XIII Jornadas de Enseñanza Universitaria de la Informática

XIII Jornadas de Enseñanza Universitaria de la Informática

Universidad de Zaragoza
Teruel, del 16 al 18 de julio de 2007

© Los autores, 2007

Portada: Fernando Naranjo y Sistemas de Impresión y Diseño – Teruel
Logo Jenui 2007: Fernando Naranjo Palomino
Maquetación: Comité de Organización

Primera edición: julio de 2007

ISBN: 978-84-9732-421-2
Depósito legal:
Impresión y encuadernación:

ACTAS DE LAS
XIII JORNADAS DE ENSEÑANZA UNIVERSITARIA
DE LA INFORMÁTICA (JENUI 2007)

ORGANIZADAS POR:

Escuela Universitaria Politécnica de Teruel, Universidad de Zaragoza

Asociación de Enseñantes Universitarios de la Informática (AENUI)

ENTIDADES PATROCINADORAS:

Teruel Digital

Fundación Universitaria Antonio Gargallo

ENTIDADES COLABORADORAS:

Thomson

Novática

COMITÉ DE PROGRAMA

PRESIDENTE

Joaquín Ezpeleta Mateo	Universidad de Zaragoza
Marian Díaz Fondón	Universidad de Oviedo
Jesús Joaquín García Molina	Universidad de Murcia
Alberto Gómez Mancha	Universidad de Extremadura
Cristóbal Pareja Flores	Universidad Complutense de Madrid
Rosalía Peña Ros	Universidad de Alcalá
Fermín Sánchez Carracedo	Universitat Politècnica de Catalunya

COMITÉ DE ORGANIZACIÓN
UNIVERSIDAD DE ZARAGOZA

PRESIDENTE DE HONOR
Felipe Pétriz Calvo
(Rector de la Universidad de Zaragoza)

MIEMBROS

Sergio Albiol Pérez
Carlos Catalán Cantero
Piedad Garrido Picazo
Raquel Lacuesta Gilaberte
Francisco J. Martínez Domínguez
Fernando Naranjo Palomino

COLABORADORES EN EL PROCESO DE REVISIÓN

Albiol Pérez, Sergio	UNIZAR
Álvarez García, Fernando	UNIOVI
Álvarez Pérez-Aradros, Pedro Javier	UNIZAR
Araque Cuenca, Francisco	UGR
Arévalo Rosado, Luis Jesús	UNEX
Aznar Gregori, Fidel	UA
Baldassarri , Sandra	UNIZAR
Bañares Bañares, José Ángel	UNIZAR
Bengochea Martínez, Luis	UAH
Blesa Pons, Pedro	UPV
Bofill Soliguer, Pau	UPC
Burgos Ortiz, José Manuel	UPM
Calvo Sánchez, Tomasa	UAH
Catalán Cantero, Carlos	UNIZAR
Compañ Rosique, Patricia	UA
De Vicente Rodríguez, Antonio J	UAH
Delgado Domínguez, Adelaida	UIB
Díaz Esteban, Alberto	UCM
Díaz Labrador, JosuKa	DEUSTO
Escribano Otero, Juan José	UEM
Ferrandez Pastor, F. Javier	UA
García Almiñana, Jordi	UPC
García Famoso, Montse	URV
García García, Eduardo	UVM
García García, María José	UEM
García López, Pedro	URV
García Mateos, Gines	UM
García Rubio, Félix Óscar	UCLM
García Zubía, Javier	DEUSTO
Garre Rubio, Miguel	UAH
Garrido Picazo, Piedad	UNIZAR
Gavaldà Mestre, Ricard	UPC
Gayo Avello, Daniel	UNIOVI
Gil Larrea, María José	DEUSTO
Giménez Canovas, Domingo	UM
Gómez Mancha, Alberto	UNEX
González Ferez, Pilar	UM
González Rodríguez, Julia	UNEX
Grau , Antoni	UPC
Grediaga Olivo, Ángel	UA
Gutiérrez Martínez, José María	UAH
Gutiérrez Pérez, Diego	UNIZAR
Jacob Taquet, Inés	DEUSTO
Juan Lizandra, M. Carmen	UPV
Julián Inglada, Vicente J.	UPV
Labra Gayo, José Emilio	UNIOVI
Lacuesta Gilaberte, Raquel	UNIZAR
Lapeña Marcos, María Jesús	UNIZAR
Lara Bercial, Pedro J.	UEM
Lemus Zúñiga, Lenin G.	UPV
Llopis Pascual, Fernando	UA
Llorens Largo, Faraón	UA
López Álvarez, David	UPC
López de Teruel Alcolea, Pedro Enrique	UM
López Nozal, Carlos	UBU
Lozano Tello, Adolfo	UNEX
Luján Mora, Sergio	UA
Marroquín Alonso, Olga	UCM
Martí Campoy, Antonio	UPV
Marticorena Sánchez, Raúl	UBU
Martínez Domínguez, Francisco José	UNIZAR
Martínez Santamarta, Margarita	URJC
Martínez Vidal, Gloria	UJI
Más Estellés, Jorge	UPV
Messeguer , Roc	UPC
Miguel Villalba, Gurutze	UEM
Miró Julià, Joe	UIB
Molina Carmona, Rafael	UA
Moncusí Mercadé, M. Angels	URV
Moreno Ribas, Antonio	URV
Naranjo Palomino, Fernando	UNIZAR
Neira Parra, José	UNIZAR
Ortego Carazo, Santiago	UPC
Otero Calviño, Beatriz	UPC
Pavón Pulido, Nieves	UHU
Pérez Poch, Antoni	UPC
Pérez Serrano, Pedro Luis	UNEX
Perles Ivars, Ángel F.	UPV
Poza Luján, José Luis	UPV
Pujol López, Mar	UA
Ramió Aguirre, Jorge	UPM
Ramos Román, Isabel	US
Riesco Albizu, Miguel	UNIOVI
Saénz Pérez, Fernando	UCM
Salguero Hidalgo, Alberto Gabriel	UGR
Sánchez Alonso, Manuel	UPCT
Sánchez Alonso, Salvador	UAH
Sánchez Carracedo, Fermín	UPC
Satorre Cuerda, Rosana	UA
Segura Díaz, Clara María	UCM
Urquiza Fuentes, Jaime	URJC
Urretavizcaya Loinaz, Maite	EHU
Vega Rodríguez, Miguel Ángel	UNEX
Villalba de Benito, María Teresa	UEM
Virgós Bel, Ferran	UPC
Zarazaga Soria, F. Javier	UNIZAR

PRESENTACIÓN

Introducción

El presente libro contiene los trabajos seleccionados para la decimotercera edición de las Jornadas de Enseñanza Universitaria de la Informática (JENUI), celebradas en Teruel los días 16, 17 y 18 de julio de 2007, y organizadas por la Escuela Universitaria Politécnica de Teruel.

Los trabajos de las jornadas se han encuadrado en un conjunto de áreas de interés, que incluyen todas las materias de informática habitualmente tratadas en estudios universitarios, así como aquéllas que inciden de una manera transversal en nuestra docencia: adaptación al espacio europeo de educación superior, atención a la diversidad, calidad y evaluación de la docencia, evaluación del alumnado, formación para la profesión, métodos pedagógicos innovadores, organización curricular y planes de estudio.

Contenidos: conferencias y otras actividades

Las Jornadas se iniciarán con la conferencia titulada "Roles emergence and balance in collaborative design", impartida por la Dra. Françoise Détienne. La profesora Détienne es la investigadora responsable del grupo EIFFEL2, perteneciente al Institut National de Recherche en Informatique et en Automatique (INRIA) de Francia. EIFFEL2 centra su investigación en las herramientas para el trabajo colaborativo ("groupware") en el ámbito de las tecnologías de la información y la comunicación. La profesora Détienne tiene publicados múltiples trabajos y diversos libros en este campo.

Como es tradicional, se han programado dos mesas redondas. La primera, titulada "Docencia sobre pruebas de software: más allá de los planes de estudio", moderada por el Dr. Faraón LLorens, de la Universidad de Alicante, debatirá sobre la importancia y el peso dado a la enseñanza de las pruebas como parte importante de los métodos de aseguramiento de calidad del "software".
La segunda, titulada "Atribuciones profesionales de los ingenieros informáticos a la vista de los nuevos títulos de grado y postgrado", moderada por el Dr. Rafael Navarro de la Universidad de Zaragoza, reunirá a representantes de la profesión informática para debatir sobre un tema que está generando una encendida controversia en el mundo de las ingenierías.

Contribuciones y proceso de revisión

Tradicionalmente, las JENUI han organizado las contribuciones en torno a ponencias y recursos docentes. En esta edición se ha añadido un tercer tipo de trabajo, los pósteres. Con esta nueva modalidad hemos querido dar cabida en JENUI a aquellos trabajos interesantes que no encajan en las categorías de ponencia o recurso docente, ya sea por su extensión o por su estado de elaboración. En un póster se pueden presentar propuestas de innovación y mejora en el desarrollo docente, proyectos que todavía no se han puesto en marcha, ejemplos de problemas y prácticas útiles, ideas y actividades para el trabajo en grupo, métodos de evaluación alternativos, herramientas y programas de apoyo a la docencia, etc. El objetivo final de un póster es fomentar la discusión y recibir realimentación de otros colegas.

La calidad y el interés de los trabajos presentados es una de las principales preocupaciones de las JENUI, siendo el proceso de revisión uno de sus pilares. Este proceso es anónimo, y guiado por unas directrices que tratan de unificar los criterios aplicados: las revisiones sirven como filtro y como orientación constructiva para mejorar distintos aspectos de los trabajos seleccionados. Las opiniones de los revisores han sido el elemento fundamental para la aceptación de un trabajo. En caso de haber diferencias importantes de criterio en las evaluaciones, se ha solicitado una revisión adicional. Es posible, seguramente inevitable, que se hayan cometido errores en este proceso, pero estamos seguros de que los trabajos aceptados son trabajos de alta calidad.

Como resultado del proceso de revisión se han aceptado el 59% de los trabajos enviados como ponencias, el 80% de los recursos docentes y el 90% de los pósteres.

No quisiéramos terminar este apartado sin mostrar de manera explícita nuestro agradecimiento a todos los revisores, sin cuyo trabajo sería imposible la realización de estas jornadas. Y de entre ellos destacar a Pau Bofill y Joe Miró por su ayuda especial para la sección de pósteres.

Publicación

Como viene siendo habitual en los últimos años, estas actas se han publicado en la editorial Thomson, lo que supone un importante incentivo para los autores de los trabajos de este congreso. Queremos por ello agradecer a la editorial Thomson y, en especial, a Andrés Otero su interés por JENUI. También queremos agradecer a la revista Novática, y su coordinador editorial Llorenç Pagés, la publicación de algunos trabajos seleccionados de entre los presentados.

Sede

La sede de JENUI 2007 es la Escuela Universitaria Politécnica de Teruel (EUPT) centro perteneciente a la Universidad de Zaragoza. Creada en 1992 imparte las titulaciones de Ingeniería Técnica de Telecomunicación (Sistemas Electrónicos) e Ingeniería Técnica en Informática de Gestión. Está ubicada en el campus de Teruel de la Universidad de Zaragoza.

Una de las mayores preocupaciones del centro ha sido desde siempre impartir una docencia de calidad. Los profesores del centro, además de su trabajo investigador y de colaboraciones con empresas, participan habitualmente en congresos y reuniones sobre la enseñanza universitaria, siendo las JENUI una de ellas. La EUPT asume con gran agrado, y a la vez responsabilidad, la organización de estas jornadas.

En los últimos años, la ciudad de Teruel está siendo objeto de profundas mejoras. Así, se está rehabilitando su centro histórico, algunas de cuyas actuaciones han sido merecedoras de premios a nivel nacional e internacional de arquitectura y urbanismo. Por otro lado, Teruel tiene muchas caras, se puede hablar del Teruel Mudéjar, declarado Patrimonio de la Humanidad por la UNESCO; del Teruel Modernista, con edificios diseñados por discípulos de Gaudí; del Teruel tierra de dinosaurios, con el reciente descubrimiento de una nueva especie de saurópodo en Riodeva, o el parque temático Dinópolis. Y todo sin olvidar que el entorno de Teruel también es digno de visitar, con Albarracín y su comarca, con la sierra de Gúdar-Javalambre, con el Maestrazgo, etc.

JENUI 2008

La XIV edición de la jornadas tendrá lugar en la Universidad de Granada. Los detalles serán accesibles en la dirección https://JENUI2008.ugr.es, siendo la dirección de contacto jenui2008@atc.ugr.es.

Conclusión

Esperamos que estas actas os resulten de provecho, que disfrutéis de estos días y que les saquéis a estas Jornadas el mayor partido en lo personal y en lo profesional.

Teruel, 16 de julio de 2007
Comité de Programa y Comité Organizador

CONTENIDO

Conferencia

Roles emergence and balance in collaborative design **5**
Françoise Détienne
Responsable scientifique de l'équipe EIFFEL2(INRIA)

Ponencias

Adaptación al espacio europeo de educación superior.....9

Diseño integral de una asignatura para una formación basada en competencias.....11
Francisco Ruiz, Félix García
Universidad de Castilla-La Mancha

La enseñanza de Estructura de Computadores en el EEES 19
Fermín Sánchez, Agustín Fernández, Josep Llosa
Universitat Politècnica de Catalunya

El idioma vehicular docente; un reto a la movilidad en el Espacio Europeo de Enseñanza Superior (EEES) 27
Germán Santos Boada, Josep Solé Pareta
Universidad Politécnica de Cataluña

Cómo congeniar los exámenes y los proyectos en asignaturas PBL 33
Pablo del Canto, Isabel Gallego, Rubén Hidalgo, Johann López, José Manuel López, Javier Mora, Eva Rodríguez, Eduard Santamaría, Miguel Valero
Universitat Politècnica de Catalunya

Cómo transformar e impartir una asignatura bien adaptada a ECTS sin morir en el intento: patrones para la reducción del trabajo del profesor.....41
Ray Fenández Rupérez, Miren Bermejo Llopis
Universidad del País Vasco

Evaluación formativa de capacidades transversales en el marco EEES 49
Ferran Virgós Bel(1), Joan Segura Casanovas(2), Edmundo Tovar(3)
Universitat Politècnica de Catalunya(1,2) , Universidad Politécnica de Madrid(3)

Experiencias en un plan piloto de adaptación al EEES: Hoja de ruta para coordinadores 57
Jordi Freixenet, Josep Ll. Marzo, Josep Soler
Universitat de Girona

Metodología de trabajo y experiencias de aprendizaje colaborativo y evaluación continua en la disciplina de Sistemas Multiagente 65
Francisco Bellas Bouza, Amparo Alonso Betanzos
Universidad de A Coruña

Experiencia en la adaptación de la asignatura Sistemas Operativos I al EEES 73
José Antonio Gómez, Buenaventura Clares
Universidad de Granada

Análisis de Herramientas y Estrategias para las Nuevas Titulaciones en Informática 81
Antonio Polo Márquez, Jorge Martínez Gil, Luis J. Arévalo Rosado
Universidad de Extremadura

Estudio de la carga de trabajo en asignaturas con un gran número de alumnos 89
Joaquín Gracia Morán, María A. Pinar Sepúlveda
Universidad Politécnica de Valencia

FAQs sobre la adaptación de las asignaturas al EEES 97
Miguel Valero-García, Juan J. Navarro
Universidad Politécnica de Cataluña

La agenda de aprendizaje como herramienta de adaptación a la metodología educativa en ECTS del EEES..... 105
Lluís Ribas Xirgo, A. Josep Velasco González
Universitat Autònoma de Barcelona

Una aproximación al Espacio Europeo de Educación Superior basada en el desarrollo de proyectos software en Ingeniería del Conocimiento..... 113
Bertha Guijarro Berdiñas, Amparo Alonso Betanzos
Universidad de A Coruña

Una experiencia educativa en la asignatura de Estadística impartida en la Ingeniería Técnica Informática de la Universidad Politécnica de Valencia 121
Ángeles Calduch, Rosa Mª Alcover, Rosa Albert
Universidad Politécnica de Valencia

Estudio de Adecuación al Espacio Europeo de Educación Superior en Informática..... 129
E. Ferrando Julià[(1)], V. Brotons Gil[(1)], D.Ruiz Fernández[(1)], M. Moreno Rando[(2)]
Universidad de Alicante[(1))], Instituto de Enseñanza Secundaria Mare Nostrum[(2)]

Robótica e informática industrial/arquitectura de ordenadores..... 137

Una experiencia práctica de aprendizaje basado en proyecto en una asignatura de robótica 139
Nourdine Aliane, Sergio Bemposta, Javier Fernández, Verónica Egido
Universidad Europea de Madrid

Análisis de tecnologías sw para laboratorios remotos..... 145
Javier García Zubía, Pablo Orduña, José María Sáenz Ruiz de Velasco,
Inés Jacob Taquet, Jesús Luis Díaz Labrador, Javier Oliver Bernal
Universidad de Deusto

SWECAI: Sistema web centrado en el alumno inteligente 153
Juan Julián Merelo Guervós[(1)], Carmel Hassan-Montero[(1)], Fernando Tricas [(2)],
Juan Luis Fernández Laredo[(1)]
Universidad de Granada[(1)], Universidad de Zaragoza[(2)]

Evaluación del alumnado **161**

Análisis del rendimiento académico en los estudios de informática de la Universidad Politécnica de Valencia aplicando técnicas de minería de datos 163
R. Alcover, J. Benlloch, P. Blesa, M. A. Calduch, M. Celma, C. Ferri, J. Hernández-Orallo, L. Iniesta, J. Más, M. J. Ramírez-Quintana, A. Robles, J. M. Valiente, M. J. Vicent, L. R. Zúnica
Universidad Politécnica de Valencia

Evaluación continuada sin morir en el intento 171
David López, Alex Pajuelo, José R. Herrero, Alejandro Duran
Universidad Politécnica de Cataluña

Hacia la Evaluación Continua Automática de Prácticas de Programación 179
Juan Carlos Rodríguez del Pino, Margarita Díaz Roca, Zenón Hernández Figueroa, José Daniel González Domínguez
Universidad de Las Palmas de Gran Canaria

Predicción del rendimiento de los estudiantes y diagnóstico usando redes neuronales 187
Zulma Cataldi, Fernando Salgueiro, Fernando Javier Lage
Universidad Tecnológica Nacional (Facultad Regional Buenos Aires)

Formación para la profesión **195**

Una e-revista donde publiquen los estudiantes de Informática 197
José M. Claver[1], Germán Fabregat[2]
Universitat de Valencia[1], Universitat Jaume I[2]

Expectativas laborales de nuestros alumnos, ¿debemos adaptarnos? 205
Julia González[1], José Luis Sánchez-Carrasco[2], Carolina González[3]
Universidad de Extremadura[1], Grupo Extela[2], Junta de Extremadura[3]

Desarrollo del proyecto de reciclaje de equipos informáticos en el marco de Teruel Digital 213
Francisco Martínez, Fernando Naranjo, Piedad Garrido
Universidad de Zaragoza

Informática en otras carreras **221**

Desarrollo de docencia universitaria de informática documental con herramientas de software libre 223
Jesús Tramullas Saz, Piedad Garrido Picazo
Universidad de Zaragoza

¿Aprendizaje Basado en Proyectos? ¡Pero si mi carrera no es técnica! 231
Francisco J. Gallego Durán, Faraón Llorens Largo
Universidad de Alicante

Métodos pedagógicos innovadores.....239

Mejora del sistema Moodle mediante personalización de contenido y generación evolutiva de actividades aleatorias.....241
Antonio Javier Gallego, José Requena, Mar Pujol, Javier Montoyo
Universidad de Alicante

Aplicación del portafolio como estrategia de evaluación formativa 249
María A. Pinar Sepúlveda, Joaquín Gracia Morán
Universidad Politécnica de Valencia

Aplicación del método del caso a una parte de la asignatura Redes de Computadores.....257
Guillermo Azuara Guillén
Universidad de Zaragoza

La relevancia de los documentos en Google: Una actividad común de las asignaturas Programación y Computación Numérica 263
Assumpció Casanova, Juan Garayoa, Francisco Marqués, Vicente Vidal
Universidad Politécnica de Valencia

Hacia un Sistema Inteligente basado en Mapas Conceptuales Evolucionados para la Automatización de un Aprendizaje Significativo. Aplicación a la Enseñanza Universitaria de la Jerarquía de Memoria 269
L. Moreno, E.J. Gonzalez, J.D. Piñeiro, B. Popescu, A. Hamilton, J. Sigut, J. Torres, J. Toledo, J. Merino, C. González
Universidad de La Laguna

La revisión entre iguales como herramienta de aprendizaje y evaluación en la asignatura de sistemas operativos.....277
Miguel Riesco Albizu, Marián Díaz Fondón
Universidad de Oviedo

El Aprendizaje Basado en Proyectos como modelo docente. Experiencia interdisciplinar y herramientas Groupware. 285
Juan R Reverte Bernabeu, Antonio Javier Gallego Sánchez, Rafael Molina Carmona, Rosana Satorre Cuerda.
Universidad de Alicante

Una experiencia docente orientada a incrementar el trabajo personal del estudiante.....293
Beatriz Otero Calviño, Jaume Martí-Farré, Ernest Garriga Valle, Arantxa Alonso Maleta, Lluis Prat Viñas
Universidad Politécnica de Cataluña

Uso de herramientas colaborativas que reducen la carga de gestión en la docencia. 301
Juan José Escribano Otero, Enrique Puertas Sanz, César Alberto Escribano Otero
Universidad Europea de Madrid

Organización curricular y planes de estudio 309

¿Podemos darle la vuelta a la enseñanza del desarrollo del software? 311
Josep M. Marco-Simó, Isabel Guitart Hormigo, M. Jesús Marco-Galindo,
Àngels Rius Gavidia, M.Elena Rodríguez González
Universitat Oberta de Catalunya

La influencia de la imagen social de la profesión en las titulaciones universitarias de informática 319
Luis Fernández
Universidad Europea de Madrid

La Informática Médica en el Espacio Europeo de Educación Superior 327
J. Martín Moreno, V.J. Berenguer Miralles, D. Ruiz Fernández, A. Soriano Payá, D. Gil Méndez
Universidad de Alicante

Programación, algoritmos y estructuras de datos 335

Metodología para la enseñanza aprendizaje de la lógica de la programación orientada a objetos.....337
Leobardo López Román
Universidad de Sonora

Adoptando el Paradigma de la Programación Orientada a Atributos 345
Raúl Marticorena Sánchez, Carlos López Nozal, Carlos Pardo Aguilar
Universidad de Burgos

La práctica monitorizada: una herramienta válida en el aprendizaje activo de la asignatura Estructuras de la Información 353
Gabriel Fiol Roig, Catalina Fiol Roig, Margarita Miró Julià
Universidad de las Islas Baleares

Recursos docentes 361

Herramientas para la creación de tutoriales inteligentes en un entorno de teledocencia utilizando redes bayesianas.....363
R. Laza, R. Pavón, D. Glez-Peña, J. R. Méndez, F. Fdez-Riverola
Universidad de Vigo

ALT: Algorithm Learning Tool 371
R. Laza, D. Glez-Peña, J. R. Méndez, F. Fdez-Riverola, J. Baltasar García, M. Reboiro
Universidad de Vigo

Una herramienta para la enseñanza y aprendizaje de lenguajes formales y teoría de autómatas.....379
Jose Jesus Castro-Schez, Ester del Castillo, Julian Hortolano
Universidad de Castilla-La Mancha

Herramienta para la gestión de software y enlaces de Investigación Operativa 387
Victoria Fernández, Ana Zelaia, Pablo Aramendia
Universidad del País Vasco

Simuladores de Jerarquía de Memoria en el Contexto de un Proceso de Investigación-Acción 393
L. Moreno, E.J. González, B. Popescu, J. Torres, J. Toledo, C. González
Universidad de La Laguna

PROIEKGES: Gestión y Planificación Estratégica de Proyectos Fin de Carrera.....401
Jose Antonio Lozano Alonso , Miren Bermejo Llopis, Julián Gutiérrez Serrano
Universidad del País Vasco-Euskal Herriko Unibertsitatea

AIBench: framework de programación para la enseñanza de técnicas de IA en tercer ciclo 409
R. Domínguez, D. Glez-Peña, J. R. Méndez, F. Fdez-Riverola
Universidad de Vigo

Invocación de Métodos Remotos: prácticas de laboratorio.....417
Coromoto León Hernández, Gara Miranda Valladares
Universidad de La Laguna

Enseñanza asistida de teoría de autómatas y lenguajes formales mediante el uso de THOTH.....425
César García Osorio, Andrés Arnáiz Moreno, Älvar Arnáiz González
Universidad de Burgos

DES: un recurso para el aprendizaje de bases de datos deductivas 433
Fernando Sáenz Pérez
Universidad Complutense de Madrid

Generador-corrector de problemas de la Unidad de Control.....441
Antonio J. de Vicente, Manuel Prieto, Rafael Fernández
Universidad de Alcalá

El Control de Versiones en el aprendizaje de la Ingeniería Informática: Un enfoque práctico 447
Fran J. Ruiz-Bertol, Francisco Javier Zarazaga-Soria
Universidad de Zaragoza

ECTS y Moodle: Guía docente y aplicación práctica.....455
Pascual Timor Hermano, Pablo Moreno Meseguer, Jordi Forcada Cerdà
Escola Superior de Cerámica de Manises, Escola D'Art i Superior de Disseny de Castelló, Escola D'Art i Superior de Disseny d'Alcoi

Tutorial gráfico interactivo de estructuras de datos.....461
Mariano J. Cabrero Canosa, Javier Blanco Gutiérrez
Universidade da Coruña

Calidad y evaluación de la docencia 469

Doce Propuestas y una Reflexión 471
Rosana Satorre[(1)], Faraón Llorens[(1)], Pere Palmer[(2)], Joe Miró[(2)]
Universidad de Alicante[(1)], Universitat de les Illes Balears[(2)]

¿Compensa el Esfuerzo Realizado al Aplicar Técnicas de Aprendizaje Activo?.....479
Carlos E. Vivaracho Pascual, Mª Aránzazu Simón Hurtado, Alejandra Martínez Monés, Noelia de las Heras Osorno.
Universidad de Valladolid

Un estudio sobre el absentismo y el abandono en asignaturas de programación.....487
Agustín Cernuda del Río, Sonia Hevia Vázquez, María del Carmen Suárez Torrente, Daniel Gayo Avello
Universidad de Oviedo

¿Cómo cambia a los alumnos la asignatura de Gestión de Sistemas Informáticos?. 495
Daniel González Morales, José Luis Roda García, Luz Marina Moreno de Antonio
Universidad de La Laguna

Trabajos fin de carrera, practicum y participación de alumnos en la investigación 503

La automatización de un laboratorio de ingeniería química para uso docente:
una plataforma multidisciplinar para la realización de trabajos de fin de carrera e investigación.....505
Gerard Escudero, Moisès Graells, Josep M. Guerrero, Sebastián Tornil
Universitat Politècnica de Catalunya

Hacia una metodología para el desarrollo de trabajos y Proyectos Fin de Carrera en
Ingeniería Informática.....511
Antonio Polo Márquez, Jorge Martínez Gil, Luis J. Arévalo Rosado
Universidad de Extremadura

PFC: Los dos lados del espejo. Proyectista – Director ¿Una visión compartida? 519
Josep Fernández(1), Antonio B. Martínez(1), Raul Marín(2), Juan Pablo Arboleda(1)
Universitat Politècnica de Catalunya(1), Universitat Jaume I(2)

Atención a la diversidad 527

El estilo de aprendizaje como característica distintiva entre alumnos: un enfoque estadístico 529
Carlos T. Calafate, Juan-Carlos Cano, Pietro Manzoni
Universidad Politécnica de Valencia

Adaptación de contenidos docentes informáticos para alumnos sordos: una propuesta para abordar
esta problemática 537
Fco. Javier Bueno Guillén, Soledad García, José Raúl Fernández del Castillo, Jaime Oyarzo Espinosa
Universidad de Alcalá (Centro de Recuperación de Minusválidos Físicos)

Pósteres Sesión Abierta. 543

7 minutos para un objetivo transversal: ¡L E A A T E N T A M E N T E! 545
Rosalía Peña y Miguel Garre
Universidad de Alcalá de Henares

Una herramienta para la evaluación automatizada de la disciplina de Bases de Datos 547
Piedad Garrido Picazo, Gabriel Fuertes Muñoz, Jesús Tramullas Saz
Universidad de Zaragoza

Enseñanza de la Informática en el marco de la Educación Musical.....549
M. Isabel Dorta González, Pablo Dorta González
Universidad de La Laguna

Experiencia en Proyectos Fin de Carrera de cooperación con países en vías de desarrollo. 551
Javier Alonso, David López, Javier Larrosa
Universidad de Politécnica de Cataluña.

Iniciativas para la adaptación al EEES de la asignatura de Fundamentos Físicos de la Informática.....553
Esmeralda Úbeda, Josep-Abel González
Universidad de Girona

Una invitación a la participación de la mujer en Ingeniería Informática.....555
Itziar Cortés, Victoria Fernández, Edurne Larraza, Montse Maritxalar, Kepa Sarasola, Aitziber Zumarraga
Universidad del País Vasco UPV/EHU

Pósteres Sesión Especial. 557

Las fases del aprendizaje: Un esquema para el análisis y diseño de actividades de enseñanza/aprendizaje.....559
Pau Bofill, Joe Miró
Universitat Politécnica de Catalunya, Universitat de les Illes Balears

Las fases del aprendizaje como pauta para el diseño de un libro interactivo de Introducción a los Ordenadores 561
Beatriz Otero, Pau Bofill
Universitat Politécnica de Catalunya

Fomento de la Motivación y uso de mapas conceptuales en el primer día de clase 563
Luis J. Saiz Adalid, Joaquín Gracia Morán
Universidad Politécnica de Valencia

Diseño y evaluación de actividades de aprendizaje para un curso de introducción a la programación ..565
Pedro J. Clemente, Alberto Gómez, Julia González, Roberto Rodríguez, Encarna Sosa
Universidad de Extremadura

Índice de Autores 567

Conferencia

Roles emergence and balance in collaborative design

Roles emergence and balance in collaborative design

Françoise Détienne
Eiffel2 Team, INRIA
Domaine de Voluceau, Rocquencourt, 78153 Le Chesnay, France
Francoise.Detienne@inria.fr

In this talk we will present and discuss a framework based on forms of participation in collaborative design through the concept of "role", that considers the participants' activities on a collective level. We consider roles as phenomena that emerge from the interaction between design stakeholders rather than institutionally given.

In our previous studies (Baker et al. 2007; Barcellini et al. 2006; D'Astous et al. 2001; Détienne, 2006; Détienne et aL 2003; Sack et al. 2006) we have examined roles along several dimensions: epistemic and cognitive, discursive and interactional, social and institutional. We have analysed role emergence in the dynamic of design. We will illustrate our framework by studies in architectural design and software design conducted in various spatio-temporal settings (co-located meetings and distant asynchronous technology-mediated situations) and in various socio-organisational settings (traditional software organisation and open-source software communities).
We will discuss our framework of « role emerging design » with respect to the more traditional framework of « participatory design » and discuss about socio-technical environments enabling role emergence and role balance, constituting this way "enabling environments" (Falzon, 2005) for participants.

Baker, M., Détienne, F., Lund, K., & Séjourné, A. (2007) Etude des profils interactifs dans la conception collective en architecture. In F. Détienne, V. Traverso (Eds) : Méthodologies d'analyse de situations coopératives de conception, Nancy : PUN

Barcellini, F., Détienne, F., Burkhardt, J.M., Sack, W. (2006) Visualizing Roles and Design Interactions in an Open Source Software Community. In Workshop on Supporting large scale software development at CSCW' 06. 4-8 November 2006. Banff, Alberta, Canada.

D'Astous, P., Détienne, F., Robillard, P. N., & Visser, W. (2001) Quantitative measurements of the influence of participants roles during peer review meetings. Empirical Software Engineering, 6, 143-159.

Détienne, F. (2006) Collaborative design : managing task interdependencies and multiple perspectives. Interacting With Computers. 18(1), 1-20.**

Détienne, F., Burkhardt, J-M., & Visser, W. (2003) Cognitive effort in collective software design: methodological perspectives in cognitive ergonomics. in « Proceedings of the 2nd Workshop in the Workshop Series on Empirical Software Engineering "The Future of Empirical Studies in Software Engineering" », pages 17-25, Monte Porzio Catone (Rome, Italy), 29 September, 2003.

Falzon, P. (2005). Ergonomics, knowledge developpement and the design of enabling environments. In Conference on Humanizing Work and Work Environment, Guwahati, Inde.

Sack, W., Détienne, F., Ducheneaut, N., Burkhardt, J-M., Mahendran, D., & Barcellini,F., (2006) A methodological framework for socio-cognitive analyses of collaborative design of Open Source Software. Computer Supported Cooperative Work (CSCW), the Journal of Collaborative Computing, 15 (2-3), 229-250.

Ponencias

Adaptación al espacio europeo de educación superior

Diseño integral de una asignatura para una formación basada en competencias

Francisco Ruiz, Félix García
Dpto. de Tecnologías y Sistemas de Información
Escuela Superior de Informática, Universidad de Castilla-La Mancha
Paseo de la Universidad, 4, 13071 Ciudad Real
{Francisco.RuizG, Felix.Garcia}@uclm.es

Resumen

En esta ponencia se presenta una experiencia de adaptación de una asignatura a los principios y recomendaciones del EEES, siguiendo los paradigmas de enseñanza centrada en el aprendizaje del alumno y de formación basada en competencias. Se ha ideado y aplicado un método de diseño integral *top-down*, definiendo los elementos del modelo formativo de la asignatura desde lo general a lo concreto: objetivos y competencias, estructura general de contenidos, tipos de actividades de enseñanza-aprendizaje, esfuerzo del alumno, calendario de actividades y sistema de evaluación continua. También se comentan los resultados del primer año de experiencia, con una mejora en el rendimiento académico y mayor satisfacción de los alumnos.

1. Introducción

Entre los objetivos del "Espacio Europeo de Educación Superior" (EEES) cabe destacar el cambio de paradigma educativo que se propone. Dicho cambio se resume en la célebre frase de "*pasar de una educación basada en la enseñanza a otra basada en el aprendizaje*". Detrás de esta frase hay mucho mensaje: enseñar es el esfuerzo que realiza el profesor mientras que aprender es el esfuerzo que realiza el alumno. Por tanto, se pretende cambiar todo el sistema docente para que el centro de todo el proceso sea el esfuerzo del alumno por aprender en vez del trabajo de enseñar del profesor. Este cambio tiene múltiples consecuencias; desde las meramente docentes (manera de actuar el profesor en clase, nuevas actividades académicas) hasta otras de tipo administrativo (se debe valorar el esfuerzo del alumno y no sólo lo que demuestra que sabe en una o varias pruebas). Lo que está claro es que dicho cambio, en el caso de la Universidad española, es bastante revolucionario ya que implica un cambio muy grande en los roles habituales que tenemos los profesores y alumnos: el profesor pasa de ser un "transmisor de conocimientos" a un guía que ayuda a aprender, mientras que el alumno tiene un papel más central y activo, ya que es su esfuerzo de aprendizaje la clave de todo.

En nuestra opinión, la mayor dificultad a la que se enfrenta la universidad española en este cambio es que para implantar con éxito esta nueva modalidad del proceso de enseñanza-aprendizaje es necesario un planteamiento global, de forma que los problemas, objetivos y planes deben ser pensados con una perspectiva holística, que engloba a todas las dimensiones de una manera integral. Por el contrario, los profesores (y también los alumnos y el PAS) estamos acostumbrados a pensar y funcionar en parcelas (nuestra asignatura, nuestro examen, etc.).

Por otro lado, aunque los profesores y alumnos estemos dispuestos a llevar a cabo el esfuerzo para implantar tal cambio, en la actualidad existe un impedimento de tipo legal que hace imposible materializar, de verdad, el nuevo paradigma: mientras no cambie la legislación de planes de estudios, no es posible reformarlos para que reflejen la aplicación plena del nuevo paradigma educativo. Donde más se hace sentir esta restricción es en que seguimos con unas normas que obligan a que todo se maneje a nivel de asignaturas sueltas (un examen es de una asignatura, una clase es de una asignatura, ..), mientras que con el nuevo paradigma se deberían poder establecer mecanismos transversales (horizontales) en los planes de estudios, tanto a nivel de formación, como de aprendizaje y evaluación. Por ejemplo, en las titulaciones de Informática, se podría "imitar" la manera en que se trabaja en el mundo real de las empresas desarrollando la formación y evaluación de un grupo de asignaturas relacionadas (por ejemplo, bases de datos, ingeniería del software y gestión

de proyectos) mediante un aprendizaje basado en un proyecto común que los alumnos desarrollarían en grupos reducidos, bajo la guía de los profesores a lo largo de 2 o más semestres. O en lugar de un examen para cada asignatura de programación, se podría establecer una prueba integral que evaluara la capacidad global del alumno para programar computadoras y que supusiera la consecución de todos los créditos correspondientes.

En consecuencia, mientras no se produzca dicho cambio legislativo, en el mejor de los casos, sólo cabe hacer experiencias a la fuerza parciales, pero que nos permiten a los profesores ir probando y aprendiendo, es decir, prepararnos de cara a este futuro (y ahora parece que próximo) cambio. En nuestra opinión, estas experiencias se pueden clasificar en dos categorías:

a) Aplicación de alguna técnica educativa o pedagógica concreta; o
b) Aplicación del nuevo paradigma educativo de forma integral, aunque restringido en su alcance a una asignatura o curso (por el impedimento legal ya comentado).

Desde hace algunos años, en las universidades españolas se están desarrollando múltiples experiencias, pero la gran mayoría son de la primera categoría. Así, en diversos foros especializados, como las JENUI, o en eventos organizados de forma expresa[1], es posible encontrar experiencias sobre aprendizaje basado en problemas, formación basada en competencias, evaluación por proyectos, aprendizaje cooperativo o estimación del esfuerzo del alumno.

En cambio, son muy pocas las experiencias que se pueden encuadrar en la segunda categoría. De hecho, no conocemos ninguna que cumpla las características básicas de la aquí presentada (ver sección 3). Por razones de espacio, en esta ponencia nos centramos en explicar el método de trabajo y el diseño realizado, y sólo se presentan brevemente los resultados y lecciones aprendidas, cuyo detalle queda para otra ponencia posterior. El resto del artículo está organizado de la siguiente manera: la sección 2 explica el contexto (centro, titulación, asignatura, alumnos) en que se ha llevado a cabo la experiencia y su motivación. La sección 3 presenta el método seguido, mientras que en la 4 se muestra el diseño de asignatura obtenido al aplicarlo. La sección 5 continúa con un resumen de los resultados en el primer año. Por último, en la sección 6 se exponen algunas conclusiones, lecciones aprendidas y trabajo futuro.

2. Contexto de la experiencia

La Universidad de Castilla-La Mancha (UCLM) está siendo conservadora en cuanto a la implantación del EEES. En esta línea, hasta el curso 2005/2006 en la Escuela Superior de Informática de Ciudad Real tan sólo se había procedido a realizar experiencias piloto aisladas en algunas asignaturas optativas de segundo ciclo de la titulación de Ingeniería en Informática. A partir de septiembre de 2005 se decide ampliar a asignaturas obligatorias, incluida "Planificación y Gestión de Sistemas de Información", cuyas características básicas, antes del cambio, se resumen en la tabla 1.

Carácter:	obligatoria
Temporalidad:	4º curso, 2º semestre
Créditos:	9 antiguos
Alumnos:	92 (2004-2005)
Grupos de clase:	1 de teoría, 3 de laboratorio
Resultados académicos:	49% aptos, 11% suspensos, 40% no presentados
Objetivos:	Formar en planificación, gestión y seguimiento de proyectos de sistemas de información, utilizando o desarrollando técnicas específicas. Conocer las herramientas software para dichas tareas.

Tabla 1. La asignatura PGSI antes del cambio

La propuesta de implantar el nuevo sistema en PGSI partió de los propios profesores y no se debió a que fuéramos unos entusiastas del cambio (en otras asignaturas que impartimos seguimos esperando) sino a que la situación nos tenía muy descontentos. Teníamos una asignatura que era muy de contar en clase, dar documentación y el alumno adoptar una aptitud muy pasiva. Habíamos fracasado en diversos intentos previos de cambiar la situación con retoques puntuales y habíamos llegado a un punto en el que los profesores nos aburríamos en clase y, por tanto, hacíamos que los alumnos se aburrieran. Los alumnos que participaban y asistían a clase eran

[1] Jornadas Nacionales de Intercambio de Experiencias Piloto de Implantación de Metodologías ECTS, Badajoz, sep-2006 (http://sofd.unex.es/jornadas/index.htm).

menos de la mitad y los resultados académicos eran pobres (menos del 50% de aprobados). Después de un período de reflexión llegamos a la conclusión de que teníamos que hacer algo en serio para cambiar y nos pusimos a estudiar durante varios meses documentos y experiencias sobre el EEES, con el objetivo de cambiarle la cara a la asignatura. Queríamos conseguir que fuese una asignatura interesente para los alumnos porque realmente aprendieran a planificar y a gestionar proyectos, y no se centraran en "empollar" apuntes llenos de literatura.

3. Método de trabajo

Una decisión clave fue seguir la recomendación del EEES de orientar los estudios universitarios hacia una formación basada en **competencias** [7] [2]. Esto nos planteó un problema: antes de poder empezar a trabajar a nivel de asignatura o curso concreto, es necesario conocer bastantes cosas de carácter más global: metas y objetivos de la titulación, perfil académico y profesional del centro y, por supuesto, competencias profesionales de los egresados. Además, se debe disponer de un nuevo tipo de plan de estudios que integre todo lo anterior definiendo un programa formativo completo: objetivos, competencias, contenidos, modalidades de actividades de enseñanza-aprendizaje, sistemas de evaluación globales, etc.

Obviamente, el contexto nuestro no era ni es el anterior. El dilema lo resolvimos "simulando", en cuanto tuviera que ver con nuestra asignatura, cómo creemos que podría/debería ser el nuevo programa formativo basado en competencias. Nuestra principal referencia fue el informe "Adaptación de los Planes de Estudio al Proceso de Convergencia Europea" [5]. Aplicando sus indicaciones, seguimos un enfoque *top-down*, abordando los diferentes aspectos a considerar desde lo más general a lo más concreto. En dicho informe se establecen las siguientes etapas para diseñar un plan de estudios adaptado al EEES y basado en competencias [3]:

a) Delimitación de los objetivos y el perfil académico y profesional de la titulación.
b) Establecer la estructura organizativa del plan así como las competencias y los contenidos propios del mismo.
c) Concretar las modalidades del proceso de enseñanza-aprendizaje para el desarrollo metodológico del programa formativo.
d) Efectuar una previsión sobre los recursos humanos y materiales necesarios.
e) Especificar los requisitos administrativos que regulan la gestión de todos los aspectos implicados en las enseñanzas del plan.

Basándonos en lo anterior y realizando una adaptación motivada por las limitaciones ya comentadas, ideamos el siguiente método de trabajo para el diseño de asignaturas guiadas por competencias:

1. Identificar **competencias**, (en base al libro blanco y otras propuestas curriculares).
2. Establecer la **estructura general** de los **contenidos,** (establecer unidades docentes compuestas de 1-n temas)
3. Definir los **tipos de actividades** de enseñanza-aprendizaje, obteniendo una lista de los tipos de actividad adecuados y su importancia (peso). El peso de cada tipo se determina en función de las competencias y los contenidos.
4. Estimar el **esfuerzo del alumno**, es decir, horas necesarias para cada tipo de actividad en cada contenido (tema o práctica).
5. Elaborar el **calendario**, indicando cada día o semana los tipos de actividad y contenidos que se trabajarán.
6. Definir un método de **evaluación continua**, basado en un sistema de puntos acumulativos; de forma que el alumno conozca desde el principio la lista de indicadores (maneras de conseguir puntos) y a lo largo del curso vaya conociendo de forma continua su evolución (puntos acumulados hasta la fecha).

4. Diseño de la asignatura

A continuación se describen con mayor detalle los resultados que obtuvimos al aplicar cada uno de los pasos del método anterior, es decir, se presenta el diseño o modelo de asignatura que hemos utilizado durante el curso 2005-2006.

---

[2] En nuestra opinión sería más correcto hablar de formación guiada por competencias.

[3] Los autores tienen una presentación detallada de este método de adaptación de planes de estudios (http://alarcos.inf- cr.uclm.es/per/fruiz/conf/eees/eees.htm).

4.1. Competencias

Para identificar las competencias tuvimos en cuenta el papel que juega la asignatura de PGSI en la formación profesional de un Ingeniero en Informática. Para ello analizamos las siguientes fuentes: Libro blanco de la titulación [3], currículos internacionales de ACM [1], SWEBOK [6] y las propuestas profesionales PAFET [2] y *Career Space* [4]. Como resultado, se estableció como objetivo de la asignatura contribuir a que el alumno adquiera las competencias profesionales, tanto transversales (generales para diversos tipos de profesionales) como específicas de la Ingeniería Informática, incluidas en la tabla 2.

Competencias	
Transversales	**Específicas**
Capacidad de organización y planificación	Conocimientos en Planificación, estrategia y organización empresarial (en cuanto a sistemas y tecnologías de información)
Comunicación oral y escrita	
Capacidad de gestión de la información (captación y análisis de la información)	
Toma de decisiones	Conocimientos en Dirección, planificación y gestión de proyectos informáticos
Trabajo en equipo	
Habilidades de relaciones interpersonales	
Razonamiento crítico	Documentación técnica
Compromiso ético	Empleo de herramientas informáticas
Aprendizaje autónomo	

Tabla 2. Competencias de la asignatura PGSI

Todas y cada una de estas competencias tienen reflejo real en alguna actividad y/o contenido de la asignatura. Por ejemplo, la "toma de decisiones" y las "habilidades de relaciones interpersonales" se trabajan con el mecanismo de reparto de puntos de los trabajos: los mismos alumnos miembros de cada grupo son los que deben negociar para determinar el reparto de los puntos de cada trabajo en base al esfuerzo que realizó cada uno y otras circunstancias que consideren.

4.2. Estructura de contenidos

Los contenidos deben estar basados en las competencias que se desean adquirir y, por supuesto, en las indicaciones legales en vigor. La tabla 3 muestra la estructura general y el peso de los contenidos que consideramos adecuados para alcanzar los objetivos de la asignatura. Información mucho más detallada puede conseguirse en la web de la asignatura (http://alarcos.inf-cr.uclm.es/doc/pgsi/).

4.3. Actividades de enseñanza-aprendizaje

El centro estableció que a la asignatura reformada de PGSI le correspondían 7'5 créditos ECTS (prorrateo de los créditos antiguos del 4º curso para obtener el equivalente a 60 ECTS al año). Esto equivale a 200 horas de esfuerzo total (*workload*) del alumno. Del análisis de las competencias buscadas, los contenidos a impartir, y las recomendaciones que al respecto incluye el Libro blanco [3], se decidió establecer los tipos de actividades que se muestran en la tabla 4. Dicha tabla también muestra la importancia que se asignó a cada tipo de actividad, es decir, el reparto general de las 200 horas. Respecto del sistema antiguo, las horas de aula normal se reducen considerablemente porque ahora esas mismas horas se reparten entre clases (magistrales-pasivas o de debate/dudas-activas), tutorías en grupos reducidos, exámenes parciales y otras actividades (seminarios, conferencias, visitas de expertos, ..).

TEORÍA (65%)	
I. Planificación de Sistemas de Información (25%)	Empresa, TI y SI. Planificación estratégica. Planificación de SI y TI
	La Información es un activo fundamental de las organizaciones
II. Gestión de Proyectos Informáticos (65%)	Fundamentos de Gestión de Proyectos. Técnicas para proyectos informáticos: integración y alcance, tiempos, riesgos, costes
	¿Cómo hacen los proyectos los 'ingenieros clásicos'? Nuestras herramientas de trabajo
III. Aspectos Sociales y Humanos de la Informática (10%)	*Realidad ≈ Algoritmos + Sentimientos*
LABORATORIO (35%)	
IV. Herramientas generales – MS Project (70%)	
V. Herramientas específicas – USC COCOMO (30%)	

Tabla 3. Contenidos de PGSI rediseñada

Tipo de Actividad	Peso	Horas
Clases Magistrales	11%	22
Clases de Debate/Dudas	6%	12
Tutorías en grupo e individuales	4%	8
Estudio	31,5%	63
Trabajos y Ejercicios	44%	88
Exámenes	2,5%	5
Otras	1%	2
Total	100%	200

Tabla 4. Tipos de actividades y su importancia

Como se aprecia en la tabla anterior, desde el punto de vista del alumno su participación y esfuerzo se dedica, por orden de prioridad, a:

- **Trabajos y ejercicios**. Al centrarse los objetivos académicos en la adquisición de competencias (saber hacer), los esfuerzos en los cuales el alumno aprende haciendo, son los más valorados. Todos los trabajos de la asignatura son llevados a cabo en grupos de 4 a 6 personas. Los alumnos eligen un caso real (una empresa o institución), que recibe el visto bueno del profesor. Primero realizan el plan de sistemas y tecnologías de información de dicha empresa (trabajo 1 de teoría), que debe incluir un proyecto de desarrollo de un nuevo software; y después desarrollan el plan para dicho proyecto software: integración y alcance del proyecto en el trabajo 2, planificación de riesgos en el 4 y estimación del tamaño en el 5. En paralelo, se realizan los dos trabajos de laboratorio que complementan lo anterior: implementar con MS Project el plan del proyecto y simular su seguimiento, y emplear los resultados del trabajo 5 de teoría como entrada para estimar el esfuerzo con la herramienta USC. COCOMO II. También se consiguen puntos por trabajos adicionales individuales, cuya temática es abierta (pero relacionada con la asignatura), y por realización y entrega individualizada de los ejercicios propuestos por el profesor.
- **Estudio**. Se trata del estudio que el alumno debe realizar para asimilar los contenidos de la asignatura. En la parte de teoría se han estimado tres horas de estudio por cada hora de clase magistral. En laboratorio, al ser sus contenidos más prácticos, la estimación ha sido de una hora de estudio por cada hora de explicación del profesor.
- **Asistencia y participación en clase.** Además de evaluar la asistencia a clase, decidimos que era importante incentivar la participación. Por ello, después de cada tema y práctica, se realiza una clase de debate/dudas donde el profesor anota el interés e intensidad de las intervenciones de cada alumno. Por otro lado, al disminuir considerablemente el número de horas dedicadas a transmitir conocimientos de profesor a alumno, se hizo necesario un nuevo enfoque de las clases magistrales: a los alumnos se les entrega previamente una guía de aprendizaje de cada tema (arduo trabajo el primer año) y el profesor se dedica a presentar el tema, justificarlo e incidir en los aspectos más importantes o complejos de comprender.
- **Tutorías.** Además de las tradicionales tutorías individuales en el despacho del profesor, los alumnos tienen tutorías en grupos reducidos (4-6 alumnos), que se desarrollan durante el horario previsto de clases. En estas tutorías el profesor realiza un seguimiento del desarrollo de los trabajos en grupo y resuelve dudas y problemas sobre dichos trabajos.
- **Exámenes**. El alumno cuenta con dos exámenes parciales de teoría, además de los exámenes finales ordinario y extraordinario.

4.4. Calendario

El reparto del esfuerzo del alumno a lo largo del tiempo debe ser cuidado y meditado. Por ello, es importante la elaboración de un calendario que prevea el desarrollo las actividades con la temporalidad y orden adecuados. Este calendario es una ayuda para los alumnos a la hora de planificar su trabajo en la asignatura (y con respecto a las demás también). Por ello, debe ser público antes del comienzo de las clases.

La elaboración del calendario no es una tarea fácil. Es necesario conocer el número de alumnos matriculados para decidir el número total de grupos de trabajo reducidos (de teoría y laboratorio) y, en función de este número, planificar los horarios de las tutorías para cada grupo. Cada hora de actividad debe ser planificada e indicada en el calendario. Además, durante su elaboración se debe garantizar que todos los alumnos tengan las mismas oportunidades y las mismas obligaciones. En la tabla 5 se presenta una parte del calendario del curso 2005-2006. Se puede comprobar que las distintas actividades se intercalan siguiendo una secuencia lógica: clases magistrales, debate y dudas, tutorías de grupo sobre el trabajo, tiempo para realizar el trabajo, entrega del trabajo.

4.5. Sistema de evaluación

Siguiendo los principios del sistema ECTS, el método de evaluación tiene en cuenta los tres aspectos siguientes: a) Los conocimientos adquiridos, ¿qué conoce?; b) Las competencias adquiridas, ¿qué sabe hacer?; c) El esfuerzo realizado, ¿cuánto ha hecho?

PGSI-2006		TEORÍA 4h/semana (M=1h, X=2h, J=1h)					PRÁCTICAS 2 h/semana (L=G2, X=G3, J=G1)				COMÚN
Fecha	Día	Clases magistrales	Clases de debate/duda	Tutorías en Grupo	Entrega Trabajos	Examenes parciales	Clases magistrales	Clases de debate/duda	Tutorías en Grupo	Entrega Trabajos	Otras (conferencias, …)
14-feb	M										presentación
15-feb	X	t1									explicar evaluación
. . . . .											
23-feb	J	t3									
28-feb	M	t3									
01-mar	X		t3	Trabajo T1							
02-mar	J			Trabajo T1							
06-mar	L						project-G2x2				
07-mar	M			Trabajo T1							
08-mar	X			Trabaj. T1x2			project-G3x2				
09-mar	J					temas 1-3	project-G1x2				
13-mar	L						project-G2	project-G2			
14-mar	M	t4									
15-mar	X	t5	t4				project-G3	project-G3			
16-mar	J	t5					project-G1	project-G1			
17-mar	V				trabajo T1						
. . . . .											

Tabla 5. Fragmento del calendario durante el curso 2005-2006

Una pieza clave de todo el diseño formativo realizado es la **evaluación continua**, con dos connotaciones igual de importantes:

i) se garantiza el reparto de los esfuerzos de aprendizaje a lo largo de todo el tiempo, y
ii) el profesor evalúa dichos esfuerzos y realimenta al alumno con una periodicidad alta (entre una y dos semanas).

Para implementar estas ideas sobre la manera de evaluar, se ideó un método de acumulación de puntos, cuyas principales características son:

- Por cada esfuerzo de aprendizaje el estudiante consigue una cantidad de puntos cuya cuantía depende del tipo de actividad, el esfuerzo (estimado en horas) y/o la calidad del resultado (calificación del profesor).
- El máximo de puntos acumulable por un alumno son 100, 65 de teoría y 35 de prácticas (ver tabla 6), de forma que la nota oficial en la asignatura es igual al número de puntos dividido por 10. Para aprobar el único requisito es obtener 50 puntos (un 5), pudiendo cada alumno decidir enteramente la estrategia de cómo y cuando conseguirlos.
- No es necesario superar el examen final para aprobar la asignatura. De hecho, éste examen se ha convertido en tan sólo una última oportunidad para los que quedaron cerca de los 50 puntos. Tampoco es suficiente sólo con el examen final para aprobar (solo supone 30 puntos); pero sí es posible aprobar, sin asistir a clase, realizando el examen final y otras actividades (en especial trabajos) que demuestren el aprendizaje de las competencias buscadas. Esto último se implantó pensando en el tercio de estudiantes que trabajan, a veces a más de 100 kms de distancia.

A la hora de decidir el papel del examen final tuvimos una duda importante: no tiene mucho sentido organizar un sistema completo de aprendizaje y evaluación continua si después la ley actual obliga a que los alumnos tengan derecho a aprobar simplemente estudiando contenidos y demostrando su conocimiento en un examen final. Esta duda fue consultada al Vicerrectorado de Convergencia Europea y Ordenación Académica de la UCLM, con la siguiente respuesta: la legislación actual concede el derecho a tener dos convocatorias de examen por cada matrícula, pero nada indica sobre que sólo con eso sea suficiente para aprobar.

Actividades	Total	Teo	Lab
Asistencia a clase	7	5	2
Participación en clases de debates/dudas	8	6	2
Tutorías en grupo	8	5	3
Tutorías individuales	7	5	2
Trabajos en grupo planificados	58	32	26
Exámenes parciales	12	12	-
Total Actividades Básicas	**100**	**65**	**35**
Ejercicios entregados	3	3	-
Trabajo Especial	4	4	-
Otras (conferencias, etc..)	1	1	-
Total Actividades Complementarias	**8**	**8**	**-**
Examen Final	30	20	10
TOTAL (máximo x alumno)	**100**	**65**	**35**

Tabla 6. Puntos máximos que se pueden conseguir **6**

El sistema de evaluación incluye una lista de **indicadores** que permiten conocer cómo y cuantos puntos se consiguen por cada actividad. Por ejemplo, se obtiene un punto por cada cuatro asistencias a clase (de 1 hora). En el caso de los trabajos en grupo, cada trabajo tiene un máximo de puntos que los propios alumnos reparten entre ellos, aunque existe un límite superior por persona y trabajo (para animar a que el esfuerzo se reparta entre todos los miembros del grupo). Los puntos que acumula cada alumno con cada trabajo son C*R/10, siendo C la calificación obtenida (nota del profesor entre 0 y 10) y R los puntos que el propio grupo asignó al alumno en el reparto. La lista completa de indicadores puede ser consultada en el apartado de evaluación de la web de la asignatura.

5. Primeros resultados

Tras el primer año de aplicación del nuevo diseño de la asignatura hemos **analizado** dos factores:

i) evolución de los resultados académicos; y
ii) comparación de nuestras estimaciones con lo realmente acontecido.

En la tabla 7 se resumen los resultados académicos en el curso 2005-2006. Si se comparan con los del método anterior (tabla 1), se puede concluir que se ha producido una mejora sustancial en los resultados académicos: 18% más de aptos, 9% menos de no presentados, y 9% menos de suspensos.

Alumnos totales:	92
Alumnos participando en la experiencia:	72
Grupos de clase:	1 de teoría, 3 de laboratorio
Grupos de trabajo:	11 de teoría, 11 de laboratorio
Resultados académicos:	67% aptos, 2% suspensos, 31% no presentados

Tabla 7. Resultados académicos con el nuevo diseño

Para evaluar la bondad de las estimaciones iniciales de esfuerzo, los profesores fuimos registrando información sobre todo lo ocurrido durante el curso. Además, se preparó una **encuesta** anónima para que los alumnos también anotaran sus esfuerzos y sugerencias. Para evitar suspicacias dicha encuesta se recogió después de publicadas las notas finales. La encuesta se dividió en tres bloques:

- Actividades Individuales, tanto de teoría (dos exámenes parciales y cinco trabajos) como de laboratorio (dos trabajos).
- Asignatura completa. **,** valoración global sobre el conjunto de la teoría, el laboratorio y la asignatura completa.
- Mejoras o cambios, propuestas de los alumnos para mejorar la asignatura en los siguientes aspectos: contenidos, método de enseñanza, sistema de evaluación, material educativo, página web y otros.

El bloque 3 se evaluó con preguntas abiertas, mientras que en los bloques 1 y 2 se siguió el siguiente esquema de preguntas:

- ¿Te parece adecuado realizar esta actividad (SI/NO)?
- ¿Cuántas horas totales le has dedicado?
- ¿Cuántas horas de esfuerzo por alumno crees que se deberían prever para su realización?
- ¿Cambiarías algo en esta actividad?. En caso afirmativo, indica brevemente el qué.

En cuanto a las estimaciones de los esfuerzos, en promedio, los alumnos dedicaron a la asignatura el 84% del tiempo previsto, y opinaron que el esfuerzo planificado total debería reducirse en un 9%. En cuanto a actividades individuales, las mayores desviaciones fueron en el esfuerzo del segundo parcial (sobrevalorado en 7 horas) y en el segundo trabajo de teoría (infravalorado en 2 horas).

Toda la planificación de la asignatura para el curso 2006-2007 se ha realizado teniendo en cuenta las anotaciones de los profesores y las respuestas de la encuesta a los alumnos. Así, los principales cambios realizados en el reparto del esfuerzo han sido incrementar la teoría (trabajos 1 y 2) y reducir las prácticas (ambos trabajos) y el segundo parcial.

6. Conclusiones

Se ha presentado una experiencia de adaptación de una asignatura a los principios y recomendaciones del EEES, siguiendo el paradigma de enseñanza centrada en el aprendizaje del alumno. Se ha ideado y aplicado un método de diseño *top-down* pensado para una

formación basada en competencias. Comenzando por lo más general hasta llegar a lo más concreto, con este método se definen los elementos del modelo formativo de la asignatura: objetivos y competencias, estructura general de los contenidos, tipos de actividades de enseñanza-aprendizaje y su importancia, estimación de los esfuerzos del alumno, calendario de actividades previstas, y sistema de evaluación continua basado en la acumulación de puntos mediante un lista de indicadores.

Los resultados del primer año son positivos y animan a continuar.

Las principales lecciones aprendidas son las siguientes:

El **esfuerzo del profesor** para diseñar una asignatura de este tipo es muy grande, sobre todo el primer año, pero este esfuerzo se compensa en años posteriores (ya lo estamos comprobando) y, lo más importante, es que se pueden conseguir significativas mejoras en los resultados académicos. En cuanto al esfuerzo de implantación, la mayor carga de trabajo del profesor se produce al comienzo del curso (planificación), se mantienen en un nivel intermedio durante casi todo el curso (seguimiento y evaluación continua), para acabar siendo más bajo en el periodo de exámenes.

Es muy importante realizar una correcta **planificación**. Además, el primer año es muy recomendable registrar todas las actividades para poder realizar un análisis posterior.

Al estimar el **esfuerzo de los alumnos** se debe evitar la tendencia de muchos profesores a inflar la carga de trabajo en sus asignaturas o a inflar los contenidos. En este tipo de formación la clave es aprender a hacer más que conocer conceptos.

Es necesario dedicar tiempo y atención (documentación en papel, página web, explicación en clase) para ayudar a los alumnos a asimilar lo antes posible la dinámica del nuevo método. Se debe insistir en que los tiempos son diferentes y ahora deben trabajar antes para descansar antes. Es importante que las **reglas del juego** sean públicas y estén claras.

Para el éxito de este nuevo método es clave el seguimiento, evaluación y realimentación **continuas** de los alumnos por parte del profesor. Si los alumnos no conocen su situación (puntos) de forma periódica, pueden desorientarse y desmotivarse.

Los alumnos están **contentos** porque, dicen, aprenden más y mejor, aunque al mismo tiempo están **asustados** de que se generalice el sistema a nivel de cursos completos o titulaciones por la sobrecarga de esfuerzo que les podría suponer. Esto lleva a una reflexión importante a la hora de diseñar los planes de estudios futuros: es necesario y muy importante tener en cuenta que un plan de estudios no es una mera acumulación de asignaturas. Si no pensamos con un punto de vista holístico y los profesores actuamos de forma coordinada, podemos llegar a situaciones tan poco recomendables como que algunas semanas del año los alumnos tengan una carga de 300 horas y otras una carga de 3 horas.

Además de seguir mejorando la asignatura, como **trabajo futuro** estamos evaluando el realizar una experiencia integrada con las asignaturas de ingeniería del software I y II, también obligatorias de cuarto y quinto años. La idea base sería que el mismo caso de los trabajos de PGSI se empleara para enseñar ambas ingenierías del software.

Referencias

[1] ACM Curricula Recommendations. Disponible en http://www.acm.org/education/curricula.html.

[2] AETIC: PAFET IV. Perfiles Profesionales TIC para la implantación de servicios y contenidos digitales, 2005.

[3] ANECA. Libro Blanco del Título de Grado en Ingeniería Informática. Proyecto EICE, 2005.

[4] Career Space. Perfiles de Capacidades Profesionales Genéricas de TIC. Oficina de Publicaciones de las Comunidades Europeas. Disponible en http://www.career-space.com/downloads/index.htm.

[5] Dirección General de Universidades. Adaptación de los Planes de Estudio al Proceso de Convergencia Europea. Proyecto EA 2004-0024. Octubre, 2004.

[6] IEEE CS. SWEBOK - Guide to the Software Engineering Body of Knowledge 2004 version.

[7] Cepeda, J.M. Metodología de la Enseñanza Basada en Competencias. Revista Iberoamericana de Educación, Vol 34(4), 2004.

La enseñanza de Estructura de Computadores en el EEES

Fermín Sánchez, Agustín Fernández, Josep Llosa
Dept.. Arquitectura de Computadors
Universitat Politècnica de Catalunya
c/Jordi Girona 1i 3, Mòduls D6 y C6
{fermin, agustin, josepll} @ac.upc.edu

Resumen

En este artículo se describe el proceso de diseño de la organización docente de una asignatura de Estructura de Computadores dentro del marco del EEES. A partir de las competencias profesionales relacionadas con la Estructura de Computadores se definen los objetivos formativos de la asignatura, para a continuación establecer los contenidos técnicos y la metodología docente que permiten conseguir los objetivos propuestos. El proceso de diseño se ha aplicado a la asignatura Estructura de Computadores II, de la Facultat d'Informática de Barcelona. Los indicadores muestran que la asignatura alcanza un alto grado de consecución de los objetivos propuestos.

1. Introducción

Los nuevos títulos de Grado [1] y Postgrado [2] harán especial énfasis en la formación profesional del titulado, a diferencia del enfoque académico de las actuales titulaciones. Adaptar las titulaciones actuales al EEES exigirá un esfuerzo docente para cambiar de paradigma educativo: (i) el protagonista es el alumno, no el profesor, (ii) el objetivo deja de ser la enseñanza del profesor para convertirse en preparar al alumno para el aprendizaje permanente y (iii) se pasa de simplemente "enseñar" a "aprender a aprender"

El cambio de paradigma educativo tiene unas consecuencias inmediatas para los alumnos: es imprescindible que lleven las asignaturas al día y aumenten su participación en las clases.

El ECTS [3] será la unidad de valoración de la actividad académica. Un crédito ECTS es equivalente a una dedicación del estudiante de entre 25 y 30 horas. Mide el trabajo del estudiante, no del profesor, e integra estudios teóricos y prácticos, actividades académicas dirigidas y el trabajo personal del estudiante. El volumen de trabajo de un alumno durante un año académico completo son 60 créditos ECTS.

Los estudios de Estructura de Computadores se centran en la relación entre los lenguajes máquina y los de alto nivel, la jerarquía de memoria y su influencia en el rendimiento de los programas y, finalmente, el manejo de dispositivos de entrada/salida desde el punto de vista del programador de lenguaje máquina.

En JENUI 2001 se abordó como tema estratégico "¿Cómo enseñamos memoria cache en nuestros planes de estudios?". En los últimos años, diversos autores han elaborado o usado simuladores o pequeños programas para ayudar a los alumnos a comprender los conceptos relativos a la estructura de los computadores mediante la realización de prácticas.

En [4] se describe un simulador gráfico de memorias cache multinivel que permite simular hasta 3 niveles de cache e incluye optimizaciones como la prebúsqueda. En [5] se expone un código, escrito en lenguaje máquina, que permite comprobar experimentalmente el efecto de la memoria cache sobre las prestaciones de un PC.

En [6] se detalla un conjunto de prácticas realizadas sobre Xilinx, orientadas a que el alumno comprenda mejor la organización y funcionamiento de la memoria principal.

En [7] se presenta una práctica para analizar, usando los contadores hardware del procesador, la influencia en el tiempo de ejecución de algunas optimizaciones sobre el código escrito en lenguaje máquina de la arquitectura IA32. Con el mismo objetivo, en [8] se presenta una práctica que enlaza ensamblador con C para mejorar el rendimiento de un conjunto de programas.

En [9] se describe un entorno de simulación de entrada/salida de sistemas basados en la arquitectura IA32. Con un enfoque muy distinto, en [10] se presentan los diseños de una tarjeta PCI y de una mochila que se conecta al puerto paralelo del computador. Ambos dispositivos permiten hacer prácticas de entrada/salida sin la necesidad de usar un simulador.

En este artículo se describe la organización de la asignatura "Estructura de Computadores II" (EC2), perteneciente al plan de estudio de la Facultat d'Informàtica de Barcelona. Se trata de una asignatura de tercer cuatrimestre, precedida por dos asignaturas que también abordan conceptos de estructura de computadores. En la asignatura de primer cuatrimestre, "Introducción a los Computadores", el alumno estudia un computador sencillo formado por un procesador RISC muy básico de 16 bits, un sistema de memoria con un único módulo y un único nivel y un sistema de entrada/salida con muy pocos periféricos y gestionado por encuesta. El segundo cuatrimestre, el alumno cursa "Estructura de computadores I", donde al computador que ya conoce se le añade una memoria cache directa de escritura retardada. En cuanto al procesador, se le añaden interrupciones. En EC2 se completa el modelo de computador mediante el estudio del lenguaje máquina de un procesador real con arquitectura IA32 y el sistema de memoria y de entrada/salida de un PC compatible.

El resto del artículo se organiza del siguiente modo. La Sección 2 enumera las competencias profesionales sobre las que incide la asignatura EC2 y sus objetivos formativos. En la Sección 3 se describe la organización docente que permite a EC2 conseguir los objetivos definidos. La Sección 4 presenta algunos resultados obtenidos los cinco últimos semestres y, finalmente, en la Sección 5 se presentan las conclusiones.

2. Competencias profesionales y objetivos formativos

En esta sección se clasifican y describen las competencias profesionales y los objetivos formativos, y se detallan las competencias en las que incide y los objetivos cubiertos por EC2.

2.1. Discusión preliminar

El Libro Blanco de la Ingeniería Informática [11] define un conjunto de competencias deseables en un Ingeniero en Informática. En su redacción participaron 53 universidades y la totalidad de centros públicos y privados que imparten estudios universitarios de informática en España. Para definir las competencias profesionales de un Ingeniero en Informática, se deben considerar también los trabajos desarrollados en el marco de la Comunidad Europea por *Career Space*, un consorcio formado por once compañías de las TIC y la Asociación Europea de Industrias TIC [12,13].

En este trabajo clasificamos las competencias profesionales de una titulación en tres categorías, tal como se describe en [14]:

- Competencias Técnicas: relativas a los conocimientos técnicos propios de la titulación
- Competencias Transversales: necesarias en un titulado con ese nivel académico, pero no relacionadas con sus conocimientos técnicos
- Competencias Deontológicas: recomendables en la actitud personal del titulado respecto a temas relacionados con la sociedad y con su formación personal como individuo

Una titulación debe diseñarse a partir de un conjunto de competencias profesionales, mientras que una asignatura se diseña a partir de un conjunto de objetivos formativos. Los objetivos formativos de las diferentes asignaturas deben permitir al titulado adquirir las competencias profesionales de la titulación.

Tanto ACM como IEEE [15,16] consideran que el nivel de competencia (la palabra *competencia* se usa aquí con otro significado) que debe alcanzar un ingeniero es el nivel de *Aplicación* y, en algunas materias, el de *Análisis*. Los niveles de *Síntesis* y *Evaluación* pertenecen al ámbito del doctorado. Sobre este tema destaca el trabajo descrito en [17], donde se adapta la taxonomía de Bloom [18] a una titulación técnica, respetando la definición de sus niveles (conocimiento, comprensión, aplicación, análisis, síntesis y evaluación) pero ofreciendo, para cada uno de ellos, definiciones y ejemplos propios de una Ingeniería Informática. Un trabajo similar fue realizado por la universidad de Pittsburg [19].

La taxonomía de Bloom distingue dos clases de objetivos: *generales* y *específicos*. Los objetivos generales tienen una granularidad más gruesa que los específicos y hacen referencia a capacidades que el alumno debe desarrollar de forma genérica. Los objetivos específicos, por el contrario, detallan de forma precisa los conocimientos y aptitudes que el alumno debe adquirir sobre un tema determinado.

A nivel de asignatura, y con una relación directa con la clasificación de competencias, en [20] los objetivos se clasifican en tres grupos:

- Técnicos: relacionados con los contenidos técnicos de la carrera.

- Transversales: relacionados con capacidades y aptitudes. Hacen referencia a capacidades generales, no ligadas a conceptos técnicos de la titulación.
- Deontológicos: relacionados con actitudes, valores y normas. Se refieren a la disposición personal del estudiante con respecto a la sociedad.

En este artículo usaremos esta clasificación.

2.2. Competencias profesionales y objetivos formativos de EC2

Las competencias técnicas sobre las que incide EC2 son las siguientes:

- Capacidad de implementar código crítico siguiendo criterios de valor estratégico, urgencia, rendimiento y seguridad. En particular, capacidad de diseñar software que se ejecute con el máximo rendimiento sobre una arquitectura concreta
- Capacidad de evaluar los requerimientos hardware que debe tener el sistema informático de una organización

En cuanto a competencias transversales, EC2 contribuye a desarrollar las siguientes:

1. Capacidad para resolver problemas
2. Adquisición de hábitos de trabajo y estudio
3. Capacidad de trabajo en equipo
4. Capacidad de razonamiento crítico
5. Capacidad de análisis y síntesis
6. Capacidad de organización y planificación
7. Capacidad para tomar decisiones

EC2 no incide en competencias deontológicas.

A partir de las competencias hemos definido los objetivos de EC2. Los objetivos técnicos de tipo general son los siguientes:

8. Capacidad de desarrollar aplicaciones en C que integren rutinas en ensamblador
9. Capacidad de describir cómo se realiza en una máquina real la comunicación con los dispositivos básicos de entrada/salida
10. Capacidad de explicar las implicaciones de una configuración de entrada/salida en el rendimiento del computador
11. Capacidad de evaluar el rendimiento de una determinada configuración de entrada/salida
12. Capacidad de evaluar los esquemas básicos de memorias cache
13. Capacidad de evaluar el rendimiento de un esquema real de memoria principal
14. Capacidad de justificar la existencia de una jerarquía de memoria y de explicar su funcionamiento

Los objetivos transversales de EC2 son mejorar incrementalmente cada una de las competencias transversales descritas en esta sección. Esto se consigue con una adecuada metodología docente. A continuación se clasifican por temas los objetivos técnicos específicos.

Objetivos específicos de Lenguaje Máquina

15. Capacidad de realizar programas de hasta 25 líneas en ensamblador IA32
16. Capacidad de utilizar un depurador (*debugger*) para eliminar errores en un programa escrito en ensamblador y/o C
17. Capacidad de enlazar programas escritos en ensamblador con programas escritos en C en un entorno de trabajo Linux
18. Capacidad de describir cómo se almacenan las estructuras de datos complejas y cómo acceder a ellas de forma eficiente
19. Capacidad de traducir a ensamblador un código escrito en C y realizar optimizaciones simples sobre dicho código

Objetivos específicos del sistema de memoria

20. Capacidad de describir las características de los diferentes tipos de DRAM
21. Capacidad de explicar las implicaciones (tiempo de respuesta, complejidad, coste, etc.) de la implementación hardware de los diferentes esquemas de memoria cache
22. Capacidad de seguir una traza de referencias a memoria, indicando aciertos y fallos de cache, TLB y página. La traza puede ser explícita o estar implícita en un programa escrito en C o ensamblador
23. Capacidad de evaluar configuraciones de memoria cache que incluyan prebúsqueda, *buffers* de escritura, varios niveles de cache, cache de víctimas, cache no bloqueante y optimizaciones en el tiempo se servicio en caso de fallo
24. Saber evaluar el impacto de realizar transformaciones simples en un programa escrito en C en una jerarquía de memoria determinada
25. Capacidad de explicar los conceptos básicos de memoria virtual: traducción de direcciones,

fallo de página, implementación de la tabla de páginas, TLB y fallo de TLB

26. Capacidad de explicar la relación entre la memoria virtual y la cache cuando se realiza la traducción de direcciones

Objetivos específicos del sistema de entrada/salida

27. Capacidad de explicar el funcionamiento de los mecanismos de sincronización de entrada/salida en un computador real
28. Capacidad de explicar las implicaciones que tiene la transferencia vía DMA en un sistema con jerarquía de memoria
29. Capacidad de explicar las características de los diferentes tipos de buses y sus sistemas de arbitraje: síncronos o asíncronos, serie o paralelo, ciclo partido o completo, arbitraje centralizado o distribuido
30. Capacidad de justificar la existencia de una jerarquía de buses y de evaluar el ancho de banda de cualquier bus de la jerarquía
31. Capacidad de describir los buses estándar de un PC compatible y relacionarlos con los periféricos que soportan
32. Capacidad de explicar el concepto de RAID y de calcular el rendimiento de un sistema RAID de un solo nivel (RAID 0 – RAID 6)

Siguiendo las recomendaciones descritas en [21], a partir de los objetivos formativos definidos para EC2 hemos diseñado los contenidos, las estrategias docentes y el método de evaluación.

3. Organización docente

Esta sección describe la forma en que se ha organizado EC2 para conseguir los objetivos propuestos en la Sección 2. EC2 se imparte de forma cuatrimestral durante 5 horas semanales. Está organizada tres tipos de clases:

- Clases de teoría
- Clases de problemas
- Clases de laboratorio

Las clases de teoría y problemas son de dos horas y se realizan conjuntamente. Las clases de laboratorio se distribuyen semanalmente a lo largo de las 13-14 semanas que dura el curso a razón de una hora semanal.

3.1. Contenidos técnicos y objetivos técnicos cubiertos por cada tipo de actividad

Las clases de teoría y problemas están estructuradas en tres grandes bloques temáticos:

- Arquitectura y Lenguaje Máquina IA32: cubre los objetivos 15, 17, 18 y 19
- Jerarquía de memoria: cubre los objetivos 12, 13, 14, 20, 21, 22, 23, 25 y 26
- Subsistema de entrada/salida: cubre los objetivos 9, 10, 11 y 27-32

Las clases de laboratorio cubren los objetivos 8, 15, 16, 17, 19, 22 y 24, y permiten fijar y profundizar en la mayoría de los objetivos trabajados en las clases de teoría y problemas. En [22] pueden encontrarse los enunciados de las prácticas, los enunciados de problemas y un detalle completo de los contenidos.

3.2. Método de evaluación:

La evaluación sirve para determinar si el alumno ha alcanzado los objetivos de la asignatura, pero también es una herramienta muy potente para ayudar a que el alumno alcance dichos objetivos.

La característica fundamental del método de evaluación de EC2 es que los alumnos pueden superar la asignatura mediante la evaluación continuada sin necesidad de hacer el examen final. De esta forma, se fomenta que los alumnos lleven la asignatura al día.

En la evaluación de EC2 distinguimos 2 componentes: teórica (80%) y práctica (20%). La componente práctica se evalúa a partir de un conjunto de ejercicios de laboratorio, y no es imprescindible aprobarla. La componente teórica se evalúa o bien a partir de un examen final o bien a partir de un mecanismo de evaluación continua que describimos a continuación.

Durante el curso, y en horas de clase, se realizan tres controles de dos horas de duración sobre la materia. Los controles están distribuidos uniformemente en el tiempo, cada cuatro o cinco semanas. El nivel de estos controles no es inferior al del examen final. El segundo y tercer control incluyen conceptos de los controles anteriores. Para superar la componente teórica, el alumno ha de aprobar los tres controles de forma independiente o bien obtener una nota media superior o igual a 6. Los alumnos que no superen la componente teórica deben ir al examen final.

El profesor evalúa las prácticas sesión a sesión en función del trabajo y los resultados de los

alumnos, sin necesidad de realizar ningún examen, entrega de prácticas o entrevista. La evaluación es individual pese a que las prácticas se hacen en grupos de dos.

Aunque no es necesario aprobar las prácticas para aprobar la asignatura, la experiencia nos muestra que casi ningún alumno con el laboratorio suspendido aprueba la asignatura.

3.3. Metodología docente

La metodología docente influye tanto en la organización de las clases como en el método de evaluación. El primer día de clase se explica en detalle al estudiante lo que se espera de él. Se definen los objetivos de la asignatura y se le detalla cómo se esperan conseguir. Esto contribuye a que el estudiante comprenda que debe trabajar de forma continuada desde el principio si quiere alcanzar los objetivos que se le han propuesto (y ahorrarse el examen final).

Clases de teoría

Las clases de teoría se imparten con el soporte de transparencias que se ponen a disposición de los alumnos al empezar el curso, tanto en formato impreso como en la web de la asignatura. El uso de transparencias aporta ciertas ventajas, como la minimización de los errores de trascripción (tanto por parte del profesor en la pizarra como por parte de los alumnos al copiar sus propios apuntes) y la agilización de las clases, especialmente aquéllas en las que hay dibujos complejos. Sin embargo tiene ciertas desventajas, como una rápida disminución de la atención de los alumnos a medida que avanza la clase o una menor asistencia de los alumnos a las clases, que creen erróneamente que es suficiente con estudiar las transparencias para aprobar la asignatura.

Para combatir los efectos negativos de las transparencias, éstas incorporan puntos de ruptura de la dinámica de la clase (*breaks*) consistentes en pequeños problemas que los alumnos deben realizar en unos minutos. En ocasiones, los *breaks* consisten en una pregunta o un tema de discusión. Los *breaks* permiten, por un lado, recuperar la atención del alumno mediante un cambio de actividad y, por otro, consolidar los conceptos recientemente explicados. Además, los *breaks* fomentan la cooperación de los alumnos, el espíritu crítico y la participación en clase (competencias 1, 3 y 4).

Además de las transparencias, el profesor recurre con frecuencia a la pizarra, ya sea para ampliar o clarificar algún concepto o para introducir algún concepto secundario o ejemplo. El uso de la pizarra contribuye a recuperar la atención de los alumnos.

Clases de problemas

Las clases de problemas están mezcladas con las de teoría, es decir, no hay un horario especial dedicado a ellas. Esto permite al profesor dedicar tiempo a problemas en el momento más oportuno, cuando el alumno ya ha adquirido y madurado los conocimientos necesarios. También contribuye a mejorar la atención de los alumnos, ya que permite dedicar una parte de la clase a la resolución de problemas y otra a la adquisición de conocimientos teóricos.

Los alumnos disponen de un conjunto de enunciados de problemas desde el primer día y las clases de problemas requieren su participación. Los problemas a resolver son anunciados con antelación y los alumnos disponen de varios días para poder resolverlos por su cuenta. En ocasiones se les da tiempo en clase para que resuelvan un determinado problema. El profesor aprovecha este tiempo para pasear por la clase y comprobar el grado de comprensión de la materia que han alcanzado los estudiantes. El profesor no suele resolver los problemas en la pizarra. Durante la clase, algunos alumnos convenientemente seleccionados por el profesor explican sus soluciones en la pizarra y se entablan discusiones públicas (moderadas, guiadas y fomentadas, por el profesor) sobre las diversas soluciones propuestas. Este tipo de actuación contribuye a conseguir las competencias 1, 2, 4 y 6.

Los enunciados de la colección de problemas no contienen las soluciones, con el objeto de facilitar la discusión entre los alumnos y fomentar que exploren diversas soluciones. En presencia de colecciones con soluciones, los estudiantes tienden a intentar entender la solución propuesta en lugar de tratar de encontrarla por sí mismos. No obstante, en aquellos problemas en que el resultado es único o sencillo de exponer, se les da la solución numérica final o bien algún dato que les permita comprobar si su solución es correcta.

Clases de laboratorio

Las clases de laboratorio se realizan en sesiones de una hora semanal en grupos de dos alumnos (competencia 3) sobre un entorno Linux y en PCs compatibles. Estas clases consisten en la resolución e implementación de pequeños problemas hasta su correcto funcionamiento (competencia 1) o en la realización de pequeños experimentos y de su posterior análisis (competencias 4 y 5). Son, por tanto, un refuerzo adicional de la teoría y están perfectamente sincronizadas con ella.

Para fomentar la preparación de las prácticas y garantizar el correcto aprovechamiento de las sesiones de laboratorio, los alumnos deben entregar un trabajo previo al inicio de cada sesión, que consiste en la realización de pequeños ejercicios que aportan pistas o les guían hacia la correcta realización de la práctica. El trabajo previo contribuye a conseguir las competencias 1, 2 y 6. En algunas sesiones se les pide un informe posterior, fomentando las competencias 3, 4 y 5. Ambos informes requieren poca carga para los alumnos, ya que pueden ser manuscritos y suelen tener una extensión de entre una y tres páginas.

Evaluación formativa

EC2 implementa una evaluación formativa, ya que ha sido diseñada con el objetivo de fomentar que los alumnos lleven la asignatura al día y contribuir a su aprendizaje. Por tanto, el método de evaluación de la asignatura es una parte importante de la metodología docente.

Además de los tres controles de la evaluación continuada, se realizan otros dos o tres controles sorpresa durante el curso, intercalados con los controles de la evaluación continuada. Estos controles tienen una duración de entre media hora y una hora y se resuelven inmediatamente después de su elaboración.

El objetivo de los controles sorpresa no es evaluar a los estudiantes, y por ello no tienen ningún impacto negativo en la nota final. No obstante, y como motivación para los estudiantes, estos controles son tenidos en cuenta en ciertas circunstancias siempre positivas, como redondeos al alza, liberar del examen final a un alumno con un control suspendido y media muy próxima a 6 o aprobar a un alumno con nota final cercana a 5.

Desde el punto de vista formativo, los controles sorpresa cumplen múltiples funciones. En primer lugar, permiten al alumno realizar un autoseguimiento. En muchos casos hace que se dé cuenta de la necesidad de incrementar el esfuerzo dedicado a la asignatura sin esperar a los días previos al control evaluativo. Esto contribuye a que el alumno lleve la asignatura al día y aproveche mejor las clases (contribuyendo además a las competencias 2 y 6). En segundo lugar, obliga al alumno a resolver varios problemas de forma individual (competencia 1), procurando el máximo aprovechamiento de la posterior sesión de corrección/discusión. Los controles sorpresa son autocorregidos por los alumnos. Para ello, los estudiantes son organizados en grupos de dos o tres y se les distribuyen otros tantos controles de sus compañeros. El profesor explica la solución y la forma de puntuar y los alumnos corrigen los controles de sus compañeros. Esto les hace darse cuenta no sólo de los errores que han cometido, sino también de errores cometidos por sus compañeros. Este tipo de corrección les obliga a ponerse en la piel del profesor durante unos minutos y a entender lo difícil que puede resultar corregir un examen y lo importante que es que esté bien organizado y escrito (competencia 4). También les permite que vean soluciones distintas de la suya y de la propuesta por el profesor, y les obliga a evaluar la calidad de las diversas soluciones (competencia 7). Por supuesto, estas sesiones de corrección de los controles sorpresa fomentan también la participación en clase.

El hecho de que las prácticas se evalúen de forma individual y que también se evalúe el trabajo previo fomenta que los alumnos realicen algunos ejercicios de forma semanal contribuyendo a las competencias 2 y 6. El que todas las sesiones de laboratorio sean ‘actos evaluables’ y que las diferentes prácticas estén muy relacionadas con las clases teóricas incentiva a los alumnos a seguir el laboratorio de forma constante y regular. Aunque los laboratorios no son obligatorios para poder aprobar la asignatura, el método de evaluación hace que prácticamente todos los alumnos asistan a todas las sesiones habiendo realizado el trabajo previo, aprovechando así de forma satisfactoria dichas sesiones. Además, evaluar todas las sesiones de laboratorio permite al profesor conocer a sus alumnos e ‘incentivarlos’ cuando sea necesario.

Finalmente, esta metodología docente no sólo no ha aumentado el trabajo de los profesores en cuanto a corrección se refiere, sino que ha conseguido distribuirlo a lo largo del curso.

3.4. El forum

La asignatura dispone de un forum de discusión vía web. En este forum los estudiantes plantean preguntas a sus compañeros o proponen soluciones a problemas. Este forum fomenta la discusión colectiva de los problemas fuera de las aulas y contribuye al aprendizaje del estudiante (competencias 1, 4 y 7). Asimismo, incrementa su capacidad para explicar los conocimientos adquiridos.

Para fomentar la participación en el forum, los profesores proponen semanalmente problemas o temas de discusión en el forum. La misión de los profesores es la de ejercer de moderadores y, si es necesario, guiar a los alumnos. En ningún caso el profesor resuelve el problema ni en el forum, ya que iría en contra de la participación. Este forum tampoco es un mecanismo para realizar preguntas a los profesores: para esto ya existen las horas de consultas.

3.5. Bibliografía

La bibliografía de la asignatura se divide en básica y complementaria [22]. No existe un libro de texto como tal, aunque en alguno de los temas se utilizan como base determinados capítulos de la bibliografía básica.

Los estudiantes pueden acceder a toda la documentación de la asignatura a través de la página web de la asignatura [22]. Entre otras cosas, los alumnos disponen de las casi 600 transparencias del curso, la documentación de prácticas, los enunciados de problemas y exámenes de cursos anteriores. Toda la documentación de la asignatura está bajo una licencia de Creative Commons [23] y protegida por un *username* y *password*, que se pone en conocimiento de los estudiantes a comienzo de curso, para evitar su uso no autorizado.

3.6. Dedicación del estudiante:

Desde el curso 2005-2006 se realiza en la FIB una experiencia para correlacionar la dedicación de los alumnos a una asignatura con las horas especificadas en el Plan Docente. Los datos son introducidos de forma voluntaria por los alumnos semana a semana, y luego se agrupan en función de su nota final de cada asignatura. Por ello, sólo disponemos de datos de un 8% de los alumnos (Entre 150 y 250 matriculados cada semestre).

Los resultados experimentales nos indican que la asignatura está bien dimensionada. El esfuerzo estimado por los profesores para seguir la asignatura correctamente es de 150 horas (6 créditos ECTS). Los alumnos declaran que dedican en media entre 116 y 130 horas, pero los que han sacado mejores notas (>7) declaran una dedicación media de entre 130 y 150 horas. Sin embargo, las encuestas oficiales que realizan la mayoría de los estudiantes indican una dedicación media de entre 1 y 2 horas de estudio por cada hora de clase, lo que supone entre 130 y 195 horas de trabajo. Dado que la encuesta ECTS se realiza semana a semana y las encuestas oficiales se realizan a final del curso, consideramos más exactos (aunque menos significativos) los resultados de la primera encuesta.

4. Resultados

La Tabla 1 muestra resultados académicos de los últimos cuatrimestres. En todos ellos se ha usado la metodología descrita. En media, el 58% de alumnos aprueba la asignatura con la evaluación continuada (Aprob. curso). En los últimos cuatrimestres este número ha disminuido porque hasta el curso 2005-06 no se exigía aprobar los 3 controles independientemente, sólo la media final. De forma complementaria, ha ido aumentando el número de alumnos que aprueba el examen final (37% en media), desde el 16% del primer cuatrimestre al 50% del último cuatrimestre analiazado. Al margen del número de aprobados, consideramos que la mayoría de los alumnos alcanzan los objetivos de la asignatura.

Curso	Núm. Alum.	Aprob. curso	Aprob. total	Susp. total	NP
2004/05 Q1	262	65,6%	71,4%	25,6%	3,1%
2004/05 Q2	196	73,0%	82,7%	12,2%	5,1%
2005/06 Q1	236	52,1%	73,7%	16,9%	9,3%
2005/06 Q2	177	46,9%	67,8%	20,9%	11,3%
2006/07 Q1	180	56,1%	78,3%	14,4%	7,2%

Tabla 1. Resultados académicos

La valoración de la asignatura que realizan los alumnos en las encuestas oficiales es buena (3.7 sobre 5 en los 2 últimos cursos), y está por encima de la valoración media de las asignaturas del centro (3.4).

5. Conclusiones

En este artículo hemos presentado una propuesta de asignatura de Estructura de Computadores dentro del marco definido por el EEES. A partir de las competencias profesionales se han diseñado los objetivos de la asignatura, y a partir de estos los contenidos, el método de evaluación y la metodología docente. La propuesta se centra en el aprendizaje del alumno, de forma que las actividades lectivas y evaluatorias están orientadas a fomentar la participación activa del estudiante y su trabajo constante y regular a lo largo del curso.

La asignatura se ha impartido con éxito durante los últimos cinco cuatrimestres. Los resultados obtenidos por los alumnos (porcentaje de aprobados), la sensación subjetiva de los profesores y la evaluación realizada a partir de las encuestas de los estudiantes sugieren que los alumnos alcanzan los objetivos propuestos en el tiempo estimado.

Referencias

[1] *Real Decreto 55/2005, de 21 de Enero, por el que se establece la estructura de las enseñanzas universitarias y se regulan los estudios universitarios oficiales de Grado,* BOE núm. 21, Martes 25 de Enero de 2005

[2] *Real Decreto 56/2005, de 21 de Enero, por el que se establece la estructura de las enseñanzas universitarias y se regulan los estudios universitarios oficiales de Postrado,* BOE núm. 21, Martes 25 de Enero de 2005

[3] http://europa.eu.int/comm/education/programmes/socrates/ects_en.html

[4] R. Almisas, R. Paz, A. Linares, C. Amaya y J.L. Sevillano. *Un simulador de memorias cache multinivel.* JENUI2001.

[5] J. Sahuquillo, J. Flich y J. Real. *Prácticas experimentales de memoria cache.* JENUI2003.

[6] J. Flich, J. Real y J. Sahuquillo. *Prácticas de diseño de sistemas de memoria.* JENUI2003.

[7] A. Linares, R. Paz, S. Vicente, M.A. Rodríguez y F. Díaz . *Análisis a bajo nivel de Procesadores superescalares reales.* JENUI2001.

[8] M. Anguita, F.J. Fernández, A.F. Díaz, A. Cañas y A. Prieto. *Práctica de optimización para asignaturas de Estructura de los Computadores.* JENUI2004.

[9] M. Prieto, A.J. Vicente y J.A. Vargas. *Simulador de dispositivos de entrada/salida programables.* JENUI2002.

[10] M. Prieto, A.J. De Vicente, R. Aldea y E. Bonillo. *Soluciones para las prácticas de entrada/salida.* JENUI2006.

[11] J. Casanovas, J.M. Colom, I. Morlán, A. Pont y M.R. Sancho, *El libro blanco de la Ingeniería en informática*: el proyecto EICE, JENUI2004,http://www.aneca.es/modal_eval/docs/libroblanco_informatica.pdf

[12] *Directrices para el desarrollo curricular*, Career Space, CEDEFOP. www.cedefop.eu.int

[13] *Perfiles de capacidades profesionales genéricas de TIC*, Career Space, CEDEFOP. www.cedefop.eu.int

[14] J. García, F. Sánchez, y R. Gavaldà. *Cómo diseñar un Grado en Informática.* JENUI2006.

[15] *IEEE / ACM Computing Curricula*. http://www.computer.org/education/cc2001/

[16] *Accreditation Criteria*. Accreditation Board for Engineering and Technology, Inc., http://www.abet.org/

[17] M. Valero-García y J.J. Navarro. *Niveles de competencia de los objetivos formativos de las ingenierías.* JENUI2001.

[18] B.S. Bloom, J.T. Hastings y G.F. Manaus. *Taxonomía de los objetivos de la educación, Tomo I (conocimientos) y Tomo II (dominio afectivo).* Ed. Marfil, Alcoy 1973.

[19] *Bloom and Krathwohl Definitions of Levels and McBeath Action Verbs*. The University of Pittsburg, 2000. http://www.engrng.pitt.edu/~ec2000/ec2000_downloads.html

[20] F. Sánchez y R. Gavaldà. *Objetivos formativos del primer curso de las ingenierías informáticas y estrategias docentes relacionadas.* SINDI2005

[21] F. Sánchez. *¿Cómo serán las asignaturas del EEES?* JENUI2005

[22] http://docencia.ac.upc.edu/FIB/EC2/

[23] http://creativecommons.org/licenses/by-nc-sa/2.5/es

El idioma vehicular docente; un reto a la movilidad en el Espacio Europeo de Enseñanza Superior (EEES)

Germán Santos Boada y Josep Solé Pareta
Departamento de Arquitectura de Computadores
Universidad Politécnica de Cataluña
C/ Jordi Girona Salgado s/n, 08034 Barcelona
german@ac.upc.edu pareta@ac.upc.edu

Resumen

La asignatura Servicios Públicos de Datos (SPD) de Ingeniería Informática de la Universidad Politécnica de Cataluña se está adaptando a los requerimientos que plantea el Espacio Europeo de Enseñanza Superior introduciendo conceptos de nuevos paradigmas docentes en base a la integración de la enseñanza expositiva y el trabajo cooperativo.

La utilización del Campus Virtual basado en tecnología Moodle de la Universidad permite además una mayor flexibilidad y una mayor adaptación de los estudiantes a las nuevas metodologías docentes, destacando todo aquello que hace referencia a los sistemas de evaluación ECTS y al trabajo cooperativo con interdependencia positiva.

Ante un escenario como el indicado, un nuevo reto surge en el horizonte educativo relacionado con el idioma vehicular docente.

La movilidad de los estudiantes europeos, uno de los fundamentos del EEES, abre nuevas fuentes de problemas docentes relacionados con la comunicación entre estudiantes y profesores, ya que la predominancia del idioma inglés como vehicular a nivel mundial, no debe necesariamente ser incompatible con la utilización de los idiomas locales.

La búsqueda del equilibrio en el uso de distintos idiomas docentes, respetando el derecho de la movilidad y la comunicación de los estudiantes europeos, no debe estar reñido con el mantenimiento de la cultura lingüística de cada región.

Este artículo explica la experiencia de los últimos años en SPD en este sentido, considerando la casuística propia de Cataluña que dispone de dos idiomas oficiales.

Los resultados son esperanzadores en el sentido de que la presencia en los últimos años de estudiantes Erasmus da suficientes pistas de cómo se puede afrontar el tema, y de que la *flexibilidad* se postula como la herramienta más eficaz a la hora de afrontar con garantías de éxito la problemática.

1. Motivación

La aplicación de los conceptos relacionados con el Espacio Europeo de Enseñanza Superior (EEES) suponen modificaciones destacables en las carreras universitarias europeas. En primer lugar se pretende evolucionar el paradigma docente balanceando el centro de gravedad del profesor al alumno y proporcionando mecanismos que le permitan a éste último aprender a aprender.

En segundo lugar, introduciendo los créditos ECTS (European Credit Transfer System) con los que se pretende valorar del alumno todo aquello que éste hace en el aprovechamiento de las asignaturas, tanto dentro como fuera de las aulas.

Y en tercer lugar hay una profunda remodelación de las titulaciones buscando la homogeneización europea en las mismas basada en un proceso correlativo de Grado, Master y Doctorado.

Todo ello conduce a facilitar la movilidad de los estudiantes europeos de forma que éstos se encuentren en cualquier lugar de Europa con unas metodologías, procedimientos y titulaciones equiparables, y con ello facilitar la posibilidad de estudiar en cualquier universidad.

Esta movilidad hace que alumnos de distintos idiomas maternos se encuentren estudiando la misma titulación en una universidad de cualquier lugar de Europa.

En el caso de la asignatura objeto de esta ponencia, SPD, en una misma aula se puede dar el

caso de que para un estudiante determinado le coincidan hasta cuatro idiomas vehiculares, español, catalán, el propio del estudiante, y el inglés.

La presencia en las aulas en los últimos años de estudiantes Erasmus ha permitido elaborar estrategias de resolución de esta problemática, y ello ha motivado que se disponga de información y resultados de la integración lingüística de los mismos que puede ser de gran utilidad para afrontar este problema ante los retos del EEES.

2. Posición oficial

Desde el documento de Bolonia de junio de 1999 en el que se define el Espacio Europeo de la Enseñanza Superior, hasta las siguientes reuniones de los ministros de la Unión Europea en Berlín 2003, o Bergen 2005 se hace una promoción de la movilidad mediante la eliminación de los obstáculos al ejercicio efectivo del derecho a la libre circulación haciendo hincapié en los siguientes aspectos:

- para los estudiantes, en el acceso a las oportunidades de enseñanza y de formación
- para los profesores, los investigadores y el personal administrativo en el reconocimiento y valoración de los periodos de investigación, enseñanza y formación en un contexto europeo.

Los Ministros de Educación de la UE enfatizan la importancia de la movilidad por razones académicas, culturales, políticas y sociales así como económicas.

Además se indica de una forma clara que se intensificarán los esfuerzos para salvar los obstáculos de la movilidad, instando a los estudiantes y a las instituciones a que hagan pleno uso de los programas de movilidad apoyando el reconocimiento pleno de periodos de estudio en el extranjero.

En ninguno de los comunicados de la Conferencia de Ministros de la UE responsables de la Educación Superior se hace mención de la problemática lingüística, aunque esta cuestión podría implícitamente considerarse como uno de los obstáculos a los que se hace referencia para conseguir la plena movilidad [5].

El Ministerio de Educación español recoge en su documento marco todas estas inquietudes y se hace eco de la necesidad de reducir las barreras de la movilidad, pero no se define cómo se debe resolver el problema del idioma docente para que esta movilidad sea una realidad. En la Ley Orgánica de modificación de la LOU que actualmente está en trámite en el Senado, se indica en el artículo sobre el régimen jurídico de las universidades que los poderes públicos y las universidades en sus estatutos establecerán los mecanismos para que en los procesos de acogida de los miembros de la comunidad universitaria se favorezca el conocimiento suficiente de las lenguas oficiales.

Por parte de Cataluña y bajo el programa NORMA se han aprobado subvenciones para las universidades catalanas bajo el epígrafe de Atención a la Movilidad en un contexto multilingüe para la acogida sociolingüística y cultural en el entorno universitario.

La UPC ha establecido un programa de acogida en el que se informa y se da al estudiantado foráneo todas las herramientas necesarias para su integración lingüística [9].

Queda, por tanto, en manos de las Universidades definir cómo debe ser el modelo lingüístico a aplicar en sus aulas para resolver el problema de la movilidad de estudiantes europeos.

3. La asignatura SPD y el EEES

La aplicación de los conceptos EEES en la asignatura SPD se realiza utilizando la metodología propia de los créditos ECTS en la que se valora todo aquello que hace el estudiante en su formación, tanto dentro como fuera de las aulas, y aplicando metodologías de trabajo cooperativo e interdependencia positiva [8]. Como soporte fuera de las aulas se utiliza el Campus Virtual ATENEA, propio de la Universidad Politécnica de Catalunya [1].

Los estudiantes disponen desde el primer día de clase, y conforme al ritmo de avance en el programa , de la documentación del contenido del programa que se va a estudiar en los próximos días en el campus virtual de la asignatura. El formato es electrónico en forma de transparencia documentada, autosuficiente en sí misma para su comprensión aunque elaborada en forma esquemática, proviniendo de la bibliografía especificada en el programa de la asignatura. El profesor introduce los temas con un método expositivo y siguiendo estrictamente la

documentación, sin añadidos ni saltos, pero utilizando lo menos posible la pizarra, estableciendo un diálogo con los estudiantes para que sean ellos mismos los que vayan desarrollando sus notas sobre la documentación que les va a servir como base de estudio.

La metodología de explicación busca la máxima interactividad entre estudiantes, tanto entre ellos como con el profesor, intentando encontrar la participación del estudiante, en base a los conceptos del ejemplo, el caso de estudio, y la resolución de problemas con la participación de los estudiantes en la resolución de los mismos[6].

Los estudiantes, después del primer mes de clase, han de desarrollar un trabajo en grupo formado por tres personas, relacionado con la asignatura. El objetivo del trabajo es que el alumno adquiera alguna competencia especial sobre los objetivos de la asignatura, considerándolo como un complemento al programa de la misma. En el campus virtual el profesor presenta una relación de posibles temas, y es el propio grupo el que elige el tema. El grupo puede presentar un tema propio con la única condición de que sea un complemento al programa oficial y por tanto no conste en la documentación de la asignatura o que en cualquier caso sea una profundización de los temas tratados en clase. En función de sus inquietudes, el grupo presenta al profesor durante el segundo mes el índice de la actividad que desea desarrollar. El profesor lo valida y a partir de su aprobación inicia un proceso de tutoría, facilitándoles la obtención de información y la síntesis de la misma. El resultado final ha de ser un documento electrónico de un máximo de 30 transparencias, con figuras y notas explicativas. El objetivo es que el estudiante lea mucho y escriba muy poco, es decir, incentivar la capacidad de abstracción y de recogida de referencias bibliográficas. Finalmente el trabajo se entrega electrónicamente antes del examen final y debe quedar preparado para que el alumno pudiera presentarlo públicamente si fuera preciso, y es el profesor el que lo evalúa con la posibilidad de una entrevista con el estudiante [7].

La asignatura presenta un modelo de evaluación basado en otros aspectos además del examen. Éste sigue siendo la base fundamental por la que el estudiante superará la asignatura, pero matizado con la adquisición de competencias que le supone el trabajo en grupo. En primer lugar existe evaluación continuada con dos controles equidistantemente ubicados en el periodo lectivo y un examen final. Además se realiza un control de asistencia y un trabajo en grupo. El examen final queda como una oportunidad adicional que se da a los estudiantes que no hayan podido hacer los controles o como medio de subir nota.

Nota = [70 % Máximo {Nota media controles, Nota examen final}] +20 % Nota trabajo + 10 % Nota asistencia (se ha de acreditar un mínimo del 80 % de asistencia a clase)

Los exámenes están basados en la documentación base de la asignatura y en los objetivos de aprendizaje, discutidos en clase, y no en la bibliografía adicional, y se permite la utilización de cualquier libro, apuntes o documentación en los mismos.

4. El idioma vehicular

Como hemos podido ver antes en el apartado de la posición oficial sobre el multilingüismo, al final de tanta documentación y declaraciones, y a pesar de todas las buenas intenciones de las normativas vigentes, es responsabilidad del profesor decidir cuál va a ser en cada momento el idioma vehicular docente. En comunidades autónomas en las que existe un único idioma oficial, el español, la decisión se combina entre el uso de este idioma, o la utilización del inglés como idioma universal de intercambio de conocimientos.

Cada vez más, las universidades ofrecen asignaturas en las que el inglés es el idioma vehicular, pero no hay que confundir multilingüismo con el uso del inglés. En las comunidades autónomas con dos idiomas oficiales, las universidades recomiendan por razones culturales el uso del idioma propio siempre que el profesor lo asuma. Por ejemplo en Cataluña entre el 60 y el 80% en función de la universidad las clases son en catalán [9].

En la asignatura SPD, objeto de este artículo, el idioma vehicular es el catalán, por lo que la mayoría de actividades docentes se realizan en esta lengua, siendo asumido de forma natural por el alumnado.

En la UPC, el 60% de las asignaturas utiliza el catalán como idioma vehicular docente.

4.1. Clases expositivas

Las clases expositivas de SPD se realizan en catalán. Difícilmente un estudiante que provenga del estado español y se enfrente de origen con este idioma tiene dificultades para en poco tiempo entender con suficiencia las explicaciones del profesor. No obstante, otro caso distinto es el de los estudiantes Erasmus que provienen de países europeos, que llegan a la Universidad con la idea errónea de que las clases son exclusivamente en español y con una determinada preparación del mismo. Si provienen de países latinos como Francia, Portugal o Italia se adaptan al catalán con facilidad. Si provienen de países sajones como Suecia tienen mayor dificultad.

En cualquier caso se les aconseja el uso de los Servicios Lingüísticos de la Universidad que disponen de programas de acogida para que el paso del español al catalán sea lo más suave posible. Aquellos que lo utilizan, en pocos días se encuentran más cómodos con el uso del catalán para atender a las clases.

4.2. Documentación

La documentación de la asignatura está fundamentalmente escrita en catalán, y en determinados apartados que provienen de documentación original en español o inglés se mantienen su idioma.

Ante dificultades planteadas por los estudiantes extranjeros en el entendimiento de la documentación en catalán se les recomienda la utilización de traductores automáticos [3].

El más recomendado es el que proporciona el Instituto Cervantes que permite la traducción en los dos sentidos entre los idiomas catalán, gallego, español, inglés y francés, de cualquier longitud de texto y completamente gratuito [2], aunque recientemente la Generalitat de Catalunya ha puesto en servicio un traductor automático gratuito de catalán a francés o inglés y viceversa de gran potencia y con resultados satisfactorios y que se ha empezado a recomendar también [4].

4.3. Campus virtual

El Campus Virtual utilizado se denomina ATENEA y en su definición están presentes tanto el idioma catalán como el español y el inglés. Por tanto se puede utilizar en cualquiera de los tres idiomas, aunque cuando se hace referencia a la asignatura SPD en concreto, todos aquellos documentos que se incluyen pueden estar en cualquiera de los tres idiomas. En este caso, el propio Campus Virtual permite acceder a los traductores automáticos.

4.4. Exámenes y libro de problemas

Los exámenes y el libro de problemas están siempre redactados en catalán, aunque se permite la resolución de los mismos tanto en catalán como en inglés o español.

Los profesores que vigilan los exámenes están pendientes de forma especial de los estudiantes con movilidad, con los cuales ya se ha mantenido durante el curso un seguimiento de su aprovechamiento lingüístico, para ayudarles a comprender el contenido del examen.

Los estudiantes, a los que se les ha advertido con anterioridad esta disponibilidad, la usan con soltura, discreción y rapidez. Los profesores están muy atentos para que la comprensión del examen no suponga un handicap en su rendimiento.

Los exámenes utilizan un formato idéntico a los problemas que los alumnos tienen en su libro de ejercicios, lo cual facilita la comprensión de los mismos.

4.5. Consultas

Las consultas se realizan en el idioma que resulte más cómodo para el estudiante. Se intenta que se hagan en catalán, pero ante cualquier dificultad de entendimiento o expresión, se utiliza el español o bien el inglés.

Se procura que las consultas de estudiantes extranjeros se hagan simultáneamente con estudiantes españoles para facilitar su integración lingüística y la comprensión.

4.6. Trabajo en grupo

En el contenido del trabajo se permite la redacción tanto en catalán, español o inglés.

El trabajo en grupo se procura que se realice entre estudiantes Erasmus de la misma nacionalidad, y si no es posible se sugiere su inclusión en grupos exclusivamente Erasmus.

Aunque parece que para una mayor integración sería mejor incluir estudiantes extranjeros con nacionales, la experiencia aconseja no hacerlo así ya que dificulta de forma

especial la realización del trabajo por parte de los estudiantes foráneos y disminuye su participación en el mismo, aspecto contradictorio con el carácter de trabajo cooperativo base de la metodología docente empleada.

5. Resultados

En los seis últimos años han pasado 28 estudiantes con movilidad por la asignatura lo que supone una media de 2-3 alumnos por semestre. Siguiendo la metodología docente explicada y la flexibilidad aportada en la resolución de cada uno de los entornos, nunca se ha planteado ningún conflicto relacionado con el idioma docente, y los estudiantes han salido satisfechos de su experiencia lingüística. Para ello se ha hecho un seguimiento personalizado de sus impresiones una vez finalizado el curso.

En relación a los resultados académicos de los estudiantes de movilidad y siguiendo la metodología de la Facultad de Informática de Barcelona en las mediciones, se constata que ha habido un 62 % de aprobados sobre los presentados y un 10 % de no presentados, lo que supone una media de aprobados del 70 % de esta tipología de estudiantes que coincide con la media general de la asignatura.

En la tabla 1 se indican los orígenes de los estudiantes que han cursado la asignatura SPD en la UPC y que están considerados como estudiantes de movilidad. Incluye alumnos Erasmus europeos, sudamericanos y de comunidades autónomas españolas no catalanoparlantes.

Estudiantes	Origen	Aprobados
3	Colombia	2
3	Suiza	3
7	Italia	5
8	Portugal	6
1	Grecia	0
1	Francia	1
1	Suecia	0
4	País vasco	4

Tabla 1. Distribución estudiantes con movilidad

6. Conclusiones

El idioma docente utilizado en los estudios puede influir de forma importante en la movilidad de los estudiantes europeos en el marco del EEES, y no hay una postura oficial de cómo se debe garantizar el derecho a la movilidad reconocido en la declaración de Bolonia, dejando en manos del profesor la decisión de cómo se debe aplicar el idioma docente en las aulas.

Las Universidades están desarrollando actividades tendentes a resolver la problemática generada entre movilidad e idioma docente, todas ellas centradas en proporcionar los medios lingüísticos necesarios, en habilitar procedimientos de acogida y en incentivar la realización de cursos en inglés.

Los resultados docentes obtenidos en SPD con los estudiantes con movilidad concluyen que el idioma no ha supuesto para ellos una mayor dificultad en el proceso de aprendizaje.

La experiencia desarrollada en estos últimos cursos en SPD indica que el problema se debe afrontar con un seguimiento personalizado y con la flexibilidad docente necesaria, para que manteniendo la presencia cultural propia del catalán, sin menoscabo del español y el inglés, se pueda mantener la calidad docente y resultados de aprendizaje deseados, consiguiendo en definitiva una situación multilingüe reflejo de la realidad europea.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el MEC bajo el proyecto CATARO (TES 2005-08051-C03-01).

Los autores agradecen el soporte recibido por el ICE (Instituto de Ciencias de la Educación) de la Universidad Politécnica de Cataluña para la utilización de ATENEA, el Campus Virtual de la UPC.

Referencias

[1] https://atenea.upc.edu/moodle/login/index.php

[2] http://oesi.cervantes.es/traduccionAutomatica.html

[3] Delgado, Ana María; Oliver Cuello, Rafael y Salomón Sancho, Lourdes, *El aprendizaje multilingüe en un aula virtual.*

www.formatex.org/micte2006/pdf/401-405.pdf
[4] http://traductor.gencat.net/index_es.jsp
[5] Ministerio de Educación. *La integración del sistema universitario español en el espacio europeo de enseñanza superior*. Documento marco. Febrero 2003. www.mec.es
[6] Santos Boada, Germán y Solé Pareta, Josep, *Informática y Telecomunicaciones: el acicate de la sobrenota.* JENUI2006. Universidad de Deusto. Bilbao, julio 2006, páginas 161-167.
[7] Santos Boada, Germán y Solé Pareta, Josep, *Evaluación participativa en ingeniería informática*. Jornadas nacionales de intercambio de experiencias piloto de implantación de metodologías ECTS. Universidad de Extremadura Badajoz. Septiembre 2006.
[8] Santos Boada, Germán y Solé Pareta, Josep, *Positive interdependence and participative assessment as cohesive elements in group teaching.* 8th International Symposium on Computers in Education SIEE06. Universidad de León. Octubre 2006.
[9] http://www2.upc.edu/slt/ajudam/where/pmf.htm

Cómo congeniar los exámenes y los proyectos en asignaturas PBL[1]

Pablo del Canto, Isabel Gallego, Rubén Hidalgo, Johann López, José Manuel López, Javier Mora, Eva Rodríguez, Eduard Santamaria, Miguel Valero
Departamento de Arquitectura de Computadores, Escola Poltècnica Superior de Castelldefels
Universitat Politècnica de Catalunya
Av. del Canal Olímpic, s/n. Castelldefels (08860)
pcanto@ac.upc.edu, isabel@ac.upc.edu, rhidalgo@ac.upc.edu, johannnl@ac.upc.edu, jolopez@ac.upc.edu, mora@cimne.upc.edu, evar@ac.upc.edu, esantama@ac.upc.edu, miguel.valero@upc.edu

Resumen

La ponencia describe un nuevo método de evaluación para asignaturas diseñadas siguiendo la metodología del aprendizaje basado en proyectos (PBL). En asignaturas PBL con frecuencia los alumnos nos sorprenden con sus proyectos, pero nos decepcionan con sus exámenes. El método de evaluación que proponemos pretende que exámenes y proyectos coexistan de forma armoniosa en asignaturas PBL. Para ello, proponemos que uno los elementos del método de evaluación sean las pruebas de mínimos en vez de los exámenes tradicionales. En las pruebas de mínimos los alumnos han de demostrar que han adquirido unos ciertos conocimientos mínimos que han sido definidos a partir de los objetivos mínimos de aprendizaje que se han de alcanzar en la asignatura.

1. Introducción

En esta ponencia se describe un nuevo método de evaluación para la asignatura de Introducción a los Computadores (IC) [1], destinada al aprendizaje de la programación de ordenadores, de las titulaciones de Ingeniería Técnica de Telecomunicación e Ingeniería Técnica Aeronáutica que se imparten en la Escuela Politécnica Superior de Castelldefels [2] de la Universidad Politécnica de Cataluña.

La asignatura IC ha sido adaptada al sistema de crédito europeo (ECTS) y ahora tiene asignados 4.6 ECTS lo que implica unas 128 horas de dedicación total de los alumnos, distribuidas a lo largo de 16 semanas. Por tanto, la dedicación del alumno es de unas 8 horas por semana. De estas 8 horas de trabajo semanal, 2 corresponden a una sesión de clase y el resto son horas de trabajo adicional fuera de clase, para realizar tareas individuales o en grupo. Algunas semanas la sesión de clase tiene 4 horas y fuera de clase los alumnos deben dedicar otras 4 para completar las 8 semanales.

La asignatura IC, que se imparte el primer cuatrimestre del primer curso del plan de estudios, representa para los alumnos un primer contacto con el mundo de la programación de ordenadores. El objetivo de esta asignatura es que los estudiantes aprendan a construir programas sencillos utilizando el lenguaje de programación C. Las actividades que se realizan en esta asignatura durante la primera mitad del curso están destinadas a adquirir los conocimientos elementales que permitirán realizar programas sencillos. En la segunda mitad del curso, se realiza un proyecto, que consiste en diseñar e implementar un programa de tamaño medio, en grupo, que deberá funcionar perfectamente al final del curso. En realidad, durante esta segunda mitad del curso se continúan haciendo actividades similares a las de la primera mitad del curso, pero la novedad importante es ahora los alumnos deben aplicar lo que aprenden al proyecto de programación, de acuerdo con los compañeros del grupo.

Los alumnos en general se sienten motivados por los proyectos que les proponemos, y al estar diseñados de forma que pueden añadir funcionalidades adicionales, muchos grupos realizan proyectos brillantes. Pero cuando han de realizar un examen tradicional, los resultados no han sido tan buenos como hubiéramos esperado en función de los proyectos que han realizado. Esta situación nos desconcierta ya que la realización del proyecto es un medio para que aprendan, y sus resultados se tendrían que ver reflejados en los resultados de los exámenes. Sin embargo, un análisis más detallado de la cuestión pone de manifiesto que la realización de un proyecto no es

[1] Este trabajo ha sido realizado con la ayuda del Departamento de Arquitectura de Computadores y de la Escuela Politécnica Superior de Castelldefels

la mejor manera de preparar un examen final tradicional y viceversa, un examen final no es la mejor manera de evaluar todo lo aprendido en la realización de un proyecto. Por tanto, los proyectos y los exámenes tradicionales conviven mal en nuestras asignaturas. De hecho, este problema se observa de forma habitual en otros escenarios PBL [3] en los que los alumnos nos sorprenden con sus proyectos, pero nos decepcionan con sus exámenes [4].

En esta ponencia se discuten diferentes métodos de evaluación para que los exámenes y los proyectos convivan de forma equilibrada, sin molestarse, en asignaturas diseñadas siguiendo la metodología del PBL.

2. Proyectos y exámenes en una asignatura de programación

En esta sección presentamos las diferentes alternativas que nos planteamos para congeniar exámenes y proyectos en asignaturas PBL. Las tres opciones que nos planteamos fueron suprimir los exámenes; exigir una nota mínima en el examen tradicional o que el examen estuviera basado en los conocimientos mínimos.

La primera opción que consideramos fue la de no realizar el examen. Estamos tan convencidos de que la realización adecuada de los proyectos conduce a los alumnos inexorablemente al aprendizaje que no es necesario verificar ese aprendizaje a través de un examen, que por otra parte parece que no es el instrumento más adecuado. De hecho, el examen tradicional es un instrumento casi exclusivo del entorno académico, que poco tiene que ver con el entorno profesional para el que preparamos a nuestros alumnos.

Esta opción se descartó porque no nos da elementos de juicio para asegurar que todos los miembros del grupo han alcanzado un aprendizaje mínimo. Es evidente que el hecho de que un proyecto en grupo esté muy bien no garantiza que todos los miembros del grupo hayan trabajado y aprendido lo suficiente.

La segunda opción consistía en seguir realizando exámenes convencionales pero fijando una nota mínima, lógicamente inferior a 5, que los alumnos tendrían que superar en los exámenes para aprobar la asignatura. Esta opción ha sido aplicada en muchas ocasiones a asignaturas que combinan trabajos o proyectos con exámenes. El método de evaluación para estas asignaturas consiste en dar un peso en la nota final a los trabajos y al examen, fijando una nota mínima para cada una de estas dos partes. La decisión sobre cuál debe ser la nota mínima produce siempre una cierta tensión entre alumnos, profesores y responsables académicos. Lo habitual es que esa nota mínima acabe situándose entre el 3,5 y el 4.

Lo cierto es que cuando se aplica este método de evaluación, si el alumno aprueba la asignatura gracias a la nota del proyecto, con una baja nota de exámenes (por ejemplo un 3,5), el profesorado sigue con la duda de que los alumnos hayan aprendido lo suficiente (precisamente una calificación de 3,5 indica que no ha aprendido). En definitiva, este esquema no mejora significativamente la sensación del profesorado, en relación al esquema anterior.

Finalmente la tercera opción que nos planteamos fue la de definir unos conocimientos mínimos (y no una calificación mínima) que los alumnos tendrán que demostrar para aprobar la asignatura, con independencia del resultado del proyecto. Estos conocimientos mínimos establecen una cota inferior para el aprendizaje, y son fijados a partir de los objetivos mínimos de aprendizaje que se han de alcanzar en la asignatura. Por tanto, para que un alumno supere la asignatura ha de demostrar que ha adquirido todos los conocimientos mínimos en las diferentes pruebas individuales de mínimos que realizamos durante el curso, y por supuesto, debe realizar de forma adecuada y en grupo el proyecto de la asignatura.

En las próximas secciones desarrollamos la idea de los conocimientos mínimos y mostramos su aplicación en nuestra asignatura.

3. Como identificar los conocimientos mínimos

Los conocimientos mínimos para IC han sido definidos a partir de los objetivos específicos y del temario de la asignatura, que se describen en las dos secciones siguientes.

3.1. Temario de IC

El temario es la lista organizada de los temas que son relevantes para conseguir los objetivos de una

asignatura. El temario de la asignatura de IC es el siguiente:

- Estructura de un programa, datos, variables y expresiones
- Las sentencias básicas: Asignación; Sentencias condicionales; Sentencias iterativas
- Tipos de datos elementales: Enteros; Caracteres; Reales
- Tipos de datos estructurados: Vectores; Matrices; Estructuras
- Procedimientos y funciones: Definición e invocación; Paso de parámetros; Variables locales
- Ficheros: Operaciones para abrir, cerrar, leer y escribir fichero de texto
- Esquemas algorítmicos: Recorrido; Búsqueda
- El entorno de programación Visual C++: Edición, compilación, montaje y ejecución de proyectos; Ejecución paso a paso; Uso de breakpoints; Visualización de variables, estructuras de datos y ficheros

3.2. Objetivos específicos de IC

A continuación presentamos una lista con los objetivos específicos que describen en detalle todo lo que deben aprender los alumnos en relación a la programación en C, clasificados por temas:

1. Tipos de datos elementales y sus operaciones (enteros, caracteres y reales)
   - Describir los tipos de datos elementales y las operaciones que actúan sobre ellos
   - Escribir la declaración de datos de cualquiera de los tipos elementales
2. Sentencias básicas (asignación, condicionales, iterativas y de entrada/salida)
   - Describir el funcionamiento de las sentencias básicas
   - Predecir el resultado de una secuencia de sentencias básicas
   - Codificar una tarea convenientemente especificada, utilizando la secuencia de sentencias básicas adecuada
3. Tipos de datos estructurados (vectores, estructuras, matrices y combinaciones)
   - Describir las estructuras de datos fundamentales, y las operaciones típicas sobre ellas
   - Escribir la declaración de una estructura de datos convenientemente especificada
   - Escribir el código necesario para acceder a un elemento o conjunto de elementos de una estructura de datos
   - Elegir la estructura de datos más adecuada para una aplicación determinada
4. Esquemas algorítmicos (recorrido y búsqueda)
   - Explicar los esquemas de recorrido y búsqueda
   - Adaptar los esquemas de recorrido y búsqueda a una situación convenientemente especificada, identificando con claridad cada uno de los elementos del esquema
   - Elegir el esquema adecuado para resolver un problema determinado
   - Aplicar por iniciativa propia los esquemas algorítmicos estudiado
5. Procedimientos y funciones
   - Describir los conceptos de procedimiento y función, la diferencias entre ellos, y su utilidad
   - Definir los conceptos: cabecera de procedimiento o función, parámetros formales, variables locales, resultado de la función, activación del procedimiento o función, parámetros reales, paso de parámetros
   - Describir la diferencia entre paso de parámetros por valor o por referencia
   - Codificar convenientemente una llamada a procedimiento o función, pasando correctamente los parámetros Codificar en forma de procedimiento o función una tarea convenientemente especificada, estableciendo adecuadamente los parámetros necesarios
   - Proponer una organización en bloques (procedimientos o funciones) de una aplicación determinada
6. Ficheros
   - Explicar el concepto de fichero, para qué sirve, y cuáles son las operaciones típicas sobre ficheros de texto (crear, abrir, leer, escribir, preguntar por fin de fichero y cerrar)
   - Escribir las sentencias necesarias para realizar las operaciones básicas con fichero de texto

- Escribir las sentencias necesarias para determinar el tipo de error que se ha producido al realizar una operación con un fichero
- Determinar el formato adecuado para almacenar los datos en un fichero de texto, en función de las necesidades de la aplicación

Además de estos objetivos, la asignatura IC tiene también otros objetivos de carácter transversal, relacionados con la capacidad para trabajar en grupo, para aprender de forma autónoma o para comunicarse con los demás de forma efectiva. Estos objetivos no se enumeran aquí porque no están relacionados directamente con el objeto de la ponencia.

4. Conocimientos mínimos para IC

Para la asignatura de IC hemos identificado ocho conocimientos mínimos

Los conocimientos mínimos han sido definidos en función de los objetivos mínimos de aprendizaje que se han de alcanzar en la asignatura que hemos presentado en la sección anterior. A continuación detallamos los conocimientos mínimos definidos:

- *Conocimiento mínimo 1:* Escribir un programa que lea información de línea de comandos (caracteres, enteros, reales o palabras), tome decisiones a partir de esa información (usando solamente sentencias condicionales) y escriba en la pantalla los resultados (enteros, reales, palabras, frases).
- *Conocimiento mínimo 2:* Escribir un programa que realice un recorrido o una búsqueda en una secuencia de dígitos, números consecutivos o números leídos del teclado.
- *Conocimiento mínimo 3:* Escribir un programa que haga un recorrido o una búsqueda en un vector o matriz de caracteres o números.
- *Conocimiento mínimo 4:* Escribir un programa que requiera la combinación de dos esquemas algorítmicos (recorrido y/o búsqueda) aplicados sobre un vector o matriz de enteros o caracteres.
- *Conocimiento mínimo 5:* Diseñar la estructura de datos más adecuada para representar la información de un problema dado.
- *Conocimiento mínimo 6:* Codificar un procedimiento o función que cargue una estructura de datos con la información contenida en un fichero de texto o que salve en un fichero de texto la información contenida en la estructura.
- *Conocimiento mínimo 7:* Escribir un procedimiento o función que recorra, busque o añada información en un vector de estructuras.
- *Conocimiento mínimo 8*: Escribir un programa principal que realice llamadas a procedimientos o funciones pasándole adecuadamente los parámetros, leyendo del teclado la información necesaria para realizar las llamadas y escribiendo en pantalla la información correspondiente a los resultados de las llamadas.

5. Cómo se presentan los conocimientos mínimos a los alumnos

Durante la presentación de la asignatura que realizamos el primer día de clase, cuando introducimos el método de evaluación hacemos hincapié en el hecho de que para superar la asignatura tendrán que demostrar que han adquirido unos conocimientos mínimos en diferentes pruebas que realizarán a lo largo del curso y que para superar la asignatura tendrán que demostrar que han adquirido como mínimo 7 de los 8 conocimientos mínimos de la asignatura.

También indicamos a los alumnos que en la intranet de la asignatura tienen el recurso Conocimientos Mínimos, con ejemplos y soluciones para cada uno de los conocimientos mínimos de la asignatura.

Para cada conocimiento mínimo hemos preparado tres ejemplos que representan los ejercicios tipo que se podrían encontrar en las pruebas de mínimos y para cada ejemplo también disponen de la solución.

La Tabla 1 presenta los ejemplos que hemos preparado para los conocimientos mínimos 2 y 4.

Conocimientos mínimos – Ejemplos con sus soluciones	
Conocimiento mínimo 2 ***Escribir un programa que realice un recorrido o una búsqueda en una secuencia de dígitos, números consecutivos o números leídos del teclado.***	
Ejemplos	**Solución**
2.1 Realizar un programa que pida un número entero positivo por teclado y diga cuál es su mayor digito par y su menor digito impar.	solución
2.2 Realizar un programa que determine la suma de los 10 primeros múltiplos de 6.	solución
2.3 Identificar el mayor de los tres primeros múltiplos de 7 en la secuencia de números escritos por teclados, acabada en 0.	solución
Conocimiento mínimo 4 ***Escribir un programa que requiera la combinación de dos esquemas algoritmicos (recorrido y/o búsqueda) aplicados sobre un vector o matriz de enteros o carácteres.***	
Ejemplos	**Solución**
4.1. Escribir un programa que lea del teclado un número enero y lo inserte en un vector de enteros que está ordenado de menor a mayor, de forma que el vector se mantenga ordenado de menor a mayor.	solución

4.2 Suponer las siguientes declaraciones:
`#define MAX 20`
`int vector [MAX];`
Suponer que el vector ya contiene números enteros en todas las posiciones. Escribir un programa que elimine del vector el primer múltiplo de 7 que encuentre. Para eliminar un número hay que mover todos los que tiene a su derecha una posición a la izquierda.
	solución

4.3 Suponer las siguientes declaraciones:
`#define MAX 20`
`char frase [MAX];`
Suponer que el vector frase ya contiene caracteres en todas las posiciones. Escribir un programa que inserte un carácter en blanco después de la primera 'j' que encuentre
	solución

Tabla 1. Ejemplos de ejercicios para verificar conocimientos mínimos, tal y como se presentan a los alumnos

Una de las tareas que proponemos realizar a los estudiantes durante la semana 6 del curso, para que se preparen para la primera prueba de mínimos que realizará en la semana 7, es un ensayo de la primera prueba de mínimos y una autoevaluación. Les proporcionamos un enunciado tipo con un ejercicio para cada uno de los cuatro primeros conocimientos mínimos que tendrán que resolver y la solución oficial. Una vez hayan resuelto el ejercicio se tendrán que autoevaluar en función de los criterios establecidos y resolverán si su código es correcto (bien, regular o mal) y si es claro (bien, regular o mal).

Además les proponemos que realicen varios ensayos con los ejercicios tipo que les hemos preparado y que los comparen con las soluciones que les proporcionamos.

Del mismo modo durante la semana 13 les proponemos que se preparen para la segunda prueba de mínimos, que se realiza durante la semana 14. En esta prueba tendrán que demostrar que han adquirido los conocimientos mínimos relacionados con estructuras, procedimientos y funciones y ficheros que se corresponden con los mínimos del 5 al 8. La Tabla 2 muestra un ejemplo de ejercicios para verificar los conocimientos mínimos 5 y 7.

En la intranet de la asignatura podemos encontrar tres ejemplos para cada conocimiento mínimo.

Conocimientos mínimos – Ejemplos con sus soluciones	
Conocimiento mínimo 5 ***Diseñar la estructura de datos más adecuada para representar la información de un problema dado.***	
5.1. Diseñar una estructura de datos para que contenga la siguiente información: título del CD, autor, ranking en la lista de mas vendidos y precio de CD.	solución
Conocimiento mínimo 7 ***Escribir un procedimiento o función que recorra, busque o añada información en un vector de estructuras.***	
7.1. Partiendo de las mismas declaraciones que el ejercicio 5.1, escribir el código de una función que tendrá la siguiente cabecera: `int posición (Ttienda t, TCD mi_cd);` La función debe retornar la posición que ocupa `mi_cd` en el vector de CD que tiene la estructura Ttienda. Si `mi_cd` no está en la tienda entonces la función debe retornar un -1.	solución

Tabla 2. Ejemplos de ejercicios para verificar conocimientos mínimos, tal y como se presentan a los alumnos

6. Método de Evaluación

El esquema de evaluación para IC se muestra en la Tabla 3

Elemento	Peso
Entregas del curso	20%
Conocimientos mínimos	30%
Proyecto Versión 1	10%
Proyecto Versión 2	20%
Ejercicio individual ampliación del Proyecto	10%
Actitud y participación	10%

Tabla 3. Esquema de evaluación IC

Las características más relevantes del esquema de evaluación de IC se detallan a continuación. Las entregas son el resultado de las tareas que realizan los alumnos a lo largo del curso (por ejemplo, el informe de autoevaluación del ensayo de mínimos que se ha mencionado antes). En IC se valora un 20% el mero hecho de realizar las entregas a tiempo, con independencia de si están bien o mal. Es un elemento de la calificación que usamos para estimular a los alumnos a que recorran el camino que hemos preparado para ellos. De hecho, si no realizan el 80% de las tareas del curso no podrán superar la asignatura y la calificación final de la asignatura será no presentado. Si realizan todas las tareas a tiempo o solamente se retrasan en la entrega de una tarea obtendrán un 10. Mientras que si no realizan una o dos tareas o se retrasan en la entrega de dos o tres obtendrán un 7,5, en el resto de casos, siempre que hayan realizado más del 80%, obtendrán un 5 en esta componente de la nota.

La calificación del proyecto la obtenemos a partir de tres elementos. El desarrollo de proyecto que consta de dos entregas, una primera parcial y un prototipo final, que representan un 10% y 20% respectivamente de la nota final y que se califican en función de ciertos criterios establecidos para la evaluación del proyecto. La calificación de las dos versiones del proyecto es la misma para todos los miembros de grupo. El tercer elemento es un ejercicio de ampliación individual del proyecto sobre ordenador, en la que cada miembro del grupo deberá realizar una pequeña ampliación del proyecto, mediante la que podrá demostrar que conoce perfectamente como está organizado y construido el mismo. Se pretende evitar así que un miembro del grupo se ocupe sólo de su parte, desentendiéndose del resto. La calificación de esta componente se asigna de la forma siguiente:

- 0 si no se hace bien la ampliación
- 5 si se hace bien
- 10 si todos los miembros del grupo la hacen bien

Respecto a la calificación de los conocimientos mínimos cabe destacar que para aprobar la asignatura será necesario demostrar al menos 7 de los 8 conocimientos mínimos en las pruebas escritas que se realizarán las semanas 7 y 14 del curso y en la semana de exámenes finales. En caso contrario no se aprobará la asignatura. En la prueba de la semana 7 los alumnos pueden demostrar los conocimientos mínimos del 1 al 4.

En la prueba de la semana 14 pueden demostrar los conocimientos mínimos del 5 al 8, y los que no demostraron en la primera prueba. En la semana de exámenes finales se realiza una última prueba en la que los alumnos pueden demostrar los mínimos que no hayan superado en las pruebas anteriores.

La calificación de la componente de conocimientos mínimos se determina en función de las reglas siguientes:

Para superar la asignatura es necesario superar al menos 7 de los 8 conocimientos mínimos. En caso contrario se suspende, con independencia del proyecto, y la calificación de la asignatura estará entre 0 y 3,5, en función del número de mínimos demostrado.

Si se superan los 7 ó 8 mínimos, la calificación será:

10 si se demuestran todos, sin necesidad de la última prueba

8,5 si se demuestran todos, pero se necesita la última prueba

7,5 si se demuestran 7, sin necesidad de la última prueba

5 si se demuestran 7, pero se necesita la última prueba

Finalmente, en la actitud y participación se valora la participación frecuente en el Foro de Consultas, la seriedad en las tareas de autoevaluación y evaluación de compañeros, la asistencia regular a clase y la observación del comportamiento en el grupo.

7. Consideraciones sobre la calificación de los mínimos

Una característica importante del esquema de evaluación propuesto es que los ejercicios de mínimos no se califican en una escala de 0 a 10. Basta con determinar si el ejercicio está bien o no, ya que la calificación de los conocimientos mínimos se establece en base al número de conocimientos superados, y en el número de pruebas que se han necesitado para superarlos.

La prueba de mínimos se realiza sobre papel y en determinadas ocasiones pueden surgir algunas dudas a la hora de determinar si un ejercicio con un pequeño fallo es correcto o no. Entonces lo que valoramos es si el pequeño fallo lo podrían resolver fácilmente si dispusieran de un ordenador, en cuyo caso se da el mínimo por superado.

Para confirmar que, a pesar de una falta de criterios explícitos para la corrección de los mínimos, diferentes profesores de la asignatura tomarían las mismas decisiones en la valoración de las respuestas de los alumnos, realizamos el siguiente experimento. Cuatro de los profesores de la asignatura corregimos la segunda prueba de mínimos de un grupo de la asignatura de IC, sin necesidad de fijar criterios explícitos. La hipótesis es que los cuatro coincidiríamos en gran medida sobre qué ejercicios están superados y cuáles no. Los resultados del experimento fueron los siguientes: en un 68,81% de los ejercicios los cuatro profesores coincidían en la valoración (pasa o no pasa). En un 23,65% de los casos sólo tres de los cuatro profesores coincidían y en un 7,52% de los casos sólo coincidían dos profesores.

Ciertamente, esperábamos un mayor nivel de coincidencia, por lo que probablemente en la próxima ocasión fijaremos de forma explícita algún criterio que ayude a aumentar el grado de coincidencia.

Por otra parte, el decidir simplemente si el mínimo se pasa o no (sin tener que adjudicar una calificación entre 0 y 10) hace que el tiempo necesario para la evaluación se reduzca de forma muy notable. Se da incluso el caso de que durante las pruebas de mínimos, en las que hay alumnos que acaban mucho antes que otros (porque son más rápidos o porque tenían menos mínimos para demostrar), el profesor que supervisa el examen puede ir corrigiendo a medida que entregan, y finalizado el tiempo de examen ya sólo quedan por corregir una pequeña parte de ejercicios (los entregados al final de la sesión).

8. Resultados obtenidos

A continuación detallamos los resultados para las pruebas de conocimientos mínimos obtenidos durante el curso 2006-2007Q1 en la asignatura de IC. A la hora de elaborar los resultados solo hemos tenido en cuenta los alumnos que han seguido la asignatura durante todo el cuatrimestre, y por tanto han realizado el proyecto y han entregado como mínimo el 80% de los entregables. Los resultados que hemos obtenido son:

- 50% de los estudiantes han demostrado todos los conocimientos mínimos a la primera
- 24,48% de los estudiantes han demostrado todos los conocimientos mínimos, pero han necesitado la convocatoria de la semana de exámenes finales
- 6,12% de los estudiantes han demostrado todos los conocimientos menos uno sin necesidad de la convocatoria de la semana de exámenes finales
- 9,18% de los estudiantes han demostrado todos los conocimientos mínimos menos uno, usando la convocatoria de la semana de exámenes finales
- 10,2% de los estudiantes han realizado el proyecto bien, pero no han demostrado los mínimos y, por tanto, suspendieron.

Los resultados obtenidos muestran que una gran parte de los estudiantes que siguen la asignatura, realizan las tareas que proponemos durante el curso y un buen proyecto, demuestran que han adquirido los conocimientos mínimos de la asignatura. Solamente un 10,2 por ciento de los estudiantes que han seguido la asignatura no han demostrado que han adquirido los conocimientos mínimos.

Si comparamos estos resultados con los obtenidos en otros cursos en los que realizábamos exámenes tradicionales, vemos que el número de estudiantes que superan los mínimos es superior al de estudiantes que superaban el examen. Por tanto, una de las percepciones que tenemos es que la evaluación basada en mínimos favorece a los estudiantes menos brillantes.

9. Conclusiones

En esta ponencia hemos descrito cómo congeniar los exámenes y proyectos en asignaturas diseñadas siguiendo la metodología del aprendizaje basado en proyectos. En concreto el método de evaluación para estas asignaturas ha sido diseñado en función de la adquisición de unos conocimientos mínimos definidos a partir de de los objetivos mínimos de aprendizaje que se han de alcanzar. El método pretende garantizar que todos los alumnos alcancen unos objetivos mínimos de aprendizaje, pero sin interferir con el esfuerzo que necesitan para realizar los proyectos del curso, cosa que suele pasar cuando se pretende que los alumnos hagan proyectos y, al mismo tiempo, exámenes tradicionales (en los que se pretende que los alumnos demuestren todos los objetivos del curso).

En función de los resultados obtenidos al sustituir el examen tradicional por la prueba de mínimos, si simplemente comparamos los enunciados de ambos tipos de pruebas podría dar la sensación que estamos bajando el nivel de exigencia de la asignatura. Pero, se ha de tener en cuenta que en las pruebas de mínimos exigimos que realicen correctamente los ejercicios y que han de demostrar todos los mínimos (o todos menos uno) para poder superar la asignatura. Por otra parte, la valoración del nivel no puede hacerse sin tener en cuenta los proyectos que realizan los alumnos, que en la mayoría de los casos satisfacen plenamente las expectativas.

Referencias

[1] La páginas web de IC puede consultarse en atenea.upc.edu (entrar como visitante y buscar el curso *Introducció als Computadors*, de la EPSC)

[2] Escola Politècnica Superior de Castelldefels (EPSC), http://www.epsc.upc.edu

[3] D.R. Woods et al., The future of engineering education. Developing Critical Skills. *Chem. Engr. Education,* 34 (2), 108-117.

[4] Jesús Alcocer, Silvia Ruíz y Miguel Valero-García, Evaluación de la implantación de aprendizaje basado en proyectos en la EPSC (2002-2003) *XI Congreso Universitario de Innovación Educativa en Enseñanzas Técnicas*, julio 2003

Cómo transformar e impartir una asignatura bien adaptada a ECTS sin morir en el intento: patrones para la reducción del trabajo del profesor

Ray Fenández Rupérez, Miren Bermejo Llopis
Dpto. Lenguajes y Sistemas Informáticos
Universidad del País Vasco / Euskal Herriko Unibertsitatea
Paseo M. Lardizabal 1, 20018 Donostia
{ray.fernandez, miren.bermejo}@ehu.es

Resumen

La adaptación de nuestras titulaciones al EEES nos está llevando a plantear modelos docentes en los que el alumnado pasa a ocupar el centro de la asignatura, contando con el profesor como guía o, en general, a plantear tareas distintas de las tradicionales: *preparar lo que va a decir la profesora*, *ir a clase*, *post-procesar lo que ha dicho la profesora* y *estudiar los apuntes/libros*. El escenario más habitual es el uso de tareas o *microexámenes*: trabajos asignados a cada alumno o cada equipo con evaluación de resultado; y esto dispara al alza el trabajo del profesor.

En un modelo simple pero suficiente para la mayoría de los casos, la dedicación del docente se puede cuantificar como: *A* número de horas de clase + B * número de alumnos + C * número de grupos*. Este artículo estudia cómo reducir esa dedicación mediante patrones de diseño de la asignatura, mejorando la formación de competencias del alumno y manteniendo o mejorando la calidad docente. Para ello se analiza la estructura de A, B, C y se comparan tres escenarios: una asignatura tradicional, otra basada en trabajos asignados a los alumnos y esta misma pero utilizando los patrones que aquí se presentan.

Todo esto se ilustra con el ejemplo de la asignatura Planificación y Gestión de Proyectos Informáticos (PGPI) para la cual los profesores han propuesto una docencia orientada a tareas en la que la carga de trabajo del docente es similar o menor a la de una asignatura tradicional.

1. Introducción

Medir una asignatura tradicional en ECTS es posible, incluso fácil. Tomemos por ejemplo una asignatura de 6 créditos que hoy consiste en impartición de teoría y un examen final. Basta con seguir dos pasos. Paso 1: listar las competencias específicas; es algo que puede hacerse bien en no mucho tiempo. Paso 2: declarar que la asignatura usa dos tipos de tarea, y que ambos tipos se usan en todas las competencias: (t1) tarea *exposición teórica*, con un número de horas para el alumno igual a las que ahora tiene la asignatura en presencialidad (habitualmente 10 por crédito); (t2) tarea de tipo *preparación y asimilación de los conceptos teóricos*, con un coeficiente 1,5 (es decir, 1,5 horas de preparación y asimilación por cada hora de clase). Como resultado obtenemos una flamante asignatura con competencias y tareas y medida en ECTS.

Pero también es posible cambiar simultáneamente el núcleo de la estrategia didáctica utilizada hasta el momento para que sea el alumno el protagonista del aprendizaje[1]. Es decir, (1) preparar tareas que no se reduzcan a atender las explicaciones dadas en clase y después estudiar las notas tomadas, (2) que esas tareas aporten de forma bien definida a las competencias a lograr en el alumno, y (3) evaluar cada tarea por separado. Normalmente esto transforma la asignatura en una sucesión de labores, a realizar por parte del alumno, que necesitan una evaluación posterior.

Pero, al hacer esto, el trabajo del profesor se dispara (ver apartado 4). A un docente dedicado a tiempo completo a dar clase e investigar, esto le puede afectar a su investigación. ¿Debe optar

[1] Los autores de este artículo opinan que no existe otra forma *correcta* de adaptar una asignatura al EEES, o de medir *bien* en ECTS. Tratar tal hipótesis no es objeto de este artículo, por lo que se ha evitado toda alusión a la misma y a sus consecuencias.

entre *dar clase bien* o investigar? Aún más, cuando el profesor es asociado a tiempo parcial y se le contrata estrictamente el número de horas de clase presencial y de tutoría mínimas, ¿debe donar gratuitamente su tiempo a la universidad? La respuesta habitual es *mirar hacia otro lado*, y tiene consecuencias. Sin entrar en ellas, baste una referencia del mundo de la dirección de proyectos: en toda tarea con un objetivo preciso, hay que lograr un compromiso entre plazos, calidad y recursos necesarios; si se fijan dos, el tercero se ajustará. En una asignatura, el plazo es fijo. Por tanto, si se fijan los recursos (por ejemplo: *hay que adaptarse al EEES con lo que hay*), la calidad cambiará (estrategia *time-box planning* [1]) si se fija la calidad (por ejemplo, *seguir evaluando tan individualmente como antes*), los recursos necesarios subirán.

De la experiencia de los autores y de otros compañeros realizando el proceso de adaptación, vemos que son mucho más importantes y efectivas las técnicas concretas, los pequeños *trucos* usados, que las profundas y sesudas discusiones sobre la definición de competencia o el modelo de influencia entre tarea y competencia. El objetivo inicial de este artículo fue plantear las técnicas que hemos usado en los últimos nueve años, aunque finalmente apareció un nivel mayor de formalismo.

A continuación se presenta y se discute cualitativamente el modelo de cálculo de cargas de trabajo. El apartado 3 propone patrones que modifican las cargas de trabajo, en especial del profesor. El apartado 4 analiza las cargas en tres casos: asignatura tradicional, asignatura basada en trabajos asignados, y la anterior tras aplicar los patrones. El apartado 5 presenta una asignatura en la que se han utilizado los patrones, y cómo se han empleado varios de ellos. Las conclusiones cierran el artículo.

2. Modelo de cálculo de carga de trabajo del profesor

De cara al cálculo de la carga de trabajo que un profesor tiene al impartir una asignatura, en el modelo tradicional de clase más prueba final de evaluación, el número de alumnos únicamente afecta a la hora de calcular el tiempo necesario para corregir los exámenes. Sin embargo, la adaptación de nuestras titulaciones al EEES nos está llevando a plantear otros modelos docentes en los que la evaluación continua es la manera de valorar el aprendizaje de nuestros alumnos.

Esta evaluación continua se traduce la mayoría de las veces en solicitar trabajos que los estudiantes deben ir completando a lo largo del curso, o en pequeños exámenes o controles parciales, o en trabajos desarrollados por equipos, etc [2,3,4,5]. En todos estos casos, el número de alumnos pasa a ser un factor fundamental a la hora de determinar el trabajo del docente antes y después de acudir al aula. No sólo influye la cantidad de alumnos (que determina también cuántos equipos[2] se forman) sino también el número de grupos.

De forma sencilla pero suficiente para la mayoría de los casos, la expresión *(A * número de horas de clase + B * número de alumnos en la clase + C * número de grupos)* nos puede servir como modelo para contabilizar la carga de trabajo del profesor. En este polinomio hay tres elementos sobre los que el docente tiene poca influencia: las horas de clase dependen del número de créditos de la asignatura y de su coeficiente de presencialidad; el número de alumnos depende de la matrícula; el número de grupos depende de coeficientes fijados por la universidad y de los recursos disponibles (instrumentos en el laboratorio, ordenadores,…). A continuación se analiza cualitativamente cómo dependen A, B, C de la naturaleza y contexto de la asignatura. Un análisis cuantitativo y de simulación queda fuera del alcance de este artículo, y aún no ha sido publicado.

Coeficiente A. Es siempre mayor que 1 (impartir una hora de clase implica, al menos, una hora de trabajo del profesor). La *tasa de cambio* de contenidos de una asignatura, cuánto cambie cada año, es el factor que más influye en este coeficiente: una asignatura nueva, o un

[2] En este artículo se usa el término *grupo* para referirse a las agrupaciones administrativas en que se separa a quienes cursan una asignatura para recibir las sesiones teóricas, prácticas y tutorías, debido a la capacidad de los laboratorios, al máximo de alumnos por aula o profesor, etc., y *equipo* (de trabajo) para referirse a las pequeñas agrupaciones de alumnos, espontáneas o inducidas por el profesor, que realizan juntos tareas y encargos en la asignatura.

cambio importante en una existente, exige contenido y materiales nuevos para cada hora de clase, esto lleva un tiempo que varía con la naturaleza y calidad de los mismos. Igualmente, una tasa baja de cambios permite un mayor efecto de la *curva de aprendizaje*: cuantas más veces se haya impartido una clase, menos cuesta prepararse para darla la siguiente vez. Hay otros aspectos más prosaicos, pero igualmente reales, en este factor, como son los costes de *ejecutar* la impartición (que dependen realmente del número de sesiones, pero basta con conocer la duración media de cada sesión): desplazamiento al centro/aula, puntualidad, tiempo extra post-clase. Se puede obtener una conclusión relevante para el objeto de este artículo: el coeficiente A *no depende* de que cómo esté diseñada la asignatura (como tareas de alumno o no).

Coeficiente B. A este coeficiente le aporta todo aquello que requiere un trato individual: que la tutoría sea presencial o por internet (email, web, etc.); la corrección de trabajos individuales; informar de resultados.

Coeficiente C. Impartir una misma clase a dos grupos distintos implica no sólo repetirla, sino una labor adicional de coordinación entre ambos grupos para atender a las diferencias que van a surgir entre ambas imparticiones (o entre ambos experimentos, entre ambas clases prácticas, etc.), o por ser profesores distintos los de cada grupo.

Este modelo puede (y de hecho, lo hace) tener en cuenta más factores: diferenciar clases magistrales, de laboratorio, de trabajo de campo; ligar el número de alumnos a la dificultad de la asignatura con una *tasa de repetidores*; diferenciar en A el caso de asignatura no estable (es decir, que acaba de ser creada, o que el profesor ha impartido pocas veces, o que ha tenido un cambio no sencillo en contenidos y calidad de los métodos docentes); calcular el trabajo del alumno en la misma asignatura con otra expresión y coeficientes; etc. Pero esto no afecta a su uso en este documento, por lo que pasamos, en el siguiente apartado, a presentar directamente patrones que permiten al profesor variar los coeficientes B y C. *Alterar*, en el enfoque planteado en este documento, implica *reducir*.

3. Patrones para modificar las cargas de trabajo de una asignatura

En este apartado se presentan una serie de patrones simples para modificar las condiciones de una asignatura estructurada en tareas. Tal y como se ha comentado previamente, este tipo de planteamiento en tareas aumenta de forma desorbitada el trabajo del profesor, de manera que se hace inviable si la matrícula supera las 20-25 personas. Sin embargo, las posibilidades que se muestran a continuación han sido puestas en práctica con éxito en grupos más grandes (ver apartado 5).

Cuando el profesor corrige o evalúa una tarea que ha diseñado para que los alumnos adquieran cierta competencia (CA), utiliza sus propias competencias (CP) asociadas a la capacidad de evaluar. La hipótesis básica para los mecanismos propuestos es la siguiente: si se practican las competencias CP se forman también las CA, y con más intensidad que lo hace el desarrollo de la propia tarea (pensada para adquirir CA). Por tanto, si se diseñan actividades que sean evaluables por los propios alumnos, la labor de evaluación puede incorporarse como tarea a la asignatura. De esta manera, los estudiantes además de practicar las competencias que la tarea valorada les plantea, trabajan otras adicionales y transversales (como resolución de problemas, negociación, comunicar malas noticias, defender la propia posición, etc.). El trabajo que esto supone para ellos se contabiliza también dentro de la carga de créditos de la asignatura.

Hay que tener en cuenta que las medidas que se proponen pueden resultar negativas si no se realizan correctamente. Muchas veces hemos escuchado cómo algún profesor ha optado por dejar parte del temario para que lo preparen los alumnos y lo apliquen fuera de clase o lo expliquen al resto de compañeros. Esto conlleva una reducción de la calidad de la enseñanza a no ser que previamente se haya preparado a los estudiantes y enseñado cómo hacerlo consumiendo, por lo tanto, el tiempo que se pretendía ahorrar. Es decir, estos patrones deben utilizarse en un entorno en el que se garantice que se mantiene la calidad de la enseñanza.

Por otro lado, consideramos que el verdadero valor de las técnicas que aquí se presentan está en que puedan servir también a otros profesores. Para ello se está estudiando la posibilidad de crear una web en la que colocarlos y permitir que todo aquel que desee muestre sus propias maneras de resolver la situación.

A continuación se explican los patrones propuestos.

3.1. El alumno evalúa.

Mediante esta estrategia los alumnos colaboran en la evaluación de tareas, y se reduce el trabajo del docente. Aunque aparecen un factor de eficacia (ya que no lo hace con la misma calidad que el profesor) y de eficiencia (al alumno le cuesta más), estos se compensan haciendo una corrección al modelo: el trabajo del profesor sería como corregir a un único estudiante. Además, no debe olvidarse que, en la estrategia de adaptación al EEES que analiza este artículo, es *realizar* de las tareas lo que forma, no la evaluación de las mismas. Por tanto, que un alumno que ha llevado a cabo las actividades evalúe a otro (aumentando más su competencia, con esta labor derivada), es una situación bien distinta al –imposible– equivalente *un alumno corrige los exámenes en vez del profesor*.

En los puntos siguientes se muestran algunos patrones que se pueden considerar dentro de este tipo en el que el alumno se transforma en evaluador.

- *Se corrigen pequeñas tareas unos a otros.* Ya sea dentro del equipo o con la clase al completo, se reparten los trabajos y ellos mismos evalúan. Dejar claros cuáles son los criterios de evaluación que se deben utilizar es indispensable para obtener valoraciones similares.
- *Evaluación del resto de equipos a equipo*. Los distintos equipos formados en clase evalúan el trabajo de otro y cada equipo emite una única valoración que debe estar consensuada.
- *Evaluación de una persona a equipo.* En cada equipo se nombra a una persona que se va a encargar de ayudar a otro equipo a llevar el trabajo al día, de ir valorando lo que van haciendo y de tratar de resolver las dudas que tengan [6,7].
- *Evaluación de un equipo a otro.* En este caso dos equipos intercambian su trabajo y cada uno evalúa la labor realizada por el otro.

3.2. Delegar algunas tutorías sobre teoría

Se solicita a algunos alumnos que hagan de guías de otros equipos. Su labor consiste en resolverles las dudas más comunes y en caso de no poder hacerlo acudir al profesor.

3.3. Tiempo de clase para reuniones

Esto garantiza que todos los miembros del equipo se puedan ver a la vez, y que se puedan exigir cosas concretas a desarrollar en la propia reunión. Además, esas mismas tareas deben acabar evaluadas ya que se redacta un acta por cada una de las reuniones realizadas.

3.4. Evaluar el hecho de obtener una solución

Es decir, valorar la propia existencia de un resultado (que refleja por tanto un grado de trabajo desarrollado) y no la calidad de la misma (superados unos mínimos de fácil constatación). Un ejemplo puede ser cuando la mera entrega de algo es ya una evaluación.

3.5. Secuencias de tareas dependientes

En este caso se plantean una serie de tareas en las que la finalización de unas es indispensable para comenzar otras. Si no logran desarrollar las precedentes, no podrán hacer las sucesoras. Por tanto, que estén realizando las sucesoras ya evalúa las antecesoras, y su aportación a la competencia.

3.6. Logros concretos para fechas concretas

El planeamiento en este caso sería el siguiente: para cierta fecha deben lograr algo, si en la fecha límite ese algo se ha obtenido, entonces el hito está superado. Por ejemplo, encontrar un cliente al que desarrollarle el proyecto como máximo al final de la segunda semana de clase.

3.7. Muchos entregables a desarrollar

La estrategia consiste en plantear una gran cantidad de entregables que tienen que ir generando. Es necesario que estos entregables sean distintos entre sí ya que de lo contrario nos podemos encontrar con copias de otros años o con *generadores de ejercicios*. Deben ser pequeños proyectos singulares que les permitan tener un mismo patrón de actuación.

3.8. Mantener la estructura del trabajo: la figura del Registrador

El *Registrador* mantiene una estructura predefinida para la observación del trabajo de varios alumnos (un informe, una carpeta física con cierta estructura, unas carpetas en un sitio web, etc.). Esto ayuda por un lado, a que los propios alumnos se organicen y por otro, a que el profesor sea capaz de reconocer rápidamente cómo va dicho trabajo o de comparar sin demasiado esfuerzo los resultados de dos personas o dos equipos. Esta posibilidad es más efectiva cuando existen un número elevado de pequeños trabajos o cuando hay que desarrollar un trabajo más amplio que exige una estructuración (una memoria o un informe). Por ejemplo, decidir en qué carpetas, con qué nombre de archivo o con qué índice hay que entregar una memoria. El efecto de aplicar el patrón es de oportunidad (puede evaluar cuando más le conviene), y de cantidad (cada evaluación cuesta menos tiempo).

3.9. Evaluador externo

Consiste en solicitar la opinión de alguna persona externa sobre el trabajo desarrollado. Esta persona puede ser el beneficiario de los trabajos realizados, o algún profesional invitado a hacer esa evaluación, alumnos de años anteriores que lo hicieron bien en la asignatura y mantienen el interés, etc.

3.10. Dividir en partes independientes

Tratar de dividir el contenido en partes independientes. Esto permite utilizar con cada una de esas partes un modelo adaptado al mismo. Al reducirse el gránulo mínimo gestionable, la viabilidad de aplicar los patrones anteriores aumenta.

4. Comparación de tres escenarios

Supongamos una asignatura en la que los datos en los que no influye el profesor son: 6 créditos; 100 alumnos; 60 horas de clase para cada alumno, repartidos en 2 grupos durante 35 de las horas (aula para clase magistral) y 4 grupos durante 15 horas (aula especial, laboratorio). Analizamos cada escenario en la misma situación de *estabilidad* de la asignatura, cuando lleva un tiempo impartiéndose por el mismo profesor, y el tiempo de preparación es el mismo. En tal caso, el coeficiente A es el mismo en todos los casos, y el C varía entre escenarios sólo porque se incrementa en el 2 y en el 3 (por ejemplo, si cuesta una hora para preparar la notificación de la nota final de la asignatura a cada grupo, lo cuesta en los tres escenarios). En caso de tratarse de la primera vez que se llevan a cabo estas estrategias, habría que añadir un coste en el desarrollo de las diferentes plantillas, planificaciones e infraestructura necesaria para ellas.

Escenario 1. Situación tradicional. La evaluación es mediante examen final individual. El tiempo de preparación no se considera (reutilización, etc.). 30’ de corrección por examen, 15’ por cada revisión y un 20% de peticiones de revisión. El coeficiente B es 30’ + 15’ * 20%. Horas de evaluación: 55.

Escenario 2. Una vez modificada la asignatura a las nuevas estrategias pedagógicas, se han introducido 4 tareas. Por cada tarea hace falta 1 hora para plantearla y asignarla (el coeficiente C se incrementa respecto al escenario 1 en 4 horas). Por cada tarea y alumno: 10’ de corrección, 10’ de guía previa (mientras la hacen) y/o posterior (son evaluaciones durante el curso que, para ser efectivas, requieren orientar al alumno según el resultado), y no hay revisión. El coeficiente B es 4*20’, es decir, 133 horas. Las horas necesarias pasan a ser unas 137. Se produce *un incremento del 250%* respecto al escenario 1.

Escenario 3. Es el escenario 2, añadiéndole algunos de los patrones antes planteados: los alumnos se evalúan entre sí; 5 alumnos velan porque todas las evaluaciones se informen en el mismo formato; 10 alumnos revisan, cada uno, el 10% de las evaluaciones en cada tarea. Para formar a estos alumnos en cada grupo necesita-

mos unas 5 horas, que incrementan C respecto al escenario 2. El profesor verifica el 10% de las evaluaciones hechas durante el año por cada alumno revisor (aporte al coeficiente B: 20'*10%*4). Supondremos que hay un *mal revisor* cada año, y que el profesor debe rehacer todo su trabajo (aporte al coeficiente B: 20'*10%*4). Las horas necesarias pasan a ser unas 40h. Añadamos un 25% de tiempo para atender a los riesgos e incidentes en un modelo así y obtenemos un total de 50h. Es decir, *el mismo número de horas que en el escenario 1*. Por lo tanto, el escenario 3 es posible sin trastornos al profesorado, haciendo al alumno protagonista de su aprendizaje, y con mejor formación de competencias (lo que es seguro para al menos 15 de 100 alumnos).

El contraste que se muestra en este análisis puede parecer extremo, pues se basa, como se verá en la exposición que viene a continuación, en una asignatura que lleva ya nueve años implementando patrones como los descritos. Sin embargo (y precisamente por eso se presentan como *patrones*, no como métodos) pueden ser usados como guía, adaptados y aplicados a muchas asignaturas, siempre que lo sean por el profesorado responsable de la misma formado en estos patrones, o con el apoyo necesario.

5. Aplicación de patrones en PGPI

La asignatura PGPI [7] en la que se han puesto en marcha los patrones que se describen en un apartado posterior se imparte en la Facultad de Informática de San Sebastián (FISS) dentro de la titulación de Ingeniería Informática. Se trata de una asignatura troncal de 9 créditos.

Su orientación es práctica: equipos de 6 alumnos llevan a cabo un proyecto real para desarrollar capacidades [6]. Cada proyecto aplica al completo la propuestas de PMBOK [8]. Al principio del curso cada alumno elige qué nota va a obtener, y esto implica un nivel de compromiso, de carga de trabajo, y unas tareas concretas que debe cumplir. Por otro lado, si todos los alumnos de un equipo se ponen de acuerdo, pueden optar por recibir su calificación al final, tras ser evaluado a medias entre el resto de los equipos y el profesor. Cursan unos 100 alumnos en 2 grupos (uno en castellano, otro en euskera). El número de equipos ha oscilado entre 14 y 20. Una semana de clase estándar tiene tres sesiones de hora y cuarto y una sesión de dos horas y media (pensadas para poder dividir a los alumnos en grupos menores según los recursos del laboratorio).

Este tipo de asignatura encaja en el escenario 3 del apartado 4 (aunque no es el mismo). El diseño docente de esta asignatura requirió, en su momento, una gran inversión de trabajo. Este trabajo fue el mismo que hubiera sido necesario para diseñar una asignatura en el escenario 2, *excepto* por el tiempo adicional que nos requirió crear los patrones del apartado 3. Repitiendo el trabajo hoy en día, para otra asignatura, costaría el mismo esfuerzo rediseñar una asignatura para el escenario 2 que para el escenario 3. La mayoría de los patrones descritos en el apartado 3 se han utilizado en alguna ocasión desde 1998, y muchos están en uso en la actualidad de la forma que se describe a continuación.

Corrección de tareas. El volumen de trabajo es tal que es necesario el reparto del mismo entre los diferentes miembros del equipo. Es decir, no es posible llegar en plazos con el proyecto finalizado si deciden *hacerlo todo todos, por separado o juntos*. Por lo tanto es necesario delegar y con ello se hace necesaria también una revisión del trabajo hecho. Normalmente son los propios alumnos los que se organizan para ir revisándose unos a otros los documentos que van generando. Suelen definir algún sistema rotatorio de forma que todos van controlando el trabajo de todos.

Evaluación del resto de equipos a un equipo. Durante la última semana del cuatrimestre cada equipo muestra y presenta el trabajo realizado al resto de la clase. En algunos casos, además, el resto de los equipos emite una nota relativa a lo que acaba de presenciar. La calificación la da el equipo al completo y por tanto es necesario que lleguen a un consenso sobre la misma antes de hacerla pública.

Evaluación de una persona a equipo: *la figura del Controller*. Los Controllers son alumnos que deciden tener una implicación con la asignatura mayor, optan a una nota mayor, y dedicarán más tiempo que el resto. Ese tiempo adicional lo invertirá en tareas a solas (de autoformación y para realizar la evaluación), tareas

conjuntas de todos los Controllers y el profesor (formación específica, 3 o 4 sesiones de 15'), y a trabajo individualizado con el profesor. En los dos últimos casos, aumenta el tiempo del docente (según se ha visto en los apartados anteriores).

A cada Controller se le asigna un equipo, del cual semanalmente recibe los entregables desarrollados y los partes de trabajo donde cada miembro indica el tiempo que ha dedicado a cada tarea de la asignatura. El Controller comprueba que se están cumpliendo las fechas en que hay que ir creando los entregables, y que los contenidos de dichos entregables *son razonables* (es decir, sin calificar el contenido, sí pueden indicar, por ejemplo, si algún entregable refleja claramente que el equipo no entiende lo que tiene que hacer). Por supuesto, para ello el Controller debe tener un dominio suficiente de la asignatura. El Controller también revisa si el reparto de trabajo es equitativo en el equipo, revisando los partes de trabajo semanales. Este sistema permite detectar rápidamente aquellos equipos que se están *descolgando* y tomar las medidas oportunas (hasta ahora, son los propios equipos los que se han autorregulado y normalmente aumentando el ritmo de trabajo, han conseguido llegar a finalizar el proyecto).

Evaluación de un equipo a otro. Aunque no se utiliza, se ha estudiado la posibilidad de aplicarlo. Consistiría en intercambiar los proyectos de los equipos de dos en dos. Esto se realizaría a mitad del desarrollo y la primera labor del equipo tendría que ser evaluar el trabajo que se le ha entregado. Además pondrían todo el empeño en ello puesto que ellos han de continuar con el proyecto que reciben y no podrían, en el futuro, achacar ningún problema a la herencia recibida salvo que lo identificaran en esta evaluación.

Delegar alguna tutoría de teoría. La figura del Controller es la que realiza esta labor. Parte de su cometido es resolver las dudas más comunes que su equipo le plantee y si no sabe resolverlas acudir al profesor. Este trabajo refuerza los conceptos y hace que el Controller adquiera un conocimiento más profundo de la materia.

Tiempo de clase para reuniones. Los alumnos se reúnen tanto en clase como fuera de ella. Normalmente ellos mismos son los que deciden el orden del día. Sin embargo, en algunas ocasiones los temas a tratar los decide el profesor y se ponen de forma que haya que llevar a cabo alguna tarea concreta (teniendo en cuenta el tiempo que tienen para ello de manera que puedan tenerla finalizada durante la reunión). El acta de esa reunión da cuenta del trabajo realizado.

Evaluar el hecho de tener una solución. Uno de los aspectos que los estudiantes de PGPI deben saber hacer es estimar el tiempo que les va a costar realizar las diferentes tareas del proyecto. Esta estimación es uno de los elementos indispensables para una correcta planificación. Sin embargo, y aunque se les enseñan diferentes técnicas para hacerla, los alumnos carecen de la experiencia suficiente como para poder dar unos valores que se aproximen en grado suficiente a la realidad posterior. Por lo tanto, lo que se tiene en cuenta en este caso es el esfuerzo de haber desarrollado una estimación de tiempos (aunque se revisa el contenido para comprobar que lo que proponen no es una invención).

Secuencias de tareas dependientes. Tal y como se ha comentado con anterioridad, los equipos terminan desarrollando en torno a 100 entregables. Pero estos no se pueden crear en el orden que uno quiera, es decir, los alumnos no pueden dividir directamente el trabajo entre 6, repartírselo y quedar un mes más tarde para ver qué ha hecho cada uno. Existen dependencias entre los entregables de forma que la información que está en unos es indispensable para poder crear otros. Así, las personas encargadas de desarrollar los documentos que dependen de otros son las primeras interesadas en que aquellos se hagan bien y a tiempo.

Logros concretos para fechas concretas. El ejemplo que se explica en el apartado anterior es el más ilustrativo en relación a la asignatura pero además del logro del cliente existen otros como la elección del proyecto que van a desarrollar.

Muchos entregables a desarrollar. Nuestra decisión de aplicar el PMBOK[8] completo lleva a que los alumnos tengan que generar gran cantidad de documentos. No pueden copiarse entre ellos porque son específicos de cada proyecto; al contrario, entre miembros de diferentes equipos se explican qué han entendido o cómo han resuelto un documento concreto.

Mantener la estructura del trabajo: *la figura del Registrador*. Un alumno Registrador tiene, entre otras, la responsabilidad de mantener la integridad del *archivo del proyecto* en el que se va dejando todo el trabajo. Ello implica tener todos los entregables organizados en tiempo real según la estructura (prefijada) del archivo. Esto permite al profesor acceder a comprobar el estado del proyecto en cualquier momento, y evaluar en menos tiempo ya que conoce dónde debe estar cada tipo de entregable y con qué nombre. Esta figura no implica trabajo adicional del profesor (los conceptos ya se imparten para todos), y el trabajo adicional que le implica al alumno se ve compensado por la nota mayor a la que opta.

Evaluador externo. En PGPI se invita a algún profesional del área a las defensas de los proyectos. Las preguntas o comentarios que esta persona realiza nos pueden dar una idea de cómo se consideraría fuera la labor desarrollada.

6. Conclusiones

Una asignatura en la que deben evaluarse varios trabajos o ejercicios por cada alumno exige alternativas a la evaluación directa del profesor. El ejemplo mostrado, la asignatura PGPI, requeriría a cada profesor corregir en torno a 1000 pequeños documentos. Con los patrones que se han usado, los alumnos producen resultados con una calidad media que viene avalada por dos vías: los propios miembros de cada equipo, y los Controllers.

Estas estrategias son posibles porque es *realizar* de tareas lo que *forma competencias*. La evaluación es sólo una verificación; por tanto, que un alumno que ha llevado a cabo las actividades evalúe a otro no sólo es una opción, sino que refuerza su competencia. Es una situación bien distinta al –imposible– equivalente *un alumno corrige los exámenes en vez del profesor*.

Aunque las posibilidades que se dan en este artículo se han utilizado con éxito en una asignatura concreta, son fácilmente adaptables a asignaturas de otra índole. No es necesario plantear trabajo en equipo para utilizar la mayoría de ellas. Pueden modificarse para equipos más pequeños, o al trabajo individual, sin equipos. Aplicándolos, hemos mostrado que puede rediseñarse una asignatura para usar métodos didácticos en los que el alumno tome parte activa, e impartirla *sin requerir más horas que antes del cambio*.

En el caso que nos ocupa los autores usan una docencia completamente orientada a tareas, siendo su carga de trabajo similar o menor a la de una asignatura tradicional. Las encuestas de opinión anuales, que muestran valoraciones de los alumnos altas y sostenidas año a año, así como los correos electrónicos de exalumnos en activo profesionalmente, en los que agradecen tanto lo aprendido como la forma en que lo lograron, nos permiten pensar que, efectivamente, la estrategia escogida es correcta.

Referencias

[1] I.Graham, *Object Oriented Methods*, Addison Wesley, 2002

[2] M. Marqués, V.R. Tomás, I. Sanz, *Tratando de fomenter la motivación del estudiantado*, Actas Jenui, 329-336, 2004

[3] P. Bofill, B. Otero, E. Toribio, J.M. Aroca, M. Breitman, P. Garcias, J.M. Sancho, *ANCORA: Aprendizaje Organizado por Tareas,* actas Jenui, 71-77, 2005

[4] I. Fortes, J. Medina, S. Sánchez, *Una experiencia de implantación del sistema de créditos europeos en los planes de estudios vigentes,* Actas Jenui, 179-185, 2005

[5] V. Traver, J.A. Javier y Traver, *¿Por qué no enseñamos a aprender cooperativamente?* Actas Jenui, 297-304, 2004

[6] M. Bermejo, R. Fernández, *Alumno Rupérez ¡está Ud. despedido!,* Novática Nº 182, pp. 40-43, 2006

[7] M. Bermejo, R. Fernández, *Documentación de soporte al modelo de docencia basado en la simulación del funcionamiento de una empresa en la asignatura Planificación y Gestión de Proyectos Informáticos,* Informe Interno Dpto. LSI (UPV/EHU), UPV7EHU/LSI/TR1-05, 2005

[8] Project Management Institute, *A Guide to the Project Management Body of Knowledge*, pmi, 2000

Evaluación formativa de capacidades transversales en el marco EEES

Ferran Virgós Bel[1], Joan Segura Casanovas[2], Edmundo Tovar[3]
Dpto. LSI [1,3] , Dpto ESAII[2])
EUETIB, UPC [1,2]), FI, UPM [3]
Urgell 187, 08036 [1,2])
{Ferran.Virgos|Joan.Segura}@upc.edu ;etovar@fi.upm.es

Resumen

El marco del nuevo Espacio Europeo de Enseñanza Superior (EEES) ha estado presente en la totalidad de los foros universitarios más recientes. El concepto de *European Credit Transfer System* (ECTS) constituye uno de los pilares que da permeabilidad y flexibilidad al espacio único. El otro gran pilar se basa en un cambio estructural más profundo con la incorporación de nuevos enfoques y nuevas metodologías. También, y sobre todo, de la necesidad de una *orientación de formación para la profesión*. Por eso se nos dice que los nuevos diseños curriculares deberán basarse en la definición de las *competencias profesionales* y, en consecuencia, empezar por la identificación de las capacidades asociadas.

Se habla, de hecho, de capacidades, habilidades y actitudes y al referirnos a capacidades, se entiende que nos interesan también las capacidades transversales. Se está pensando en un perfil completo.

Las capacidades transversales raramente llegan a tratarse en profundidad en la bibliografía, aunque ya aparece un primer estudio en el libro blanco del grado en informática [1]. basado en una ordenación de las capacidades consideradas por parte de los colectivos seleccionados.

En este sentido, el presente trabajo, pretende andar un paso más proponiendo un modelo que ayude a la valoración de las capacidades trasversales, no sólo en forma absoluta sino, fundamentalmente, con detección de capacidades con déficit o "*gaps*", planteando, una metodología de evaluación formativa que pueda personalizarse a cada entorno y que incorpore en forma implícita la adaptación y mejora continuas. Como valor añadido, se comentar los resultados obtenidos en una primera aplicación del prototipo de herramienta construida.

1. Presentación

El nuevo marco del EEES se ha ido abriendo paso comenzando por los foros de mayor proximidad temática, incluyendo el propio JENUI. Pero la publicación de los reales decretos de grado y postgrado (ver referencias [2] y [3]) lo ha convertido definitivamente en elemento motor de la actividad universitaria actual.

Pero ¿cuál es el planteamiento novedoso del marco de Bolonia?, o ¿es realmente un planteamiento novedoso?, o ¿es más de lo mismo con otro envoltorio?. Algunos han ironizado sobre la percepción del profesor que ve los créditos ECTS como una operación aritmética respecto los valores del marco anterior. Otros, más alineados con el nuevo enfoque, lo han definido como una evolución *"de la enseñanza al aprendizaje"*. En este caso, normalmente, se ha puesto el énfasis en aspectos metodológicos desde aprendizaje cooperativo hasta aprendizaje basado en proyectos (PBL), o problemas (en su acepción más ligera y flexible). Otros muchos aspectos metodológicos como biblioteca cooperativa, congreso de alumnos, etc. han aparecido, también, en algunos trabajos (ver, por ejemplo, [4] y [5]).

En este planteamiento metodológico global novedoso, no podía faltar el inestimable soporte de la tecnología, de modo que la *extranet* docente pasa a constituir un elemento esencial en la mayoría de paradigmas propuestos, compartiéndose los sistemas propios con una tendencia a utilizar el soporte de una plataforma genérica (normalmente *Moodle)*.

De todos modos en este movimiento de "replanteamiento" no siempre es fácil la conversión a la nueva fe, de modo que las universidades han debido contar con sus propios misioneros en el duro trabajo de la evangelización metodológica. ¡Y está bien! pero yo creo que el EEES no se debe ver (sólo) desde la perspectiva

metodológica. Por ello, en todos los foros en que he tenido ocasión, he aprovechado para intentar transmitir la idea que la supuesta nueva religión, si existe, debe basarse en unos nuevos principios y que la metodología y, por supuesto, la tecnología a aplicar, deben pasar a segundo término. En [7] se describe un modelo centrado en defender que el planteamiento de las materias no debería hacerse nunca por mimetismo ni basarse en el concepto clásico de "programa", sino partiendo de la identificación de unas capacidades y sub-capacidades a aprender (que aquel trabajo denomina "ejes de actividad"), cuya consolidación debería dar lugar a las competencias deseadas en el alumno.

En este punto, recupero unas notas tomadas en una conferencia del profesor *Meyer* en Jenui 2004 donde decía que los "objetivos de formación en la universidad" deberían ser dos:

- *"professional competentes"*
- *"help find a job"*

Sin duda podríamos añadir otros aspectos de formación integral pero, como idea de transición, ¡No se puede decir más claramente!: no podemos basarnos en modelos "formales" de la estructura del conocimiento porque la realidad actual es muy cambiante. Decía el profesor Meyer que debe buscarse un equilibrio entre los principios de la disciplina, los principios de la educación universitaria y los intereses de los alumnos. En estos intereses, aparte otros de tipo personal, interpreto yo que, el prof. Meyer consideraba incluidas las necesidades y visión de las organizaciones que deben emplear nuestros egresados. Así, organizaciones (normalmente empresas), alumnos y universidad (léase profesores) son los elementos básicos a considerar de forma equilibrada en el nuevo marco. Actores a interpretar la nueva obra en el nuevo escenario.

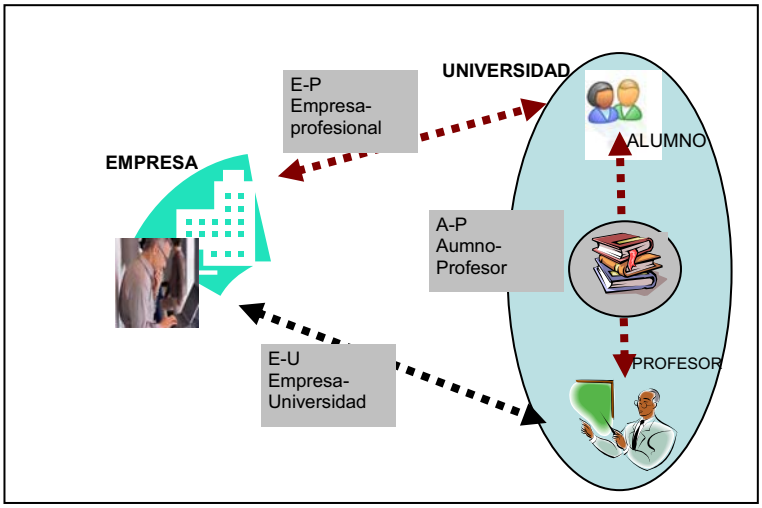


Figura 1. ALUMNO-PROFESOR-EMPRESA, actores y argumento del proceso enseñanza-aprendizaje

2. Miremos hacia las capacidades transversales

¡El EEES no se hizo en un día!. Hay que ir con cautela. Por ello, en el presente trabajo decidimos avanzar en la línea de dar un poco de luz en relación a las capacidades transversales. Respecto a ellas, parece necesario considerar diferentes aspectos convertibles en fases:

- Identificación de capacidades transversales.
- Validación/ valoración de su relevancia, idealmente de tipo formativo (realimentación).
- Mecanismos de incorporación al proceso formativo de adaptación curricular.

En el libro blanco del grado en informática [1], encontramos un estudio que incluye la identificación de 19 diferentes capacidades transversales (lista establecida por el mismo a partir de un examen bibliográfico previo). Las capacidades identificadas se incluyen en la tabla 1, a continuación, ya ordenadas según la valoración resultante de aquel trabajo:

CT1.	Capacidad para resolver problemas.
CT2.	Trabajo en equipo.
CT3.	Capacidad de análisis y síntesis.
CT4.	Capacidad de organización y planificación.
CT5.	Capacidad de gestión de la información (captación y análisis de información).
CT6.	Capacidad para tomar decisiones.
CT7	Motivación por la calidad y la mejora continua.
CT8.	Conocimiento de alguna lengua extranjera.
CT9.	Capacidad de trabajo en equipo multidisciplinar.
CT10.	Comunicación oral y escrita.
CT11.	Razonamiento crítico.
CT12.	Habilidades de relaciones interpersonales.
CT13.	Capacidad para dirigir equipos y organizaciones.
CT14.	Conocimientos básicos del ámbito de formación.
CT15.	Conocimientos en alguna especialidad del ámbito de formación.
CT16.	Capacidades directivas.
CT17.	Trabajo en un contexto internacional.
CT18.	Reconocimiento de la diversidad y multiculturalidad.
CT19.	Sensibilidad por el medio ambiente.

Tabla 1. Capacidades transversales consideradas [1]

El estudio incluye la valoración de las 19 capacidades identificadas por los mismos tres colectivos que aparecen en nuestro planteamiento de la figura 1: empresa, profesores y estudiantes (hay que aclarar que en aquel trabajo, el colectivo "estudiantes" es, en realidad, el de "titulados").

Para guiar la valoración, en el estudio citado se pidió a cada persona encuestada una "ordenación" de la lista presentada. Así se eliminaban empates y se evitaban las respuestas ambiguas (aunque, a cambio, no facilitaba la discriminación ponderada).

El trabajo es un excelente punto de inicio ya que, de entrada, nos facilita una primera lista de 19 capacidades (aunque, lógicamente, deberá avanzarse en la identificación e incorporación de otras capacidades nuevas.). Por otra parte, la ordenación por los tres colectivos nos facilita un primer punto de referencia de cara a la futura planificación docente.

Competencia	COLECTIVO (orden)			
	emp	tit	Prof	Total
CT1	2	1	2	5
CT2	1	2	3	6
CT3	3	4	1	8
CT4	4	3	5	12
CT5	8	6	6	20

Tabla 2. Las 5 primeras capacidades, según [1]

Nótese que existen pequeñas diferencias entre los colectivos, totalmente esperables, por otro lado. Así, la *capacidad de análisis y síntesis* (CT3) es la más importante para los profesores, mientras los titulados resaltan la *capacidad de resolver problemas* (CT1) y la empresa valora la *capacidad de trabajo en equipo* (CT2) En cualquier caso, es de resaltar que el nivel de acuerdo de las 5 primeras capacidades entre los diferentes colectivos, resultaba muy alta.

Lo cierto, no obstante, es que esta inestimable referencia inicial no nos sería de mucha utilidad práctica en el momento de desear definir la cobertura de capacidades para una planificación curricular específica con mejora continua, ya que el resultado da luz sobre la conveniencia de disponer de la capacidad pero la evaluación no es "formativa" al ser de objetivos pero no tener una medida del error. Es una evaluación, por decirlo en términos de servo-sistemas, de lazo abierto, sin realimentación. Permite saber que hay que facilitar una determinada capacidad pero el tipo de medición no aporta nada en la "situación" respecto el "ideal". En particular si ya se ha alcanzado, si nos hallamos lejos o, incluso, si estamos abusando.

3. Una herramienta de evaluación "Formativa"

Llegado este punto, nos planteamos la creación de una metodología (y una herramienta asociada) que permitiera conocer no sólo la "evaluación" de los niveles deseables de capacidades transversales, sino, también, la detección del nivel alcanzado en un determinado entorno, cosa bien diferente. En definitiva, se trataría de plantear una metodología similar a la descrita en [6] y [8] para las funciones de los informáticos en las empresas. En este caso, proyectada sobre las "capacidades" transversales en una determinada titulación.

En definitiva, se buscaba que el modelo pudiera actuar como evaluación "formativa". Es decir, una evaluación que no se limitara a dar una referencia deseable sino que ofreciera una medición del nivel realmente alcanzado y, por tanto, ofreciera un diagnóstico de los aspectos a mejorar.

Finalmente, optamos por realizar una modificación al modelo descrito en [8], incluyendo una pregunta no del valor alcanzado sino del "déficit" o "gap" respecto al valor deseable, para clarificar las respuestas. De este modo se preguntaba directamente por el "error" que debía considerarse en la replanificación.

Naturalmente no todos los "*gaps"* deben recibir el mismo tratamiento. Parece lógico centrarnos especialmente en aquellos que podamos clasificar como "críticos".

Por ello, en definitiva, decidimos evaluar para cada capacidad, las variables siguientes:

- Nivel de acuerdo (variable ACU)
- Prioridad o criticidad (variable PRI)
- *"Gap"* o desviación (variable GAP)

Variable	Descripción	Medición
ACU	Nivel de acuerdo con la capacidad. Se trataría de identificar el grado de consenso en la necesidad de incorporar la formación en la capacidad al currículo docente	Escala Likert de 7 valores con signo, donde el 0 es el elemento neutro +3=totalmente de acuerdo +2=bastante de acuerdo +1=más bien de acuerdo 0=neutro -1=más bien en desacuerdo -2=bastante en desacuerdo -3= totalmente en desacuerdo
PRI	Prioridad o criticidad de la capacidad (notar que alguien puede estar de acuerdo en que es bueno la capacidad "saber jugar al golf" pero no se considera crítico para un ingeniero (¿o si?)	Escala Likert de 5 posiciones de 0 a 4 que pondera el nivel de acuerdo: 0= nada importante 1= algo importante 2= bastante importante 3 = muy importante 4= absolutamente esencial
GAP	Detección de falta, déficit o *"gap"* en la capacidad	Escala Likert de 7 valores con signo, donde el 0 es el elemento neutro. Los valores positivos indican extra conocimientos ociosos de la capacidad, mientras los valores negativos indican déficit +3= se hace o se sabe exageradamente +2= se hace o sabe más de lo necesario +1= se sabe o hace algo más de lo necesario 0= está bien lo que se hace o se sabe -1 = falta algo de conocimiento -2 = falta bastante -3 = falta mucho

Tabla 3. Variables a considerar en el modelo de evaluación propuesto

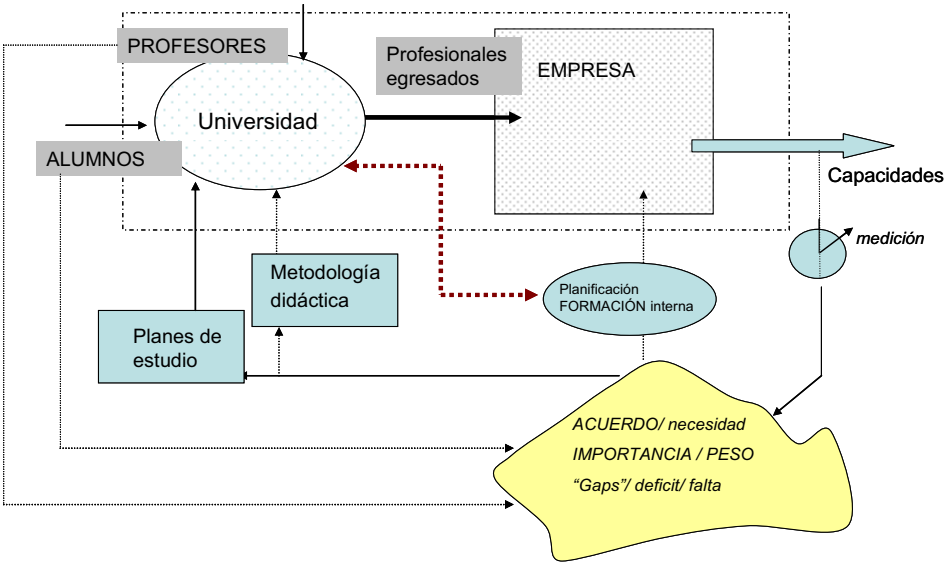


Figura 2. Modelo de evaluación formativa

4. El trabajo de campo

Una vez diseñado el modelo, construimos una herramienta para consultar los colectivos implicados. La herramienta se basaba en la consulta de las variables descritas en la tabla 3, en relación a las capacidades transversales de la tabla 1. Naturalmente el documento construido realizaba la presentación del estudio e incluía las instrucciones necesarias para cumplimentar la encuesta.

El objetivo básico de nuestro trabajo se centraba, en esencia, en la propuesta del modelo y herramienta, no tanto en la obtención de unos resultados específicos, dado que éstos se entendían como dependientes del entorno en el propio planteamiento conceptual del modelo. No obstante, aportar datos concretos complementarios era, obviamente, un objetivo subsidiario, aunque fuera a título de experiencia piloto y, además, servía para "rodar" la herramienta de cara a su versión final.

En consecuencia, nos pusimos manos a la obra y planificamos la obtención de unos datos de referencia entre los colectivos definidos en el propio modelo:

1. Para la encuesta de alumnos, elegimos una porción del colectivo matriculado de nuestra asignatura de Fundamentos de informática de la titulación de ingeniería técnica en electrónica de la EUETIB (troncal, primer cuatrimesre). Para ello, aprovechando el día de examen del cuatrimestre de otoño del curso 2006 a 2007. Obtuvimos un total de 37 respuestas válidas

2. Como muestra del colectivo de visión empresarial decidimos seleccionar un grupo de alumnos del Master en dirección y organización de empresas de la UPC. Este planteamiento era especialmente cómodo y eficaz, aparte eficiente, dada la inexistencia de recursos extra o financiación institucional para nuestro estudio. Obtuvimos un total de 17 respuestas válidas.

3. Finalmente para el colectivo de profesores, nos basamos en un grupo de voluntarios del área de informática de la EUETIB, así como otros del área de electrónica, complementados por un pequeño grupo de la Escuela de informática y gestión de la universidad de Andorra, aprovechando nuestra condición de

profesor visitante. Contabilizamos un total de 16 respuestas válidas.

En todos los casos pusimos especial énfasis en clarificar los matices de las variables de la encuesta entre las personas, consultadas, conscientes de que la herramienta tenía un cierto grado de dificultad en la correcta interpretación (que, quizás, habría que intentar mejorar en el futuro). También detectamos que los colectivos se podrían ampliar a 4 diferenciando entre alumnos (idealmente primer cuatrimestre) y titulados egresados (día de defensa del PFC).

5. Análisis de resultados

Para cada uno de los tres grupos de encuestas evaluamos la media y desviación tipo de cada una de las tres variables (ACU, PRI, GAP) y, además, calculamos tres variables derivadas:

- El producto (ACU*PRI)
- El producto (GAP*PRI)
- El producto completo (ACU*PRI*GAP)

Siempre procedimos a calcular la desviación típica de la media para tener una orientación de la significación estadística.

	ACU	PRI	GAP	ACU*PRI	GAP*PRI	ACU*PRI* GAP
C1	2,27	3,03	-0,03	6,87	-0,08	-0,19
C2	1,86	2,43	-0,35	4,54	-0,85	-1,59
C3	1,97	2,70	-0,19	5,33	-0,51	-1,01
C4	1,89	2,32	-0,08	4,40	-0,19	-0,36
C5	1,65	2,22	0,03	3,65	0,06	0,10
C6	2,11	2,68	-0,54	5,64	-1,45	-3,05
C7	1,81	2,43	-0,32	4,40	-0,79	-1,43
C8	1,92	2,54	-1,51	4,88	-3,85	-7,38
C9	1,65	1,95	-0,30	3,21	-0,58	-0,95
C10	1,54	2,08	0,00	3,21	0,00	0,00
C11	1,84	2,54	0,03	4,67	0,07	0,13
C12	1,24	1,86	-0,08	2,32	-0,15	-0,19
C13	1,78	2,30	-0,84	4,10	-1,92	-3,43
C14	1,68	2,30	0,03	3,85	0,06	0,10
C15	1,43	2,05	-0,27	2,94	-0,56	-0,80
C16	1,46	1,89	-0,73	2,76	-1,38	-2,01
C17	1,43	1,97	-0,95	2,83	-1,87	-2,67
C18	1,19	1,73	-0,78	2,06	-1,36	-1,61
C19	1,38	2,30	-1,11	3,17	-2,55	-3,51

Tabla 4. Resultados encuesta a alumnos

	ACU	PRI	GAP	ACU*PRI	GAP*PRI	ACU*PRI * GAP
C1	2,353	3,000	0,235	7,059	0,706	1,66
C2	2,294	2,647	0,059	6,073	0,156	0,36
C3	2,294	2,706	0,765	6,208	2,069	4,75
C4	2,118	2,706	0,353	5,730	0,955	2,02
C5	1,882	2,353	-0,059	4,429	-0,138	-0,26
C6	1,706	2,235	-0,353	3,813	-0,789	-1,35
C7	2,059	2,235	0,000	4,602	0,000	0,00
C8	2,294	2,353	-0,235	5,398	-0,554	-1,27
C9	1,765	1,765	0,176	3,114	0,311	0,55
C10	2,353	2,647	-0,529	6,228	-1,401	-3,30
C11	2,000	2,118	-0,118	4,235	-0,249	-0,50
C12	1,765	2,176	-0,353	3,841	-0,768	-1,36
C13	1,706	2,235	-0,765	3,813	-1,709	-2,92
C14	1,353	1,471	0,059	1,990	0,087	0,12
C15	1,118	1,294	0,059	1,446	0,076	0,09
C16	1,059	1,647	-0,824	1,744	-1,356	-1,44
C17	1,412	1,529	-1,000	2,159	-1,529	-2,16
C18	0,647	1,176	-0,824	0,761	-0,969	-0,63
C19	1,471	1,765	-1,000	2,595	-1,765	-2,60

Tabla 5. Resultados encuesta a empresas

	ACU	PRI	GAP	ACU*PRI	GAP*PRI	ACU*PRI* GAP
C1	2,813	3,063	-0,125	8,613	-0,383	-1,08
C2	2,500	2,375	-0,688	5,938	-1,633	-4,08
C3	2,563	3,000	-0,375	7,688	-1,125	-2,88
C4	2,063	2,563	-0,813	5,285	-2,082	-4,29
C5	2,000	2,125	-0,875	4,250	-1,859	-3,72
C6	1,813	2,250	-1,063	4,078	-2,391	-4,33
C7	1,938	2,375	-0,500	4,602	-1,188	-2,30
C8	2,438	2,250	-2,063	5,484	-4,641	-11,31
C9	1,875	2,125	-1,063	3,984	-2,258	-4,23
C10	1,750	2,500	-1,438	4,375	-3,594	-6,29
C11	2,125	2,313	-0,813	4,914	-1,879	-3,99
C12	1,563	1,813	-0,125	2,832	-0,227	-0,35
C13	1,125	1,438	-1,500	1,617	-2,156	-2,43
C14	1,688	2,188	0,563	3,691	1,230	2,08
C15	1,563	1,938	0,625	3,027	1,211	1,89
C16	0,875	1,250	-1,250	1,094	-1,563	-1,37
C17	1,188	1,188	-1,750	1,410	-2,078	-2,47
C18	0,813	1,000	-0,938	0,813	-0,938	-0,76
C19	1,625	1,875	-1,000	3,047	-1,875	-3,05

Tabla 6. Resultados encuesta a profesores

Para cada una de las variables (ACU, ACU*PRI, GAP y GAP*ACU*PRI) construimos unas tablas resumen incluyendo los valores de cada colectivo (alumnos, empresa y profesores) y añadimos su valor medio equiponderado.

5.1. Análisis del nivel de acuerdo (y criticidad)

Como era esperable, el análisis de nivel de acuerdo (ACU) y nivel de acuerdo ponderado con la prioridad o criticidad (ACU*PRI) no muestra datos relevantes respecto la ordenación inicial (recogida en [1]).

Si acaso, citar que CT5 baja en todos los colectivos (y la media), cediendo su lugar entre los cinco primeros , mientras que suben con fuerza CT8 (que ocupa la posición 4), CT11 y CT10 (CT8 y CT10, valorado sobre todo en empresa y profesores, mientras CT11 es menos valorada en la empresa y más por los profesores).

	ALU	EMP	PROF	MED
CT1	6,87	7,06	8,61	7,51
CT3	5,33	6,21	7,69	6,41
CT2	4,54	6,07	5,94	5,52
CT8	4,88	5,40	5,48	5,25
CT4	4,40	5,73	5,29	5,14
CT11	4,67	4,24	4,91	4,61
CT10	3,21	6,23	4,38	4,60
CT7	4,40	4,60	4,60	4,54
CT6	5,64	3,81	4,08	4,51
CT5	3,65	4,43	4,25	4,11

Tabla 7. Ordenación según media de ACU*PRI

5.2. Análisis de los déficit (GAP)

La novedad (esperada, por otra parte) aparece al examinar las valoraciones del GAP y *gap* ponderado (GAP*ACU*PRI).

En este caso, la nueva herramienta nos permite confirmar que las consideraciones iniciales sobre acuerdo o peso de las capacidades apenas tienen nada que ver con los déficit observados. En efecto, se produce una notable inversión en la ordenación de la lista que pasa a ser encabezada en forma abrumadora por CT8 (lengua extranjera).

	ALU	EMP	PROF	MED
CT8	-1,51	-0,24	-2,06	-1,27
CT17	-0,95	-1,00	-1,75	-1,23
CT19	-1,11	-1,00	-1,00	-1,04
CT13	-0,84	-0,76	-1,50	-1,03
CT16	-0,73	-0,82	-1,25	-0,93
CT18	-0,78	-0,82	-0,94	-0,85
CT10	0,00	-0,53	-1,44	-0,66
CT6	-0,54	-0,35	-1,06	-0,65
CT9	-0,30	0,18	-1,06	-0,39
CT2	-0,35	0,06	-0,69	-0,33

Tabla 8. Ordenación según media de GAP

El pequeño tamaño de la muestra da lugar a unos valores relativamente grandes de la desviación de la media pero la mayoría se mantienen entre 0,2 y 0,3 por lo que existe una alta significación de los resultados (que hemos desestimado calcular por innecesario dado el objetivo puramente piloto del estudio).

Al considerar la variable GAP*ACU*PRI, podemos contrastar los resultados anteriores. La lista de los 10 más importantes sigue siendo la misma con la única salvedad de que desaparece C18. Los mecanismos del modelo funcionan como se esperaba y recoge que nuestros encuestados han detectado *"gap"* en esa capacidad pero no sólo hay un bajo nivel de acuerdo sino que no lo consideran crítico. En su lugar, se ha colado C19.

	MED
CT8	-6,65
CT10	-3,20
CT19	-3,05
CT13	-2,93
CT6	-2,91
CT17	-2,43
CT2	-1,77
CT16	-1,61
CT9	-1,55
CT11	-1,45

Tabla 9. Ordenación de *"gaps"* críticos

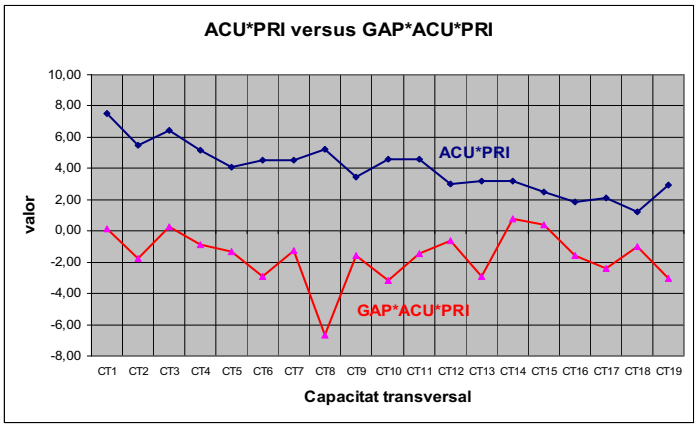


Figura 3. Visión gráfica de los *"gaps"* más críticos en relación a la tendencia ACU*PRI

6. Conclusiones

A partir de justificar la conveniencia de disponer de un mecanismo de evaluación formativa de las capacidades transversales, hemos propuesto un modelo y una herramienta asociada que no solamente permiten identificar el nivel de acuerdo con las capacidades sino también la detección de la desviación en un instante o institución determinados.

Por ello, pensamos que la propuesta podrá ayudar en la planificación curricular en el nuevo marco EEES y también en su mejora continuada, dada la característica formativa de la herramienta.

Además, para poner a punto tanto el modelo como la herramienta, hemos realizado un estudio de campo piloto con un notable nivel de significación estadística a pesar del reducido tamaño de las muestras. Las conclusiones permiten confirmar la sensibilidad de la herramienta a los *"gaps"*.

Referencias

[1] [ANECA 2004] Agencia Nacional de evaluación de la calidad y acreditación, *Libro blanco del Título de grado en ingeniería informática (proyecto EICE)*. www.aneca.es 2004.

[2] [BOE 2005-1] BOE: *Real decreto 55/2005 de 21 de enero por el que se rigen los estudios oficiales de grado*. Publicado martes 25 de enero 2005 (BOE num 21, pag. 2842 y siguientes).

[3] [BOE 2005-2] BOE: *Real decreto 56/2005 de 21 de enero por el que se rigen los estudios oficiales de post-grado*. Publicado martes 25 de enero 2005 (BOE num 21, pag. 2846 y siguientes).

[4] [Ibáñez et al, 2001] Jesús Ibáñez, Julián Gutierrez, Jon Ander Elorriaga y Alfredo Goñi. *El congreso de alumnos como recurso didáctico*. Actas JENUI 2001. Palma de Mallorca. Julio 2001. Pag. 126 a 131.

[5] [Pérez & Virgós] Pérez-Poch, Antoni & Virgós Bel, Ferra. *Un modelo para aplicación sistemática de aprendizaje cooperativo*. Libro de actas de Jenui 2002.

[6] [Virgós 2004] Virgós Bel, Ferran. *La Función TSI en las organizaciones: una evaluación formativa para detección de gaps críticos.* Actas de Jenui 2004. Thomson. ISBN 84-9732-334-3. Pag. 79 a 87. Alicante. Julio 2004.

[7] [Virgós & Tovar], 2005.] Virgós Bel, Ferran; Tovar, Edmundo. *Elementos a considerar en el diseño curricular del nuevo grado en informática.* Actas de JENUI 2005. Ed. Thomson. ISBN 84-9732-421-8. 2005, pag. 155 a 162.

[8] [Virgós 2005] Virgós Bel, Ferran. *Funciones, capacidades, habilidades y actitudes de los informáticos en las empresas: ¿Cómo podemos mejorar?*. Actas de SINDI 2005. Ed. Thomson. ISBN 84-9732-443-9. 2005, pag. 33 a 40. Granada, septiembre 2005.

Experiencias en un plan piloto de adaptación al EEES: Hoja de ruta para coordinadores

Jordi Freixenet[1], Josep Ll. Marzo[1], Josep Soler[2]
[1]Dpto. de Electrónica, Informática y Automática, [2]Dpto. Informática y Matemática Aplicada
Escuela Politécnica Superior, Universitat de Girona
Ed. P4 Campus Montilivi, 17071 Girona
[jordi.freixenet, joseluis.marzo, josep.soler]@udg.edu

Resumen

En este artículo se presenta una guía en formato "hoja de ruta para coordinadores de titulación" que relata la experiencia del equipo de dirección del ámbito informático, sobre la puesta en marcha del ECTS en las titulaciones de ITIG e ITIS. Durante el curso 2004-05 en la Universidad de Girona se inicia un plan piloto de adaptación de titulaciones al EEES. Desde entonces hemos desarrollado un conjunto de actividades que van desde la información, la formación, la coordinación de cursos, la puesta en marcha de herramientas y métodos, la definición de competencias, hasta la evaluación tanto a nivel de asignatura como de titulación. Los destinatarios de estas actividades han sido principalmente el colectivo de profesores y los estudiantes de la titulación, pero también se ha involucrado al PAS y profesores de otras titulaciones. En este artículo se resumen las acciones que hemos llevado a cabo en estos tres cursos, y se invita a reflexionar sobre los éxitos y los problemas que nos hemos encontrado.

1. Introducción

En la primavera del 2004, nuestra universidad nos plantea participar en un plan piloto para adaptarnos al Espacio Europeo de Educación Superior (EEES). Después de varias reuniones sobre la conveniencia de la participación, entendimos que era una oportunidad para rediseñar nuestra docencia bajo el nuevo paradigma. Inmediatamente vimos que no todo debía cambiar radicalmente, sí una asignatura funciona, desde el punto de vista del profesor, de los estudiantes y del aprendizaje que se lleva a cabo ¿para que cambiar?, en todo caso realizar pequeñas adaptaciones. Como punto de partida nos planteamos el paradigma del sistema europeo de transferencia y acumulación de créditos (ECTS) bajo una perspectiva positiva, recogiendo ideas de diferentes fuentes [2, 3, 4, 6, 7]. Después de reflexionar sobre el tema consideramos distintas evidencias, entre ellas:

- Los estudiantes deben saber de antemano y de forma clara cuales son las competencias / objetivos de cada asignatura.
- Los estudiantes deben disponer de herramientas que les faciliten saber si progresan adecuadamente en la asignatura a lo largo del curso.
- Es mejor aprender lo esencial de la materia a fondo que impartir un temario muy extenso, con detalles de dudosa utilidad.
- El deseo de que nuestros alumnos fuesen más autosuficientes, fueran más ingeniosos y capaces de aprender por si mismos.
- Deberíamos formar a los alumnos en competencias genéricas, como el saber liderar, expresarse en público, dominio de varios idiomas, etc.

A partir de estas evidencias, consideramos que era el momento adecuado para la revisión de la titulación y los métodos docentes. Pero, ¿Cómo aplicar todos estos conceptos en una titulación? ¿Cómo convencer a todos los profesores? En este artículo no vamos a facilitar "la solución perfecta" a estas preguntas, pero vamos a relatar nuestra experiencia como coordinadores y promotores del plan piloto de adaptación de las Ingenierías Técnicas en Informática de Gestión y de Sistemas (ITIG e ITIS respectivamente) al EEES.

1.1. Contexto del plan piloto ITIG e ITIS

La participación de los estudios de ITIG e ITIS al plan piloto se aprobó por consenso en el consejo de estudios de las dos titulaciones. La prueba piloto ha consistido en adaptar todas las

asignaturas de estos planes de estudios al formato europeo. Desde el inicio nos planteamos la adaptación como una prueba a largo plazo. Ahora que estamos en el ecuador del tercer año valoramos satisfactoriamente las acciones que hemos desarrollado, pero también somos conscientes del mucho camino que nos queda por recorrer. La ventaja de una prueba de tres años, es que permite ir planificando la formación del profesorado adecuadamente y disponer de tiempo suficiente para cambiar y probar nuevas metodologías docentes. La actitud de los profesores de la titulación ha sido siempre constructiva (también crítica) y el éxito del plan, en nuestra opinión, se fundamenta en el trabajo intenso de los mismos. Por otro lado también hemos ido informando y recogiendo el *feed-back* de los estudiantes.

En el desarrollo del plan piloto se han visto implicados gradualmente alumnos, profesores y todas las asignaturas de los planes de estudios, tal como se muestra en las distintas tablas.

Curso	Primero		Segundo		Tercero	
	ITIG	ITIS	ITIG	ITIS	ITIG	ITIS
04/05	97	94	--	--	--	--
05/06	69	69	79	70	--	--
06/07	69	65	54	52	55	50

Tabla 1. Alumnos plan piloto

Curso	Primero	Segundo	Tercero
04/05	33	--	--
05/06	32	36	--
06/07	30	35	31

Tabla 2. Profesores implicados

Curso	Primero	Segundo	Tercero
04/05	8	--	--
05/06	8	12	--
06/07	8	12	28

Tabla 3. Asignaturas adaptadas

2. Plan de actividades desarrolladas

Las actividades que se han desarrollado para la implantación del plan piloto se basan en cuatro aspectos básicos: coordinación, formación de profesores, diseño de asignaturas dentro del marco ECTS e información a los alumnos.

2.1. Coordinación del plan

Para organizar el desarrollo del plan piloto se establecen tres niveles de coordinación: a) una comisión de coordinación de universidad dirigida por el Vicerrectorado de Docencia y formada por los coordinadores de todos los planes piloto; b) una comisión de coordinación del plan piloto ITIG ITIS, formada por el Subdirector del ámbito de la ingeniería informática y los Coordinadores de las titulaciones ITIG, ITIS; c) y la coordinación de curso, donde se reúnen los responsables de las asignaturas de cada curso con los coordinadores. A continuación detallamos las acciones más importantes de cada comisión.

a) Coordinación a nivel de Universidad. La comisión liderada desde el Vicerrectorado de docencia se reúne con el objetivo de coordinar el desarrollo de todos los planes piloto de la Universidad. Fruto de esta comisión se han desarrollado y mejorado las aplicaciones informáticas básicas de soporte a los planes piloto y que serán las herramientas que se usaran con la puesta en marcha de los futuros Estudios de Grado. Esta aplicación permite a los coordinadores introducir las competencias de su carrera e irlas distribuyendo en las asignaturas correspondientes. Dispone también de todo tipo de cuadros resumen para ver qué competencias trabaja cada asignatura, qué actividades ha planificado el profesor para llevarlas a cabo, etc. Por otro lado la aplicación permite a los profesores responsables de cada asignatura planificar las actividades y los contenidos asociados que llevaran a cabo para que el alumno adquiera las distintas competencias (tanto genéricas como específicas de la asignatura). El objetivo final es que cada asignatura se plantee según su "diseño en formato ECTS". En las figuras 1 y 2 se puede ver la interfaz de entrada de competencias y las actividades de una asignatura.

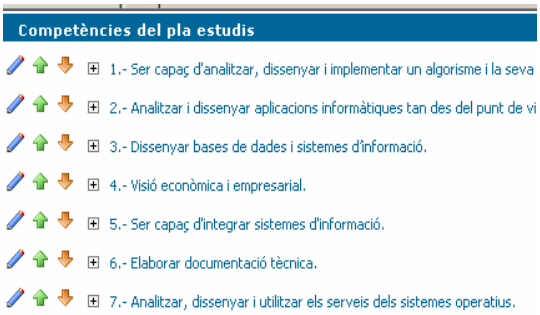


Figura 1. Herramienta para asignar competencias a la titulación.

Tipus activitat	Hores AP	Hores SP	Total hores
Altres	0	39,50	39,50
Classes expositives	1,00	0	1,00
Classes participatives	25,00	0	25,00
Classes pràctiques	10,00	7,50	17,50
Prova d'avaluació	9,00	0	9,00
Resolució d'exercicis	13,00	32,50	45,50
TOTAL	**58,00**	**79,50**	**137,50**

Figura 2. Herramienta para asignar las competencias de la titulación a las asignaturas.

b) Coordinación a nivel de titulación. En esta comisión los protagonistas son los coordinadores de titulación y en nuestro caso nos reunimos al menos quincenalmente. En estas reuniones se planifican y coordinan las acciones del plan piloto, el desarrollo del curso, los resultados académicos, etc.

c) Coordinación a nivel de curso. A mediados y al final de cada semestre se analizan, tanto las opiniones de los estudiantes (recogidas en las encuestas), como de los profesores y tutores en el caso de primer curso (ver apartado 2.4). Al finalizar cada curso nos reunimos con los profesores responsables de asignatura con el objetivo de planificar, coordinar y consensuar el diseño de las asignaturas para el siguiente curso. Es decir, se informa, analiza y discute las actividades que cada una de las asignaturas piensa llevar a cabo. También evaluamos los resultados académicos y el grado de satisfacción acerca del desarrollo de las actividades de cada asignatura. En el presente curso estas reuniones de coordinación se llevan a cabo de manera independiente para cada uno de los tres cursos.

2.2. Formación de profesores

La formación de los profesores se ha organizado en tres tipos de acciones, ya que según ha ido avanzando el plan hemos tenido diferentes necesidades.

- Jornada de formación inicial. Durante tres años se ha organizado esta jornada que supone el punto de partida para muchos de los profesores de la titulación. Aunque la jornada va dirigida especialmente a los profesores de un determinado curso, es abierta a cualquier otro profesor interesado. Durante la jornada se realizan cuatro presentaciones:

1) El EEES y los créditos ECTS.

2) Experiencias y ejemplos concretos de asignaturas y técnicas docentes en el contexto ECTS. Algunas presentadas por profesores de otras universidades.

3) El plan piloto ITIS e ITIG. Presentación de las acciones llevadas a cabo. Presentación de la web del plan piloto.

4) Presentación de la ficha de asignatura del portal docente de nuestra universidad.

Finalmente, se aprovecha la jornada para coordinar y consensuar el calendario para la programación de las asignaturas del curso.

- Los viernes docentes. Durante las primaveras del segundo y tercer año se organiza y desarrolla el ciclo "Los viernes docentes ITIG ITIS", como un conjunto de coloquios, intercambio de experiencias de los profesores ITIG ITIS que han intervenido en el diseño de asignaturas o nuevas metodologías docentes en el marco del plan piloto. En cada sesión, de las 5 realizadas, se presentan dos asignaturas, y se analiza y discuten sus diseños, las experiencias obtenidas y los planes de mejora. Los materiales del ciclo se publican en la web del plan que se desarrolla bajo la coordinación del Instituto de Ciencias de la Educación (ICE).

- Cursos y jornadas organizadas por el ICE y el Vicerrectorado de Docencia, que van dirigidas a todo el conjunto de profesores de la universidad. Destacar que en estos dos últimos años se ha incrementado notablemente el número de participantes, especialmente en los cursos donde se debaten y exponen técnicas docentes concretas (aprendizaje cooperativo, técnicas de evaluación, etc.).

2.3. Diseño de las asignaturas en formato ECTS

En la actualidad todas las asignaturas de ITIG e ITIS ya están formuladas siguiendo la filosofía del EEES, y se puede consultar el diseño y el plan de actividades de cada asignatura a partir de la web. Los estudiantes tienen acceso a la intranet docente donde todas las asignaturas han sido formuladas con la misma herramienta, y entre otros recursos se dispone de un calendario/agenda de curso, acceso a la plataforma de *e-learning ACME* [5] utilizada en distintas asignaturas, un forum, etc.

Definición y revisión de competencias.

Los planes de estudios de ITIG e ITIS corresponden a titulaciones oficiales reguladas por el BOE, de manera que la adaptación al EEES por enseñanza basada en competencias parte de unas asignaturas con contenidos ya fijados.
Así, el trabajo llevado a cabo en competencias no ha sido de definición y construcción de un plan de estudios, es decir no nos planteamos cuales eran las competencias que queríamos para los titulados y en función de estas competencias definimos el plan de estudio. De manera que la definición de competencias en el plan piloto resulta un ejercicio de revisión y actualización a partir de los objetivos de cada una de las asignaturas. En una fase posterior estudiaremos que competencias no se trabajan (o no suficientemente) y de que manera resulta apropiado introducirlas.
A continuación relatamos el proceso metodológico que hemos seguido. Inicialmente, se definieron las competencias genéricas basadas en el Libro Blanco [1] y fueron los coordinadores quien asignaron competencias a la titulación. A partir de esta primera propuesta, se desarrollaron reuniones con los profesores responsables de asignatura con el objetivo de fijar qué competencias trabajaría cada asignatura. Se dispuso en la ficha de la asignatura de un apartado especial llamado "otras competencias" y que cada profesor responsable puede definir de forma libre para describir un mayor grado de concreción.
Como resultado de distintas reuniones de trabajo y conjuntamente con la comisión a nivel universidad, se establecieron unos criterios generales de redacción de competencias:
- Cada asignatura debe tener entre 4 y 8 competencias (de las cuales no debería haber más de 3 o 4 específicas).
- Las competencias que aparecen en el apartado "otras competencias" no van a aparecer en el cómputo general. De esta forma se facilita la definición de las competencias de la titulación.
- Recomendamos seguir algunas normas sencillas que ayudan a definir y uniformizar competencias para dar la máxima claridad posible y orientar tanto a profesores como estudiantes. Por ejemplo evitar el uso de verbos del tipo "saber", conocer", etc. y utilizar expresiones como "analizar", "evaluar", etc. Se proporcionó una lista completa de expresiones y ejemplos claros. Aunque al principio puede dar la sensación de ser un sistema excesivamente rígido, al final se agradece y se evitan discusiones estériles sobre su semántica. Con el tiempo se asumen perfectamente las nuevas expresiones.

2.4. Actividades con los estudiantes

Sin lugar a dudas el colectivo de los alumnos juega un papel muy importante en este plan piloto. Desde un primer momento nos propusimos que estuvieran bien informados respecto al EEES y al plan piloto que estamos llevando a cabo. Para ello

establecimos varios cauces de formación e información.

A mediados de abril se realiza una jornada de puertas abiertas en la que acuden la mayoría de alumnos de Bachillerato y Ciclos Formativos interesados en los estudios de ITIS/ITIG. Además de mostrarles una visión general de los estudios se les informa del plan piloto que se está realizando y de lo que va a suponer el EEES.

A finales de julio, durante la jornada de bienvenida, se reúne a los nuevos alumnos y se les informa del plan piloto y de lo que les va a suponer de cara a su formación. Durante este acto se les entrega la guía docente en la cual se les indica quien va a ser su profesor tutor. El tutor es un profesor de la titulación que se ocupa de aconsejar y guiar al estudiante durante su primer año en la universidad. Ningún alumno podrá matricularse sin antes acudir a una primera tutoría. El tutor en función de la situación/características del alumno le orientará sobre su matrícula, le informará en detalle sobre el plan piloto, fijarán la fecha de su próxima cita y finalmente activará la aplicación de matrícula para que el alumno pueda formalizar el proceso de auto-matrícula.

Otro cauce de información son los profesores y el diseño de cada asignatura. Durante la primera sesión de cada asignatura el profesor informa del plan piloto, de las competencias que deben adquirir, de las actividades planificadas, del trabajo que conlleva un crédito ECTS, etc. El alumno encontrará esta información en la ficha diseño de la asignatura.

Quizás pueda parecer que insistimos mucho en mentalizar a nuestros alumnos del nuevo sistema pero consideramos que es una labor fundamental para facilitar su integración al plan piloto.

Finalmente consideramos que tan importante es tener bien informados a los alumnos como recabar sus opiniones. Para ello realizamos dos encuestas cada semestre (inspiradas a partir de [6]) en las que se pregunta a los estudiantes, por ejemplo, si se sienten bien informados sobre su progreso en la asignatura, que nos indiquen su nivel de dificultad, las horas que le dedican, etc. Además les pedimos que escriban un mínimo de dos aspectos positivos y dos negativos de cada asignatura para detectar posibles problemas y deficiencias. Los profesores valoran muy positivamente estas encuestas por doble motivo: por un lado les permite reaccionar delante de problemas y por otro el estudiante percibe que su opinión es importante para el profesor.

2.5. Transferencia de experiencias a otras titulaciones de la EPS

En la Universidad de Girona (UdG), los Estudios de ITIG/ITIS se imparten en la Escuela Politécnica Superior (EPS) junto con otras titulaciones. Desde el inicio del segundo año del plan piloto, se creó en la EPS una Comisión para la formación inicial del profesorado en términos del EEES. Este grupo de profesores formado por representantes de todas las áreas de conocimiento se reúne una vez al mes y trabaja en la formación e intercambio de experiencias en el marco de nuevas metodologías docentes y utilización de las herramientas desarrolladas por nuestra universidad. El objetivo es formar como mínimo un profesor de cada área de conocimiento de la EPS para que sea partícipe de todas las acciones desarrolladas en la UdG relacionadas con el EEES. Las sesiones organizadas han sido las siguientes:

- Sesión 1. Créditos ECTS. Formación en competencias.
- Sesión 2. Libros blancos de titulación de grado. Inicio del taller "Diseño en ECTS y competencias". Actividades de aprendizaje.
- Sesión 3. El Plan piloto de adaptación al EEES en la UdG. El plan piloto ITIG ITIS.
- Sesión 4. Información sobre el catálogo de grado. Taller: diseño de asignatura con la herramienta de la universidad. Introducción al aprendizaje cooperativo.
- Sesión 5. Información directrices. Grado Química y Grado Informática. Experiencias en Aprendizaje cooperativo.
- Sesión 6. Presentación de la plataforma de *e-learning ACME*. Experiencias en diferentes asignaturas.
- Sesión 7. Uso de la herramienta de diseño de la asignatura de la universidad. Debate sobre el documento "competencias" editado por la universidad.
- Sesión 8. Técnicas de evaluación.
- Sesiones 9 y 10. Presentación de la ficha de la asignatura adaptada al EEES por cada uno de los miembros de la comisión.

2.6. Otras acciones

Entre otras acciones que también se han realizado, queremos destacar dos jornadas de información dirigidas al PAS, informándoles sobre las acciones del plan y de todos los cambios que se avecinan. Consideramos importante que este colectivo esté bien informado respecto al nuevo EEES.

Finalmente comentar que se ha constituido una Comisión para estudiar y elaborar los nuevos planes de los Estudios de Grado y Master. La experiencia de estos tres años de plan piloto será sin duda, una buena base para empezar a trabajar.

3. Satisfacción de estudiantes y profesores.

Para evaluar la satisfacción de los estudiantes se analizan los resultados de las encuestas presentadas en el apartado 2.4. En general, los estudiantes se muestran satisfechos ante las acciones del plan piloto, como por ejemplo la utilización de la herramienta de *e-learning ACME*, la inclusión de evaluación continua, el hecho que los profesores resalten qué es realmente importante, indiquen exactamente qué problemas hacía falta solucionar y el tiempo estimado por su resolución, etc. En la figura 3 se detalla el resultado de la encuesta al contestar la pregunta "¿Te sientes bien informado de tu progreso en la asignatura?". En general, un porcentaje mayoritario de estudiantes se manifiesta bien informado, y en algunas asignaturas se observa un incremento en el grado de afirmación.

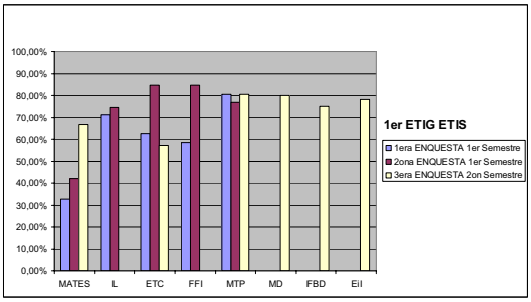


Figura 3. Resultados de las encuestas a la pregunta "¿Te sientes bien informado de tu progreso en la asignatura?" dirigida a los estudiantes ITIG e ITIS

Por otra parte, el grado de satisfacción de los profesores con respecto al plan se ha evaluado mediante encuestas específicas para profesores. La tabla 4 muestra los resultados de algunas de las cuestiones, donde puede observarse un grado muy alto de satisfacción. Además, fruto de las reuniones que hemos mantenido con ellos, en general queremos subrayar un grado de compromiso, participación e implicación máximo (100%). Si bien, como es normal y enriquecedor hay opiniones y críticas de todo tipos, pero no hay absentismo. En este sentido queremos constatar que no habido ningún profesor que haya expresado que no quiere participar en el plan.

	Totalmente de acuerdo	Bastante de acuerdo	Bastante desacuerdo	Total. en desacuerdo	NS/NC
1.1.- La participación en el plan piloto ha sido positiva para mi	55%	45%			
1.3.- Las reuniones de coordinación de curso han sido útiles	10%	90%			
1.4.- Las encuestas que responden los estudiantes sobre el plan son útiles	20%	35%	25%	5%	15%
1.6.- Utilizar la herramienta de diseño de la asignatura es una buena práctica	40%	35%	20%	5%	
1.8.- He dedicado más tiempo a la docencia	55%	35%	5%		5%
1.9.- He realizado mejor mis tareas docentes	25%	60%		5%	10%
1.10.- Hemos estado perdiendo el tiempo			15%	75%	10%

	SI	NO
2.1.- Has participado en alguna actividad de formación durante el plan piloto?	75%	25%
3.1.2.- He rediseñado mi asignatura siguiendo la filosofía ECTS	100%	
3.1.3.- He adecuado los contenido	100%	
3.1.4.- He incorporado nuevas actividades de aprendizaje	95%	5%
3.1.5.- He incorporado nuevos materiales	90%	10%

Tabla 4. Resultados de algunas preguntas de la encuesta dirigida a los profesores participantes en el plan.

4. Resultados académicos

En general, por los datos analizados hasta el momento, no creemos que se puedan extraer conclusiones sobre el impacto del plan en los resultados académicos. El curso 2005-06 ha sido el segundo año de plan piloto para las asignaturas de primero, y creemos que podemos valorar con más precisión sus resultados académicos. Pese a

esto, insistimos ya de entrada que no creemos que tras dos años se pueda establecer una relación directa entre la implantación del plan piloto y la mejora o empeoramiento de los resultados académicos. Es evidente que el plan está sirviendo para reflexionar en nuestra tarea docente y mejorar nuestra docencia, pero esto no implica una mejora directa de los resultados académicos. Con seguridad hay otros factores que afectan a los resultados académicos, por ejemplo la preparación de base de los nuevos estudiantes, la nota de acceso de una determinada promoción, o la sinergia que se pueda crear en el grupo, etc.

En el balance del curso pasado, si bien hay más asignaturas dónde los resultados académicos mejoran, en otras empeoran. Cabe decir que nos queda pendiente realizar un análisis más detallado de los resultados académicos, que incluya aspectos como la calidad de los estudiantes aprobados, o por ejemplo, tener en cuenta la nota con la cual han accedido a la universidad, etc.

5. Conclusiones

En este artículo se han descrito las acciones realizadas como coordinadores de ITIG e ITIS en el marco del plan piloto de adaptación al EEES. Después de casi tres años de adaptación somos capaces de señalar evidencias, aciertos y carencias. Agrupamos las consideraciones finales, bajo tres puntos de vista: el profesor, los estudiantes y las asignaturas.

Consideraciones que hacen referencia al profesorado:

- A pesar de algunas reticencias iniciales por parte de algún profesor, finalmente todo el profesorado se ha involucrado de forma activa en el plan.
- Es difícil determinar con precisión todas las competencias de una titulación y repartirlas en las distintas asignaturas. En el proceso de definición de competencias se sufrieron etapas alternativas entre excesiva simplificación y excesivo detalle. Con el tiempo se vio la conveniencia de reducir y ajustar el número y el grado de detalle de las mismas.
- La asignación de una misma competencia a varias asignaturas complica su gestión, por ejemplo en su evaluación.
- La opinión de los profesores acerca del nivel de formación de los nuevos alumnos, les sirve de justificación del bajo rendimiento académico. La percepción que tienen algunos profesores es de poca capacidad y dedicación, excesivo absentismo y una actitud muy pasiva, especialmente en estudiantes de primer curso.
- El mero hecho de rediseñar las asignaturas en el nuevo formato ECTS, conlleva a un mejor enfoque docente. Se han replanteado las materias impartidas en todas las asignaturas.
- Sin lugar a dudas ha supuesto una mayor dedicación y esfuerzo por parte de todos los profesores.

Consideraciones acerca de los estudiantes:

- El sistema de evaluación continuada que han propuesto la mayoría de asignaturas, con entregas de prácticas más a menudo, pequeños test de seguimiento, entregas de problemas, etc. ha sido valorado positivamente, aunque en algunos momentos pueda resultarles un poco estresante.
- El correcto seguimiento por parte del alumno de la evaluación continua supone el éxito en la mayoría de los casos.
- Consideramos que algunos estudiantes se resisten a cambiar de hábitos, y están muy acostumbrados a estudiar sólo cuando se acercan los exámenes.
- La compaginación de estudios y trabajo es un obstáculo a veces insalvable. Muchas asignaturas se han diseñado con dos opciones de evaluación, una para los que siguen la evaluación continua y otra para el resto. Algunos profesores se plantean como opción adicional un enfoque semipresencial en algunas asignaturas.
- La asistencia a clase ha descendido en los últimos años. Este es uno de los retos que todavía tenemos pendientes. Uno de los motivos de la baja asistencia y que nos comentan los alumnos es “que todo lo se explica en clase lo encontramos en la web de la asignatura”. Este es un tema de reflexión.

- A partir de las encuestas realizadas (tanto a profesores como a estudiantes) hemos observado problemas en la estimación del tiempo que los estudiantes dedican a las asignaturas.

Consideraciones acerca de las asignaturas:

- En las asignaturas con un gran número de estudiantes es más difícil y complejo aplicar nuevas metodologías docentes. Las asignaturas con menor número de estudiantes se prestan mucho más a todo tipo de innovaciones. Hay que seguir trabajando y probando nuevas metodologías.
- Es muy importante el establecimiento de las competencias de cada asignatura, las actividades que se desarrollaran para llevarlas a cabo y los contenidos que se van a trabajar.
- Es esencial disponer de una página web de cada asignatura, en donde figure toda la información necesaria para el alumno, desde el diseño de ésta en formato ECTS, la evaluación, la bibliografía y en donde se pueda dejar el material.
- Una de los temas que tenemos pendientes es la evaluación de las competencias genéricas.
- Otro de los temas pendientes es la realización de prácticas/proyectos conjuntos entre dos o más asignaturas.
- La utilización de plataformas de *e-learning* motiva a los alumnos y en general el rendimiento obtenido ha sido muy bueno.

Tal como hemos descrito en este artículo, en nuestra universidad hemos ido trabajando nuevas propuestas y mecanismos para adaptar las titulaciones de ITIG e ITIS al EEES. Tenemos la impresión de haber hecho muchas cosas, pero al mismo tiempo somos conscientes que aún nos queda mucho camino por recorrer.

Agradecimientos

Queremos agradecer a todos los profesores de ITIG e ITIS, y a todos aquellos que nos han ayudado con sus ideas y su trabajo durante estos tres años de plan piloto. También queremos agradecer a los estudiantes sus iniciativas e ideas y su predisposición a colaborar. Gracias también a la dirección de la EPS, al ICE, al Vicerrectorado de Docencia y al Gabinete de Planificación y Evaluación, por el apoyo y el *feedback* que nos han dado.

Referencias

[1] Casanovas J, Colom JM, Morlán I, Pont A, Ribera M. *El libro blanco de la Ingeniería en informática: el proyecto EICE.* JENUI 2004, http://www.aneca.es/modal_eval/docs/librobl anco_informatica.pdf

[2] Jacob I, Oliver J, García J, Sáenz JM, Díaz J. *Formando en competencias, el caso práctico de una facultad.* JENUI 2006, pág. 49-55.

[3] Pavón, N. *¿Están los alumnos preparados para el Tour de Francia? Comportamientos, hábitos y Sistema de Créditos Europeo* JENUI 2004, pág. 39-46.

[4] Sánchez F. *¿Cómo serán las asignaturas del EEES?* JENUI 2005, pág. 147-154.

[5] Soler J, Prados F, Boada I, Poch J. *Utilización de una plataforma de e-learning en un plan piloto de adaptación al EEES.* 4to Congreso de Docencia Universitaria e Innovación CIDUI 2006.

[6] Valero M. *¿Cómo nos ayuda el Tour de Francia en el diseño de programas docentes centrados en el aprendizaje?* NOVATICA, num. 170, Julio-Agosto 2004, pág. 42-47.

[7] Zabalza M.A. *Guía para la planificación didáctica de la docencia universitaria en el marco del EEES.* Documento de trabajo, Universidad de Santiago de Compostela. http://www.net.upcomillas.es/innovacioneduc ativa/Documentos/guiaplan.pdf. 2004.

Metodología de trabajo y experiencias de aprendizaje colaborativo y evaluación continua en la disciplina de Sistemas Multiagente

Francisco Bellas Bouza y Amparo Alonso Betanzos
Dpto. de Computación. Facultad de Informática
Universidad de A Coruña
Campus de Elviña, 15071 A Coruña
fran@udc.es, ciamparo@udc.es

Resumen

En este trabajo se exponen las experiencias realizadas para la adaptación al Espacio Europeo de Educación Superior de la asignatura Sistemas Expertos, de la titulación de Ingeniería Informática en la Universidad de A Coruña. El nuevo planteamiento se centra principalmente en la realización de actividades colaborativas, la incorporación de recursos virtuales y el sistema de evaluación continua empleado, que son posibles, en gran parte, debido a que el número de alumnos matriculados en la asignatura (una media de 21 en los dos últimos cursos), es adecuado para este tipo de experiencias. Con este planteamiento, el 95% de los alumnos matriculados (100% de los presentados) superaron la materia y demostraron un alto nivel de asimilación de los conceptos.

1. Introducción

La asignatura de Sistemas Expertos en la Universidad de A Coruña es una asignatura optativa de segundo ciclo que se imparte en la titulación de Ingeniería Informática, con una carga lectiva total de 6 créditos, 4 de los cuales son teóricos y 2 son prácticos. Esta asignatura forma parte de un grupo de asignaturas de la Facultad de Informática en nuestra Universidad, que planifican una nueva Guía Docente para la adaptación al Espacio Europeo de Educación Superior. En nuestro caso, la nueva guía adapta esta asignatura a 5 créditos ECTS. Estas guías docentes han sido elaboradas empleando la plantilla creada por la Universidad de A Coruña mediante el proyecto GADDU (Guía de Armonización para el diseño de la Docencia Universitaria) [1].

Dado el carácter optativo de esta asignatura, y que además se imparte en el segundo ciclo de una ingeniería superior, la filosofía que se ha tratado de imponer al desarrollo de los objetivos docentes de la misma se basa en incentivar el trabajo personal del alumno y la capacidad de resolución de problemas por sí mismo. De esta forma, se valorará altamente la asistencia y la participación en las clases, pudiendo el alumno obtener la calificación final de la asignatura mediante trabajos y sin necesidad de realizar un examen final escrito. Además, consideramos importante fomentar el trabajo en equipo y la colaboración, fundamentales en la vida profesional de los Ingenieros Informáticos, además de potenciar el autoaprendizaje, por supuesto guiado, de los alumnos. La metodología usada y los objetivos de la asignatura se ven claramente favorecidos por el hecho de que el número de alumnos matriculados es moderado (durante los dos últimos cursos la media es de 21 alumnos).

Desde el curso 2004/2005, el contenido temático básico de esta asignatura ha cambiado de los Sistemas Expertos clásicos del campo de la Inteligencia Artificial, a los Sistemas Multiagente que se imparten en la actualidad. Este es un campo de estudio reciente en sus planteamientos y desarrollos, que aumentan cada día, lo cual confiere unas características especiales a la asignatura en cuanto a la novedad de los contenidos. Esto nos ha permitido, por una parte, acercar a los alumnos al mundo de la investigación cuando están a punto de finalizar el segundo ciclo, y por la otra, incorporar conocimientos recientes que se utilizan ya a nivel industrial, siendo éste un aspecto muy importante en carreras como la Ingeniería Informática.

Hemos establecido una serie de objetivos concretos para la asignatura, que desde el punto de vista de los conocimientos son:

- Introducir el concepto de Sistemas Multiagente a partir de la necesidad de

arquitecturas distribuidas en los sistemas inteligentes.
- Presentar las diferentes aproximaciones a las arquitecturas de los Sistemas Inteligentes.
- Presentar la noción de Negociación como un aspecto básico inherente a los sistemas multiagente.
- Presentar las nociones y los aspectos básicos de la coordinación, la cooperación y la comunicación, también imprescindibles en sistemas de este tipo.
- Presentar los aspectos básicos de una herramienta que cumple los estándares del campo, como JADE, y capacitar al alumno para la implementación de sistemas sencillos usando esta herramienta.

Los objetivos de la asignatura en cuanto a competencias y actitudes son los siguientes:

- Lograr que el alumno adquiera la capacidad de llevar a cabo la resolución de problemas, en este caso particular, relativos al desarrollo y la implementación de un sistema multiagente.
- Lograr que el alumno adquiera la capacidad de aplicar los conocimientos teóricos a la práctica.
- Lograr que el alumno adquiera la capacidad de adaptarse a nuevas situaciones.
- Lograr que el alumno sea capaz de trabajar de forma colaborativa con sus compañeros.
- Fomentar el aprendizaje autónomo del alumno

2. Metodología

Como hemos mencionado anteriormente, el objetivo principal de esta asignatura se puede resumir en *enseñar al alumno a resolver problemas que necesiten de arquitecturas computacionales distribuidas y cooperativas flexibles*. En líneas generales, nuestra propuesta para adaptarnos a una distribución de créditos ECTS consiste en:

- Utilizar las clases magistrales para establecer los conceptos fundamentales de la materia.
- Desarrollo de trabajos tutelados por los profesores de la asignatura. Los trabajos están basados en los contenidos desarrollados en las clases teóricas y pretenden ampliar los conocimientos del alumno en la materia. El profesor guiará al alumno en el desarrollo de estos trabajos para establecer unos contenidos mínimos y delimitar el alcance de los mismos.
- Desarrollo de trabajos en grupos cooperativos durante las clases presenciales. El objetivo de estos trabajos es favorecer el aprendizaje en grupo del alumno así como fomentar diversas competencias, entre ellas la de trabajo en grupo.
- Desarrollo de prácticas de programación en una herramienta que se ajuste a los estándares del campo. Estos ejercicios pretenden que el alumno adquiera experiencia en la construcción, depuración y prueba de programas y su uso en diversas aplicaciones prácticas.
- Desarrollo de foros de discusión sobre temas puntuales de la asignatura.
- Estudio de casos en las clases presenciales, con trabajo previo del alumno.
- Empleo y manejo de recursos virtuales para favorecer el autoaprendizaje del alumno y su autonomía.

Para llevar a cabo cada uno de estos apartados y como soporte al alumno, se emplea mayoritariamente el programa informático Moodle [2], una plataforma de formación a través de la Web que permite la gestión de un curso virtual y que ya hemos utilizado el curso 2005/06 con éxito notable. Este curso se pretende potenciar su uso para fomentar la capacidad de aprendizaje autónomo del alumno. La herramienta está basada en diversos principios pedagógicos para ayudar a los educadores en el desarrollo de comunidades de aprendizaje on-line. Moodle es una herramienta de libre distribución y gratuita, por tanto su implantación no supone ningún coste económico adicional, aunque sí es necesario un coste temporal en cuanto a la instalación y configuración del sistema. Paralelamente, también se utilizarán los recursos disponibles en la Facultad Virtual de la Universidade da Coruña [3], aunque esta herramienta posee ciertas limitaciones respecto a Moodle, por lo que será utilizada básicamente como apoyo. Esta decisión se debe a que Moodle permite realizar pruebas de autoevaluación interactivas, entrega de actividades individuales y de grupos, encuestas sobre la asignatura, etc. y otros aspectos atractivos que no

es posible realizar en el entorno que nos proporciona la Universidad. En la Figura 1 se muestra la página principal de la asignatura en el entorno Moodle.

3. Estructura teórica de la asignatura

Las clases teóricas de la asignatura suponen 4 de los 6 créditos de la misma, es decir, un 66% de su contenido total. Son de carácter presencial y se basan en una clase magistral impartida haciendo uso de transparencias que son proyectadas con un cañón de video. La asistencia a las clases de teoría no es obligatoria, pero es valorada en la calificación final de la asignatura.

Los contenidos teóricos están agrupados en cuatro bloques temáticos básicos, que se detallan a continuación:

3.1. Bloque temático I: Conceptos básicos

Se introducen las nociones básicas de sistemas inteligentes distribuidos y de agentes. Este bloque contiene los tres primeros temas del temario oficial de la asignatura:

- Tema 1. Introducción.
- Tema 2: Agentes. Aspectos Generales.
- Tema 3: Agentes Inteligentes.

En este bloque inicial se proporcionará al alumno una visión moderna de los sistemas inteligentes y los recursos necesarios para consultar los trabajos de investigación más relevantes del campo.

Trabajos propuestos (no presenciales)

Se propone un trabajo inicial de búsqueda en Internet de aplicaciones reales y actuales de sistemas multiagente. El objetivo es que el alumno adquiera una dimensión del campo del que trata la asignatura. La actividad está tutorada por los profesores de la asignatura de forma presencial y de forma virtual. Un segundo objetivo de esta

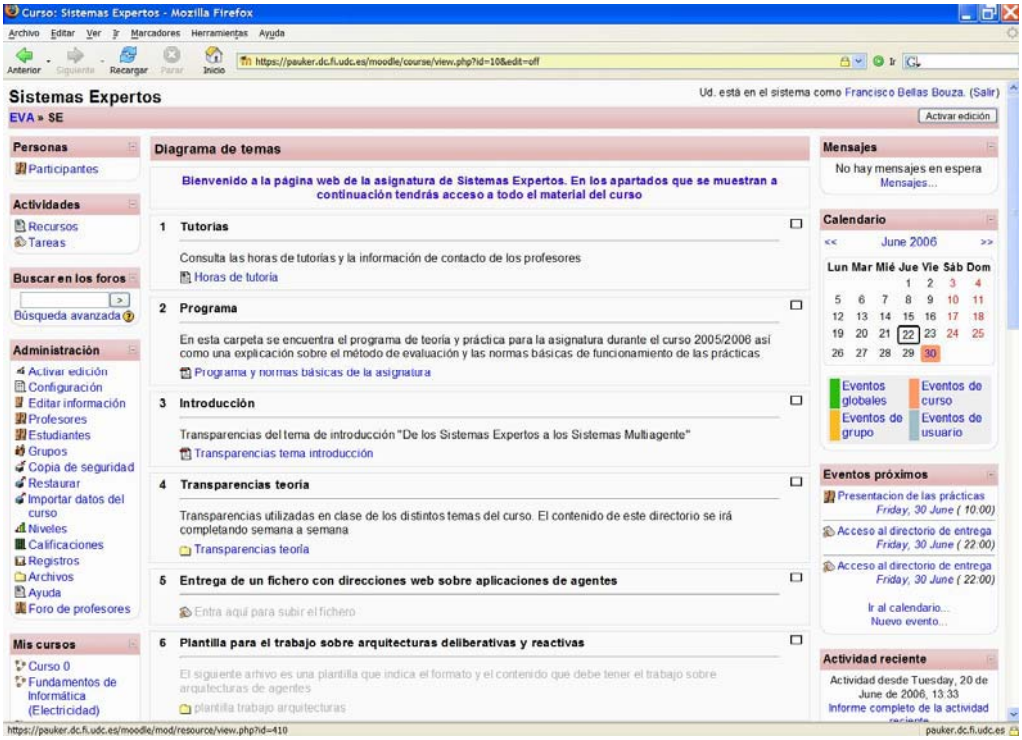


Figura 1. Página principal de la asignatura en la herramienta docente Moodle

actividad es que el alumno pueda plantearse qué tipo de aplicación le interesaría presentar en el trabajo teórico final, otra de las actividades que se propone a los alumnos de la asignatura al finalizar el cuarto bloque temático.

Este trabajo es de carácter individual y las especificaciones en cuanto a longitud y características del contenido son proporcionadas por el profesor mediante la herramienta docente *Moodle*.

3.2. Bloque Temático II: Arquitecturas de Agentes

Los temas concretos a los que se hace referencia en este bloque son:

- Tema 4.- Agentes de razonamiento deductivo
- Tema 5.- Agentes de razonamiento práctico
- Tema 6.- Agentes reactivos y Agentes híbridos

En estos temas se trata un aspecto básico de los agentes, la arquitectura que les proporciona capacidades inteligentes. Se hace un repaso exhaustivo a las principales características de los tres tipos de arquitecturas existentes: deliberativas, reactivas o híbridas.

Trabajos propuestos (no presenciales)

Antes de comenzar la explicación del tema 6 sobre arquitecturas deliberativas, se plantea a los alumnos un trabajo donde se muestren con claridad las diferencias entre las arquitecturas deliberativas y reactivas para agentes. El objetivo de esta actividad es mostrar que la problemática del tipo de aproximación escogida, desconocida para la mayoría de los alumnos, ha sido muy relevante en estas últimas décadas. La principal fuente de información debe ser, de nuevo, Internet, pero en este caso también es posible recurrir a los textos de referencia en la bibliografía de la asignatura. El alumno debe comentar como conclusión la solución que ha sido adoptada como más eficiente en el campo, esto es, una aproximación híbrida. Dicha conclusión se puede extraer de una lectura calmada de cualquiera de los textos que tratan esta problemática.

Para la realización de este trabajo, se proporciona a los alumnos (por medio de la Web de la asignatura en *Moodle*), una plantilla con el formato básico y la longitud mínima (y máxima), en páginas. Asimismo, se establece un directorio de entrega del trabajo, ambas cosas a través de la página Web de la asignatura en Moodle. El planteamiento que hemos elegido para la implementación de este trabajo es la variante de aprendizaje colaborativo Jigsaw [4]. Los grupos constan de dos alumnos, uno de los cuales se especializaría en arquitecturas deliberativas y el otro en arquitecturas reactivas. Posteriormente, los grupos se intercambiarán de forma que los alumnos que estudian el mismo tipo de arquitectura se reúnan en un mismo grupo, que elaborará en detalle el aspecto a estudiar. Finalmente, cada estudiante retornará a su grupo original para elaborar un trabajo final.

3.3. Bloque Temático III: Interacción, cooperación y negociación en Sistemas Multiagente

Los temas concretos a los que se hace referencia en este bloque son:

- Tema 7.- Interacciones multiagente
- Tema 8.- La negociación entre agentes.
- Tema 9.- La comunicación entre agentes.
- Tema 10.- La cooperación entre agentes.

En los temas de este tercer bloque se trata un aspecto básico de los sistemas de agentes, la comunicación e interacción entre ellos. El estudio general de las interacciones entre agentes requiere establecer un protocolo de comunicaciones eficiente y fiable y un protocolo de negociación general que permitan que los agentes cooperen en la realización de una tarea.

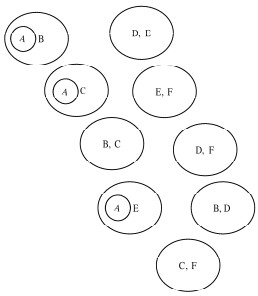
Trabajos propuestos (no presenciales)

Durante este bloque, los alumnos deben realizar una tarea de discusión sobre el tema de metodologías de agentes. Así, se organizan en grupos de dos alumnos, cada uno de los cuales está especializado en una metodología particular, de entre un grupo de metodologías a estudiar que propondrá el profesor. Por tanto, cada grupo estará especializado en dos metodologías. Posteriormente, en una segunda fase, se formarán grupos homogéneos con los alumnos que se hayan

especializado en cada una de las distintas metodologías, para profundizar en ellas. Finalmente, los alumnos vuelven a su grupo inicial, en donde se debe elaborar un informe que compare las ventajas e inconvenientes de las dos metodologías, realizando un estudio comparativo (figura 2) . Como vemos, de nuevo aplicamos una estrategia de aprendizaje colaborativo en su variante de Jigsaw [4]. Con esta actividad, se pretende introducir al alumno en la problemática de la falta de estandarización en el campo de los sistemas multiagente, además de fomentar su capacidad crítica. La plantilla para la realización del trabajo y el directorio de entrega del mismo están disponibles en la página Web de la asignatura. En la plantilla se incluye un apartado de comparativa con otras metodologías y de ventajas e inconvenientes de la metodología presentada.

METODOLOGÍAS PROPUESTAS : A, B, C, D, E, F

GRUPOS 1ª FASE



1	A,B
2	A,C
3	D,C
4	D,E
5	E,F
6	D,F
7	A,E
8	B,D
9	C,F

GRUPOS 2ª FASE : El alumno del grupo especializado en una metodología , por ejemplo , la A, forma un nuevo grupo . En este ejemplo , los nuevos grupos son de tres alumnos , todos especialistas en la misma metodología .

METODOLOGÍA

A	1,2,7
B	1,3,8
C	2,3,9
D	4,6,8
E	4,5,7
F	5,6,9

Figura 2. Estrategia de Jigsaw aplicada en el trabajo sobre metodologías de agentes.

3.4. Bloque Temático IV: Ejemplos de Sistemas Multiagente

El tema al que se hace referencia en este bloque es:

- Tema 11.- Aplicaciones

Los contenidos teóricos de este tema serán presentados por los propios alumnos mediante trabajos y la exposición en clase de los mismos.

Trabajos propuestos (no presenciales)

En este bloque, el alumno realizará de nuevo en grupos de dos, un trabajo teórico, consistente en el estudio de una aplicación práctica de los sistemas multiagente de su elección. Este trabajo deberá ser presentado ante el resto de la clase, además de entregado por escrito. Con esta actividad, se persigue que el alumno, además de adquirir conocimientos sobre el tema, adquiera las capacidades necesarias para exponer en público un trabajo, y para abstraer las ideas generales más importantes, ya que el tiempo de exposición se reducía a 10-12 minutos para cada uno de los dos alumnos del grupo. Por otra parte, de esta forma también podíamos puntuar a cada uno de los dos integrantes del grupo por separado, en función de su capacidad de abstracción, resumen de ideas fundamentales, claridad en la exposición, dominio del tema y capacidad de contestar a las preguntas formuladas por el profesor y por los demás alumnos.

La plantilla para la realización del trabajo y el directorio de entrega del mismo están disponibles en la página Web de la asignatura.

Finalmente, se propone a los alumnos una última actividad no presencial, que realizan en grupo, y que consiste en el desarrollo de una aplicación en la metodología seleccionada. En este momento, los alumnos ya tendrán una versión crítica de las metodologías que se pueden usar, así como de los posibles campos de aplicación de la disciplina, por lo que consideramos que están capacitados para realizar un diseño de un sistema de este tipo.

4. Estructura práctica de la asignatura

En cuanto a las clases prácticas, éstas son presenciales y la asistencia es obligatoria. La organización de las prácticas está sincronizada con las clases de teoría, de forma que los alumnos programen aquellos aspectos de un Sistema Multiagente que ya han visto de forma teórica.

También pueden ser estructuradas en cuatro bloques básicos, aunque no guardan una relación directa con los bloques de teoría:

4.1. Introducción a JADE

En este primer bloque se introduce la herramienta de desarrollo de sistemas multiagente que se utilizará en la asignatura: JADE. Al estar realizada completamente en JAVA, los alumnos de segundo ciclo están altamente familiarizados con el lenguaje, y la comprensión de los elementos básicos es rápida. Además, esta herramienta cumple totalmente el estándar FIPA de creación de agentes, lo que la convierte en muy indicada para docencia.

Se dedican dos clases prácticas a la explicación de la herramienta, dónde se llevan a cabo ejemplos simples que toquen los aspectos básicos de JADE (creación y eliminación de agentes, configuración de la herramienta, entorno gráfico, threads, etc). Estas prácticas no serán entregadas al profesor, sino que se plantean para que el alumno pueda fijar dichos conceptos básicos.

4.2. Conceptos básicos de Sistemas Multiagente

En este bloque, el alumno deberá realizar una práctica de programación obligatoria consistente en la modificación de un ejemplo básico de sistema multiagente dedicado a la compra-venta de libros. Este problema fue planteado en una versión básica dentro de las dos primeras clases prácticas que se utilizaron para introducir los conceptos básicos de JADE y los lenguajes orientados a agentes. De esta forma, el alumno no parte de cero a la hora de implementar un sistema tan complejo, sino que debe completar y aumentar un código ya existente. La práctica hace hincapié en los principales aspectos de JADE, como son el uso de comportamientos, las comunicaciones, el uso de la GUI, etc.

De nuevo, se habilita un directorio de entrega en la página Web de la asignatura y se proporciona una plantilla para la realización de la memoria de la práctica.

4.3. Conceptos avanzados de Sistemas Multiagente

La segunda práctica obligatoria implica la realización por parte del alumno de un sistema multiagente visto en clase de teoría y que trata de realizar la exploración óptima de un planeta, con el objetivo de que un grupo de agentes robóticos recojan el mayor número de piedras posible. Para la realización de esta práctica, los alumnos deben organizarse previamente en grupos de 6 alumnos, es decir, grupos formados por 3 de las parejas ya existentes. Cada uno de estos grupos de 6 alumnos debía crear un simulador gráfico con el objetivo de poder realizar una competición entre los sistemas creados por cada pareja. Esta tarea inicial se planteó con el objetivo de fomentar el trabajo en grupo y, dado que los alumnos son de último curso, de mostrar qué implica organizarse en un grupo de programadores para realizar una aplicación conjunta (Ingeniería del Software). De hecho, el profesor de prácticas no proporcionó ninguna indicación sobre cómo debían organizarse los alumnos. El hecho de plantear una competición final hizo que los alumnos se tomasen en serio la premisa básica de realizar una implementación optimizada, con el objeto de no resultar "perdedores" delante del profesor y de sus compañeros.

Desde el punto de vista de los contenidos, esta práctica trata de hacer énfasis en aspectos más realistas de los sistemas multiagente, como son la necesidad de un simulador gráfico que permita analizar los resultados de forma simple, la necesidad de colaboración entre los agentes para lograr una meta común, la necesidad de comunicarse, etc.

De nuevo, se habilita un directorio de entrega en la página Web de la asignatura y se proporciona una plantilla para la realización de la memoria de la práctica

4.4. Práctica optativa

Todos los trabajos anteriores, a excepción del trabajo inicial sobre búsqueda de referencias, fueron realizados por los grupos (parejas) creados al inicio del curso. La idea de la práctica optativa fue que el alumno que desease incrementar su nota final, pudiese hacerlo de forma individualizada. Así, el profesor de prácticas plantea una serie de posibles implementaciones y los alumnos interesados escogen una de ellas. También, aquellos alumnos que lo deseen, pueden plantear al profesor alguna idea propia como práctica optativa. Una posibilidad es mejorar alguna de las dos prácticas obligatorias, de modo que no supone la realización de un sistema nuevo desde cero.

De nuevo, se habilita un directorio de entrega en la página Web de la asignatura y se proporciona una plantilla para la realización de la memoria de la práctica.

5. Evaluación

En cuanto a la evaluación, los alumnos pueden elegir entre dos vías:

- Evaluación continua: El alumno debía entregar los distintos trabajos tanto teóricos como prácticos a lo largo del curso. A continuación se muestra la valoración sobre la nota final de cada una de las actividades:

Actividad	Porcentaje sobre la nota final
Trabajo links agentes	2.5 %
Trabajo reactivo vs deliberativo	5 %
Trabajo metodologías	10%
Trabajo aplicaciones	15%
Diseño aplicación	15%
Practica 1	15%
Practica 2	22,5%
Practica 3	15%

El valor de cada actividad fue fijado por los profesores de la asignatura a principio de curso con el objetivo de que ambas partes (teoría y práctica) pesasen lo mismo sobre la nota final. Sin la realización de la práctica optativa, el alumno no puede optar a la calificación máxima.

Decir por último que las calificaciones acumuladas tras la entrega de cada trabajo eran publicadas en la Web de la asignatura de forma que los alumnos pudiesen ir viendo a lo largo del curso la calificación que acumulaban.

- Evaluación final: El alumno debía entregar únicamente las dos prácticas obligatorias y el examen se centra en la parte teórica. La ponderación de cada parte es al 50%.

6. Resultados

De cada una de las actividades propuestas, hemos extraído una serie de conclusiones de cara al próximo curso:

1. Búsqueda de referencias sobre aplicaciones existentes sobre sistemas multiagente: Los conceptos básicos que se presentaron en el tema introductorio y que podían parecer abstractos adquirieron tras este trabajo una dimensión real. Nos ha parecido una actividad introductoria muy adecuada porque permitió que los alumnos se familiarizasen con la entrega de trabajos a través de Moodle (hubo varias incidencias a este respecto), y que se hiciesen una idea de la dimensión real del tema que se trata.
2. Trabajo sobre arquitecturas deliberativas y reactivas: Permitió que los alumnos se hiciesen una idea de la problemática de las arquitecturas antes de que el profesor las explicase, lo cual es necesario dada la complejidad del tema. En este curso, la implantación de la metodología Jigsaw [4] persigue un entendimiento más profundo del tema, así como favorecer el aprendizaje colaborativo de los alumnos.
3. Entrega de la primera práctica obligatoria: Sirvió para que los alumnos se familiarizasen con el entorno JADE, y con la creación de sistemas altamente distribuidos. Además, aquellos grupos que fueron calificados con suspenso, se tomaron más en serio el trabajo a realizar en la segunda práctica y el de los siguientes trabajos teóricos.
4. Entrega de un trabajo sobre metodologías: Durante el curso pasado, se logró el objetivo buscado de mostrar la falta de estandarización a la hora de crear una metodología para sistemas multiagente. Se pedía a los alumnos que realizasen un estudio comparativo con otras metodologías, así como que resaltasen las ventajas y los inconvenientes de la metodología principal a estudiar. De esta forma, no sólo debían conocer una metodología particular, sino también relacionarla con otras. En este curso, la metodología que se pretende usar busca de nuevo un aprendizaje autónomo del alumno mayor, junto con fomentar el aprendizaje colaborativo y el espíritu crítico.
5. Entrega y exposición del trabajo sobre aplicaciones de agentes: Esta actividad sirvió para que los alumnos adquiriesen una dimensión realista de las aplicaciones que se han llevado a cabo en este campo, ya que cada

grupo debía exponer durante 10-12 minutos en clase delante de sus compañeros. El profesor trató además de fomentar la participación y la discusión tras las exposiciones, iniciativa que resultó muy satisfactoria. Este aspecto favorece las capacidades de crítica y de síntesis, así como la adquisición de competencias para comunicarse de modo eficaz y correcto con los demás, aprendiendo a argumentar con claridad, hablando en público. Además, debe elaborar documentación audiovisual para la presentación de su trabajo, así como redactar un informe. Todas estas actividades serán importantes para enfrentarse a otras asignaturas, fundamentalmente el Proyecto Fin de Carrera, así como a su futuro profesional. Los demás alumnos deben organizar un debate, argumentando con el grupo que está exponiendo su trabajo, sobre determinados aspectos del mismo.

6. Entrega del diseño de un sistema multiagente sobre una aplicación elegida por los alumnos. La estrategia utilizada es el aprendizaje orientado a proyectos [5], de manera que en el aprendizaje autónomo del alumno cobra especial importancia el proceso investigador y la realización de un trabajo escrito.
7. Entrega de la segunda práctica obligatoria: La creación de "grupos de grupos" que se debían de organizar de cara a realizar el simulador resultó muy satisfactoria y todos los simuladores desarrollados funcionaban perfectamente y permitieron la realización de competiciones. El hecho de plantear dichas competiciones hizo que los alumnos se tomasen en serio la premisa básica de realizar una implementación optimizada, con objeto de no resultar "perdedores" delante del profesor y de sus compañeros. Por otro lado, esta práctica sirvió para que los alumnos se diesen cuenta de qué implica crear un sistema "inteligente" sin la posibilidad de usar memorias, es decir, las limitaciones de los sistemas puramente reactivos.
8. Entrega de la práctica optativa: Esta actividad ha servido para mostrar, por un lado, que los alumnos llegan al final del curso con demasiados exámenes y trabajos y parte de ellos no son capaces de realizar la entrega a tiempo. Los mejor organizados o los más interesados en una buena calificación final han realizado implementaciones muy interesantes.

7. Conclusión

En cuanto a las conclusiones finales que podemos extraer tras nuestra experiencia en este curso cabe destacar:

1. El nº de alumnos de la asignatura (26 en el curso 2005/06), ha permitido la realización de las actividades mencionadas anteriormente. Los profesores de la asignatura podían plantear trabajos colaborativos, y corregirlos rápidamente, informando a los alumnos de sus notas acumuladas a lo largo del cuatrimestre. De esta forma el alumno también podía medir el esfuerzo que necesitaba realizar en posteriores trabajos para lograr una determinada nota acumulada. En el caso de que el nº de alumnos aumentase de forma considerable en el futuro, sería mucho más difícil el poder seguir este tipo de enseñanza.
2. El porcentaje de aprobados en la asignatura es superior al 95%, con unos resultados a nivel de conocimientos por parte de alumnado muy satisfactorios.
3. Todos los alumnos han elegido la forma de evaluación continua y han participado de forma activa en el aprendizaje de la asignatura durante todo el cuatrimestre.

Referencias

[1] http://www.udc.es/eees/docs_eees/doc_gaddu/gaddu_proyecto02_gal_final.pdf
[2] http://moodle.org/
[3] http://www.fv.udc.es/
[4] M. de Miguel Díaz. "Metodologias de enseñanza y aprendizaje para el desarrollo de competencias". Alianza Editorial, 2006.
[5] J.W. Thomas. "A Review of research on Project-based learning". The Autodesk Foundation, 2000

Experiencia en la adaptación de la asignatura Sistemas Operativos I al EEES

José Antonio Gómez, Buenaventura Clares
Dpto. de Lenguajes y Sistemas Informáticos
Universidad de Granada
Periodista Daniel Saucedo Aranda s/n, 18071Granada
{jagomez, bclares}@ugr.es

Resumen

En el presente artículo se exponen los cambios metodológicos realizados en la asignatura de Sistemas Operativos I encaminados tanto a una próxima adaptación al EEES como una mejora de la actual docencia. Con la mejora de la docencia se pretende generar un conocimiento más sólido de la disciplina en nuestros estudiantes, mejorar los resultados finales de los estudiantes, y reducir la falta de asistencia a clase. Los cambios metodológicos usados pretenden trasladar la responsabilidad del aprendizaje al estudiante. Para ello se fomenta el trabajo autónomo, se aplican técnicas de aprendizaje cooperativo y aprendizaje basado en problemas para el desarrollo de las clases presenciales, y se plantea una revisión de la evaluación. Si bien no se utilizan metodologías novedosas, hay que destacar que si lo es el empleo conjunto de éstas, y su articulación a través de las guías didácticas y de trabajo autónomo. Los resultados obtenidos muestran una mejora los resultados académicos de nuestros alumnos.

1. Motivación

La materia de sistemas operativos se imparte en la Universidad de Granada en tres asignaturas. Sistemas Operativos I y Sistemas Operativos II, que se imparten en el segundo curso de las tres titulaciones de Informática, en el primer y segundo cuatrimestre, respectivamente. Diseño de Sistemas Operativos que se imparte en quinto curso de Ingeniería Informática y es optativa.

La asignatura de Sistemas Operativo I, con un programa y método de trabajo similar a los que podemos encontrar en otras universidades, seguía, tras algunas reformas, enfrentándose a dos graves problemas. Uno, el alto porcentaje de no presentados en las diferentes convocatorias de examen. Otro, las dificultades para adquirir las capacidades adecuadas en la asignatura. Este último punto provoca además una dificultad añadida en el seguimiento de la asignatura de Sistemas Operativos II, que impartimos en el segundo cuatrimestre.

Para intentar evitar estos problemas, nos planteamos modificar la forma de impartir y evaluar la asignatura. Las modificaciones propuestas se acercan a la filosofía propuesta por el Espacio Europeo de Educación Superior, es decir, tratan de centrarse en el aprendizaje más que en el proceso de enseñanza.

En el siguiente apartado, mostraremos todos los cambios realizados y justificaremos su uso. En el apartado 3, analizaremos tanto los resultados de las encuestas pasadas a los estudiantes como los obtenidos de las calificaciones finales. Para finalizar, expondremos nuestras conclusiones y líneas de mejora para próximos cursos.

2. Trabajo desarrollado

Los intentos de mejora realizados con anterioridad en la asignatura estaban orientados hacia la enseñanza [1]. En ellos, los profesores tratan de mejorar la docencia mediante una preparación más cuidada de las clases magistrales, ajustar los contenidos para que estuviesen actualizados (evitando explicar cuestiones en desuso), preparar de material de apoyo para ilustrar los conceptos más problemáticos, etc.

Desgraciadamente, este esfuerzo no se veía suficientemente recompensado tanto en las calificaciones de los alumnos como en un decremento de la falta de asistencia a los exámenes y clases.

Por ello, decidimos afrontar una reforma en la forma de desarrollar la docencia de la asignatura con el objetivo de centrarnos más en el aprendizaje del alumno, y, por tanto, delegando la

responsabilidad de su aprendizaje en ellos mismos.

Las medidas tomadas comprendían las siguientes acciones:

- Emplear las clases magistrales de una manera reducida y en temas concretos.
- Fomentar el trabajo en grupo fuera y dentro del aula. Esto genera unas clases presenciales más participativas y dinámicas.
- Desarrollar un tipo de docencia similar al aprendizaje basado en problemas en aquellos casos en que sea posible.
- Elaborar una guía didáctica y otra de trabajo autónomo que facilite la labor del alumno guiando su proceso de aprendizaje.
- Modificar el sistema de evaluación para que ésta sea una medida real de valoración del aprendizaje y no un mero sistema de clasificación de estudiantes.

En los siguientes epígrafes vamos a detallar cada uno de estos puntos y a ejemplificarlo en el contexto de nuestra asignatura de Sistemas Operativos I.

2.1. La clase magistral

Dado que queríamos dar más importancia al trabajo del alumno, decidimos reducir el número de clases magistrales. En general, su uso esta encaminado a la transmisión de información, y por ello pensamos que parte de la adquisición de esa información se podría realizar de igual forma mediante la lectura de una de las referencias clásicas de la disciplina. En nuestro caso, optamos por el libro de Stallings [2]. Podría haber sido cualquiera, pero lo vimos más adecuado por dedicar una mayor atención a aspectos de interacción entre los sistemas operativos y el hardware que otros libros. La elección de un único texto de referencia permite unificar las lecturas de los estudiantes, y no excluye que se aporte nuevo material procedente de otras referencias, como veremos al hablar de las guías.

Las clases magistrales impartidas tienen ahora básicamente dos objetivos. Primero, se usan a comienzo de un tema, y están enfocadas a dar una visión global y conductora del tema (visión que a veces se pierde en las lecturas concretas de la bibliografía). Segundo, conocidos los principales escollos de la asignatura, la clase magistral pretende avisar a los alumnos de los problemas concretos que se pueden en encontrar y su correcta interpretación.

Como ejemplo del primer uso de la clase magistral, centrémonos en la primera clase del tema de Gestión de Memoria, típico de cualquier asignatura de fundamentos de sistemas operativos. En ella, presentamos una visión general con la línea argumental que pasamos a exponer.

La gestión de memoria tiene tres ejes fundamentales: el primero, relacionado con el hardware, y que si bien nosotros veremos algunos aspectos, se tratará con mayor detalle en otras asignatura (en nuestro caso en Estructura de Computadores II, que se imparte en el segundo cuatrimestre); el segundo, la generación de código por parte del compilador, que se estudiara con mayor detalle en asignaturas de procesadores de lenguajes; por último, el sistema operativo, que interacciona con los otros dos elementos para llevar a cabo una gestión eficiente de la memoria.

A continuación se expone algún esquema simple de gestión de memoria, con el objetivo de establecer el principal concepto relacionado con el tema, *la fragmentación*. Se explica que éste es el fenómeno que desencadenó la generación de nuevos esquemas de gestión de memoria más complejos debido a que producía un gran desperdicio de memoria. Surge así la segmentación, que, con la idea de trocear un programa en fragmentos, se pensaba que reduciría la fragmentación de forma notable. La segmentación presenta el problema de ser un mecanismo de asignación dinámica de memoria, por lo que penaliza la ejecución de los programas y sigue produciendo fragmentación. A partir de este problema, se ideó dividir el proceso en fragmentos iguales a los huecos de memoria para eliminar el problema de la asignación dinámica y el de fragmentación externa, surge así la *paginación*.

En los sistemas actuales y condicionados por el hardware, se utilizan estos dos últimos esquemas, que si bien son en cierta medida redundantes, tienen encomendadas diferentes funciones: la segmentación gestiona los diferentes niveles de privilegio, la paginación gestiona las necesidades de memoria de los procesos.

Reduciendo el número de clases magistrales, se nos planteaba el problema de cómo transmitir al estudiante la información necesaria. Optamos por la solución de encomendar parte de este

trabajo al alumno, que realiza lecturas seleccionadas del libro de texto como parte de su trabajo en casa. Recordaremos que la asignatura tiene 5.33 ECTS, lo que supone unas tres horas semanales de trabajo no presencial. Esto tiene como ventaja, frente a otros esquemas, que permite estimar el tiempo de trabajo del estudiante en casa de una manera fácil.

2.2. El trabajo en grupo

La reducción del número clases magistrales permite dedicar una parte importante de las clases presenciales a un trabajo activo del alumno, frente al pasivo que fomenta la clase magistral. Para el trabajo en clase, optamos principalmente por fomentar el trabajo en grupo. Esto obedece básicamente a dos razones: primero, el trabajo en grupo en el aula permite al profesor supervisar el proceso de aprendizaje y, por tanto, intervenir de una forma más eficiente en su marcha; segundo, resuelve el problema frecuente que se deriva de la dificultad de poner en contacto a los componentes del grupo fuera del aula, debido en muchas ocasiones a la incompatibilidad horaria.

En nuestro departamento, un grupo de profesores venimos usando diferentes técnicas de aprendizaje cooperativo desde que en 2005 se impartiera un taller al respecto [3].

El aprendizaje cooperativo en nuestra asignatura consume un porcentaje importante de clases presenciales. Desde el primer día de clase, se forman grupos de tres alumnos, cuyo objetivo es abordar durante el cuatrimestre diferentes trabajos en grupo, tanto fuera como dentro del aula. Como después veremos al analizar los resultados, creemos que esta forma de trabajo influye positivamente a la hora de que los alumnos no vayan dejando la asignatura a lo largo del cuatrimestre.

Los trabajos propuestos para resolver en grupo están encaminados a solventar las dudas de las lecturas previas, y plantear y resolver problemas que posiblemente el estudiante no percibió cuando realizó el estudio. En definitiva tiene por objetivo generar un conocimiento más profundo sobre la materia.

Para ilustrarlo en el contexto de la disciplina, y siguiendo con el ejemplo del tema de memoria, una vez leído en el libro el apartado sobre paginación, se plantea a los grupos un ejercicio en el que se les propone que construyan la tabla de páginas para un proceso cuyo espacio de direcciones (divido en código, datos y pila con tamaños determinados) se da como dato. Este ejercicio permite que entiendan como funciona la paginación y pone de manifiesto aspectos que leyeron pero que no saben como manejar en la práctica. Por ejemplo, las dudas que les surgen en este ejemplo proceden de cómo reflejar la presencia de *heap* en la tabla de páginas (en general, suelen no reservar entradas para él) o qué hacer con las páginas compartidas de código y datos (que suelen separarlas). Además, éste ejercicio sirve de ilustración para que posteriormente entiendan por que es necesario, como vemos en Sistemas Operativos II, superponer sobre la paginación otros esquemas de gestión de memoria, como las *vm-áreas* de Linux, que permiten una representación cómoda de espacios de direcciones dispersos.

Estos ejercicios de clase permiten al profesor guiar el aprendizaje, pues cuando se detecta que algún concepto no se ha entendido bien se pueden plantear nuevos ejercicios en la misma línea pero desde otras perspectivas para que lo asimilen.

Además de los trabajos en grupo de clase, también se asignan algunos trabajos fuera del aula. En concreto, este año se ha realizado uno sobre los algoritmos de planificación de la CPU y otro en algoritmos de sustitución de páginas. Estos trabajos tienen como interés, además de que adquieran los conocimiento propio de los temas, fomentar la capacidades relativas a la expresión escrita y oral, pues se entregan sendos documentos y se exponen en clase.

2.3. Aprendizaje basado en problemas

Los ejercicios, o problemas, que se proponen en clase suelen mostrar en general una parcela pequeña del tema de estudio que nos ocupa, y, por tanto, no suelen dar una visión de funcionamiento global del sistema.

Para solventar este inconveniente, hemos desarrollado un ejercicio que se resuelve poco a poco, pero que da una visión conjunta de varios temas de la asignatura, integrando contenidos que de otra forma parecen confinados en temas independientes.

Para su desarrollo, partimos de los conocimientos previos de los alumnos en una asignatura de introducción a la informática de

primer curso. En ella, se estudia el funcionamiento de un sistema de computador incluidos los mecanismos básicos de entradas/salidas: sondeo, interrupción, y mediante DMA. A partir de aquí, desarrollamos en problema en partes estructuradas según los temas de la asignatura.

En el Tema 1 de la asignatura se hace una introducción a la funcionalidad del sistema operativo y cómo esta se estructura en componentes. En especial hablamos de las rutinas de entrada/salida o manejadores de dispositivo. En un ejercicio se pide al grupo que traslade el diagrama de entradas/salidas controladas por interrupción visto en primer curso a seudo-código de forma que separen las diferentes actividades del diagrama de flujo y las asignen a los módulos del sistema operativo que las realiza. Este ejemplo permite reflexionar sobre el tratamiento de las interrupciones por parte del sistema operativo, y comprender las bases de la multiprogramación al visualizar que se debe ceder el control a otro proceso una vez instruido el dispositivo para realizar una operación.

En el segundo tema, dedicado a la implementación de procesos, se retoma el mísmo ejemplo. Este permite de forma natural ver la necesidad de introducir el concepto de estado de un proceso y esto a su vez deriva en transiciones entre estados, en especial del estado bloqueado. La Figura 1 recoge un instante de construcción de este macro ejemplo. También, nos permite ver como se implementa tiempo compartido a través de las oportunas modificaciones de la rutina de servicio de interrupción.

De nuevo, en el Tema 3, podemos retomar el ejemplo, para introducir y resolver un problema de productor-consumidor centrado en una entrada de teclado.

Como podemos observar, este gran ejercicio nos permite, mejor dicho les permite a ellos, ir creando un *sistema operativo de papel* que explica y justifica la aparición de muchos conceptos que aparecen en sistemas operativos.

2.4. Guía didáctica y de trabajo autónomo

Como se recoge en [4], “el desplazamiento inicial de interés por la enseñanza hacia el aprendizaje, que aparece como uno de los postulados del EEES, obliga al profesor a planificar y desarrollar la enseñanza de un modo diferente. No es suficiente con facilitarle al alumno el material que tiene que estudiar sino que hay que indicarle cómo deber abordar el proceso de aprendizaje”.

Desde este planteamiento, decidimos realizar una guía didáctica y otra de trabajo autónomo para la asignatura. Debemos de indicar que es precisamente esta última, la de trabajo autónomo, la que nos ha consumido el mayor esfuerzo, ya que la guía didáctica contiene información sobre la asignatura que ya estaba disponible de forma dispersa: objetivos, temario, tutorías, forma de evaluación, etc.

En nuestro caso, la guía de trabajo autónomo contiene un programa de trabajo detallado al día que pretende guiar el aprendizaje de los alumnos. En ella se recogen tanto la dedicación temporal de cada tema, como las actividades diarias a realizar: qué hay que leer, que ejercicios o cuestiones se van a tratar en clase, cuales son los puntos importantes, etc.

Estas guías han mostrado ser muy útiles, y así lo han valorado los alumnos, por los siguientes motivos:

- Reducen la incertidumbre del alumno ante la docencia, ya que sabe qué va a hacer en cada momento. Además, si algún alumno falta un día, sabe qué trabajo se ha realizado y por tanto qué debe recuperar.
- Marcan el ritmo de trabajo, que en un modelo tradicional podría acelerarse o retrasarse por diversas causas. Esto no quiere decir que las guías deber ser inflexibles.
- Permiten alterar el hilo conductor, o enfoque, de determinados temas y añadir material adicional al libro de texto, al permitirnos

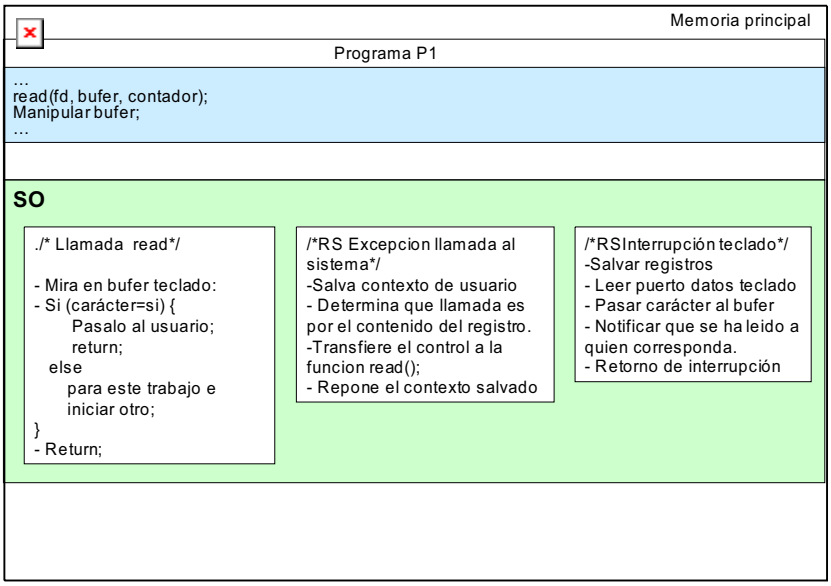


Figura 1. Una instantánea de la evolución del sistema operativo de papel

reordenar las lecturas o intercalar nuevos documentos.

- Permiten que el alumno se centre más en su aprendizaje que en la toma de apuntes.

En el siguiente apartado se mostrará cómo han valorado los alumnos las primeras guías que hemos realizado este curso.

2.5. Evaluación del aprendizaje

Todos estos cambios nos obligaban a replantearnos la evaluación del aprendizaje, que debería estar más encaminada a fomentar el conocimiento profundo de la materia que a ser un sistema de clasificación de alumnos.

Dada nuestra falta de experiencia en la aplicación de métodos de evaluación continua, no nos atrevimos a dejar el peso de la evaluación exclusivamente en la valoración del trabajo del alumno, y se opto por plantear un porcentaje del 60% para la realización de pruebas individuales, frente al 30% dado al trabajo en grupo. Un 10% de la calificación se deja a potestad del profesor, el cual lo asigna en función de la asistencia y participación en clase, la asistencia a tutorías, etc., es decir, tiene cómo principal objetivo fomentar la participación.

La evaluación realizada tiene las características de ser formativa y acumulativa.

Al hablar de evaluación formativa me refiero a que el proceso de evaluación tiene como principal objetivo ver el proceso de aprendizaje del alumno y, a la vista de los resultados, tomar las acciones oportunas. Para ello, comenzamos el cuatrimestre realizando ejercicios en grupo, muchos de estos ejercicios se corrigen en clase, y puntualmente es el profesor quién corrige alguno de ellos. Esta corrección permite conocer la marcha del grupo, tanto al profesor como a los miembros del mismo, pues la corrección se realiza en un plazo breve de tiempo y se devuelven los ejercicios a los grupos con las anotaciones pertinentes.

Cuando está finalizado el tema, se procede a realizar una prueba individual similar a las pruebas realizadas en grupo. El objetivo de la misma es conocer la marcha de cada uno de los alumnos de forma individual. Al igual que en las pruebas en grupo, esta se corrige rápidamente y se devuelve debidamente anotada al alumno, lo que permite el estudiante corrija lo más rápidamente posible las deficiencias de aprendizaje observadas.

La evaluación es acumulativa por diferentes motivos. Uno de ellos proviene del deseo de

fomentar en ellos la capacidad de razonamiento y resolución de problemas, por lo que creemos que no podemos aislar los temas unos de otros, sino ver la materia como un todo. Otro proviene de la propia operatoria a la que están acostumbrados de la docencia tradicional, me refiero al examen parcial. No deseamos que vean las pruebas/ejercicios individuales como exámenes parciales en los que se da una materia, se estudia y después se desecha.

3. Resultados obtenidos

Con el objetivo de poder evaluar la docencia de la asignatura en este curso, preparamos dos encuestas para los alumnos, que fueron pasadas al final del cuatrimestre a los alumnos presentes en clase. Una encuesta destinada a evaluar la asignatura y al profesor; la otra, destinada a valorar las guías y la metodologías empleadas.

Respecto a los resultados de la encuesta sobre el profesor y la asignatura, la calificación del profesorado de la asignatura es superior a la obtenida en las encuestas de satisfacción del profesorado que en cursos anteriores realizó nuestra universidad. La calificación a la pregunta "ha aumentado mi interés por los estudios de informática" ha sido de de 3 sobre 4 (que corresponde a un "bastante"). Respecto a "si han llegado a dominar los principios de la asignatura" la puntuación media es de 3.3 sobre 4. Y dan una puntuación de 3 a la pregunta "he mejorado mi capacidad para extraer conclusiones".

Teniendo en cuenta que la calificación media del item peor puntuado es de 2.3, hay que tener presente que quedan algunas cosas por mejorar, como son: aumentar su curiosidad por investigar y descubrir, lograr mayor confianza en sus capacidades, y un aumento de su capacidad y actitud crítica.

En cuanto a la evaluación de las guías y la metodología, hay que resaltar que merecen una calificación media de 7.3 y 8.2 sobre 10, respectivamente. También, es de resaltar que un 91% de los estudiantes opinan que esta metodología hace al alumno más responsable de su aprendizaje y es más adecuada al nivel universitario.

Otro punto a destacar es que el 77% contesta que exige más trabajo del alumno. Esto es obvio pues, frente al mero esfuerzo de asistir a clase en un método tradicional, a nuestros alumnos les exigimos una participación muy activa tanto en el aula como fuera.

Volviendo al planteamiento inicial, que exponíamos en la Introducción y que nos condujo al cambio de concebir nuestra docencia, indicar que el absentismo de clase ha bajado del 50 al 31%, que aproximadamente el 70% de los alumnos que comenzaron asistiendo a clase finalizaron el cuatrimestre, y que han superado con éxito la asignatura un 82% de los la han seguido (frente a un escaso 50% de los años en que se ha seguido un enfoque tradicional) .

Para concluir, en la Tabla 1 se muestran las calificaciones de los tres últimos cursos de la asignatura.

Cursos	2004-05	2005-06	2006-07
Por calificación:	%	%	%
Suspensos	31,46	15,63	13,95
Aprobados	4,49	21,88	23,26
Notables	3,37	9,38	30,23
Sobresalientes	3,37	1,56	4,65
Matrícula H.	1,12	1,56	0
No presentados	56,18	50	27,91
Por resultados:			
No superados	31,46	15,63	13,95
Superados	12,36	34,38	58,14
No presentados	56,18	50	27,91

Tabla 1.- Calificaciones de los últimos tres cursos.

Como se puede observar, hay una reducción de alumnos no presentados. Respecto a los presentados, se observa claramente una mejora de las calificaciones. Si observamos existe una mejora en el curso 2005-06 respecto al curso 2004-05. Esta mejora es justificable debido a que ya en el curso 2005-06 comenzamos a aplicar algunos de los métodos citados en los apartados anteriores.

4. Conclusiones y trabajos futuros

Desde nuestro punto de vista, y tras analizar los resultados expuestos en el apartado anterior, creemos que estamos en un buen camino dado que los resultados han mejorado notablemente respecto a cursos anteriores.

No obstante quedan elementos que hay que mejorar. Por ejemplo, recuperar parte de ese 27% que se matricula en la asignatura pero no la cursa, o fomentar un mayor uso de la tutorías para

resolver las dificultades de aprendizaje detectadas durante el desarrollo de la asignatura.

El grupo de clase con el que hemos trabajado este curso tenía 45 matriculados, de los cuales sólo 33 cursaron la asignatura. Este pequeño número nos a permitido dedicarles más atención a los estudiantes y a la asignatura, pero entendemos que el método empleado es aplicable a grupos mayores con pequeños ajustes. Esto se debe a que el principal esfuerzo del profesorado ha estado centrado en la elaboración de las guías y la metodología, pero este trabajo es amortizable para cursos posteriores.

Respecto a la evaluación, que podría parecer que consume mucho tiempo, indicar que los trabajos en grupo disminuyen el número de ejercicios a corregir. En nuestro caso, al formar grupos de 3 alumnos, solo necesitamos corregir, un tercio de ejercicios. Creemos que es posible formar grupos algo mayores en clases donde sea necesario, de incluso 5 alumnos, pero el principal inconveniente con la estructura de las aulas actuales es que un grupo de este tamaño no trabajaría de una forma cómoda.

Respecto a los ejercicios individuales, al ser preguntas muy concretas se pueden corregir para este número de alumnos en dos o tres horas por prueba. Es decir, todas las correcciones se pueden realizar en parte del horario oficial de tutorías. Lo que es más importante, el trabajo global de las pruebas individuales es el mismo que el de corregir un examen final con 10 preguntas.

Para el próximo curso pretendemos seguir mejorando las guías. Además creemos conveniente, dado el gran número de ejercicios, trabajos y pruebas que realizan, poner en marcha una metodología tipo *portafolio* en el que los alumnos guarden todo ese material para ver su evolución y aplicar cómo usarlo desde el punto de vista de la evaluación del aprendizaje.

De forma más inmediata, se pretende agrupar el material desarrollado, en especial la guía didácticas y la guía de trabajo autónomo, y hacerlo accesible para todos los interesados en el tema en la páginas web de la asignatura con dirección http://lsi.ugr.es/~jagomez/sisopi.html.

5. Referencias

[1] Gómez, J. A. *La asignatura Sistemas Operativos I en el 2mil2,* Actas de las VIII Jornadas de Enseñanza Universitaria de la Informática, Jenui 2002, páginas 533-540, Cáceres, Julio 2002.

[2] W. Stallings, *Sistemas Operativos : Aspectos internos y principios de diseño (5ª de)*, Prentice Hall, 2005.

[3] J. Bará, J. Domingo, M. Valero, *Taller de formación: Técnicas de Aprendizaje cooperativo*, Universitat Politècnica de Catalunya, 24-25 Nov. Granada, 2005.

[4] S. Camacho, *Planificación de la docencia universitaria: Las Guías didácticas*, curso organizado por el Vicerrectorado de Planificación , Calidad y Evaluación de la Universidad de Granada en 2005, en www.ugr.es/~vic_plan/formacion/ceguido/ceguido1/Documenta/PDU_GD1_Guia.pdf

Análisis de Herramientas y Estrategias para las Nuevas Titulaciones en Informática

Antonio Polo Márquez, Jorge Martínez Gil, Luis J. Arévalo Rosado[1]
Dpto. de Ingeniería de Sistemas Informáticos y Telemáticos
Universidad de Extremadura
Avda. de la Universidad s/n, 10071 Cáceres
{polo, jmargil, ljarevalo}@unex.es

Resumen

Antes de aplicar cualquier herramienta o estrategia para el proceso de enseñanza-aprendizaje en cualquier titulación, pensamos que es imprescindible reflexionar sobre la situación actual. Por ello presentamos unas pautas de análisis que pretenden descubrir las fortalezas y debilidades de los sistemas utilizados en las titulaciones actuales. Este estudio permite definir el conjunto de requisitos de las herramientas que puedan ser útiles en el proceso de docencia y aprendizaje. Destacamos como puntos relevantes la necesidad de abordar las asignaturas en el ámbito de la titulación, permitir la personalización del sistema a cada alumno o profesor y potenciar el seguimiento y evaluación global del alumno. Ante los cambios hacia nuevas estrategias docentes con la aparición de los créditos ECTS, planteamos la necesidad urgente de potenciar el enfoque de asignaturas hacia créditos presenciales, frente a la idea extendida de potenciar créditos no presenciales. Aunque las reflexiones se han realizado en base a la titulación de Ingeniería Informática en la Escuela Politécnica de Cáceres de la Universidad de Extremadura (UEX), pensamos que muchas de las conclusiones son extrapolables a cualquier Ingeniería en nuestro entorno universitario.

1. Motivación

Aunque son numerosas las herramientas informáticas para docencia que se proponen en el mercado, en la mayoría de los casos sus adaptaciones a los distintos niveles de estudios no consiguen crear espacios en los que tanto el profesor como el alumno se sientan plenamente integrados. En nuestro país esta situación se agudiza en el entorno universitario. ¿Qué problemas plantean los sistemas actuales? ¿Cómo pueden resolverse? En este trabajo abordamos estas preguntas planteadas desde una perspectiva de Ingeniería de Software. Para ello aplicamos una metodología de análisis mediante la cual se detectan las fortalezas y debilidades de los sistemas de información actuales y se propone un conjunto de requisitos para su solución.

Nuestro enfoque está guiado por la idea de Titulación, considerando que las herramientas deben estar al servicio de la formación individual e integral de cada alumno. El problema no sólo radica en las herramientas que se encuentran en el mercado, sino en la propia organización universitaria y en los hábitos de funcionamiento de los trabajadores de dicha empresa. De entrada, la organización docente universitaria está basada en Departamentos y Áreas de Conocimiento, siendo muy débil la coordinación a nivel de titulación.

Una nueva perspectiva parece abrirse con la implantación de nuevos Planes de Estudio, impulsados por un espíritu de convergencia europea que se refleja en la implantación de los créditos ECTS (European Credit Transfer System) [3]. Sin embargo, creemos que la formación y disposición actual, tanto del profesorado como del alumnado, deben hacernos ser cautos antes de pensar que es el momento propicio para un cambio drástico en las estructuras.

En este trabajo se presentan los resultados del análisis realizado dentro del Proyecto Fin de Carrera SEDA (Sistema de EDocencia y Aprendizaje) [1] mostrando los problemas detectados en la titulación de Ingeniería Informática en la Escuela Politécnica de la UEX,

[1] Este trabajo ha sido financiado por el MEC a través del proyecto "Perspectivas de ICARO": TIN2005-09098-C05-05 y TIN2005-25882-E.

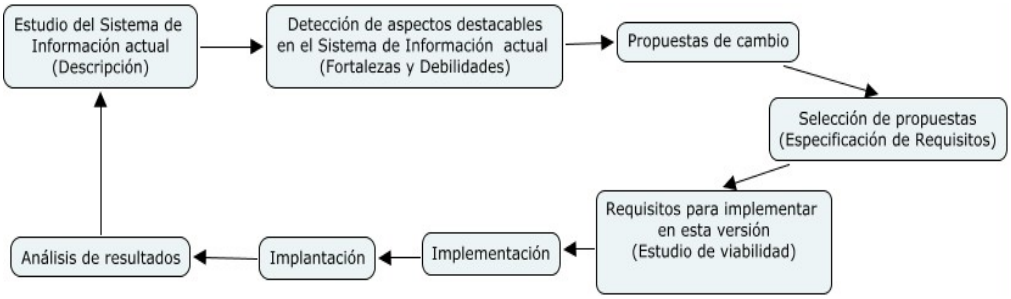


Figura 1. Metodología general para desarrollo de proyectos

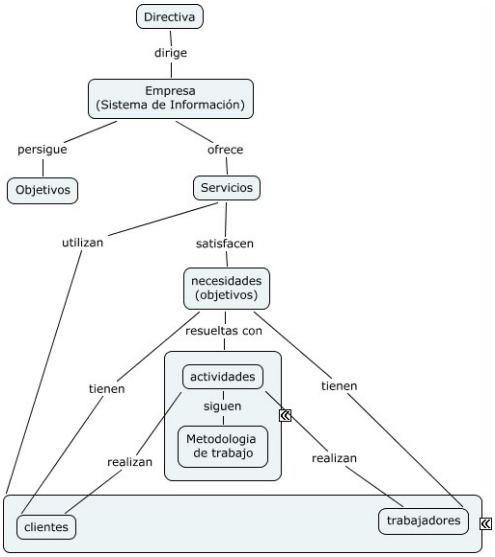


Figura 2. Modelo general con distinción de clientes y trabajadores en la empresa

que se pueden generalizar a la mayoría de las titulaciones similares en nuestro entorno universitario, y las herramientas y alternativas de solución que se han propuesto.

Este trabajo se inicia exponiendo la metodología llevada a cabo en el proceso de análisis definiendo un modelo general de funcionamiento de tipo empresarial en el que se destacan la disparidad de objetivos para cada uno de los usuarios del sistema. A continuación se indican las principales actividades que definen las necesidades de los usuarios, detectando los problemas asociados en cada una de ellas. En la sección 3 se completa este análisis considerando las nuevas directivas europeas para la implantación de las nuevas titulaciones. En la siguiente sección se clasifican las fortalezas y debilidades del sistema actual y se proponen algunas sugerencias de solución, distinguiéndose soluciones de tipo estructural u organizativo, junto a soluciones que pueden ser aportadas por sistemas de información. Finalmente, se resumen las soluciones organizativas y la especificación de requisitos que deberían cumplir los sistemas de información que se implanten; y se proponen algunas sugerencias para abordar el futuro, incierto como ya es tradicional, de la docencia universitaria.

2. Modelo de sistema de información empresarial en el ámbito universitario

Para el desarrollo de un Sistema de Información podemos seguir una metodología de desarrollo iterativo como la reflejada en la Figura 1. En ella destacamos la fase inicial de análisis partiendo de un modelo que describa el Sistema de Información actual.

Para el caso del ámbito universitario hemos decidido tomar un modelo empresarial, como el que se muestra en la Figura 2. En dicho modelo aparecen tres perfiles de usuarios: Directiva, Clientes y Trabajadores.

La Directiva se encarga de definir los perfiles de usuario y objetivos, y en definitiva, es el usuario responsable de la política e interpretación de funcionamiento del Sistema.

Este modelo general se debe aplicar al entorno concreto de la Titulación en nuestra Universidad.

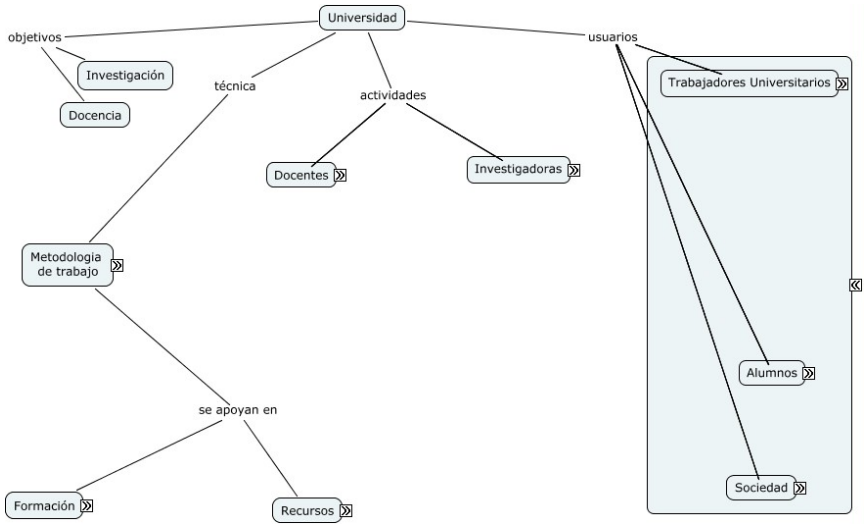


Figura 3. Esquema general del Sistema de Información en la empresa Universitaria

La interpretación de este modelo se ha realizado de forma muy genérica, por lo que se puede identificar con la situación habitual en cualquiera de nuestros centros, como se representa en la Figura 3.

Este esquema se puede expandir hasta llegar a situaciones de un mayor detalle como el que ofrece la Figura 4.

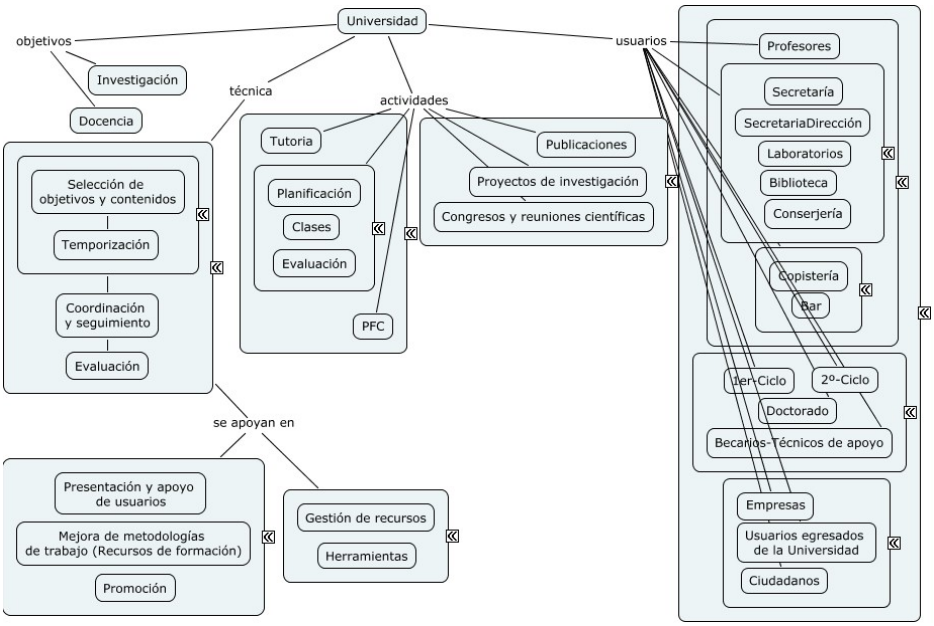


Figura 4. Esquema detallado del Sistema de Información en la empresa Universitaria

En esta aplicación del modelo aparecen los siguientes elementos:

Objetivos: a) *Docentes* que consisten en formar nuevos profesionales, y en cuyo análisis nos centraremos en este artículo, y b) *Investigadores* orientados al desarrollo de nuevas tecnologías y herramientas que resuelvan problemas para la sociedad.

Perfiles de usuario: Donde distinguimos:

- Directiva: Difícil de identificar en nuestro sistema.; pues la responsabilidad directiva aparece disgregada en distintos niveles y figuras como el Rector, Directores de Departamento y Decanos o Directores de Centro.
- Empleados: La Universidad, como empresa, genera diferentes puestos de trabajo como refleja la Figura 4. Todos deberán contemplarse en nuestro Sistema de Información. Sin embargo, desde el aspecto docente, nos centraremos en los *profesores* como los encargados de enseñar a los alumnos conocimientos y habilidades (destrezas) definidos a principio de curso. La mayoría de estos profesores también escriben artículos o se dedican a la investigación.
- Clientes: Los principales clientes de una Universidad son los *alumnos*. El alumno establece un contrato con la empresa universitaria en el proceso de matriculación, pagando por recibir una educación en una determinada especialidad, y en el que se compromete a realizar una serie de exámenes para demostrar que ha adquirido los conocimientos suficientes para merecer el título que lo acredite. Las empresas también pueden ser clientes de la Universidad participando en proyectos de investigación y desarrollo.

Una de las características esenciales en la empresa, que debe planificar la dirección de la misma, es la definición de una **metodología de trabajo**. Nosotros hemos propuesto una metodología de trabajo, general para todas las actividades desarrolladas por cualquiera de los perfiles de usuario, consistente en desarrollar cualquier actividad en los tres pasos siguientes:

- Planificación.
- Coordinación y seguimiento.
- Evaluación.

Organización estructural de la empresa: La Universidad, desde el punto de vista docente, presenta dos organizaciones ortogonales: a) Una división en Centros donde se imparten las correspondientes titulaciones, y b) Una división en Departamentos y Áreas de Conocimiento, que agrupa a los profesores que imparten asignaturas afines e investigan en campos comunes.

Por tanto, una visión organizativa que se realice en base al concepto de Titulación, irá ligada al Centro en que se imparte, frente a la estructura de organización basada en Departamentos en que se estructura la Universidad. Esto produce un conflicto a nivel de equipos directivos, ya que la estructura *fija* de la Universidad son los Departamentos que permiten, como aspecto positivo, una mayor flexibilidad para movilidad docente entre titulaciones y movilidad investigadora entre proyectos y grupos de investigación.

2.1. Actividades y necesidades de usuarios

Para seguir aplicando nuestro modelo, es preciso determinar las actividades que debe facilitar cualquier Sistema de Información, y que estarán guiadas por las necesidades que se plantean los usuarios para conseguir sus objetivos. Si nos centramos en los perfiles de profesor y alumno, como actores principales del proceso docente, podemos destacar las actividades reflejadas en la Tabla 1. En dicha Tabla aparecen en cada línea una misma actividad o muy relacionada, pero apreciada desde el punto de vista del usuario.

Así, el profesor debe realizar una planificación previa de la asignatura que determinará: el programa y contenidos de la asignatura, horario semanal (*Horario* en la última fila de Gestión de Actividades) y asignación temporal de las actividades (*Planificación* de la asignatura en la primera fila de la Tabla y que muestra simplemente las actividades y su duración).

A partir del calendario lectivo (*Calendario* en la última fila de Gestión de Actividades) y el *Horario* se puede generar el *Calendario de la Asignatura*, que combinado con la *Planificación* permite obtener la ***Agenda de la asignatura*** (*Agenda* en la última fila de Gestión de Actividades). La *Agenda* indica por tanto las actividades de la asignatura en el Calendario lectivo.

Estos tres elementos: Horario-Calendario-Agenda (HCA) que prepara el profesor constituyen el elemento básico para las actividades de la asignatura, tanto de teoría como de prácticas y problemas. Así, el alumno realiza la matriculación en las asignaturas confeccionando previamente su HCA personal, que le permite evaluar el esfuerzo y disponibilidad temporal de la carga lectiva en que se matricula.

Además, la Gestión de Datos Personales permite acceder a la Ficha e Historial de cada alumno (Certificado Académico), o al Curriculum Vitae del profesor.

Sin embargo, el desarrollo de estas actividades suele presentar los problemas que se indican en las próximas secciones.

Profesor	Alumno
Planificación de asignaturas	Matriculación
Control de asistencia a prácticas	Registro de asistencia
Elaboración de material docente	Descarga de material docente
Recuperación de prácticas entregadas	Elaboración de prácticas y subida al servidor
Redacción y generación de noticias	Acceso a noticias
Calificación de prácticas	Recepción de calificaciones
Gestión de datos personales (Curriculum)	
Gestión de Actividades (Horario / Calendario / Agenda)	

Tabla 1. Actividades de profesores y alumnos

3. Problemas y esbozos de solución

3.1. Deficiencias del modelo universitario actual

Algunas de las debilidades del sistema universitario actual se pueden modelar, describiendo la Universidad como un sistema de tipo **EFE**, para indicar que se trata de un sistema que es **E**stático, **F**ragmentado y **E**xclusivo.

Entendemos que es un sistema **Estático**, ya que resulta muy difícil cambiar el contenido de una asignatura, no sólo a lo largo de un curso, sino a lo largo del ciclo de vida de la titulación.

Es **Fragmentado** porque la universidad está dividida en Áreas de Conocimiento, en asignaturas, etc., de modo que cada profesor tiene una visión individual de su asignatura y de sus alumnos. Entendemos que hay que ver la universidad de forma global, con el punto de mira en la formación final que va a obtener el alumno.

Y finalmente, los contenidos de las asignaturas son **Exclusivos** de cada una de ellas, cuando sería deseable que dichos contenidos pudieran ser compartidos por diversas asignaturas, de manera que aparecieran interrelacionadas las asignaturas y si una materia evoluciona, también pueda evolucionar el contenido compartido con el resto de asignaturas.

En la actualidad, podemos considerar que los Sistemas de Información utilizados por los perfiles de usuario descritos anteriormente están limitados por los problemas anteriores, por lo que no facilitan las actividades, ni proporcionan una metodología de trabajo adecuada. Por ello, la mayoría de estos usuarios no logran desempeñar sus tareas de forma eficiente, gastando mucho tiempo y esfuerzo en su realización. Muchas de las tareas son repetitivas, teniendo que hacer lo mismo una y otra vez, y en numerosas ocasiones duplicando datos al no poder compartirlos automáticamente.

- **Duplicidad de datos y tareas**

¿Cuántas veces a lo largo de la carrera un alumno debe entregar una ficha al profesor? ¿Qué ocurre si cambia alguno de los datos del alumno?

En cuanto al profesor, cada curso se encuentra con la rutina de tener que revisar todos los contenidos, cambiar las fechas, preparar las prácticas, asignar un calendario para las clases teóricas y prácticas, tener que llevar un control de alumnos que asisten a clase, control de los alumnos que han entregado las prácticas, evaluación, publicación de actas, etc.

- **Restricciones de acceso**

En numerosas situaciones los datos son pobremente compartidos, en ocasiones debido a reticencias de compartición de datos por restricciones de privacidad. Sin embargo, podemos tomar como premisa para resolver estas restricciones el siguiente principio:

Principio de privacidad de datos: Los usuarios del mismo perfil compartirán todos los datos que faciliten su tarea de docencia/aprendizaje.

Esto implica, por ejemplo, que el curriculum de cada alumno podrá ser accesible por todos los profesores que participan en su educación; o que

el material de cualquier asignatura puede ser consultado por cualquier profesor de la titulación. O también que los comentarios y anotaciones de alumnos sobre un material docente pueda ser compartido por todos los alumnos de la titulación.

3.2. Otros problemas detectados

Estos son algunos de los problemas que nos parecen más destacables, obtenidos de la lista de debilidades observadas en el Sistema de Información actual, y que deben ser incluidos en la lista de requisitos de un Sistema de Información y ayuda a la docencia para la Ingeniería Informática:

- *Mecanismos de comunicación profesor-alumno*. A) Sistemas de notificación: Por ejemplo para informar al alumno de los cambios en el temario o contenido de la asignatura, o de la suspensión o cambio de horarios. B) Mejora de la comunicación virtual. C) Gestión de listas de correo.
- *Intervención del alumno en la elaboración de contenidos de forma supervisada por el profesor.* El alumno debe intervenir directamente en el contenido (por ejemplo, proponiendo nuevos cambios en el contenido, si encuentra alguna errata, etc.).
- *Definición de responsables para supervisión y coordinación.* ¿Quién se encarga de mantener y coordinar los horarios? En numerosas ocasiones no hay una estructura de coordinación bien definida. Habría que definir desde las personas que intervienen en la *realización* de la titulación hasta la figura del delegado/subdelegado de curso, que actualmente tiene escasa utilidad real.
- *Potenciar la estructura organizativa de la Universidad centrada en la titulación;* definiendo para cada perfil la noción de contrato que explicita los derechos y obligaciones para cada uno de los perfiles, así como la noción de hoja de reclamaciones que permita asegurar la realización del contrato por ambas partes.
- *Mecanismos de acceso y comunicación con los órganos de gobierno de la Universidad.*
- *Dotar a cada perfil de usuario de una metodología de trabajo***.** Es necesario optimizar la forma de trabajo buscando puntos de trabajo en común, interrelación directa entre asignaturas, evitar repeticiones innecesarias del temario, y facilitar la evolución coherente de las asignaturas.
- *Mecanismos de autentificación de prácticas,* pues se detecta una alta tasa de prácticas copiadas.
- *Mecanismo de control de impartición de clases,* esencial para actualizar la agenda real de clases presenciales.
- *Disponibilidad de datos.* Es difícil disponer de bases de datos estadísticas, y en especial de datos históricos. Por ejemplo, los datos de encuestas de evaluación de las asignaturas al finalizar el curso académico no están disponibles, y mucho menos los objetivos que persiguen dichas encuestas.
- *Gestión de Proyectos Fin de Carrera.* No existe ninguna herramienta que gestione los Proyectos de Fin de Carrera, ni normas de desarrollo, ni criterios de evaluación, ni listados y bases de datos con los proyectos ya presentados, …
- *Mecanismos de publicidad e interconexión de trabajos de investigación.*
- *Información de servicios auxiliares de la Universidad específicos a nivel de titulación.*

4. Impacto de la Convergencia Europea y créditos ECTS

Una de las tareas que nos planteamos en la fase de análisis fue considerar el impacto del cambio de la actual universidad hacia el nuevo modelo propuesto por la Oficina de Convergencia Europea (OCE) [4]. Para preparar el proceso de transición hacia las nuevas titulaciones normalizadas a nivel europeo, nuestra universidad está utilizando cursos o titulaciones (denominados piloto) que lo experimentan de forma optativa [2]. Los profesores deben elaborar una ficha de la asignatura con el programa, horas presenciales y no presenciales dedicadas a cada parte de ese programa y los créditos ECTS [3] que irán en relación con el número de horas. Hasta ahora los profesores elaboran toda esta información a mano.

- **Problema HCA de planificación**

Siguiendo las propuestas de la OCE, surgieron nuevas tareas que aún no disponen de herramientas para su realización. Para el autocontrol del alumno y su seguimiento por parte del profesor, se necesitan la agenda del alumno y agenda del profesor, que no son más que la

planificación del curso académico para ambos según se explicó en la Sección 2.1. Por ahora, estas agendas se elaboran con una hoja de cálculo o, en su defecto, un editor de textos. El profesor realiza la planificación al inicio del curso, pero resulta muy costoso compartir esta información con el alumnado. Además, si por alguna circunstancia el programa cambia, no existe ninguna herramienta que facilite la replanificación. Más difícil aún resulta para el alumnado integrar todas las agendas de las asignaturas en las que se matricula en una única Agenda Personal.

- **¿Qué intenta medir la Agenda Personal?**

Creemos que las asignaturas y cursos piloto han tenido varios aspectos positivos como el sentar a una mesa a profesores y debatir sobre las asignaturas. Sin embargo, el objeto de medida se ha centrado en muchas ocasiones en aspectos de asignación o sincronización temporal que nos desvían del núcleo central de debate, por lo que deben recordarse las siguientes ideas:

(A) Contenidos antes que metodología. El cambio de metodología no define un nuevo Plan de Estudios. Es decir, la introducción de nuevas técnicas docentes son sólo eso, mejoras para que el alumno aprenda. Lo importante es lo que aprende el alumno, los contenidos y las habilidades que debe adquirir. Desgraciadamente abundan las discusiones para ajustar los créditos y horarios antes que lo que realmente se pretende conseguir en esas horas de clase.

(B) Diseño global y descendente de los contenidos. No tiene sentido hablar de contenidos en asignaturas de forma aislada sin considerar los objetivos globales de la titulación y el resto de asignaturas.

(C) Presencia antes que ausencia. En nuestra experiencia nos da la sensación de que los créditos presenciales empiezan a tener menos peso que los no presenciales. Es cierto que los créditos ECTS han introducido nuevos aspectos de reflexión para calcular mejor las cargas de trabajo que mandamos a nuestros alumnos. Sin embargo, hay varias consideraciones que debemos tener para evitar situaciones extremas.

La carga de trabajo para el aprendizaje es tan personal que es muy difícil de medir. Es cierto que podemos indicar una aproximación de esfuerzo, pero influyen muchos aspectos que no podemos considerar como para dar una medida fiable. En cambio es más fácil y claro indicar qué tipo de habilidades se esperan y qué tipo de problemas debe ser capaz de resolver el alumno.

El tiempo libre que dispone un alumno, en especial cuando no ha sido educado para trabajar de forma no presencial, no va a cambiar. En cambio, se tiende a trasladar gran parte de las actividades del aula al ámbito no presencial. Nuestra experiencia ha detectado que el volumen de carga no presencial ha aumentado (en muchos casos simplemente porque ahora se contabiliza y antes no), y se ha disminuido la carga presencial; pero el alumno no ha aumentado su dedicación fuera del aula de forma que, en ocasiones, la carga no presencial a la semana es más del doble de la que realmente dispone en su planificación. Por tanto, proponemos como táctica para la implantación de nuevas titulaciones *usar al máximo las horas lectivas y minimizar la carga no presencial,* de esa forma aprovecharemos mejor los créditos de nuestra asignatura con el tiempo *disponible* del alumno… al menos hasta que se alcance un equilibrio razonado con el resto de créditos de la titulación.

5. Principios para un Sistema de Información para Ingenierías en Informática

Después del análisis desarrollado en las Secciones anteriores, podemos proponer una serie de principios para desarrollar un Sistema de Información que resuelva las deficiencias detectadas. En ese sentido nos gustaría que fuera Dinámico, Integral y Compartido, lo que podríamos llamar sistema tipo **DIC**.

- **Dinámico** para que durante el transcurso del período lectivo se pueden detectar errores y realizar actualizaciones. Cualquier cambio suscitado debería poder proponerse al responsable para que pueda ser certificado y en su caso, reflejarse en las demás asignaturas.
- **Integral** porque la formación debe verse como un todo y no sólo a través de las asignaturas individuales. Se deben obtener datos estadísticos a nivel general dándonos una visión exacta del estado actual de la formación que se oferta, e invitando o asesorando por personal docente al alumnado a través de un itinerario guiado, pudiendo así aprovechar el alumno todos sus esfuerzos de

una forma más homogénea y didáctica, como propone el *Plan de Acción Tutorial de la Titulación* (PATT) [5].

- **Compartido**, porque todas las asignaturas deben tener una interrelación. De esa forma, el Sistema debería indicar si un concepto se ha tratado en otra asignatura, proporcionando enlaces adecuados. El alumno dispondrá de todo el material estudiado para poder revisarlo y cualquier cambio en una asignatura será coherente con el resto de asignaturas.

6. Conclusiones y trabajos futuros

En el análisis que se realiza en este trabajo se han detectado las siguientes necesidades que deben cubrir los sistemas de información en el ámbito universitario para titulaciones de Ingeniería Informática:

1. Sistemas de personalización para cada perfil de usuario con las siguientes utilidades:
   - Utilidad de planificación HCA: Horarios, Calendarios y Agendas.
   - Sistema de notificación.
2. Integración del sistema HCA en el proceso de matriculación.
3. Integración de los contenidos:
   - A nivel de titulación/curso
   - A nivel de curriculum personal

Para poder conseguir estas funcionalidades es necesario un cambio en las estructuras de organización docente universitaria que potencie:

- El nivel de titulación-curso-asignatura como el ámbito de planificación y coordinación de los estudios.
- El seguimiento y evaluación personalizada de cada alumno.

Algunas de las propuestas que aparecen en las directivas europeas de nivelación de estudios parecen impulsar estos cambios, por ejemplo mediante las figuras de coordinadores de titulación o planes de acciones tutoriales. Sin embargo debemos ser prudentes ante cambios cualitativos, como la introducción de créditos no presenciales que afectan al comportamiento tanto del profesor como el alumno.

En todos los casos, tanto el hábito adquirido como la inmovilidad administrativa son los principales muros con los que se debe luchar para lograr estas mejoras en el ámbito educativo de la empresa universitaria.

Una característica esencial en este Proyecto es el enfoque Dinámico, Integral y Compartido de la educación. Por ello se ha abordado la adquisición de conocimientos desde la perspectiva del alumno como referencia, de forma que el concepto de Titulación es el de mayor prioridad frente a otras organizaciones administrativas como Departamentos o Áreas de Conocimiento que suelen conducir a una educación universitaria Estática, Fragmentada y Exclusiva.

Agradecimientos

El desarrollo del Proyecto Fin de Carrera SEDA v1.0 ha sido llevado a cabo por los alumnos: Fernando Cabrerizo Guzmán, Damián Carbajo Fernández y Pedro Palomino Suero. Sin su trabajo no habría sido posible este artículo, que está basado en la fase de análisis llevada a cabo durante dicho proyecto.

Referencias

[1] Fernando Cabrerizo Guzmán, Damián Carbajo Fernández y Pedro Palomino Suero. *Proyecto SEDA: Sistemas de e-Docencia-Aprendizaje v. 1.0*. Proyecto Fin de Carrera en Ingeniería Informática. Dpto. de Informática. Escuela Politécnica. Universidad de Extremadura, 2006.

[2] *Informe de Evaluación de Experiencias Piloto ECTS de Curso Completo.* Oficina de Convergencia Europea de la Universidad de Extremadura. Curso 2005-2006. http://www.unex.es/unex/oficinas/oce/archivos/ficheros/documentos/informe_pilotos_2006.pdf

[3] *European Credit Transfer and Accumulation System* (ECTS) (http://en.wikipedia.org/wiki/European_Credit_Transfer_and_Accumulation_System)

[4] *Oficina de Convergencia Europea de la Universidad de Extremadura.* [OCE] (http://www.unex.es/unex/oficinas/oce)

[5] [PATT] *Plan de Acción Tutorial de la Titulación*.(http://www.unex.es/unex/servicios/sofd/areas/oyt/archivos/ficheros/orientacion/pattdelauex.pdf)

Estudio de la carga de trabajo en asignaturas con un gran número de alumnos

Joaquín Gracia Morán, María A. Pinar Sepúlveda
Dpto. de Informática de Sistemas y Computadores
Escuela Técnica Superior de Informática Aplicada
Universidad Politécnica de Valencia
Camino de Vera s/n, 46022 Valencia
{jgracia, mapinar}@disca.upv.es

Resumen

La transformación en créditos ECTS de los créditos actuales va a implicar el redefinir las horas asignadas a una asignatura de tal forma que se habrá de tener en cuenta todas las actividades que realiza el alumno en dicha asignatura.

Además, con la implantación de metodologías activas que conlleva la adaptación al Espacio Europeo de Educación Superior (EEES), no sólo es conveniente estudiar la carga de trabajo del alumno, si no que también es importante estudiar dicha carga de trabajo en el profesor, sobre todo en aquellas asignaturas con un gran número de alumnos matriculados.

En este trabajo se presenta el estudio realizado durante el primer cuatrimestre del curso 2006/07 respecto a la carga de trabajo que tienen los alumnos y el profesor en una asignatura obligatoria de redes de ordenadores, lo que conlleva un gran número de estudiantes matriculados.

Además, en este caso concreto, y con el fin de empezar a adaptarnos al EEES, se ha introducido la posibilidad de que los estudiantes optasen por una evaluación continua mediante portafolio, o por una evaluación 'tradicional', consistente en un único examen al final de la asignatura.

1. Motivación

Con la llegada del EEES [10] las estrategias didácticas desarrolladas hasta ahora deben cambiar. Se va a pasar a realizar una docencia centrada en el aprendizaje del estudiante, en vez de realizar una educación centrada en la enseñanza, convirtiendo de esta manera la universidad en un lugar al que se va a aprender y no a enseñar. En esta línea, un punto importante es la conversión del alumno en un elemento activo dentro de su aprendizaje, incentivando su participación, de tal forma que se sienta parte activa del proceso de enseñanza–aprendizaje.

Con este objetivo, y con el fin de incentivar esta participación, uno de los efectos colaterales de la llegada del EEES es la aplicación de diferentes metodologías activas [1][3].

De esta manera, el centrar el aprendizaje en el estudiante ha provocado que los profesores deban realizar la llamada Guía Docente, que se redacta teniendo en cuenta el tiempo que le cuesta a un estudiante medio superar la asignatura.

La realización de la Guía Docente se puede hacer 'a mano', es decir, los docentes suponen el tiempo que se consumirá en cada una de las tareas en las que se divide la asignatura.

Sin embargo, la realización de la Guía Docente sin tener en cuenta las metodologías activas que se pondrán en marcha con la entrada del EEES puede suponer una estimación poco realista de la carga de trabajo del estudiante, siendo los perjudicados los alumnos.

Otra forma de realizar la Guía Docente es preguntando directamente a los alumnos [2][8][9]. En este último caso, es de crucial importancia que los alumnos respondan de forma realista a la encuesta que se les proporciona.

Por otro lado, si en la Guía Docente se tienen en cuenta todas las actividades del alumno, ¿por qué no tener en cuenta todas las actividades que tiene que realizar un profesor a la hora de preparar la asignatura, sobre todo si tenemos en cuenta la aplicación de metodologías activas en el aula [4][5]?

Con esta perspectiva en mente, en este trabajo se aborda el estudio de la carga de trabajo de los estudiantes durante el curso 2006/07 a la hora de estudiar una asignatura básica de redes de computadores de la Universidad Politécnica de Valencia, así como la dedicación del profesor para

preparar las diferentes actividades que se han realizado a lo largo del curso.

El presente trabajo se ha organizado de la siguiente forma. La sección 2 presenta el contexto de la asignatura, la sección 3 describe el estudio de la carga de trabajo de los alumnos mientras que la sección 4 estudia la carga de trabajo del profesor. Para finalizar, la sección 5 presenta las principales conclusiones de este trabajo.

2. Contexto de la asignatura

En este apartado se van a comentar las razones por las que se eligió esta asignatura para el estudio realizado. Además, se realizará un resumen de la misma.

2.1. Selección de la asignatura

Un aspecto importante a la hora de incorporar innovaciones docentes consiste en la elección de la asignatura donde realizar dichas innovaciones.

Suele ser común usar metodologías activas en los últimos cursos, donde la predisposición de los alumnos suele ser mayor, o el número de estudiantes es reducido.

Sin embargo, ¿qué pasa cuando se aplican metodologías activas en asignaturas con un gran número de alumnos matriculados? En este caso, si bien la carga docente del alumno está especificada en la Guía Docente, no pasa lo mismo con la carga del profesor, el cual puede llegar a verse desbordado.

Siguiendo este razonamiento, para el estudio que se presenta en este trabajo, se ha seleccionado una asignatura obligatoria con 170 alumnos matriculados, divididos en tres grupos. En concreto, los alumnos estaban matriculados de la siguiente forma:

Grupo Teoría	Nº Alumnos Matriculados
Grupo 1	71
Grupo 2	53
Grupo 3	46

Los diferentes grupos están bajo la responsabilidad de dos profesores (Grupo 1 de un profesor y Grupos 2 y 3 el otro profesor), siendo el grupo seleccionado para el análisis el Grupo 1 por ser el más numeroso.

Así pues, el objetivo que se persigue con este estudio es doble. Por un lado, se quiere estudiar la dedicación por parte de los alumnos, según se evalúen utilizando metodologías activas o el examen tradicional. Por otro lado, también se va a estudiar la carga de trabajo del profesor y cómo influye en dicha carga la aplicación de metodologías activas.

2.2. Presentación de la asignatura

El estudio se ha realizado en la asignatura de Fundamentos de Redes de Computadores (FRC), obligatoria dentro de la titulación de Ingeniero en Informática de la Facultad de Informática de la Universidad Politécnica de Valencia [7]. En concreto, se cursa en el primer semestre del tercer curso. Es, además, la primera asignatura relacionada con las redes de computadores que estudian los alumnos, siendo sus contenidos básicos:

- Estudio básico de las comunicaciones entre sistemas informáticos.
- Presentación de la arquitectura de comunicaciones TCP/IP.
- Comprensión del esquema de comunicaciones Cliente/Servidor.
- Estudio de las principales aplicaciones que actualmente funcionan en Internet, así como de su interfaz con el resto de la pila de protocolos.

El temario resumido de FRC es el siguiente:

- TEMA 1. INTRODUCCIÓN.
- TEMA 2. CONFIGURACIÓN AUTOMATICA DE DIRECCIONES (DHCP) Y SERVICIO DE NOMBRES DE DOMINIO (DNS).
- TEMA 3. LA APLICACIÓN DE WORLD WIDE WEB.
- TEMA 4. EL CORREO ELECTRÓNICO.
- TEMA 5. OTRAS APLICACIONES EN RED.
- TEMA 6. INTERFACES DE PROGRAMACIÓN.
- TEMA 7. SEGURIDAD EN LA RED.

Como se puede ver, los contenidos de la asignatura están relacionados con aplicaciones de uso común entre los estudiantes de Ingeniería Informática (como es el correo electrónico o Internet), con lo que los alumnos no tienen miedo a presentarse al examen, tal como se puede ver en la Figura 1, donde se puede observar que el porcentaje de alumnos presentados suele ser del 90% respecto a los alumnos matriculados.

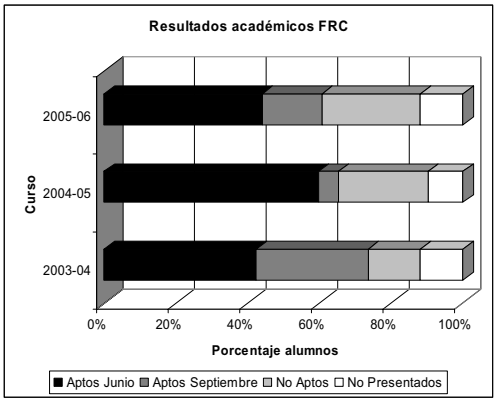


Figura 1. Resultados académicos FRC

Durante el curso 2006/07, y con el fin de empezar a adaptarnos al EEES, se ofreció a los alumnos dos formas de evaluación:

- Evaluación tradicional: examen al final del curso, con un valor del 100% en la nota final.
- Evaluación por portafolio: evaluación continua a lo largo del curso. Básicamente, la realización del portafolio ha consistido en la elaboración de:
  - 5 colecciones de ejercicios,
  - 4 memorias de prácticas,
  - 2 exámenes parciales
  - 1 trabajo realizado en grupo y
  - 1 examen final.

En este primer año de aplicación del portafolio, de todos los alumnos matriculados, el 46% se apuntaron al principio de curso, llegando al final del mismo el 27%. En concreto, en el grupo bajo estudio, de los 71 alumnos matriculados, 25 se apuntaron al portafolio (35%), mientras que llegaron al final 19 (27%), borrándose a lo largo del curso 6 alumnos (8%).

3. Estudio de la carga de trabajo de los alumnos

A continuación se muestra la encuesta que rellenaron los alumnos del grupo bajo estudio. Una vez por semana y durante todas las semanas que duró el curso, se rellenaba este formulario con el fin de poder estudiar el tiempo dedicado a la asignatura por parte de los alumnos.

Horas que has dedicado a:	Aula	Laboratorio	Casa
Teoría			
Prácticas			
Trab/Probl/etc			

¿Has ido a tutorías? ¿Cuántas horas?
Nº asignaturas matriculado en el curso 2006/07:
Cuatr. A Cuatr. B
Nº convocatoria FRC:
¿Estás haciendo el portafolio? SI NO
¿Quieres comentar algo sobre la asignatura?

Además de averiguar el tiempo que dedicaban los alumnos a las distintas tareas que se realizan durante el curso, como puede ser la asistencia a clases teóricas o prácticas o el trabajo realizado en casa, también se intentó situar al alumno dentro de su contexto, preguntándole para ello si se evaluaba mediante portafolio, el número de convocatoria de la asignatura y el número de asignaturas de las que estaba matriculado en cada cuatrimestre. También se pedía a los alumnos su opinión sobre el funcionamiento de la asignatura.

Una cuestión importante fue el concienciar a los alumnos para que rellenaran la encuesta sin 'exagerar' sus méritos, con el fin de obtener los datos más realistas posibles.

Los datos recogidos se muestran en la Figura 2, donde podemos ver el promedio de horas que ha dedicado cada alumno durante cada semana del curso, sin tener en cuenta si se evaluaba mediante portafolio o con el examen final, mientras que la Figura 3 y la Figura 4 muestran la dedicación media de cada alumno en función de si se evaluaban mediante el portafolio o mediante evaluación tradicional.

De todas estas figuras podemos extraer varias conclusiones interesantes:

- A tutorías sólo asisten los alumnos que siguen el portafolio. Sólo un alumno que no seguía el portafolio vino durante la última semana del curso (cuando se acercaba el examen final).
- A medida que se acerca el final del curso (y con ello el examen final), los alumnos que se evalúan únicamente con este examen dedican más tiempo a la asignatura.
- Los alumnos que no siguen el portafolio no hacen nada 'extra' de la asignatura. Por ejemplo, en la semana 12 se hizo el segundo examen parcial el mismo día que se pasaba la encuesta. Como se puede ver, a este examen

parcial sólo vinieron alumnos que se evaluaban mediante portafolio, a pesar de insistir en clase que cualquier alumno podía realizar dicho examen.

- Con el portafolio se ha conseguido que los alumnos trabajen de forma más continuada durante todo el curso, tal como se planteó al principio al aplicar esta forma de evaluación.
- Como se puede ver en la Figura 5, casi todos los alumnos que seguían el portafolio asistían a clase, mientras que los alumnos que no lo seguían, no sólo no venían a las sesiones teóricas, si no que tampoco asistían a las sesiones prácticas.
- El número de alumnos repetidores que hicieron el portafolio es mínimo. En el grupo donde se ha estudiado la carga de trabajo, sólo un alumno repetidor hizo el portafolio, mientras que el resto de repetidores ni siquiera asistían a clase, tal como se puede ver en la Figura 5.
- Respecto al portafolio, podemos ver que hay cuatro picos, uno de ellos bastante marcado. En concreto, estos picos se corresponden a las semanas 7, 10, 12 y 13, que se corresponden con el primer examen parcial (semana 7), el tiempo que dedicaron a la asignatura durante la semana del 6-8 de diciembre y la semana siguiente (semana 10), las dos semanas navideñas junto con el segundo examen parcial (semana 12) y la entrega del trabajo en grupo (semana 13). Es decir, que a pesar de llevar la asignatura más o menos al día, aún esperaban a la última semana para realizar el trabajo que se tenía que entregar.

La recopilación de datos presentada en este trabajo servirá para ajustar las diferentes tareas a realizar en futuros cursos dentro del portafolio, de tal forma que el tiempo que los alumnos dediquen a la asignatura no les requiera más horas de las marcadas por los ECTS. De esta manera, la elaboración de la Guía Docente de FRC se hará de una forma realista, teniendo en cuenta el tiempo que dedican los alumnos, y no suponiendo dichos tiempos desde el punto de vista del profesor.

Por otra parte, estos datos también nos han servido para ver que los alumnos repetidores no asisten ni realizan ninguna actividad en la asignatura, siendo el caso más común que únicamente aparezcan a la hora de realizar el examen final.

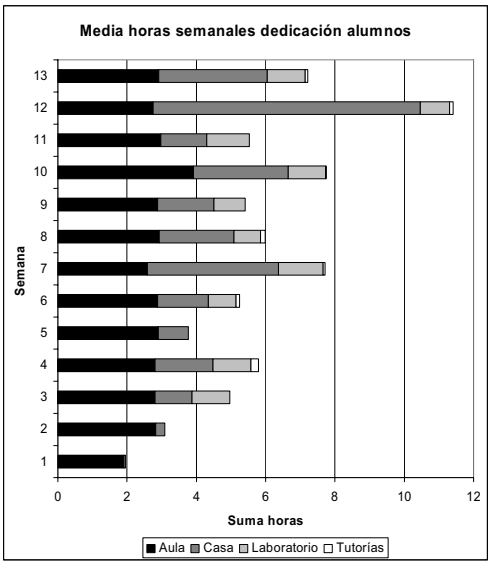


Figura 2. Media semanal horas por alumno (general)

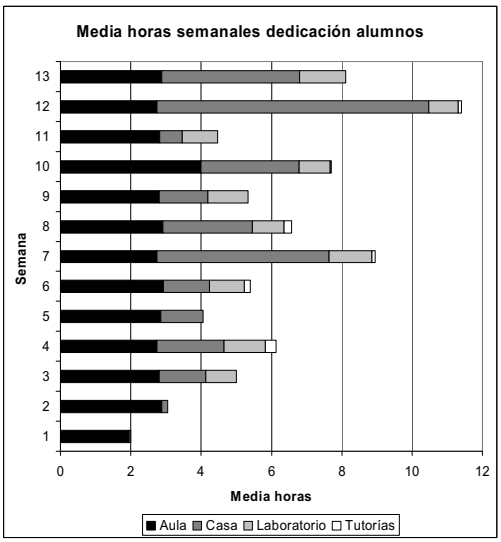


Figura 3. Media semanal horas por alumno (portafolio)

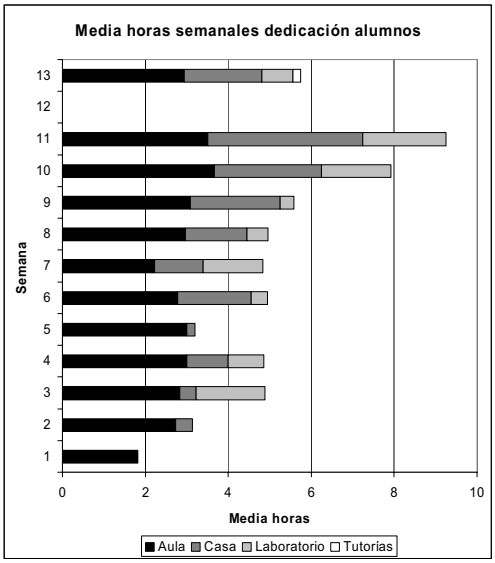


Figura 4. Media semanal horas por alumno (sin portafolio)

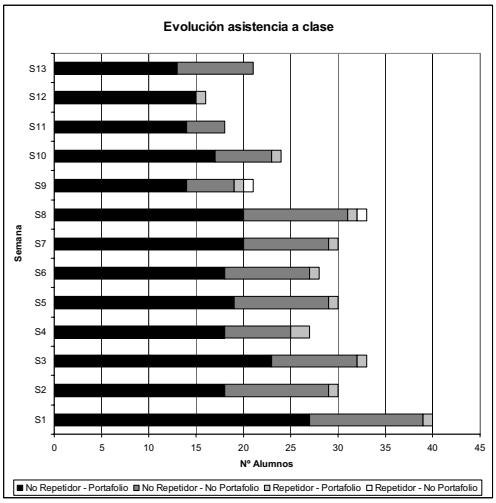


Figura 5. Asistencia a clase

4. Estudio de la carga de trabajo del profesor

Respecto a la dedicación del profesor, la Figura 6 muestra únicamente las horas consumidas en todas las actividades relacionadas con la asignatura FRC durante las semanas en las que hay docencia.

Como en el apartado anterior, en la Figura 6 también podemos ver dos picos, correspondientes a las semanas 4 y 10. En este caso, en la semana 4 del curso se iniciaron las entregas del portafolio. Como suele ser habitual, al principio los alumnos están bastante motivados, por lo que esa entrega fue la más numerosa así como durante esa semana, bastantes alumnos se acercaron a tutorías. En cuanto a la semana 10, ésta incluye el tiempo de la semana del 6-8 de diciembre.

Por otra parte, la Figura 7 muestra la dedicación del profesor a lo largo de todo el curso. Esta dedicación incluye el trabajo previo realizado (marcado como S0 en la figura), así como el trabajo de preparación, realización y corrección del examen final (marcado en este caso como S14).

De la Figura 6 y la Figura 7 podemos deducir:

- Aunque la mayor parte del tiempo se dedica a la teoría, las actividades relacionadas con el portafolio también tienen un impacto importante. Cuando este método de evaluación sea obligatorio para todos los alumnos, el tiempo que habrá que dedicar a este apartado será mucho mayor.
- Como se puede observar en la Figura 6, el tiempo mínimo de dedicación es de unas 12 horas semanales, aunque el tiempo medio supera las 16 horas por semana.
- Si dividimos el tiempo dedicado a las distintas actividades realizadas durante el curso, obtenemos que el tiempo medio semanal dedicado por el profesor del Grupo 1 a las diferentes actividades es de 8’5 horas a la teoría, 3 horas a las prácticas y 5 horas a otras actividades. Estas 5 horas son las añadidas principalmente por la aplicación del portafolio en la evaluación. Es decir, que teniendo en cuenta que en el Grupo 1 sólo se apuntaron un 35% de los alumnos al portafolio, si tuviésemos el 100% de alumnos apuntados, el tiempo medio que habría que dedicar a las otras actividades sería de unas 15 horas semanales, que sumadas a las 8’5 horas de teoría y 3 horas de prácticas, nos daría una jornada semanal de unas 26’5 horas, sólo para FRC. Si la jornada laboral es de 35 – 40 horas, esto implica que para el resto de actividades

que debe desarrollar el docente, tendremos entre 8'5 – 13'5 horas disponibles.

- Otro tiempo que habrá que tener en cuenta será el de las tutorías. A pesar de poderse hacer de forma presencial o utilizando la red [6], este tiempo será mayor cuantos más alumnos participen en el portafolio, restando de esta manera tiempo para otras actividades.
- Un aspecto interesante es la interacción con el alumnado. El hecho de tener un contacto continuado con el estudiante provoca un mayor conocimiento del mismo, lo que mejora el funcionamiento general de la asignatura.
- La evaluación continua introducida por el portafolio provoca que antes de la realización del examen final el docente tenga una idea bastante aproximada del resultado final de sus alumnos, lo que facilita la evaluación de los mismos.

En resumen, las metodologías activas suponen un incremento del tiempo que el profesor debe dedicar a la asignatura. Si tenemos en cuenta que con la llegada del EEES, todas las asignaturas seguirán metodologías activas, esto implica que la dedicación del profesor puede llegar a exceder la jornada laboral asignada.

Sin embargo, la proximidad que se consigue con los alumnos es tal que el ambiente en el aula y el funcionamiento de la asignatura es bastante agradable.

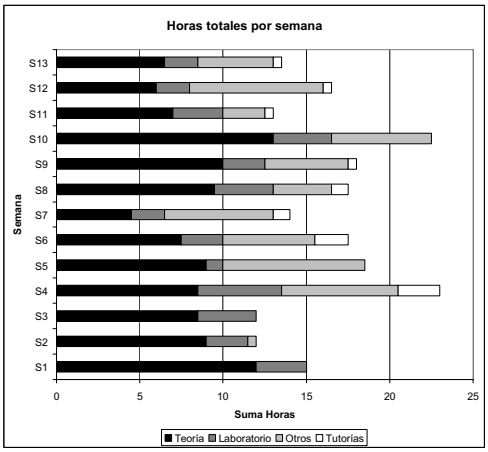


Figura 6. Dedicación del profesor (semanas docencia)

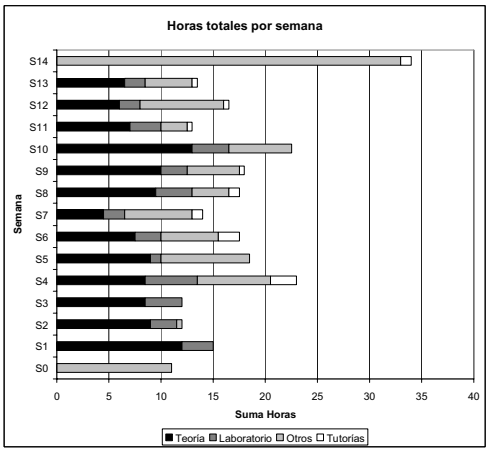


Figura 7. Dedicación del profesor (total)

5. Conclusiones

Este trabajo presenta el estudio de la carga de trabajo del docente y del discente en una asignatura con un gran número de alumnos matriculados, y donde se han empezado a aplicar metodologías activas con el fin de adaptarse al EEES.

Gracias a este estudio, ya se tienen las bases para el desarrollo de una Guía Docente de la asignatura lo más realista posible.

En cuanto a los resultados obtenidos, un dato curioso es la media de horas utilizadas por los alumnos y por el profesor, sin tener en cuenta el tiempo que han dedicado a preparar el examen. Respecto a los alumnos, aquellos que no han seguido el portafolio han dedicado una media de 62 horas a la asignatura. Si FRC tiene 6 créditos actualmente, esto implica que estos alumnos se han limitado a ir a clase y prácticas durante el curso, y al final del mismo es cuando han empezado a mirar algo de la asignatura.

Sin embargo, aquellos alumnos que han realizado el portafolio, su media es de unas 80 horas. En este caso, sí que se ha conseguido que los alumnos trabajen durante todo el curso.

En cuanto al profesor, y tal y como se ha comentado en la sección 4, la media de horas dedicadas a la asignatura supera las 16 semanales durante las semanas con docencia. Si tenemos en cuenta que solamente un tercio de los alumnos ha realizado la evaluación por portafolio, este tiempo

crecerá hasta unas 26'5 horas cuando todos los alumnos estén implicados en este método de evaluación, lo que va a provocar que al docente le falten horas en su jornada laboral para poder realizar todas sus tareas (docencia, investigación y gestión).

Por otra parte, la aplicación del portafolio ha permitido que los alumnos que lo siguen hayan realizado un trabajo más continuado y repartido a lo largo del curso, y por otro lado, la interacción con el profesor ha sido mayor, llegando a generarse un clima de confianza que ha provocado un buen ambiente en el aula.

Sin embargo, varias dudas han surgido a la hora de aplicar metodologías activas en el aula:

- ¿Qué pasará con los alumnos repetidores? En este curso hemos observado que la mayor parte de estos alumnos sólo han venido para hacer el examen, no habiendo participado en ninguna de las actividades de la asignatura (ni siquiera han ido a clases teóricas o prácticas).
- ¿Qué pasará con la jornada laboral del docente? Si con los ECTS se persigue que un alumno *trabaje* entre 35 – 40 horas a la semana, los docentes también deberían hacer lo mismo. Si todas las asignaturas se desarrollan aplicando metodologías activas, al profesor le va a faltar tiempo para poder realizar todas las actividades que se le exigen (docencia, gestión e investigación).
- Por último, también sería interesante contestar a la pregunta ¿cuántos ECTS ha de impartir un profesor?

Referencias

[1] Blanc Clavero, S. et al. *Metodologías activas para facilitar el aprendizaje en un curso básico de Tecnología de Computadores.* VII Congreso en Tecnologías Aplicadas a la Enseñanza de la Electrónica, Madrid, 2006.

[2] Conejero Casares, J.A. et al. *La dedicación del alumno en la Facultad de Informática de la Universidad Politécnica de Valencia.* 4º Congreso Internacional de Docencia Universitaria e Innovación, Barcelona, 2006.

[3] Fernández March, A. et al. *Metodologías activas para la formación de competencias.* Material de trabajo. Instituto de Ciencias de la Educación, Universidad Politécnica de Valencia, 2006.

[4] Fernández Nieto, M.J. et al. *Comparación de la carga de trabajo y evaluación de resultados en una experiencia piloto de innovación educativa en el primer curso de Ingeniería Técnica Agrícola, especialidad en Hortofrutícola y Jardinería.* 4º Congreso Internacional de Docencia Universitaria e Innovación, Barcelona, 2006.

[5] García Guzman, J. et al. *Análisis experimental de la carga de trabajo requerida para completar una asignatura universitaria de cara a la transición hacia el Espacio Europeo de Educación Superior*. XII Jornadas de Enseñanza Universitaria de Informática (JENUI'06), pp. 153-159, Bilbao, 2006.

[6] Gracia Morán, J. et. al. *Nuevos retos para el profesorado universitario: experiencias de la tutoría como estrategia metodológica.* 4º Congreso Internacional de Docencia Universitaria e Innovación, Barcelona, 2006.

[7] http://www.fiv.upv.es

[8] Martínez Cortés, J.P. et al. *Evaluación de la carga discente de la titulación de Ingeniería de Telecomunicación: Asignación de créditos ECTS.* 4º Congreso Internacional de Docencia Universitaria e Innovación, Barcelona, 2006.

[9] Posadas Yagüe, J.L. et al. *Estudio de la carga de trabajo del alumnado en las titulaciones de ITIG e ITIS para la adaptación al EEES.* XII Jornadas de Enseñanza Universitaria de Informática (JENUI'06), pp. 17-24, Bilbao, 2006.

[10] Tuning Educational Structures in Europe project, "Approaches to teaching, learning and assessment in competence based degree programmes", 2005. http://www.unideusto.org/tuning.

FAQs sobre la adaptación de las asignaturas al EEES[1]

Miguel Valero-García* y Juan J. Navarro**

(*) Escuela Politécnica Superior de Castelldefels
(**) Facultad de Informática de Barcelona
Departamento de Arquitectura de Computadores
Universidad Politécnica de Cataluña
e-mail: {miguel, juanjo}@ac.upc.es

Resumen

Muchas son las preguntas y las dudas que se plantean los profesores a la hora de adaptar una asignatura al Espacio Europeo de Educación Superior. Pasar de un modelo de docencia centrado en la enseñanza a un modelo centrado en el aprendizaje plantea diferentes tipos de preguntas, que suelen repetirse en todos los foros de discusión. Algunas de estas preguntas tienen que ver con la pertinencia de los planteamientos que se hacen *(¿Es este enfoque realmente mejor?*). Otras tienen que ver con la dificultad de implantar los nuevos enfoques en el contexto actual *(¿Puede hacerse esto en nuestras aulas y con nuestros alumnos?).* Y finalmente, algunas preguntas tienen que ver con aspectos técnicos, que surgen cuando uno se pone manos a la obra *(¿Cómo se estima el tiempo de dedicación del alumno a una tarea?)*. En este artículo respondemos a ocho de estas preguntas típicas, agrupadas en los tres apartados comentados, con la esperanza de que las respuestas ayuden a los profesores a adaptar sus asignaturas al nuevo entorno.

1. Introducción

La adaptación al Espacio Europeo de Educación Superior (EEES) y, en particular, la adopción del sistema europeo de créditos (ECTS) está propiciando un debate intenso sobre la necesidad de reforma de los métodos docentes en la enseñanza universitaria. En concreto, se dice que es necesario adoptar un enfoque centrado en el aprendizaje y en el alumno. Este enfoque requiere una formulación detallada de los objetivos formativos incluyendo conocimientos, habilidades y actitudes junto con una programación meticulosa de las tareas que debe realizar el alumno dentro y fuera del aula para conseguir esos objetivos. Se dice también que la planificación debe incluir una estimación del tiempo de dedicación a cada tarea y que deben usarse enfoques docentes que ayuden al estudiante a seguir las actividades del plan, como son el trabajo cooperativo, el aprendizaje activo, la evaluación continuada, el aprendizaje basado en problemas, etc.

Los autores de esta ponencia acumulamos una cierta experiencia en reuniones de debate y actividades formativas en torno a este proceso de adaptación al EEES. Hemos observado que en estos foros se repiten habitualmente el mismo tipo de preguntas, con independencia del contexto local en que se desarrolle la actividad. Por ese motivo, hemos estimado oportuno realizar una recopilación de estas preguntas habituales y nuestras respuestas favoritas. Esperamos que este material ayude a algunos profesores a ponerse en marcha para este largo viaje, o haga el viaje más sencillo para los que ya han empezado a caminar.

Hemos agrupado las preguntas en tres secciones según hacen referencia a la validez de los nuevos métodos docentes, a la dificultad de implantar los nuevos enfoques en el contexto actual de nuestra universidad y a los problemas técnicos de implementación de los nuevos métodos docentes.

2. ¿Son pertinentes los métodos docentes centrados en el aprendizaje?

2.1. Objetivos formativos específicos

Estamos acostumbrados a formular los objetivos *de la asignatura en términos bastante generales. Ahora parece que debemos especificarlos con mucho más detalle. En una asignatura en la que, por ejemplo, se hacen pruebas de evaluación continua, la lista de objetivos específicos dice*

[1] Este trabajo se ha realizado con la ayuda del Departamento de Arquitectura de Computadores, la Facultad de Informática de Barcelona y la Escuela Politécnica Superior de Castelldefels, ambos de la Universidad Politécnica de Cataluña

prácticamente lo que se le va a preguntar al alumno en esas pruebas ¿No es éste un planteamiento excesivamente dirigido y paternalista?¿No es mejor sorprenderlos en cada examen para que así estudien más?

Los objetivos específicos de una asignatura detallan lo que el alumno debe saber (conocimientos), *saber hacer* (habilidades) y *querer hacer* (actitudes) al final del curso. Cada objetivo del ámbito de los conocimientos y las habilidades se formula mediante una frase en la que el sujeto es el alumno y el verbo es un verbo de acción, evaluable, que sirva para hacer una pregunta. No se pueden usar verbos como comprender, entender, etc., pero si veros como definir, comparar, calcular, deducir, analizar, diseñar, etc. sí que son verbos útiles para hacer una pregunta de examen. Además, la frase que formula un objetivo específico debe tener suficientes complementos para concretar al máximo el tema del objetivo y las circunstancias en que se desarrolla la acción. Nos gusta decir que la lista de objetivos específicos esta bien formulada si cumple la prueba del profesor Ruso (un profesor que sabe mucho del tema de nuestra asignatura, pero no ha asistido a ninguna de nuestras clases, ni sabe qué documentación seguimos, ni qué problemas hemos hecho). Con solamente la lista de objetivos específicos el profesor Ruso debería poder poner los exámenes de nuestra asignatura y que nuestros estudiantes no se sorprendieran.

Si el estudiante conoce los objetivos específicos [1]:

- tiene claro, desde el comienzo, qué es lo que se espera de él, lo cual disminuye la tensión de cara a los exámenes porque hay menos terreno para las sorpresas y le permite enfocar los esfuerzos en la dirección correcta desde el primer momento,
- tiene todos los elementos de juicio para autoevaluar su proceso de aprendizaje,
- está preparado, desde el primer momento, para diferenciar lo importante de lo secundario (con respecto a lo que se le explica en clase o está en los libros),
- está más motivado porque le resulta fácil ver el resultado final como asequible.

Imaginemos una situación que nos puede hacer pensar: si los profesores tuviéramos que formar a nuestros alumnos para que una agencia independiente los evaluara y a nosotros nos pagaran una parte sustancial de nuestro sueldo en función del número de nuestros alumnos aprobados, ¿no seriamos nosotros los primeros en exigir la lista detallada de los objetivos que van a ser evaluados? Seguro que conocidos los objetivos conseguiríamos un mayor éxito de manera más efectiva y eficiente.

Pero también hay ventajas para los profesores. La lista de objetivos específicos facilita enormemente la coordinación entre diferentes profesores de una misma asignatura, aclarando qué preguntar en los exámenes y sobretodo ayudando a seleccionar los métodos docentes más adecuados para cada uno de los objetivos perseguidos, lo cual facilita una recomendable diversificación de estrategias.

En este sentido puede ayudarnos mucho, en asignaturas de ingeniería, la clasificación de los objetivos en tres niveles de competencia nivel 1 (conocimiento), nivel 2 (comprensión) y nivel 3 (que llamamos aplicación pero engloba al nivel 3 y superiores de la taxonomía de Bloom [2]). Una pregunta para evaluar un objetivo de nivel 1 se puede responder “de memoria”. Para resolver un ejercicio que evalúa un objetivo de nivel 2 se debe seguir un procedimiento conocido y la solución del ejercicio es única. Sin embargo, la resolución de un problema de nivel 3 requiere tomar decisiones, elegir entre varios métodos, generar una idea nueva, analizarla, evaluarla y modificarla si no sirve para solucionar el problema. Estos problemas suelen admitir varias soluciones válidas.

No deja de ser extraño que en nuestros exámenes apenas aparezcan objetivos de nivel 1 cuando la mayor parte de nuestras clases son expositivas, que es una forma adecuada de transmitir conocimientos de este nivel, pero no de niveles superiores. Sin embargo tenemos pocas clases de problemas o de laboratorios guiados, que son adecuadas para conseguir objetivos de nivel 2, siendo las preguntas de este nivel las que más aparecen en nuestros exámenes. Por otro lado, para conseguir objetivos de nivel 3 son más adecuadas asignaturas diferentes a las clásicas, que se basen, por ejemplo, en proyectos y con un seguimiento más individualizado del profesor. Estos objetivos de nivel 3 son los que nos gustaría que tuvieran nuestros alumnos, sobre todo en los

últimos años de carrera, pero seguimos haciendo las clases como siempre.

En definitiva, una buena formulación de objetivos es un elemento fundamental, tanto desde el punto de vista de los alumnos como del de los profesores.

2.2. Planificación de actividades y entregas

Los métodos docentes que se están planteando requieren una planificación detallada de todo lo que debe hacer el alumno *a lo largo del curso ¿No es este planteamiento, otra vez, excesivamente paternalista?¿No corremos el peligro de producir titulados con escasa iniciativa propia y poca autonomía para hacer las cosas por sí mismos?*

Efectivamente, el elemento clave de la programación docente centrada en el aprendizaje es la planificación detallada de todo lo que deben hacer los estudiantes (dentro y fuera de clase) para alcanzar los objetivos de aprendizaje. Se deben establecer también con detalle cuáles son los productos de las tareas encomendadas (que llamamos entregas) y cuándo deben estar preparadas estas entregas. Esto contrasta con lo que podríamos llamar programación centrada en la enseñanza, en la que el elemento clave es la planificación de las tareas que va a realizar en profesor y, más en concreto, sus explicaciones en clase. En otras palabras, los profesores tenemos que planificar y gestionar las horas de trabajo del alumno que se corresponden con la asignación de ECTS de nuestra asignatura. ¿Es ésto excesivamente paternalista?

Lo primero que hay que decir es que un buen plan formado por pasos que se perciben como asequibles, es un elemento clave de motivación para conseguir que los alumnos realicen las tareas y con ello aprendan. Pero no sólo los alumnos. Se sabe que las personas, en general, funcionamos mejor cuando tenemos un plan detallado para desarrollar una determinada actividad. Esto es exactamente lo que hacen los ingenieros cuando se enfrentan a un proyecto: establecer un plan detallado de lo que hay que hacer, cuándo hay que hacerlo y cuál es el resultado de cada paso del plan. No podemos aceptar que aquello que en el mundo profesional se considera una buena práctica, en el mundo académico sea considerado como excesivamente paternalista.

Sabemos también que si el trabajo a realizar conduce a un resultado ambicioso, los alumnos se sienten doblemente motivados. Y justamente, cuanto más ambicioso sea el resultado final, más necesario es establecer un plan detallado paso a paso del camino a seguir. Uno puede plantearse subir a la montaña de su pueblo sin excesiva planificación. Simplemente echa un bocadillo en la mochila y se pone a caminar. Ya parará a comérselo cuando lo considere oportuno. Pero si lo que se propone es escalar el Everest, entonces necesita establecer un plan muy minucioso de todo lo que va a hacer, dónde va a establecer los diferentes campamentos base, qué va a ingerir en cada momento, y cuántos metros va a avanzar en cada esfuerzo. En realidad, la falta de un plan detallado de lo que deben hacer los alumnos en nuestras asignaturas nos impide conseguir objetivos más ambiciosos con ellos.

Finalmente, hay que decir que no necesariamente el profesor establece siempre todos los detalles del plan de trabajo de sus alumnos. Lo adecuado es que el profesor de primeros cursos sí establezca con detalle el plan de trabajo y las herramientas para que el estudiante lo siga (trabajo cooperativo, evaluación continua,…). Los estudiantes que siguiendo el plan obtienen buenos resultados, viven en su propia piel lo importante del trabajo continuo, del esfuerzo y de una planificación adecuada. Estas planificaciones rígidas de los cursos iniciales sirven de ejemplo para los alumnos. A medida que avanzan en los estudios, se deben encontrar con planes más abiertos e indefinidos, de forma que sean ellos mismos los que acaben de concretar los detalles de cómo y cuando van a hacer las tareas e incluso decidan qué tareas van a hacer.

En resumen, un plan detallado de trabajo no sólo no es un elemento paternalista, sino que es un elemento indispensable para la buena ingeniería. Es nuestra responsabilidad enseñar a los alumnos primero a seguir planes de trabajo y luego también a diseñarlos.

2.3. Evaluación

Con frecuencia se aportan datos que indican que con los métodos centrados en el alumno se obtiene un mayor rendimiento *académico, medido en términos de porcentajes de aprobados. Pero estos datos suelen suscitar la preocupación de que una excesiva valoración del trabajo que hacen los*

alumnos fuera del control directo del profesor (y en particular, del trabajo en grupo) y una reducción del peso de los exámenes en la evaluación conduzca a un menor nivel de exigencia. En otras palabras, con todo esto ¿no estamos bajando el nivel?

En el contexto de la programación docente centrada en el aprendizaje, la tarea fundamental del profesor es diseñar un camino que conduzca inexorablemente al aprendizaje y conseguir, además, que los alumnos recorran ese camino. Para ello, debe utilizar todos los elementos de motivación que tenga a su alcance, como por ejemplo, el trabajo en grupo [3]. Otro elemento de motivación es el método de evaluación, que debe premiar a los alumnos que realizan a tiempo las tareas previstas. Las pruebas o controles puntuables deben ser parecidos a las tareas que han realizado, de forma que el que las ha realizado apruebe esos controles. En ese contexto, el clásico examen final, en el que se decide gran parte de la calificación del alumno, con independencia del trabajo que ha realizado durante el curso, es un elemento que no encaja. Es más, el examen final puede ser incluso perjudicial, especialmente si proyecta ante los ojos del alumno la ilusión de que existe un camino alternativo más corto, un atajo, para superar la asignatura.

Cuando hablamos del nivel de una asignatura usualmente nos referimos al nivel de dificultad de los exámenes que los alumnos deben superar. Desde ese punto de vista, algunos métodos centrados en el alumno pueden suponer una bajada del nivel de dificultad de los exámenes, puesto que éstos pierden protagonismo en beneficio de otras actividades, e incluso pueden llegar a desaparecer.

Pero si bien estamos en contra del clásico examen final, no tenemos nada en contra de los exámenes/pruebas/controles en un entorno de evaluación continua y sobre todo para evaluar objetivos de niveles de competencia 1 (conocimiento) y 2 (comprensión). Así por ejemplo, una estrategia que estamos experimentando consiste en usar los exámenes para verificar de forma individual que los alumnos han adquirido los objetivos de los niveles de competencia 1 y 2, claramente establecidos al inicio del curso. El resto de los objetivos, de niveles de competencia más altos, deben demostrarlos por otras vías (por ejemplo, realizando un proyecto en grupo). Los exámenes de niveles de competencia bajos son de una dificultad claramente inferior a los exámenes de antes, cuando el examen final era el único elemento de evaluación y se pretendía evaluar allí también los objetivos de niveles de competencia más altos (que por otro lado son difíciles y a veces imposibles de evaluar en un examen que dura unas pocas horas). No creemos que esto sea bajar el nivel.

Por otro lado, hay otros elementos para valorar el nivel de una asignatura. Podemos fijarnos, por ejemplo, en los resultados de las tareas realizadas por los alumnos. Desde ese punto de vista no podemos aceptar que se esté bajando el nivel. Tenemos suficientes evidencias para afirmar que los alumnos, bien organizados en grupo, son capaces de realizar tareas de elevada complejidad y relevancia profesional, que con frecuencia sorprenden a sus profesores y a ellos mismos. Esto es especialmente cierto en asignaturas en las que se trabaja por proyectos [4], en las que se establece un objetivo ambicioso para los alumnos, y un plan detallado de pasos asequibles.

3. ¿Se puede implementar la docencia centrada en el aprendizaje en el contexto actual de nuestra universidad?

3.1. Número de alumnos por aula

Los métodos centrados en el aprendizaje *requieren sin duda un escenario más apropiado que el que tenemos ahora: aulas masificadas y con un mobiliario que no facilita el trabajo en grupo, que parece ser un elemento clave. ¿Cómo nos enfrentamos a estas circunstancias "hostiles"?*

El escenario es el que es, no porque no haya alternativas, sino porque es un escenario que funciona bien para los métodos tradicionales y no hemos tenido la necesidad de cambiarlo. El esfuerzo que hagamos ahora los profesores para cambiar los planteamientos docentes de nuestras asignaturas debe venir acompañado de un esfuerzo de los responsables académicos para ir adaptando el escenario, cambiando el mobiliario

de las aulas y el tamaño de los grupos de clase. Y así debemos reclamarlo.

No obstante, teniendo en cuenta que las cosas no van a cambiar de forma inmediata, tenemos que estar dispuestos a experimentar incluso en condiciones no óptimas. Para ello, podemos apoyarnos en técnicas que son bien conocidas y que pueden facilitarnos mucho la labor.

Así por ejemplo, sabemos que la dificultad más grande que plantea el tener un grupo numeroso de alumnos en clase es la de poder darles a todos ellos retroalimentación con frecuencia y a tiempo, a partir de las entregas y las pruebas que vayan realizando a lo largo del curso. Para enfrentarse a ese reto pueden ser de gran utilidad estrategias de autoevaluación o evaluación entre compañeros, en virtud de las cuales, los propios alumnos ayudan al profesor a evaluar los trabajos realizados por ellos mismos, o por compañeros, a partir de unos criterios claramente establecidos por el profesor. Además, estas estrategias inciden en otros objetivos formativos importantes, tales como la capacidad de reflexionar sobre el propio trabajo realizado, o la capacidad de emitir juicios de valor sobre el trabajo de otros.

Finalmente, y también al hilo del problema del número elevado de alumnos en clase, que parece ser una de las barreras de la innovación docente más frecuentemente mencionadas, conviene tener en cuenta que en muchas escuelas y facultades el problema empieza a ser ya el contrario: una falta de alumnos de nuevo ingreso. Eso es especialmente cierto en algunas ingenierías, como telecomunicaciones o informática. No es razonable que, de forma casi instantánea, pasemos de la preocupación por la masificación a la preocupación por la falta de alumnos. En algún punto intermedio de esta transición debe haber una oportunidad para una organización docente basada en grupos pequeños de clase (entre 30 y 40) que facilite la implantación de métodos innovadores que puedan ayudar a recuperar el interés de los alumnos en nuestros planes de estudio. Esta debe ser también una de nuestras reclamaciones para los responsables académicos.

3.2. Normativas de evaluación

Otro elemento *del contexto, que está muy arraigado, es la normativa de evaluación que suelen tener muchas universidades y que establece el derecho de los alumnos a ser evaluados en un examen final y, en algunos casos, el derecho a diversas convocatorias extraordinarias. Tal y como se ha indicado antes, parece que estos elementos pueden ser perjudiciales de cara la implantación de los nuevos métodos docentes ¿Cómo se enfrenta uno a esta situación?*

En primer lugar, insistimos, exigiendo que la normativa también se adapte al EEES. Existen casos de universidades que, por ejemplo, han eliminado las convocatorias extraordinarias de septiembre (e incluso las de julio) sin que se haya producido ninguna revuelta de los estudiantes.

Debería ser fácil convencer a los responsables académicos de nuestras universidades de que hay objetivos en el ámbito del conocimiento y sobre todo de las habilidades y de las actitudes que es imposible evaluar mediante un examen de una convocatoria extraordinaria.

No obstante, mientras se producen esas adaptaciones normativas, que no serán inmediatas, conviene tomar las medidas necesarias para eludir las dificultades que pueda plantearnos la normativa en vigor. En concreto, tal y como se ha indicado antes, lo esencial es evitar que la presencia de un examen final o de convocatorias extraordinarias proyecte ante el alumno la ilusión de que existe un camino más corto para aprobar. No es deseable que algunos (o muchos) alumnos acaben optando por ese atajo en el momento en que el curso se pone difícil. Probablemente las normativas no impiden, por ejemplo, que las pruebas de esas convocatorias extraordinarias estén condicionadas a las tareas realizadas durante el curso, de forma que sea *de facto* imposible superar esas pruebas si no se han realizado las tareas, o una parte importante de ellas. Nuestra experiencia es que aquellos profesores que han adoptado este planteamiento han conseguido que las convocatorias extraordinarias no representen una dificultad significativa para el despliegue de sus programas centrados en el aprendizaje.

3.3. Compaginar estudio y trabajo

¿Y qué hacemos con los alumnos que compaginan estudios y trabajo? Está claro que, al menos en el ámbito de las ingenierías, el número de alumnos que trabajan es elevado, y también está claro que estos métodos docentes son mucho más exigentes

en cuanto a la asistencia a clase y al trabajo continuado. *¿Cómo podemos tener en cuenta esta realidad?*

Una primera reflexión al respecto es que, a nuestro entender, hay dos tipos de alumnos que compaginan estudios y trabajo: los que inevitablemente necesitan trabajar para pagarse los estudios y los que trabajan porque encuentran poco valor en lo que ocurre en clase y consideran más rentable usar su tiempo trabajando, consiguiendo de alguna manera el material del curso y preparándose los exámenes cuando toque. Estamos convencidos que la mayoría de estudiantes son de este segundo grupo. Uno de los objetivos de nuestra enseñanza debería ser el conseguir que nuestros alumnos adjudicaran tanto valor a lo que ocurre en clase y a las tareas que hemos preparado para ellos, que se dijeran a si mismos: *"Esto no me lo puedo perder. Ya trabajaré cuando acabe la carrera"*.

En cualquier caso, no conviene poner en el sistema más dificultades de las estrictamente imprescindibles. No pongamos énfasis en la obligatoriedad de la asistencia a clase (lo cual puede llevar a un debate conflictivo con los alumnos), sino en la obligatoriedad de realizar las tareas y entregar los resultados en el momento establecido para ello (lo cual es mucho menos discutible). Además, conviene ofrecer una cierta flexibilidad en las entregas, que se negocia en privado, en función de las circunstancias particulares. También conviene tomar algunas precauciones a la hora de formar grupos (por ejemplo, conviene poner en el mismo grupo a alumnos que trabajan, pero que pueden reunirse si es necesario durante los fines de semana). Esto suele ser suficiente para que muchos de los alumnos que trabajan puedan seguir el curso, con mucho esfuerzo, pero sin poner en cuestión los principios organizativos de la programación centrada en el aprendizaje.

Los problemas serios acaban surgiendo en el caso de los alumnos que simplemente no disponen del número de horas de dedicación que requieren los ECTS asignados a la asignatura. En este caso la respuesta no puede ser otra: "Vuelve a matricularte cuanto tengas ese tiempo disponible".

En cualquier caso, conocemos ejemplos en los que el profesor ha sido capaz de arbitrar un sistema en el que, desde el primer momento se detecta cuáles son los alumnos que no pueden seguir el plan establecido por falta de tiempo, y negocia con ellos un plan alternativo (el plan B), que puede incluir algunas horas de consulta y dos o tres exámenes, más o menos tradicionales, sin perjudicar con ello la dinámica general del curso. Después de todo, ese plan B es el que hubiesen seguido, en el formato tradicional, los alumnos que no tienen tiempo para venir a clase.

4. ¿Qué problemas técnicos aparecen al poner en práctica la docencia centrada en el aprendizaje?

4.1. Créditos ECTS. Tiempo de cada actividad

La programación centrada en el aprendizaje requiere una estimación razonable del tiempo que debe dedicar el alumno a cada una de las tareas que han sido planificadas. Sin embargo, todos sabemos que puede haber una elevada varianza en ese tiempo de dedicación. Habrá alumnos que podrán hacer la tarea en poco tiempo y otros requerirán mucho más, en función de su competencia. ¿Cómo hacemos entonces para calcular el número de ECTS de una asignatura? ¿Y para calcular el tiempo que tarda un estudiante en realizar cada tarea?

En primer lugar, el número de ECTS de una asignatura debe venir determinado por los responsables del plan de estudios. Este debe ser, por tanto, un dato de entrada en el proceso de diseño de la asignatura. Mientras no se tomen ese tipo de decisiones, un planteamiento que suele funcionar bien es asumir que los alumnos deben dedicar una hora de tiempo adicional por cada hora de clase del esquema horario actual, basado en créditos LRU. Esto nos lleva a una sencilla fórmula para pasar de créditos LRU a ECTS: créditos_ECTS = 0.8 x créditos_LRU.

Dicho esto y conocidos los créditos ECTS de nuestra asignatura, nuestro trabajo como profesores es llenar de actividades significativas todo el tiempo correspondiente a los ECTS asignados a nuestra asignatura. Por ello, es muy importante hacer la estimación del tiempo de cada tarea del programa de actividades. De esto tenemos cierta experiencia ya que siempre hemos calculado con bastante precisión el tiempo que

dejamos a los estudiantes para realizar las preguntas y los problemas de los exámenes.

Efectivamente, sabemos que la varianza en el tiempo que requieren los alumnos para hacer una tarea puede ser elevada, especialmente si el profesor no ejerce control sobre ese proceso. Con frecuencia, hay alumnos que se encallan en una tarea y acaban dedicando muchas horas de forma totalmente improductiva, con grave perjuicio para otras asignaturas. Esta situación acaba volviéndose en contra del profesor, que recibe quejas de los alumnos por una presunta mala estimación del tiempo de dedicación, o de profesores de otras asignaturas del mismo curso, que lo acusan de acaparar un excesivo tiempo de dedicación de los alumnos.

Nosotros proponemos ejercer un control sobre la situación, de la forma siguiente: en las instrucciones del curso debe quedar muy claro que los alumnos deben dedicar a cada tarea el tiempo que se ha previsto para ella, pero no más. Si en ese tiempo la tarea no se ha acabado, entonces el alumno debe identificar sus dudas y prepararse para exponerlas a sus compañeros o al profesor, al inicio de la clase siguiente. A la vista de esas dudas, el profesor puede decidir que el alumno debe dedicar más tiempo a la tarea, pero ahora la situación está bajo su control. Obviamente, es necesario ceder un cierto tiempo de clase para que los alumnos puedan resolver las dudas, quizá con una pequeña reunión de grupo o con la ayuda del profesor.

Este planteamiento tiene varias ventajas adicionales. Por un lado, el alumno se hace más consciente de su proceso de aprendizaje y aprende a identificar sus dudas y a poner remedio a tiempo. Por otro lado, estamos enseñando al alumno a adoptar una actitud muy propia del ingeniero: hacer el mejor trabajo posible en el tiempo que nos han dado para ello o que nosotros hemos planificado para hacerlo.

Como complemento a esta medida, es importante que el profesor recoja sistemáticamente datos sobre el tiempo de dedicación de sus alumnos. El procedimiento puede ser muy simple: hacer circular una hoja con la lista de alumnos de clase, de forma que cada alumno apunte junto a su nombre el tiempo que ha dedicado fuera de clase a la asignatura. En nuestra experiencia, para que los datos sean fiables, conviene hacer esta recogida con frecuencia (por ejemplo, cada semana). Otro procedimiento de recogida del tiempo dedicado a cada actividad consiste en preguntarlo en cada entrega (ya que cada actividad, o una secuencia de ellas, suele terminar con un entrega). Sin embargo, puesto que las normas no se aplicarán de forma completamente estricta, los datos recogidos pueden poner de manifiesto problemas de mala estimación, o situaciones de dedicación fuera de lo normal en algún alumno o grupo de alumnos.

Por último, esa insistencia de estar en condiciones de reportar sobre cómo se ha usado el propio tiempo tiene un cierto valor pedagógico, puesto que ayuda a inculcar a los alumnos dos ideas básicas: que aprender requiere tiempo (y esfuerzo) y que el tiempo es un recurso muy valioso, y que hay que administrar con sabiduría.

4.2. Trabajo en grupo

¿Y qué pasa si un alumno se niega a trabajar en grupo, argumentando que se le da mejor trabajar solo?

Lo primero que hay que decir es que éste no es un problema habitual. Al *contrario*, a los alumnos les gusta trabajar en grupo (el trabajo en grupo siempre aparece como uno de los elementos más valorados por los alumnos en las encuestas de satisfacción).

En todo caso, podemos imaginar dos situaciones posibles. Por una parte, es posible que el trabajo en grupo sea una estrategia de apoyo al aprendizaje, un recurso más. En este caso, quizá no sea difícil encontrar una manera en que los alumnos que no quieren trabajar en grupo puedan hacerlo de forma individual.

El segundo escenario es aquel en el que el trabajo en grupo no es sólo una estrategia docente, sino un objetivo de *aprendizaje* del curso. Después de todo, trabajar en grupo es una de esas competencias transversales que demandan los empleadores y que debemos ir incluyendo entre los objetivos formativos de nuestras titulaciones. En esta situación, el trabajo en grupo simplemente no es negociable. Es como si el alumno nos hubiese dicho: "*A mi la Transformada de Fourier no me va mucho. Por favor, no me examines de ese tema*".

Si nuestro escenario es el segundo, asegurémonos de que en nuestra lista de objetivos formativos está el aprender a trabajar en grupo y

busquemos la complicidad de los responsables académicos del plan de estudios al que pertenece nuestra asignatura.

5. Conclusión

Muchas han sido las preguntas y también las respuestas. Pero aun suponiendo que hemos convencido a un lector escéptico o que hemos aclarado algunas dudas a un lector ya convencido, muchos profesores se sienten abrumados por lo que se les viene encima, y entonces hacen la última pregunta: *¿Cómo podemos enfrentarnos a este reto sin morir en el intento?*

Pues la mejor respuesta a esto es que comencemos ya mismo a cambiar cosas. Desde luego, lo que no es recomendable es cambiarlo todo de golpe. Conviene ir haciendo cambios poco a poco.

Una buena manera de empezar, por ejemplo, es elegir un tema del curso que esté bien documentado y sustituir las clases expositivas en las que se desarrolla habitualmente el tema, por

- una tarea previa en la que el alumno debe trabajar con la documentación en casa y
- una actividad en clase, en pequeños grupos, para aclarar las dudas del tema y hacer ejercicios de profundización.

Otra posibilidad para dar un primer paso consiste en transformar las colecciones de problemas, que seguramente ya tenemos preparadas, en actividades y entregas. Por ejemplo, cada tema puede tener una colección de problemas que los estudiantes deben realizar primero individualmente en casa y después, en grupo durante la clase o fuera de ella, pueden reunirse para aclarar dudas y comprobar que lo han hecho bien. El resultado de esta reunión de grupo puede ser una entrega. Después se puede hacer una prueba individual con ejercicios muy parecidos a los de la entrega. Para fomentar la ayuda mutua en el aprendizaje de todos los miembros del grupo se puede puntuar a todos con la nota del que obtuvo la peor calificación. O, en plan más positivo, si todos los miembros del grupo sacan más de un 6, por ejemplo, se sube un punto a cada uno de ellos.

También se pueden ir introduciendo en clase técnicas de aprendizaje cooperativo, sin cambiar excesivamente toda la estructura de la asignatura.

Lo que hay que tener claro es que al principio, cuando introducimos estos pequeños cambios, lo que en realidad se pone a prueba no es tanto el método, sino a nosotros mismos. Se pone a prueba nuestra capacidad y confianza para hacer las cosas de otra manera y para justificar ante los alumnos el porqué de estos cambios. Por parte de los alumnos, esos pequeños cambios no van a suponer una gran alteración de sus planteamientos, e incluso serán vistos como interesantes para romper la monotonía.

En realidad, hasta que hayamos cambiado al menos un 30% del programa, y el correspondiente 30% del método de evaluación, no apreciaremos un cambio serio de actitud por parte de los alumnos. Y también hay que tener claro que ese cambio de actitud puede incluir inicialmente un cierto rechazo a los nuevos métodos por parte de los alumnos, en cuanto se ponga de manifiesto que deben trabajar más horas, en casa, y estarán sometidos a un régimen estricto de entregas. Pero también hay que tener claro que las experiencias previas indican que persistiendo en el método y mejorando los aspectos que no funcionan bien se acaban obteniendo mejores resultados y tanto los profesores como los estudiantes se muestran más satisfechos. Por último decir que aunque todas las técnicas actúan positivamente de forma individual, cuando se aplican varias de ellas de forma coordinada y coherente su efecto positivo se multiplica.

Referencias

[1] Juan J. Navarro, Miguel Valero-García, Fermín Sanchez y Jordi Tubella, Formulación de los objetivos de una asignatura en tres niveles jerárquicos, JENUI 2000, pag. 457

[2] B.S. Bloom et al, *Taxonomy of Educational Objectives: Handbook I, Cognitive Domain.* Nueva York: David McKay, 1956.

[3] Johnson, D. W., R. T. Johnson, and K. A. Smith. (1991). *Active learning: Cooperation in the college classroom*. Edina, MN: Interaction Book Company.

[4] Project Based Learning, a guide to Standard-focused project based learning for middle and high school teachers. *Buck institute for education*, 18 Commercial Boulevard, Novato, California 94949.

La agenda de aprendizaje como herramienta de adaptación a la metodología educativa en ECTS del EEES

Lluís Ribas Xirgo, A. Josep Velasco González
Dpt. Microelectrónica i Sistemes Electrònics
Universitat Autònoma de Barcelona
ETSE, Campus UAB, 08193 Bellaterra
{Lluis.Ribas, Josep.Velasco}@uab.cat

Resumen

En la adaptación al nuevo modelo educativo en ECTS es necesario que el alumno cambie de actitud ya que pasa a ser el actor central del proceso de aprendizaje. Para favorecer esta transformación se propone presentar el plan docente de las asignaturas en ECTS en forma de agenda de actividades de aprendizaje. Así, el alumno toma conciencia del camino que debe seguir para lograr los objetivos docentes de cada asignatura y es capaz de observar su evolución. Con ello, también se fomenta el desarrollo de la habilidad de gestionarse el tiempo y la competencia en la asunción de responsabilidades. En este trabajo se presenta un método para la generación de dichas agendas a partir de los planes docentes de las asignaturas. Las experiencias desarrolladas hasta el momento han sido positivas.

1. Motivación

La introducción de nuevas metodologías docentes a raíz de la implantación del sistema de ECTS en el ámbito del EEES [1, 2] ha supuesto un importante trastorno tanto para profesores como para alumnos.

En este sentido, la experiencia recogida tras los cuatro primeros semestres de aplicación del Plan Piloto del DURSI [3] en la titulación de Ingeniería Informática ha servido para confirmar la importancia de la elaboración de los planes docentes de las asignaturas tanto por lo que hace referencia a la coordinación entre profesores como, sobre todo, por lo que implica para los alumnos. (El DURSI es el departamento con competencias para las universidades de la Generalitat de Catalunya que propuso la ejecución de un plan piloto, que se inició en el 2005, para la adaptación de las titulaciones al EEES.)

Así pues, la adopción de un formato común de los planes docentes ha favorecido la coordinación entre varias asignaturas de unos mismos estudios. Por otra parte, también ha facilitado su comprensión por parte de los alumnos y, consecuentemente, ha conseguido que se adaptaran de forma menos traumática.

De todas maneras, la adecuación de los planes docentes a ECTS se podría completar con una mejor información de los mismos a los alumnos. Además de ponerlos a su disposición, podrían servir de base para la elaboración de unas *agendas de aprendizaje*.

Una agenda de aprendizaje es una lista de *actividades de aprendizaje* ordenada en el tiempo que puede visualizarse de distintas maneras. Una actividad de aprendizaje es una tarea que debe realizar el alumno para conseguir un determinado objetivo parcial en su aprendizaje. Puede ser de carácter presencial, como asistir a una clase magistral, o no presencial, como la resolución de un determinado problema.

Estas agendas servirían para presentar a los alumnos el camino de actividades que tienen que hacer para lograr los objetivos fijados en el plan docente correspondiente. (Como curiosidad, es interesante saber que, con un nombre parecido, la Universidad de Santander en Colombia dispone de una herramienta dentro de su campus virtual que denomina "Agenda de Avance de Aprendizaje" [4] y que se corresponde a lo que sería una posible visualización de una agenda.)

En cualquier caso, la finalidad de este trabajo es formalizar una agenda de aprendizaje cuyos contenidos puedan visualizarse adecuadamente y cuya formalización permita la sistematización de los procesos que se involucran en su generación y visualización. (Cabe hacer notar que hay algunos trabajos interesantes en el establecimiento de caminos de aprendizaje a través de unidades de

conocimiento [5] cuyos métodos podrían ser adaptados para las agendas.)

Evidentemente, el uso de estas agendas facilitaría que los alumnos se hiciesen responsables de su propio aprendizaje. En este sentido, la idea es similar a la del *project-based learning* (PBL) [6, 7], donde se intenta que el alumno asuma la responsabilidad de realizar el proyecto que se le propone a partir de la base que el reto lo motiva y que para su consecución tendrá que hacer el aprendizaje de los elementos previstos en el plan docente correspondiente. Se diferencia, sin embargo, porque no se trata de un método docente sino de un complemento en que tiene que servir de guía de aprendizaje para el alumno en asignaturas organizadas en ECTS.

El trabajo que se presenta, pues, se orienta a determinar una metodología para la generación de estas agendas de aprendizaje para el alumno.

Por cuestiones de complejidad, se ha trabajado sólo con un tipo de asignaturas. En este caso, las de las titulaciones de Ingeniería Informática (II) e Ingeniería Técnica Informática (ITI), en las dos especialidades, de Gestión y de Sistemas. De todos modos, los resultados se pueden extender con facilidad hacia cualquier otro tipo de asignatura e, incluso, aplicarse para un curso completo.

El procedimiento que se ha establecido aprovecha el modelo de programa docente que se usa en II a raíz de la aplicación del Plan Piloto para la adaptación al EEES. Las actividades que se describen son tareas que ha de hacer el alumno para completar su proyecto de aprendizaje. Cada tarea tiene una duración prevista y un conjunto de recursos asociado

1.1. Objetivos

El objetivo principal del proyecto en el que se enmarca este trabajo es el apoyo al alumno para que asuma un rol activo en su aprendizaje.

Para ello se le proporcionará una agenda en la que verá reflejadas todas las actividades que debe hacer para lograr el conocimiento, las competencias y las habilidades que se le piden. Se tendrá en cuenta especialmente que se dé cuenta del trabajo que debe de realizar con independencia de aquellas actividades presenciales programadas como clases y sesiones de laboratorio, a las que ya está habituado.

De esta manera, se pretenden los objetivos siguientes: que el estudiante desarrolle la habilidad de la autogestión del tiempo (en especial, para las actividades no presenciales) y que adquiera una cierta competencia en la asunción de responsabilidades (por ejemplo, en el caso del cumplimiento de los plazos de entrega).

Uno de los mecanismos que se proponen para conseguir los objetivos parciales ya mencionados es el empleo de las agendas de aprendizaje. Éstas se pueden generar a partir de la información de los programas docentes en ECTS.

Así pues,

- se establecerá un modelo de plan docente que se adecue a las necesidades de las agendas de aprendizaje y que, además, sea compatible con el modelo actual en ECTS,
- se hará una clasificación sistemática de las diferentes actividades que un estudiante puede llevar a término en vistas a facilitar el proceso de preparación de la agenda,
- se diseñará un procedimiento para la elaboración de una agenda de aprendizaje tipo, y
- se preparará un sistema de información para los alumnos de forma que tengan un acceso fácil.

Hay que tener en cuenta, no obstante, que la complejidad de esta tarea obliga a limitar el alcance del trabajo a una asignatura, de forma que se pueda hacer una prueba piloto con los alumnos para corregir todas las deficiencias que se puedan encontrar antes de extender el sistema a todas las asignaturas y para todos los alumnos.

1.2. Organización del artículo

En el apartado siguiente se hace una revisión de los trabajos previos al presente y que también lo son a la introducción del plan piloto para la adaptación de las titulaciones al EEES.

En el proceso de implantación de este plan se instauró un modelo para los planes docentes de las asignaturas. Este modelo se revisa en la sección 3.

En relación al ámbito de este trabajo vale la pena hacer notar que se trata como sinónimos plan, programa y guía docente. Los tres deben de contener, como mínimo, una descripción implícita o explícita de las actividades que hay que llevar a cabo para cumplir con los objetivos de aprendizaje propuestos.

La sección 4 se dedica a discutir el proceso de generación de las agendas de aprendizaje. En la siguiente se hace una reflexión sobre los posibles métodos de automatización que pueden emplearse para su implantación.

Finalmente, la última sección se dedica a presentar los resultados previos obtenidos y las conclusiones de este trabajo.

2. Antecedentes

El proyecto en el que se enmarca este trabajo empezó a desarrollarse formalmente el curso 05/06, pero tiene precedentes que se remontan al 2004. La idea previa era la de proporcionar a los estudiantes información detallada de todas las actividades presenciales que implicaba el seguimiento de una determinada asignatura, como ya se hacía en otras universidades.

En este sentido, ya se habían hecho experimentos en los que se daba a los estudiantes un calendario (normalmente, accesible vía web) en el que podían ver todas las actividades, incluidas las de entrega de evidencias.

Los calendarios de actividades, no obstante, ya son una cosa habitual en los campus virtuales de la mayoría de universidades. Por ejemplo, el campus virtual de la UAB [8] dispone de uno en el que se pueden situar todas las actividades de cada asignatura. (Desgraciadamente no se usan en demasiadas asignaturas, especialmente si se trata de asignaturas de carácter presencial.)

En el campus virtual de la UOC [9], los calendarios de actividades son una parte fundamental de la planificación docente y, como tal, todas las asignaturas lo tienen. (El alumno pueda visualizar el calendario individual para todas las asignaturas.)

Así pues, la coincidencia de dos experiencias docentes diferentes (estudios presenciales a la UAB y virtuales a la UOC) hizo que se planteara la idea de dar soporte a los estudiantes presenciales elaborando estos calendarios y poniéndolos a su disposición con los medios que había.

2.1. Agendas de actividades presenciales

Desde el año 2004, en la asignatura de Codiseño HW/SW (II) y en la de Diseño de sistemas electrónicos (de la titulación de Ingeniería Electrónica) se presenta la guía docente a los alumnos de manera que incluye un calendario de todas las actividades presenciales.

El hecho de presentar el calendario junto con la guía de actividades (cada tema, por ejemplo, junto con la fecha, hora y lugar de las clases y laboratorios en los que se trata) permite al alumno observar cómo se ha previsto el desarrollo del plan docente y qué trabajo habrá de hacer.

Aun así, estas experiencias se realizaron en asignaturas de segundo ciclo y con pocos alumnos, por lo que el impacto de esta forma de presentar la agenda de actividades presenciales es de difícil medida. En todo caso, el seguimiento de las asignaturas por parte de los alumnos ha sido más fluido que antes de adoptar este sistema. El número de consultas en relación con el programa docente ha sido prácticamente nulo.

De hecho, este fue un indicio que marcó la posibilidad de desarrollar un trabajo parecido para las asignaturas que debían de adaptarse a ECTS, especialmente aquellas que se impartían a los alumnos de primer curso.

2.2. Agendas en estudios no presenciales

En los estudios no presenciales se cuenta, habitualmente, con una guía docente y un calendario de actividades que el alumno debe de consultar para hacer un seguimiento correcto de la asignatura.

En este sentido, se puede ver el calendario como una vista particular de la agenda de actividades de aprendizaje. Sin embargo, adolece de una relación directa con el contenido de la guía docente.

En general, los calendarios permiten a un alumno ver todas las actividades pendientes en todas las asignaturas en las que participa, pero no su evolución en cada una de ellas ni, mucho menos, ofrecen una visión global de las mismas.

Aun así, lo interesante es poner de manifiesto al alumno el trabajo a desarrollar. Así pues, en una asignatura del Master en Software Libre de la UOC ya se introdujo el concepto de agenda de aprendizaje adaptada a los alumnos: debían programar una pequeña aplicación que, a partir de un programa de estudio y de los horarios que ellos podían dedicar al mismo, generara una agenda de actividades que les mostrara el camino para lograr los objetivos de aprendizaje de la asignatura. Esta experiencia se ha ido repitiendo en los cursos posteriores.

2.3. Adaptación a programas en ECTS

En los planes docentes en ECTS es necesario tener en cuenta no sólo el tiempo del alumno invertido en las actividades presenciales sino también en las no presenciales.

Con esta experiencia previa, que combina los dos tipos de planificación (presencial y no presencial) y la puesta en marcha del Plan DURSI para las titulaciones universitarias en el marco del EEES se vio interesante de continuar con la línea seguida para transmitir al alumno la idea de que el responsable de su aprendizaje era él mismo.

3. Modelo de los planes docentes

Por conveniencia para los fines de este trabajo, se ha adoptado el modelo del plan docente de las asignaturas de la titulación de Informática en la que se distinguen, entre otras, los apartados siguientes:

- *Temario detallado*. Cada entrada incluye una descripción del tema y una lista de las actividades por desarrollarlo, con la duración en horas de cada una.
- *Ciclo semanal de aprendizaje*. Un esquema de la distribución horaria recomendada por semana que incluye las horas en actividades presenciales y no presenciales. Como que las de las primeras se hacen durante los horarios de clase/laboratorio/tutorías de la asignatura, se pueden considerar regulares y fijas para todo el semestre. Las actividades no presenciales pueden ser pautadas, pero podrán tener más variaciones semana a semana, según los diversos plazos de presentación de las memorias correspondientes. En la plantilla propuesta hay, sin embargo, una pauta semanal media. Cosa que permite al estudiante tener una idea genérica de la dedicación que deberá tener semanalmente a la asignatura.
- *Evaluación*. Descripción del sistema que se usará para evaluar el aprendizaje. Evidentemente, se priorizan los métodos de evaluación continua respecto de los otros, con lo que se confiere más importancia al hecho de seguir adecuadamente el plan docente previsto.
- *Relación del material de apoyo al aprendizaje*. Incluye referencias bibliográficas clasificadas y herramientas y material de consulta en formato digital (probablemente accesible vía web).
- *Profesores*. Cada uno en relación a las actividades de las que se hace cargo. De esta manera, los alumnos saben a quién dirigirse para resolver sus dudas.

En este modelo sería conveniente añadir actividades de evaluación, que incluyen entregas en puntos determinados del proceso de aprendizaje. Al tratarse de una clase especial de actividades puede ser difícil de incluir dentro del temario. Ahora bien, sería conveniente de ponerlo en el apartado de evaluación en relación al temario, de forma que se pudiera captar fácilmente la vinculación entre un determinado tramo de aprendizaje y la actividad dedicada a evaluarlo.

La dificultad para establecer una forma estándar para hacerlo (hay actividades de laboratorio que son evaluadas, entregas de evidencias no vinculadas directamente con el aprendizaje de un tema, exámenes, etcétera) ha hecho que tampoco se proponga ningún modelo concreto.

3.1. Clasificación de las actividades

Las actividades asociadas al aprendizaje de un tema se pueden dividir entre presenciales y no presenciales.

En las presenciales hay las más convencionales, que son las clases magistrales (teoría), los seminarios (práctica, habitualmente resolución de problemas) y las sesiones de laboratorio (práctica). Hace falta añadir también tutorías y exámenes.

Entre las actividades no presenciales hay las de estudio, resolución de problemas, preparación de seminarios y prácticas, desarrollo de proyectos y elaboración de documentos.

3.2. Recursos docentes

Es necesario tener presente que las actividades de aprendizaje, sean o no presenciales, requieren de unos recursos para poderse llevar a cabo.

En las no presenciales, la mayoría de los recursos pueden ofrecerse al alumno a través de Internet o consisten en referencias a material del que puede disponer con facilidad.

En las actividades presenciales, estos recursos incluyen especialmente tanto horarios como lugares (aulas, laboratorios, seminarios, etcétera).

Por lo que atañe a la guía docente, es necesario que se detallen todos. Respecto de la agenda de aprendizaje, los elementos clave son horarios y lugares.

3.3. Adaptación de los planes docentes para la generación de agendas

Aunque la información presente en el modelo de guía docente es básica para la confección de las agendas de actividades de aprendizaje, es necesario tener presente que no suelen incluir relaciones explícitas entre actividades.

Generalmente, se parte de que las actividades se desarrollan de forma secuencial, tal como aparecen en el plan docente y con una precedencia implícita entre ellas. (Por ejemplo, una clase magistral se debe de hacer antes de una de resolución de problemas y ésta antes de alguna sesión de laboratorio.)

Sin embargo, esto no tiene porqué estar programado así en la realidad (piénsese en que, por algún motivo, haya dos clases magistrales sucesivas) y tampoco resuelve los vínculos con el proceso de evaluación.

Así pues, será imprescindible determinar estas relaciones entre actividades de forma explícita para poder proceder tanto a la confección de un calendario de la asignatura como a la de una agenda.

4. Procedimiento para la obtención de las agendas de aprendizaje

Para la confección de las agendas de actividades de aprendizaje se ha optado por hacer una correspondencia entre plan docente y *proyecto*.

De hecho, el plan docente no es más que la planificación del desarrollo de un proyecto el objetivo del cual es, precisamente, lograr los objetivos de aprendizaje que se determinan en una asignatura dada.

En este sentido, la idea es que el proyecto docente de una asignatura se transfiera al alumno como "un proyecto de aprendizaje" a través de una agenda individualizada.

En cuanto a la correspondencia entre los elementos de un proyecto docente y un proyecto genérico (o de gestión, si se quiere) cada actividad de aprendizaje se hace equivaler a una tarea. Las tareas llevan asociadas diversos atributos como la descripción, el tiempo que requieren y, muy especialmente, una serie de recursos necesarios por poderlas llevar a término.

Hay tantas clases de tareas como clases de actividades. La distinción, no obstante, se hace a través de alguno de los recursos que necesitan: los seminarios, por ejemplo, necesitan disponer de un recurso que se puede denominar "aula de seminario" y que incluye el lugar físico dónde se realizará y el horario en qué se puede hacer. Otros recursos son los profesores, de forma que el alumno siempre tenga conocimiento del responsable de una determinada actividad, y los mismos alumnos.

Al módulo gestor de proyectos también se le debe proporcionar el calendario académico de forma que pueda sincronizar el proyecto docente con los días y horas del semestre en qué se imparte la asignatura. (En una primera versión los alumnos no podrán añadir restricciones horarias ni a las tareas no presenciales.)

5. Automatización de la generación de las agendas de aprendizaje

El hecho de hacer la correspondencia entre un proyecto de aprendizaje y un proyecto genérico responde a la conveniencia de usar alguna aplicación ya existente para la gestión de proyectos, como MS Project [10] o GanttProject [11]. Sin embargo, aunque estas aplicaciones permitan la generación de la agenda, no son adecuadas para presentar contenidos.

Así pues, para garantizar unos mínimos de comunicación de la información a los alumnos y que sea realmente útil, sería conveniente contar con algún gestor de contenidos (CMS, según las siglas en inglés) que permita a sus usuarios diferentes visualizaciones. De esta manera, se permite que determinados usuarios (los profesores) puedan intervenir en el proyecto docente y que otros (los alumnos) puedan añadir restricciones o condicionamientos de carácter personal para ver la planificación individual que les permitan lograr los objetivos fijados.

Sin embargo, entre los diferentes CMS de software libre (sólo se examinaron éstos para posibilitar una distribución más abierta de los

resultados de este trabajo), sólo eGroupWare [12] dispone de un módulo de gestión de proyectos.

Desgraciadamente, la versión del módulo de gestión de proyectos es inestable (no funciona igual en todas las versiones del CMS) e incompleta. Aun así, se hicieron diversos experimentos de cara a la generación de agendas, pero todos produjeron resultados insatisfactorios.

Así pues, para cumplir con los dos aspectos (automatización de la generación de las agendas y posibilidad de visualizaciones adaptadas a su funcionalidad) lo más conveniente es emplear algún entorno de aprendizaje virtual (VLE en sus siglas en inglés) como Moodle [13] o Dokeos [14] y añadir un módulo adaptado para la generación de agendas.

El módulo de generación de agendas de aprendizaje se ocupará de sincronizar el calendario académico con las actividades de aprendizaje que se describan en el plan docente de acuerdo con las restricciones que impongan los recursos de que se dispongan (horarios, espacios, profesores, etcétera).

6. Resultados y conclusión

Los objetivos perseguidos en este trabajo están incluidos en el campo de las metodologías docentes y, más concretamente, en el aspecto de la planificación docente.

Dado que la motivación principal ha sido la de contribuir a una mejor adaptación y aplicación de los planes docentes en ECTS; es decir, centrados en el estudiante, la aplicación de los resultados de este trabajo se extiende a todas aquellas titulaciones que ya se imparten según criterios de ECTS. La limitación, en todo caso, radica en el hecho que las agendas de aprendizaje se han inspirado en un modelo docente aplicado a estudios de ingeniería. Cosa que puede hacer que haya elementos, como por ejemplo la clasificación de las actividades, que se hayan de adaptar.

En este trabajo se pretendía impulsar la concienciación del alumno en el rol que ha de asumir en el modelo docente en ECTS. El objetivo, por una parte, era el de ofrecerle apoyo para la adquisición de la habilidad de autogestionarse el tiempo y, por otra parte, hacerle evidente el compromiso que tiene con el proceso de aprendizaje de forma que desarrollara una cierta competencia en la asunción de responsabilidades.

El mecanismo para lograr estos objetivos ha sido el de transmitir el plan docente de las asignaturas en forma de agenda de actividades de aprendizaje. Para hacer esta transformación, primero se ha analizado el modelo de planes docentes que se ha establecido en la aplicación del Plan DURSI para las titulaciones en el marco del EEES y, después, se ha determinado una correspondencia entre los diversos elementos del plan docente con los de un proyecto de cariz genérico que, en este ámbito, acaba siendo un proyecto educativo.

Aunque se contaba con experiencia previa en trabajos similares, el primer semestre del curso 06/07 ha sido el primero en el que se ha confeccionado una agenda de aprendizaje que sigue los principios aquí citados. De forma manual, aunque siguiendo el procedimiento mencionado en la sección cuarta de este artículo, se preparó la agenda de aprendizaje de la asignatura de Fundamentos de computadores de la titulación de ITI.

La agenda no incluye información de las actividades no presenciales puesto que dependen de cada alumno en particular y, al carecer de herramientas para su automatización, no era factible ofrecerles esta posibilidad.

La vista de la agenda es una tabla por semanas (véase tabla 1) de las actividades presenciales que los alumnos deben de realizar, con fechas y horas por grupos. La columna de los días sólo refleja los días de teoría (primer número) y de seminario de problemas (segunda cifra). Las horas de dichas actividades así como las horas y los días de los de las de laboratorio se han ocultado por claridad.

Para que los estudiantes tuviesen que mirarla (y poder tener, con ello, un elemento más para conseguir que asumieran un cierto rol activo en su aprendizaje), esta misma tabla contiene enlaces al material que debían de emplear en cada actividad. Generalmente, para las clases de teoría se trataba de un enlace a las “transparencias” de clase, para los seminarios se trataba de un enlace al problemario correspondiente y para las sesiones de laboratorio, de la documentación necesaria para el trabajo con simuladores y para el desarrollo del proyecto de la asignatura.

Semana		Días	Tema	Seminario	Laboratorio	Proyecto
			…			
5	16/10	M: 17, 19; T: 11, 17!	Funciones lógicas	Problemas		
6	23/10	M: 24, 26; T: 18!, 24	Álgebra de Boole	Problemas	Herramientas	Presentación
7	30/10	M: 31, 2; T: 25, 31!	Optimización de funciones	Problemas		
8	06/11	M: 7, 9; T: 8, 14!	Circuitos combinacionales	Problemas	Manipulación de funciones	Parte 1
9	13/11	M: 14, 16; T: 15, 28!	Módulos combinacionales	Problemas	Circuitos con puertas lógicas	Parte 2
			…			
14	18/12	M: 19, 21; T: 20, 12!	Arquitectura de Von Neumann	Procesador educativo	Circuitos secuenciales	Parte 5
15	08/01	M: 9, 11; T: 10, 9	Organización de la memoria	Problemas	Lenguaje máquina	Parte 6
16	15/01	M: 16, 18; T: 17, 16	Comunicaciones	Periféricos	Ensamblador	Parte 7
17	22/01		Enunciados y soluciones de exámenes de convocatorias anteriores			
18	29/01	Examen: 02/02	Enunciado/solución			
			…			

Tabla 1. Vista simplificada de la agenda de la asignatura (porción)

En la tabla 2 se resume el número de visitas por alumno que la agenda recibió. La media está en 47 accesos, con una desviación típica de 30. (Se han descartado aquellos alumnos no presentados.) Esto es, la mayoría la consultó una media de tres veces por semana, con una variación que puede oscilar entre una vez y cinco veces, cosa razonable.

Intervalo	Alumnos	
#accesos	Número	Porcentaje
0 – 15	31	14
16 – 30	51	23
31 – 45	38	17
46 – 60	42	19
61 – 75	23	11
76 – 90	14	7
> 90	20	9

Tabla 2. Resumen estadístico del acceso a la agenda de aprendizaje

La buena acogida que ha tenido esta primera experiencia nos anima a seguir con el proceso de automatización de la generación de este tipo de agendas. En una primera fase se tratarán de obtener las que atañen a las actividades presenciales. La segunda y última fase incluirá aquellas no presenciales que dependen de cada alumno. (Para la generación de agendas individualizadas es necesario crear un entorno en el que el alumno pueda introducir sus propias restricciones, cosa que la complica.)

En todo caso, se espera que para el curso 07/08 ya se disponga de una versión completa, cosa que permitirá generar agendas de aprendizaje individualizadas y una adaptación dinámica a las incidencias que puedan ocurrir durante el desarrollo de una asignatura.

Agradecimientos

El trabajo ha sido financiado por los proyectos de mejora de la calidad docente siguientes:

- "A4/ECTS: Herramienta básica para el desarrollo de la competencia de la autogestión del tiempo del alumno en el marco del EEES (A4 = Autogestión de la agenda de aprendizaje del alumno)", Ref. PID2005-13. Financiado por la UAB. Curso 2005/2006.
- "Individualización de la programación docente en Fundamentos de computadores", Ref. MQD2006-105. Financiado por AGAUR, Generalitat de Catalunya. 2006–2008.

La formalización del modelo de los planes docentes es, en gran parte, debida a la implantación del Plan Piloto del DURSI y a las distintas comisiones y grupos de trabajo que a tal efecto se crearon. Por ello, se agradece a sus miembros la buena disposición para con los participantes de este trabajo.

Referencias

[1] Real Decreto 1044/2003, "por el que se establece el procedimiento para la expedición por las universidades del Suplemento Europeo al Título". BOE núm. 218, del 11 de septiembre de 2003.

[2] Real Decreto 1125/2003, "por el que se establece el sistema europeo de créditos y el sistema de calificaciones en las titulaciones universitarias de carácter oficial y validez en todo el territorio nacional". BOE núm. 224, del 18 de septiembre de 2003.

[3] "Pla pilot d'adaptació de titulacions a l'espai europeu d'educació superior", Departament d'Educació i Universitats, 14 de enero de 2006. [http://www10.gencat.net/dursi/ca/un/eees_pla_pilot.htm]

[4] "Agenda de Avance de Aprendizaje", Campus Virtual UDES, Univ. Santander, Bucaramanga, Colombia. [www.cvudes.edu.co/AulaVirtual/recursos.aspx]

[5] Zhao, C. and Wan, L. "A Shortest Learning Path Selection Algorithm in E-Learning." In *Proc. Sixth IEEE Int'l. Conf. on Advanced Learning Technologies* (July 05 – 07, 2006). ICALT. IEEE Computer Society, Washington, DC, 94-95.

[6] "Introduction to project-based learning", Project-based handbook, BIE, 05/06/2006. [http://www.bie.org/pbl/pblhandbook]

[7] "Project-based learning", Instructional Module, The George Lucas Educational Foundation, 2003. [web: http://www.glef.org]

[8] Campus virtual, OAID, UAB. [web: https://cv.uab.es/cv/entrada.jsp]

[9] Demostración del campus virtual, UOC. [http://www.uoc.edu/presentacions/campus/]

[10] "MS Project", en *Project Management Software Review*. Top Ten Reviews. 2007. [http://toptenreviews.com/]

[11] D. Popov, "A quick look at GanttProject 2.0", 17/03/2006. [www.ganttproject.org]

[12] R. Becker, Projectmanager, eGroupWare, [www.egroupware.org]

[13] Course Management System Moodle. [www.moodle.org]

[14] Dokeos Open Source eLearning. [www.dokeos.com]

Una aproximación al Espacio Europeo de Educación Superior basada en el desarrollo de proyectos software en Ingeniería del Conocimiento.

Bertha Guijarro Berdiñas, Amparo Alonso Betanzos
Dpto. Computación
Universidad de A Coruña
Facultad Informática
Campus de Elvira, nº5. 15071 A Coruña.
{cibertha,ciamparo}@udc.es

Resumen

En esta ponencia se presenta una propuesta para la docencia en la asignatura de Ingeniería del Conocimiento de la Ingeniería en Informática. Esta propuesta supone un esfuerzo de cara a la adaptación de dicha asignatura al Espacio Europeo de Educación Superior, para la que uno de los principales problemas suele ser el elevado número de alumnos en las aulas. En este artículo se expone cómo hemos gestionado este problema para poder llevar la adaptación de la asignatura, utilizando el aprendizaje orientado a proyectos, y las ventajas e inconvenientes encontrados. Además, el sistema utilizado, con el que en general hemos obtenido resultados positivos, puede ser fácilmente extrapolable a otras asignaturas presentes en los planes de estudio de las Ingenierías Informáticas, como aquellas relacionadas con la Ingeniería del Software.

1. Introducción y motivación

En este artículo se presenta el método docente empleado en la asignatura *Ingeniería del Conocimiento* (IC). La orientación que hemos querido dar a esta asignatura viene determinada por tres cuestiones:

1. La importancia que diversas fuentes [1,10] dan a la realización de proyectos software *en grupo* en las carreras de Informática.
2. La necesidad de adaptación a las directrices de enseñanza en el marco del Espacio Europeo de Educación Superior (EEES).
3. La visión ingenieril que hemos querido dar a esta asignatura, en la que la IC se enfoca como una ingeniería que enseña a desarrollar proyectos de calidad al estilo de la Ingeniería del Software (IS).

Al intentar llevar a la práctica un proyecto docente comprometido con estos tres puntos, nos encontramos con un problema común a la mayoría de las asignaturas de las carreras de Informática en la Universidad Española: la relación del número de alumnos por profesor es, generalmente, muy elevada. En esta situación, indudablemente el tiempo dedicado a la docencia se convierte en un elemento crítico que condiciona las actividades docentes que se pueden realizar ya que, además, nos vemos obligados a atender nuestras obligaciones docentes sin por ello descuidar nuestras obligaciones en investigación. Por este motivo, es fundamental encontrar mecanismos que permitan aplicar el enfoque metodológico del EEES con una inversión en tiempo, de profesores y alumnos, acorde con los créditos asignados a las asignaturas. A lo largo de este artículo trataremos de exponer nuestra experiencia docente para mostrar cómo hemos gestionado este problema, con el ánimo de que sea útil a otros profesores.

1.1. Los proyectos software en las titulaciones de Informática

Los cursos de computación deben contener uno o más proyectos de grupo, motivados por consideraciones prácticas y pedagógicas. El uso de grupos de trabajo en la Educación Superior aporta muchos beneficios tal como está recogido en numerosos trabajos bibliográficos [5,7-11]. En concreto, los trabajos citados identifican seis cuestiones principales para forzar a los estudiantes a realizar proyectos prácticos en grupo:

1. Aplicación del conocimiento adquirido en la teoría
2. Motivación, si el proyecto elegido es el adecuado

3. Habilidades cognitivas de nivel superior, ya que los estudiantes desarrollan una mayor comprensión del material en el que trabajan, y las correspondientes habilidades de aprendizaje
4. Autonomía, los estudiantes tienen control sobre lo que aprenden y cómo lo aprenden
5. Asesoramiento, las prácticas en grupo son efectivas para distinguir entre los buenos y los malos estudiantes.
6. Ideológicas, los estudiantes están más preparados para la participación en una sociedad que, cada vez más, promociona la colaboración y la participación.

En el contexto de las Ciencias de la Computación, se obtienen además ventajas adicionales: 1) La mayoría de los proyectos de software profesionales, a los que los alumnos se incorporarán tras graduarse, implican una actividad en grupo, a la vez que existe una amplia proporción de gente atraída por el desarrollo de software que encuentra en esto una dificultad, y 2) la realización de trabajos en grupo implica la aplicación de buenas prácticas de IS. Por otra parte, también el *Computer Science Curriculum* (CSC) [1] lo encuentra "extremadamente importante" y recalca que "Aprender a trabajar en grupo no es un proceso natural para muchos estudiantes", por lo que se les debe proporcionar las herramientas para aprender a trabajar de forma efectiva como parte de un equipo, en lugar de como un programador aislado. De este modo, se establece la necesidad de desarrollar este tipo de proyectos a lo largo de la carrera, lo que normalmente significa incluir prácticas en grupo en el que distintos programadores trabajen juntos para producir un software. Además, las indicaciones de prestigio como el CSC o la British Computer Society también establecen criterios que deben satisfacer estos trabajos de grupo:

- Deben exhibir una aproximación estructurada a las prácticas, implicando un número de etapas del ciclo de vida del desarrollo de software
- Cada estudiante debe identificar claramente su contribución en el proyecto general, y la evaluación tendrá en cuenta la contribución personal de cada uno de los individuos
- El producto final debe tener calidad, fiabilidad, puntualidad en su finalización y ser mantenible
- El proyecto debe incluir un informe que describa las etapas del ciclo de vida, la verificación y la validación realizadas en cada etapa y la documentación técnica.

1.2. Las nuevas directrices de enseñanza en el EEES

La convergencia hacia el EEES implica un cambio en nuestros hábitos docentes. Entre otras cuestiones, se establecen recomendaciones referentes a las metodologías docentes a emplear y se enfatiza la necesidad de no sólo enseñar los contenidos de las asignaturas, sino de promover el aprendizaje autónomo y ciertas competencias en el alumno. Estas recomendaciones incluyen:

- Mayor implicación y autonomía del estudiante
- Metodologías docentes más activas como son trabajo en equipo, tutorías, seminarios, tecnologías multimedia, etc.
- Creación de entornos que estimulen el aprendizaje de los alumnos.

Por otro lado, el proyecto Tuning [6] desarrollado por las instituciones europeas de educación superior con el fin de enlazar este sector a los objetivos políticos de la Declaración de Bolonia, establece como primeras diez competencias a desarrollar por el alumnado las siguientes:

- Capacidad de análisis y síntesis
- Capacidad de aprender
- Resolución de problemas
- Capacidad de aplicar los conocimientos
- Capacidad de adaptarse a nuevas situaciones
- Preocupación por la calidad
- Habilidad para la gestión de la información
- Habilidad para trabajar de forma autónoma
- Trabajo en equipo
- Capacidad para organizar y planificar.

1.3. La orientación de la asignatura "Ingeniería del Conocimiento"

El contenido de los temarios en las universidades españolas para la enseñanza de la IC en la Ingeniería en Informática es muy variado. Algunos proyectos docentes presentan un enfoque orientado a profundizar en las técnicas empleadas en la IC (adquisición y representación de conocimiento, etc.). Otras veces los contenidos de esta asignatura complementan a otras relacionadas con la Inteligencia Artificial (IA) y así se centran, por ejemplo, en lógica, métodos de resolución de

problemas o aprendizaje. En una ponencia previa [3] los autores de este artículo exponíamos y justificábamos el enfoque metodológico que dábamos a esta asignatura en la Facultad de Informática de la Universidad de A Coruña (UDC). Éste es un enfoque práctico centrado en la gestión de proyectos de desarrollo de software de Sistemas Basados en Conocimiento (SBC) similar al perseguido en IS. Es decir, asumimos la definición de la IC como "La disciplina tecnológica que se centra en la aplicación de una aproximación sistemática, disciplinada y cuantificable al desarrollo, funcionamiento y mantenimiento de SBC. En otras palabras, el objetivo último de la IC es el establecimiento de metodologías que permitan abordar el desarrollo de SBC de una forma más sistemática" [13]. En concreto, en la asignatura se estudia la metodología *CommonKADS* [2,12] basada en el modelado de conocimiento y que presenta una clara tendencia convergente con las técnicas de IS, en el sentido de que es una aproximación sistemática al análisis, diseño, implementación y mantenimiento de software. No es el ánimo de este artículo, por lo tanto, discutir de nuevo sobre los contenidos de la asignatura, sino exponer el método docente empleado en el que la visión ingenieril de la asignatura hace que la parte práctica cobre una importancia especial.

2. Metodología de trabajo

"Ingeniería del Conocimiento" en la UDC es una asignatura obligatoria y cuatrimestral de 5º curso que tiene un total de 4,5 créditos, de los cuales 3 son teóricos y 1,5 prácticos (Suplemento al B.O.E. 23/11/94). En el curso 2005/06, había 148 alumnos matriculados para un único grupo de teoría, 3 grupos de prácticas y dos docentes.

2.1. Contenidos docentes y orientación de las clases presenciales de los créditos teóricos

Como se ha comentado, la parte fundamental de la docencia de esta asignatura se dedica al aprendizaje de una metodología que, al igual que las de IS, comprende unas fases estructuradas en un ciclo de gestión del proyecto de desarrollo del software basado en el modelo en espiral. En concreto, los créditos teóricos de la asignatura se articulan en 8 temas:

1. Introducción a la Ingeniería del Conocimiento
2. Metodologías para la construcción de SBC
3. Análisis de viabilidad e impacto: modelado del contexto en CommonKADS
4. Descripción conceptual del conocimiento en CommonKADS
5. Del análisis a la implementación en CommonKADS
6. Gestión de proyectos de SBC en CommonKADS
7. Técnicas para la adquisición del conocimiento
8. Evaluación de los sistemas basados en el conocimiento

Los temas 3 a 6 comprenden todos los pasos de análisis, modelado, diseño e implementación que, según la metodología explicada, deben seguirse en el desarrollo de SBC.

Es importante recalcar que, en consonancia con las nuevas directrices, durante las clases presenciales teóricas se expone sólo un núcleo básico de conocimientos que luego los alumnos tendrán que saber utilizar y ampliar durante las prácticas. Como técnica didáctica, se emplea el método de aprendizaje basado en el *estudio de casos*. En este método, se presenta a los estudiantes una situación real, se suministra cierta información, y basándose en los conocimientos adquiridos, se les pide que tomen y razonen las decisiones oportunas. En este caso, se sigue un ejemplo que corresponde a un Proyecto Fin de Carrera de un curso anterior, de forma que los alumnos pueden conocer a fondo el proyecto, ejecutar el sistema, y consultar el material que deseen. Este método de aprendizaje ayuda al estudiante a aplicar los conocimientos adquiridos en la parte de teoría (en la que se usan ejemplos parciales más pequeños), y a analizarlos dentro de un ejemplo que abarca un proyecto completo, que además les resulta temporalmente muy cercano, puesto que ellos mismos estarán realizando su proyecto fin de carrera. Dada la dificultad que supone tratar un análisis intensivo del caso con grupos tan grandes, la estrategia didáctica utilizada es la centrada en la aplicación de principios, de forma que los estudiantes se ejerciten en la aplicación y selección de las estructuras adecuadas. Esta es una de las novedades mejor valoradas en el cuestionario de evaluación que realizamos al finalizar el cuatrimestre.

2.2. Contenidos docentes y orientación de los créditos prácticos

El programa de prácticas contempla la aplicación paralela de los conocimientos adquiridos en el programa teórico. En concreto las prácticas de las asignatura se conducen siguiendo las fases de un proyecto real de análisis, diseño e implementación de una aplicación determinada en un dominio de conocimiento concreto siguiendo la metodología enseñada y con la ayuda de un entorno de desarrollo determinado, específico para este tipo de sistemas. De esta estrecha relación entre teoría y práctica se obtienen varias ventajas:

1. Los profesores de esta asignatura estamos convencidos de que dar a conocer una metodología de desarrollo de sistemas sólo de forma teórica no ayuda al alumno a comprenderla ni a asimilarla de forma perdurable ni, por supuesto, a que conozca la problemática particular asociada a la construcción de SBC y los roles que participan en un proyecto de este estilo.
2. Permite acercar la imagen de la IA a la línea principal de las Ciencias de la Computación, y evitar que sea vista como perteneciente al ámbito investigador y lejana del empresarial, al obligar al alumno a convertir las teorías explicadas en programas concretos que resuelvan problemas del mundo *real*.
3. Este método permite reducir el contenido de las clases teóricas (muy importante en una asignatura de sólo 3 créditos) ya que en ellas se expone el núcleo básico de conocimientos que luego los alumnos tendrán que saber utilizar y ampliar durante las prácticas.

De este modo, en el enfoque docente de esta asignatura las prácticas cobran una importancia especial y es en ellas donde el alumno tendrá que centrar el mayor esfuerzo. El método didáctico empleado en este caso es el *aprendizaje tutorizado orientado a proyectos* [4]. Se trata de un método de enseñanza-aprendizaje en el que los estudiantes realizan un proyecto en un tiempo determinado con el fin de abordar un problema mediante la planificación, diseño y realización de una serie de actividades, aplicando y ampliando el conocimiento adquirido. Este método pretende que los estudiantes asuman una mayor responsabilidad sobre su propio aprendizaje, así como aplicar en proyectos reales las habilidades y conocimientos adquiridos en su formación. La importancia del aprendizaje no radicará sólo en el resultado del proyecto, sino también en las competencias que se desarrollan al realizarlo. Tal como se describe en [4] esta técnica permite desarrollar numerosas competencias: las basadas en conocimientos y capacidades vinculadas al mundo profesional, a la profundización de conocimientos de la materia y habilidades técnicas, de comunicación oral y escrita, trabajo en equipo, organización del trabajo, responsabilidad personal y grupal, etc.

2.3. Organización de los grupos y el trabajo de prácticas

Este enfoque requiere una planificación temporal y una coordinación estricta con el programa de teoría [3]. El principal obstáculo en la aplicación de este enfoque lo encontramos en el elevado número de alumnos, ya que para que el método tenga éxito en su aplicación son necesarios un contacto estrecho con el alumnado y un seguimiento de los trabajos que realizan, en la línea además de las recomendaciones de los nuevos enfoques docentes de cara al EEES.

Ya el primer día de clase, se explica la orientación y organización de la asignatura, y se hace público el calendario de entregas de materiales. Dado el elevado número de alumnos, para una correcta gestión de la asignatura, es imprescindible que estas fechas sean inamovibles.

En nuestra asignatura, los trabajos de prácticas se realizan en grupos de cuatro alumnos que también se forman ya durante la primera semana. Además, una de las premisas del aprendizaje orientado a proyectos es que se deben abordar temas reales. En este caso, el SBC que construyan a lo largo del curso debe resolver un problema real de entre los habituales en IC, y que se les comentan en el *Tema 1* del programa de teoría. En dos semanas deben proponer el dominio de aplicación de su SBC, una elección que irá supervisada por el profesor. Pasado este período se publica la composición de los grupos, el código identificativo asignado y el tema elegido.

Al mismo tiempo, las tres primeras semanas de clase se utilizan para, durante las horas de prácticas, instruir a los alumnos en la herramienta de programación específica que utilizarán para la implementación del sistema. Pasado este período se comienza el desarrollo del proyecto, que seguirá la metodología *CommonKADS*, y del que

cada grupo tendrá que entregar tres informes a lo largo del cuatrimestre:

1. El *análisis de viabilidad* del SBC.
2. Los *modelos conceptuales* que suponen la especificación de requisitos de su SBC.
3. El modelo de diseño y la implementación del SBC, que seguirá las especificaciones establecidas en los modelos anteriores y las pautas explicadas en el programa de teoría.

De estos tres trabajos sólo el último es de entrega opcional, y engloban los temas 3 a 8 del programa teórico.

La aplicación del método se puede estructurar en cuatro tareas:

1. *Información*. Los estudiantes recopilan información necesaria para la resolución del problema. Para ello, previamente a cada clase de prácticas se les informa del objetivo de la misma y se les recomienda la lectura de ciertos materiales.
2. *Planificación*. Los alumnos deben elaborar un plan de trabajo para poder cumplir con el calendario de entregas establecido.
3. *Realización* del proyecto. Gracias a las etapas previas de Información y Planificación (que se realizan de forma no presencial) se aprovecha al máximo la clase de prácticas, que se dedica a revisar el plan de trabajo y discutir con cada equipo la orientación y los progresos de su proyecto. Este es un aspecto imprescindible en el aprendizaje orientado a proyectos.
4. *Evaluación*. Los estudiantes deben informar de los resultados conseguidos y poder discutirlos con el profesor. Este apartado se discutirá en mayor detalle más adelante.

2.4. Una herramienta fundamental: la plataforma de "Facultad Virtual".

Con las restricciones con las que se trabaja resulta fundamental realizar una organización y gestión temporal de la asignatura perfectas, con el fin de optimizar el tiempo que tanto alumnos como profesores le dedican. Para ello, es clave la página Web de la asignatura, construida sobre la plataforma de "Facultad Virtual" ofrecida por la UDC. Esta plataforma, con acceso restringido a profesores y alumnos, se utiliza como medio de comunicación e intercambio de documentos. En concreto:

- En ella y desde el inicio del curso, los alumnos tienen disponible el calendario con las fechas de entrega de las prácticas, y ejemplos de temáticas de prácticas realizadas en cursos anteriores.
- Semanalmente se publican los objetivos de las clases prácticas y los materiales que han de leer antes de ellas.
- Se hace disponibles a los alumnos apuntes de apoyo a las clases teóricas, así como el Proyecto Fin de Carrera que se utiliza en el estudio de casos.
- También se pueden encontrar las plantillas que deberán seguir los alumnos al entregar sus trabajos de prácticas, así como los aspectos valorables de cada uno de ellos.
- Se publican los calendarios de defensa de las prácticas y las notas.
- Permite la utilización de *tutorías virtuales*.
- Actúa como repositorio donde los alumnos depositan sus prácticas, para lo que se habilitan directorios sobre los que dejan de tener acceso una vez que la fecha de entrega ha caducado. Las ventajas de este sistema frente al tradicional, en que se entregan los trabajos directamente al profesor, son obvias: 1) se produce un ahorro considerable en el soporte, sea disquetes o papel; 2) el alumno gana en flexibilidad en cuanto al momento y lugar de entrega; 3) como consecuencia la entrega de prácticas no supone ninguna interrupción en el trabajo diario del profesor; y 4) la utilización de un método automático de recogida de prácticas evita por completo la intención de entrega fuera de plazo.

2.5. Uso de las tutorías

En este esquema de prácticas, las tutorías resultan un recurso fundamental muy utilizado por los alumnos. Éstas se utilizan desde el inicio del curso, ya que es donde los alumnos comentan sus ideas sobre posibles dominios de aplicación de su SBC. Al mismo tiempo el profesor se asegura de que el dominio finalmente elegido sea factible como práctica de la asignatura. Más tarde, las tutorías se utilizan para comentar las numerosas dudas que surgen en la elaboración de los documentos de las prácticas.

Como se ha comentado, los alumnos pueden realizar dos tipos de tutorías: virtuales y

presenciales. Las primeras, que realizan a través de la Facultad Virtual, pueden utilizarlas para realizar dudas muy concretas de respuesta rápida. Las más comunes se irán depositando en un apartado de "Preguntas Frecuentes" que deberán consultar antes de enviar una nueva pregunta. Respecto a las tutorías presenciales, dado el elevado número de alumnos, en los períodos de mayor demanda aquellos que deseen utilizar las tutorías deberán apuntarse en una lista en la que se conceden 10 minutos a cada uno de los grupos. Así evitamos que los alumnos pierdan tiempo esperando su turno, y aprenden a hacer un uso más efectivo de las tutorías al saber de antemano de cuánto tiempo disponen.

2.6. Evaluación de las prácticas y la asignatura en general

Las recomendaciones didácticas establecen que el alumno debe conocer cuanto antes los criterios bajo los que va a ser evaluado para que pueda orientar su trabajo de la forma adecuada. Por este motivo, el esquema de evaluación se hace público en la página Web desde el inicio del curso.

El trabajo realizado a lo largo del curso se refleja en la evaluación final que se hará basándose en la media de dos notas: la correspondiente a un examen escrito sobre los conocimientos teóricos, y la correspondiente a las prácticas desarrolladas a lo largo del curso. Como se puede observar, y en comparación con una docencia más tradicional, el examen teórico pierde peso, al tiempo que se simplifica su contenido al haber sido evaluada gran parte de la materia ya en las prácticas.

En cuanto a las prácticas, los alumnos tienen disponibles el porcentaje de participación en la nota final de cada uno de los trabajos que entregan, así como los aspectos que más se valoran de cada uno de ellos. Se ha comprobado que, esta última cuestión, es muy valorada entre ellos e influye muy positivamente en la calidad de los trabajos.

Además, con el fin de valorar las prácticas, cada grupo tendrá dos reuniones obligatorias con el profesor a lo largo del cuatrimestre. Estas reuniones se producen al terminar cada una de las dos fases que constituyen el grueso de las prácticas: el *análisis de viabilidad* y el *modelado conceptual*. De este modo, inmediatamente después de finalizar cada uno de los períodos de entrega de estas prácticas, se publica en la página Web de la asignatura una relación de los trabajos recibidos y un calendario donde se establece cuándo tendrá cada grupo la reunión con el profesor y cuánto durará esta. Las reuniones se realizan siempre en la semana posterior a la entrega de la práctica. Su duración se establece en 20 minutos máximo para la primera práctica y 30 para la segunda, y se realizan a lo largo de cuatro tardes. Los objetivos fundamentales de estas reuniones son proporcionar al alumno rápidamente información acerca de los errores o aspectos más sobresalientes de sus prácticas, controlar el trabajo de grupo y desarrollar su capacidad de síntesis y exposición de conocimientos. De este modo, durante las reuniones, los alumnos deberán explicar ciertos aspectos de sus prácticas a requerimiento del profesor. Además, para hacer más eficaces estas reuniones deberán llevar preparada una presentación en formato electrónico, cuyo contenido esperado pueden encontrarlo también en la página Web de la asignatura y se corresponde con algunas partes del documento previamente entregado. Los alumnos saben, por tanto, sobre qué aspectos fundamentales se discutirá en esta reunión, pero no saben cuál de los miembros del grupo tendrá que responder de qué parte de la práctica. Respecto a la corrección de la práctica, el profesor utiliza un formulario que contempla todos los aspectos puntuables de la práctica y que rellena durante la reunión. De este modo, la nota se establece, en la mayoría de los casos, tras la exposición de cada grupo. Una vez terminada la ronda de reuniones, esto es, una semana después de la entrega de cada práctica, los alumnos podrán encontrar en la página Web de la asignatura la calificación asignada a cada grupo.

3. Resultados

La bondad del método expuesto se puede analizar en tres aspectos: el éxito del alumno en la asignatura, su opinión acerca de la metodología empleada y las ventajas respecto a la organización del trabajo docente del profesor.

El éxito de los alumnos se refleja en el porcentaje de alumnos presentados y aprobados en la asignatura. El porcentaje de alumnos presentados en la convocatoria ordinaria de Febrero en el curso 2005/2006 fue del 95,89% y el de aprobados el 80,82% del total de alumnos matriculados. Estos porcentajes llegan a alcanzar

casi el 100% si tenemos en cuenta las convocatorias extraordinarias de Septiembre y Diciembre. Creemos que, en relación a otras asignaturas de la carrera[1] y al elevado número de alumnos matriculados, estos números reflejan porcentajes bastante aceptables.

Por otro lado, al finalizar el cuatrimestre se realizan unas encuestas entre el alumnado destinadas a evaluar el método expuesto. Respecto a la capacidad formativa del método, la Tabla 1 muestra algunas de las preguntas realizadas y los resultados obtenidos[2].

Además, también se observa el cumplimiento de las ventajas de los proyectos prácticos en grupo mencionadas anteriormente. Algunas son obvias, como la *aplicación del conocimiento* o las *ideológicas*. En cuanto a la *motivación*, tal como se refleja en la Tabla 1, los alumnos encuentran el aprendizaje de la metodología más ameno y les atrae comprobar la aplicabilidad real de un proyecto de IC. En este sentido, creemos que también es importante el que el tema de la práctica quede a su elección.

Respecto a las *habilidades cognitivas de nivel superior*, el hecho de tener que realizar todas las etapas del ciclo de vida, y de que la calidad de una dependa de la calidad de las otras, obliga a los alumnos a pensar en profundidad y valorar su trabajo desde el principio del proceso. Además, las decisiones que tomen tendrán que ser defendidas en la presentación, lo que refuerza todavía más esta idea. Otras habilidades se adquieren con las presentaciones y defensa de las prácticas. Por último, la *autonomía* y el *asesoramiento* se consiguen dando a los miembros del grupo libertad para elegir en qué aspecto del proyecto centrarse y repartirse tareas, es decir, empleando un enfoque de aprendizaje cooperativo, frente a las técnicas más tradicionales de aprendizaje grupal. No obstante, este enfoque no implica que cada miembro del grupo desconozca lo que hacen los demás, (sino más bien al contrario, les obliga a coordinarse), ni permite que los miembros participen en distinto grado. Este aspecto, por otro lado, se comprueba durante las presentaciones de las prácticas.

He conseguido motivarme con esta asignatura				
Siempre	Casi siempre	A veces	Casi nunca	Nunca
27,27%	49,09%	18,18%	1,82%	3,64%
¿Cómo valoro la realización de prácticas?				
Interesantes (de 1(mínimo) a 5 (máximo))				
1	2	3	4	5
3,57%	17,86%	28,57%	37,50%	12,50%
Formativas (de 1(mínimo) a 5 (máximo))				
1	2	3	4	5
1,89%	7,55%	30,19%	39,62%	20,75%
El esquema planteado para la evaluación de la asignatura (50% prácticas, 50% examen) es…				
Adecuado	Inadecuado	No sé valorar		
89.47%	8.77%	1.75%		
Según las clases teóricas y mi conocimiento de la asignatura, las prácticas son...				
Adecuadas = 98,25%	Inadecuadas =1,75%			
Nº de miembros adecuado para los grupos				
1	2	3	4	> 4
1,79%	8,93%	17,86%	60,71%	10,71%
Fijar unos plazos de entrega de prácticas es..				
Positivo	Negativo	Indiferente		
81,48%	7,41%	11,11%		
Realizar las PRESENTACIONES de las prácticas es…				
Positivo	Negativo	Indiferente		
96,43%	1,79%	1,79%		
Cómo valoras conocer la nota de prácticas en un tiempo cercano a su entrega				
Positivo = 88,24%	Negativo = 11,76%			

Tabla 1. Resultados de encuestas realizadas por 140 alumnos en el curso 05/06.

Además de la orientación de la asignatura, también nos preocupaba estar aumentando en exceso la carga de trabajo del alumnado. Sin embargo, en una pregunta en la que se pedía calificar la dificultad de las prácticas de 1(muy fácil) a 5(muy difícil) los porcentajes se repartieron entre un 8.16% que las calificaron de dificultad *2*, un 26.53% que las calificaron de dificultad 4, mientras que el porcentaje más alto (65.31%) las situó en un segmento de dificultad media (3). Respecto al tiempo de preparación de

[1] Según los últimos datos institucionales proporcionados por la UDC y teniendo en cuenta todas las convocatorias de un curso, las medias de alumnos presentados y aprobados en las asignaturas troncales y obligatorias en la titulación de Ingeniería Informática son, respectivamente, el 76,09% y el 58,03% (http://udc.es/informacion/ga/estatistica/20052006).

[2] Estos resultados se refieren a las encuestas realizadas en el curso 2005/2006 y son porcentajes sobre el total de encuestas realizadas.

las presentaciones el 13.21% dice haber necesitado menos de 1 hora, el 60.38% entre 1 y 2 horas, y el 26.42% entre 2 y 3.

Es también importante resaltar la importancia que los alumnos dan al hecho de que este esquema favorece mucho la relación alumno/profesor y que el 96.49% juzga de muy positivo.

Finalmente, este método ha aportado también muchas ventajas al profesor. En el curso 2001/2002 comenzamos a aplicar esta orientación en que las prácticas suponían una aplicación paso a paso de los conceptos explicados en clase de teoría. La cantidad de alumnos por grupo era algo menor, pero el método para gestionar las prácticas seguía un esquema más tradicional. El trabajo que suponía la atención de la asignatura era casi inabordable debido al tiempo que consumía. Desde entonces, y con esta nueva aproximación, hemos disminuido considerablemente el uso de horas de tutorías fuera del horario oficial, así como el tiempo empleado en la corrección de las prácticas, hasta el punto de no necesitar del profesor mucho más tiempo que el que pueda requerir una asignatura con un enfoque más clásico. Todo ello se ha conseguido además, sin no sólo no disminuir sino aumentar la calidad de las prácticas y la implicación del alumnado.

4. Conclusiones y trabajo futuro

En esta ponencia se ha presentado una metodología docente con el fin de adaptar la asignatura de Ingeniería del Conocimiento a los nuevos enfoques de aprendizaje del EEES. El elevado número de alumnos en la Universidad Española nos obliga a establecer métodos que permitan esta adaptación. Creemos que la metodología presentada lo permite. Con ella hemos conseguido evaluar al alumno siguiendo un procedimiento continuo, estimular su proceso de aprendizaje y contribuir al desarrollo de muchas de las competencias establecidas en el proyecto Tuning, gracias a un esquema que es extrapolable a otras asignaturas de los planes de estudio de las Ingenierías Informáticas.

Agradecimientos

A nuestros alumn@s.

Referencias

[1] ACM/IEEE-CS Joint Curriculum Task Force. *Computing Curricula 2001*. Disponible en: http://www.sigcse.org/cc2001/ (accedido el 15/02/2006).

[2] Alonso Betanzos, A., Guijarro Berdiñas, B., Lozano Tello, A., Palma Méndez, JT., Tabeada, MJ. *Ingeniería del conocimiento. Aspectos metodológicos.* Pearson Ed., 2004.

[3] Alonso Betanzos, A., Guijarro Berdiñas, B., Lozano Tello, A. Un enfoque metodológico para la docencia en Ingeniería del Conocimiento. IX JENUI, pp. 291-298, 2003.

[4] De Miguel Díaz, M (coord.). *Metodología de enseñanzas y aprendizaje para el desarrollo de competencias. Orientaciones para el profesorado universitario ante el EEES*. Alianza, 2006.

[5] Gibbs, G. *Learning in teams, a tutor guide*. Oxford Centre for Staff Development, Oxford Brookes University, 1995.

[6] González, J., Wagenaar, R. (eds.) *Tuning Educational Structures in Europe*. Final Report Phase 1, Univ. Deusto, 2003.

[7] Henry, J. *Teaching through projects*. Routledge Falmer, 1994.

[8] Hughes, B., Cotterell, M. *Software Project Management*. McGraw-Hill, 1999.

[9] Jaques, D. *Learning in groups*. 3ª ed. London: Kogan Page, 2000.

[10] Joy, M. *Group projects and the computer science curriculum*. Innovations in Education & Teaching Int, Vol. 42(1), pp. 15-25, 2005.

[11] Reynolds, M. *Groupwork in Education and Training. Ideas in Practice*. The Educational and Training Technology Series, 1994.

[12] Schreiber, G., Akkermans, H., Anjewierden, A., de Hoog, R., Shadbolt, N., van de Velde, W., Wielinga, B. *Knowledge engineering and management. The CommonKADS methodology.* MIT Press, 2000.

[13] Shaw, M.L.G., Gaines, B.R. The synthesis of knowledge engineering and software engineering. E: Loucopulos, P. (Ed.), Advanced Information Systems Engineering, LNCS, vol. 593, 1992.

Una experiencia educativa en la asignatura de Estadística impartida en la Ingeniería Técnica Informática de la Universidad Politécnica de Valencia

[1]Ángeles Calduch, [1]Rosa Mª Alcover, [2]Rosa Albert
[1] Dpto. de Estadística e Investigación Operativa Aplicadas y Calidad
[2]Becaria Escuela Técnica Superior de Informática Aplicada
Universidad Politécnica de Valencia
Camino de Vera s/n, 46022 Valencia
mcalduch@eio.upv.es

Resumen

En este trabajo se describe una experiencia piloto de adaptación en la asignatura de primer curso de Estadística de la titulación de Ingeniería Técnica en Informática de Sistemas (ITIS), impartida en la Escuela Técnica Superior de Informática Aplicada (ETSIAp) de la Universidad Politécnica de Valencia (UPV), en el contexto del proceso de convergencia al Espacio Europeo de Educación Superior (EEES). Para ello presentamos la planificación de la asignatura en relación a las actividades de los alumnos: clases teóricas, prácticas, evaluación y tutorías; la utilización y aplicación de metodologías activas de enseñanza, y también los resultados que han obtenido los alumnos en la asignatura. Finalmente, se comparan estas calificaciones del grupo piloto con las obtenidas en el resto de los grupos de la asignatura.

1. Motivación

Desde la UE se está haciendo un esfuerzo para impulsar el proceso de Convergencia Europea en Educación Superior. Con él se pretende, en líneas generales, un reconocimiento más fácil de las titulaciones, asegurando una formación óptima de los estudiantes y su integración en el mercado laboral unificado y sin fronteras.

Actualmente, el sistema universitario español está inmerso en el proceso de adaptación al EEES, intentando, con más o menos dificultades, la aplicación de los principios y criterios de Convergencia Europea. Entre los objetivos globales de este proceso cabe destacar "el establecimiento de un sistema educativo de calidad". La universidad española ha elaborado diferentes acciones en materia de innovación educativa con el fin de mejorar la calidad de la enseñanza. Históricamente, en la UPV se han desarrollado con este fin diversos planes, tales como el Plan de Innovación Educativa (PIE), los Proyectos de Innovación Docente (PIDs), el Proyecto Europa y el Programa de Adaptación al EEES (PAEEES).

Todo ello ha desembocado en el diseño de un Proyecto General para nuestra universidad, en el marco del EEES, que supone para el profesorado nuevas competencias que afectan a conceptos, convicciones y actitudes sobre la enseñanza de las materias y el aprendizaje de los alumnos.

En este proyecto, los centros de la UPV son elementos clave para dinamizar o agilizar el proceso de convergencia europea. Por ello, se les propone diversas acciones, con el fin de facilitar el proceso de adaptación. En este sentido, una de las acciones contempladas es la propuesta de Proyectos Piloto de Adaptación al EEES.

La ETSIAp, consciente de la necesidad de promover medidas que permitan avanzar tanto en el proceso de la convergencia europea como en el terreno de la calidad de nuestra docencia, ha promovido la puesta en marcha de acciones concretas a través de experiencias de enseñanza-aprendizaje en un *grupo piloto* de primer curso en la titulación de Ingeniero Técnico en Informática de Sistemas. El objetivo de este grupo es el de experimentar la puesta en práctica de nuevas metodologías y la obtención de resultados que permitan conformar una opinión y adquirir experiencia de cara a la futura reestructuración de las titulaciones.

Desde la unidad docente de Estadística en los estudios de Informática se ha realizado una profunda revisión de contenidos y se ha planteado la adaptación tanto de metodología docente como

de los contenidos y del sistema de evaluación de la asignatura troncal Estadística (6 créditos, 3cr Teoría + 3cr Práctica), impartida en el segundo cuatrimestre de primer curso en las dos especialidades de Ingeniería Técnica Informática. Este proceso de revisión ha incluido una reflexión sobre los objetivos formativos de la asignatura en relación a las competencias transversales de la titulación.

Además, la experiencia que se ha realizado con esta asignatura unifica los objetivos perseguidos por la ETSIAp en la formación del grupo piloto con los planteados por la asignatura en sintonía con el plan de estudios vigente.

El trabajo que exponemos a continuación está organizado de la siguiente manera: en el apartado 2 se presentan las recomendaciones generales elaboradas por la ETSIAp en la creación y seguimiento del grupo piloto. En el apartado 3 se revisa la trayectoria de la docencia de la asignatura de Estadística en la ETSIAp en cursos anteriores. En el apartado 4 presentamos la descripción de las actividades docentes y de evaluación llevadas a cabo en la asignatura de Estadística en el grupo piloto. En el apartado 5 se estudian los resultados de la asignatura obtenidos por los alumnos en el grupo piloto y se comparan con los resultados de los otros grupos. Finalmente, en el apartado 6 se presentan las conclusiones.

2. Recomendaciones generales en el grupo piloto

La ETSIAp de la UPV, que imparte las titulaciones de Ingeniería Técnica de Informática en Gestión (ITIG) e Ingeniería Técnica de Informática en Sistemas (ITIS), teniendo en cuenta las recomendaciones que podemos encontrar en [4], llevó a cabo en el curso 2004 – 2005 el proyecto educativo "Evaluación de la carga de trabajo del alumnado y profesorado en las asignaturas troncales y obligatorias de las titulaciones ITIG e ITIS". Con este proyecto se introdujeron las bases para realizar una acción en uno de los 8 grupos de primer curso en el año escolar 2005 – 2006, y al que se ha denominado *grupo piloto*.

En concreto, en el grupo piloto, la enseñanza durante el pasado curso 2005 – 2006 fue impartida de manera más similar a los créditos ECTS, de modo que es un sistema que se centra más en el aprendizaje del alumno en vez de en la enseñanza del profesor. La evaluación del alumno se realiza de un modo continuo, con lo que el profesor tiene muchos más elementos para valorar la calificación final del alumno en la asignatura. En cuanto al número de alumnos, el grupo piloto era de menor tamaño que los otros siete grupos de primer curso. Estaba limitado a 60 – 80 alumnos matriculados en teoría, por la mayor complejidad en la docencia. El grupo de teoría se dividió en 3 subgrupos de prácticas de laboratorio de 20 – 25 alumnos. Aún así, creemos que la interacción profesor-alumno hubiera funcionado mejor si el número de alumnos hubiera sido más reducido.

Con el fin de trabajar con una muestra de alumnos lo más aleatoria y representativa posible, al efectuar éstos su matrícula en el primer curso de la ETSIAp, no fueron informados de la puesta en marcha del proyecto. De esta manera se intentó evitar que los que ingresaran en él fueran alumnos voluntarios, quizá con un mejor expediente académico, y con una mayor predisposición al estudio. Además, el total de alumnos repetidores de primer curso se distribuyó en la matrícula de forma uniforme entre los ocho grupos de teoría.

El objetivo planteado con la creación del grupo piloto fue la comparación de los resultados de las calificaciones obtenidas por estos alumnos a final de curso con las notas de los otros alumnos de primero matriculados en los siete grupos restantes. Esto nos permitiría comprobar si la nueva metodología docente impartida en él les ayudaba a superar el curso y conseguir mejores calificaciones en las asignaturas que sus compañeros.

Al iniciar el curso escolar, a los alumnos del grupo piloto se les entregó un calendario con las fechas de los exámenes parciales que tendrían que realizar, así como las fechas de entrega de trabajos. Cada asignatura fijó sus propios pesos o importancia sobre la calificación final de un alumno a los exámenes parciales, entrega de trabajos y examen final. Con ello, los alumnos podían planificarse el curso completo desde septiembre.

Cada profesor firmó con cada uno de los alumnos un contrato de enseñanza-aprendizaje, en el que el profesor se comprometía a evaluar al alumno de la forma indicada, y el alumno se comprometía a asistir como mínimo a un 80% de las clases de la asignatura. En cualquier momento,

si el alumno consideraba que no podía cumplir el contrato, éste se podía romper, y se llevaría a cabo la clásica evaluación de examen final. El examen final de cada una de las asignaturas de primer curso sería el mismo para todos los alumnos de todos los grupos, con el fin de que fuera posible comparar así los conocimientos y las calificaciones finales obtenidas entre el grupo piloto y el resto de los grupos, aunque aquí volvemos a incidir en la ponderación del citado examen. Hay que tener en cuenta que *el cambio metodológico que se propugna no debe suponer en ningún momento una disminución del nivel de exigencia al alumno,... debería ser entendido como un instrumento de mejora de calidad*, como se indica en [2].

3. Trayectoria de la asignatura Estadística en la ETSIAp

Hay que resaltar que durante los últimos cursos, el número de alumnos inscritos en las titulaciones de Ingeniería Técnica Informática ha descendido con respecto a cursos anteriores. Como consecuencia de ello, también ha sido más baja la nota de acceso de sus estudios previos. El número de alumnos de nuevo ingreso que se vienen admitiendo en cada una de las titulaciones es de 200. Sin embargo, con el número de alumnos repetidores de cursos anteriores, primer curso sigue contando con un número de alumnos bastante elevado.

El enfoque que se le ha venido dando a la asignatura Estadística en la ETSIAp ha sido totalmente aplicado, y la metodología docente que se ha seguido ha sido bastante interactiva, respondiendo al modelo PIE introducido en 4º curso de la Facultad de Informática en 1990, y atendiendo a las recomendaciones planteadas en [3]. Sin embargo, como se muestra a continuación, los resultados obtenidos con esta metodología en la ETSIAp no han sido tan buenos como se esperaba.

Para el curso 2004 – 2005, de un total de 712 alumnos matriculados en la asignatura, la cantidad de alumnos que no realizaron el examen en ninguna de las dos convocatorias fue de 297, un 41,71% del total. El porcentaje de alumnos suspensos fue de un 23,46%. Tan sólo un 22,89% aprobó la asignatura en la convocatoria de junio, y un 11,94% la superaron en septiembre. Estos porcentajes se representan gráficamente en la Figura 1. Como se puede apreciar a partir de los datos anteriores, el número de alumnos que no se ha presentado a los exámenes de la asignatura es muy alto, siendo este valor superior al 40% en los datos que disponemos. Estos datos ponen de manifiesto el alto índice de abandono en la asignatura (al igual que ocurre en otras asignaturas de primer curso). Por ello decidimos introducir la nueva metodología de enseñanza-aprendizaje en el grupo piloto, para intentar lograr, entre otras cosas, que este índice descendiera. Principalmente, al hacer más exámenes parciales esperábamos que los alumnos consideraran más accesible y llevadera la asignatura. También hay que tener en cuenta que una parte de este porcentaje de abandono se debe a que la asignatura Estadística es de segundo cuatrimestre, por lo que los alumnos que hayan obtenido malas calificaciones en las asignaturas del primer cuatrimestre, probablemente centrarán sus esfuerzos en junio en recuperar estas asignaturas, ya conocidas por ellos, en vez de enfrentarse a una nueva asignatura.

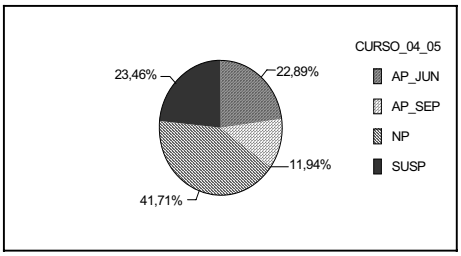


Figura 1. Diagrama de sectores con los porcentajes de las calificaciones de los alumnos en el curso 04-05

4. La asignatura de Estadística en el grupo piloto

Los contenidos básicos o bloques temáticos que se imparten en la asignatura troncal de Estadística son los siguientes:

- Estadística descriptiva.
- Probabilidad.
- Distribuciones de probabilidad.
- Inferencia en poblaciones normales.

En el grupo piloto, la asignatura fue planteada de modo que se conjugaran los contenidos con la realización de pruebas parciales. A continuación pasamos a exponer el planteamiento de las clases.

La parte de Estadística descriptiva se explicó a los alumnos sobre ejemplos del área de su interés (Informática) en prácticas de laboratorio, utilizando ordenadores, en lugar de en clases de aula como se venía haciendo en cursos anteriores. El software estadístico que se utilizó en las prácticas de ordenador fue el *Statgraphics* [4]. La duración de las prácticas de ordenador fue de dos horas para cada uno de los tres subgrupos, por lo que se propuso una práctica autodirigida de una hora y media, y en la media hora restante, los alumnos realizaron una breve evaluación de esta parte de la materia con preguntas similares a las realizadas durante la práctica.

Con esta estructura, las clases de teoría y las prácticas de aula comienzan con la parte de Probabilidad, llevando a cabo un desarrollo introductorio realizado por la profesora, para pasar a mostrar diferentes aplicaciones prácticas de lo estudiado y a la realización de ejercicios por parte de los alumnos. Finalmente, se evaluó a los alumnos con un examen parcial sobre este bloque temático.

La experiencia de cursos anteriores nos ha demostrado que los alumnos encuentran una gran dificultad en el aprendizaje de los temas relacionados con las distribuciones de probabilidad, ya que les supone un especial esfuerzo comprender los conceptos que incluyen estos temas y sobre todo el diferenciar los distintos tipos de distribución.

Por este motivo, en el bloque temático de Distribuciones de probabilidad se llevaron a cabo en el grupo piloto las siguientes acciones: dar a los alumnos los conocimientos mínimos acerca de las distribuciones de probabilidad y dividir el grupo de teoría en cuatro subgrupos de aproximadamente 12 alumnos. Cada uno de estos subgrupos estudiaba una distribución de probabilidad. A su vez, cada subgrupo se fragmentó en tres de cuatro personas cada uno, y se les dejó unos días para que recopilaran material sobre la distribución de probabilidad correspondiente. El trabajo realizado por los alumnos fue supervisado por la profesora. Posteriormente, cada subgrupo tuvo que explicar al resto de compañeros del grupo las características más relevantes de la distribución de probabilidad sobre la que habían trabajado y sus posibles aplicaciones en el entorno informático, para efectuar después una serie de ejercicios evaluables acerca de todas las distribuciones, no sólo la propia sobre la que han trabajado, sino de todas las que se fueron exponiendo. De esta manera, la profesora, al supervisar el desarrollo del trabajo pudo valorar, además de los conceptos teóricos aprendidos, la capacidad de trabajar en equipo. En la evaluación de estos trabajos en equipo, también se tuvo en cuenta la exposición del trabajo al resto de alumnos, y, por último, la asimilación de los conocimientos acerca de las otras distribuciones de probabilidad que les fueron explicadas por sus compañeros.

Entre las exposiciones de los trabajos y la resolución del examen parcial de esta parte de la asignatura, los alumnos dispusieron de unos días en los que la profesora se dedicó a resolver las dudas acerca de los conceptos y ejemplos que surgieron en las exposiciones de los compañeros. Hay que destacar que el alumno de cada subgrupo que realizó la exposición fue elegido por sorteo de entre los integrantes del subgrupo, con el fin de que todos los alumnos se vieran involucrados en el desarrollo del trabajo, ya que la nota de la exposición del alumno era común para todo el subgrupo.

Finalmente, respecto al bloque temático de Inferencia en poblaciones normales, los contenidos se introdujeron tanto en las clases de aula (teoría y problemas) como en las clases de laboratorio. En este bloque, a diferencia de los dos anteriores, no se llevó a cabo evaluación parcial, puesto que consideramos que al impartirse al final de curso, cuando los alumnos tienen muy cerca los exámenes finales, hubiera sido una sobrecarga adicional de trabajo para ellos. Además, hay que tener en cuenta que unos días más tarde se examinaron de toda la asignatura.

Con esta organización, pensamos que se trabajan las siguientes competencias transversales genéricas: capacidad de análisis y síntesis, capacidad de organización y planificación, comunicación oral y escrita, trabajo en equipo, capacidad de gestión de la información, resolución de problemas y toma de decisiones, trabajo en equipo, razonamiento crítico y aprendizaje autónomo.

Respecto a las calificaciones de los alumnos, los pesos que se asignaron a las evaluaciones de los diferentes bloques temáticos y actividades realizadas durante el curso fueron los siguientes:

- Examen: 0,4.
- Prácticas: 0,2.
- Prueba de ejercicios de Probabilidad: 0,1.
- Presentación trabajo de Distribuciones: 0,15.
- Prueba de ejercicios de Distribuciones: 0,1.
- Participación activa en clase: 0,05.

Se consideró que para superar la asignatura, la nota del examen debía ser mayor o igual a 3.

Hay que hacer notar que los profesores implicados en la docencia del grupo piloto en todas las asignaturas de primer curso de la ETSIAp llegaron al acuerdo siguiente: la nota de cada alumno en cada asignatura sería el máximo entre su calificación obtenida en dicha asignatura en el grupo piloto, y la que hubieran obtenido en caso de estar matriculados en cualquiera de los otros siete grupos. La calificación de un alumno de la asignatura de Estadística matriculado en cualquiera de los otros siete grupos de primer curso se obtiene como la nota del examen más la nota de prácticas, siempre que la nota del examen sea mayor o igual a 4,5, y teniendo en cuenta que la nota de prácticas varía entre -1 y +1.

Por otra parte, en lo referente a tutorías, hemos constatado en cursos anteriores que la mayoría de los alumnos que asisten lo hacen los días previos al examen. Con el fin de que las tutorías se aprovecharan desde el principio por parte de los alumnos, se organizaron de la siguiente manera: dos de las seis horas semanales de tutorías/consultas a la profesora se realizaron en el aula docente del centro, en lugar de en el despacho de la profesora. Así, las tutorías podían desarrollarse tanto de manera individual como en grupo. Se consideró la posibilidad de tutorías grupales, pues al ser Estadística una asignatura eminentemente práctica, hay muchos alumnos que estudian juntos para resolver ejercicios, por lo que en numerosas ocasiones tienen las mismas dudas. Con ello también se intentó evitar las largas colas formadas por alumnos en el despacho del profesor en vísperas del examen, y se procuró que el alumno comenzara a trabajar la asignatura desde el inicio del cuatrimestre.

5. Resultados obtenidos por los alumnos

Los alumnos recibieron muy bien la propuesta del grupo piloto, y en particular de la asignatura Estadística, ya que para ser una asignatura de 2º cuatrimestre, con el abandono previo que se le presupone, a los 71 alumnos matriculados inicialmente hubo que añadir 7 alumnos repetidores que solicitaron su inclusión en el grupo. Sabemos que al aceptarlos se desvirtuaba un poco la comparación que quería hacerse entre grupos, ya que los porcentajes de repetidores y no repetidores variaban, pero en algunos casos eran alumnos que arrastraban años de fracaso en la asignatura, y pensamos que si este sistema podía ayudarles a superar la asignatura, valía la pena que el grupo fuera un poco más grande.

Finalmente, después de las nuevas incorporaciones que se produjeron en las dos primeras semanas, el grupo piloto en la asignatura Estadística quedó formado por 78 alumnos. De éstos, 2 seguían matriculados, pero habían abandonado la titulación en el primer cuatrimestre. De los 76 restantes, hubo 16 que no vinieron a clase ningún día, y 4 que abandonaron la asignatura en las primeras semanas de docencia. Así, 56 alumnos firmaron el contrato de enseñanza-aprendizaje. De ellos, 5 alumnos que no cumplieron el mínimo de asistencia a clase (del 80% como se ha indicado en el apartado 2 de este trabajo) por lo que finalmente, la cantidad de alumnos que cumplían todos los requisitos para ser evaluados mediante esta modalidad fue de 51. Estos abandonos se produjeron antes del primer parcial.

Hay que resaltar que el clima creado en el aula era agradable y relajado, y que los alumnos se mostraron muy receptivos con el método de enseñanza y de evaluación.

El primer examen parcial fue la prueba en la que se obtuvieron peores resultados, ya que fue superada por 34 alumnos, mientras que la tercera parte de los alumnos matriculados obtuvo una nota inferior a 5. Todos los alumnos superaron la parte de la evaluación correspondiente a los trabajos de distribuciones de probabilidad, pero sin embargo, el segundo parcial, en el que se examinaban de la parte de la asignatura que habían presentado en los trabajos, el número de alumnos que consiguió una calificación superior o igual a 5 fue de 29. Fueron 18 los alumnos que

salieron voluntarios a la pizarra para resolver ejercicios y explicarlos a sus compañeros, y 14 de ellos lo hicieron en dos ocasiones. Respecto a las prácticas, los resultados fueron muy buenos, ya que la nota más baja que se obtuvo fue de un 6,17.

Respecto al examen final de la convocatoria de junio, en el que recordemos que los alumnos tenían que obtener una nota superior o igual a 3 para que se les pudiera calcular la nota media con los otros apartados de la evaluación, los resultados que se obtuvieron fueron los siguientes:

- Sólo 3 de los alumnos obtuvieron una nota inferior a 3, por lo que no aprobaban con ninguno de los dos tipos de evaluación.
- 9 alumnos obtuvieron una nota mayor o igual a 3 e inferior a 4,5. Estos alumnos, que no hubieran superado la asignatura con la evaluación tradicional aplicada a los otros grupos, sí que lo hicieron con la evaluación del grupo piloto.
- Así pues, 39 de los 51 alumnos (un 76,47%) hubieran superado la asignatura evaluándose como el resto de grupos (realizando las prácticas). De ellos, 6 fueron calificados con una nota inferior a 5, 24 tuvieron una nota de aprobado, 7 fueron calificados con notable, y para 2 su nota fue mayor o igual que 9, y por lo tanto de sobresaliente.

Como vemos, un 76,47% de los alumnos superaron la asignatura con el método tradicional. Se trata de un buen porcentaje comparado con el obtenido el curso anterior (recordemos que era de un 22,89%). Por otra parte, las calificaciones de los alumnos del grupo piloto (1G) en el examen final se han comparado con las obtenidas por los alumnos matriculados en los otros grupos impartidos por la misma profesora (1A y 1C). Gráficamente, la comparación se muestra en la Figura 2. Como puede apreciarse en los gráficos de caja y bigotes de esta figura, los resultados del grupo piloto (1G) fueron superiores al resto.

Además, también se ha realizado una comparativa de las notas entre los alumnos del grupo piloto y las obtenidas por los alumnos matriculados en el resto de grupos. Los resultados pueden verse en la Figura 3. Como se observa en los gráficos de caja y bigotes de esta figura, los resultados del grupo piloto (1G) fueron superiores al resto.

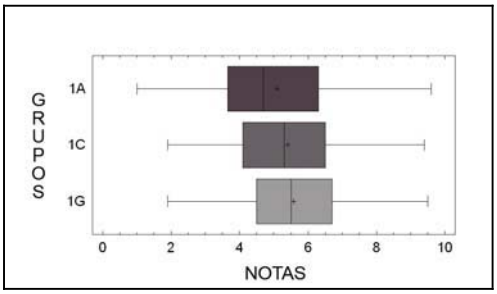


Figura 2. Gráfico de caja y bigotes para las notas del examen de los alumnos del grupo piloto frente a los otros grupos impartidos por la misma profesora.

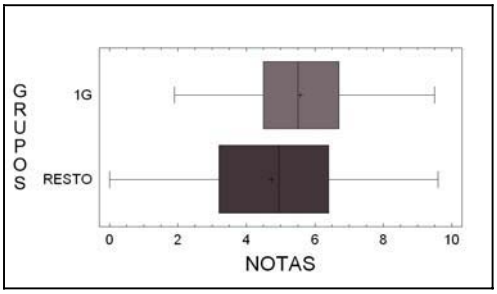


Figura 3. Gráfico de caja y bigotes para las notas del examen de los alumnos del grupo piloto frente al resto de alumnos.

6. Conclusiones

La experiencia docente llevada a cabo en la asignatura de Estadística en el grupo piloto ha sido, en general, satisfactoria. Por una parte, la relación con el alumnado ha sido más estrecha que con el sistema tradicional. Por otra, las calificaciones obtenidas por los alumnos del grupo piloto han sido más altas que las de los alumnos de otros grupos.

Un elemento que desempeña un papel crucial en el éxito de la nueva experiencia es el nivel de motivación e implicación del alumnado. El nuevo sistema les exige una mayor dedicación y trabajo que el sistema anterior. Es por tanto necesario buscar un equilibrio entre la dedicación demandada al alumno y la realmente ofrecida, pues si no hay sintonía entre estos dos parámetros se corre el riesgo de fracasar con este tipo de metodologías docentes. Por otra parte, esta clase

de experiencias deben, en nuestra opinión, implantarse en todas las asignaturas de un grupo, pues en otro caso podría ocurrir que el alumno invierta una mayor dedicación y esfuerzo en el seguimiento de las asignaturas con esta modalidad de docencia, en detrimento de otras con tipos de docencia más tradicionales.

Otro problema en el seguimiento de este sistema corresponde al de los alumnos que no asisten sistemáticamente a clase, bien porque trabajan o bien por otros motivos, y que no pueden dedicar mayoritariamente su tiempo a los estudios. Los alumnos con este perfil parece que se decantan por metodologías más tradicionales que les exigen una menor implicación en las clases y en el desarrollo de exámenes parciales, ya que es distinto pedir un día de trabajo para realizar un examen final que para llevar a cabo uno parcial.

Pero en lo que respecta a este grupo en particular con el que se ha trabajado, los alumnos solicitaron seguir teniendo este tipo de docencia en segundo curso, ya que estaban motivados por el experimento del primer curso, y querían que continuara.

Así pues, aunque hay pocos datos, ya que sólo tenemos los resultados de un curso académico, la gente que ha participado está contenta en general, tanto los profesores como los alumnos, y para el siguiente curso, el actual 2006 – 2007, se está llevando a cabo un tipo de docencia mixta en el primer curso de la ETSIAp de la UPV, aprovechando la experiencia positiva ya realizada, y teniendo en cuenta que no todo el profesorado está motivado a embarcarse en un proyecto como fue el del grupo piloto.

Agradecimientos

A la ETSIAp de la UPV por haber dotado de los recursos y el apoyo necesarios para poder llevar a cabo esta experiencia, y en especial a Mª Engracia Gómez Requena.

Referencias

[1] Escuela Técnica Superior de Informática Aplicada. Universidad Politécnica de Valencia. *Informe sobre la matrícula de los cursos 2001-2002, 2002-2003, 2003-2004, 2004-2005 y 2005-2006.* Microweb de la ETSIAp.

[2] Escuela Técnica Superior de Informática Aplicada. Universidad Politécnica de Valencia. *Propuesta de implantación de un grupo piloto en primer curso en la Escuela Técnica Superior de Informática Aplicada durante el curso 2005-2006*. 2005.

[3] Peña, D., Prat, A. y Romero, R. *La enseñanza de la Estadística en las Escuelas Técnicas.* Estadística Española, 32, pp. 147-200, 1990.

[4] Statgraphics Plus v.5.0. Manugistics Inc. (2000).

[5] Vicerrectorado de Planes de Estudio y Títulos Propios. Universidad Politécnica de Valencia. *Recomendaciones para el Programa Docente de las asignaturas que participan en el Proyecto Piloto de Experimentación del ECTS.* Curso 2004-2005. 2004.

Estudio de Adecuación al Espacio Europeo de Educación Superior en Informática

E. Ferrando Julià[(1)], V. Brotons Gil[(1)], D.Ruiz Fernández[(1)], M. Moreno Rando[(2)]
(1)Dpto. de Tecnología Informática y Computación
Universidad de Alicante
Carretera San Vicente s/n, 03690 San Vicente del Raspeig (Alicante)
{eferrando, vbrotons, druiz}@dtic.ua.es
(2)Instituto de Enseñanza Secundaria Mare Nostrum
C/ Panamá s/n, 03008 Alicante
mmoreno@terra.es

Resumen

En el presente trabajo, a través de una encuesta, se realiza un análisis de la situación actual de las ingenierías informáticas, los perfiles profesionales que más se buscan y la aceptación de adaptación al Espacio Europeo de Educación Superior (EEES) en la titulación de Informática. La encuesta ha sido realizada a alumnos preuniversitarios procedentes de bachillerato y de ciclos formativos, universitarios de últimos cursos, ingenieros titulados y personal docente universitario.

Una vez analizados los resultados, se pone de manifiesto especial interés por asignaturas relacionadas con tecnologías basadas en Internet y el descontento por la ausencia de asignaturas relacionadas con el entorno empresarial. También se han estudiado las perspectivas profesional y la actitud frente al cambio de plan de estudio debido al Espacio Europeo de Educación Superior (EEES).

1. INTRODUCCIÓN

En todos los países se considera la educación como uno de los pilares del progreso, concretamente en Europa, existe una elevada preocupación porque los países miembros consigan un elevado nivel académico que permita el continuo desarrollo de la sociedad [4].

La declaración de Bolonia de 1999 [5] marca el inicio oficial del proceso de convergencia hacia un espacio común de educación, en el que participan todos los estados miembros de la Unión Europea [7]. El proceso hace referencia a cambios como la implantación de una nueva estructura de titulaciones universitarias, un sistema de créditos ECTS (European Credit Transfer System) y un mejor programa de movilidad, de manera que se garantice la calidad de la educación y el permanente aprendizaje en todo el territorio europeo [3][7].

Con motivo de la adaptación al Espacio Europeo de Educación Superior (EEES), hemos realizado un estudio para conocer la valoración por parte de diferentes colectivos de aspectos relacionados con el plan de estudios de ingeniería en informática, concretamente en la Universidad de Alicante [1]. También intentamos analizar las perspectivas profesionales del alumnado y su visión de los nuevos planes adaptados al EEES.

2. OBJETIVOS Y MÉTODO

En primer lugar, queremos valorar los contenidos de las titulaciones de informática en diferentes áreas de especialización; para ello, hay preguntas en la encuesta que recogen la opinión de los miembros del entorno educativo en temas relacionados con asignaturas optativas ofertadas, implantación de asignaturas en inglés o docencia a distancia, no presencial. Entre otras cuestiones también se pregunta por el nivel de exigencia que hay en la titulación y el nivel de preparación del personal docente.

En segundo lugar, se quiere conocer las expectativas profesionales de los futuros ingenieros informáticos, al igual que, su opinión sobre su nivel de preparación respecto a sus vecinos europeos; así pues, en la encuesta se pregunta directamente sobre estos temas. A todos los colectivos exceptuando al personal docente e investigador. Resulta interesante para este trabajo la opinión del colectivo universitario sobre la realización de un proyecto final de carrera, la

realización de prácticas en empresa o las líneas de investigación que les sugiere más interés. Para tratar estas cuestiones se recoge a modo cualitativo la información de las preguntas de la encuesta.

Por último, en lo referente a la adaptación al EEES, pretendemos conocer cuales son las inquietudes frente al cambio o la posible mejora de la calidad en la enseñanza universitaria en Europa.

Para intentar dar respuesta a las cuestiones anteriores, se ha realizado una encuesta formada por 28 preguntas. Dicha encuesta ha sido adaptada dependiendo de las características del colectivo al que se estuviera encuestando, modificando la orientación de la pregunta o, a través del encuestador, explicando el significado de la pregunta (especialmente en alumnos de Bachillerato y formación profesional). De todas las cuestiones podemos clasificar 14 relacionadas con la valoración de los contenidos de las titulaciones, 6 se centran en las perspectivas profesionales y las restantes se dedican a cuestiones relacionadas con la adaptación al EEES.

La encuesta ha sido realizada a diferentes colectivos, relacionados con los estudios universitarios de informática: alumnos pre-universitarios de bachillerato interesados en ingeniería informática, alumnos de ciclo formativo de informática, alumnos universitarios de últimos cursos, titulados universitarios y personal docente de la universidad. Del primer grupo fueron encuestados 21 estudiantes, para saber las expectativas que tienen los alumnos acerca de esta carrera. Del segundo grupo, 53 alumnos de ciclos formativos de grado medio y superior, fueron escogidos por estar en contacto con la informática y saber, si se diera el caso, por qué desean acceder a la universidad tras cursar un ciclo formativo. El tercer grupo, 68 estudiantes universitarios, fue elegido por estar directamente relacionados con los planes de estudio de informática actuales y con el EEES. En el grupo de ingenieros titulados hay 34 encuestados, nos interesaba saber los conocimientos adquiridos y cómo les han sido útiles para sus perspectivas laborales. Por último, 10 encuestados pertenecen al colectivo de personal docente e investigador y han sido elegidos con el fin de valorar su visión respecto la preparación de los alumnos, la programación impartida y cómo consideran de beneficioso los cambios de adaptación al EEES.

3. ANÁLISIS DE DATOS OBTENIDOS

El estudio que hemos realizado está centrado en tres áreas: valoración de los contenidos de las titulaciones, perspectivas profesionales y adaptación al espacio europeo de educación superior (EEES).

3.1 Valoración de los contenidos de las titulaciones

Los resultados obtenidos en las encuestas realizadas a distintos ciclos formativos de grado medio de Exploración de Sistemas Informáticos, de grado superior de Administración de Sistemas Informáticos y de Desarrollo de Aplicaciones Informáticas del instituto Mare Nostrum de Alicante reflejan que la mayoría de los encuestados decidieron hacer el ciclo formativo porque les gusta la informática, sin atender a criterios de expectativas laborales. La opinión general es que no es necesario tener conocimientos ni de matemáticas ni de física para estudiar informática, información que responde a la pregunta en la que deben de valorar del 1(bajo) a 5(alto) el nivel de matemáticas y de física que debería impartirse. Esta pregunta se plantea debido a que en el RD 55/2005 se exponen las directrices para la adaptación al EEES la cantidad mínima de créditos debería dedicarse a estas materias [6]. Por otro lado, se proponen asignaturas optativas muy prácticas tales como un taller de reparación de computadores, o programación de videojuegos. Así mismo, piensan que se debería añadir asignaturas de domótica y redes inalámbricas, ya que este sector está en pleno crecimiento en el mercado. Hay que tener en cuenta que el elevado grado de pragmatismo de este sector viene influenciado por la orientación eminentemente práctica de los ciclos formativos. En lo que respecta a cambios en la docencia, la mayoría señalan que las asignaturas en una segunda lengua, como el inglés, serían de gran utilidad ya que, en informática, es el idioma más utilizado. No hay una opinión unánime sobre la docencia no presencial: a una parte del colectivo les parece de gran utilidad que se oferten asignaturas a distancia, no presenciales mientras

otros discrepan considerando que son más útiles las clases presenciales.

Según los resultados obtenidos en las encuestas realizadas a estudiantes de bachillerato, desean matricularse en ingeniería informática porque les gusta este campo. Piensan que es necesario tener más conocimientos de matemáticas que de física para estudiar esta titulación. Por otro lado les atraen especialmente asignaturas optativas como programación de páginas web y administración de sistemas. También les gustaría que se impartieran asignaturas por parte de empresas especialistas del sector ya que ayudaría a entender el funcionamiento de una empresa real y facilitaría su incorporación al mundo laboral. Asimismo, encuentran interesante la docencia en inglés ya que posibilita el acceso a gran cantidad de recursos. Además, la mayoría creen que sería de gran utilidad que se ofertasen asignaturas a distancia, no presenciales.

Casi todos los estudiantes encuestados de la Universidad de Alicante decidieron estudiar informática por las expectativas laborales que la titulación presenta. Afirman que el nivel de matemáticas debe ser más elevado y manifiestan especial interés por asignaturas relacionadas con la programación web y bases de datos; también consideran que existe una carencia en asignaturas optativas del campo empresarial. Los estudiantes opinan que están adquiriendo suficientes conceptos durante su carrera y están preparados para el mundo laboral, aunque también existe un porcentaje elevado que cree que es sólo una base para "enseñarles a ser ingenieros" (y todo lo que el término conlleva) y que donde realmente se forman es en el trabajo. Este colectivo está de acuerdo con el nivel de exigencia de los estudios, aunque opina que el tiempo que se le dedica a las prácticas de las asignaturas es excesivo. Respecto al nivel de exigencia en los exámenes consideran que, en la mayoría de las asignaturas, es coherente con el nivel que se imparte en clases, y que la competencia de los profesores que imparten las asignaturas en las titulaciones de informática, es apropiada, aislando casos muy concretos. Aún así, proponen distintos cambios en la docencia, como que se den más asignaturas en inglés, ya que sería una manera de mejorar en su futuro profesional. Creen que sería de gran utilidad que se ofertasen asignaturas a distancia, no presenciales, y que algunas de las asignaturas fuesen impartidas por empresas del sector que transmitiesen su modo de trabajo, funcionamiento y su conocimiento adquirido a base de la experiencia.

En lo que respecta a los ingenieros titulados, la mitad de los encuestados opinan que en el primer curso debería haber un mayor nivel de matemáticas y de física. Las asignaturas optativas que consideran más importantes desde su punto de vista son tecnologías web, domótica y proponen otras como informática aplicada al derecho, bioingeniería, informática audiovisual y redes inalámbricas. La mayoría de los encuestados apuestan por la aplicación de la informática a todos los procesos que se puedan automatizar, siendo los más propuestos: financiero, automóviles, psicología, educación y recursos energéticos alternativos.

Todos los titulados coinciden en que es interesante cursar asignaturas en las que las empresas enseñen su modo de funcionamiento, qué tareas desempeñan, así como los conocimientos adquiridos con la experiencia. Este colectivo admite no haber cursado asignaturas en otros países de Europa, porque la mayoría consideran el idioma como una barrera. Pero afirman que en caso de hacerlo habrían preferido países de habla anglosajona, Alemania u Holanda.

La opinión general en cuanto a la exigencia en los estudios de las titulaciones en la Universidad de Alicante es que el nivel es intermedio-alto. En opinión de un gran grupo de los encuestados el nivel que se exige en los exámenes es coherente con el de las clases impartidas. La valoración general sobre la competencia de los profesores que imparten asignaturas en las titulaciones de informática, es positiva, exceptuando algunos casos aislados.

El colectivo perteneciente al personal docente universitario cree que en general el nivel educativo en la universidad ha disminuido debido

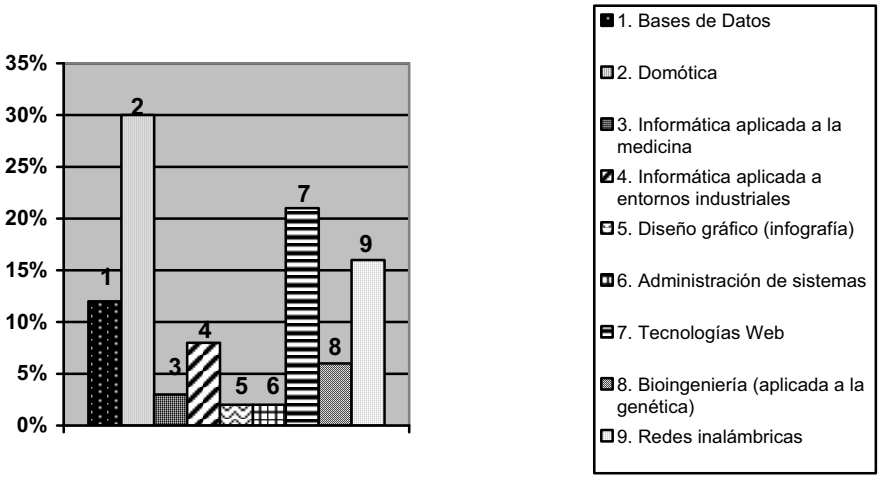


Figura 1. Porcentaje global de interés por las distintas áreas propuestas en la encuesta relacionadas con asignaturas optativas

al bajo conocimiento con el que los alumnos acceden a la universidad. Respecto a la exigencia en los exámenes consideran que, en la mayoría de las asignaturas, es coherente con el nivel que se imparte en clases. Por otra parte, proponen distintos cambios en la docencia, como que se den más asignaturas en inglés, ya que sería una manera de mejorar el acceso al mundo laboral. La mayoría creen que sería de gran utilidad que se ofertasen asignaturas a distancia, no presenciales, pero también hay algunos que piensan que de esta manera se pierde la relación personal con los profesores y los compañeros. Consideran favorable que algunas de las asignaturas fuesen impartidas por empresas del sector de manera que les proporcione a los alumnos una mejor visión del mundo que se encontraran al terminar sus estudios.

En general, tras la valoración global de los diferentes colectivos se presenta en la figura 1, un resumen del interés de las diferentes áreas propuestas relacionadas con asignaturas optativas. Se puede observar que el área de domótica o automatización del hogar es la que despierta un mayor interés. Otra conclusión importante de este apartado es el elevado interés por la docencia en inglés.

3.2 Perspectiva profesional

En los aspectos relacionados con el futuro profesional los alumnos de ciclos formativos consideran de mayor interés aquellas salidas profesionales relacionadas con la programación web, las aplicaciones de gestión y la administración de sistemas. Este colectivo para finalizar sus estudios tiene que realizar prácticas en empresa (formación en centros de trabajo, FCT), la mayoría lo valoran positivamente, considerándolo una manera de poner en práctica lo que han aprendido en clase, aunque también hay quienes creen que si se aumentara el número de créditos de prácticas durante las clases lectivas no sería necesario realizar la FCT. Hay una pregunta en la encuesta que lista 10 líneas de investigación, y que los encuestados deben clasificar del 1 (menor interés) al 10 (mayor interés), considerando las más interesantes: la informática médica y bioingeniería, inteligencia artificial y seguridad. Por otro lado, cabe destacar que no se ven capacitados para competir por un trabajo en el extranjero por la dificultad con el idioma. Se hacen preguntas relacionadas con el idioma anglosajón, porque en el RD 55/2005[7], se destaca como objetivo fundamental que lo titulados deben saber leer y hacer una presentación en al menos uno de los idiomas

comúnmente utilizados en la disciplina con especial atención al inglés.

Los alumnos de bachiller consideran de mayor interés aquellas salidas profesionales relacionadas con la investigación, el análisis o la programación. Opinan que las prácticas en empresa son muy útiles para introducirse en el mundo laboral y no se ven capacitados en el futuro para competir por un trabajo en el extranjero debido a la dificultad con el idioma. Consideran que la realización del proyecto de fin de carrera y del doctorado aportará una ampliación de sus conocimientos y una mejora en su currículum. Además, para los que se inclinan por la investigación, las líneas más interesantes son las de inteligencia artificial, seguridad y redes de comunicaciones.

Los alumnos de últimos cursos de las titulaciones en informática sienten interés por aquellas salidas profesionales relacionadas con la programación web, la docencia y la investigación. Gran parte de ellos consideran importante la realización del proyecto de fin de carrera y aconsejan a todos aquellos que estén interesados en la docencia e investigación la realización del doctorado, aunque creen que no se valora lo suficiente para el gran esfuerzo que conlleva. Además, para los que deseen orientar su vida profesional hacia la investigación, consideran que las líneas más interesantes son inteligencia artificial, informática médica y bioingeniería. No se encuentran capacitados para competir por un trabajo en el extranjero aunque con la posibilidad de hacer prácticas en empresa ven otra alternativa para introducirse en el mundo laboral. Según las directrices del RD 55/2005 [2], uno de los objetivos que deben cumplir las nuevas titulaciones adaptadas al EEES, es impulsar a los alumnos a que hagan prácticas tuteladas, así como que las prácticas sean en universidades o empresas de otros países.

Los ingenieros titulados afirman que las profesiones más demandadas son la enseñanza, la consultoría y la investigación. La mayoría de los encuestados creen que los conocimientos adquiridos durante la carrera son suficientes para entrar en el mundo laboral. En caso de dirigir la vida profesional hacia la investigación, la mayoría de los encuestados elegirían la rama de redes de comunicación o bioingeniería. Su perspectiva para competir por un trabajo a nivel europeo una vez finalizados los estudios es, en general, de pocas oportunidades, al igual que para encontrar el primer empleo. En este último aspecto, destacan que las principales causas son el intrusismo y que hay más demanda que oferta laboral.

El personal docente universitario considera imprescindible cursar el doctorado sobretodo para quienes se quieren dedicar a su misma profesión. Gran parte de los encuestados opinan que el proyecto final de carrera es fundamental, y lo consideran un resumen de todos sus esfuerzos a lo largo de la carrera. La pregunta acerca de proyecto final de carrera (PFC), fue planteada debido a que el RD 55/2005 [6] marca que el PFC debería cumplir un mínimo de 30 ECTS a diferencia que actualmente, son 15 créditos.

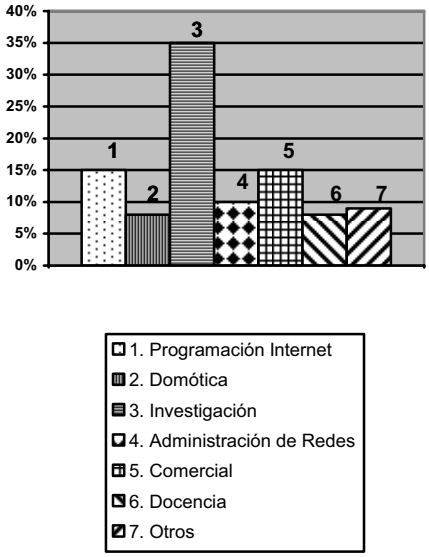


Figura 2. Interés global por las salidas profesionales propuestas.

En la figura 2 se presenta como resumen el interés por parte de todos los colectivos, de las salidas profesionales propuestas siendo, curiosamente, la investigación la que predomina.

3.3 Adaptación al EEES

Con lo que respecta al EEES, los alumnos de los ciclos formativos piensan que se ha de valorar el tiempo que se invierte en casa como horas de dedicación a sus estudios. Esta cuestión se les plantea por la propia definición de los créditos ECTS [1][5]. Respecto a la calidad de la

educación, cuando entre en vigor la reforma, la mayoría no contesta por desconocimiento y los que lo hacen creen que no se mejorará.

Por su parte, los alumnos de bachiller creen que es importante que se valore el tiempo invertido fuera del horario lectivo ya que también cuenta como trabajo dedicado a una asignatura. Una parte piensan que con la adaptación al EEES no se mejorará la calidad de la educación aunque la mayoría de este colectivo no sabe exactamente en qué consiste la reforma y no contesta.

En cuanto a los propios alumnos de las titulaciones en informática admiten que es de gran importancia la posibilidad de que se puedan cursar asignaturas en otros países de Europa pero sobretodo en países que les permita perfeccionar el inglés. Además consideran importante que se valore el tiempo invertido en casa dedicado al estudio y al desarrollo de las prácticas. Al contrario que el grupo anterior (alumnos de ciclos formativos y de bachiller), los alumnos universitarios de informática, confían que con la reforma al EEES sí que subirá el nivel educativo y mejorará su capacidad competitiva respecto a otros estudiantes europeos.

Los ingenieros titulados no consideran que tener en cuenta el tiempo invertido en casa para trabajar sea una ventaja, ya que es muy subjetivo y hará que los alumnos trabajen menos; finalmente esto redundará una disminución del nivel académico. Afirman rotundamente, que el problema está en la enseñanza en secundaria obligatoria (ESO).

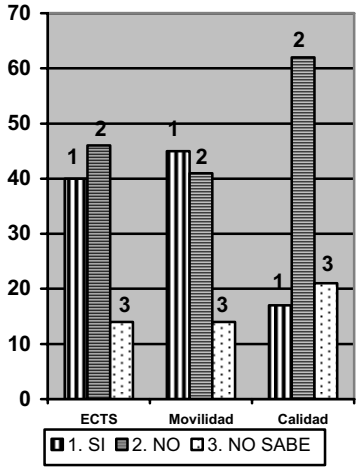


Figura 3. Adaptación al espacio europeo. Las columnas que acompañan a "ECTS", representan los porcentajes a favor de los créditos europeos. Las que acompañan a "Movilidad", representan los porcentajes a favor de cursar asignaturas en el extranjero. Las que acompañan a "Calidad", representan los porcentajes sobre sí aumentará la calidad de educación con el cambio al EEES.

En la figura 3 se puede observar los porcentajes de aceptación de los cambios que introducirá el EEES, representados por los nuevos créditos europeos, la facilidad para la movilidad y el incremento de la calidad en la enseñanza. Como posible conclusión destacable se puede ver cómo, en general, no se opina que los cambios vayan a mejorar la calidad de la enseñanza.

4. CONCLUSIONES

Por lo que se refiere a las asignaturas preferidas para cursar por los encuestados, hay unanimidad en las relacionadas con programación web y domótica. Basándonos en estas afirmaciones, propondríamos que en los nuevos planes de estudios de informática, se introdujera la domótica en el apartado de "Aspectos profesionales de la Ingeniería Informática". En cuanto a la programación web, ya se tiene en cuenta en las directrices para los nuevos planes de estudio.

La mayoría de los encuestados opinan que gran parte de los conocimientos que se estudian en la carrera están desfasados; los instrumentos con los que se trabaja son obsoletos y esto dificulta el

acceso al mundo laboral. Por otro lado, muchos encuestados hacen diferentes alusiones en sus comentarios a la baja valoración social de las titulaciones de informática.

En cuanto a las perspectivas profesionales, los encuestados piensan que los conceptos adquiridos durante la carrera son útiles, aunque no suficientes para enfrentarse al mundo laboral. Se consideran con menos capacidades a la hora de competir por un puesto de trabajo frente a otros titulados europeos, debido principalmente a la dificultad con el idioma. La conclusión sería que no se sienten en peores condiciones por la educación en sí, sino principalmente por el idioma.

Por otra parte, los colectivos estudiados muestran mayor interés en salidas profesionales relacionadas con la docencia, la investigación y las distintas áreas relacionadas con Internet.

Otro aspecto fundamental relacionado con el EEES es que todos los colectivos implicados, consideran de gran importancia la posibilidad de cursar asignaturas en otros países del espacio europeo, sobretodo en países de habla anglosajona (aunque esto puede estar influenciado por las reiteradas necesidades de aprendizaje de inglés). Con los nuevos planes de estudio se pretende favorecer el intercambio entre universidades europeas, tanto de reconocimiento de créditos como de movilidad curricular, por lo que esta demanda de los encuestados quedaría cubierta. Para finalizar, cabe destacar que todavía existe un importante desconocimiento general sobre los cambios que introducirá el EEES.

Referencias

[1] Acción de convergencia de la Universidad de Alicante. www.eees.ua.es/acciones_convergencia.htm

[2] Ficha técnica de propuesta de título universitario de grado según RD 55/2005, de 21 de enero. Enseñanzas de grado en Ingeniería informática. www.upc.edu/eees/contingut/arxius/WebEEES_DirectriusInformatica.pdf

[3] Fundación estatal para la gestión de la calidad en las universidades europeas. www.aneca.es/present/que_es.html

[4] Instituto universitario de desarrollo y cooperación. www.ucm.es/info/IUDC/guia/OCDE.htm

[5] Ministerio de educación. http://www.mec.es/universidades/eees/index.html

[6] Proyecto EICE. El Libro Blanco sobre las titulaciones universitarias de informática en el nuevo EEES. www.uv.es/%7Eoce/web%20castellano/Libro%20Blanco%20EICE.pdf

[7] UNESCO. Oficina internacional de educación. http://portal.unesco.org/es/

[8] Web universia relacionada con el EEES. http://eees.universia.es/documentos/otros/index.htm

Robótica e informática industrial
Arquitectura de ordenadores

Una experiencia practica de aprendizaje basado en proyecto en una asignatura de robótica

Nourdine Aliane, Sergio Bemposta, Javier Fernández, Verónica Egido

Departamento de Arquitectura de computadores y Automática
Universidad Europea de Madrid,
Villaviciosa de Odón, s/n (28670) Madrid (Spain)
nourdine.aliane@uem.es

Resumen

En este trabajo se presenta una experiencia de introducción de la metodología de aprendizaje basado en proyecto (ABP) en una asignatura de robótica. La metodología de ABP ha sido implementada de forma parcial y solo afecta a los créditos prácticos de la asignatura. Un solo proyecto ha sido propuesto para toda la clase y consiste en el desarrollo de un sistema de juego de damas contra un robot manipulador. Es un proyecto multidisciplinar e integra varias tecnologías: la robótica, la visión por ordenador y la inteligencia artificial.

En este articulo, en primer lugar, se presentan el entorno académico y los condicionantes para la introducción de la metodología ABP. Seguidamente, se describen los elementos claves para la elección del proyecto así como la descripción de algunos detalles técnicos del mismo. A continuación, se detallan las acciones realizadas para llevar a la práctica la metodología ABP. Finalmente se presenta una valoración cualitativa de nuestra experiencia.

1. Introducción

En el nuevo escenario educativo que resulta del proceso de armonización del Espacio Europeo de Educación Superior [6], se pone de manifiesto la necesidad de un modelo de enseñanza-aprendizaje centrado en el alumno y se plantea claramente la necesidad de desarrollar en el alumno no solamente habilidades específicas al campo de conocimiento al que estén orientados los planes de estudios, sino también otras competencias complementarias y generales muy demandas en el mundo profesional.

En este contexto, la educación superior se está adaptando a las nuevas circunstancias realizando cambios en el paradigma educativo a través de diversas iniciativas de innovación docente basadas en las metodologías docentes activas como el aprendizaje basado en problemas, método el caso, aprendizaje basado en proyectos, etc.

En nuestro caso, queremos introducir una acción docente activa en una signatura de robótica de quinto de informática marcando como objetivos elevar la motivación de los alumnos, mejorar su aprendizaje y desarrollar competencias generales como el trabajo en equipo, la innovación y la iniciativa, etc. Para esto, pensamos que la metodología de aprendizaje basado en proyectos (ABP en adelante) es la más apropiada.

Al ser una metodología completamente nueva para nosotros, hemos optado por una implantación parcial de la misma, y que la metodología ABP solo afecta a los créditos prácticos de la asignatura. Por otro lado, hemos optado en proponer un solo proyecto para toda la clase, el cual consiste en el desarrollo de un sistema del juego de damas contra un robot manipulador.

El resto del artículo se organiza de la siguiente forma: en la sección 2 se darán las claves de la metodología ABP. La sección 3 establece el entorno docente y los diferentes condicionantes de la puesta en marcha de ABP. En la sección 4, se formulan las razones que avalan la elección de nuestro proyecto de apoyo a la metodología ABP. En la sección 5, se explican los pasos más importantes en la implantación de metodología ABP. La sección 5 recoge la valoración de nuestra experiencia de forma cualitativa, y finalmente, la sección 6 concluye este trabajo.

2. La metodología ABP

La metodología ABP se empezó a aplicar a finales de los 70 en la enseñanza de la medicina en la Universidad de McMaster, Canadá [11] para combatir el problema de desmotivación de los estudiantes. Desde entonces, esta metodología ha ido ganando adeptos y actualmente se considera especialmente adecuada para abordar muchos de los retos de la educación superior [16]. Su aplicación en el campo de la informática llega más tarde, pero ya se considera como una metodología madura en este campo [3], [4].

El proceso de aprendizaje en la metodología ABP [14] se basa en el desarrollo de un proyecto que establece una meta como producto final. Su consecución exigirá el aprendizaje de conceptos técnicos y de actitudes. La metodología ABP solo estará en sintonía con los objetivos del EEES si el alumno toma un papel importante en el desarrollo del proyecto, y por ende, en el proceso de aprendizaje en el que estará inmerso.

Las características más relevantes de la metodología ABP son:

- La metodología del ABP se desarrolla en un entorno real y experimental. Esta circunstancia ayuda a los alumnos a relacionar los contenidos teóricos con el mundo real, y esto recae en la mejora de la receptividad para aprender los conceptos teóricos.
- El alumno tiene un papel activo en el proyecto y marca el ritmo y la profundidad de su propio aprendizaje.
- El ABP motiva a los alumnos, por lo que se puede tomar como un instrumento ideal para mejorar el rendimiento académico de los alumnos y su persistencia en los estudios.
- El ABP crea un marco ideal para el desarrollo de varias competencias generales como el trabajo en equipo, la planificación, la innovación y la creatividad, la iniciativa, etc.

2.1. ABP y la robótica

La utilización de la tecnología como medio en la docencia no se limita a la utilización del ordenador y a los medios multimedia clásicos, sino estamos asistiendo a la utilizando de los robots a modo de herramienta educativa y está ganando cada vez más popularidad [5]. En efecto, en muchos casos, los robots están presentes en el aula no con el fin de enseñar la disciplina de la robótica propiamente dicha, sino aprovechar su carácter multidisciplinar para activar procesos cognitivos que propicien un aprendizaje significativo y un acercamiento al mundo de la ciencia y la tecnología. El sistema de LEGO MINDSTORM es un buen ejemplo y está siendo utilizado a gran escala no solamente en la educación, sino también en proyectos de investigación [7], [8].

En general, un proyecto de robótica integra varias tecnologías y, en consecuencia, propicia un aprendizaje multidisciplinar. Es más, el reto que supone para un alumno diseñar y construir un sistema real, le permite adquirir los conceptos con cierta profundidad y le permite entrar en una dinámica de aprendizaje autónomo y el desarrollo de varias competencias como la iniciativa y la innovación. Estas características hacen que los proyectos de robótica sean muy apropiados en la implantación de la metodología de ABP [10]. En los trabajos [9], [13], [12] y [15], podemos encontrar varias experiencias relacionadas con la utilización de los robots en el marco de ABP.

3. El entorno docente

La asignatura de robótica es una asignatura optativa del plan de estudios de ingeniero en informática. Es cuatrimestral y se cursa en el quinto curso. Es una asignatura especialmente pensada para ingenieros en informática y su contenido está relacionado con los aspectos computacionales como la implementación de los algoritmos del control cinemático, la planificación de trayectorias, la programación de los robots y la integración de otras tecnologías como la visión por computador o la inteligencia artificial.

El carácter multidisciplinar de la asignatura hace que sea relativamente fácil proponer proyectos de cierta complejidad, y por lo tanto, ofrece condiciones idóneas para introducir la metodología ABP. Por otro lado, la implantación por primera vez de una metodología nueva, y sin contar con la experiencia previa de otros, nos ha planteado varias dudas. Ante esta situación, hemos optado por una implantación parcial de la metodología ABP. Esto nos lleva a definir un modelo mixto que consiste en dividir la asignatura en dos bloques. El primer bloque consta de contenidos teóricos e impartidos siguiendo la enseñanza tradicional. El segundo bloque es

práctico y enfocado al desarrollo de un proyecto dentro del marco de la metodología ABP, y esta parte representa aproximadamente el 25% la carga lectiva de toda la asignatura.

El proyecto y su alcance técnico constituyen el elemento central del proceso de aprendizaje, y por lo tanto, su elección no es trivial. Para garantizar el éxito de la implantación de la metodología ABP, hemos optado por proponer un solo proyecto para toda la clase, y permitir así, un ambiente de aprendizaje colaborativo.

4. El proyecto

El proyecto de apoyo a la metodología ABP propuesto para el conjunto de la clase consiste en el desarrollo de una aplicación de juego de damas contra un robot manipulador. Es un proyecto multidisciplinar e integra contenidos de robótica, de visión por ordenador y de inteligencia artificial. El carácter lúdico del proyecto permite mejorar sustancialmente la motivación de los alumnos, lo cual supone un valor añadido a nuestra experiencia. La elección del proyecto se apoya principalmente en los siguientes criterios:

- El proyecto es perfectamente realizable con los recursos hardware y software disponibles en nuestros laboratorios.
- Es un proyecto multidisciplinar e integra varias tecnologías que tienen una relación directa con varias asignaturas optativas de la titulación como la visión por computador o la inteligencia artificial.
- Es un proyecto con una cierta complejidad y se puede dividir en varios sub-proyectos. Esto nos permite crear varios pequeños grupos de trabajo y reforzar el desarrollo de algunas competencias.
- Es un proyecto abierto y admite varias soluciones. Los alumnos tendrán que buscar el mejor compromiso entre la sencillez de las propuestas y el tiempo necesario para su realización.
- El profesor puede brindar un asesoramiento optimo ya que la misma aplicación ha sido desarrollada con anterioridad por el equipo del profesor [1], [2].

4.1. Los aspectos técnicos del proyecto

El proyecto de "*un robot que juega a las damas*" consiste en desarrollar una aplicación que permite a una persona jugar a las damas contra un robot. Es un sistema que involucra un robot manipulador de tipo SCORBOT ER-IV y un sistema de visión por ordenador basado en el software MIL de MATROX. La siguiente fotografía, se puede ver el aspecto general del robot jugando a las damas.

Figura 1. Un robot jugando a las Damas

El proyecto pretende desarrollar e integrar varios elementos software como la captura y procesamiento de imágenes, el control del robot, el motor de juego, las comunicaciones entre el controlador del robot y el ordenador central, etc. En la siguiente figura, podemos ver los diferentes módulos de nuestra aplicación, así como las interacciones entre si.

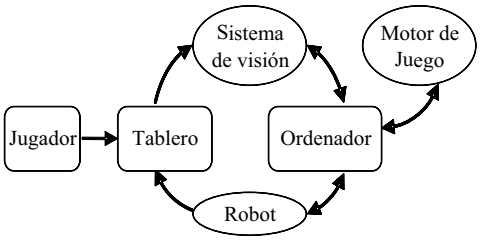


Figura 2. La interacción entre los subsistemas

La operatividad de la aplicación es cíclica que empieza con la captura de la imagen del tablero y su análisis determina la situación actual del juego. A continuación, el motor de juego basado en la estrategia Mini-max calcula las mejores jugadas. Finalmente, el robot manipulador culmina la ejecución de las jugadas. El lector puede encontrar

todos los detalles técnicos de la aplicación en las referencias [1] y [2].

5. Desarrollo de la metodología ABP

En este apartado, se van a presentar los pasos y aspectos más relevantes en lo se refiere al desarrollo de la metodología docente ABP.

5.1. La propuesta

La idea de realizar cambios en la asignatura con respecto de otros años integrando el proyecto de "*un robot que juega a las damas*" en la asignatura ha sido bien recibido por el conjunto de la clase.

En la presentación oficial del proyecto, el profesor ha hecho hincapié en que se va a desarrollar un método alternativo de aprendizaje y que el verdadero proyecto es el aprendizaje a través de un proyecto. Aún así, los alumnos han demostrado su plena predisposición para esta nueva metodología. Se ha informado correctamente a los alumnos haciendo una breve descripción de la metodología ABP dando sus principales características, y presentado la metodología como una alternativa de aprendizaje que merece la pena de explorar.

5.2. Tormenta de ideas

El desarrollo del proyecto empezó con la organización de un debate en forma de tormenta de ideas moderada por el profesor y que formula preguntas al conjunto de la clase para alentar y encauzar el debate. Todas las propuestas y sugerencias son aceptadas y registradas.

Al final de esta primera sesión, se han acordado los siguientes puntos:

- Se han identificado cuatro (04) partes independientes del proyecto, por lo que se ha alcanzado un acuerdo de dividir el proyecto en 4 sub-proyectos.
- Se ha fijado de forma aproximada los objetivos técnicos que se deben alcanzar.
- Se han inventariado los recursos necesarios para el desarrollo del proyecto.
- Se conformaron los grupos necesarios (3-4 alumnos) para la realización de los sub-proyectos.
- La asignación de las tareas fue consensuada y se ha hecho con el criterio de afinidad a los proyectos de fin de carrera que estaban desarrollando los alumnos.
- El primer objetivo de cada grupo ha sido la elaboración de un ante-proyecto para definir con claridad los objetivos, los hitos y una planificación de las tareas.

5.3. Método de trabajo

La clase de 14 alumnos se ha organizado en 4 grupos de (3-4) alumnos. Cada grupo tiene su sub-proyecto y han evolucionado en un ambiente de aprendizaje colaborativo; es decir, todos los integrantes han intervenido en todas y cada una de las tareas programadas en su proyecto. Por otro lado, cada grupo ha designado a un delegado para cooperar con el resto de los grupos en los aspectos de intercambio de información entre los todos los módulos y su integración en la aplicación final.

El carácter abierto del proyecto admite diversos enfoques y varias soluciones. Esto ha originado un verdadero debate sobre las posibles soluciones y se ha creado un entorno favorable para el desarrollo del trabajo en equipo, la innovación y la creatividad.

El rol del profesor también es importante y no debe limitarse a observar a los estudiantes, sino, tiene que ser un catalizador del proceso de aprendizaje. Debe crear una atmósfera de confianza y fomentar la colaboración de todos los alumnos. Es importante que el profesor ceda el liderazgo del proceso y hacer que los propios alumnos sean los protagonistas del proyecto. Pero, no significa aislarse, sino tiene que estar presente, observar y ayudar a valorar las iniciativas de los alumnos. Si bien la metodología permite una cierta libertad a los alumnos, el profesor debe conocer las decisiones que se toman para poder corregirlas a tiempo y encauzar a los alumnos de nuevo para no alejarse demasiado de los objetivos marcados.

La duración del proyecto ha sido de 8 semanas, dedicando 2 horas semanales en horas laboratorio y en presencia del profesor. Por su parte, los alumnos han invertido más tiempo en las diferentes tareas de búsqueda de la información, en la asimilación de conceptos nuevos, en la realización de pruebas parciales, etc.

5.4. La sesión final

Al finalizar el proyecto, cada grupo ha tenido que entregar una memoria recalcando los aspectos técnicos más relevantes de su trabajo, y han tenido

que realizar una comunicación oral para explicar y divulgar los logros alcanzados.

Además, se ha reservado la última sesión del laboratorio para realizar una valoración técnica del proyecto y una valoración cuantitativa del proceso de aprendizaje. Los alumnos y profesor comentan y discuten conjuntamente los resultados conseguidos. La valoración técnica consistió en realizar varias demostraciones del sistema con la participación de varios alumnos ajenos a la clase. La valoración del método de aprendizaje se ha hecho en forma de debate para recoger las opiniones de los alumnos en torno a la metodología de APB.

5.5. Método de evaluación

Unos de los aspectos más difíciles de la metodología ABP es la concepción de un sistema de evaluación que sea justo. En el desarrollo de un proyecto, la evaluación se vuelve difusa ya que no se sabe exactamente que es lo que se tiene que evaluar: los conocimientos adquiridos por cada alumno o su desempeño y su aportación al proyecto, y cómo reflejar la evaluación individualizada de cada alumno que han evolucionado dentro de un grupo.

En esta primera experiencia, no hemos acordado mucha importancia a este aspecto, y nos hemos limitado en otorgar una nota simbólica que no influye en la nota global de la asignatura, y que ha tratado de reflejar la opinión del profesor sobre el desempeño de cada alumno.

6. Valoración de la experiencia

La valoración de nuestra experiencia es cualitativa basándose en la opinión del profesor y de los alumnos involucrados en la experiencia.

El profesor considera que enseñar utilizando la metodología ABP es estimulante y que es una experiencia gratificante. El nivel de entendimiento entre el profesor y los alumnos es muy elevado y se crea un entorno de aprendizaje marcado por la predisposición total de los alumnos. En términos de objetivos docentes, las actividades desarrolladas en el marco del proyecto han permitido alcanzar un nivel de aprendizaje claramente mayor que el aprendizaje que hemos ido desarrollando con las clásicas practicas de laboratorio.

Por oro lado, los alumnos ponen de manifiesto su satisfacción por la formación recibida a través de la metodología ABP y perciben claramente que han adquirido una experiencia investigadora por su propia cuenta. Se ha valorado positivamente el nivel de motivación y de estimulación que les ha producido la experiencia.

En lo que se refiere a las competencias, los alumnos afirman que el proyecto les ha permitido adquirir una experiencia valiosa de trabajo en equipo con sus compañeros, que el proyecto les ha permitido tener sus propias iniciativas, y que no han tenido que seguir ningún guión. Finalmente, varios alumnos han afirmado que les hubiese gustado haber cursado más asignaturas siguiendo la metodología ABP.

Como consideración final, el proyecto en la metodología de ABP es en realidad un instrumento de aprendizaje. Sin embargo, los alumnos desconocen las reglas básicas de gestión de proyectos. Para optimizar el aprendizaje y el desarrollo de las competencias antes mencionadas, pensamos que el profesor tiene que proporcionar un guión de trabajo en forma de Portafolio, donde se marcan las pautas a seguir en el desarrollo del proyecto y recalcar el cumplimiento con algunas obligaciones como la entrega de resultados parciales o la exposición oral de los avances del proyecto.

7. Conclusión

En este trabajo, hemos presentado una experiencia práctica de implantación de la metodología docente de aprendizaje basado en proyectos en una asignatura de robótica.

Al ser una metodología nueva para nosotros, hemos optado por una implantación parcial y que la metodología ABP solo afecta a la parte práctica de la asignatura. La parte teórica ha sido impartida siguiendo el método de enseñanza tradicional.

Un solo proyecto ha sido propuesto para toda la clase, el cual consiste en la realización de un sistema titulado "*un robot que juega a las damas*". Su carácter multidisciplinar y su complejidad nos han permitido dividirlo en varios sub-proyectos y estructurar la clase en grupos reducidos. El desarrollo del proyecto ha creado un marco ideal para el aprendizaje colaborativo y su aspecto lúdico ha permitido mantener un nivel de motivación muy alto.

El desarrollo del proyecto ha permitido a los alumnos experimentar un aspecto novedoso directamente relacionado con las competencias específicas como es la integración de varias tecnologías. El nivel de aprendizaje ha sido claramente mayor que el aprendizaje basado las tradicionales practicas de laboratorio. Por otro lado, el alumno ha tomado conciencia de los aspectos positivos y formativos relacionados con las competencias generales como el trabajo en equipo, la planificación, la innovación o la iniciativa. En definitiva, los alumnos se han sentido como verdaderos ingenieros que han participado en la concepción de un sistema complejo.

En conjunto, la experiencia ha sido positiva y que ha sido acogida favorablemente por el conjunto de los alumnos.

Referencias

[1] Aliane, N, Bemposta, S, Gachet, D, *Robotics Lab Practices: Solving the Tower of Hanoi & Checker Playing Robots,* Weingarten, Germany, 2001.

[2] Aliane, N, Bemposta, S, Gachet, D, *Un Robot que Juega a las Damas,* XXIV Jornadas de Automática CEA-IFAC, León, 2003.

[3] Barg, M, *et al*, *Problem-Based Learning for Foundation Computer Science Courses*, Computer Science Education 10(2), pp1-20, 2000.

[4] Dart, P *et al*, *Enhancing Project-Based Learning: Variations on Mentoring,* Australian Software Eng. Conference, Proceedings pp. 112-117, 1996.

[5] Druin, A, & J. Hendler, *Robots for kids: Exploring new technologies for learning.* San Diego, CA: Academic Press, 2000.

[6] EEES (En línea)

[7] Gawthrop P, J *et al*, *A Lego Based Control Experiment,* IEEE Control Systems Magazine Vol 24 (5), pp 43-56, 2004

[8] Gawthrop P, J *et al, Using LEGO in Control Education,* Proceeding Advances in Control Education", pp 31-38, 2006

[9] Grimheden M & M. Hanson, *How might Education in Mechatronics benefit from Problem Based Learning,* 4th International Workshop on Research and Education in Mechatronics, Bochum, Germany, pp 211-218, 2003 http://www.crue.org/Bolet_educ_ESP25.htm

[10] Hung, D, *Situated cognition and ABP: Implications for learning and instruction with technology*. Journal of Interactive Learning Research, 13(4), pp-393-414, 2002.

[11] McMaster (En línea) University 2001 *Problem-Based-Learning*: http://www.chemeng.mcmaster.ca/pbl/pbl.htm

[12] Mingyang G, *A Case to Do Empirical Study Using Educational Projects,* Journal of: Issues in Informing Science and Information Technology, Vol pp-509-520, 2004

[13] Piguet Y, *et al, Hands-On Mechatronics: Problem-Based Learning for Mechatronics.* IEEE Int. Conference on Robotics & Automation, Washington D.C., USA, 2002

[14] Solomon & Gwen, *Project-Based Learning*: Technology and Learning, 23(6), pp 20-30, 2003.

[15] Spong, M W, *Project Based Control Education,* Proceeding in Advances in Control Education, pp 40-47, 2006

[16] Woods, D. R *et al*. *The future of engineering education. Developing Critical Skills*. Chem. Engr. Education, 34 (2), pp-108-117, 2000

Análisis de tecnologías sw para laboratorios remotos

Javier García Zubía *, Pablo Orduña *, José María Sáenz Ruiz de Velasco **, Inés Jacob Taquet **, Jesús Luis Díaz Labrador ** y Javier Oliver Bernal **
* Dpto. de Arquitectura de Computadores, Automática y Electrónica y Telecomunicaciones
** Dpto. Ingeniería del software
Universidad de Deusto
Avda. Universidades 24, 48007, Bilbao, España
zubia@eside.deusto.es

Resumen

Los Laboratorios Remotos o WebLab son ya un recurso didáctico de primer orden en las facultades de ingeniería, sin embargo en muchos casos adolecen de un pobre diseño sw, tanto en el cliente como en el servidor, lo que degrada su calidad y su utilidad académica. El presente trabajo analiza y selecciona las mejores tecnologías para implementar el cliente y el servidor de un WebLab.

1. Introducción

Actualmente los WebLabs han demostrado sobradamente su utilidad académica no tanto para sustituir a los laboratrios presenciales, como para complementarlos y potenciarlos. En una primera etapa, los WebLabs estaban organizados y promovidos por un laboratorio o departamento, pero su éxito ha conllevado que deba ser la propia universidad la encargada de ofrecer este servicio. Este cambio supone un reconocimiento para los WebLabs, pero también nuevos requisitos (seguridad, accesibilidad, universalidad, etc.) que generalmente no son tomados como esenciales al inicio del diseño, pero que son los que conforman un servicio profesional. Un planteamiento incorrecto es diseñar primero un prototipo que funcione –it runs- y luego añadirle otras funcionalidades, desgraciadamente este planteamiento no es válido y muchas veces acaba en que hay que rehacer la totalidad de la aplicación. Esto es fácilmente asumible por un profesional de la informática, pero no es tan evidente en otros casos.

Por ejemplo, en el mes de noviembre del 2006 la Universidad de Deusto organizó un workshop invitando a una decena de investigadores en el área de los laboratorios remotos [1]. Buena parte de las dicusiones se centraron en los aspectos hw y académicos, pero sin embargo quedó patente que no era lo mismo desarrollar en Java, que en Adobe o en AJAX, y que una mala elección inicial de la tecnología lastraba la calidad del laboratorio remoto, sobre todo en aspectos esenciales como la universalidad, la seguridad y la accesibilidad. Las razones son que mayoritariamente los investigadores tienen un perfil electrónico o de regulación y que las tecnologías web 2.0 son de reciente aparición.

Otro ejemplo clarificador se obtiene al leer el imponente trabajo [2]. En él se analizan más de un centenar de artículos centrados en laboratorios, pues bien, solo uno de ellos [3] relaciona sw y laboratorios remotos, centrándose el resto en el hw y en los aspectos académicos.

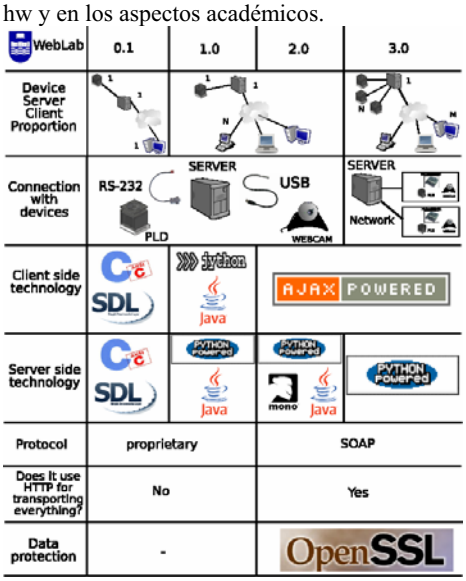


Figura 1. Evolución tecnológica del WebLab-Deusto.

El presente trabajo aprovecha la experiencia del equipo investigador desde el 2001 en el desarrollo de WebLabs (ver Fig. 1 y [4][5][6]) para analizar y seleccionar entre las tecnologías propias del cliente y del servidor la más adecuada para diseñar un laboratorio remoto. El trabajo describe las necesidades de un WebLab, las posibilidades de las diferentes tecnologías para el cliente –aplicaciones de escritorio, ActiveX, Java, Adobe Flash, AJAX y HTML– y las correspondientes al servidor –Python, .NET y Java.

Los apartados 2 y 3 describen y analizan las distintas tecnologías para implementar el cliente y el servidor de un WebLab, seleccionando justificadamente una para cada entorno, AJAX y Python, respectivamente. El apartado 4 refleja las conclusiones del trabajo.

2. Aplicación del cliente

El cliente en un laboratorio remoto es el sw que el usuario va a utilizar para acceder al servicio. Dependiendo del experimento, el cliente puede necesitar enviar un fichero, o recibirlo; puede necesitar ver por WebCam qué está pasando en el laboratorio; puede necesitar interactuar con el experimento; o puede necesitar otros servicios.

La parte cliente debería evitar cualquier restricción innecesaria en el usuario más allá de las funcionalidades, es decir, cualquier alumno con cualquier PC, SO y navegador debe poder acceder al WebLab, ya que lo contrario es difícilmente asumible por la universidad (aunque sí podría serlo por el laboratorio). Así pues, un cliente es mejor cuanto más "fino" es, cuanto más SO soporta, cuanto más accesible es, cuanto menos dependiente es de plug-in, etc. Los WebLabs deben ser herramientas didácticas universales.

2.1. Tecnologías del cliente

Actualmente existe un abanico muy amplio de tecnologías que pueden ser utilizadas para implementar el cliente de un laboratorio remoto, desde la más ligera aplicación web hasta la más pesada aplicación de escritorio. Todas ellas pueden ser clasificadas en dos grupos:

- Aplicaciones de escritorio. Aquellas que se ejecutan en el escritorio del ordenador del usuario.
- Aplicaciones web. Aquellas que son ejecutadas en el navegador del escritorio del usuario.

Una aplicación de escritorio puede ser desarrollada en muchas plataformas (C, C++, Delhi, Java, .NET, Python, etc) y tener muy pocas restricciones, pero es poco portable, más intrusiva que las aplicaciones web, ya que usualmente tienen acceso a todo el sistema del usuario, y necesita de un proceso de instalación. Estas desventajas se compensan con una mayor potencia y complejidad, ya que al no tener las restricciones propias de un navegador y de los estándares asociados a él, estas aplicaciones pueden explotar por cuenta propia recursos como 3D, protocolos binarios específicos para la aplicación, etc. En cualquier caso la calidad y seguridad de una aplicación de escritorio recae totalmente en su diseñador y programador, y esto desde el punto de vista del administrador de sistemas de la universidad suele ser totalmente inaceptable a no ser que ellos hayan sido los desarrolladores; no se debe olvidar que este departamento es el encargado de asegurar y ser responsable de la integridad del servicio informático de la universidad, y que un mal diseño puede comprometerla.

Atendiendo a las razones anteriores, el trabajo se va a centrar principalmente en las aplicaciones web, que junto a las aplicaciones de escritorio, pueden reclasificarse en:

- Aplicaciones intrusivas. Estas aplicaciones tienen los mismos privilegios que las aplicaciones del usuario, por ejemplo pueden acceder al disco duro, pueden leer y escribir ficheros, pueden ejecutar programas, pueden abrir y cerrar conexiones, etc.
- Aplicaciones no intrusivas. Estas aplicaciones garantizan al usuario que pueden ejecutadas sin poner en riesgo su seguridad, ya que no pueden hacer nada que no pueda hacer el navegador en el que están siendo ejecutadas.

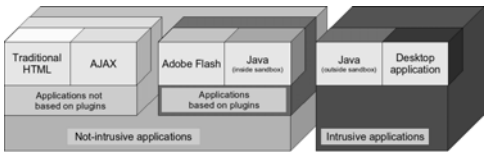


Figura 2. Clasificación de las tecnologías del cliente

Viendo la Fig. 2 se puede decir que cuanto más potente es una tecnología, menos universal y más intrusiva es. Conceptualmente, la mejor tecnología es aquella que cubriendo todos los requisitos del sistema, más universal sea. El resto del trabajo va a analizar las diferentes estrategias para acabar seleccionando AJAX, ya que aunque no es la más potente sí es la más universal, y en el caso de un servicio académico su "universalidad" es un requisito irrenunciable.

Aplicaciones de escritorio

Son las más intrusivas y las que más comprometen la seguridad del usuario. Específicamente las tecnologías Java y .NET pueden ser no intrusivas bajo control del usuario [7] [8]. En cualquier caso su calidad depende claramente del equipo de desarrollo, y por tanto no serán tenidas en cuenta en el análisis final.

ActiveX

Java y ActiveX son probablemente, entre las tecnologías de aplicaciones web, los sistemas más potentes en términos de flexibilidad, pero ActiveX solo está disponible bajo Internet Explorer y sus aplicaciones son por defecto intrusivas, aunque el cliente debe aceptar la ejecución de sw intrusivo. Estas características hacen que las aplicaciones en ActiveX estén más cerca de las aplicaciones de escritorio que de las aplicaciones web.

Java applets

Java es una plataforma potente para desarrollar Rich Internet Applications, RIA. Para poder usar Java, el cliente debe tener instalada la máquina virtual de Java, JRE [9] [10].

Una buena característica de Java es que puede ser instalada en diferentes sistemas operativos, y puede ser embebida en muchos navegadores. La desventaja de JRE es su disponibilidad, que si bien era mucha hace tres o cuatro años, ahora está empezando a decrecer, y no todos los ordenadores la tienen instalada. Otra desventaja es que si el WebLab ha sido desarrollado para JRE 1.5, no funcionará para JRE 1.4, y el usuario tendrá que actualizar la máquina virtual, pero además, y aun siendo extraño, una aplicación hecha bajo JRE 1.3 puede no funcionar bajo JRE 1.5, todo esto exige al usuario tener disponibles varias máquinas virtuales y saber bajo cuál debe ejecutar cada aplicación, y/o exige a los desarrolladores tener diferentes versiones de un mismo WebLab para las diferentes JRE, con el problema de mantenimiento que esto supone.

Es interesante volver sobre la disponibilidad de la JRE, ya que si no está instalada y el usuario está en un cibercafé o en un ordenador universitario sin privilegios de administrador, no podrá instalar la máquina virtual y por tanto no podrá acceder al laboratorio remoto, quedando degradada su universalidad.

Por último cabe destacar que una aplicación Java no es inicialmente intrusiva ya que se ejecuta en la sand box, pero si el cliente debe acceder a algún fichero, entonces la aplicación deberá "abandonar" la sand box, convirtiéndose en una aplicación intrusiva, perdiendo una de sus principales ventajas. La solución a este problema podría implementarse en HTML o JavaScript dentro de la aplicación Java, pero esto complicaría innecesariamente el desarrollo y mantenimiento del sistema.

Adobe Flash

Adobe Flash [11] es actualmente la tecnología líder en RIA. El usuario de una aplicación Adobe Flash debe tener instalado el Adobe Flash Player, el cual interpretará los ficheros en formato swf. Una vez instalado el Adobe Flash Player, las aplicaciones desarrolladas serán no intrusivas, multiplataformas y muy potentes: vídeo, vídeo en tiempo real, sonido, acceso no intrusivo a ficheros, uso de ActionScript, acceso a servicios web, etc. La combinación del potencial de Adobe Flash en servicios web y en animación hacen de esta tecnología una de las más adecuadas para implementar laboratorios remotos.

El uso de Adobe Flash está muy extendido y está disponible en Windows, Linux y Mac OS [12]. En cualquier caso esta disponibilidad es relativa ya que todavía no está disponible para 64 bits, lo que es una clara desventaja. Además, la versión 7 es la única disponible para Linux hasta mediados de enero de 2007, mientras que para Windows ya está desarrollada la versión 9 [13]. Otro problema importante es que solo tiene un gran distribuidor, Adobe, y es por tanto un sw propietario. Por ejemplo, en diciembre de 2006 fue descubierto un error de seguridad en Adobe Reader que comprometía gravemente a Windows [14], de tal forma que si el *site* tenía disponible un pdf, la sesión del cliente podía ser robada. Esto mismo podría pasar con el Adobe Flash Player y

quedar remarcado por la posición aislada de Adobe.

AJAX

En los últimos dos años la tecnología de referencia en RIA es AJAX [15]. AJAX es la combinación de varias tecnologías web ya existentes (XHTML, Javascript, CSS, DOM, etc) con un nuevo componente: XHTMLHttpRequest. Este componente permite llamar a servicios web XML asíncronamente desde Javascript. AJAX es el acrónimo de Asynchronous Javascript And XML.

El aspecto principal de AJAX es que todos los componentes, excepto XMLHttpRequest, son estándares que los navegadores ya soportan. Así, si un navegador soporta el nuevo componente, entonces cualquier aplicación cliente en AJAX se podrá ejecutar en dicho navegador del usuario. Esta característica es muy importante, y hace de AJAX la más portable y universal de las tecnologías explicadas hasta ahora, ya que esta característica no reside tanto en el SO como en el navegador que es el encargado de soportar los estándares. De esta forma, una aplicación implementada en AJAX es directamente ejecutable en un teléfono celular, PDA, etc. siempre que su navegador soporte XMLHttpRequest. Este es el caso del navegador Opera usado por muchos teléfonos celulares [16], el de las últimas versiones del Explorer para Windows CE y el del navegador en código abierto desarrollado por Nokia para sus últimos modelos. Por tanto, un laboratorio remoto implementado en AJAX es accesible desde una multitud de dispositivos, lo que no hace sino otorgarle más universalidad, que es uno de los principales requisitos de este tipo de aplicaciones.

Grandes empresas como Google o Yahoo han desarrollado algunas de sus más famosas aplicaciones en AJAX, por ejemplo Google Maps, Google Mail o Flickr, y por ello AJAX está siendo utilizado en diversidad de aplicaciones, lo que no hace sino aumentar su potencial.

La principal desventaja de AJAX es que no tiene recursos de vídeo y audio. Un laboratorio remoto con bajas exigencias de vídeo y sin audio, bien puede ser desarrollado solo en AJAX, pero si se necesita un buen rendimiento, la aplicación necesita integrar funciones implementadas en Adobe Flash, por ejemplo.

Aplicaciones tradicionales en HTML

Las aplicaciones HTML son aplicaciones web que solo usan estándares como HTML, XHTML, CSS, etc. Estas aplicaciones no tienen por sí mismas capacidad de interacción con el servidor, de vídeo, de audio, etc., lo que supone una clara desventaja en el caso de un WebLab, más allá de la ventaja que supone su perfecta integración en cualquier navegador.

Un aspecto destacable es que es posible desarrollar aplicaciones accesibles en HTML, lo que permite que personas discapacitadas puedan acceder a los servicios web así implementados. Lo anterior no es tan fácil en el resto de tecnologías, excepto para Adobe Flash, cuya versión 6 ayuda al diseñador con funciones para la accesibilidad [17]. Lo anterior puede parecer poco importante, pero los responsables de un laboratorio remoto y la propia universidad no pueden ignorar el alcance de las leyes que favorecen la integración de los discapacitados en la enseñanza [18] [19].

2.2. Análisis y selección de la tecnología del cliente

La pregunta que surge es, ¿cuál es la mejor tecnología para implementar el cliente de un laboratorio remoto? Teniendo claros los requisitos del WebLab a implementar, la pregunta sería ¿pueden cubrirse los requisitos con HTML? Si la respuesta es sí, entonces esta es la tecnología; si la respuesta es no, entonces se repite la pregunta para AJAX; y así sucesivamente para Adobe, etc. Es decir, se debe recorrer la Fig. 2 de izquierda a derecha hasta encontrar la tecnología adecuada para los requisitos.

La Tabla 1 puntúa de 1 a 5 las características de cada tecnología para implementar la aplicación del cliente. Las características utilizadas son:

- Paradigma: ¿Es la tecnología correspondiente el paradigma actual de diseño?
- Multiplataforma: ¿Es ejecutable la aplicación bajo cualquier SO?
- Intrusividad: ¿Hasta qué punto no es intrusiva la aplicación?
- Proveedores: ¿Cuántos proveedores tiene la tecnología de desarrollo?
- Instalación previa: ¿Require la aplicación la instalación previa de un sw, plug-in, máquina virtual, etc?
- Precio: ¿Es gratuita la tecnología?

- Dispositivos móviles: ¿Es ejecutable directamente la aplicación en un teléfono móvil, PDA, etc?
- Flexibilidad: ¿Es utilizable la tecnología en diferentes contextos?
- Accesibilidad: ¿Es adecuada la tecnología para desarrollar aplicaciones accesibles por discapacitados?
- Comunidad de desarrolladores: ¿Existe una comunidad de usuarios y desarrolladores detrás de la tecnología?
- Protocolos: ¿Son diversos y potentes los protocolos que soporta la tecnología?
- Herramientas de desarrollo: ¿Son potentes las herramientas de desarrollo?
- Estandarización: ¿Esta la tecnología basada en estándares?
- Ancho de banda: ¿Cuánto ancho de banda necesita la tecnología?
- Audio y vídeo: ¿Permite la tecnología el uso de audio y vídeo?
- Integración en el navegador: ¿Es la tecnología parte intrínseca del navegador?

Analizando los resultados de la Tabla 1:

- AJAX es la tecnología mejor valorada.
- Mirando solo a los aspectos más destacables, AJAX es también la tecnología mejor valorada (ver Tabla 2).
- Si la aplicación del cliente necesita vídeo y audio de calidad, al menos debe ser usado Adobe Flash.
- Si se necesita interacción con el hw, como es usual en un laboratorio remoto, entonces HTML debe ser descartado.
- Java Applets es similar a Adobe Flash en la mayoría de características, pero está menos disponible en términos de máquina virtual.
- ActiveX no es recomendable para desarrollar WebLabs porque no aporta ninguna ventaja que otras tecnologías no tengan, y sin embargo no es multiplataforma.

Para un WebLab específico, el equipo de desarrollo debe elegir de la Tabla 1 los requisitos más importantes, o añadir nuevos a la tabla. Por ejemplo la Tabla 2 muestra la comparación entre AJAX y Adobe Flash para el desarrollo del WebLab-Deusto. La tecnología más adecuada para el WebLab-Deusto es AJAX.

	Java Applets	Adobe Flash	AJAX	HTML	Active X
Paradigma	***	****	*****	****	*
Multiplataforma	** (1)	**** (2)	***** (3)	***** (3)	* (4)
Intrusividad	*****/* (5)	*****	*****	*****	*
Proveedores	***	*	*****	*****	*
Instalación previa	**	***	*****	*****	*
Precio	*****	*****/** (6)	*****	*****	*** (7)
Dispositivos móviles	** (8)	** (8)	**** (9)	****	** (8)
Flexibilidad	****	****	***	*	*****
Accesibilidad	**	****	**	*****	**
Comunidad de desarrolladores	*****	*****	*****	*****	*****
Protocolos	*****	*****	****	**	*****
Herramientas de desarrollo	*****	***	*****	*****	***
Estandarización	****	***	****	*****	**
Ancho de banda	*****	*****	***	**	*****
Audio y vídeo	***	*****	**	*	*****
Integración en el navegador	*	*	*****	*****	*** (10)
Suma	**56**	**57**	**65**	**64**	**45**

Tabla 1. Análisis de las tecnologías del cliente

1. Mientras la máquina virtual de Java está disponible bajo varios SO, no es posible asumir que esté siempre instalada, sobre todo la última.
2. Es común encontrar Adobe Flash Player instalado (más que la máquina virtual). En cualquier caso, no está disponible para arquitecturas de 64 bits.
3. Es totalmente asumible que todos los usuarios tienen instalados navegadores.
4. Solo se ejecuta bajo un único navegador, el Explorer, y en único SO, Windows.
5. Depende de si se trabaja en la sand box o no.
6. El Player es gratuito, pero el diseñador sí paga por el editor de Adobe, aunque ya hay alternativas gratuitas.
7. ActiveX exige Windows, que no es gratuito.
8. Con restricciones y dependiendo del dispositivo.
9. Es necesario usar navegadores con AJAX, como Opera o Nokia OSS Web Browser.
10 ActiveX es solamente parte integral del Explorer, no del resto de navegadores.

	Adobe Flash	AJAX
Paradigma	****	*****
Multiplataforma	****	*****
Intrusividad	*****	*****
Instalación previa	***	*****
Dispositivos móviles	**	****
Herramientas de desarrollo	***	*****
Audio y vídeo	*****	**
Integración en el navegador	*	*****
Suma	**27**	**36**

Tabla 2. Análisis de las tecnologías del cliente centrado en WebLab-Deusto

Las ventajas que AJAX tiene en términos de disponibilidad, independencia de un único proveedor, buena carga de trabajo e integración en el navegador, hacen de ella la tecnología más adecuada cuando la página web necesita interacción. La principal desventaja de AJAX para WebLabs es que no dispone de capacidad para vídeo y audio de calidad, que sí son aportadas por Adobe Flash o Java. Pero como Adobe y Java son interoperables con AJAX, entonces la integración de módulos desarrollados para audio y vídeo en Adobe Flash o Java en la aplicación es trivial. Por ejemplo, Google Mail es una aplicación AJAX que soporta conversaciones online y usa un módulo en Adobe Flash para generar sonidos cada vez que alguien manda un mensaje [20]. Si el usuario no dispone del Adobe Flah Player, todo el Google Mail funcionará con normalidad, excepto los sonidos de mensaje; esta misma situación se puede dar al diseñar laboratorios remotos.

2.3. Herramientas de implementación del cliente

Cada una de las tecnologías explicadas con anterioridad tiene asociadas al menos dos herramientas de desarrollo. Por ejemplo, para desarrollar aplicaciones de escritorio hay muchas herramientas para cada lenguaje, y en HTML lo mismo; lo mismo ocurre para desarrollar applets de Java; en peor situación está Adobe Flash, pero ya existen algunas herramientas distintas de la propia de Adobe; y finalmente hay docenas de librerías disponibles para integrar AJAX en diferentes plataformas de desarrollo web. Así pues, el equipo de desarrollo puede optar entre multitud de herramientas: Google Web Toolkit, OpenLazslo, AJAX.NET, AJAX para PHP, etc. Las dos primeras van a ser descritas en detalle.

Recientemente la plataforma de código abierto Google Web Toolkit [21][22] ofrece al programador la API con varios widgets y controles escritos en Java, y de hecho el programador desarrolla al completo la web en Java, pero finalmente Google Web Toolkit compilará el cliente en AJAX.

Otra herramienta de desarrollo muy interesante para RIA es OpenLazslo [23], que desarrollada en código abierto ofrece al programador la API en un lenguaje llamado LZX (el lenguaje de programación OpenLazslo consiste en un dialecto de XML que puede crear interfaces de usuario y aceptar métodos y retornos escritos en Javascript). El punto más interesante de OpenLaszlo (ver Fig. 3) es que desde la versión 4 soportará múltiples runtimes, así el diseñador describirá la aplicación en LZX y la compilará en Adobe Flash, en AJAX o en J2ME (Java para dispositivos móviles). Se espera que en el futuro el número de runtimes se incremente [24]. El problema es que de momento esta versión 4 no está disponible excepto como versión beta, el trabajo realizado permite ya compilar en AJAX, y Laszlo Systems ya está trabajando con Sun Microsystems para implementar el compilador en J2ME [25].

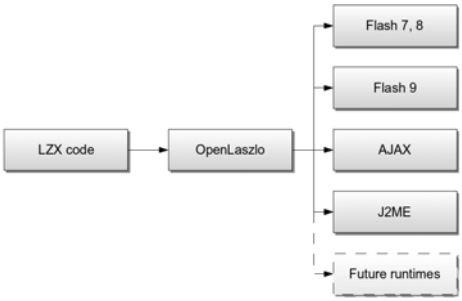


Figura 3. Plataforma OpenLazslo

Cuando la plataforma esté disponible, el equipo de diseño solo trabajará en un lenguaje, LZX, pero podrá obtener tantos clientes como desee, incluso clientes para distintas versiones de por ejemplo Adobe Flash, y así dar servicio a Microsoft, Linux, etc. Así la portabilidad no dependerá del equipo de diseño, sino de la plataforma, y por tanto siempre habrá un cliente para cada usuario, y no un cliente para todos los usuarios.

3. Aplicación del servidor

Aunque una parte muy importante del laboratorio remoto es la parte del cliente y las tecnologías asociadas a él, la mayor parte del esfuerzo y de la potencia residen en el servidor (el cliente es responsable de la universalidad). Si el servidor es bueno y el cliente es pobre, todo el sistema será pobre, pero si el servidor es pobre, no tendrá ninguna importancia la calidad del cliente, el sistema lo será.

El problema reside en que puede que una decisión tomada en el servidor, afecte al cliente, y viceversa. Este situación debe ser tenida en cuenta desde el principio, ya que va a afectar al correcto desarrollo del proyecto. Por ejemplo, si se elige Google Web Toolkit u OpenLaszlo, el equipo de desarrollo necesitará usar Java al menos en una parte del servidor, lo que podría hacer recomendable el usarlo en todo él.

Estas dependencias y la portabilidad no son un grave problema para el servidor, tanto como lo eran para el cliente, simplemente exigen al administrador del servicio utilizar un determinado entorno de explotación.

En cualquier caso un buen diseño en el servidor no depende de la tecnología usada, y hay muchas disponibles; no al menos en el mismo sentido que en el caso del cliente. La tecnología elegida puede hacer más fácil desarrollar un servidor escalable, seguro y mantenible, pero esto solo se logrará si el desarrollador sabe usar la herramienta. La Tabla 3 analiza tres de las tecnologías más usadas para implementar servidores.

	Python	.NET	Java
Multiplataforma	*****	* (1)	****
Precio	*****	** (1)	**** (2)
Comunidad de desarrolladores	*****	*****	*****
Herramientas de desarrolladores	***	*****	*****
Velocidad de desarrollo	*****	***	***
Librerías de servicios web	**	*****	*****
Lenguaje ***	*****	***	**
Robustez	***	****	****
Dinamismo ***	*****	****	****
Suma	**38**	**32**	**36**

Tabla 3. Análisis de las tecnologías del servidor

1. Hay una plataforma popular en código abierto desarrollada por Novell llamada Mono [26], la cual es compatible en muchos sentidos con .NET. Si el WebLab se ejecuta bajo Mono, el costo decrecerá y podrá ser utilizado en diferentes plataformas.
2. Depende de la herramienta y del framework usado.

La tecnología elegida para desarrollar WebLab-Deusto es Python [27] porque es un lenguaje de programación dinámica muy potente, tiene una gran comunidad de código abierto y permite desarrollar prototipos muy rápidamente. Como desventaja de Python cabe destacar que su rendimiento no es muy óptimo. Python es usado internamente por Google, Yahoo, NASA, Industrial & Magic y otras empresas [28], incluso Microsoft y Sun han desarrollado intérpretes de Python para sus entornos .NET y Java, llamados IronPython [29] y Jython [30], respectivamente.

4. Conclusiones

Utilizando la experiencia acumulada desde 2001 como diseñadores del WebLab-Deusto el trabajo ha remarcado en primer lugar la importancia del sw en la calidad final de un laboratorio remoto, sobre todo si este va a ser utilizado como una herramienta didáctica de la universidad, y no solo del laboratorio. En segundo lugar se han analizado diferentes tecnologías para el desarrollo de las aplicaciones cliente y servidor.

Al analizar la aplicación del cliente se proponen como básicas dos características ordenadas por su importancia: la universalidad y la potencia, entendida la primera como la posibilidad de que cualquier usuario o alumno pueda acceder al laboratorio remoto. El análisis concluye con la elección de AJAX como la tecnología más adecuada y potente a la hora de diseñar un laboratorio remoto, seguida de Adobe Flash que destaca por su potencia en el tratamiento de audio y vídeo.

El escenario de diseño del servidor no es tan exigente como el del cliente, ya que aunque su calidad condiciona el del resto del laboratorio remoto, no hay tantas restricciones en cuanto a la universalidad. En el caso de WebLab-Deusto, la elección es Python ya que permite un rápido prototipado, aunque su rendimiento no es muy óptimo.

Referencias

[1] International Meeting on Professional Remote Labs. Bilbao, 12-13 noviembre de 2006. http://weblab.deusto.es

[2] Ma, J. y Nickerson, J.V., 2006, "Hands-on, Simulated, and Remote Laboratories: A Comparative Literature Review", *ACM Computing Surveys*, Vol. 38, Nº 3, Article 3.

[3] Kolberg, S. y Fjeldly, T.A., 2004, "Web Services remote educational laboratories", Proceedings of the International Conference on Engineering Education, Gainesville, FL 1-6.

[4] Garcia-Zubia et al, 2006, "Questions and answers for designing useful WebLabs", *International Journal of Online Engineering,* VOL II, Nº 3, ISSN: 1861-2121, www.ijoe.org, Austria.

[5] Garcia-Zubia et al, 2005, "Evolving towards better architectures for remote laboratories: a practical case", *International Journal of Online Engineering,* VOL I, Nº 2, ISSN: 1861-2121, www.ijoe.org, Austria.

[6] García Zubía, J. y Sáenz Ruiz de Velasco, J.M., 2005, "Diseño de laboratorios remotos virtuales: WebLab", Actas de JENUI 2005, pp: 405-412, ISBN: 84-9732-421-8.

[7] [secJava] http://java.sun.com/javase/6/docs/technotes/guides/security/permissions.html

[8] [secNet] http://msdn2.microsoft.com/en-gb/library/930b76w0(vs.71).aspx

[9] [appletJava] http://java.sun.com/applets/

[10] [downloadJava] http://java.sun.com/javase/downloads/

[11] [adobeMacromedia] http://www.adobe.com/aboutadobe/invrelations/adobeandmacromedia.html

[12] [platformsFlash] http://www.adobe.com/products/flashplayer/productinfo/systemreqs/

[13] [flash9linux] http://blogs.adobe.com/penguin.swf/2007/01/flash_player_9_for_linux_x86.html

[14] [adobeReaderBug] http://www.securityfocus.com/archive/1/455790/30/0/

[15] [ajax] http://www.adaptivepath.com/publications/essays/archives/000385.php

[16] [operaDevices] http://www.opera.com/products/mobile/products/

[17] [flashAccessibility] http://www.adobe.com/resources/accessibility/best_practices/bp_fp.html

[18] [BOCG, 2002] Boletín Oficial de las Cortes Generales, Num. 68-13, 3 de julio de 2002, http://www.congreso.es/public_oficiales/L7/CONG/BOCG/A/A_068-13.PDF

[19] [BOE, 2003] Boletín Oficial del Estado, Num. 289, 3 de diciembre de 2003, http://www.boe.es/boe/dias/2003/12/03/pdfs/A43187-43195.pdf

[20] [flashNeededForGMail] https://mail.google.com/support/bin/answer.py?ctx=%67mail&hl=en&answer=35877

[21] [GWTOpenSource] http://googlewebtoolkit.blogspot.com/2006/12/gwt-13-release-candidate-is-100-open_12.html

[22] [GWT] http://code.google.com/webtoolkit/

[23] [OpenLaszlo] http://www.openlaszlo.org/

[24] [OLLegals] http://www.openlaszlo.org/legals

[25] [Orbit] https://orbit.dev.java.net/

[26] [Mono] http://www.mono-project.com

[27] [Python] http://www.python.org

[28] [PythonSuccess] http://www.python.org/about/success/

[29] [IronPython] www.codeplex.com/Wiki/View.aspx?ProjectName=IronPython

[30] [Jython] http://www.jython.org

SWECAI: Sistema web centrado en el alumno inteligente

Juan Julián Merelo Guervós, Carmel Hassan-Montero, Fernando Tricas*,
Juan Luis Fernández Laredo
Depto. Arquitectura y Tecnología de Computadores
ETS Ingenierías Informáticas y Telecomunicaciones
Universidad de Granada
* Departamento de Informática e Ingeniería de Sistemas
Centro Politécnico Superior
Universidad de Zaragoza

Resumen

La introducción de tecnologías de la información y la comunicación (TIC) en una clase con conexión permanente a Internet implica un cambio del concepto de la clase, que puede llevar a aumentar las fuentes de distracción del alumno, o, por el contrario, aumentar la participación por parte del alumno en la asignatura. Lo que nosotros pretendemos es conseguir lo último, y por ello, nos hemos sentido motivados para la búsqueda de un CMS (*Content Management* System, o sistema gestor de contenidos) que centralizara esfuerzos y facilitara el seguimiento tanto virtual como presencial de las clases teórico-prácticas de forma transparente para el alumno. El resultado obtenido es el portal web SWECAI (Sistema Web Centrado en el Alumno Inteligente, `http://swecai.ugr.es`), construido sobre el CMS Drupal, que incluye el uso integrado de estas herramientas para la toma de apuntes y la realización de ejercicios de autoevaluación, además de proporcionar soporte para la incorporación de un sistema de envío y corrección de prácticas y trabajos finales. En esta ponencia se presentarán las decisiones de diseño tomadas para elegir el CMS, así como los primeros resultados de la experiencia. Se expondrá también la metodología seguida para integrar las herramientas en la clase presencial, así como posibles aplicaciones dentro y fuera del aula.

1. Introducción

Uno de los problemas principales de las asignaturas de Informática (o, para el caso, de cualquier otra carrera) que cubren un tema periférico, como las optativas o las que se ofrecen como de libre configuración, consiste en despertar el interés del alumno en las mismas. Suelen ser asignaturas con alumnos procedentes de diferentes cursos, con intereses, horarios, y disponibilidad muy diferentes, y que apenas coinciden durante la propia clase. En ese estado, el concepto de *apuntes* tradicional es difícil de aplicar. Las notas de clase del profesor y los guiones de prácticas dan una visión parcial de la asignatura, y, en todo caso, contienen poca información sobre la temporización y el énfasis que se hace en un apartado determinado. Los apuntes, incluso aunque se tomen, darían una visión fragmentada, pero el hecho es que no existe en Informática una cultura de elaboración de los mismos, mediante la toma de notas en clase, su compartición a través de la fotocopiadora o usando la red social del alumno.

En ese entorno, el profesor se encuentra que los alumnos ignoran lo que se ha dicho en clase, y, en general, todo lo que no esté en los apuntes disponibles en el sitio web del profesor o en las transparencias que se pasan al alumno. Y estas explicaciones dadas en clase son en muchos casos imprescindibles para realizar una práctica, contribuyen a comprenderla, o da conocimiento de cuándo termina el plazo de entrega de la misma. Algunas veces esta ignorancia se debe a que no conocen a nadie en clase que les haga llegar esa información, y otras veces, simplemente, a que convierten una asignatura presencial en una asignatura virtual, pensando que sólo con mirarse los apuntes que están en Internet y bajarse los guiones de prácticas, enviando los trabajos también por Internet (como sucede en esta asignatura) tienen suficiente para sacar una nota aceptable en la asignatura.

Un problema adicional es que este comportamiento perpetúa el modelo de interacción unidireccional profesor alumno, basado en el algoritmo

- El profesor explica (o escribe los apuntes)
- El alumno realiza una prueba
- El profesor corrige la prueba

La Internet y los medios virtuales se convierten en una barrera, más que una forma de unir. Incluso aunque se provea de otros medios al alumno para participar en clase, como bitácoras [1] o listas de correo, a veces lo que se consigue es virtualizar aún más la interacción profesor-alumno. Lo que se ha pretendido con la introducción de diferentes canales de comunicación entre el profesor y el alumno tales como wikis, bitácoras y foros como método para tomar apuntes de clase, almacenarlos y aportar información adicional es no sólo guardar un registro de lo expuesto, sino también que se use esa conexión para que los alumnos complementen (o, porqué no, corrijan) lo que el profesor transmite, incluyendo, por ejemplo, recursos educativos adicionales, o páginas web que ilustren lo explicado en clase.

Los wikis tienen una serie de características que los hace atractivos para esta tarea:

- Tienen una sintaxis de etiquetas simplificada: es muy fácil añadir enlaces, realces o crear listas, añadiendo un sólo carácter al texto
- Es muy fácil crear nodos[1] nuevos: sólo incluir una palabra con `EsteFormato` (es decir, con mayúsculas iniciales y en el interior de la palabra) crea automáticamente un nodo nuevo; el nodo se puede editar pinchando en la palabra, que se convertirá automáticamente en enlace a ese nodo nuevo. En algunos casos se usa otro formato diferente (por ejemplo, rodear una palabra con corchetes []), pero, en todo caso, la creación de nodos nuevos es relativamente fácil.
- Guardan una historia de todos los cambios que se han hecho al documento, permitiendo hallar fácilmente quién ha trabajado en él y qué cambios ha hecho. Esto permite la colaboración espontánea, pero también el volver a una versión anterior en caso de equivocación (y en caso de *spam*, que suele ser lo más habitual).
- El resultado está inmediatamente publicado en Internet, permitiendo una interacción no sólo presencial, sino virtual.

Sin embargo, también tienen una serie de inconvenientes, el principal de los cuales es el hecho de que dos personas no pueden trabajar simultáneamente en el mismo nodo; el último que envíe sobreescribirá al anterior, con lo cual la *granularidad* del trabajo es relativamente gruesa, es decir, la cantidad de trabajo que se suele hacer antes de guardarlo en el wiki es relativamente alta. Hay otros productos que permiten una edición más fina, como el SubEthaEdit, pero no se pueden integrar dentro de una herramienta online, ni tienen su resultado publicado en Internet directamente.

El trabajo se organiza como sigue: a continuación se expone el estado del arte en la materia de uso de herramientas de participación y colaboración online en un entorno de enseñanza virtual o presencial; y experiencias anteriores en el uso de esos medios de los autores. La siguiente sección (3) explica la experiencia actual de integración de diferentes modos de comunicación, para concluir en la sección 4 con las conclusiones y la exposición de posibles líneas de trabajo futuro.

2. Material y métodos

A la hora de implementar un LCMS, caben dos opciones: partir de un CMS (un sistema de gestión de contenidos, tal como Plone , Zope , o Drupal) y adaptarlo, o partir de un LMS ((*Learning Management System*, o sistema de gestión de aprendizaje tal como Moodle o ILIAS) y, en cualquier caso adaptarlos, ya que los sistemas que sean capaces de hacer *de fábrica* ambas cosas (los denominados propiamente LCMS o *Learning and Content Management* Systems, sistemas de gestión de contenido y aproendizaje) son escasos y poco conocidos. En todo caso, entre las alternativas sólo pueden estar productos de software libre, pues habrá que adaptarlos a las necesidades particulares de las asignaturas (empezando por el idioma), y hacer tantas instalaciones y modificaciones como sean necesarias. Y entre las capacidades que se requieren del sistema, están las siguientes:

- Bitácoras, es decir, páginas web organizadas cronológicamente, para aportar información adicional a clase, y enviar los ejercicios de autoevaluación.
- Wikis, páginas web editables por parte del usuario, para tomar apuntes en clase.
- Sistema de corrección de prácticas y trabajos, con resúmenes y listados que el profesor pueda ver.
- Inclusión de apuntes dentro de la misma plataforma, con gestión por parte del profesor.

[1]Nodo es el nombre con el que nos referimos a una página en un wiki

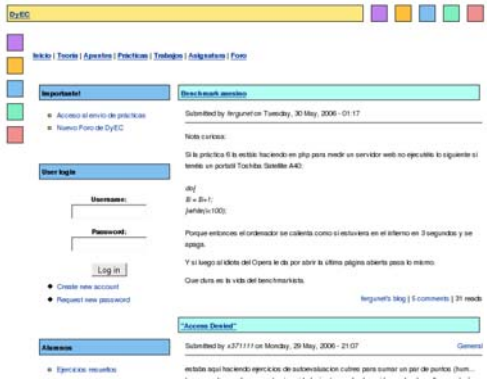


Ilustración 1: Captura de pantalla del portal basado en Drupal del que se habla en este trabajo.

De todos los sistemas examinados anteriormente, Zope, que es en realidad un servidor de aplicaciones, y Plone, un CMS basado en él, se encontraron muy completos, pero también muy difíciles de usar (al menos, para una persona que se aproxime por primera vez al lenguaje Python, que es el que se usa). Su extensión para uso como LMS puede ser una buena opción si ya se tiene experiencia en el uso de la plataforma. Por otro lado, tanto Moodle como Ilias ofrecen la posibilidad de programar *plugins* (o módulos que añaden funcionalidad al programa principal), y, de hecho, tienen módulos para lo indicado anteriormente, pero son poco flexibles en los tipos de documentos que se pueden añadir y, además, algunos de los módulos anteriores, en la práctica, no funcionan. En ambos casos, están más enfocados a la presentación de contenidos ya elaborados y a la evaluación del alumno a partir de ellos, que a la creación espontánea de nuevos contenidos, que es lo que pretendíamos nosotros en este proyecto.

Por eso, la elección final recayó en el CMS Drupal, que tiene la ventaja de estar desarrollado sobre una plataforma bien conocida (el sistema de gestión de bases de datos MySQL y el lenguaje de programación PHP), tener una base amplísima de plugins desarrollados (además de un modelo simple para desarrollar nuevos, en caso necesario), y una comunidad de usuarios considerable, a la que se puede acudir en caso necesario. El portal resultante se muestra en la ilustración 1. Sobre la instalación base de Drupal, se añadieron una serie de plugins para adaptarlo a nuestras asignaturas, tales como book (para crear libros, que contendrían los apuntes de la asignatura) y wiki, para que los usuarios puedan crear y modificar wikis.

Ilustración 2: Área de trabajo del profesor. En la parte superior están los accesos rápidos a los apuntes (wiki para tomar apuntes en clase), y a otra serie de contenidos estáticos, mientras que a la izquierda hay diferentes opciones de creación de contenido, tales como páginas o páginas de un libro. La más habitual es la creación de entradas del diario o bitácora.

Al principio de las clases, se llevó a cabo un breve seminario para explicar a los alumnos como usarlo. En principio, tanto alumnos, como administradores, como profesores pueden crear contenido de tres tipos diferentes: entradas en la bitácora o diario, nuevos nodos de apuntes (que se implementan mediante un wiki), hebras en el foro aparte de comentarios a todos los nodos anteriores. El área de trabajo del profesor se muestra en la ilustración 2 y la del alumno en la ilustración 3.

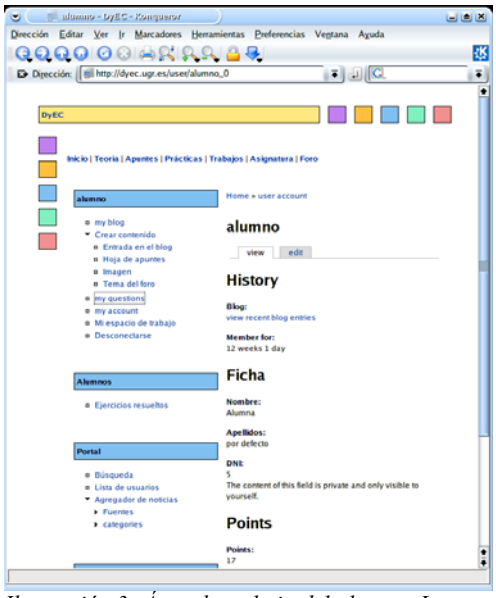


Ilustración 3: Área de trabajo del alumno. La funcionalidad a la que tiene acceso está limitada a crear apuntes, entradas en la bitácora, y hebras en el foro; el resto de los enlaces son de mantenimiento.

En general, el tipo de contenido que se puede crear por parte de cada rol es totalmente configurable. Una plataforma de CMS como Drupal está más enfocada a un flujo de trabajo entre diferentes roles y, de ahí, a la página principal del portal flexible y configurable, de forma que, por ejemplo, se podían configurar alumnos con menos privilegios (digamos *alumnos de libre configuración*, o alumnos presenciales frente a alumnos virtuales), y, en general, permite regular de forma bastante precisa qué tipo de contenido se puede crear por parte de cada rol y quién puede verlo.

Durante el uso del sistema, se presentaron algunos problemas de mantenimiento del propio servidor, que hizo que se tuviera que establecer una serie de métodos de vigilancia de la carga, y se tuvieran que instalar módulos para mejorar la rapidez. Asimismo, se atendió a las solicitudes de los usuarios, en el sentido de añadir funcionalidades como el propio foro (que al principio no estaba previsto).

El uso del sistema se premió mediante una *nota de clase*, que comprendía una sexta parte de la nota total de la asignatura. Con esta nota de clase se califica tanto la asistencia presencial, como la participación en los diferentes aspectos del portal. Por otro lado, el único contenido obligatorio del portal eran los ejercicios de autoevaluación, una serie de actividades planteadas a lo largo de los apuntes, que los alumnos pueden hacer y entregar a través de una entrada en su bitácora.

Por otro lado, en este proyecto hemos intentado que los alumnos usen el sistema desde la propia clase. Se anima a los alumnos a que se lleven los portátiles a clase, y se usa el *Aula Móvil Centrino* de la UGR[2] para que los alumnos que no dispongan de portátil puedan pedir prestado uno y usarlo durante la clase.

Las diferentes actividades en el portal se premian también con una puntuación, que permite luego establecer ránkings y también asignar la nota de clase dependiendo de esas actividades. Se asignaban 5 puntos por entrada en el wiki (toma de apuntes), 2 por historia en la bitácora, y un punto por comentario y por fichero subido. Los resultados se reflejaban en un ránking, que se mostraba y comentaba en clase periódicamente.

A continuación expondremos los resultados obtenidos en esta experiencia.

3. Resultados

Para empezar, el portal está todavía en uso (a fecha de mayo de 2007, aunque los resultados son los disponibles a la fecha de entrega del trabajo, febrero 2007), y esperamos que los alumnos lo sigan usando más allá del fin de la asignatura, si así lo desean. El URL es `http://swecai.ugr.es`. Aparte de esto, el principal resultado, como sucede en estas nuevas experiencias, es si, efectivamente, los alumnos han usado el sistema, y han obtenido algo de él. Las estadísticas de uso, para la asignatura Diseño y Evaluación de Configuraciones, una optativa de 3º de Informática con un porcentaje alto (alrededor del 50%) de alumnos de libre configuración, se muestran en la tabla 1.

[2]Un aula de portátiles donada por Toshiba, a disposión del profesorado

Nodos totales	792
Nodos publicados	789
Ejercicios de autoevaluación	34
Comentarios	621
Usuarios	75

Tabla 1: Estadísticas de uso del portal desde el momento de su inauguración, a principios de marzo de 2006, hasta el momento de la escritura de este artículo, el 31 de mayo de 2006. Las diferencias entre nodos totales y publicados se deben posiblemente a la existencia de borradores que no han sido publicados, pero se quedan en la base de datos

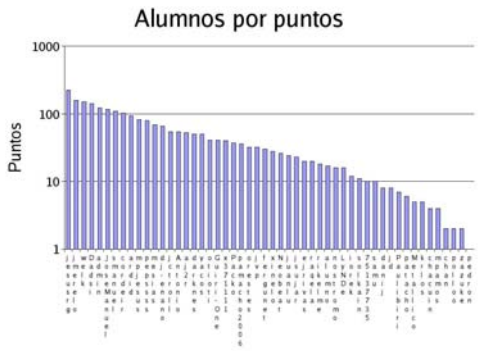


Ilustración 4: Ranking de usuarios por puntos, con escala vertical logarítmica. Curiosamente, siguen aproximadamente una ley de potencias, más concretamente una ley de Zipf.

Como se ve, la publicación de contenido ha sido bastante entusiasta, correspondiendo a cada alumno (72 de ellas, el resto de los usuarios del portal son los profesores, la administradora y alguna cuenta de prueba) unos 11 nodos (historias en bitácora o contenido de otro tipo: imágenes, hebras en el foro) y aproximadamente 9 comentarios. Evidentemente, el reparto de creación de contenido es bastante desigual. Si nos atenemos al ránking por puntos asignados por el portal, cuyo gráfico se muestra en la ilustración 4, como corresponde a una ley de potencias, la mayoría del contenido creado se concentra en unos cuantos usuarios. Mientras que la mediana está en 29 puntos, la media está en 45.6 puntos. Por esa razón no se puede decir que haya un *alumno medio* con respecto al uso del portal. Hay unos pocos que lo usan con entusiasmo (alrededor del 10%), otro grupo mayor que aporta contenido con regularidad (un 20-30%), y el resto que aporta contenido esporádicamente.

En general, el contenido que aportan está relacionado con la asignatura, pero hay una cierta tendencia a hablar de las cosas relacionadas con la vida informática: derechos de autor, juegos, y también, en algunos casos, crónicas de eventos sucedidos en la escuela.

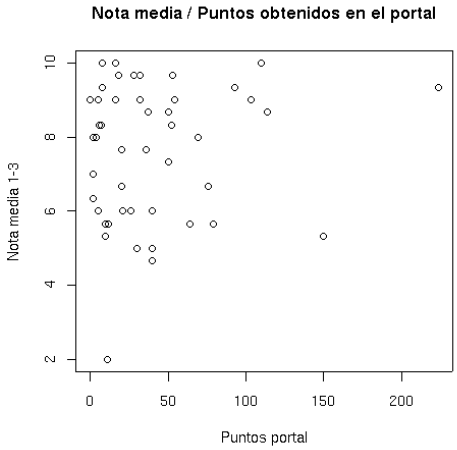
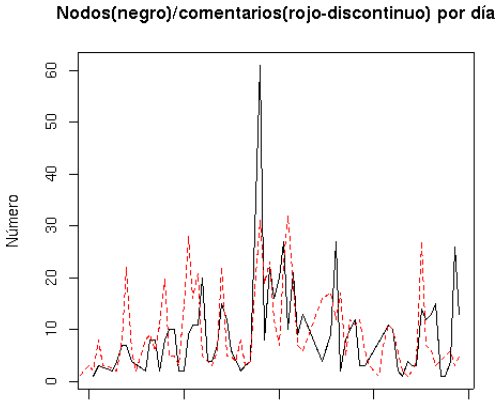


Ilustración 5: Puntuación en el portal frente a la nota media obtenida en las tres primeras prácticas por una parte de los alumnos.

Por otro lado, la participación en el portal a base de aportación de contenidos y comentarios ha sido más o menos continua a lo largo del tiempo, como se muestra en la ilustración 5. Hay una cierta correlación entre el número de nodos (historias o hebras en el foro) y el número de comentarios, como es natural, y también una cierta naturaleza cíclica, que concentra la actividad alrededor del martes, día en que tenía lugar la clase.



El impacto que ha tenido el uso del portal sobre la nota obtenida en la asignatura (si consideramos ésta, claro está, como el mejor indicador del aprendizaje de los alumnos en clase) es difícil de evaluar. Sin embargo, para una parte de los alumnos[3] , podemos trazar la puntuación obtenida en el portal frente a la nota obtenida en las prácticas, que se muestra en la ilustración 5. En general, los usuarios más entusiastas (puntuaciones mayores que 100), tienen una nota media por encima de 6.5. También en general, la mayoría de los usuarios identificados del portal (cuya participación, recordemos, es voluntaria) tienen una media por encima de 5 puntos (salvo tres usuarios); sin embargo, hay alumnos con una puntuación relativamente alta que tienen una nota media en prácticas baja; y, por el contrario, hay alumnos con un uso del portal bajo que tienen una nota media relativamente alta. En resumen, es difícil decir que la puntuación en el portal resulta un predictor de la nota en prácticas, aunque si es cierto para los casos que tienen una puntuación más alta, lo que nos obliga a buscar otros parámetros de uso que estén más relacionados con la nota, o visto de otra forma, a asignar un sistema de puntuación que esté más correlacionado con el desempeño que el alumno pueda tener en la asignatura.

Todas las clases a partir de la segunda han sido reflejadas en el wiki, que además incluye un nodo de FAQ (*Frequently Asked Questions*, preguntas frecuentemente preguntadas), enfocada sobre todo a resolver dudas relativas a la entrega del trabajo final que sirve para evaluar la asignatura. En general, las páginas del wiki, tal como observamos en otras experiencias [2], tienen una sola edición; por este motivo se dedicó una clase específicamente a crear una versión de los apuntes de cada día que fusionara lo escrito por cada uno de los tomadores de apuntes, lo limpiara, y aportara información adicional. Es difícil evaluar también el impacto que ha tenido el wiki sobre el aprendizaje de los alumnos.

Ninguna de las páginas del wiki aparece entre las páginas más visitadas, pero, de hecho, se visitan bastante, sobre todo hacia el final de la asignatura, cuando se quiere buscar una aclaración. En todo caso, el problema es que no hay una cultura que te lleve a buscar por los recursos de la asignatura, en vez de, por ejemplo, hacer una pregunta en el foro, o directamente acudir a buscar al profesor para que explique directamente qué es lo que hay que hacer. En principio, pensamos que el uso se incrementará según sea más común este tipo de entornos para enseñanza presencial

[3]Los alumnos usan un nick o alias en el portal, y hasta el momento de la calificación, no se les puede asociar a su identidad real.

4. Conclusiones

La principal conclusión de este proyecto, que todavía se está desarrollando, es que si se van a usar diferentes medios de aportación de contenidos y de evaluación del alumnado (y del profesor), es mejor tenerlos centralizados en un portal, el *portal de una asignatura*. En general, este es el enfoque de los cursos de enseñanza virtual; en enseñanza presencial o mezclada (*blended*) la conclusión es la misma. Por otro lado, los LMS enfocados a enseñanza virtual son poco flexibles (en el sentido de que no tienen un interfaz de programación abierto para incorporar nuevas funcionalidades, en caso de que sea necesario), y es más conveniente usar un CMS flexible y libre, instalando los módulos necesarios para adaptarlo a un uso como sistema de gestión de aprendizaje, y desarrollando los módulos que sean necesarios.

En general, es complicado que los alumnos se interesen en el uso de este tipo de plataformas, sobre todo teniendo en cuenta que no puntúan, o no tanto como el resto de las actividades relacionadas con la asignatura. Sin embargo, forma parte de ese tipo de habilidades *transversales* que están más relacionados con la habilidad de comunicación que con el aprendizaje de técnicas informáticas específicas. En todo caso, la iniciativa parece haber sido bien recibida por los principales clientes de la misma.

En resumen, aunque la participación ha sido desigual, para los que asisten habitualmente a clase ha sido un estímulo más para aprender; y a los que asisten esporádicamente, les ha permitido estar en contacto con los compañeros y no perder contacto con la misma. Una de las modificaciones que se van a hacer en el futuro es crear nuevos roles, entre ellos el de *antiguo alumno*, que pueda, por ejemplo, actualizar la bitácora, o comentar en foros y bitácoras, pero sin tener acceso a la modificación de los apuntes (aunque esto tampoco sería una mala idea).

Por otro lado, hay cosas que mejorar en el sistema, que se enfocarán como trabajo en el futuro. Una de ellas es ajustar bien el sistema de puntuaciones, de forma que se premie más, por ejemplo, la participación mediante ejercicios de autoevaluación, o la corrección de apuntes, más que la toma de apuntes. Quizás implantar algún sistema de *karma*, es decir, de puntuación *paritaria*, para que los alumnos puedan puntuar las historias y comentarios unos de otros.

Otra mejora al sistema será integrar el envío y recepción de prácticas y trabajos a través del mismo sistema; ahora mismo se sigue usando un sistema antiguo, alojado en otra base de datos y ordenador diferente.

Por otro lado, en cursos sucesivos se van a incorporar nuevas asignaturas al sistema, y ello planteará una serie de retos; principalmente para el flujo de trabajo, pero también para la medición y optimización de la carga del sistema. Finalmente, el sistema necesitará una serie de evaluaciones pedagógicas, de usabilidad, y de *satisfacción del cliente*. Trataremos de recolectar estadísticas sobre uso del mismo, y aplicar algoritmos de *data mining* para tratar de sacarle el máximo partido, como ya han hecho en la Universidad de Sevilla [3]. Pero el verdadero reto es posiblemente ver cómo se puede evaluar la eficacia en el aprendizaje de estas técnicas, cualitativamente, y, si es posible, cuantitativamente.

5. AGRADECIMIENTOS

Este proyecto está financiado en parte por el Vicerrectorado de Planificación, Calidad e Innovación Docente de la UGR, con el proyecto 06-03-17 y siguientes.

6. References

[1] Rune Baggetun. Prácticas emergentes en la web y nuevas oportunidades educativas. *TELOS*, (67), Abril-Junio 2006. Disponible en `http://www.campusred.net/telos/articulocuaderno.asp?idarticulo=5&rev=67%`. V0.1-4.

[2] Juan Julián Merelo Guervós and Fernando Tricas García. Integración de wikis en una clase de informática: un paso más en la creación de comunidad. In Manuel Ortega Cantero, editor, *Actas I Simposio Nacional de Tecnologías de la Información y de las Comunicaciones en la Educación, SINTICE05*, Nuevos retos científicos y tecnológicos en Ingeniería Informática, pages 19–26. Thomson, Septiembre 2005.

[3] CR Morales, SV Soto, and CH Martínez. Estado actual de la aplicación de la minería de datos a los sistemas de enseñanza basada en web. In *Actas de TAMIDA 2006, Taller sobre Minería de Datos*. Thomson-Paraninfo, 2005.

Evaluación del alumnado

Análisis del rendimiento académico en los estudios de informática de la Universidad Politécnica de Valencia aplicando técnicas de minería de datos

R. Alcover[1], J. Benlloch[2], P. Blesa[3], M. A. Calduch[1], M. Celma[3], C. Ferri[3], J. Hernández-Orallo[3], L. Iniesta[5], J. Más[4], M. J. Ramírez-Quintana[3], A. Robles[2], J. M. Valiente[2], M. J. Vicent[3], L. R. Zúnica[1]

[1]Dpto. de Estadística e I.O. Aplicadas y Calidad, [2]Dpto. de Informática de Sistemas y Computadores, [3]Dpto. de Sistemas Informáticos y Computación, [4]Dpto. de Física Aplicada, [5]Becaria PACE

Universidad Politécnica de Valencia

Camino de Vera s/n, 46022 Valencia

jmas@fis.upv.es

Resumen

En este trabajo presentamos un análisis del rendimiento académico de los alumnos de nuevo ingreso en la titulación de Ingeniería Técnica en Informática de Sistemas de la Universidad Politécnica de Valencia (UPV) a lo largo de tres cursos, aunque también se ha trabajado con las titulaciones de Ingeniería Técnica en Informática de Gestión y de Ingeniería Informática.

Este análisis relaciona el rendimiento con las características socioeconómicas y académicas de los alumnos, que se obtienen en el momento de su matrícula, y que se recogen en la base de datos de la universidad. Hemos definido un indicador del rendimiento para cada alumno, teniendo en cuenta las calificaciones obtenidas y las convocatorias utilizadas.

Para el estudio utilizamos técnicas de minería de datos, que pretenden determinar qué nivel de condicionamiento existe entre dicho rendimiento y características como el nivel de conocimientos de entrada del alumno, su contexto geográfico y sociocultural, etc… Esto proporciona una herramienta importante para la acción tutorial, que puede apoyarse en las predicciones de los modelos que se obtienen para encauzar sus recomendaciones y encuadrar las expectativas y el esfuerzo necesario para cada alumno, lógicamente dentro de la cautela habitual a la hora de tratar modelos inferidos a partir de datos.

1. Motivación

Desde hace algún tiempo se le está dedicando una creciente atención al rendimiento académico universitario. Esta mayor atención viene determinada por factores tanto de índole económica como política y social; la permanencia de los estudiantes en la universidad durante prolongados períodos de tiempo es un aspecto controvertido y causante, en parte, de la demanda cada vez mayor de las titulaciones de primer ciclo frente a las de segundo ciclo [6].

Por otra parte, el proceso de convergencia europea de la educación superior está obligando a las universidades al diseño de estrategias de adaptación debido a la profunda reforma que este proyecto implica. Los procesos de evaluación y acreditación de títulos puestos en marcha en los últimos años para llevar a término esta adaptación se sustentan en la construcción de indicadores, de muy variada índole, que permiten descubrir las fortalezas y debilidades de las enseñanzas universitarias actuales. Como se comenta en [9], "reflexionar sobre todos los elementos que la evaluación del rendimiento del alumnado proporciona se convierte en un mecanismo claro para la mejora de la calidad del proceso educativo". En este sentido, es interesante analizar el rendimiento académico de los estudiantes para tomar medidas oportunas tanto de forma individual como global. Asimismo, según [5], en la propuesta de los futuros títulos de Grado y Máster se deberá incluir una estimación de los valores de ciertos indicadores de resultados, como las tasas de graduación, abandono y eficiencia. En este aspecto, el Ministerio de Educación y Ciencia ya recoge una lista de indicadores [7] para estos estudios. Más aún, la viabilidad académica de los títulos (que estos puedan ser completados en el tiempo previsto por la mayoría de los estudiantes

a tiempo completo) será un criterio clave para la aprobación de los mismos, así como para la evaluación de las universidades y sus centros.

Algunos factores podrían, en gran medida, explicar el éxito o fracaso de un estudiante, como sus características socioeconómicas, edad, estudios previos, entorno al inicio de sus estudios, actividad, o no, laboral durante los estudios, características organizativas y docentes de los centros, planes de estudios, métodos evaluativos, etc... Conocidos estos factores, tanto la universidad como los estamentos responsables de los estudios preuniversitarios, podrían estudiar acciones que mejoraran el rendimiento de colectivos específicos, como ya ocurre en el llamado Programa de Acción Tutorial Universitario (PATU). Este programa de tutorías, adoptado institucionalmente por nuestra universidad desde hace más de cinco cursos, pretende favorecer tanto la integración del alumno como su formación integral. Es en esta línea en la que se enmarca nuestro trabajo, para el cual hemos utilizado la investigación directa en las bases de datos que recogen información tanto sobre las características e historial del alumnado, como de la universidad, mediante técnicas de minería de datos [4]. A pesar de que existen innumerables trabajos descriptivos sobre rendimiento académico en universidades, no tenemos conocimiento de que las técnicas de minería de datos se hayan aplicado de forma exhaustiva a este tipo de estudios. Sí que aparece algún trabajo en el que se utiliza la técnica de *"clustering"* [1] o de regresión lineal mediante procedimientos tradicionales [3].

2. Metodología

La metodología que se ha seguido para la obtención del rendimiento se puede resumir en las siguientes etapas:

1. Establecer el/los objetivo/s del estudio.
2. Definir la población y la muestra de estudiantes implicada en el estudio.
3. Obtención de la vista minable, a partir de la información contenida en la base de datos de la universidad.
4. Elección del tipo de análisis de datos requerido.
5. Generación y validación de los modelos.
6. Interpretación de los resultados.

Seguidamente se presentan las tareas llevadas a cabo en cada una de estas etapas.

2.1. Objetivo del estudio

Nuestro estudio, a nivel global, pretende aplicar técnicas de minería de datos para analizar la influencia de los parámetros (socioeconómicos, características personales, nota de entrada…) más relevantes sobre el rendimiento académico de un alumno de primer curso en las titulaciones de informática de la UPV, de forma que nos permita predecir este rendimiento disponiendo únicamente de la información aportada por el alumno en el momento de su matrícula. Estas titulaciones corresponden a Ingeniería Informática (II) impartida por la Facultad de Informática (FI), Ingeniería Técnica en Informática de Gestión (ITIG) e Ingeniería Técnica en Informática de Sistemas (ITIS), estas últimas impartidas por la Escuela Técnica Superior de Informática Aplicada (ETSIAp). En este trabajo se presentan únicamente resultados correspondientes a ITIS.

La adopción de políticas encaminadas a corregir situaciones de fracaso académico a partir de las conclusiones de este trabajo no es un objetivo del estudio. Estas políticas deben ser acometidas bien por los centros, bien por los estamentos responsables.

2.2. Población

La población objeto de nuestro estudio está constituida por todos los alumnos de nuevo ingreso en cualquiera de las tres titulaciones de informática de la UPV antes mencionadas. Con el fin de trabajar con una muestra representativa de la población, se ha considerado a los alumnos de nuevo ingreso matriculados en primero de alguno de los títulos de informática durante los cursos 01-02, 02-03 y 03-04, esto es, desde el último cambio del plan de estudios. Así, el estudio se ha realizado sobre 569 alumnos de II, 646 alumnos de ITIG y 572 alumnos de ITIS.

2.3. Obtención de la vista minable

Como ya se ha comentado, la base de datos de una universidad incorpora, curso tras curso, una gran cantidad de información relativa al alumnado, procedencia, matrículas, calificaciones, planes de estudios,… Por lo tanto, esta base de datos

contiene información sobre diferentes factores, *a priori*, potencialmente relevantes para el rendimiento de un alumno. Desde los dos centros implicados en nuestro estudio se dispone de una vista parcial de la base de datos de la UPV que contiene la información de la FI y de la ETSIAp.

Un aspecto importante es la despersonalización de los datos puesto que, de acuerdo a la Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal, se debe proceder tratando los datos personales de modo que la información que se obtenga no pueda asociarse a persona identificada o identificable. El proceso de despersonalización ha consistido en eliminar o sustituir cualquier información que pueda identificar directa o indirectamente al alumno (como el DNI, nombre y apellidos, domicilio, etc…).

Con el fin de crear un almacén de datos y un entorno que facilitara la obtención de datos para realizar el estudio, se decidió integrar los mismos en *Oracle*. De esta forma se ha podido utilizar como herramienta OLAP el *Oracle Discoverer*. Con ella se han extraído las vistas minables.

Una vista minable puede definirse como una colección de individuos sobre los cuales queremos realizar un determinado estudio, con todas sus características (atributos), que tiene como finalidad poder aplicar el proceso de la minería de datos sobre ella para poder extraer conocimiento útil. Así, hemos creado una vista minable por titulación. Cada una de ellas contiene las notas y datos personales de los alumnos de la muestra seleccionada. Para generarla, se ha utilizado el generador de informes del *Discoverer*, seleccionando, de entre todos los atributos disponibles, aquellos que se consideraron, a priori, que podrían tener mayor influencia en el rendimiento académico, filtrando el resto. Dichos atributos son:

- *Ocupacio P*: Ocupación del padre.
- *Ocupacio M*: Ocupación de la madre.
- *Ocupacio A*: Ocupación del alumno.
- *Ing Nota*: Nota con la que el alumno aprueba estudios de acceso.
- *Ing Est*: Estudios con los que accede a la titulación.

Seguidamente, se procedió a la agrupación de valores de algunos atributos por su elevado número de alternativas, con el fin de reducirlas y hacer más fácilmente interpretables los resultados obtenidos. Tales atributos son:

- *D_Altr Estud*: Otros estudios universitarios del alumno al ingresar en la titulación.
- *D_Estudis P*: estudios del padre.
- *D_Estudis M*: estudios de la madre.
- *Dpaises*: Derivado del país de nacimiento del alumno, agrupando por zonas geográficas.
- *Residencia Alumno*: Derivado de la provincia y el código postal donde reside el alumno durante el curso.
- *Residencia Familia Alumno*: Derivado de la provincia y el código postal donde reside la familia del alumno durante el curso.
- *Edad Ingreso*: Atributo derivado calculado como la diferencia entre el año de ingreso del alumno y año de nacimiento.

Finalmente, se especificó el tipo de cada atributo como nominal (o categórico) o numérico, siendo todos nominales excepto la nota de acceso a los estudios y la edad del alumno.

Además de todos estos datos personales del alumno, para completar la vista minable se incorporó una columna con el *Rendimiento* de cada alumno, *R*, calculado mediante la siguiente fórmula:

$$R = \frac{\sum_j 0,8^{c-1} \cdot Calif_j \cdot C_j \cdot 10}{\sum_j C_j}$$

donde *c* es la convocatoria en que el alumno supera la asignatura (1 en el caso de ordinaria y 2 en el caso de extraordinaria). De esta forma penalizamos un 20% el rendimiento de los aprobados en segunda convocatoria; no caben más opciones en este primer curso. $Calif_j$ es la calificación numérica que obtuvo el alumno en la asignatura *j* cuando aprobó dicha asignatura, tomando un 0 si el alumno no se presentó. En el caso de que el alumno suspendiera dicha asignatura, se toma la nota de la última convocatoria presentada. C_j es el número de créditos con que figura la asignatura *j* en el plan de estudios. Obsérvese que el rendimiento de un alumno así definido es un atributo que toma valores entre 0 y 100 [10].

2.4. Elección del tipo de análisis de datos requerido

Para lograr los objetivos marcados en nuestro trabajo (predecir el rendimiento de un alumno en su año de ingreso usando únicamente datos anteriores a su entrada en la universidad), los modelos de minería de datos que hemos elaborado son de tipo predictivo.

Hay que tener en cuenta que en minería de datos las técnicas estadísticas tradicionales pueden completarse con técnicas de inteligencia artificial. En este sentido, métodos adaptativos como los algoritmos genéticos y las redes neuronales permiten realizar predicciones muy acertadas, sobre todo en casos de gran complejidad y con relaciones internas no lineales, pero tienen el problema de ser menos legibles, por lo que no las hemos utilizado en nuestro trabajo.

De entre las técnicas de minería de datos existentes, hemos utilizado dos de ellas para generar los modelos predictivos del rendimiento: los árboles de decisión y la regresión multivariante.

- Los árboles de decisión son una serie de decisiones o condiciones organizadas de forma jerárquica, a modo de árbol. Son muy útiles para encontrar estructuras en espacios de alta dimensionalidad y en problemas que mezclan datos categóricos y numéricos. Básicamente, un árbol de decisión es un árbol donde cada nodo representa una condición o test sobre algún atributo y cada rama que parte de ese nodo corresponde a un posible valor para ese atributo. Finalmente, las hojas representan el valor de la variable predicha. Esta técnica se usa en tareas de clasificación, agrupamiento y regresión. Los árboles de decisión que se usan para predecir variables categóricas se llaman árboles de clasificación, mientras que los árboles de decisión que se utilizan para predecir variables continuas se llaman árboles de regresión.
- La regresión multivariante es un método estadístico clásico que permite establecer una relación matemática entre un conjunto de variables independientes $X_1, X_2, ..., X_k$ y una variable dependiente Y. Se utiliza fundamentalmente en estudios en los que no se puede controlar por diseño los valores de las variables independientes. Los objetivos de un modelo de regresión pueden ser dos: obtener una ecuación que nos permita "predecir" el valor de Y una vez conocidos los valores de las variables independientes, y cuantificar la relación entre las variables independientes y la dependiente con el fin de conocer o explicar mejor la relación. Se trata en este caso de modelos explicativos.

2.5. Generación y validación de los modelos

Para la generación de los modelos se ha utilizado la herramienta *SPSS Clementine v.9.0* [8]. En concreto, de los árboles de decisión que incorpora el Clementine, hemos utilizado para regresión el *árbol C&R*, que es un tipo de algoritmo de aprendizaje de árboles que se basa en el algoritmo CART de Leo Breiman et al. [2]. Este árbol realiza particiones binarias con el objetivo que la media de cada rama sea diferente y, por tanto, discrimine con la suficiente precisión en un número de particiones razonable como para poder asignar a cada hoja un valor cercano a la media de los elementos que caen en ella. Asimismo, también hemos aplicado el método *regression* del Clementine, que implementa una regresión lineal.

Una vez generados los modelos es necesario evaluarlos para determinar su calidad y utilidad. Dependiendo de la tarea de minería de datos existen diversos criterios que pueden usarse para evaluar los modelos, como, por ejemplo, la precisión predictiva (porcentaje de aciertos) generalmente utilizada en el contexto de la clasificación. Si la tarea es de regresión, tal y como la que nos ocupa, una forma de evaluar un modelo es mediante la raíz cuadrada positiva del error cuadrático medio (RECM), calculada como

$$RECM = \sqrt{ECM} = \sqrt{\frac{\sum_{i=1}^{n}\left(y_i - R_i\right)^2}{n}}$$

es decir, la diferencia entre el valor que para el rendimiento predice el modelo (y_i) y el valor real (R_i) que podemos calcular con la fórmula establecida anteriormente, siendo n el tamaño del conjunto de datos. Así, el parámetro obtenido viene expresado en las mismas unidades que los datos originales.

Si la cantidad de datos lo permite, la forma de entrenar y validar un modelo consiste en partir aleatoriamente los datos en dos subconjuntos disjuntos: el de los datos de entrenamiento

(*training set*), con el que se genera el modelo, y el de prueba o test (*test set*), con el que se evalúa el modelo. En ese caso, el RECM se calcula únicamente para los datos del conjunto de test.

En nuestro estudio, se ha realizado la siguiente partición de los datos: un 77% de los mismos se ha utilizado para el entrenamiento del modelo, y el 23% restante para validarlo.

A continuación mostramos, a modo de ejemplo, los modelos obtenidos para la vista minable de ITIS, así como sus errores.

La Figura 1 muestra el árbol de decisión generado por la herramienta.

```
D_Altr Estud in [ 1 ] => 58,0 (16)
D_Altr Estud in [ 2 ]  (424)
    Ing Nota <= 7,390  (339)
        Edad Ingreso <= 18,500 (160)
            Ing Nota <= 6,420 (82)
                D_Estudis P in [ 2 ] => 20,482 (26)
                D_Estudis P in [ 1 3 ] => 28,907 (56)
            Ing Nota > 6,420 (78)
                Ing Est in [ 2 ] => 25,854 (24)
                Ing Est in [ 10 ] => 40,863 (54)
        Edad Ingreso > 18,500 => 23,939 (179)
    Ing Nota > 7,390 (85)
        Ocupacio A in [ 1 ] => 28,229 (18)
        Ocupacio A in [ 2 3 ] (67)
            Edad Ingreso <= 19,500 => 50,43 (20)
            Edad Ingreso > 19,500 (47)
                D_Estudis_ M in [1 2] => 33,7 (36)
                D_Estudis_ M in [3] => 52,94 (11)
```

Figura 1. Árbol C&R para la titulación de ITIS

La notación que se sigue es: cada línea corresponde a un nodo del árbol, y contiene el nombre del atributo usado en ese nodo así como su/s valor/es entre corchetes, si es un atributo nominal, o bien una expresión de la forma ≤ V ó >V, si es un atributo numérico, siendo V un valor comprendido entre los valores mínimo y máximo de ese atributo en los ejemplos en ese nodo; además, si se trata de una hoja, se añade el símbolo "=>" tras el que se indica el valor predicho (el rendimiento, en nuestro caso); asimismo, el número que aparece entre paréntesis indica el número de instancias del conjunto de entrenamiento en ese nodo del árbol. Por ejemplo, en la Figura 1, el primer atributo seleccionado es *D_Altr Estud.* El primer hijo, que corresponde a aquellas instancias que tienen un valor de 1 (estudios universitarios) para este atributo (en concreto 16), es una hoja y el valor medio predicho para el rendimiento en este caso es 58. El otro nodo hijo es el que corresponde a las 424 instancias cuyo atributo *D_Altr Estud* tiene un valor de 2 (otros estudios). Este último nodo continúa expandiéndose ya que no es una hoja. A partir de los errores calculados por el Clementine para este modelo con respecto al conjunto de test, hemos obtenido la raíz cuadrada del error cuadrático medio (que no está incluido en la herramienta) que resultó ser 17,95.

El modelo de regresión lineal multivariante generado para los alumnos de la titulación de ITIS se muestra en la Figura 2. Dicho modelo debe interpretarse de manera que determinados valores de ciertos atributos (que se indica a continuación del nombre del atributo para los atributos categóricos, y separado de éste por un guión) hacen que el rendimiento de un alumno varíe en un determinado valor, dado por el coeficiente de cada atributo del modelo, y cuyo signo indica si el rendimiento aumenta debido a esa característica, o disminuye. Además, también aparece un término independiente de cualquier atributo.

```
Rendimiento =
    D_Altr Estud_1 (21)· 35,55 +
    D_Estudis M_3 (140)· 4,063 +
    Ing Est_10 (248)· 9,673 +
    Ing Est_4 (3)· -36,54 +
    Ing Nota (572)· 7,572 +
    Ocupacio M_8 (73)· 6,573 +
    -27,52
```

Figura 2. Modelo de regresión para la titulación ITIS

En cada atributo se muestra, entre paréntesis, el número de alumnos afectados por dicho factor, en este caso, sobre el total de la muestra estudiada. El orden en el que aparecen los atributos en el modelo no presupone mayor o menor influencia en el rendimiento. El RECM asociado al modelo es 17,45.

Finalmente, para verificar la robustez de los resultados, los modelos se generaron dos veces, con diferentes particiones de los datos en conjuntos de entrenamiento y test realizadas aleatoriamente, y se comprobó que los modelos inducidos eran parecidos. También se calculó el rendimiento medio del total de alumnos estudiados, resultando un valor de 30,7, con una desviación típica de 19,6.

2.6. Interpretación de los resultados

Como puede observarse, los dos modelos mostrados en la sección anterior tienen un error similar. Sin embargo, no tienen en cuenta los mismos atributos, algo que no parece descartable por el hecho de haber utilizado técnicas diferentes. Incluso, dentro de un mismo modelo (por ejemplo, el árbol de decisión), no todos los atributos tienen la misma importancia, existiendo algunos que ni siquiera se tienen en cuenta. Para determinar la relevancia de cada atributo con respecto al rendimiento hemos analizado por separado cada uno de los modelos.

Análisis del Árbol C&R. Ya que un atributo puede utilizarse a diferentes niveles del árbol (véase *Ing Nota* en la Figura 1) y además repetidamente, hemos calculado para cada atributo el número de ejemplos para los cuales dicho atributo se utiliza (si el atributo se utiliza varias veces para el mismo ejemplo se cuenta tantas veces como se use). El número de ejemplos viene anotado en el Clementine entre paréntesis (tal y como ya hemos comentado) y hace referencia al conjunto de entrenamiento. Con esto tenemos que un atributo es relevante si se utiliza para un número importante de casos. Este número absoluto, que llamamos I (por importancia), lo dividimos entre el número total de ejemplos utilizados para el entrenamiento, obteniendo un valor denominado IR (importancia relativa). Finalmente, sumamos todas las IR de cada atributo y normalizamos, obteniendo un valor IR2 tal que todos los IR2 para todos los atributos sumen 1 y nos muestre un valor de importancia que permita comparar más fácilmente entre diferentes árboles. La Tabla 1 incluye esta información para el árbol de la Figura 1.

Atributo	I	IR	IR2
D_Altr Estud	440	1	0,343
Ing Nota	584	1,327	0,455
Edad Ingreso	406	0,922	0,316
D_Estudis P	82	0,186	0,063
D_Estudis M	47	0,106	0,036
Ocupacio A	85	0,193	0,066
Ing Est	78	0,177	0,060
TOTAL	1282	2,913	1

Tabla 1. Análisis de los atributos en el árbol de decisión para ITIS

Hemos considerado que un atributo es relevante si IR2 > 0,05. Esto quiere decir que ese atributo se utiliza, en promedio, al menos en un 5% de las decisiones. Como hay varias decisiones por caso (todas las condiciones hasta un nodo hoja), un valor del 5% indica que el porcentaje de ejemplos afectados por este atributo será, por lo general, sensiblemente mayor que un 5%. De acuerdo a este criterio, los atributos relevantes son: *D_Altr Estud*, *Ing Nota* y *Edad Ingreso*, y en mucha menor medida, *D_Estudis P*, *Ocupacio A* e *Ing Est*. Ya sólo falta analizar si estos atributos contribuyen positivamente o negativamente, viendo cómo afecta al rendimiento medio de las dos hojas hijas según la partición del atributo. En el caso de que un atributo aparezca varias veces, se debe ver si la manera de afectar es consistente o algunas veces afecta positivamente o negativamente. Nótese que el hecho de decir que un atributo (o los valores del mismo) afecta positiva o negativamente es según las ramas del árbol en las que aparece. Podría ocurrir que un atributo afectara positivamente (o negativamente) en general (para casi todos los ejemplos), pero en un subconjunto de ellos representados por una rama del árbol sucediera al revés. El análisis ha dado los siguientes resultados:

- Los mejores rendimientos se obtienen para el valor 1 del atributo *D_Altr Estud,* es decir, los alumnos que ya poseen estudios universitarios, aunque esta condición sólo la cumple un porcentaje relativamente pequeño de alumnos.
- El atributo *Ing Nota* (nota de ingreso) afecta positivamente.
- El atributo *Edad Ingreso* afecta negativamente (cuanto mayor es, peor rendimiento).
- El atributo *D_Estudis P* afecta positivamente para los valores 1 y 3 (padre sin estudios o estudios superiores), y negativamente para el valor 2 (padre con estudios equivalentes a bachillerato).
- El atributo *Ocupacio A* afecta positivamente para los valores 2 y 3 (alumnos con una ocupación inferior a 15 horas o que no realiza trabajo renumerado), y negativamente para el valor 1 (alumnos con una ocupación mayor o igual a 15 horas semanales).
- El atributo *Ing Est* afecta positivamente para el valor 10 (alumnos que acceden desde

bachillerato LOGSE con PAU), y negativamente para el valor 2 (alumnos que acceden con prueba de acceso pero no provienen de bachillerato LOGSE).

Debemos tener en cuenta que los últimos tres resultados no son tan generales y afectan a un número relativamente pequeño de alumnos (según los valores de IR2 mostrados en la Tabla 1, la relevancia de estos atributos está en el límite considerado).

Análisis de la regresión lineal. En este modelo aparece en primer lugar el atributo *D_Altr Estud* con el valor 1 (*D_Altr Est_1*), indicativo de que poseen ya estudios universitarios. Este factor hace que el rendimiento medio de estos alumnos se incremente en 35,6 puntos, el coeficiente positivo más grande, aunque el número de alumnos que cumple esta condición (21) es reducido.

Los 248 alumnos con el atributo *Ing Est_10* (alumnos procedentes de LOGSE con PAU), con un coeficiente de 9,7 positivo tienen rendimiento medio superior a la media. En cambio, aquellos individuos que tienen el atributo *Ing Est_4* (titulados universitarios), aparecen con un coeficiente muy negativo, (-36,5), aunque sólo 3 individuos, con edades de ingreso muy superiores a la media (entre 35 y 55 años) aparecen en esta situación. Probablemente, cargas familiares y de trabajo podrían explicar su rendimiento muy por debajo de la media.

El valor 3 del atributo *D_Estudis M* (alumnos cuyas madres poseen estudios medios o superiores), afecta positivamente a un número importante de individuos (140) aunque con un coeficiente pequeño (4,1). Este atributo está relacionado con el valor 8 del atributo *Ocupacio M* (alumnos cuyas madres no tienen un trabajo remunerado), con un coeficiente de 6,6 positivo. Interpretamos estos dos atributos como que aquellos alumnos cuyas madres tienen estudios y además tienen disponibilidad para atender a sus hijos, influyen positivamente en el rendimiento de estos. Es obvio que esta positiva influencia no tiene porqué haberse producido durante el primer año de estudios universitarios, sino durante toda la enseñanza previa.

Finalmente aparece el atributo numérico *Ing Nota* con un coeficiente de 7,6. En este caso, al ser este atributo numérico, el valor del coeficiente debe ser multiplicado por el valor del atributo, que por ser la nota de acceso debe ser siempre igual o superior a 5.

3. Conclusiones

Como primera conclusión, podemos afirmar que las técnicas de minería de datos proporcionan una herramienta que permite determinar qué características de los alumnos de nuevo ingreso son más relevantes de cara a estimar su rendimiento académico el primer año. La desviación típica obtenida con el simple cálculo de la media aritmética del rendimiento es algo mayor que en los modelos (19,6 vs 17,95 y 17,45), pero la posibilidad de establecer factores determinantes del rendimiento es, a nuestro entender, la mayor ventaja de esta técnica. En el caso estudiado, factores como los estudios previos del alumno y la nota de ingreso en la titulación aparecen de manera repetida como claramente correlacionados con el rendimiento académico el primer año. También aparecen factores que podrían influir en el rendimiento, como las ocupaciones y estudios de los padres, o la edad de ingreso del alumno, aunque estos dependen de la técnica utilizada. En cambio, el país de procedencia o el lugar de residencia (del alumno o de su familia) no aparece en ningún caso.

Si bien pretendemos que los resultados específicos de este estudio sean útiles de inmediato, este trabajo se enmarca dentro de un proyecto más general que ha realizado estudios en los últimos años acerca de diferentes indicadores (abandono, duración de estudios, ...) sobre diferentes titulaciones (II, ITIG, ITIS) y diferentes cohortes. Esta es la primera vez, sin embargo, en la que toda la información se integra adecuadamente en un almacén de datos, se aplican las técnicas de minería de datos de una manera sistemática y se realiza un análisis de los modelos extraídos por equipos multidisciplinares. Estos equipos conjugan los conocimientos técnicos sobre análisis de datos con una gran experiencia en la docencia universitaria, incluyendo la participación y el apoyo de las direcciones de los centros en este análisis, así como en el despliegue y aplicación del conocimiento y de los modelos extraídos. Pensamos que este compromiso y esta continuidad es fundamental de cara a amortizar el esfuerzo realizado en las primeras etapas del

proceso: establecimiento de objetivos, entendimiento de los datos, así como de la siempre compleja limpieza e integración de datos.

En el futuro inmediato planteamos extender el análisis a otros cursos, posiblemente utilizando otros indicadores (abandono, duración de estudios), desglosar el análisis por asignaturas o por temáticas, así como ampliar las herramientas empleadas (por ejemplo *SPSS o R-project, Weka*). También estamos actualmente trabajando en un proyecto para contrastar resultados de rendimiento académico en estudios de informática en colaboración con otras universidades españolas.

No obstante, creemos que un programa estable de análisis de datos que cuente con el apoyo institucional de los centros involucrados y de la propia universidad sería fundamental para poder diseñar estrategias de mejora más adecuadas a la realidad de cada centro.

Agradecimientos

A la Universidad Politécnica de Valencia por la financiación recibida a través del programa PACE. A las direcciones de la FI y la ETSIAp, por facilitarnos los datos.

A Lorenzo Morales, subdirector del Área de Sistemas de Información y Comunicaciones, que amablemente nos resuelve todas nuestras dudas acerca de la base de datos de la universidad, y a Sara Collado, jefa de administración de la ETSIAp, que nos da luz sobre todas nuestras dudas administrativas.

Referencias

[1] Bará, J., y otros. Informe Transversal del rendimiento académico de las ingenierías técnicas. Consejo de Universidades, 2001.

[2] Breiman, L., Friedman, J., Losen, R., Stone, *Classification and Regression Trees*. Wadsworth and Brooks, 1984 (new edition 1993).

[3] Fontes, R. y otros. *Valor predictivo de algunos criterios de selección para el ingreso a la carrera de medicina*, Revista Cubana de Educación Médica Superior, vol.14, no.1, p.17-25. 2000.

[4] Hernández, J., Ramírez, M.J., Ferri, C. *Introducción a la Minería de Datos* Prentice Hall, 2004.

[5] http://www.mec.es ⇒ Universidades ⇒ Consejo de Coordinación Universitaria⇒ Directrices para la elaboración de títulos universitarios (Enero 2007).

[6] http://www.mec.es ⇒ Universidades ⇒ Estadísticas ⇒ Datos y Cifras del Sistema Universitario Español (Enero 2007).

[7] http://www.mec.es ⇒ Universidades ⇒ Estadísticas ⇒ Indicadores ⇒ Catálogo de Indicadores (Enero 2007).

[8] http://www.spss.com/clementine (Enero2007)

[9] Muñoz, S. *Indicadores de rendimiento académico del alumnado de la Universidad de La Laguna*, Jornadas sobre Políticas de Calidad en la Universidad de La Laguna, 2005.

[10] Zúnica, L., Blesa, P., Alcover, R., Más, J., Valiente., J. *Estudio del rendimiento académico de una asignatura con relación a asignaturas de cursos anteriores*, Libro de resúmenes JENUI 2003, p. 137-142, Cádiz, 2003.

Evaluación continuada sin morir en el intento

David López, Alex Pajuelo, José R. Herrero, Alejandro Duran
Dept. Arquitectura de Computadores, Universidad Politécnica de Cataluña.
UPC – Campus Nord, C/ Jordi Girona 1-3, Módulo C6. 08034 Barcelona.
{david, mpajuelo, josepr, aduran}@ac.upc.edu

Resumen

Entre los objetivos de la evaluación continuada podemos destacar: ayudar a que el alumno organice su estudio a lo largo del curso, proporcionarle *feedback* y ofrecer la posibilidad que no se juegue toda su nota en un único acto evaluador, sino en varios actos. A la hora de implementar este tipo de evaluación hay diversas metodologías, pero muchas reciben la queja de un elevado coste, tanto para el estudiante como para el profesor. En este artículo se presenta un modelo de evaluación que nos permite alcanzar los objetivos de la evaluación continuada, con un coste para el alumno y el profesor similar al de la evaluación tradicional. Los resultados muestran que los objetivos planteados han sido alcanzados.

1. Introducción

La asignatura Sistemas Operativos (SO) es una asignatura obligatoria dentro de los estudios de Ingeniería Técnica en Informática de Sistemas e Ingeniería Superior en Informática que imparte la Facultad de Informática de Barcelona (FIB), de la Universidad Politécnica de Cataluña. Se ubica en el cuarto cuatrimestre.

Es la primera asignatura relacionada con sistemas operativos que cursan nuestros alumnos, y es preparatoria para las asignaturas posteriores en el plan de estudios (*Proyecto de Sistemas Operativos* y *Administración de Sistemas Operativos*). Es un curso básico cuya dificultad radica más en la cantidad de conocimientos que se dan en poco tiempo (y su interrelación), que en la dificultad de los conocimientos en sí mismos.

Debido a esta gran cantidad de conocimientos y a su fuerte interrelación, es muy recomendable para el alumno llevar la asignatura al día. Sin embargo, observamos que los alumnos sólo estudiaban cara a los exámenes, lo que provocaba que las clases, y especialmente los laboratorios, no fueran aprovechados: al no llevar la asignatura al día, en las clases magistrales los alumnos eran meros copiadores de pizarras, mientras que en el laboratorio les faltaban conocimientos básicos, pero sobre todo la experiencia de haber resuelto muchos ejercicios con anterioridad.

En el año 2003 se realizó en nuestra facultad un cambio en el plan de estudios, que obligaba a realizar una reforma completa de la asignatura (e intentar adaptarla al EEES [14]). Se aprovechó esta oportunidad para buscar una metodología que implementara evaluación continuada y que hiciera muy atractivo para el estudiante llevar la asignatura al día. Nuestros objetivos eran:

1. Que los estudiantes no concentraran su dedicación al estudio de la asignatura en los días previos al examen parcial y al final, sino a lo largo del curso.
2. Que, al llevar la asignatura más al día, las clases y los laboratorios fueran más aprovechados (y pudiéramos ofrecer un mejor *feedback*), separando la evaluación formativa de la sumativa [15].
3. Todo esto sin aumentar las horas de trabajo del estudiante y sin incrementar en exceso la carga de trabajo del profesor.

Nuestra propuesta consiste en cambiar el sistema de evaluación tanto de la parte teórica del curso, como de la parte práctica o de laboratorio. En la parte teórica se permite que los alumnos que superen la evaluación continuada aprueben la asignatura sin sufrir el examen final. En la parte de laboratorio, se ha orientado la evaluación al intercambio de información que sostiene el alumno con su profesor. Como veremos, se han cumplido los objetivos y a un coste razonable.

Este artículo está organizado como sigue: la sección 2 presenta el trabajo relacionado. La sección 3, el método de evaluación propuesto. La sección 4 analiza el coste de la propuesta para profesores y alumnos. La sección 5 analiza los resultados obtenidos en los cinco cuatrimestres en que se ha impartido esta asignatura (el lector puede encontrar un estudio más completo en [8]). Finalmente, la sección 6 presenta nuestras conclusiones.

2. Trabajo relacionado

En [15] se introduce la distinción entre evaluación formativa y evaluación sumativa:

- Evaluación formativa: la información se usa para guiar y mejorar los procesos de enseñanza y aprendizaje.
- Evaluación sumativa: la información se usa para determinar la calificación que acredita el nivel de aprendizaje alcanzado por el alumno.

En la evaluación formativa, prima la prontitud sobre la precisión y fiabilidad de la evaluación: es más importante que el alumno sepa pronto si sus decisiones han sido acertadas, que saber con exactitud su nota. Por contra, en la evaluación sumativa está en juego el expediente del alumno, por lo que debe ser precisa y fiable, aunque no se le ofrezca al alumno con prontitud.

En el mencionado artículo, se desarrolla un sistema que ofrece prontitud a bajo coste consistente en la autoevaluación y la co-evaluación por parte de los alumnos.

Diversos artículos proponen una evaluación continuada donde ésta cuenta parte de la nota, pero en general, la evaluación continuada no libera del examen final: en [10] se ofrece una reflexión sobre cómo influye que el trabajo diario tenga un peso en la nota final, pero en su propuesta la evaluación continuada cuenta un 60%. En [11] se propone evaluación continuada de la parte de laboratorio junto con un examen final, indicando el alto coste para el profesorado. En [3] se ofrece un incremento de hasta dos puntos sobre la nota del examen final a los alumnos que siguen la evaluación continuada.

En general, se elimina el examen final en asignaturas de laboratorio [4] o con métodos educativos lejanos a los tradicionales, como el *Project-Based Learning* [2], implementado entre otros en [7][9], o la experiencia de cambio de evaluación final a continuada basada en seminarios de [12], donde los alumnos que han realizado todos los seminarios no tienen porqué presentarse al examen final; sin embargo, los autores se quejan de la gran carga de trabajo. Incluso en algunas experiencias novedosas como Áncora [1], se acaba realizando un examen final.

También encontramos en la literatura artículos sobre la problemática de los laboratorios y cómo ofrecer *feedback* rápido [13] o actividades para aumentar la motivación de los alumnos [6].

3. La propuesta

3.1. Objetivos

La primera pregunta que nos hicimos fue: ¿porqué los alumnos no llevan nuestra asignatura al día? Encontramos tres motivos:

- Falta de costumbre: los alumnos no están acostumbrados a realizar un esfuerzo continuado de manera voluntaria, sino más bien por estímulos tipo examen.
- Falta de motivación: los alumnos percibían la asignatura como "aprobable con un esfuerzo antes del examen". Que al poco tiempo se olvidaran los conocimientos parecía no tener importancia: primaba aprobar sobre aprender.
- Competencia: esta asignatura tiene una dura competencia en asignaturas de programación (al mismo nivel dentro del plan de estudios) y muy costosas en esfuerzo para los alumnos.

En resumen, los alumnos deciden que es preferible dedicar su esfuerzo durante el curso a otras asignaturas (que lo requieren, ya que son muy absorbentes) y dedicar a SO solamente un esfuerzo al final ante los exámenes, ya que se percibe como fácil de aprobar.

No estaba en nuestra mano trabajar sobre la falta de costumbre: sólo un cambio global en el sistema podría acostumbrar a los alumnos a estudiar de manera continuada. Igualmente, la solución no pasaba por pedir a nuestros compañeros de "la competencia" que suavizaran su asignatura, sino en motivar a los alumnos a prestar una atención más continua a SO.

Se podía haber aumentado el número de entregas puntuables (a semanalmente, por ejemplo), pero ello implica una gran presión para el estudiante y un gran esfuerzo para el profesor, por lo que no optamos por este método.

Para incentivar que los estudiantes lleven la asignatura al día, les ofrecemos la oportunidad de aprobar la asignatura sin necesidad de realizar el examen final. Sin embargo, no es obligatorio seguir la evaluación continuada, así que un estudiante que lo desee, puede obtener el 100% de su nota en exámenes finales.

La normativa de la nuestra universidad obliga a que la nota final no dependa de un único acto evaluador. En SO se divide la nota en dos partes: la nota de teoría (NT) y la nota de laboratorio (NL), que cuentan un 70% y un 30% de la nota

final, respectivamente. Se exige un mínimo de 3.5 de nota de laboratorio para aprobar la asignatura; si no se obtiene esta nota mínima, la nota final (NF) no puede superar el cuatro, es decir que si NL<3.5, NF=min (NT*0.7+NL*0.3 , 4). Esta es una condición indispensable para forzar a los alumnos a participar en el laboratorio.

3.2. Nota de teoría

La nota de teoría es, por definición, la que se obtiene en el examen final. Sin embargo, se ofrece la oportunidad de aprobar por controles, sin sufrir el examen final.

A lo largo del curso se realizan tres controles, cada uno de los cuales incluye todo el temario impartido hasta el momento. El curso normal es de 15 semanas reales (13 efectivas, eliminando festividades y puentes), realizándose los controles habitualmente la sexta, décima y decimoquinta semanas. Los dos primeros controles son de una hora, mientras que el tercero es de dos horas. El peso de cada control es proporcional a sus horas, de manera que el primero y el segundo tienen un peso del 25% cada uno, y el tercero del 50%.

Los requisitos para superar la evaluación continuada son: aprobar los 3 controles o, caso de haber suspendido alguno, tener una nota media igual o superior a 6.

Los alumnos que han superado la evaluación continuada pueden decidir entre quedarse con su nota de controles como nota final de teoría, o bien presentarse al examen final. Si entregan el examen final, pierden la nota de evaluación continuada y su nota de teoría será la del examen. Los alumnos que no siguen o no superan la evaluación continua tienen como nota de teoría la del examen final.

3.3. Nota de prácticas o laboratorio

A lo largo de las 13 semanas efectivas del curso se realizan 10 sesiones puntuables de laboratorio de dos horas cada una. Las dos primeras semanas no hay laboratorio, ya que aún no se dispone de los conocimientos necesarios para la primera sesión. La última semana del curso se reserva para el examen de laboratorio.

Cada sesión de laboratorio consiste en un trabajo de unas dos horas que debe realizar el estudiante antes de la sesión, más un enunciado con diversos problemas, repartido al principio de la sesión, que se debe resolver en las dos horas del laboratorio. Los grupos de laboratorio son de un máximo de 20 estudiantes, y las prácticas se realizan de manera individual.

Durante la sesión, el profesor de laboratorio habla con todos los alumnos, discutiendo sus propuestas, preguntando dónde encuentran dificultades y ofreciendo un *feedback* continuo. Las prácticas están diseñadas para resolverse en las dos horas de laboratorio, siempre y cuando se haya realizado el trabajo previo.

El alumno que no puede acabar las prácticas nota enseguida que no lleva la asignatura al día, y además es consciente de que su profesor de laboratorio también lo sabe.

Al final de la sesión, los alumnos entregan lo que han hecho, aunque han sido informados que su nota no depende de esta entrega (que sólo es información adicional), sino de la evaluación que ha hecho el profesor de sus conocimientos y evolución a lo largo de las sesiones.

Al acabar las 10 sesiones, el profesor publica una nota subjetiva para cada alumno. Esta nota refleja los conocimientos demostrados ante el profesor, y la dedicación y evolución que ha observado en el alumno. Esta nota no es revisable ni discutible.

Los alumnos que no han obtenido una nota subjetiva igual o superior a 5, o bien no están de acuerdo con su nota, pueden presentarse al examen de laboratorio que se realiza la última sesión del curso. El examen consta de entre 3 y 5 problemas similares a los realizados en cada sesión. La nota final de laboratorio es el máximo entre la nota subjetiva y la nota de examen (lo que quiere decir, en particular, que si alguien ya está de acuerdo con la nota subjetiva, no necesita presentarse al examen de laboratorio).

3.4. Bondades de la propuesta

Cada vez que hemos explicado nuestra propuesta a algún colega, nos ha planteado dudas sobre las bondades del sistema. Intentaremos contestar aquí algunas de las preguntas más comunes.

- "No me gusta eliminar el examen final, sirve para tener una visión global de la asignatura". Cierto, pero los controles son acumulativos, de manera que el tercero (que se realiza al final del curso) incide sobre la visión global.
- "Los alumnos que suspendan el primer control abandonarán la evaluación continuada". No, dado que no es necesario superar los tres controles para aprobar: en caso de suspender

alguno, la media ha de ser igual o superior a 6, pero como el tercer control cuenta un 50%, saben que pueden recuperarse y seguir la evaluación continuada (excepto aquellos que han sacado muy mala nota, que al menos han detectado el problema la sexta semana del curso). Además, los alumnos siguen percibiendo los controles como un buen entrenamiento para el examen final.

- "Los alumnos que aprueben con buena nota los dos primeros controles, descuidarán la asignatura". Al contrario: como el tercer control cuenta un 50%, un alumno con una media de 7.5 en los dos primeros controles sabe (insistimos en ello en clase) que un 4 en el tercero implica suspender la evaluación continuada (ya que la media es inferior a 6, de ahí esta restricción), y el premio -no ir al examen final- es muy apetecible.
- "La nota subjetiva del laboratorio puede ser injusta". Cierto, pero la nota proviene de las observaciones del profesor durante 20 horas de laboratorio. Las prácticas son individuales, así que los alumnos que superan el laboratorio han demostrado conocimientos suficientes. Podría ser injusto para alumnos tímidos o que les cueste comunicarse, pero para ello está el examen de laboratorio.
- "La evaluación continuada debería ser voluntaria". Estamos totalmente de acuerdo: por eso la asignatura se puede aprobar presentándose solamente al examen de laboratorio y al final.
- "Si todas las asignaturas hicieran lo mismo, seria mucha carga para el estudiante". Al contrario: si todas las asignaturas siguieran nuestro método, el estudiante distribuiría su tiempo de manera más óptima (véase el punto 4.3). De hecho, pedimos unas 150 horas de trabajo (incluyendo asistencia a clases, laboratorios, etc...) a lo largo de 15 semanas para una asignatura de 9 créditos; un estudiante se matricula en media de 37.5 créditos al cuatrimestre, por lo que si todas las asignaturas hicieran lo mismo, el alumno debería dedicar 41.6 horas semanales a sus estudios, lo que es perfectamente razonable.
- "El coste de implementar esto debe ser muy grande para el profesor". Esto lo discutimos en el siguiente punto.

4. Costes de la propuesta

Como ya se ha comentado, una de las quejas de la implementación de la evaluación continuada es el alto coste para el profesor y para el alumno. ¿Qué cuesta esta propuesta?

4.1. Para el profesor: coste de evaluación de la teoría

En la parte teórica, comparamos un método con un examen parcial y un final obligatorio contra un método con tres controles y un examen final. En la evaluación continuada, los estudiantes que aprueban la asignatura por controles no se presentan al examen final: si *entregan* su examen, saben que la nota de teoría será la de examen y ante la posibilidad de suspender una asignatura ya aprobada, sólo se arriesgan a entregar aquellos que realmente están seguros de subir la nota. En los cinco cuatrimestres de la experiencia, sólo tres alumnos de casi ochocientos han entregado el examen final con la asignatura aprobada, y los tres han mejorado su nota.

Respecto a los enunciados de exámenes, en ambos casos hay que elaborar un examen final, pero en el caso de la evaluación continuada deben elaborarse tres controles por grupo de teoría, mientras que en la evaluación final sólo era un parcial por grupo de teoría. En este caso hay un incremento de coste.

En cuanto a la corrección, como los exámenes son similares en los dos tipos de evaluación, consideramos que el esfuerzo de corrección es proporcional a las horas del examen. Si un estudiante se presenta a un parcial de dos horas y a un final de tres horas, el profesor debe corregir el esfuerzo de cinco horas de examen por parte del alumno; si el estudiante se presenta a dos controles de una hora, y un control de dos horas y libera el examen final, el profesor debe corregir el esfuerzo de cuatro horas de examen.

En el primer cuatrimestre del curso 2006-2007 la asignatura ha tenido 143 matriculados donde 131 se presentaron al primer control, 122 al segundo y 102 al tercero; finalmente, 47 alumnos se presentaron al examen final. Por tanto, los profesores corregimos el resultado de 131x1h +122x1h+102x2h+47x3h = 598 horas de examen.

Para comparar con el método tradicional, supongamos que todos los alumnos que se han presentado al primer control se hubieran

presentado al parcial: tendríamos 131 presentados a un parcial de dos horas. El ultimo cuatrimestre antes del cambio de método, en la asignatura se presentó al examen final el 89% de los alumnos; asumamos pues que se presentaran al examen final el 89% de los 143 matriculados (127). En este caso hubiéramos corregido 131x2h+127x3h = 643 horas de examen.

Esta diferencia no implica que el nuevo método tenga un coste menor que el tradicional, pero creemos que demuestra que la evaluación continuada no tiene un coste mayor. Además, la corrección está más distribuida a lo largo del curso.

4.2. Para el profesor: coste de evaluación del laboratorio

El coste principal de las prácticas consiste en la elaboración del cuaderno de laboratorio, con lecturas previas y una lista de ejercicios. Pero este trabajo se realiza una vez y no cada cuatrimestre, aunque requiera mantenerlo actualizado. Por tanto, podemos considerar este trabajo como el de creación de la asignatura. Una vez tenemos el cuaderno, el trabajo del profesor consistirá en:

- Elaborar un enunciado para cada grupo y sesión: es un enunciado corto, habitualmente basado en ejercicios de la colección, y que se reaprovechará para actualizar dicha colección. El tiempo para elaborar cada enunciado es bastante pequeño. Además, en la mayoría de los sistemas "tradicionales" hay que elaborar enunciados igualmente.
- La gran ventaja está en la corrección de las 10 sesiones, que se realiza durante las horas presenciales del laboratorio: la nota subjetiva resulta de las conversaciones profesor-alumno, lo que significan *feedback* para el estudiante e información para el profesor.
- Pocos estudiantes se presentan al examen final de laboratorio (una media del 10%), ya que la mayoría han aprobado el laboratorio o bien han renunciado, de manera que la corrección del mismo tampoco lleva mucho tiempo.

Es difícil comparar nuestro sistema con uno "tradicional", ya que hay muchas maneras de implementar un laboratorio. Sin embargo, nuestra propuesta solapa las sesiones de laboratorio con su evaluación, con lo que podemos ver que el coste es bastante pequeño.

4.3. Para el estudiante: créditos ECTS

El ECTS [5] es una unidad de valoración de la actividad académica del estudiante, que tiene en cuenta la asistencia del estudiante a clases magistrales, el tiempo de estudio personal, el de preparación de ejercicios y prácticas de laboratorio, la realización de ejercicios y entregas, así como la realización de exámenes.

En la asignatura que nos ocupa, un estudiante asiste durante 13 semanas reales a 4 horas de clases teóricas durante la semana (que incluyen los controles), a las cuales debemos añadir 4 horas más de estudio semanal. Hay 10 sesiones de laboratorio de 2 horas, a cada una de las cuales deberían dedicar dos horas más para su preparación. En total 144 horas de dedicación caso de aprobar por evaluación continuada; si el estudiante se presentara al examen de laboratorio y al de teoría se deberían sumar 5 horas de examen (2 y 3, respectivamente) y otras 6 para repasar la asignatura. En total, entre 144 y 155 horas de trabajo.

Lo más importante de este punto es que calculamos el esfuerzo que debería dedicar un estudiante a la asignatura, independientemente de si sigue un modelo tradicional o la evaluación continuada. Esta última debería servir para forzar al estudiante a distribuir su esfuerzo de manera continuada a lo largo del cuatrimestre.

5. Evaluación de la propuesta

Para evaluar nuestra propuesta, vamos a tener en cuenta cinco factores: seguimiento y resultados de la evaluación continuada; dedicación a la asignatura fuera de horas de clase; comparación de resultados con la metodología; opinión del alumnado respecto al método de evaluación y, finalmente, apreciación personal de los profesores.

5.1. Seguimiento y resultados de la evaluación continuada

Habitualmente hay entre 130 y 170 alumnos matriculados en la asignatura. El último cuatrimestre (otoño del curso 2006/07), hubo 143 estudiantes matriculados. Presentaremos los resultados sólo para este cuatrimestre, ya que los cuatrimestres anteriores son muy similares.

Por lo que respecta a la parte de teoría, en este cuatrimestre un 70% de los matriculados ha

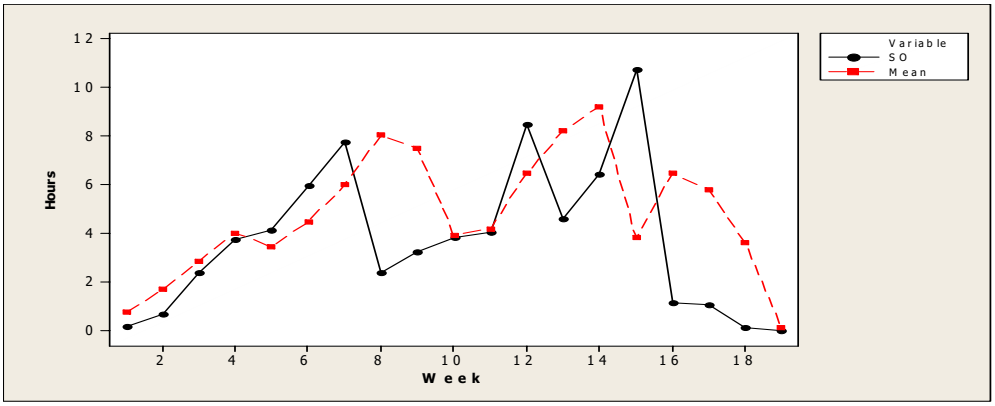


Figura 1: Horas semanales de dedicación a SO (continua) y a las asignaturas del mismo nivel (discontinua)

seguido completamente la evaluación continuada (es decir, ha realizado los tres controles) mientras que otro 15% ha empezado dicha evaluación pero han decidido abandonarla (sobre todo debido a las notas que han obtenido en los dos primeros controles). Es decir, un 85% de los alumnos han intentado seguir la evaluación continuada.

Del 70% de alumnos que siguieron la evaluación continuada de manera completa, un 78% consiguió aprobar la teoría sin tener que realizar el examen final (o sea, un 54.6% del total). Es necesario resaltar que no hay diferencia en el nivel de dificultad de los controles y el del examen final, por lo que los requisitos para aprobar la asignatura son los mismos.

En cuanto al laboratorio, el 95% de los alumnos siguieron la evaluación continuada. Controlar el nivel de aprendizaje del alumno durante todas las sesiones del laboratorio y ofrecer un *feedback* inmediato en cada sesión provocó que casi el 89% de los estudiantes aprobasen. Menos de un 2% de los estudiantes aprobados se presentaron al examen de laboratorio a subir su nota.

En resumen, la mayoría de los estudiantes han seguido la evaluación continuada con un alto grado de éxito.

5.2. Dedicación a la asignatura

Un estudiante tiene que repartir sus horas de estudio entre todas las asignaturas matriculadas, por tanto, la evaluación continuada no debe suponer más horas de estudio fuera de horarios de clase que los calculados en ECTS, ni concentrar las horas de estudio justo antes de los controles.

Nuestra Facultad tiene un proyecto en que estudiantes voluntarios indican, cada semana, el tiempo que dedican a cada asignatura distribuido en teoría, laboratorio, estudio, etcétera (proyecto ECTS). La Figura 1 muestra estos datos para el segundo cuatrimestre del curso 2005-2006, en el que el número de estudiantes que formaban la muestra fue de un 14% del total de matriculados.

En el cuatrimestre en que se ubica nuestra asignatura, hay cuatro asignaturas más, una de ellas de proyecto. La línea continua indica el número de horas en media que dedican los alumnos a nuestra asignatura, mientras que la línea discontinua indica la media las cinco asignaturas del cuatrimestre (horas de estudio personal, sin contar con las horas de clase ni laboratorio).

En la Figura 1 se aprecian dos características interesantes. La primera de ellas es el hecho de que el número de horas dedicadas a seguir la evaluación continua es parecido al del resto de asignaturas. Esto es muy importante ya que conseguimos que el alumno siga la evaluación continuada sin una carga adicional de trabajo.

La segunda característica de la Figura 1 es que los estudiantes dedican un tiempo mínimo todas las semanas a estudiar la asignatura; no desglosamos los datos de las otras asignaturas, pero queremos apuntar que SO es la única asignatura donde los estudiantes trabajan todas las semanas y la que tiene las fluctuaciones más pequeñas en la curva.

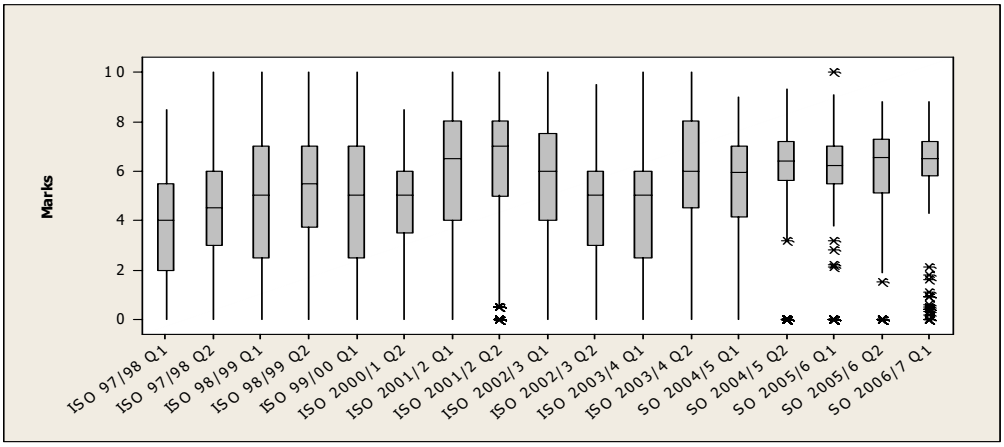


Figura 2: comparativa de notas ISO-SO

5.3. Comparativa de resultados con la metodología tradicional

La asignatura Sistemas Operativos (SO) es una ampliación de la asignatura Introducción a los Sistemas Operativos (ISO) del plan de estudios anterior. La asignatura SO ya nació con el método de evaluación que proponemos en este artículo, mientras que ISO tenia un método tradicional consistente en un examen parcial (optativo, que podía subir la nota del final, pero en ningún caso bajarla) y un examen final, mientras que la evaluación del laboratorio consistía en un examen final práctico. Son por tanto SO e ISO las asignaturas que debemos comparar.

La Figura 2 presenta las notas finales de ISO desde el curso 1997-1998 hasta su desaparición el curso 2003-2004, y las notas finales de SO a partir de dicho curso. Se pueden ver las notas divididas en cuartiles (los asteriscos corresponden a *outlayers*). Para esta comparativa es necesario resaltar que el nivel de dificultad de los exámenes en las dos asignaturas es el mismo.

De la Figura 2 destacamos dos detalles. El primero es que la nota media de SO es, en general, igual o superior a la de ISO. Deducimos que, dado que exigimos el mismo nivel de dificultad y sólo cambia la metodología, conseguimos que los alumnos aprovechen mejor su esfuerzo y adquieran los conocimientos de una manera mejor, lo que se refleja en su nota final.

La segunda observación es que la dispersión de notas es más pequeña en SO que en ISO. Deducimos de esto que, como la nota depende de más actos evaluadores, hay menos notas extremas. Es decir, en ISO el examen final marcaba la nota, mientras que el parcial y el laboratorio sólo realizaban ligeros ajustes a la nota final. Por tanto, la influencia del examen final en la nota era muy grande, y al depender tanto de un único acto evaluador era más común obtener notas extremas (dependiendo de lo bien o mal que fuera el examen final). Por tanto, consideramos que con este método evitamos perjudicar a los alumnos que "tienen un mal día". De la misma manera, hemos observado que se ha reducido drásticamente el número de sobresalientes y matrículas de honor, con lo que el factor suerte (alumnos a los que se les preguntó justo aquello que sabían) también ha desaparecido.

Estas dos características nos inducen a pensar que esta metodología provoca un mejor aprovechamiento del esfuerzo del alumno, y una evaluación más justa.

5.4. Opinión del alumnado

Hemos preguntado a los alumnos su opinión sobre el método de evaluación continuada. Para ello, les pasamos una encuesta de siete afirmaciones en la cual, entre otras cosas, les pedíamos su opinión sobre la metodología. Para la afirmación "Creo que la evaluación continuada me ha ayudado a llevar la asignatura al día", un 91% de los

alumnos marcaron "totalmente de acuerdo", mientras que un 88% marcó lo mismo para la afirmación, "Creo que la evaluación continuada me ayudará a aprobar la asignatura".

Creemos que estos números indican que el método es bueno desde el punto de vista del alumno.

5.5. Apreciación personal

Quizá la medida que más indica el éxito o el fracaso de una propuesta es la apreciación personal de los profesores que la han realizado. También es la más difícil de justificar, y donde más fácil es el autoengaño. En este caso, creemos que presentamos suficientes indicadores como para considerar que la propuesta es un éxito. Sin embargo, lo que más nos hace pensar que la nueva metodología es mejor es una sensación de que las clases están más aprovechadas y que cuando construyes una explicación sobre conocimientos explicados con anterioridad, los alumnos responden muy bien.

La única queja que tenemos es que a los alumnos les cuesta adaptarse a la filosofía del trabajo continuado, de manera que cuesta que realicen el trabajo previo a los laboratorios, problema que estamos intentado resolver en la actualidad.

6. Conclusiones

Se ha presentado una implementación de evaluación continuada que no requiere un esfuerzo adicional ni para el profesor ni para el alumno. El método separa la evaluación formativa de la evaluación sumativa, ofreciendo un *feedback* rápido a los alumnos y motivándolos a llevar la asignatura al día. La evaluación del método demuestra que el cambio no ha implicado un incremento de coste para nadie, que los alumnos siguen la evaluación continuada, y que perciben que les ayuda a aprobar la asignatura.

Agradecimientos

Queremos agradecer su entusiasmo a todos los profesores (especialmente a los tiempos parciales) que han impartido la asignatura. Este trabajo ha sido apoyado por el Ministerio de Educación y Ciencia, bajo proyecto TIN 2004-07739-C02-01.

Referencias

[1] Bofill P. et al. *ÁNCORA: Aprendizaje Organizado por Tareas*. JENUI 2005, p. 71.

[2] Buck Institute for Education. *Project Based Learning* [On line http://www.bie.org/pbl; fecha de consulta: 12 de enero de 2007]

[3] Canaleta X., Vernet D. *Propuesta de créditos ECTS para la asignatura Programación de primer curso de Ingenieria*. JENUI 2004, p. 417.

[4] Durán M., Caro A. y Rodríguez P.G. *Una apuesta por la motivación al alumnado en las asignaturas de programación: el sistema de evaluación continuada*. JENUI 2003, p. 191.

[5] European Comission. *European Credit Transfer and Accumulation System*. [On line http://ec.europa.eu./education/programmes/socrates/ects/index_en.html; fecha de consulta 15 de enero de 2007]

[6] Garrido A. *Una experiencia para fomentar la motivación en prácticas de una asignatura de desarrollo de software*. JENUI 2005 p.239.

[7] Lacuesta R. y Catalán C. *Aprendizaje Basado en Problemas: Una experiencia interdisciplinar en Ingeniería Técnica en Informática de Gestión*. JENUI 2004, p. 305.

[8] López D. et al. "Implantación de la evaluación continuada en SO". Report técnico UPC-DAC-RR-2007-11, UPC, 2007

[9] Machado S. et al. *Recomendaciones para la implantación del PBL en créditos optativos basados en la experiencia en la EPSC*. JENUI 2005, p. 21.

[10] Marqués M., Tomás V.R. y Sanz I. *Tratando de fomentar la motivación del estudiantado*. JENUI 2004, p. 329.

[11] Palomo Lozano F., Medina Bulo I. *Una propuesta para una primera asignatura de Algoritmia*. JENUI 2004, p. 393.

[12] Poza Luján J.L. y Cruz Alcázar P.P. *Una experiencia de cambio de evaluación final a contínua*. JENUI 2006, p.186.

[13] Riesco M. y Díaz M. *Sistema Docente de Realimentación Inmediata en Clases Prácticas*. JENUI 2005, p. 29.

[14] Sánchez Carracedo F. *¿Cómo serán las asignaturas del EEES?*. JENUI 2005, p. 147.

[15] Valero-García M., Díaz de Cerio L.M. *Evaluación continuada a un coste razonable*. JENUI 2003, p. 183.

Hacia la Evaluación Continua Automática de Prácticas de Programación

Juan Carlos Rodríguez del Pino, Margarita Díaz Roca, Zenón Hernández Figueroa, José Daniel González Domínguez
Dpto. de Informática y Sistemas
Universidad de Las Palmas de Gran Canaria
{jcrodriguez,mdiaz,zhernandez,dgonzalez}@dis.ulpgc.es

Resumen

En el contexto de implantación y aplicación del Espacio Europeo de Educación Superior se hace necesario disponer de herramientas docentes que faciliten el seguimiento y la orientación personalizada y continuada del proceso de aprendizaje del alumno. En este trabajo se presenta un sistema de evaluación continua automática de prácticas de programación, vía web, basado en la realimentación. De esta manera, la evaluación proporciona información inmediata al alumno de los errores cometidos en las prácticas, permitiendo que puedan ser corregidas y reevaluadas repetidas veces. Este esquema establece un marco de evaluación, donde el alumno puede aumentar su motivación y decidir la nota que desea obtener. Los errores susceptibles de ser identificados automáticamente son de diversos tipos: incumplimiento de especificaciones de interfaz, errores de estilo y de diseño de codificación, errores de funcionalidad, errores graves de ejecución e insuficiencias en las pruebas.

1. Introducción

En las titulaciones de informática, y también en otras de perfil tecnológico, existen asignaturas que incorporan en su metodología la realización de prácticas en un lenguaje de programación. Normalmente, la elaboración de estos trabajos está supervisada por los profesores y se lleva a cabo en un laboratorio donde el alumno dispone del material informático necesario para su realización. Aunque, actualmente, la mayoría de ellos disponen de este material en sus casas, lo que les ofrece mayor flexibilidad, tanto espacial como temporal, a la hora de realizar las prácticas.

Los trabajos prácticos tienen unos plazos de entrega prefijados, tras los cuales son evaluados, pero hasta que se produce la revisión por parte del profesor, el alumno desconoce si es correcto o no. Dependiendo de las circunstancias de la asignatura —no es lo mismo una troncal de primer curso que una optativa de quinto—, la evaluación puede requerir bastante tiempo y esfuerzo del profesor debido a la cantidad de alumnos y el número y complejidad de los trabajos. Este modo de actuación satisface el aspecto evaluativo, pero no facilita que el alumno aprenda de sus errores, lo que hace que se pierda una parte importante del potencial formativo que conlleva la realización de una práctica.

La adaptación de los métodos de enseñanza tradicionales al Espacio Europeo de Educación Superior plantea nuevos retos al profesor que están condicionados por el esfuerzo necesario para alcanzarlos. La disponibilidad de una herramienta docente que facilite el seguimiento y la orientación personalizada y continuada del proceso de aprendizaje del alumno posibilita superar este reto con un coste asumible.

En este trabajo se presenta un sistema de evaluación continua y automática de prácticas de programación vía web basado en la realimentación. Esta herramienta lleva en funcionamiento desde el curso 2004/2005 en diversas asignaturas del primer y segundo curso de las titulaciones de informática de la Universidad de Las Palmas de Gran Canaria, abarcando a unos 650 alumnos por curso académico.

La motivación para desarrollar este sistema de evaluación/enseñanza ha sido que el profesor pueda realizar una evaluación continua y disponer de información personalizada sobre cada discente, además de facilitar la adquisición de conocimientos al alumno mediante una información detallada de los fallos cometidos. Con este sistema de evaluación se estimula al

además de facilitar la adquisición de conocimientos al alumno mediante una información detallada de los fallos cometidos. Con este sistema de evaluación se estimula al alumno a corregir sus errores y a comprobar de inmediato sus progresos.

1.1. Estado del arte

Varios grupos de profesores de diversas universidades españolas han empezado a incorporar en la evaluación de sus asignaturas algún tipo de herramienta automática [10]. En general, se centran en valorar los conocimientos teóricos adquiridos por los alumnos mediante pruebas tipo test, o bien en el desarrollo de una herramienta de evaluación automática de prácticas con el fin de ahorrar tiempo y esfuerzo de corrección al docente, olvidando las ventajas que aporta la tecnología [2] en cuanto a la posibilidad de comunicar resultados y, en caso de que existan fallos, la posibilidad de su corrección —a los pocos que abordan esta última opción les falta integrar la herramienta con alguna plataforma de campus virtual [3]—. A nivel internacional se han desarrollado herramientas de evaluación automática con realimentación, bien dirigidas a lenguajes de programación concretos como C++ [1], Scheme [13], Ada [7], o bien dirigidas al manejo de estructuras de datos mediante ejercicios de simulación [9]. La que más se parece a la propuesta en este trabajo es BOSS [8], siendo la principal diferencia que la información que se muestra al alumno como realimentación es significativamente menor que la que dispone el profesor para la evaluación. La metodología empleada por los autores de BOSS hace más hincapié en proporcionar a los alumnos pruebas *offline* para comprobar sus prácticas.

2. Criterios típicos de evaluación de prácticas de programación.

Antes de abordar la evaluación automática de prácticas de programación, es necesario repasar brevemente los criterios típicos que se suelen usar en dicha evaluación, a fin de estar en condiciones de discernir el alcance potencial de la evaluación automática, identificando lo que es posible evaluar y descartando lo que no lo es, al menos por el momento.

La evaluación de prácticas de programación suele estar basada [6] en varios criterios: diseño, ejecución, cumplimiento de especificaciones, estilo y creatividad.

El criterio de diseño se refiere a aspectos tales como: organización, descomposición modular, selección de estructuras y algoritmos, etc. El criterio de ejecución se ocupa de los fallos en el funcionamiento del programa, tanto los que reflejan incumplimientos de las especificaciones, como los que se deben a un mal uso de los mecanismos del lenguaje. El criterio de cumplimiento de especificaciones tiene en cuenta las especificaciones no funcionales, como la obligatoriedad de usar, o no, determinadas estructuras o construcciones —por ejemplo, que un determinado subprograma tenga que ser recursivo. El criterio de estilo trata de todo aquello que ayuda a la legibilidad y comprensión de un programa —comentarios, sangrado, espacios, nombres,... Por último, el criterio de creatividad premia la implementación de soluciones especialmente interesantes o innovadoras.

3. Herramienta empleada

Como soporte para la evaluación automática se ha empleado el programa de Gestión Automática de Prácticas GAP [11][12] . Una de las características del GAP 2.0 permite la ejecución de herramientas externas de forma segura, mediante la simple configuración de líneas de comando del sistema operativo, lo que resulta muy útil a la hora de realizar las pruebas automáticas de evaluación.

Para la ejecución de las pruebas automáticas de un ejercicio basta con escribir un guión usando el lenguaje de *script* del sistema operativo; este guión se ejecutará en el momento de la entrega —si así se ha configurado—, llevando a cabo las pruebas que en el mismo se hayan dispuesto y generando información sobre los resultados, que el GAP se encargará de mostrar al alumno convenientemente filtrada y formateada. El profesor puede ejecutar el mismo guión, u otro distinto sobre todas las prácticas entregadas si desea obtener información distinta o actualizar la que recibió el alumno en el momento de la entrega.

Dado que el código ejecutado podría contener partes maliciosas, el programa GAP ejecuta estas

pruebas en un entorno con privilegios restringidos mediante una cárcel *chroot* [4]. En una ejecución, el sistema realiza los siguientes pasos:

1. Crea un directorio temporal en la cárcel.
2. Transfiere a ese directorio una copia de los ficheros entregados por el alumno y de los preparados por el profesor para las pruebas.
3. Prepara el *script* elaborado por el profesor para ser ejecutado en dicho directorio.
4. Transfiere el control al *script*, aunque encarcelándolo y controlando su ejecución.
5. Finalizada o detenida la ejecución se recogen los resultados y se almacenan.
6. Se elimina el directorio temporal.

Este trabajo, usando estas características del GAP, describe cómo se puede establecer un sistema de evaluación automática con programas específicos y la ayuda de herramientas externas.

4. La revisión automática

El primer aspecto a tener en cuenta a la hora de preparar prácticas que van a ser probadas automáticamente es la ausencia de ambigüedad en su especificación. Las tareas encomendadas deben ser claras y concretas. Una evaluación automática sólo es posible si la variabilidad permitida está bajo control. Se puede argumentar en contra que esto disminuye las posibilidades de creatividad, pero también es cierto que, especialmente cuando el alumno es novel en la materia, puede ser muy beneficioso encarrilarle hacia un conjunto más o menos limitado de opciones. La alternativa sería dejarlo frente a una multiplicidad de posibilidades que, si no es capaz de controlar, fácilmente puede conducirlo al desánimo y al abandono.

Algunas de las revisiones que se realizan requieren la utilización de un analizador sintáctico que proporcione información acerca de múltiples aspectos del código, relacionados fundamentalmente con el cumplimiento de las especificaciones y el estilo. Aunque puedan existir herramientas estándar, el desarrollo de un analizador específico integrado en las pruebas, que permita acceder programáticamente a la información que suministra, presenta muchas ventajas. En la experiencia que aquí se expone se han desarrollado analizadores simples para C, C++ y Ada.

La calificación automática se basa en la penalización de los errores cometidos. Esta evaluación evita castigar reiteradas veces por el mismo tipo de error, así como que la acumulación de errores en una parte de la práctica invalide la valoración de otras partes.

4.1. Incumplimiento de especificaciones

Esta tarea se realiza normalmente compilando el código suministrado por el alumno junto con el correspondiente código de prueba. Se comprueban aspectos como que los nombres de los procedimientos, funciones, paquetes, y otros módulos coincidan con los indicados o que los tipos o el orden de los parámetros son los establecidos en la especificación, pudiéndose alcanzar un mayor grado de detalle con el apoyo del analizador sintáctico —por ejemplo, se puede informar al alumno de que se implementa la función *X* cuando se pide la función *Y*.

Otros aspectos a comprobar son: si un subprograma es recursivo cuando se ha especificado que lo sea, si se usan funciones o procedimientos que no están permitidos —por ejemplo, se pide que se implemente un procedimiento de manipulación de ristras sin usar las funciones de la biblioteca, y se usan, o si se emplean funciones de entrada y salida cuando se pide que no se empleen. Esta tarea se lleva a cabo en base a la información suministrada por el analizador sintáctico y la captura y análisis de la salida estándar.

4.2. Análisis de estilo

El analizador sintáctico se encarga de comprobar distintos elementos de estilo del código entregado por el alumno, tales como:

- Número, extensión y distribución de los comentarios.
- Calidad del sangrado.
- Longitud de las líneas de código.

En muchos casos, existen herramientas que permiten la realización de pruebas que usan guías de estilo estandarizadas sin necesitar un analizador específico —por ejemplo, el compilador *gnat* de Ada tiene opción de hacer comprobación de estilo—, aunque se complica la integración con el esquema de revisión al tener que procesar la salida de dicha herramienta.

4.3. Análisis de diseño

El análisis sintáctico puede suministrar información relativa al diseño de la solución, tal como:

- Número de subprogramas definidos.
- Tamaño de los subprogramas.
- Definición de subprogramas que no se invocan.
- Uso de variables globales.
- Complejidad ciclomática.
- Empleo de estructuras de control inadecuadas, como *goto*.

Los compiladores actuales también pueden proporcionar información adicional, por ejemplo: si se declaran variables que no se utilizan o si existe código inalcanzable.

4.4. Pruebas de funcionamiento

Las pruebas se realizan como pruebas unitarias, estableciendo casos de prueba para cada función o procedimiento que debe implementar el alumno de forma que cada caso sea independiente del resto.

Estas pruebas proporcionan información sobre errores de incumplimiento de la especificación funcional, además de manifestar errores graves de ejecución. Cuando se encuentra un error del primer tipo, se informa al alumno del caso probado que ha generado el error para que pueda depurarlo y corregirlo. En los errores graves, también se informa al alumno de las circunstancias que lo han provocado, pero, a diferencia de los otros, no se puede continuar la ejecución del caso de prueba.

Algunos de los errores graves que se pueden producir son: punteros sin inicializar, división por cero, índice fuera de rango, bucles infinitos, acceso a ficheros no abiertos, etc.

La actuación ante los errores graves depende bastante del lenguaje de programación. En el caso de lenguajes como Java, C# y Ada, los errores se manifiestan como excepciones que pueden ser controladas mediante los mecanismos suministrados al efecto por el propio lenguaje —las excepciones incorporan información sobre las circunstancias en que se han producido a la que se puede acceder mediante funciones específicas. En lenguajes como C o C++, el tipo de error se puede conocer capturando las señales del sistema operativo, pero a diferencia de los otros lenguajes, la continuación del resto de los casos de prueba se hace más compleja.

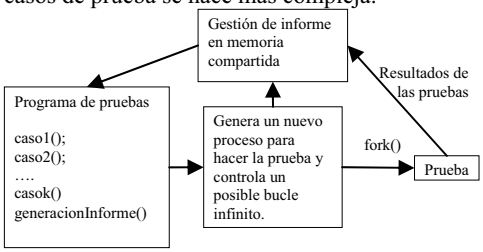


Figura 1. Gestión de errores graves en C y C++.

Para resolver este problema, cada caso de prueba en C y C++ se ejecuta en un proceso separado, lo que permite que un error grave en un caso no interrumpa la prueba del resto. El subprograma principal realiza la transferencia de información entre los distintos procesos usando memoria compartida y controla el tiempo de ejecución de cada caso para detenerlo si es necesario —por ejemplo, si hay un bucle infinito—, todo ello se ilustra en la figura 1. El subprograma en el que se ha producido el error se puede deducir de la función *backtrace* y de información sobre las funciones definidas por el alumno.

También, pueden producirse errores por mala utilización de los recursos —estos errores no siempre producen que la ejecución se detenga. Errores típicos de esta clase son, por ejemplo, en el lenguaje C: el mal uso de la memoria dinámica y el mal uso de ficheros.

La detección del mal uso de la memoria dinámica se hace sustituyendo las funciones de gestión de memoria (*malloc*, *free*, *realloc*,…) por funciones propias, que antes de realizar la tarea revisan su coherencia, pudiendo descubrir los siguientes errores:

- Toma de un bloque de memoria de tamaño erróneo (cero o exageradamente grande)
- Liberación de un bloque de memoria no tomado.
- No liberar bloques de memoria que ya no se usan.

Al igual que en el uso de la memoria dinámica, se pueden interceptar las funciones de gestión de ficheros (*fopen*, *fclose*, *fread*,…) sustituyéndolas por unas propias que comprueben la coherencia de su uso, pudiendo detectarse los siguientes errores:

- Usar un fichero no abierto.
- Abrir un fichero ya abierto.
- No cerrar los ficheros cuando termina la tarea a realizar sobre ellos.
- Uso incorrecto de parámetros.

4.5. Prueba de pruebas

Junto a la tarea de construir un código que cumpla unas especificaciones, a los alumnos se les solicita con frecuencia que añadan código que compruebe el correcto funcionamiento de la solución dada. La revisión de la prueba de los alumnos es muy compleja, ya que los casos a probar serían distintas implementaciones erróneas del código solicitado, para determinar si las pruebas encuentran el error. Normalmente, es más fácil realizar otras pruebas que obtengan información sobre la prueba del alumno. Una acción interesante es interceptar las llamadas realizadas desde las pruebas del alumno al código que prueba. Esto permite verificar cuántos casos se prueban para cada subprograma y revisar el tipo de casos examinados, aunque esto último es más complejo y difícil.

Con la experiencia, se ha comprobado que los alumnos tienden a escribir el código de prueba al final, después de haber comprobado —mediante el sistema de revisión automática— que el resto de su solución funciona bien, lo cual no es muy adecuado, por lo que se ha tomado como medida disuasoria el que, si no se realiza un número mínimo de pruebas, no se le informa de otros errores.

4.6. Herramientas externas

Existen herramientas muy útiles que pueden complementar las tareas de revisión descritas. En esta experiencia, para el caso concreto de C y C++, se han utilizado dos: *valgrind* [14] y *gcov*.[5]

valgrind. La gestión de memoria dinámica en C, la relación entre punteros y vectores, la simulación de parámetros de entrada y salida con punteros y demás flexibilidades del lenguaje hacen que sea fácil cometer errores al escribir el código. Por desgracia muchos de estos errores son espurios e irrepetibles, por ejemplo: cuando se usa una variable antes de darle un valor, puede que, por casualidad, el valor almacenado sea el que se espera como inicialización, con lo que el programa no muestra ningún síntoma; o cuando se accede fuera de rango en un vector, alcanzando una zona de memoria que ocupan otras variables del programa que no se están usando en ese momento, por lo que tampoco se manifiesta ningún error.

Este tipo de errores se da con más frecuencia de la esperada —la experiencia indica que un programa en C que ha pasado las pruebas de ejecución normales, con mucha frecuencia, contiene vicios ocultos que no se muestran.

valgrind es un intérprete de lenguaje máquina orientado a la detección de este tipo de fallos que, en la experiencia aquí descrita, se emplea como prueba final. Por desgracia, la salida que actualmente produce no es fácil de procesar automáticamente, aunque en un futuro cercano se espera que los informes sean en XML y puedan ser incorporados al informe general con facilidad.

gcov. Es un programa de GNU que se incluye con el compilador gcc y se emplea principalmente para realizar pruebas de cobertura. La prueba de cobertura permite conocer las líneas de código que no se ejecutan. En esta experiencia se emplea para saber si las pruebas del alumno ejercitan todo el código a comprobar. La salida de *gcov* es procesada para mostrar al alumno, en un formato adecuado, las líneas de código que sus pruebas no ejecutan.

4.7. Límites de la revisión automática

Aunque las pruebas enumeradas resultan bastante completas, a nadie se le escapa que existen determinados aspectos que son difícilmente evaluables de forma automática. Se trata fundamentalmente de aspectos cualitativos no cuantificables relacionados, por ejemplo, con cuestiones de estilo: si bien se puede medir el porcentaje y distribución de los comentarios, hoy por hoy resulta imposible garantizar que su contenido sea adecuado; otro ejemplo son los identificadores ¿cómo garantizar que el nombre de una variable es acorde a su función? Tampoco se puede garantizar al detalle que se estén usando los mecanismos más apropiados del lenguaje en cada situación. La valoración de la creatividad tampoco es un elemento fácilmente automatizable.

En este aspecto, pueden aparecer vicios derivados de las exigencias de las pruebas, como por ejemplo, que el alumno introduzca

comentarios sin sentido con el solo fin de garantizar el porcentaje mínimo requerido.

5. Cambio de los roles del profesor y el alumno por la revisión automática

La revisión automática ejerce un impacto visible en los papeles que juegan tanto el profesor como el alumno en el proceso evaluativo-formativo de los trabajos prácticos.

Sin la revisión automática, el profesor tiene dos momentos estelares en el proceso: la especificación del trabajo, a realizar al comienzo del proceso, y la revisión de las prácticas, entregadas al final; de estos dos, el segundo es extraordinariamente más costoso que el primero. Con la revisión automática, el momento inicial del proceso incluye no sólo la especificación, que, como ya se ha dicho, tiene que ser mucho más detallada, sino también el diseño y preparación de las pruebas; por el contrario, frente a este incremento de esfuerzo inicial, en la etapa final de revisión sólo tiene que ocuparse, en su caso, de los pocos aspectos que no son susceptibles de una evaluación automática. Esta revisión última sirve, no sólo para establecer una calificación definitiva —no debe ser costoso, ni divergir mucho de la propuesta automáticamente por el sistema—, sino también para mejorar las pruebas, en función de lo que se observe —fundamentalmente, cuando la calificación definitiva difiera significativamente de la propuesta. El profesor disminuye considerablemente el tiempo dedicado a la tarea repetitiva de evaluación pudiendo dedicar ese tiempo a establecer las pruebas automáticas.

En cuanto al alumno, su papel pasa a ser más activo. Sin revisión automática, hace su trabajo, lo entrega y se desentiende hasta que el profesor le comunica su calificación. Con revisión automática, el alumno recibe una valoración inmediata al entregar su ejercicio, como se muestra en la figura 2, y entra en un proceso de corrección/reentrega hasta que finalizan los plazos de entrega máximos establecidos.

Figura 2. Ejemplo de información mostrada al alumno en el momento de la entrega

6. Experiencia de uso

Como ya se ha mencionado, esta herramienta lleva en funcionamiento desde el curso 2004/2005 en diversas asignaturas del primer y segundo curso de las titulaciones de informática de la Universidad de Las Palmas de Gran Canaria, abarcando a unos 650 alumnos por curso académico. Como ejemplo, se pueden analizar los datos correspondientes a la asignatura "Metodología de la Programación II", de segundo curso del año académico 2006/2007. Para la segunda práctica, en la que participan 209 alumnos con 2.236 revisiones automáticas, la figura 3 muestra el número promedio de entregas que realizan, cada día, los alumnos que hacen alguna entrega ese día. Se observa que, como promedio, los alumnos, el día que están trabajando hacen 2.11 entregas. También se observa que el promedio de entregas se incrementa a medida que se acerca la fecha de finalización del plazo de realización de la práctica; ello es coherente con lo mostrado en la figura 4, que refleja que un porcentaje importante de alumnos tiende a entregar los últimos días, no aprovechando convenientemente la realimentación, aún haciendo más entregas promedio —entran en un ciclo de prueba y error.

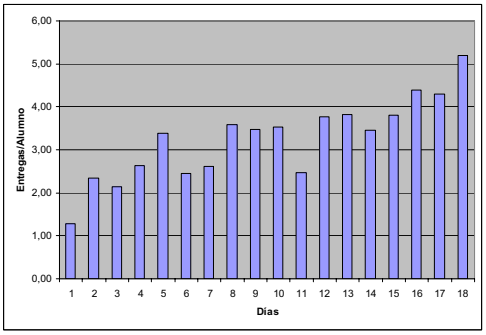


Figura 3. Entregas promedio por alumno que entrega ese día.

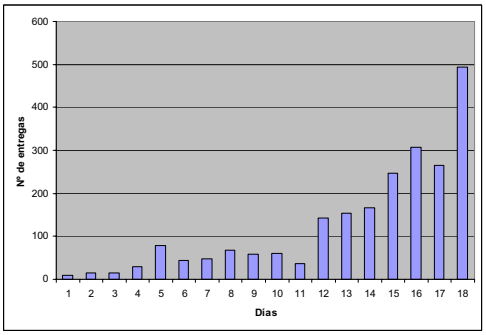


Figura 4. Número de entregas por día.

Por otro lado, la figura 5, refleja la incidencia de la realimentación en los alumnos que superaron la parte de la asignatura correspondiente al dominio del lenguaje en la convocatoria ordinaria de febrero de 2007 —y que no la habían superado por parciales. Es de destacar cómo el 73% de los aprobados usaron la realimentación proporcionada por el sistema y cómo esta realimentación ha sido más beneficiosa para los alumnos noveles que para los repetidores, lo que viene indicado por dos datos: que los alumnos noveles con realimentación son más de la mitad de los aprobados, y que la diferencia entre éstos y los alumnos noveles que aprobaron sin aprovechar la realimentación es muy superior a la que se da entre los repetidores, con y sin realimentación.

Figura 5. Incidencia de la realimentación en los alumnos aprobados en febrero de 2007.

7. Conclusión y perspectivas de futuro

La utilización de una herramienta de revisión automática de prácticas de programación es efectiva en la mejora de la calidad del esfuerzo del profesor al liberarlo de las tareas más tediosas y costosas de evaluación de código informático. Ello le permite dedicar más tiempo al diseño y mejora de las prácticas y sus pruebas. Además, se consigue una evaluación más homogénea y justa.

También es efectiva, sobre el rendimiento del alumno, al proporcionarle una realimentación inmediata de los resultados de su trabajo, que le permite mejorar su proceso formativo. No obstante, habría que conseguir que el alumno aproveche mejor esta herramienta docente a lo largo de todo el periodo de desarrollo de las prácticas, evitando el vicio generalizado de trabajar a última hora. En este sentido, se ha planteado premiar con mejor nota a los alumnos que entreguen con antelación a la finalización del plazo.

Con el tiempo, la herramienta se ha convertido, para los profesores que la usan, en un elemento importante y de difícil sustitución en el contexto de trabajo con grupos numerosos de alumnos.

Actualmente no se dispone de un seguimiento detallado del trabajo del alumno, que permita una mejor evaluación del coste de realización de las prácticas y una mayor autentificación en la autoría de éstas. Como mejora se tiene en perspectiva

incorporar herramientas que permitan hacer este seguimiento.

Referencias

[1] Ala-Mutka, K.; Uimonen, T. and Järvinen, H. *Suppporting Students in C++ Programming Courses with Automatic Program Style Assessment*. Journal of Information Technology Education. Vol. 3, pp 245-262. 2004.

[2] Barberá, E. *Aportaciones de la tecnología a la e-Evaluación*. RED. Revista de Educación a Distancia [en línea], Julio de 2006. en http://www.um.es/ead/red/M6 [consulta 18/01/2007].

[3] Durán, F.; Gutiérrez, F. y Pimentel, E. *El uso de herramientas de apoyo para la valoración de actividades prácticas de programación.* [en línea]. http://www.uca.es/... [consulta 19/02/2007].

[4] Friedl, S. *Go Directly to Jail*. Linux Magazin. December 2002.

[5] *Gcov* en GCC, the GNU Compiler Collection [en línea]. http://gcc.gnu.org/ [consulta 19/02/2007].

[6] Howatt, J. W. *On criteria for grading student programs*. ACM SIGCSE Bulletin, Vol. 26(3). September 1994.

[7] Jackson, D. and Usher, M. *Grading student programs using ASSYST*. ACM SIGCSE Bulletin, Proceedings of the twenty-eighth SIGCSE technical symposium on Computer science education SIGCSE '97, USA. Vol. 29(1), pp 335-339. March 1997.

[8] Joy, M.; Griffiths, N. and Boyatt, R. *The BOSS Online Submission and Assessment System* ACM Journal on Educational Resources in Computing (JERIC) Vol. 5(3), September 2005.

[9] Laakso, M. and Salakoski, T. and Korhonen, A. *The feasibility of automatic assessment and feedback*. Proceedings of Cognition and Exploratory Learning in Digital Age (CELDA 2005), pp 113-122. 2005.

[10] MOODLE. Moodle is a course management system (CMS). http://moodle.org/

[11] Rodríguez del Pino, Juan Carlos. *Gestión Automática de entrega de Prácticas vía web*. Actas de las VIII Jornadas de enseñanza Universitaria de la Informática. Cáceres, pp 53-58. Julio 2002.

[12] Rodríguez del Pino, J. C.; Hernández Figueroa, Z. ; González Domínguez, J. D. ; Díaz Roca, M. Experiencia de uso y características del programa Gestión Automática de Prácticas (GAP), Actas Conferencia IADIS Ibero-Americana, pp 35-42. Octubre 2005.

[13] Saikkonen, R.; Malmi, L. and Korhonen, A. *Fully Automatic Assessment of Programming Exercises*. ACM SIGCSE Bulletin, Proceedings of the 6th annual conference on Innovation and technology in computer science education ITiCSE '01. Vol. 33(3). June 2001.

[14] Valgrind [en línea]. http://www.valgrind.org/ [consulta 19/02/2007].

Predicción del rendimiento de los estudiantes y diagnóstico usando redes neuronales

Zulma Cataldi, Fernando Salgueiro, Fernando Javier Lage

Universidad Tecnológica Nacional. Facultad Regional Buenos Aires. Dirección de Postgrado.
Medrano 951. (C1179AAQ) Cdad. Autónoma de Bs. As.
LIEMA: Laboratorio de Informática Educativa y Medios Audiovisuales. Facultad de Ingeniería. UBA.
Av. Paseo Colón 850, Piso 4º (C1063ACV) Cdad. Autónoma de Bs. As.
liema@fi.uba.ar, fsalgueiro@fi.uba.ar flage@fi.uba.ar

Resumen

Según se expuso en publicaciones previas se ha observado el bajo rendimiento de los estudiantes que comienzan una carrera universitaria en sus evaluaciones parciales y finales. Esta investigación surgió para encontrar una solución a tal problema. Se toman los datos de las evaluaciones parciales y a partir del análisis de los mismos se busca efectuar un diagnóstico con base en los errores cometidos. De este modo, en función de los datos de los primeros exámenes se podrán predecir errores futuros y dar sugerencias para realizar una ejercitación correctiva a fin de mejorar la producción. Para efectuar la predicción de los próximos errores se usa una red neuronal y sobre esta base se le sugiere al estudiante una secuencia de ejercicios y problemas a fin de mejorar su producción y por lo tanto sus calificaciones.

Palabras Clave -- Sistemas Tutores Inteligentes, Redes neuronales, evaluación y predicción.

1. Introducción

En publicaciones previas se ha señalado, el bajo rendimiento de los estudiantes de Algoritmos y Programación I en sus exámenes parciales y finales durante los últimos seis cuatrimestres [1]. Este trabajo de investigación surge con fines prácticos tratando de encontrar una solución a este problema a través de la predicción tomando como datos los resultados en las evaluaciones y su análisis para poder efectuar luego un diagnóstico, a fin de sugerir estrategias de refuerzo, ya que se piensa que un sistema asesor inteligente que podría efectuar la predicción y asistir a los estudiantes en su evolución [2].

En la actualidad no existen sistemas expertos que resuelvan en forma eficiente éste problema, ya que sólo se encuentran disponibles algunos programas para enseñanza de programación, pero de tipo tutorial [3]-[4].

2. Justificación del tema

Un sistema así planteado resolvería el problema de la predicción del comportamiento de los alumnos en el marco universitario. Es decir; a partir de los primeros exámenes se podrían predecir posibles errores futuros a través de diagnóstico, pudiendo sugerir la ejercitación correctiva a fin de mejorar su rendimiento y encauzar su aprendizaje hacia conceptualizaciones incorporadas de forma más significativa y permanente [5].

Se toman los errores cometidos en cada uno de los exámenes como datos de partida, se cargan en una base de datos y permiten efectuar una primera categorización en grandes grupos.

A partir de estos datos el sistema que se propone deberá ser capaz de predecir las próximas fallas de un alumno y en consecuencia de ello deberá determinar la siguiente secuencia de problemas y/o ejercicios que el estudiante debería realizar para mejorar su rendimiento y por ende sus calificaciones [6].

El problema del diagnóstico y la predicción incluye:

- una primera etapa de diagnóstico, que está determinada por las respuestas dadas por los alumnos en sus exámenes parciales y finales,
- una segunda parte basada en la predicción de los errores futuros y en la determinación de la ejercitación que deberá resolver un alumno a fin de mejorar su rendimiento.

Este problema está relacionado directamente con los contenidos de la asignatura y además debe considerar, desde el punto de vista didáctico, las acciones del tutor y del estudiante que conforman el triángulo didáctico, en tanto es un proceso comunicacional [4].

Un sistema asesor de este tipo debe plantearse a través de la aplicación de los sistemas inteligentes [7]-[8]-[9]. En general estos sistemas suelen adoptar la forma de tutoriales en los que el estudiante puede tomar la iniciativa residiendo las diferencias más notables con respecto a los programas tutoriales convencionales en su diseño. Un programa tutorial tradicional trata de inducir en el estudiante la respuesta correcta mediante una serie de estímulos que han sido cuidadosamente planificados. En cambio un programa de este tipo, intenta predecir comportamientos futuros sugiriendo líneas de acción, es decir, debe tomar alguna decisión pedagógica [9]-[10].

Existe un *modelo de dominio* donde se encuentra el conocimiento sobre el dominio que deberá ser recomendado, que se puede dividir en: declarativo (los primeros principios, la comprensión del dominio) y procedural (el conocimiento que es utilizado para realizar una tarea) [11]. Utiliza los mismos métodos que se usan en la construcción de la base del conocimiento de los sistemas basados en el conocimiento tales como: las reglas de producción, las redes semánticas, los *frames*, etc. Este conocimiento del dominio consiste en los hechos y en las relaciones entre los hechos que, generalmente debe ser fortalecido en general por uno o más especialistas [12]-[13].

Este módulo, tiene algunas funciones básicas, ya que sirve como fuente de conocimiento a ser presentado al estudiante (lo que incluye la generación del material, generación de preguntas y respuestas, entre otras cosas) y forma un patrón que permitirá evaluar el conocimiento del estudiante. Para eso, el sistema debe ser capaz de generar soluciones a los problemas en el mismo contexto de un estudiante, para que sus respectivas respuestas puedan ser evaluadas. La base del conocimiento del dominio es un componente clave en el *sistema predictor,* ya que es ahí donde está representado el material de enseñanza [14]-[15].

En los casos en que el dominio sea de naturaleza descriptiva y teórica (como en geografía ó física), la representación utilizada es la declarativa (a través de redes semánticas ó *"frames"*). En los casos en que el dominio esté orientado a una tarea (por ejemplo: la programación) la representación tiende a ser procedural (ya que son típicamente reglas de producción) [16]. El modelo del dominio es un tema que ha sido estudiado, así como el modo en que el sistema lo puede usar para razonar [15]-[16]. Algunas de las representaciones posibles son las redes semánticas, reglas de producción y "*constraints*" [15]-[16]. La elección depende en parte de como se lo usará y es común a todos los usuarios del sistema.

A fin de dar solución a la problemática planteada de predicción y diagnóstico, se indagará sobre la aplicación de sistemas inteligentes tales como las redes neuronales que han dado buenos resultados en diversas áreas [17].

3. Las redes neuronales

Las redes neuronales (RN) son conjunto de elementos más simples que se interconectan en paralelo en forma jerárquica y que interactúan como los sistemas neuronales psicológicos [18]. A fin de poder utilizarlas para representar sistemas de mayor complejidad pueden tener retroalimentación. Una de sus características diferenciales es que pueden aprender de la experiencia a través de la generalización de casos [19].

Una red neuronal se caracteriza por cuatro elementos básicos: *su topología, el mecanismo de aprendizaje, tipo de asociación realizada entre la información de entrada y salida y la forma de representación de estas informaciones.*

Las neuronas se distribuyen en la red formando capas de un número determinado de elementos básico. Es decir, existe una capa de: *entrada* que recibe directamente la información proveniente de las fuentes externas de la red, capas *ocultas* que son internas a la red y no tienen contacto directo con el exterior (desde cero niveles hasta un número elevado), pudiendo estar interconectadas de distintas maneras, lo que determina junto a su número, las distintas topologías y una capa de *salida* que transfiere la información de la red hacia el exterior.

La topología de las redes neuronales es la forma de organización de las neuronas en la red formando capas o agrupaciones de neuronas más ó menos alejadas de la entrada y la salida de la red. Por lo tanto, los parámetros fundamentales de la red serán: el número de capas, el número de neuronas por capa, el grado de conectividad y el tipo de conexiones ente neuronas.

A1. El algoritmo backpropagation

Rumelhart, Hinton y Williams [20] desarrollaron un método de aprendizaje automático que permitió que una red neuronal basada en el

perceptron [21] aprendiera la asociación existente entre los patrones de entrada y las clases correspondientes de salidas. Como se buscaban formas de *"Aprendizaje Automático"* o "*Machine Learning*" (se puede definir como un conjunto de programas computacionales que mejoran con la experiencia), estos sistemas deben ser capaces de adquirir conocimientos de alto nivel para la resolución de problemas mediante ejemplos provistos por un instructor ó supervisor debiendo generar representaciones internas de los conceptos. Para lograr esto se modificó la red del *perceptron* de Rosenblatt [21] agregándole capas ocultas, con conexión hacia adelante y sin conexiones recurrentes [20]. Pero, no fue suficiente con introducir algunas modificaciones topológicas a la red, sino que se requerían modificaciones en el algoritmo de aprendizaje; por lo tanto fue desarrollado el método de aprendizaje no supervisado denominado también backpropagation, basado en la regla *delta generalizada* [20], logrando así, una ampliación del rango de aplicación de las redes neuronales.

El funcionamiento general de una red neuronal artificial del tipo *backpropagation*, así como el de otras redes neuronales, se puede dividir en dos partes: una etapa de entrenamiento y una etapa de puesta en marcha. La primera consiste en el aprendizaje de un conjunto predefinido de observaciones de entrada–salida dados como ejemplo (utilizando *n* atributos de entrada y un único atributo ó clase, de salida), empleando un ciclo propagación-adaptación de dos fases [17]-[22], donde:

En la *primera fase* se aplican los atributos de entrada a la capa de entrada de datos a la red y los valores generados se propagan desde esta capa hacia las superiores hasta generar una salida, en la capa de salida de la red. Para realizar el entrenamiento, se compara el resultado obtenido en cada neurona de salida con el valor deseado para cada neurona en particular y obteniéndose un error para cada una de las unidades de salida.

En la *segunda fase*, los errores de las unidades de salida se transmiten hacia atrás, pasando por todas las neuronas de las capas intermedias que contribuyan directamente a la salida, recibiendo el porcentaje de error aproximado a la participación de las neuronas intermedias en la salida original. Este proceso se repite capa por capa hasta llegar a la capa de entrada y hasta que cada neurona haya recibido un error que describa su aporte al error total. Debido a ello el algoritmo se denomina también de *retro-propagación* o propagación hacia atrás, donde los errores se calculan con respecto a los aportes de las neuronas desde la capa de salida hasta la capa de entrada y es con respecto al valor del error recibido que se reajustan los pesos de las conexiones entre cada par de neuronas en la red, de manera de que el error total cometido para ese patrón disminuya.

Dado que la fase de funcionamiento es similar a otras redes neuronales artificiales se debe realizar un análisis más profundo del método de aprendizaje. El método de *backpropagation* utiliza una función o superficie de error asociada a la red, buscando el estado de mínimo error estable a través del camino descendente de la superficie de error [20]. Es por esto que se debe realizar la retroalimentación para realizar las modificaciones en los pesos iniciales en un valor proporcional al gradiente decreciente de dicha función de error. En la Figura 1 se puede ver un esquema de este tipo de redes neuronales artificiales.

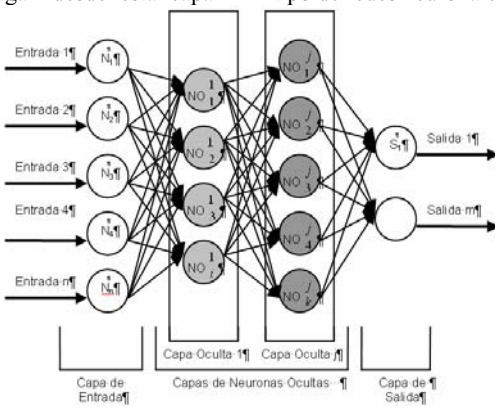


Fig.1: Modelo backpropagation de red neuronal artificial Es una red multicapa, con conexiones hacia adelante y sin conexiones recurrentes [20].

1) La etapa de funcionamiento

En esta etapa se ingresa un patrón p de entrada X_p: $x_{p1},...,x_{pi},...,x_{pN}$, que se transmite a través de los pesos w_{ji} desde la capa de entrada hacia la capa oculta. Las neuronas de esta capa intermedia transforman las señales por aplicación de una *función de activación* un valor de salida que se transmite a través de los pesos v_{kj} hacia la capa de salida. Repitiendo la misma operación que en el caso anterior, las neuronas de la última capa dan la salida de la red.

La función de activación utilizada en cada una de las neuronas debe ser derivable de primer orden.

Algunos autores como del Brio y Sanz Molina, [17] sostienen que el mejor procedimiento para entrenar una red neuronal es el *cross validation,* es decir entrenar y validar a la vez, usando el ochenta por ciento de los patrones para entrenar y el veinte restante como conjunto de pruebas.

2). La etapa de aprendizaje

En la etapa de aprendizaje, se busca minimizar el error entre la salida obtenida por la red y la salida deseada luego del entrenamiento con el conjunto de datos patrones. Es por ello, que en las redes *backpropagation* el aprendizaje es de tipo supervisado, ya que es el usuario (o supervisor) quien determina cuál es la salida deseada ante la presentación de un patrón de entrada dado

Para modificar los pesos se sigue la fundamentación matemática del algoritmo *backpropagation* basado la técnica del *gradiente decreciente* [20]. Suponiendo una red formada por dos pesos, se puede visualizar como un espacio de dos dimensiones y como el error cometido es función de los pesos de la red; en este caso, para cualquier combinación de valores de los dos pesos, le corresponderá un valor de error para el conjunto de entrenamiento donde estos valores de error se pueden visualizar a través de una superficie del error que puede diferentes topografías (especie de sábana). El proceso de entrenamiento comienza, en este caso un punto de la sábana, representado por los pesos iniciales de la red y el algoritmo de aprendizaje se centra en obtener la información local de la pendiente de la superficie o sea el gradiente. A partir de esa información se pueden ir modificando los pesos en forma iterativa en forma proporcional a dicha pendiente, a fin de asegurar un descenso a través de la superficie del error hasta alcanzar el mínimo más cercano al punto de partida. Cuanto mayor sea el número de pesos considerados el espacio se va convirtiendo en un plano multidimensional en el que aplicarán los principios mencionados.

El error o valor delta asociado a una neurona oculta j está determinado por la suma de los errores que se cometen en las k neuronas de salida que reciben como entrada la salida de esa neurona oculta j, por *propagación del error hacia atrás*.

3). El sobreajuste

Para obtener una aproximación funcional óptima se deben elegir cuidadosamente las variables a emplear, es decir se trata de incluir en el modelo las variables que realmente predigan la variable dependiente o de salida, pero que no covaríen entre sí [23], debido a que podrían provocar un sobreajuste (*overfitting*) innecesario. Esto sucede cuando el número de parámetros o de pesos de la red resulta excesivo en relación al problema a tratar y al número de patrones de entrenamiento disponibles. El sobreajuste disminuye la capacidad de la red de proporcionar una respuesta correcta ante patrones que no han sido empleados en su entrenamiento. Se entiende por generalización de la red a la capacidad de dar una respuesta correcta ante patrones que no han sido empleados n su entrenamiento [17].

Del Brio y Sanz Molina [17] recomiendan dos formas de bajarlo: un parada temprana (usando *cross validation*) o limitando el tamaño de la arquitectura de la red. La variable cuya eliminación causa el menor decremento en la ejecución de la red es eliminada. Este procedimiento se debe repetir sucesivamente hasta que la eliminación de más variables involucra una disminución sensible en la ejecución del modelo [22].

4. Objetivos del Trabajo

El objetivo general se desglosó en tres objetivos específicos:

- Implementar sistemas inteligentes (redes neuronales) para predecir errores en futuros exámenes y a partir de éstos determinar que clase de problemas o ejercicios debe resolver el alumno para mejorar su rendimiento, en un dado dominio.

– Determinar el tipo de sistema inteligente a utilizar, ya sea una red neuronal o un algoritmo genético que permita determinar en forma más eficiente la asesoría luego del diagnostico realizado.
– Evaluar el sistema utilizando datos estadísticos almacenados y con un grupo de docentes y de estudiantes disponibles.

5. Metodología

1) Se determinó la herramienta inteligente más apropiada para resolver la cuestión de la predicción en el comportamiento del alumno y la determinación de la actividad futura que deberá realizar, entre las disponibles en un dado dominio.

2) Se creó una base de datos con todos los ejercicios tomados en las evaluaciones parciales y finales de una Cátedra de Algoritmos y Programación I, indicando su tipo (ya sea práctico o teórico), su dificultad, su obligatoriedad o no en un examen, su puntaje y su objetivo (es decir a qué se apunta en dicho ejercicio). El tamaño de la muestra fue de 450 alumnos, usándose 2/3 para entrenamiento y 1/3 para pruebas.

3) Se elaboraron las bases para la creación de un sistema predictor que cumpla con los objetivos que se han propuesto en este trabajo. Para este punto se determinarán las herramientas metodológicas más apropiadas que provee la ingeniería de software y los sistemas inteligentes en cuanto a métodos, técnicas y herramientas disponibles.

4) Se realizó una prueba piloto (evaluación interna y externa del software) con un grupo estudiantes y docentes de la cátedra disponibles.

6. Parte Experimental

Las etapas seguidas fueron las que se describen en la Tabla 1.

Se utilizó la herramienta *NNclass* [25]. La misma posee un instructivo que permite efectuar el ingreso de los datos. El valor máximo es de 50 variables de entrada y 40 de ellas pueden de tipo categóricas.

.

Please make sure that there are no more than 50 neurons in Input Layer.
There should be exactly 1 Output variable - application will treat it as Categorical.
There should be no more than 40 Categorical Variables.

Omit	Cat	Omit	Omit	Cat	Cat	Cat	Cat	Cat	Cat	Cat	Cat	Cat	Cat	Cat	Cat	Cat	Cat	Cat	Cat	Cat
Errores	**Sig**	**Carrera**	**Padron**	**Sexo**	**E1**	**E2**	**E3**	**E4**	**E5**	**E6**	**E7**	**E8**	**E9**	**E10**	**E11**	**E12**	**E13**	**E14**	**E15**	**E16**	**E17**
		9	84585	M	0	1	0	1	0	0	1	0	1	0	0	0	0	0	0	1	0
		10	84722	M	0	1	0	0	0	0	0	1	1	1	0	0	0	0	0	1	0
		10	84653	M	0	1	1	1	0	0	0	0	0	0	1	1	0	0	0	1	0
		9	79600	M	1	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	1
		10	84078	M	0	0	1	0	0	0	1	0	1	1	1	0	0	0	0	1	0
		10	84134	M	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
		10	84870	M	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
		10	84620	M	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		10	85032	F	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
		10	85413	M	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
		10	84962	F	0	0	0	1	0	0	1	1	0	0	1	0	0	0	0	0	0
		9	84926	M	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
		9	84734	M	1	1	1	1	0	1	1	0	0	0	0	1	0	0	0	0	0
		10	84091	M	0	1	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0
		10	82557	M	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0
		10	84690	M	0	0	1	1	0	0	0	0	1	1	0	1	0	0	0	0	1
		10	84087	M	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1
		10	84587	M	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1
		10	84371	M	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	1

Fig. 2: Pantalla ingreso de datos codificados.

A. El ingreso de los datos.

Existen tres tipos de datos que se ingresaron en forma codificada:

– *Categóricos*: son las palabras, que no son la salida esperada de la red.

Se observa que

– *Continuos*: son datos numéricos que representan números reales.

De *Salida (Output):* son los valores que se desea que la red prediga, los que a su vez pueden ser categóricos ó continuos.

Cat E16	Cat E17	Cat E18	Cat E19	Cat E20	Cat E21	Predicted Out	Score (aprobado)	Score (desaprobado)
0	1	0	0	10	0	aprobado	0,970997572	0,027049135
0	0	10	0	0	1	desaprobado	0,001156448	0,998752892
0	0	0	0	1	0	desaprobado	0,005972461	0,994004071
100	0	0	0	11	1	aprobado	0,979461491	0,019286392
0	0	10	0	101	101	aprobado	0,999360442	0,000711223
1	0	1	0	1	1	aprobado	0,944856524	0,058509205
1	0	0	0	1	0	aprobado	0,999345362	0,000726172
0	0	1	0	0	0	aprobado	0,544088364	0,455251902
1	11	1	0	1	0	aprobado	0,983933151	0,01474512
0	0	0	0	1	1	desaprobado	0,006141305	0,993789971
0	0	0	0	0	1	aprobado	0,680459857	0,331099778
0	0	0	0	0	0	aprobado	0,779129326	0,220500425
1	0	0	0	11	11	aprobado	0,999358594	0,000712385
1	0	0	0	1	1	aprobado	0,987725079	0,011409681
0	0	0	0	0	0	aprobado	0,994639397	0,005599276
0	0	0	0	1	1	aprobado	0,945939481	0,057353366
0	0	0	0	1	1	desaprobado	0,171479136	0,828849435
0	0	0	0	0	1	desaprobado	0,000927733	0,999024689
0	0	0	0	1	1	aprobado	0,954882205	0,047749929
0	0	0	0	10	1	aprobado	0,99622333	0,003323361
0	0	1	0	0	1	desaprobado	0,030887676	0,968400836

Fig. 3: Pantalla predicción evaluación final

Paso	Entrada	Acción	Salida
1	Datos de los errores de los alumnos de la materia en exámenes	Confección de base de datos de errores de cometidos	Errores codificados
2	Datos de entrenamiento	Entrenamiento de la red	Parámetros con error mínimo. Red entrenada
3	Datos de prueba	Aplicación de la red	Pronóstico
4	Errores codificados	Aplicación de la red	Diagnóstico

TABLA 1 ETAPAS Y ACCIONES SEGUIDAS

En el caso que se muestra en la Figura 2, se muestra la salida que es de tipo categórica ya que será: *aprobado* ó *desaprobado*.

Predicted Output	Score (aprobado)	Score (desaprobado)
Aprobado	0,970997572	0,027049135
Desaprobado	0,001156448	0,998752892

TABLA 2 EJEMPLO DE PREDICCIÓN APROBADO–DESAPROBADO

Por lo tanto, la red utilizará dos neuronas para ello; una cuya salida, a través de un número real, representa a los *aprobados* y otra, que también devuelve un número real, cuya salida está asociada a los *desaprobados.* En la primera fila del ejemplo de la Tabla 2 se observa que la para la salida de 0.970 la neurona está asociada a *"aprobado"* y sólo 0.027 la asociada a *"desaprobado",* por lo que la red da como resultado el primero de éstos. En la segunda fila se presenta el caso inverso. Los datos de las evaluaciones de los estudiantes (a través de 6 instancias de aprobación: un parcial con dos recuperatorios y tres oportunidades para el examen final) han sido codificados como se observa en la Tabla 3. El resto son combinaciones de las opciones presentadas.

000	no se equivocó nunca
001	se equivocó en el parcial
010	se equivocó en el primer recuperatorio
100	se equivocó en el segundo recuperatorio

TABLA 3: CODIFICACIÓN DE LAS INSTANCIAS DE APROBACIÓN.

Con respecto a las columnas, E1 a E21 de la Figura 1, son los tipos de errores detectados en cada una de las evaluaciones agrupados luego del análisis realizado.

B. La selección de los parámetros de entrenamiento de la red.

Se recomienda utilizar valores similares a los que están en la Figura 4 repitiendo el proceso cambiando los valores hasta que se encuentre un error mínimo entre 0%-5%.

7. Resultados

En la Figura 4 se observan las características de la red. Cuando el error es inaceptable, mayor al 5%, se debe descartar la red y comenzar el proceso de entrenamiento nuevamente variando los distintos parámetros que la definen (entre los que se pueden citar el parámetro de aprendizaje α (el valor inicial y el régimen de modificación a lo largo de los ciclos), el momentum β, los pesos aleatorios iniciales, el vecindario gaussiano, la cantidad

de atributos que se utilizarán para el entrenamiento, la cantidad de las observaciones que se utilizarán y las que se descartarán, justificando por qué se descartarán.

En promedio, el tiempo de entrenamiento de una red neuronal para los valores utilizados es de aproximadamente 1.5 horas. Se entrenaron redes con más de 600 variaciones en los parámetros hasta encontrar los adecuados y los valores mínimos correspondientes al error, lo que da un tiempo total neto de 38 días de entrenamiento.

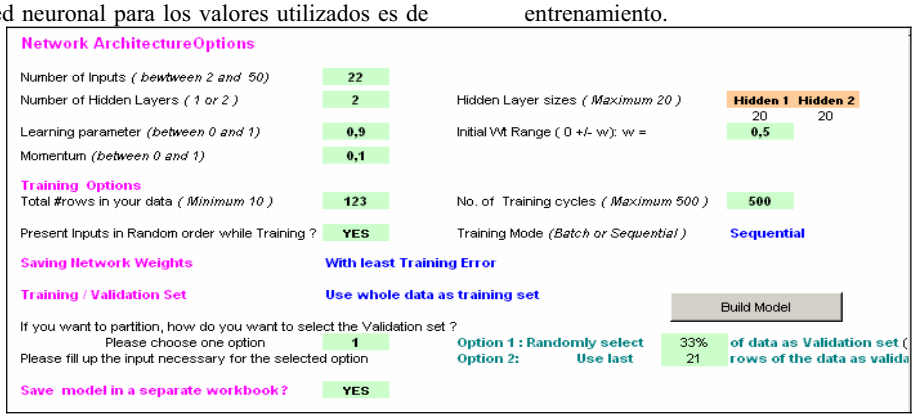


Fig. 4: Características de la red

A. La predicción de la red

Cuando un estudiante se equivoca (o no) en algunos temas, aunque no haya completado aún todo el curso, es decir, si solo rindió el parcial (y no los recuperatorios, por ejemplo) se puede predecir si aprobará o no. La idea es utilizar esta red como primer paso para el uso de una serie de dos redes. Cuando la red prediga que no aprobará, una segunda red, basada en los errores cometidos, le puede indicar qué temas debe estudiar, a fin de recomendarle los ejercicios por núcleos temáticos. Para este primer entrenamiento predictivo la red operó con un error del 4%, lo que es más que aceptable para el trabajo con grupos humanos [9].

8. Grado de Avance

Hasta ahora se trabajó en el sistema de predicción donde se buscó indagar cómo puede diagnosticar la red a partir del rendimiento obtenido en las evaluaciones parciales cómo será la *"performance"* en el final. Este dato permitirá diseñar un sistema recomendador de modo que el alumno en situación de preparar su evaluación final pueda acceder al sistema que le sugerirá una serie de ejercicios y problemas para poder internalizar los errores clave cometidos en las evaluaciones previas.

9. Conclusiones y Trabajos Futuros

Se prevé trabajar en el diseño del sistema recomendador de ejercicios y problemas de modo que el estudiante pueda adquirir una cierta autonomía en la preparación de sus exámenes finales. Se busca brindar a los estudiantes una herramienta a fin de que puedan tomar conciencia de sus propios errores para no cometer las mismas fallas en las evaluaciones finales. Esta forma de *autoevaluación* resulta un acercamiento, hacia la autonomía del alumno y la mejora del proceso de aprendizaje.

Como trabajos posteriores se propone: a) Ampliar los contenidos disponibles para la autoevaluación, b) Escalar el sistema informático de tal forma que permita realizar un seguimiento del alumno, de esta forma el docente puede tener una clusterización de su clase en cuanto a necesidades cognitivas, c) Escalar el sistema informático de tal forma que evolucione hacia bases de datos e interfaces capaces de interactuar con el alumno de manera autónoma y d) incluir un módulo de autoevaluación en los Sistemas Tutores Inteligentes cuya arquitectura se está desarrollando.

Agradecimientos

Esta comunicación forma parte de los proyectos de investigación: *Sistemas inteligentes aplicados a la predicción del comportamiento de los estudiantes y diagnóstico* 2005-2006 LIE-DC/04-07 del Laboratorio de Informática Educativa y Medios Audiovisuales (LIEMA) de la Facultad de Ingeniería, de la Universidad de Buenos Aires y C099 *Modelado del tutor basado en redes*

neuronales para un Sistema Tutor Inteligente, de la Facultad Regional Buenos Aires de la Universidad Tecnológica Nacional 2007-2008. Los autores agradecen a los alumnos que participaron de la experiencia.

Referencias

[1].Lage, F.; Cataldi, Z. y Denazis, J. M. (2000). *The Scripts of University Students and Experts in the Preparation of the Examinations: A Study in Process.* FIE 2000: 30th ASEE/IEEE Frontiers in Education Conference, Kansas City Missouri, 18-21 de octubre. Paper 1154. Proceedings en CD-ROM. ISBN 0-7803-6242/0/

[2].Copello, G.; Cataldi, Z. y Lage, F. (1999). *La comprensión de los errores.* Proceedings del V Congreso Internacional de Ingeniería Informática. Páginas 210-217. Editado por Departamento de Publicaciones de la Facultad de Ingeniería.

[3].Pozo, J. I. (1998). *Teorías cognitivas del aprendizaje*. Morata.

[4].Pozo, J. I. (1999). *Aprendices y Maestros*. Alianza.

[5].Ausubel, D.; Novak, J. y Hanessian, H. (1983) *Psicología educativa: un punto de vista cognitivo.* 2ª Ed. México: Trillas. 624p.

[6].Ohlsson, S. (1996) *Learning from performance of errors.* Psychological Review 3 (2) p. 241-262.

[7].Khuwaja, R.A. (1994) A Model of Tutoring: Facilitating Knowledge Integration Using Multiple Models of the Domain. *Ph.D., Illinois Institute of Technology*

[8].Giraffa, L.M.M.; Nunes, M. A.; Viccari, R.M. (1997) *Multi-Ecological: an Learning Environment using Multi-Agent architecture*. MASTA'97: Multi-Agent System: Theory and Applications. Proceedings. Coimbra: DE-Universidade de Coimbra.

[9]. Cataldi, Z. 2005. *Sistemas tutores inteligentes: los estilos del estudiante para selección del tutorizado.* WICC 2005. 13 y 14 de mayo. Universidad Nacional de Río Cuarto. Córdoba. RED UNCI

[10]. Salgueiro, F. A, Costa, G., Cataldi, Z., García Martinez, R. y Lage, F. J. 2005. *Sistemas inteligentes para el modelado del tutor.* GCETE'2005**,** Global Congress on Engineering and Technology Education. marzo 13-15

[11].Abbas, H. (1998) Designing a New Domain Knowledge Base for an Intelligent Tutoring System. *Ph.D., Illinois Institute of Technology.*

[12].Brachman, R.J. (1988) *The basis of knowledge representation and reasoning.* AT&T. Technical Jurnal. 67, 1:15.

[13].Brachman; R.J. (1985) On the epistemological status of semantic networks. En Brachman, R. y Levesque, H. (Eds.) *Readings in knowledge representation* (191-215). Los altos. Morgan kaufman Pub. Inc.

[14].Viccari, R. M. (1993). Inteligência Artificial e Educação: Indagações Básicas. IV Simpósio Brasileiro de Informática e Educação.

[15].Viccari, R.M. y Girafa, L.M. (1996). Sistemas Tutores Inteligentes: Abordagem Tradicional x Abordagem de Agentes. XIII Simpósio Brasileiro de Inteligência Artificial, Curitiba.

[16].Casas, M. (1999) *contribuições para a modelagem de um ambiente inteligente de educação baseado em realidade virtual.* Tesis Doctoral Universida de Federal de Santa Catarina. Programa de Pós-graduação em Engenharia de Produção.

[17].del Brio, B. M. y Sanz Molina, A. (2001) *Redes neuronales y sistemas difusos.* Paraninfo.

[18].Kohonen, T. (1988) Self-Organizing Maps Springer Series in Information Sciences, Vol. 30, Springer, Berlin, Heidelberg, NY(pp 236)

[19].Hilera González; R. y Martínez Hernando, A. (2000) Redes Neuronales Artificiales: Fundamentos, modelos y aplicaciones. Ra-ma, Madrid.

[20].Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. (1986). *Learning internal representations by back-propagating errors in Parallel Distributed Processing*: *Explorations in the Microstructure of Cognition*. Eds. Cambridge, MA: MIT Press, vol. 1, p. 318-362.

[21].Rosenblatt, F. (1958), *The perceptron: A probabilistic model for information storage and organization in the brain*. Psychological Review, 65, 386-408.

[22].Palmer, A., Montaño, J.J. y Jiménez, R. (2001) *Tutorial sobre Redes Neuronales Artificiales: El Perceptrón Multicapa.* Revista Electrónica de Psicología Vol. 5, No. 2, Julio ISSN 1137-8492.

[23].Smith, M. (1993). *Neural networks for statistical modeling.* New York: Van Nostrand Reinhold.

[24].Masters, T. (1993). *Practical neural networks recipes in C++*. London: Academic Press

[25].Saha, A. (1998) *Application of Ridge Regression for Improved Estimation of Parameters in Compartmental Models;* Tesis Doctoral. Departamento de Estadística;

Formación para la profesión

Una e-revista donde publiquen los estudiantes de Informática

José M. Claver
Depto. de Informática
Universitat de Valencia
46100 Burjassot
e-mail: Jose.Claver@uv.es

Germán Fabregat
Depto. de Ing. y Ciencia de los Computadores
Universitat Jaume I
12071 Castellón
e-mail: fabregat@uji.es

Resumen

La actividad académica de los estudiantes suele centrarse en los conocimientos y técnicas impartidas en las asignaturas del curso en que están matriculados, dejando de lado, en muchos casos, actividades y destrezas que serán muy necesarias en el desarrollo de su posterior actividad profesional como: la redacción de informes, la presentación de trabajos, las búsquedas bibliográficas, etc., y cuando éstas intentan inculcarse en los proyectos fin de carrera o en las estancias en prácticas, en los últimos cursos, ya es demasiado tarde.

Los estudiantes también poseen, en general, una falta de perspectiva respecto de los campos de estudio que les esperan en cursos posteriores, lo que dificulta elegir las asignaturas adecuadas para completar su formación. Al mismo tiempo, carecen de una referencia real respecto de la calidad esperada en los trabajos a desarrollar en las asignaturas en las que se van a matricular, o ya se han matriculado.

La propuesta de una e-revista donde los estudiantes publiquen los trabajos de calidad, realizados en asignaturas de las titulaciones de informática pretende convertirse en una herramienta que palie, en parte, algunas de estas deficiencias. En este trabajo se presentan los aspectos más destacados de la puesta en marcha de una e- revista donde publiquen los estudiantes de las titulaciones de informática de la Universidad Jaume I (UJI).

1. Motivación

La propuesta de una e-revista en la que publiquen los estudiantes de las Titulaciones de Informática (Ingeniería Informática, Ingeniería Técnica en Informática de Gestión e Ingeniería Técnica en Informática de Sistemas) de la UJI pretende conformar una actividad que complemente la actividad académica que estos desarrollan habitualmente. Con ella se quiere potenciar, en los estudiantes, destrezas que son muy valoradas en el desarrollo de la actividad profesional, y a las que a menudo no se les da la importancia que éstas requieren, como son la correcta elaboración y presentación de informes y memorandos.

Estas habilidades se ejercitan sobre todo en la realización de la memoria de los proyectos fin de carrera; pero entonces resulta un poco tarde fijar modelos de actuación y desarrollar habilidades que no han sido fomentadas durante los años anteriores en el trabajo desarrollado en el resto de asignaturas.

La valoración de dichas destrezas y su reconocimiento tendrá como consecuencia una mejor formación de los futuros ingenieros y un mejor desempeño de su profesión. Al mismo tiempo, también es esperable que se alcancen mayores cotas de excelencia en el trabajo desarrollado por los estudiantes, lo que redundará en un mayor prestigio de dichos titulados y de la propia institución.

Esta propuesta también pretende influir en el modo en que se imparte la docencia en las asignaturas de las titulaciones de informática, animando a los profesores a adoptar modelos de aprendizaje más activos para el alumno. En particular, parecen interesantes aquellos basados en problemas o casos de estudio, y colaborativos. El desarrollo de estas actividades dará al estudiante la posibilidad de potenciar su creatividad y la posibilidad de enfrentarse a situaciones más próximas a las que se va a encontrar en su vida profesional. Como es natural,

esto se dará en mayor medida en los cursos más avanzados y/o en asignaturas relacionadas con temática más especializada, cuando el estudiante ya posea el bagaje suficiente como para afrontar problemas más complejos y de forma más autónoma. Pero no tiene por qué estar restringido a éstas.

Por otra parte, el hecho de que se trate de una revista electrónica (e-revista) redundará en una reducción importante de costes y en una máxima posibilidad de difusión de los trabajos realizados por los estudiantes.

La e-revista se complementará con información de interés, tanto académico como profesional, para los estudiantes de las titulaciones de Informática. Así, se pretende incluir información de las estancias en prácticas, salidas profesionales, entrevistas a profesionales de la informática que hayan estudiado en la UJI, eventos y encuentros de interés, temas actuales en el campo de la informática, recientes proyectos fin de carrera desarrollados, líneas de investigación en informática desarrollados por los grupos de la UJI, etc.

A continuación se presentan los objetivos y características generales de la e-revista propuesta. En la sección 2 se comentan en detalle los contenidos de la revista y su organización. En la tercera sección se trata la estructura y tipo de artículos. Algunas de las alternativas evaluadas para la implementación de la e-revista se comentan en la sección 4, y en la sección 5 se presentan diversas opciones respecto del formato electrónico de los artículos. En la sección 6 se comentan otras iniciativas similares y se evalúa el impacto de esta iniciativa. Las conclusiones de este trabajo se comentan en la última sección.

1.2. Objetivos

Los objetivos que se pretenden alcanzar con esta iniciativa, de los que ya se han esbozado algunos en el punto anterior, son los siguientes:

- Potenciar una mejora en la estructura y redacción de los documentos realizados por los alumnos. Especialmente, los relacionados con el trabajo práctico, aplicado o de laboratorio.
- Fijar niveles de calidad en el desarrollo de la actividad de los estudiantes.
- Dar publicidad al trabajo bien hecho de los estudiantes, dándole el reconocimiento público que se merece y sirviendo de ejemplo y guía a otros estudiantes, tanto actuales como futuros.
- Establecer un formato más o menos uniforme en la presentación de los informes o memorias de las prácticas de las asignaturas de estas titulaciones.
- Potenciar la creatividad en el desarrollo de su aprendizaje, dándoles la posibilidad de abordar problemas para los que estén más motivados.
- Potenciar los modelos de aprendizaje activos y los basados en problemas o casos de estudio en las asignaturas de estas titulaciones.
- Incentivar el trabajo en grupo de los estudiantes, tanto en el desarrollo como en la presentación de los resultados.
- Dar a conocer a los alumnos de los primeros cursos los temas que se tratan en cursos más avanzados, ayudándoles a elegir la intensificación o asignaturas en las que estén más interesados.
- Que publicar en la revista constituya un valor añadido en el currículum de los futuros graduados.
- Incrementar el prestigio de los estudiantes de informática, de sus titulaciones y de la propia institución.

1.3. Quiénes publican

La revista está pensada para que publiquen trabajos los estudiantes de las titulaciones de Informática. Se promoverá que estos trabajos estén relacionados o sean el fruto del desarrollo de la actividad de los estudiantes en asignaturas de su carrera. Sin embargo, no se limita la posibilidad de que los estudiantes presenten trabajos de forma libre que sean de especial interés.

1.4. Incentivos

Existen varias posibilidades, no excluyentes, en cuanto a incentivar y premiar a los estudiantes que publiquen en la revista:

- Los alumnos que publiquen en la revista verán reflejado esta circunstancia en la nota de la asignatura. Evidentemente, esto supondrá la implicación y compromiso de los

profesores en el diseño de las prácticas y trabajos de sus asignaturas.

- Proponer un premio anual o semestral al mejor/es artículo/s escrito/s por los alumnos. Este premio puede ser en metálico o en especie, mediante vales de compra. Se puede buscar financiación en la Universidad, a través de las Titulaciones de Informática.
- Otra posibilidad es hacer que esta actividad conste como créditos de libre configuración de forma explícita en su expediente, indicando el tipo y características de la actividad realizada.

1.5. Periodicidad

La periodicidad de la revista debería ser semestral: Una en marzo o abril y otra en julio o septiembre Aunque sea prudente iniciarla con periodicidad anual. Esta periodicidad se refiere a la parte académica o de los trabajos de los alumnos, que quedará almacenada y podrá ser accedida posteriormente, no pudiendo ser modificada. La otra parte de la revista será renovada con mayor periodicidad, en función de las noticias que se vayan generando, de su interés y de las disponibilidades para su mantenimiento.

1.6. Temas

Los artículos enviados para su publicación en la revista estarán relacionados con las temáticas tratadas en las asignaturas de las titulaciones de informática, aunque se tendrán en cuenta aquellos artículos que por su calidad e interés en el ámbito de la informática, tanto académico como profesional, merezcan ser incluidos en la revista. En este caso se definirá un apartado especial para este tipo de artículos.

1.7. Estructura de la revista

Cada número de la revista tendrá una sección regular donde se incluirán los artículos seleccionados de entre los enviados por los estudiantes. También se pretende incluir artículos de los directores de las titulaciones de informática, artículos de profesores de relevancia en algún campo de la informática, presentaciones de líneas de investigación de algún grupo de informática, estancias en prácticas, salidas profesionales, entrevistas de graduados de la UJI, temas de interés, tanto académico como profesional o de mercado, relacionados con la informática, proyectos fin de carrera, foros sobre temas de informática, tutoriales, etc. El equipo editorial decidirá qué contenidos de la revista formarán cada número.

1.8. Formato de los artículos

Las contribuciones de los estudiantes seguirán un formato que se establecerá en su momento y que podrá variar según el tipo de artículo. Éste será similar al que utilicen los estudiantes de forma habitual en sus prácticas y trabajos. Para ello los profesores deberán asumir este formato, o parecido, como el utilizado para la presentación de las memorias o informes en su asignatura.

El formato de los artículos de los estudiantes se definirá durante el primer semestre del curso anterior en el que se lance la e-revista. Para ello es necesario un proyecto de coordinación docente que plantee la definición de dicho formato y el compromiso de su utilización por parte de un número suficiente de profesores.

Se puede plantear un formato general abierto, pero parece mejor proponer varias estructuras u organizaciones que deberían ser seguidas en los artículos. Estas estructuras se ajustarían a los diferentes tipos de trabajos con los que estén relacionados: revisiones, estudios, o proyectos (que incluirán diseños hardware y/o software, y su implementación). La longitud de los artículos será un tema a determinar, pero el máximo podría estar entre 15 y 20 páginas.

2. Contenidos de la revista

2.1. Página Principal

La dirección web de la Revista de los Estudiantes de Informática (REI) será clara y sencilla (como por ejemplo: http://rei.uji.es) y se ubicará en un servidor accesible externamente

Las siguientes secciones serían accesibles desde una barra de menú superior e inferior:

- Inicio.
- Número Actual.
- Números Anteriores.
- Entrar.

- Foros/Opinión.
- Administración.
- Enlaces.
- Envío de Trabajos.
- Búsqueda.

En el cuerpo de la página se incluirá un mensaje de presentación y el último número de la revista, pudiendo incluir una llamada a la participación.

2.2. Número Actual

Esta sección contendrá el último número publicado de la revista con los siguientes apartados en el cuerpo:

Fijos:

- Presentación equipo editorial.
- <Número_imprimible.pdf>.
- Artículo/s:

Variables:

- Lección Maestra.
- Mensaje de las direcciones de titulación.
- Grupos de investigación.
- Noticias de actualidad.
- Entrevistas.
- Proyectos Fin de Carrera.
- Tutoriales.

2.3. Números Anteriores

Incluirá una lista de enlaces a los números anteriores con formato igual al número actual y la posibilidad de composición de monográficos. Constará de una página dinámica que permita buscar y componer monográficos a partir de un patrón de búsqueda.

2.5. Grupos de Investigación

Esta sección contendrá presentaciones de líneas de trabajo de los grupos de investigación de los departamentos de informática.

2.4. Clases Magistrales

En esta sección aparecerán artículos de profesores relevantes en algún campo de la Informática.

2.6. Enlaces

Esta entrada del menú principal llevará a una página donde tendremos enlaces internos y externos a la Universidad que sean de interés para los estudiantes:

<u>*Enlaces Internos*</u>

- *Noticias de las Titulaciones de Informática*

Mensajes de Directores de las Titulaciones de Informática referentes a diversos aspectos que sean interesantes para los alumnos: planes de estudios, nuevas normativas, nuevos procedimientos, etc.

- *Estancias en Prácticas*

Enlace a la página de estancias en prácticas, noticias sobre fechas y empresas para la estancia en prácticas de las Diversas Titulaciones:

- *Proyectos Fin de Carrera*

Lista de Proyectos Fin de Carrera, de las diferentes titulaciones de Informática:

<u>*Enlaces Externos*</u>

- *Noticias de Interés*

Se recopilarán noticias académicas, profesionales y de mercado, nacionales e internacionales, relacionadas con la informática.

2.7. Entrevistas

Esta sección incluirá entrevistas a graduados de la UJI, profesores, empresarios, políticos, etc., relacionados con la informática y sus salidas profesionales.

2.8. Foros

En esta sección se propondrá y permitirá la creación de foros sobre temas de informática: asignaturas, lenguajes de programación, dispositivos, ordenadores, sistemas operativos, etc.

2.9. Tutoriales

En esta sección se tratará de recopilar tutoriales, o enlaces a los mismos, que sean de interés para los estudiantes de informática:

- Título, Autores, fecha, <Tutorial.html>, (<Tutorial.pdf>)

2.10. Administración

La sección de administración permitirá: introducir las noticias, gestionar los contenidos, distribuir los artículos, realizar las revisiones de artículos, componer los números de la revista, componer monográficos, etc.

2.11. Envío de trabajos

Esta sección será una página dinámica para enviar los trabajos por parte de los alumnos:
→Título, →Autores, →Palabras Clave, →Resumen, →<artículo.formato>.

Este proceso sufrirá modificaciones en función de las características del software utilizado para la implementación de la revista.

2.12. Encuesta de un tema de actualidad

En esta sección se propondrá a los estudiantes, a través de un marco dinámico, una encuesta sobre algún tema de actualidad que les afecte. Las repuestas pueden ser si/no o cuatro opciones.

3. Estructura y tipo de los artículos

Los trabajos de los estudiantes se organizarán siguiendo la estructura habitual de los artículos científicos y técnicos. Existen pocas variaciones en este tema, por lo que sólo se han tenido en cuenta dos ejemplos suficientemente ilustrativos de los esquemas utilizados habitualmente.

3.1. Estructura de los trabajos

Los ejemplos de estructura de artículos considerados son las siguientes:

- *ACTA*: Estructura de la compañía ACTA Press, que publica trabajos científicos y técnicos [1].
- *IMRAD:* Esquema de organización y recomendaciones de la Universidad de Wisconsin (Introduction, Methods, Results And Discussion) [4].

En la Tabla 1 se comparan las diferencias y coincidencias entre estos dos esquemas. De estos ejemplos se puede obtener un formato de contenidos para los trabajos bastante general, que presentamos a continuación:

1. Título
2. Autores/titulación/e-mail
3. Resumen
4. Palabras clave
5. Introducción
6. Metodología: Materiales, y métodos, técnicas y herramientas
7. Experimentos y resultados
8. Discusión y conclusiones
9. Agradecimientos (opcional)
10. Referencias bibliográficas

ACTA Press	*IMRAD*
Resumen	Resumen
Palabras Clave	
Introducción	Introducción
Metodología	Métodos
Resultados	Resultados
Conclusión	Discusión
Agradecimientos	
Bibliografía	

Tabla 1. Estructura de los artículos de ACTA Press. e IMRAD.

3.2. Tipo de trabajos

Todos los puntos anteriores son suficientemente conocidos. Sólo cabría hacer algunos comentarios que particularicen los contenidos de los apartados 6, 7 y 8 para los diferentes tipo de artículos. Estos pueden cambiar su nombre o subdividirse en función de las necesidades de claridad y tipo de trabajo.

Los tipos de trabajos considerados son los siguientes:

I. *Revisión*: En estos trabajos se recopila con detalle el trabajo realizado hasta la fecha en un campo del saber. Incluye frecuentemente comparaciones entre diferentes teorías o técnicas utilizadas hasta el momento y las últimas tendencias.

II. *Estudio*: En este caso el trabajo trata de aclarar, a través de la experimentación, o el trabajo deductivo, el comportamiento de un sistema, una técnica, etc. Puede incluir el uso simuladores o plataformas, tanto software como hardware.

III. *Diseño de Sistemas*: En este tipo de trabajos se plantea el diseño, y con frecuencia

la posterior implementación, de un dispositivo, más o menos complejo, que dé solución a unos requerimientos previamente planteados. El trabajo realizado incluirá normalmente una parte software y otra hardware.

IV. *Diseño de Software*: Estos trabajos están orientados al desarrollo de software que resuelva un problema práctico.

La Tabla 2 ilustra las variaciones que podrían darse entre los cuatro tipos diferentes de artículos que hemos comentado para los apartados 6, 7 y 8.

Trabajo	*Revisión*	*Estudio*	*Diseño Sistemas*	*Diseño Software*
Metodología	Pasos, mecanismos, lenguajes y arquitecturas utilizadas			
Materiales	Presentación de las fuentes, teoría o técnicas revisadas	Simuladores, plataformas, arquitectura, definiciones	Plataformas, arquitectura, componentes, …	Programas y lenguajes, Arquitectura, …
Métodos, técnicas y herramientas	Descripción, comportamiento (ventajas e inconvenientes) y discusión detallada de las teorías o técnicas revisadas,	Sistemas de reglas o procedimientos generales, análisis teórico del problema, …	Descripción de alto nivel del sistema diseñado: Hard. y Soft., discusión de opciones, …	Descripción de alto nivel del software diseñado, discusión de opciones, metodologías, …
Experimentos	Explicación de los criterios de valoración y ordenación utilizados	Indicación de las pruebas realizadas con detalle, indicar condiciones de la experimentación, …	Esquemas de sistema diseñado: Hard y Soft, restricciones de aplicación, …	Descripción del Soft. diseñado, módulos, rangos de aplicación, …
Resultados	Clasificaciones, ordenaciones, recomendaciones para diversos casos,…	Tablas y gráficas que muestren los datos obtenidos, …	Funcionalidad, prestaciones, comportamiento, …	Funcionalidad, pres-taciones, comportamiento…
Discusión y Conclusiones	Tendencias, elecciones de las mejores teorías o técnicas, …	Tendencias de los resultados obtenidos, justificación o constatación con los objetivos o hipótesis, excepciones, …	Resumen del sistema diseñado, consecución de objetivos, limitaciones, …	Resumen del software diseñado, consecución de objetivos, limitaciones, …

Tabla 2. Contenidos de las variaciones que podrían darse para los tres tipos diferentes de artículos.

4. Software para implementar la revista

Hemos realizado un estudio de diferentes tipos de plataformas para implementar la revista. En todos los casos hemos tenido en cuenta que fuera software libre y que estuvieran basados sobre estándares de programación ampliamente difundidos. Se han evaluado dos tipos de software: los de orientación generalista, conocidos como CMS (Content Management System), entre los que se encuentran Drupal [2] y Mambo [4], y otros más específicos para el diseño de revistas *on-line*, entre los que hemos evaluado SPIP [6] y OJS [5].

La plataforma para una revista electrónica tiene que poseer algunas de las capacidades de los sistemas de manejo de contenidos (CMS) y un sistema de control de flujo de trabajos (*workflow*). Además, estos sistemas deben tener un sistema de identificación y protección de detos ajustado a la legislación vigente.

A continuación resumimos algunas de las

características de los cuatro sistemas evaluados y las razones por las que nos hemos decantado por el OJS.

Drupal

Se trata de un CMS basado en PHP. Es muy modular, por lo que resulta relativamente sencillo ampliarlo con nuevas secciones no implementadas o activar/desactivar las existentes. Además, las categorías taxonómicas que soporta permiten clasificar en distintos grupos una misma noticia por lo que la posterior confección de números monográficos se simplifica. Sin embargo, se trata de un CMS demasiado generalista, siendo por ello elevado el trabajo de configuración de una revista, lo que repercutirá en su posterior mantenimiento. En particular, no lleva implementado un *workflow*, por lo que habría que diseñarlo entero.

Mambo

Se trata de uno de los CMS más utilizados actualmente y con una comunidad de usuarios muy activa. Está basado en PHP y sobre mySQL. Posee características muy similares a Drupal, aunque lo aventaja en cuanto a la sencillez de manejo de la herramienta, ya que tan sólo es necesario configurar el entorno. Existe gran cantidad de información procedente de manuales y comunidades online. Por el contrario, carece de categorías taxonómicas (el mayor punto fuerte de Drupal). Al igual que el anterior, sigue siendo un sistema muy generalista y no implementa tampoco *workflow*.

SPIP

Se trata de un sistema orientado a la redacción de revistas online por lo que resulta más sencillo que los CMS anteriores a la hora de configurar e introducir información. Implementa en su sistema procedimientos de *workflow*. Posee algunas de las posibilidades que ofrecen los CMS, aunque limitadas. Está programado en PHP y sobre mySQL. Sin embargo, las noticias se clasifican por categorías simples (tan sólo pueden estar en una de ellas), por lo que tiene menor potencia que Drupal en este sentido y los artículos que circulan por el flujo de trabajo deben ser *"introducidos"* en el sistema a través de formularios, por lo que deben estar constituidos por texto plano (a no ser que se permita la introducción de etiquetas *html*).

OJS

Al igual que SPIP, está pensado para revistas online e implementa en su sistema procedimientos de *workflow*, pero permite en el proceso la inclusión de artículos con distintos formatos: *html, pdf*, etc. Posee algunas de las posibilidades que ofrecen los CMS, aunque limitadas (permite la creación de foros de noticias). Está programado en PHP y sobre mySQL. Inicialmente parece que existe cierta rigidez en cuanto al resto de contenidos pero se pueden modificar los ficheros PHP para ir adaptándolo a las necesidades de la revista.

5. Formato electrónico de los artículos

A la hora de elegir un formato en que se deben enviar los artículos a la revista hay que tener en cuenta los siguientes aspectos:

- Visualización "on-line".
- Posibilidad de impresión.
- Flexibilidad de edición.
- Portabilidad.
- Facilidad de confección por parte de los alumnos.

De las diversas opciones posibles, una de las más utilizadas en las revistas electrónicas es el formato PDF. Sin embargo, éste es un formato estático, pensado para su impresión en papel, poco flexible para su edición y no es viable su transformación en HTML para su visualización *on-line*. Por el contrario se daría libertad al estudiante para su elaboración.

Otra opción sería utilizar LaTex, dado que es factible la obtención de una versión PDF, aunque no HTML para su visualización "on-line", y es fácil de editar, pero requiere de un largo aprendizaje previo por parte del alumno.

El HTML es ideal para la visualización *on-line*, pero posee rigidez de edición, no está pensado para su impresión en papel y es complicada su conversión a PDF.

La posibilidad más generalista es utilizar SGML/XML, en formato "docbook" [7]. Es portable, posee flexibilidad de edición y se puede convertir fácilmente a PDF y HTML. Sin embargo, su confección es tediosa, incluso con editores avanzados de software gratuito.

La última opción considerada, que engloba de alguna manera a la anterior, sería utilizar el formado SXW de OpenOffice [8], que permite generar actualmente PDF y HTML, por una parte, y está previsto que próximamente pueda guardar los documentos como XML (en formato

"docbook"). Esto facilitaría la confección de los artículos y su posterior edición. Además, éste es un editor que funciona tanto bajo Linux como Windows y es un software gratuito. El único inconveniente son ciertos fallos de estabilidad en las versiones antiguas de esta plataforma y que han sido subsanados en la versión 2.0.

6. Evaluación

Existen iniciativas de e-revistas en otras universidades en las que se promueve la publicación de trabajos por parte de los estudiantes, solos o junto a profesores. Sin embargo, las e-revistas consultadas ([9-13] entre otras) están muy focalizadas en la investigación. Todas las encontradas son extranjeras, no habiéndose encontrado ninguna nacional, ni en el campo de la Informática. No existe tampoco, en ninguna de ellas, un vínculo tan grande con el trabajo del estudiante en las asignaturas de la titulación como el que se propone en esta e-revista. Por ello, ésta puede considerarse una iniciativa novedosa, tanto en la universidad española como en el ámbito de la Informática.

Aunque en un estadio aún muy inicial de la puesta en marcha de la e-revista, existen varias iniciativas que se han derivado de ella. En las titulaciones de Informática de la UJI, y con el fin de motivar a los estudiantes, se ha aprobado convalidar por créditos de libre configuración (1,5 créditos) la publicación de un trabajo en la e-revista. Se ha realizado un proyecto de innovación docente donde se ha establecido un formato para la entrega de los trabajos igual al de la e-revista y varios profesores ya lo están utilizando en sus asignaturas. Esto facilitará que los estudiantes se animen a presentar sus trabajos a la e-revista, al tenerlos ya en el formato de ésta.

Es pronto para evaluar el impacto efectivo de esta iniciativa en la formación de los estudiantes porque el número 0 acaba de ser editado. Sin embargo ya se han recibido algunas colaboraciones y propuestas de artículos para los próximos números y buena parte del profesorado se ha comprometido a utilizar el formato propuesto para la redacción de los trabajos de sus asignaturas.

7. Conclusiones

En este artículo se ha presentado la propuesta de una e-revista donde publiquen los estudiantes de las titulaciones de informática. Se trata de una herramienta que posibilita informar y formar a los estudiantes de forma transversal a las asignaturas del plan de estudios. Al mismo tiempo, sirve como vínculo integrador de la actividad académica dentro de éstas titulaciones y refuerza el desarrollo de las competencias de escritura y presentación de informes, tan necesarias en el trabajo de los futuros ingenieros informáticos.

En el proceso de puesta en marcha de esta e-revista se han tomado un gran número de decisiones. Esperamos que este trabajo sirva de ayuda y reflexión para todos aquellos, profesores y/o alumnos, que se decidan a emprender una iniciativa similar. Las decisiones finales que tomen podrán ser distintas a éstas, pero éste puede ser un material inicial para su reflexión.

Agradecimientos

Esta iniciativa ha sido posible gracias al apoyo de la UJI, a través de su plan estratégico, la colaboración de nuestros compañeros de los departamentos de Ingeniería y Ciencia de los Computadores y Lenguajes y Sistemas Informáticos que se han implicado en este proyecto y los estudiantes que han hecho posible su número 0.

Referencias

[1] ACTA Press. http://www.actapress.com/
[2] Drupal. http://drupal.org.
[3] IMRAD. http://www-math.science.unitn.it/LRM3D2/report.htm.
[4] Mambo. http://www.mambo-foundation.org.
[5] Open Journal Systems. http://pkp.sfu.ca.
[6] SPIP. http://www.spip.net/es.
[7] N. Walsh, L. Muellner. *DocBook: The Definitive Guide.* Oreally & Associates, 1999.
[8] SUN. http://www.openoffice.org.
[9] http://law.richmond.edu/general/student.htm
[10] http://www.utexas.edu/research/student/urj/
[11] http://rideau.carleton.ca/philosophy/cusjp/
[12] http://www.loyno.edu/~history/journal/
[13] http://www-dcrp.ced.berkeley.edu/bpj/

Expectativas laborales de nuestros alumnos, ¿debemos adaptarnos?[1]

Julia González

[1] Universidad de Extremadura
Departamento de Ingeniería en Sistemas Informáticos y Telemáticos
Escuela Politécnica
Avd de la Universidad s/n
10.071 Cáceres
juliagon@unex.es

José Luis Sánchez-Carrasco

Grupo Extela
Secretario General de FIEX
C/Abilio Rosillo 6, bajo
10.001 Cáceres
jlsanchez@extela.com

Carolina González[3]

Junta de Extremadura
IES Universidad Laboral
Avd. de la Universidad
10.005 Cáceres
carolina.gonzalez.rodriguez@gmail.com

Resumen

Los contenidos de nuestros títulos se basan en los contenidos propuestos por directrices generales y desarrollados a partir de puntos de vistas de académicos y profesionales. Pero generalmente estos contenidos, y las decisiones adoptadas para llevarlos a cabo, no tienen en cuenta al alumnado, ni tampoco a sus expectativas. Esta discordancia puede ser la causa de una alta tasa de abandono, presente en las titulaciones técnicas, e incluso del descenso del número de matrículas. Intentando conocer mejor a nuestros alumnos, sus expectativas, así como las opiniones de algunos de nuestros egresados sobre la formación que recibieron, se realizó un pequeño estudio. Los resultados del mismo se muestran en este artículo. Entre los objetivos del estudio se encontraba el dar respuestas que ayuden a mejorar nuestros planes de estudios, dada la situación actual, para modificar y/o incluir nuevos contenidos que se correspondan con las expectativas del alumnado y, además, comprobar si estas expectativas coinciden con las de los empleadores e incluso con las necesidades de los actuales empleados.

1. Introducción

En la elaboración del Libro Blanco [1] los profesores fuimos preguntados sobre nuestra opinión sobre los estudios de Informática (mediante encuestas), también fueron preguntados los profesionales y los empleadores, e incluso los egresados. En este conjunto posiblemente hubiera sido interesante incluir en las encuestas a los estudiantes que empiezan, que quizás no puedan aportar datos fiables relativas a las necesidades de un futuro Ingeniero en Informática, pero sí puede ayudarnos a conocer cuáles son sus intereses y sus expectativas respecto a la titulación conlleva. Esto podría ayudarnos a evitar descontentos y frustraciones que se producen cuando lo estudiado no corresponde con ideas preconcebidas respecto a la titulación elegida.

La comprensión de las inquietudes de los alumnos puede ayudarnos a la hora de disminuir la tasa de abandono presente en nuestras titulaciones [2] y además puede contribuir a favorecer el aumento de la tasa de nuevos ingresos, en este momento en el que el descenso del número de matriculados en un factor crucial. Aunque no debemos olvidar que la situación laboral actual existente en el lugar de residencia del alumno, así como su entorno social, condiciona y sesga sus opiniones acerca de su futuro profesional y por tanto de sus expectativas.

Con el objetivo de clarificar las expectativas de nuestros alumnos, a finales de 2005 se realizó un pequeño estudio que aunque no es significativo estadísticamente, si puede ser orientativo. El desarrollo del estudio y los datos obtenidos se

[1] Desarrollado bajo los proyectos TIN2005-09405-C02-02 del MEC, 3PR05A016, PDT006A042 y PDT006A045 de la Junta de Extremadura.

muestran en este artículo. También se introduce en la sección 2 un breve análisis de la situación económica y laboral de la región, para esclarecer el contexto del pequeño estudio realizado. En la sección 3 se describe la iniciativa llevada a cabo para orientar a los alumnos en las posibles salidas profesionales y de la que se obtuvieron los datos que se muestran en este artículo y que son analizados en la sección 4. Por último finalizaremos con las conclusiones obtenidas y pautas o estrategias a seguir según los datos analizados.

2. Situación laboral actual

Las relaciones entre trabajo y educación son estrechas y sus influencias son mutuas, tal y como demuestras estudios publicados [3][4]. De forma general las tasas más altas de desempleo afectan a la población joven, entre los que se encuentran los recién egresados. Estas tasas no son uniformes en todo el territorio nacional, existiendo importantes diferencias entre las Comunidades Autónomas. Las cotas más altas se alcanzan en Andalucía (26'8%), Extremadura (24'2%) y Asturias (18'2%).

También nos muestran estos estudios que existe una relación inversa entre el nivel de estudios y la tasa de paro. El colectivo con menos paro es el que posee el título de doctor, seguido de los que han finalizado estudios universitarios de Ingeniería y Tecnología, según datos del Ministerio de Educación Cultura y Deporte, 2000 [2]. Sin embargo, estar en posesión de un título universitario no garantiza una carrera profesional ajustada al objeto de estudio.

Si nos centramos en la situación de nuestra comunidad autónoma, con el fin de poder realizar un análisis de los resultados obtenidos en el estudio presentado, podemos aclarar que la economía extremeña ha experimentado a lo largo de las últimas décadas importantes transformaciones estructurales, pasando de ser una economía agraria a otra en la que los servicios se han convertido en el eje principal de la economía, según datos suministrados por la Consejería de Economía e Industria relativos al 2002[4]. La tasa de paro en Extremadura es superior a la media española, donde el colectivo de mujeres y el de jóvenes menores de 25 años de edad son los más afectados por el problema del desempleo, alcanzando la tasa de desempleo juvenil el 44,4%.

Nos encontramos en una región poco industrializada, con tasas de paro superiores a la media nacional y europea, donde no existen grandes núcleos de población, lo que hace que la densidad de población sea muy pobre. Esto provoca la inexistencia, hasta ahora, de grandes empresas, que generen necesidades específicas que deban de ser cubiertas con mano de obra especializada, si no más bien lo contrario.

El mercado de trabajo regional para profesionales informáticos no es capaz de absorber a todos los titulados, por ello la mayoría de titulados en la Universidad de Extremadura desarrollan su carrera profesional fuera de la región, principalmente en Madrid, Sevilla y Valencia.

La demanda regional de titulados informáticos se realiza principalmente a través de:

- La Administración Pública.
- La docencia en Educación Secundaria y Universitaria.
- Sector privado no especializado. Micropymes.
- Sector privado informático: formado por empresas de servicios informáticos, delegaciones comerciales, empresas de desarrollo de software, enseñanza, etc. Aunque la demanda ha aumentado, los puestos a menudo no se corresponden con el perfil de un titulado Superior, si no de profesionales procedentes de los ciclos formativos.

La idiosincrasia de la región favorece a la emigración nuestros titulados, o a que desarrollen, en términos generales, una carrera profesional no especializada, desarrollada en pequeñas empresas y en las administraciones públicas, regionales y locales.

3. La Universidad, las empresas y los futuros ingenieros en Informática

Actualmente los tres títulos de Ingeniería Informática se imparten en al Universidad de Extremadura, en dos campus: Mérida, donde se imparten Ingeniería Técnica en Informática de Sistemas (ITIS) e Ingeniería Técnica en Informática de Gestión (ITIG) y Cáceres, donde se imparte, además de las dos titulaciones mencionadas anteriormente el título de Ingeniería en Informática (II).

Desde el curso 2004/05 se viene apreciando un descenso en la matrícula en estas titulaciones, siendo este curso, 2006/07 el decremento más que notable en la titulación de ITIG, especialmente en el campus de Mérida. Bien es cierto que el número de matriculados total en la UEX también ha disminuido y que este decremento no puede achacarse exclusivamente a nuestros títulos ni tampoco a nuestra Universidad, tal y como refleja [5].

Causas determinantes de esta situación son el descenso del número de jóvenes en edad universitaria, el aumento de oferta formativa no universitaria y una sociedad, la extremeña que, debido a sus particularidades, demanda profesionales de incorporación inmediata, donde perfil solicitado es de un titulado con conocimientos generales que pueda integrarse en pequeñas empresas, donde las más especializadas se dedican al desarrollo de software a medida, En esta situación detectamos una falta de sincronía entre los perfiles proporcionados por la Universidad, más especializados, y los solicitados por las empresas, y esgrimimos, puede ser uno de los motivos para el descenso en el número de alumnos matriculados. La solución a esta situación pasa por un acercamiento entre el mundo empresarial y la Universidad.

Con este objetivo e intentando abrir cauces de comunicación el Departamento de Ingeniería en Sistemas Informáticos y Telemáticos, de la Universidad de Extremadura y a la Federación de Asociaciones provinciales de Empresas de Tecnologías de la Información de Extremadura, decidieron unir sus esfuerzos para romper las barreras de comunicación existentes hasta el momento y realizar actividades que pudieran ayudar a los futuros titulados en Informática.

Como primera actividad de esta unión nacieron las "I Jornadas de Orientación a las Salidas Profesionales para Ingenieros en Informática", cuyo objetivo era dar comienzo a un triángulo que permitiera unir esfuerzos y mejorar la capacitación final de los alumnos, orientándolos hacia los sectores reales en los que van a verse inmersos.

Aprovechando dicha actividad, se decidió realizar un estudio "ad hoc", dada la alta participación en el evento, que aportara información sobre las expectativas profesionales de los alumnos, así como información sobre la opinión de los egresados, en términos de la formación recibida en la UEX.

El resultado del análisis de esta información se traslada a este artículo, con el único objetivo de iniciar un periodo de reflexión y quizás como punto de origen al que podría ser estudio más exhaustivo centrado en la misma temática.

4. El evento

Una vez que se acercaron el mundo empresarial y el académico, tras varias jornadas de reflexión, y varias propuestas, conjuntamente se decidió organizar un evento en el que el mundo empresarial aportara uno de sus bienes más preciados: el conocimiento del mundo laboral. El objetivo era mostrar a nuestros alumnos las salidas profesionales más habituales, para lo cual se elaboró un catálogo.

En una segunda fase se produjo un debate sobre el enfoque del evento, una vez que los contenidos habían quedado fijados en la fase previa. Se llegó a la conclusión de la necesidad de realizar un evento cercano, lleno de realidad, donde los ponentes pudieran hablar en un lenguaje muy sencillo y directo a los alumnos y que a su vez, los ponentes pudieran ser partícipes de las dudas y de los problemas a los que los alumnos se estaban enfrentando. Ante esta decisión se optó por buscar, entre los egresados en Informática de la UEX, profesionales que se adaptasen a cada uno de los perfiles seleccionados.

4.1. El desarrollo del evento

Todos los ponentes fueron alumnos de la Escuela Politécnica de la UEX, por lo que además de proporcionar información sobre su trayectoria profesional y su trabajo actual, pudieron aportar su propia experiencia como alumnos de la Escuela, ofreciendo una perspectiva humana y más cercana de la profesión.

Se escogieron ponentes con al menos cinco años de experiencia, considerados de desarrollo profesional pleno, que actualmente están trabajando en el ámbito que representaban. Y que desarrollan su actividad tanto dentro como fuera de la región, según los casos.

Por decisión organizativa el evento se celebró en Cáceres, dentro de la semana de actos de la Escuela Politécnica, en un único día.

Las presentaciones se estructuraron en tres bloques temáticos, tal y como muestra la siguiente relación:

1. Actividades por cuenta propia
- Ayudas y apoyo institucional para emprender en Extremadura
- E-emprender
2. Trabajar en una empresa privada
- Proyectos de I+D+i
- Desarrollo de sistemas de gestión comercial
- Desarrollo y telecomunicaciones en un entorno internacional
3. Trabajar en las administraciones públicas
- Enseñanza Secundaria
- Universidad privada
- Universidad pública
- Administración local y autonómica

Para el desarrollo de estas ponencias se invitó a funcionarios de la administración pública, profesores de universidad (pública y privada), jefes de proyecto de empresas de ámbito nacional, propietarios de pequeñas empresas, etc.

Cada presentación siguió siempre el mismo esquema: breve presentación del ponente, en la que se incluía su trayectoria profesional, su situación actual, su futuro más inmediato y su opinión personal sobre la profesión.

Tras cada intervención se estableció un turno de preguntas, en el que los alumnos participaron activamente, en el que se aclararon dudas sobre el trabajo desarrollado, las competencias necesarias, fases de selección superadas para obtener el puesto de trabajo, métodos de las empresas para realizar la selección, horarios de trabajo, oportunidades de promoción, etc.

El evento quedó clausurado con una mesa debate sobre la situación de la titulación en la actualidad, así como las deficiencias y puntos fuertes que los ponentes habían encontrado en su desarrollo profesional.

Paralelamente al evento se realizaron dos encuestas, una destinada a los ponentes, otra destinada a los asistentes.

La primera de ella se realizó antes del evento, y parte de la información estuvo disponible antes de las jornadas.

La segunda se realizó después, una vez que los alumnos conocían los perfiles.

En el siguiente apartado veremos con más detalle cada una de ellas.

4.2. Las encuestas

Antes de la celebración del evento a cada uno de los ponentes se le remitió una pequeña encuesta dividida en tres partes: la primera consistía en la descripción de su perfil actual, la segunda en su evolución profesional y la tercera en su opinión subjetiva sobre la titulación. Tras su recepción por parte de la organización, los datos de la primera parte fueron publicados en la web del evento.

Los datos solicitados fueron los siguientes:

1. En la actualidad
- Empresa
- Puesto
- Tiempo que lleva en la empresa
- Jornada laboral diaria
- ¿Trabaja los fines de semana?
- Asiduidad de viajes por trabajo
- Promoción académica a la que pertenece
- Ha pasado por más puestos en la empresa
- Remuneración actual
- Pasos seguidos para entrar en la empresa
2. Anteriormente
- En cuántas empresas ha trabajado
- En qué puestos
- Cuál fue su primer sueldo
- Por qué cambió de empresa
3. Sobre la titulación
- Titulación
- Fecha de finalización
- ¿Es buena la formación obtenida?
- ¿Qué ha necesitado aprender?
- ¿Qué no ha utilizado nunca?
- ¿Qué debería haber y qué falta en los planes de estudio de un Ingeniero Informático?

El objetivo de que esta información estuviese disponible previamente a la celebración de la jornada, era que los alumnos, además de conocer el perfil de cada uno de los participantes, pudieran realizar una “comparación” de cada una de las salidas profesionales propuestas según los parámetros más habituales: sueldo, horas de trabajo, viajes, estabilidad, posibilidad de promoción, etc.

Con esta información y tras las exposiciones, una vez finalizadas las jornadas, se habilitó, dentro de la web del evento, un formulario para la recogida de información.

Se pidió a los alumnos que eligieran un perfil profesional, aquel en el que deseaban desarrollar su carrera, y una justificación de la decisión tomada. Además se les solicitaba una valoración razonada sobre el evento y una opinión personal.

Se estudiaron los inconvenientes de la realización de la encuesta a los alumnos tras las jornadas, por la influencia del ponente del perfil, su poder de cautivación y su propia experiencia personal. También se valoraron los inconvenientes de hacerla antes de las mismas, pues desconocíamos cuáles eran las preconcepciones con las que contaban. Tras sopesar ambas opciones se optó por la recolección de datos después de la celebración del evento, que al menos garantizaba que la información sobre la que emitían sus valoraciones y tomaban su elección era, para todos los casos, idéntica y real.

4.3. Los datos obtenidos

- **Los ponentes**

En total fueron 8 los ponentes. Todos ellos con una experiencia media en el mundo laboral de más de seis años. El que menos había estado ejerciendo su labor como profesional eran cinco años años, el que más dieciséis años.

Actualmente, los ponentes que desarrollan su actividad en grandes empresas pertenecen a la plantilla de jefes de proyecto o directivos, todos han pasado por más de un puesto en la empresa, en la que han ido evolucionando. Todos, admiten que su jornada laboral está por encima de los límites establecidos en su contrato y que suelen viajar por cuestiones de trabajo. Según aumenta la responsabilidad del puesto que ocupan, aumenta el número de viajes que realizan al mes.

En las empresas privadas, incluida la Universidad Privada, tuvieron que realizar una selección previa a través del currículo, pasar una selección dirigida por el departamento de recursos humanos, en ocasiones mediante sesiones grupales y finalmente una entrevista personal.

Se puso de manifiesto la diferencia de trabajo y necesidad de actualización tecnológica entre unos puestos de trabajo y otros. Los trabajadores por cuenta propia, gerente y directores de sus propias empresas estaban más alertas de los cambios tecnológicos que los trabajadores por cuenta ajena, principalmente los situados en las administraciones públicas, y de manera abrumadora en la Enseñanza Secundaria, debido a que el perfil de Informática, actualmente, es inexistente.

Resultó interesante comprobar que todos los ponentes coincidían básicamente en los mismos puntos a la hora de enumerar las fortalezas y debilidades de la formación recibida.

Todos consideraron en términos generales que la formación que habían recibido en la UEX como buena. Coincidían en la necesidad de dar más importancias a los idiomas, a las habilidades sociales y a competencias transversales como la capacidad de trabajar en grupo o la realización de presentaciones orales. También proponían bajar la importancia de materias teóricas con difícil aplicación práctica.

- **Los asistentes**

El aforo inicial estaba restringido a 60 personas, que hubo que aumentar porque en menos de 24 horas se completó. Finalmente asistieron 124 alumnos, de los que 101 rellenaron la encuesta de la que se han obtenido los datos reflejados en este artículo. Sus distribución por titulación era la siguiente: 47 matriculados en Ingeniería en Informática, 30 en Ingeniería Técnica en Informática de Gestión, 17 en Ingeniería Técnica en Informática de Sistemas y 7 alumnos para los que no tenemos datos de la titulación a la que pertenecen. El grupo más numeroso pertenece a alumnos de tercer curso, con 48 alumnos. El resto se distribuye en 20 alumnos de segundo curso, 14 de cuarto, 10 de primero y 9 de quinto.

Del tratamiento de los datos obtenidos en la información facilitada por los alumnos se han obtenido los resultados expuestos en el siguiente apartado.

5. Análisis de los datos

Uno de los objetivos era constatar si la creencia de que los factores sociales influían en el carácter del alumnado de la UEX era cierta, de tal manera que uno de los perfiles profesionales más deseados fuera el de funcionario público. Queríamos conocer cuál era el ámbito profesional en el que los alumnos desean desarrollar su carrera.

Debido a nuestra premisa inicial, no nos sorprendió que el 25% de los alumnos encuestados, (ver figura 1.a), antes de finalizar sus estudios ya hayan decidido apostar por una carrera profesional estable dentro de la administración

pública. Las razones mayoritariamente que sugieren para esta elección son la seguridad, la estabilidad y la falta de responsabilidad, además del horario continuado, lo que subraya el carácter apático y de falta de iniciativa que se observa dentro de las aulas. Dentro de las posibilidades de la administración, optan principalmente por ser profesores de Educación Secundaria, (figura 1.b), basando su elección en el sueldo y en el largo periodo vacacional, aunque algún que otro alumno habla de vocación por la docencia. Estos datos contrastan con un 2%, que afirma que su única preferencia es la de no ser docente. Ambos datos vienen a exponer, de nuevo, que la docencia no está considerada socialmente, y que los estereotipos siguen manteniéndose. De hecho la ponencia explicativa del trabajo en la Universidad Pública, en la que se incluía el camino a seguir para obtener un puesto de funcionario, fue la más contestada. Aún habiéndose mostrado el camino de la LOU de manera objetiva, la respuesta unánime de los presentes fue de incredulidad, los comentarios generales indicaban que dudaban de que el camino fuese tan complicado. Según sus comentarios ellos esperaban que el acceso se realizase mediante un examen o "algo así". Aún así, 3 estudiantes mantuvieron que quieren ser profesores de Universidad, dos de ellos motivados por la investigación. Un alumno quiere trabajar en la Universidad, pero no como docente, tampoco como investigador, ¿PAS quizás?

Considerando el alto porcentaje que desea ser funcionario, resulta impactante que el 20% de los encuestados muestren vocación empresarial y su máximo objetivo sea crear su propia empresa. En éstos no están incluidos aquellos que han decidido opositar primero y una vez obtenga un puesto estable, crear su propia empresa. Incluso hay algún alumno al que le parece interesante trabajar en una empresa, y utilizar los recursos de la misma para montar su propia empresa. Cuando menos sorprendente respuesta.

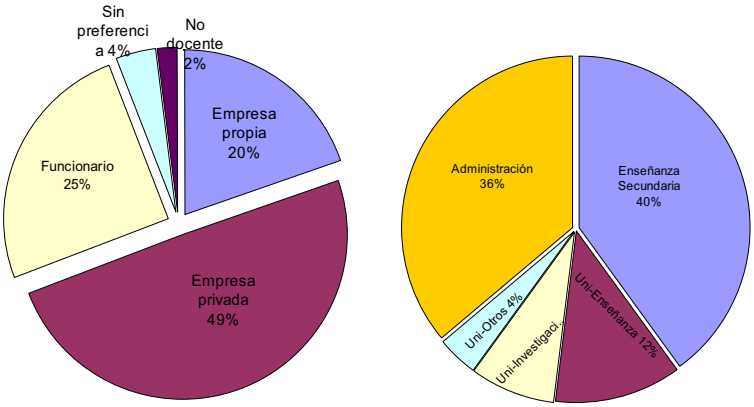


Figura 1. a) Ámbitos profesionales b) Distribución de perfil de funcionario

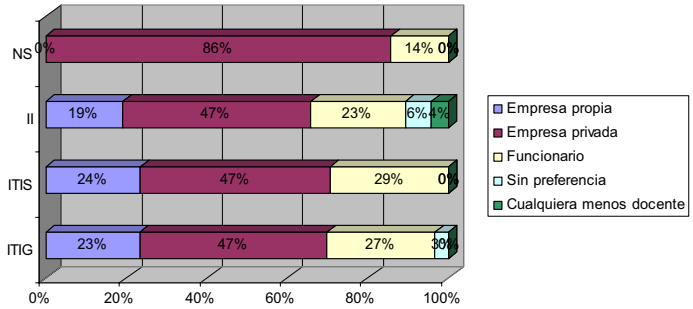


Figura 2. Distribución de perfiles por titulación

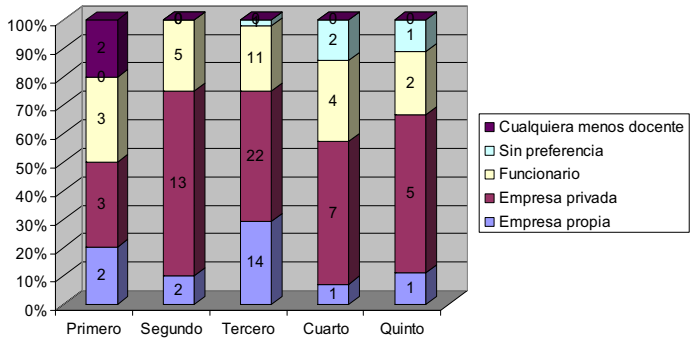


Figura 3. Distribución de perfiles por curso

Si analizamos los datos obtenidos por la titulación que cursan, podemos ver que no existen diferencias significativas entre los resultados obtenidos y que los porcentajes de los perfiles se mantienen. Disminuye ligeramente el porcentaje de alumnos que desean ser funcionarios para los estudiantes de Ingeniería en Informática, aumentando sus dudas, tal y como demuestra el 6% que ha afirmado no tener preferencia en la elección (figura 2). Posiblemente este resultado es debido a que en los cursos finales de la titulación, los alumnos disponen de una mayor oferta de contenidos, por lo que se les abren nuevas posibilidades; así lo demuestran con sus respuestas los alumnos que desean trabajar en campos concretos como son Web, multimedia, software libre o robótica.

Analizando los datos según el curso en el que está matriculado el alumno observamos que el número de personas que desea emprender se concentra mayoritariamente en el tercer curso. Posiblemente ven que el momento de salir al mercado laboral se acerca, con lo que sus preferencias laborales cambian. Sin embargo es difícil comprender el porqué de la elección de ser empresario, ya que las asignaturas dedicadas a la creación, gestión y control de empresas en el plan de estudios aparecen como optativas en los planes de Ingeniería en Informática e Ingeniería Técnica en Informática de Sistemas. Lo que nos puede llevar a reflexionar y estudiar la posibilidad de aumentar la oferta en asignaturas con estos contenidos.

Cómo era previsible, cerca del 50% de los alumnos quieren trabajar en una empresa privada. De forma general estos alumnos desean trabajar como analista y/o programadores. Sus materias preferidas están relacionadas con la programación, por lo que centran sus intereses según estas preferencias. Entre las razones que esgrimen para

esta elección se encuentran las posibilidades de promoción y de continuo reciclaje. Algunos alumnos especifican su preferencia por empresas de ámbito internacional que les permitan una mayor promoción y además viajar.

Los resultados obtenidos en este simple estudio no nos permiten la toma de ninguna decisión, pero sí nos permiten elaborar un conjunto de conclusiones que pueden ayudarnos a entender a nuestros alumnos, a adaptarnos, si lo consideramos necesario o al menos a ofrecerles más información que no le lleven a crear falsas expectativas sobres sus perspectivas laborales futuras.

6. Conclusiones

El estudio realizado no puede ser considerado relevante, tanto por la cantidad de alumnos encuestados como por la calidad de la muestra que no ha sido analizada de forma estadística, por lo que no podemos considerar que se válida para establecer un patrón de comportamiento general en nuestros alumnos. Aún así, debido a la voluntariedad de los datos recogidos, a que muestran de forma general las mismas opiniones vertidas en el evento, podemos clasificar los resultados no como significativos pero quizás sí como interesantes.

Este estudio nos permite conocer un poco mejor cuáles son los intereses de nuestros alumnos, cuáles son sus expectativas, e incluso nos permite elucubrar sobre la relación existente entre la situación social de la región y sus preferencias en su desarrollo profesional, tal y como demuestran las preferencias por la administración pública y la creación de empresas.

En un primer momento los resultados obtenidos en cuanto al carácter emprendedor que han demostrado tener nuestros alumnos, es un dato sorprendente, que puede haberse visto influido por las políticas autonómicas sobre el emprendimiento joven y que, hasta el momento del estudio, no habían sido consideradas en la titulación. Éste es un dato que no debe ser pasado por alto, y debería ser considerado en el nuevo plan de estudios, así como en el planteamiento futuro de la planificación y adaptación de las materias al nuevo Espacio de Educación Superior. Por ello debemos considerar incluir nuevas asignaturas que les ayuden en la gestión y dirección de empresas, junto con la adquisición de las competencias necesarias para cualquier emprendedor.

No podemos olvidar tampoco las referencias realizadas hacia la Enseñanza como salida profesional. Si vamos a tener futuros docentes, debemos dotarlos y potencias las capacidades adecuadas.

Además de la adaptación de la titulación, los datos de este tipo de estudio nos pueden a ayudar a guiar a nuestros futuros egresados, pero también a deshacer los equívocos que pueden conducir al estudio de las titulaciones conducentes a la obtención de algunos de los títulos en Informática a través de las "Jornadas de Puertas Abiertas" de la Universidad y de acercamiento a los centros de enseñanza pre-universitaria. De este modo podremos disminuir las tasas de fracaso y puede que conseguir un aumento en la tasa de nuevo ingreso.

Conocer los objetivos de nuestros estudiantes nos ayuda a conocerlos, a conocer a nuestra sociedad y por tanto nos ayuda a adaptarnos y dar un mejor servicio, de calidad.

Agradecimientos

A todos los ponentes que participaron en las jornadas: Julio Yuste, Pedro Palos, Carlos Rebate, Natalia González, Francisco Zarallo, Manuel Batalla, Antonio Reinoso, Juan Manuel Murillo y Pedro Rocha.

Referencias

[1] Libro blanco del título de grado en Ingeniería en Informática. ANECA 2005. http://www.aneca.es

[2] Clemente P., Gómez A., González J., Sánchez H., Sosa E. *Hacia la convergencia europea. Nuestros indicadores de calidad y su mejora.* JENUI 2005

[3] Centro de investigación y documentación educativa. *El sistema educativo español.* Ministerio de Educación, Cultura y Deporte, 2000.

[4] Consejería de Economía, Industria y Comercio. *Extremadura en cifras.* Junta de Extremadura, 2002

[5] Blesa A., Bueso P., Catalán C., Lacuesta R., Ubé M., *Actuaciones de mejora docente en una escuela universitaria politécnica.* JENUI 2005

Desarrollo del proyecto de reciclaje de equipos informáticos en el marco de Teruel Digital

Francisco Martínez, Fernando Naranjo, Piedad Garrido
Dpto. de Informática e Ingeniería de Sistemas
Universidad de Zaragoza
Pedro Cerbuna 12, 50009 Zaragoza
{F.Martinez, fnaranjo, piedad}@unizar.es

Resumen

En la presente ponencia se presentará la actuación de Reciclaje de Equipos Informáticos, que se realiza en la provincia de Teruel en el marco del proyecto Teruel Digital.

En dicha actuación colaboran varios profesores del Departamento de Informática e Ingeniería de Sistemas y dos becarios, alumnos de la Escuela, y pretende recoger los equipos informáticos que ya no utilizan los organismos públicos, particulares y empresas, para ponerlos a disposición del resto de ciudadanos.

Además se realiza asistencia técnica a las asociaciones que pueden solicitar equipos donados, así como potenciar el uso de software libre en toda la provincia.

1. Introducción

El Reciclaje de Equipos informáticos es la actuación número 7 dentro del proyecto de Ciudad Digital en Teruel [4] y se enmarca en el convenio firmado entre el Gobierno de Aragón, el Ayuntamiento de Teruel, la Diputación de Teruel y la Universidad de Zaragoza.

Esta actuación se inició el 8 de mayo de 2006 y se centra principalmente en el desarrollo de un plan de Reciclaje de Equipos Informáticos, con la finalidad de permitir la reutilización de los viejos equipos donados por parte de las Administraciones Públicas, Empresas, Asociaciones o ciudadanos de Teruel.

Estos equipos una vez revisados, reparados o mejorados en el caso de que sea necesario, y puestos a punto son entregados a las asociaciones y ciudadanos de la provincia de Teruel que los demanden.

Tanto la recogida de donaciones, como las entregas se realizan teniendo en cuenta un protocolo de prioridades que permite satisfacer al mayor número de personas posible y asegurar un correcto desarrollo del Proyecto, haciendo especial hincapié en atender las necesidades planteadas por asociaciones sin ánimo de lucro, culturales y con fines sociales.

También es importante comentar que de forma paralela al Reciclaje, se realiza una labor de Asistencia Técnica a las Asociaciones de Teruel desde el punto de vista de las Tecnologías de la Información y la Comunicación (TIC).

Finalmente, otra de las acciones a realizar es potenciar el uso de software libre por parte del mayor número de ciudadanos de toda la provincia.

Figura 1. Algunos equipos recogidos

El presente documento se organiza de la siguiente forma: en el siguiente apartado se comentará en detalle la actuación de reciclaje así como el sistema de información que se ha desarrollado expresamente para apoyar y facilitar la realización del proyecto.

En el apartado 3, se explicarán algunos detalles relacionados con la Asistencia Técnica proporcionada a las diversas asociaciones turolenses.

Posteriormente se presentarán una serie de datos que permitirán al lector hacerse una mejor idea acerca del grado de implantación del proyecto.

Finalmente se presentarán las conclusiones obtenidas después de varios meses de trabajo.

2. El Reciclaje de Equipos

En la actualidad, la rapidez con que evolucionan las tecnologías de la información, el continuo incremento de los requerimientos informáticos para satisfacer nuestras necesidades laborales o lúdicas, el progresivo abaratamiento en el precio de componentes hardware y, por qué no decirlo, la inercia de la propia sociedad de consumo, llevan a una situación en la cual numeroso equipamiento informático pasa a tener la etiqueta de "obsoleto" con gran frecuencia sin que realmente deje de funcionar.

Ahora bien, ¿qué ocurre con esos equipos? ¿a dónde van a parar? Hasta hace un tiempo, la mayoría de ellos quizás se aprovechaban en el entorno cercano a donde se hubieran usado. Sin embargo, hoy en día sólo hay un lugar al que van a parar la mayor parte de estos equipos informáticos: al trastero, al almacén o a un rincón de una habitación donde no resulte demasiado molesto para pasar.

Y, sin embargo, la pregunta es: ¿todo este equipamiento es realmente tan poco aprovechable como le parece a quien se deshace de él? La respuesta es un rotundo "no". La realidad indica que, actualmente, existe un común denominador en usos administrativos, organizativos y de introducción a la informática según el cual los requerimientos mínimos de hardware para estos fines puede encontrarse en equipos de bastantes años más atrás de la fecha actual. Lo cual, unido a la aparición y progresiva expansión del software libre, crea el entorno perfecto para el presente proyecto.

Así pues, el proyecto de Reciclaje de Equipos Informáticos pretende dar una salida a todo ese parque de equipos informáticos que se encuentra "parado" y ocupando un espacio en su lugar de origen y destinarlo a asociaciones y ciudadanos que tienen limitado acceso a estas tecnologías, favoreciendo así la expansión del uso de las tecnologías de la información en la sociedad.

Figura 2. Página web del Reciclaje de Equipos Informáticos [3]

Por otro lado, en el caso de que algún equipo no pueda ser reutilizado, se reciclará de forma correcta, evitando así que pueda acabar en el contenedor de la basura, algo no deseable desde el punto de vista de la conservación del medio ambiente.

La idea fundamental que subyace en el proyecto que nos ocupa es la siguiente:

Proporcionar un medio a las entidades potenciales donantes y a las asociaciones potenciales solicitantes para que puedan realizar sus donaciones y solicitudes respectivamente.

A continuación, realizar unas tareas organizativas, administrativas y de priorización de cara a recoger, almacenar y contactar con las entidades y asociaciones para la recogida de equipos en el caso de donaciones y petición de información y necesidades en el caso de las solicitudes.

Realizar las labores propias de reciclaje de equipos, en las cuales se incluirán:

- Inventariado de equipamiento recogido
- Diagnosis del mismo
- Proceso de reciclado

No obstante, hay que destacar que este proceso de reciclado no es inmediato: se deben desmantelar equipos y aprovechar sus componentes para otros, así como resolver problemas de incompatibilidad y configuración de los mismos.

Además, el uso de software libre y en concreto Linux como sistema operativo conlleva toda una problemática adicional de adaptación a equipos antiguos.

Una vez que los equipos son reciclados, se procede a la asignación de los mismos a las solicitudes en función de las necesidades de los solicitantes recogidas previamente y las prioridades establecidas.

Finalmente, se concreta una cita con los solicitantes y se procede a la entrega de los equipos reciclados.

En la figura 3 se muestra el proceso de ejecución del proyecto de reciclaje.

El núcleo del Reciclaje de Equipos reside en el propio proceso de reciclado, una vez que se han recogido los equipos de un donante y hasta que los mismos quedan preparados para realizar la entrega a solicitantes. Dada la importancia de este proceso, se ha definido un Work-Flow que explicita completamente el flujo de información que se sigue y las actividades que se realizan dentro del citado proceso. En la figura 4 se presenta el modelo de flujo de trabajo definido, en el cual se pueden comprobar los distintos estados por los cuales pasa un equipo en reciclaje.

Dada la propia idiosincrasia del proyecto de reciclaje, se ha establecido como uno de sus puntales base la definición y construcción de una aplicación de gestión dotada de una interfaz web accesible a través de Internet.

De esta forma los alumnos que participan en el proyecto han podido ver cómo se ha diseñado el proyecto desde su comienzo, participar en el diseño e implementación de la aplicación, así como en la puesta en marcha del Reciclaje.

La aplicación de gestión pasa a ser descrita en el punto siguiente.

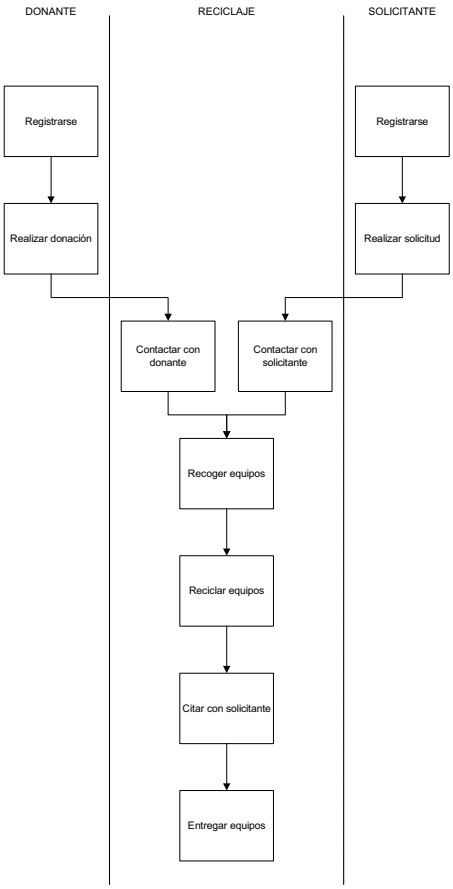


Figura 3. Diagrama de actividades

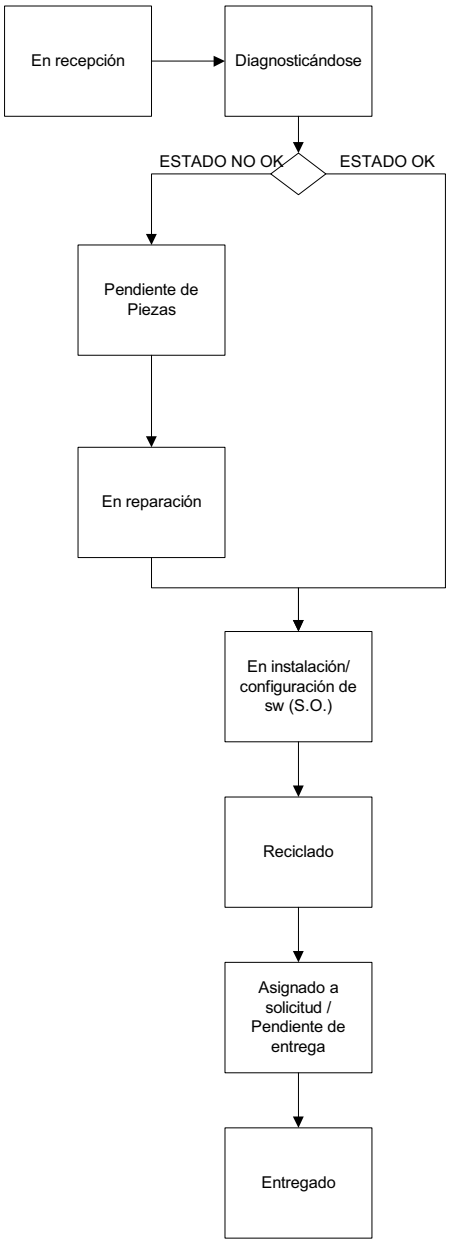


Figura 4. Modelo de Flujo de Trabajo

2.1. Descripción de la aplicación de gestión

De cara a la realización de la citada aplicación, se ha definido el proyecto informático correspondiente, con el nombre de "Sistema de Información - Plan de Reciclaje del Proyecto Ciudad Digital en Teruel".

Dicho Sistema de Información se compone a su vez de dos partes fundamentales.

- Front-end: Se ha diseñado y construido un sitio web que permite a las administraciones públicas realizar notificaciones de donaciones de equipos a reciclar. A su vez, las asociaciones y ciudadanos que lo deseen pueden realizar peticiones de equipos a través del mismo sitio web. Por último, el interfaz web permite la realización de un seguimiento de los equipos en proceso de reciclaje (trazabilidad) tanto por parte de los profesores coordinadores del proyecto como por el personal de Teruel Digital. Se han atendido a criterios de usabilidad y accesibilidad, así como se ha definido una imagen corporativa (logo, paleta de colores propios) y se ha seguido la guía de estilos previamente elaborada por Teruel Digital. La tecnología empleada para la implementación del sitio ha sido php, de forma que se favorece el uso de software libre.

- Back-end: Se ha diseñado e implementado una aplicación que se encarga fundamentalmente de soportar el Work-Flow definido en el proyecto y poder realizar el tratamiento de la información recibida a través del front-end, accediendo al repositorio de datos. La tecnología de soporte empleada en esta parte ha sido php con orientación a objetos para la lógica de la aplicación y MySQL en el repositorio de datos, de manera que seguimos favoreciendo el uso de software libre.

Figura 5. Página de Gestión del Sistema de Información

Finalmente, el sistema de información resultante posee una arquitectura de tres capas: capa de presentación, donde se encuentra la interfaz que interactúa con los usuarios (realización de solicitudes, donaciones,…) y definida por el sitio web, capa de lógica de aplicación, compuesta por las clases y sus métodos que tratan la información recibida mediante la interfaz (solicitudes, donaciones, actualización de estados de las mismas,…), y capa de datos definida por el sistema gestor de bases de datos MySQL.

Este esquema se presenta en la figura 6.

Cabe destacar que en el diseño e implementación de esta aplicación los alumnos han tenido un papel muy importante, estableciendo con ellos una adaptación de la metodología de desarrollo Métrica v3 [2] para la ejecución de dicha aplicación y pudiendo realizar su trabajo de forma muy parecida al desarrollo de un proyecto de forma profesional mediante el diseño y puesta en marcha de un flujo de trabajo en el proceso de reciclaje, de forma que puedan prepararse para una mejor incorporación al mercado laboral haciendo uso de éstas técnicas y herramientas.

3. La Asistencia Técnica

Otra tarea llevada a cabo por el equipo de Reciclaje es la llamada Asistencia Técnica. Esta tarea pretende ayudar a las Organizaciones de la provincia en su desarrollo o inicio en el mundo de las Nuevas Tecnologías.

Para ello de forma paralela al reciclaje, las asociaciones que han solicitado recibir equipos reciclados, reciben Asistencia Técnica por parte del equipo de Reciclaje.

De cara a realizar dicha asistencia, se han usado técnicas de auditoría recogidas en el estándar CoBit. [1]

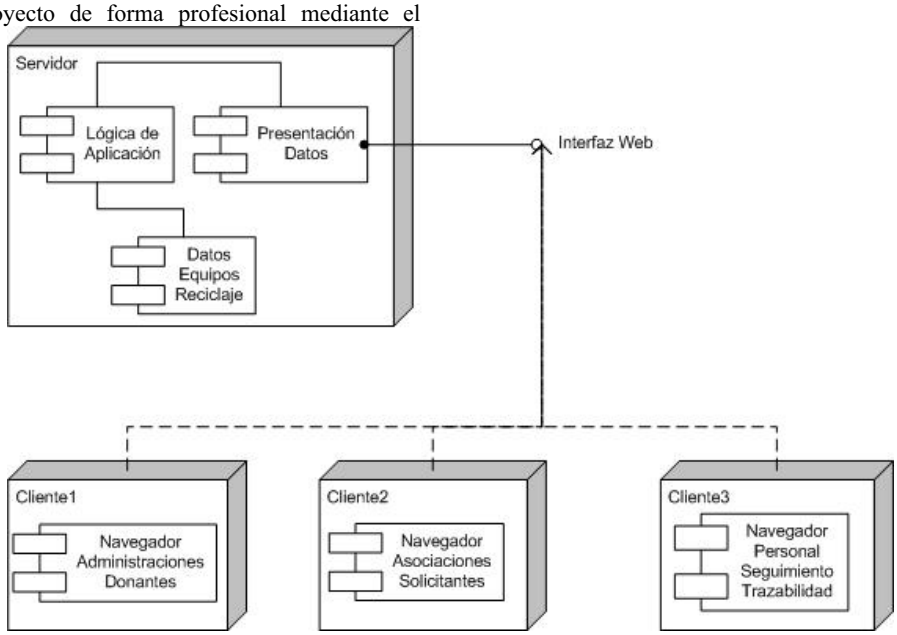


Figura 6. Diagrama de despliegue

Así pues, en primer lugar se mantienen una serie de entrevistas con algún miembro de la organización, intentando obtener la mayor información posible tanto de las actividades que realiza como de los medios técnicos de los que dispone en la actualidad. Esta información es muy

necesaria, pues permitirá adaptar al máximo la Asistencia Técnica a las necesidades particulares de la Organización.

Durante el proceso de recogida de datos, se va rellenando un documento con formato check-list en el que se detallan algunos aspectos importantes sobre la organización a asistir desde el punto de vista tecnológico y siguiendo el estándar CoBit de auditorías de Sistemas de Información.

De esta forma se estudia la situación actual de las organizaciones, de su infraestructura técnica a nivel de hardware y software, de su seguridad y de su plan de recuperación de contingencias.

Una vez realizado el estudio, se elaboran una serie de recomendaciones respecto a los puntos comentados anteriormente, con el fin de que las organizaciones puedan tenerlas en cuenta.

En la Asistencia, los alumnos han podido aprender a realizar entrevistas con "clientes", conocer los problemas derivados del trato con ellos, analizar una situación real y emitir una conclusión del análisis en función de los datos obtenidos, etc.

Hasta el momento se ha realizado asistencia técnica a 25 Asociaciones.

Esta parte del Proyecto ha sido de momento la menos desarrollada, debido fundamentalmente a que se han dedicado los mayores esfuerzos al resto de acciones que hasta el momento han tenido una especial demanda.

De todas formas, en la mayoría de los casos las características de las organizaciones son bastante similares y al no disponer de un sistema de información previo, la asistencia técnica suele ser sencilla. Esta actuación tiene mayor sentido realizarla con las asociaciones que tienen mayor tamaño, disponen ya de ordenadores e incluso tienen algún sistema de información que puede ser mejorado.

Debido a la situación comentada anteriormente, hemos planteamos la posibilidad de establecer, dentro de lo que sería el marco de actuaciones hacia la mitad del desarrollo de la actuación, es decir entre junio y septiembre, una asistencia técnica a las asociaciones que han sido objeto de entrega de equipos reciclados transcurridos unos meses desde la citada entrega, de cara a obtener una realimentación por su parte y de esta forma poder realizar posibles ajustes o mejoras en el proyecto.

4. Resultados

A medida que el proyecto ha ido avanzando, se ha conseguido una optimización de la metodología de trabajo y del flujo de trabajo empleado que nos permite afirmar que, hasta la fecha, los resultados de la actuación están siendo muy satisfactorios. Además, teniendo en cuenta las dudas que teníamos antes de comenzar, debido a la novedad y originalidad de este proyecto, pueden considerarse incluso como espectaculares.

A continuación se presentan algunos números que servirán para hacerse una idea de la magnitud del proyecto.

- hasta el momento, el número de equipos entregados es de 100
- el número de equipos recogidos alcanza en la actualidad los 500, contando ordenadores, impresoras, monitores, etc.
- en promedio, se tarda menos de 3 días en recoger un equipo, reciclarlo y ponerlo a disposición de los solicitantes
- el número de usuarios distintos que han accedido a la web de Reciclaje, supera los 3000
- el grado de participación de las organizaciones y entidades turolenses, así como de los particulares, desde nuestro punto de vista es muy bueno
- la demanda de equipos reciclados, es grande, pero no mayor que la oferta disponible
- se han realizado actividades dirigidas a la Asistencia Técnica y el análisis del estado de todas las asociaciones que de momento han recibido algún equipo (un total de 25 hasta el momento)
- se ha ampliado el ámbito de actuación de la ciudad de Teruel a toda la provincia

5. Conclusiones

Esta línea pretende alcanzar varios objetivos, en primer lugar que los alumnos participantes se preparen para su entrada en el mercado laboral, por otro lado, asistir a las diferentes asociaciones en el proceso de implantación del uso de las nuevas tecnologías, también es muy importante presentar el uso del software libre como una buena alternativa al uso de otro tipo de programas

o a la piratería, y finalmente poner a disposición de los ciudadanos los equipos que han quedado obsoletos para algunas entidades, pero que pueden ser utilizados todavía, permitiendo acercar la Tecnología a toda la ciudadanía.

Teniendo en cuenta que el número de habitantes de la ciudad de Teruel ronda los 35.000, y que en toda la provincia hay aproximadamente unos 143.000, pensamos que la marcha del proyecto está siendo muy satisfactoria y que los números obtenidos hasta la fecha son muy buenos, por lo que debemos intentar que dicha actuación pueda continuar en los próximos años permitiendo que participen más alumnos de la Universidad de forma que se siga acercando a éstos una experiencia profesional.

Agradecimientos

Este proyecto está financiado por el contrato OTRI 2006/0302. Agradecemos la colaboración a Antonio Fandos y José Manuel Martín de la Oficina Técnica de Teruel Digital, a los miembros de la corporación del Excelentísimo Ayuntamiento de Teruel, especialmente a Juan Villalba concejal encargado entre otras cosas, del proyecto Teruel Ciudad Digital. Agradecemos también la colaboración directa, y su gran trabajo a los Becarios, Alberto Molina y Mariano Utrillas.

Referencias

[1] COBIT. http://www.isaca.org/cobit/
[2] Métrica Versión 3. http://www.csi.map.es/csi/metrica3/
[3] Reciclaje de Equipos Informáticos. http://www.reciclaje.terueldigital.es
[4] Teruel Digital. http://www.terueldigital.es

Informática en otras carreras

Desarrollo de docencia universitaria de informática documental con herramientas de software libre

Jesús Tramullas Saz*, Piedad Garrido Picazo**
*Dpto. de Ciencias de la Documentación
Universidad de Zaragoza
Pedro Cerbuna 12, 50009 Zaragoza
**Dpto. de Informática e Ingeniería de Sistemas
Universidad de Zaragoza
Ciudad Escolar s/n, Teruel
tramullas@unizar.es, piedad@unizar.es

Resumen

La informática documental es una especialización de la informática, que se ocupa del tratamiento y gestión de la información recogida en documentos digitales. Este campo ha sido dominado durante largo tiempo por software privativo, lo que se reflejaba en las prácticas de las asignaturas. La creciente disponibilidad de herramientas de software libre, capaces de dar soporte a los procesos y tareas de la informática documental ha hecho posible la integración de las mismas en el proceso de aprendizaje. En este trabajo se detalla la integración de estas herramientas en el contexto de la adaptación de una asignatura especializada al Espacio Europeo de Educación Superior.

1. Introducción

La enseñanza universitaria de los métodos y técnicas de tratamiento y gestión de la información documental se organiza en las titulaciones Diplomatura en Biblioteconomía y Documentación y Licenciatura en Documentación, que se ven complementadas con los correspondientes programas de doctorado. Esta organización va a variar a causa del desarrollo del Espacio Europeo de Educación Superior, que estructura la enseñanza de la disciplina en grado, máster y doctorado.

La gestión de la información tiene como parte esencial la aplicación de tecnologías de la información y la documentación, de tal forma que no pueden entenderse una sin la otra. Ambas han dado forma a la disciplina, que ha recibido denominaciones como Documentación Automatizada o Informática documental. Las directrices establecidas por los Reales Decretos que regulan estos estudios así lo contemplan, incluyendo específicamente las Tecnologías de la Información y las Bases de Datos. El libro blanco sobre el Título de Grado en Documentación, punto de partida para los futuros planes de estudio, incide en los contenidos digitales, y en los sistemas y herramientas para su creación, implementación y gestión.

Cada universidad ha establecido en sus planes de estudio la forma en que se organizan los créditos correspondientes, generalmente en asignaturas anuales o cuatrimestrales. La variedad de títulos y de contenidos de las mismas es elevada, aunque puede rastrearse un núcleo común en todas ellas, cuyas materias corresponden a:

- Bases de datos documentales y bibliográficas
- Edición electrónica
- Técnicas y herramientas de recuperación de información
- Sistemas electrónicos de gestión documental
- Bibliotecas digitales y repositorios institucionales
- Sistemas de informatización de bibliotecas y archivos
- Servicios de información documental en Internet: motores de búsqueda, esquemas de metadatos...

Estos contenidos básicos pueden verse complementados mediante asignaturas optativas, que inciden en diferentes aspectos especializados.

2. La asignatura "Sistemas electrónicos de tratamiento documental"

La asignatura *Sistemas electrónicos de tratamiento documental* se incardina en el contexto indicado en el apartado 1. Se trata de una

asignatura troncal de seis créditos (dos teóricos y cuatro prácticos), que se imparte durante el primer cuatrimestre del segundo curso de la Diplomatura en Biblioteconomía y Documentación de la Universidad de Zaragoza.

El contenido teórico de la misma viene dado por las directrices generales de la titulación, por una parte, y por la organización interna de la docencia en la Diplomatura, lo que ha facilitado la coordinación de contenidos y actividades con el resto de asignaturas. El contenido práctico, sin embargo, venía determinado en cada momento por varios factores:

1. La orientación, demandada por el mercado, a formar usuarios finales de las herramientas informáticas
2. El dominio del ámbito de trabajo informativo-documental por parte de productos de compañías de software propietario o privativo
3. La disponibilidad gratuita, o por convenio, para la docencia de las herramientas de software privativo
4. La escasez de documentación de libre acceso para esas herramientas

La actividad práctica se llevaba a cabo en laboratorio, en el cual los alumnos desarrollaban prácticas individuales guiadas con las diferentes herramientas de software privativo.

Evidentemente, todos estos factores afectaban a la formación práctica que los alumnos podían desarrollar, a causa de la variable disponibilidad de las herramientas necesarias para ello. Si bien la enseñanza impartida hasta el momento había asegurado la formación del alumno con la calidad exigible, esta dependencia de herramietnas privativas de tereceros podía hipotecar la actividad docente en un entorno cada vez más cambiante y dinámico.

3. Adaptación al Espacio Europeo de Educación Superior

El Espacio Europeo de Educación Superior (EEES) propone un cambio efectivo en la docencia universitaria, primando el trabajo del alumno y su proceso de aprendizaje, frente a modelos docentes que podrían calificarse como más tradicionales. Son numerosas las publicaciones y trabajos que recogen las características y enfoques del EEES, por lo que no es necesario insistir sobre estas cuestiones en este trabajo.

En este contexto, y en el marco del proceso de adaptación al EEES que se está llevando a cabo en la Diplomatura en Biblioteconomía y Documentación de la Universidad de Zaragoza, se ha llevado a cabo un rediseño de la asignatura *Sistemas electrónicos de tratamiento documental.* Para ello, ha sido necesario replantear los objetivos de la misma, así como las competencias y habilidades que los alumnos deben desarrollar durante su desempeño. En consecuencia, se ha considerado pertinente llevar a cabo un completo proceso de revisión, planificación y adecuación de la actividad práctica, de forma que no se viese constreñida por los vaivenes y variaciones del mercado de software privativo y de sus productos. Inmersos en este proceso, también se rediseñó la propia actividad práctica, mediante la integración y uso de modelos de trabajo del alumno no contemplados hasta este momento.

La definición de competencias y habilidades trajo consigo un cambio del enfoque de la asignatura, que supone también cambios estratégicos en la concepción de las actividades prácticas que se llevan a cabo con herramientas informáticas. En primer lugar, se han establecido cuatro principios generales:

1. La adquisición de competencias y habilidades no debe verse limitada por el contexto comercial en el que se desarrolle la actividad profesional
2. La disponibilidad de herramientas software no debe restringir la adquisición de competencias y habilidades
3. El alumno debe poder desarrollar su actividad de aprendizaje en diferentes contextos: aula, laboratorio, entorno doméstico, otros entornos informáticos...
4. El alumno debe poder localizar y acceder a documentación externa suficiente para llevar a cabo su propio proceso de aprendizaje

El respeto de los principios generales lleva, a limitar la excesiva influencia del entonro profesional sobre la docencia, y a superar la orientación meramente instrumental de este tipo de asignaturas. Si bien no excluye la integración de herramientas de software privativo, la

dificultad para que éstas puedan cumplir los principios tercero y cuarto limita realmente su uso potencial. Puede argumentarse que la exclusión de estas herramientas privativas, adecuadas a lo que está demandando el mercado, es, en realidad, un perjuicio para el alumno, al alejar su formación de las demandas del mercado laboral. La respuesta a esta cuestión clave se abordará en el apartado 4.

El segundo aspecto de adaptación, una vez definidos los principios generales, ha sido el cambio de modelo de las actividades prácticas. Como se ha indicado anteriormente, éstas se llevaban a cabo principalmente en laboratorio. Sin embargo, el EEES propone un cambio en la actividad de aprendizaje, insistiendo en la actividad propia del alumno, independientemente de su localización y disponibilidad espacial y temporal. Además, incluye una clara orientación al desarrollo de habilidades de aprendizaje autónomo y trabajo en equipo y en colaboración.

En consecuencia, y aprovechando las posibilidades que ofrece el cambio en las herramientas informáticas, se ha optado por cambiar el modelo de práctica presencial e individual en laboratorio, or el modelo de trabajo en equipo en colaboración, en diferentes localizaciones y con flexibilidad horaria. De la misma forma, se ha producido también el paso de la actividad de práctica guiada hacia el modelo basado en el desarrollo de proyectos. Como se analizará en el apartado 4, esto no hubiese sido posible sin el cambio estratégico a herramientas de software libre. Las posibilidades que ofrece este tipo de cambio ya han sido señaladas por Faber [1].

Por último, debe destacarse que el cambio y adaptación también ha afectado a los mecanismos de evaluación. Los alumnos desarrollan un proceso personal de evaluación de su propio desempeño en los proyectos, así como del resto de sus compañeros, de las herramientas utilizadas y de la documentación de soporte disponible en cada caso.

4. Software libre para tratamiento y gestión de información documental

Como se ha señalado en el apartado anterior, la cuestión clave es la disponibilidad de herramientas de software no privativo que ofrezcan las prestaciones técnicas necesarias para que los alumnos puedan desarrollar las competencias y habilidades establecidas para la asignatura. En la bibliografía especializada pueden encontrarse trabajos que recogen experiencias en diferentes campos de aplicación de la informática [2] La experiencia docente e investigadora previa sugería que existían todas las herramientas necesarias para ello dentro del campo del software libre [3].

En consecuencia, se llevó a cabo un estudio de campo para dilucidar la cuestión. Para ello, se exploraron sistemáticamente los principales servidores para proyectos de software libre, se analizaron los contenidos de las sedes web de referencia sobre software libre aplicado al tratamiento y recuperación de información en bibliotecas y servicios de información y documentación, y se revisó detenidamente la bibliografía especializada sobre este tema [4]. El resultado del estudio ha identificado herramientas de software libre, especializadas en procesos y tareas que se llevan a cabo en los servicios de información, que pueden agruparse en las siguientes categorías:

1. Bases de datos bibliográficas: sistemas de bases de datos para la construcción de repertorios de bibliografía, tanto en soportes tradicionales como en entornos digitales. Ejemplos: Connotea (www.connotea.org), WIKINDEX (wikindx.sourceforge.net/), RefBase (refbase.sourceforge.net),
1. Motores de indización y recuperación: herramientas para la extracción de información textual de colecciones de documentos, y para su indización automática, así como para la creación de interfaces de recuperación de información. Ejemplos: Lucene/Nutch (lucene.apache.org), Xapian (www.xapian.org), SWISH-E (swish-e.org)
2. Sistemas de gestión de documentos: sistemas completos para la definición de ciclos de vida de documentos, así como flujos de trabajo, control de versiones y de acceso, etc. Ejemplos: Alfresco Content Management (www.alfrescosoftware.com), KnowledgeTree (www.knowledgetree.com)
3. Sistemas de informatización de bibliotecas: herramientas de software libre que dan soporte a todas las funciones y procesos técnicos que se llevan a cabo en una biblioteca, como

catalogación, circulación, gestión administrativa, etc. Ejemplos: Koha (www.koha.org), PMB (www.sigb.net)

4. Directorios de recursos digitales: permiten crear portales de acceso a los recursos de información digital, de todo tipo, que pueda necesitar una organización. Ponen especial énfasis en las tareas de administración y control de metadatos, ya que suelen usar Dublin Core. Ejemplos: iVIA (ivia.ucr.edu) Scout Portal (scout.wisc.edu/Projects/SPT/)
5. Bibliotecas digitales: aplicaciones para la construcción y mantenimiento de colecciones de documentos completos (textuales, gráficos, etc), organizados y estructurados alrededor del concepto de "colección". Dan prioridad al acceso de los usuarios. Ejemplos: Greenstone (www.greenstone.org)
6. Repositorios digitales: aplicaciones para el desarrollo de repositorios de documentos digitales, que contengan el acervo científico de una institución. Dan prioridad a los procesos de publicación. Ejemplos: Eprints (eprints.org), DSpace (dspace.org)

Junto a todos los anteriores cabría añadir las herramientas para edición de documentos digitales y para gestión de metadatos. Dado que las posibles elecciones son muy numerosas, no se han incluido como una herramienta específica de gestión de información documental. Igualmente, no se han considerado los sistemas de gestión de contenidos o de portales de tipo generalista, ya que se entiende, precisamente, que son herramientas básicas de gestión de información, y que su utilidad dependerá, precisamente, de las características del proyecto en el que se encuadre su utilización. No debe dejar de tenerse en consideración que la mayor parte de los procesos que tienen lugar en cualquier sistema de gestión de contenidos corresponden, en un setenta u ochenta por ciento, a tareas básicas de tratamiento y gestión de información.

Otro factor de suma importancia, inherente a las herramientas de software, es la utilización de estándares. Todas estas herramientas implementan estándares, generalmente públicos y abiertos, como XML, Z39.50, MARC, Dublin Core, OAI, etc. En el campo de la información y la documentación los estándares tienen una máxima importancia, hasta el punto de tener su propio comité específico en ISO y en las diferentes agencias nacionales de normalización, como AENOR. Los alumnos de información y documentación deben tener un claro conocimiento de las normas y estándares que afectan a su actividad, y la utilización de herramientas de software libre refuerza esta habilidad. Es este sentido destaca el recientemente aprobado estándar para documentos ofimáticos OpenDocument (ISO 26300:2006), que establece, como su propio nombre indica, el único estándar internacional para los documentos más comunes en los entorno digitales, como son los resultantes de la actividad ofimática de las organizaciones. En consecuencia, la adscripción a estándares es un requerimiento que se establece para las herramientas software que se vayan a utilizar. Igualmente, es el formato ofimático que se exige a los estudiantes para la presentación e intercambio de trabajos escritos, informes, formularios de evaluación, etc. Las herramientas seleccionadas incorporan, en sus funcionalidades, los estándares para el tratamiento y recuperación de la información señalados aneriormente en este párrafo, lo que asegura su conocimiento y utilización por parte del estudiante.

Como se ha indicado en el apartado 3, cabe discutir si este enfoque limita las posibilidades de aprendizaje del estudiante. En primer lugar, el cambio hacia un modelo basado en el aprendizaje activo de habilidades permite que el estudiante adquiera las pautas básicas para saber plantear y desarrollar actividades frente a cualquier herramienta con la que se encuentre en su actividad profesional. En segundo lugar, la libre disponibilidad de las herramientas facilita la instalación y replicación de plataformas de actividad en cualquier máquina, sin estar obligatoriamente localizada en un aula universitaria, como era el caso con el software privativo. Ambas cuestiones son complementadas por los procesos de documentación que el estudiante debe llevar a cabo para conocer y utilizar las herramientas libres, y que inciden en su capacidad de autoaprendizaje.

El desarrrollo de las actividades mediante proyectos en colaboración, en entornos distribuidos, también es posible gracias a la utilización de herramientas libres, ya que las características de las herramientas privativas

obligaban al planteamiento individual de las prácticas.

Por último, debe considerarse que la mayoría de los alumnos disponen, particularmente, de plataformas Windows, de la misma forma que las salas de usuarios de la Universidad de Zaragoza ofrecen exclusivamente este tipo de sistemas operativos. Esta incomprensible restricción impone que las herramientas que sean seleccionadas pueden instalarse sin complicaciones en plataformas GNU/Linux, Windows y Mac OS X. De esta forma se asegura el acceso de los alumnos a los medios necesarios.

5. Integración de software libre como herramienta de aprendizaje

El desarrollo de la asignatura *Sistemas electrónicos de tratamiento documental* se lleva cabo mediante la integración de herramientas de software libre, que dan soporte al proceso de aprendizaje de los estudiantes. Las herramientas se han organizado en tres grupos, cada uno de los tiene diferentes objetivos y tareas, en el marco de la planificación global de la docencia.

- Soporte ofimático: actividades en las cuales los alumnos llevan a cabo la creación, modificación e intercambio de información mediante documentos ofimáticos. Todos estos documentos deben cumplir el estándar ISO 26300 OpenDocument. El paquete de software elegido para ello es *OpenOffice*, dada su disponibilidad para plataformas GNU/Linux, Windows y Mac OS X.
- Portal de la asignatura: la asignatura pone a disposición de los estudiantes un portal propio, independiente de los sistemas de aula virtual de la universidad. Para ello utiliza un sistema de gestión de contenidos libre, *Drupal* (www.drupal.org, fig.1). Este servicio pone a su disposición foro de discusión, descarga de materiales docentes, noticias, encuestas y sindicación mediante RSS. Las relaciones entre software libre y lo que se ha dado llamado en llamar open courseware pueden verse en [5].

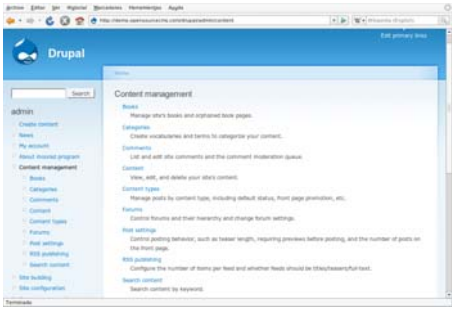


Figura 1. Portal sobre Drupal

- Sistemas de tratamiento documental: son los encargados de soportar el núcleo del proceso de adquisición de competencias y habilidades, ya que son las herramientas con las cuales desarrollarán los proyectos que se les requieran. Los dos paquetes seleccionados son *KnowledgeTree* (para gestión documental, www.knowledgetree.com), y *Greenstone* (para bibliotecas y repositorios digitales, www.greenstone.org). Los paquetes permiten gestionar los documentos digitales desde la perspectiva de tratamiento y recuperación de información documental, incluyen motores de indización y recuperación, y permiten trabajar con diferentes esquemas de metadatos.

Si bien todas las herramientas, como se ha indicado previamente, pueden instalarse en diferentes plataformas, el portal de la asignatura es instalado y gestionado por los docentes en un servidor propio (disponible en imhotep.unizar.es), sobre la distribución GNU/Linux Ubuntu 7.04 Feisty Fawn. En el mismo se dispone también de versiones de producción de *KnowledgeTree* y de *Greenstone*, para que los alumnos puedan comprobar el funcionamiento de los productos resultantes de su trabajo en un entorno real. El servidor y las herramientas van siendo actualizadas conforme aparecen nuevas versiones.

Una vez seleccionadas las herramientas que van a ser utilizadas en el proceso de aprendizaje, es necesario determinar la forma en que las mismas se van a incorporar a la actividad del alumno. Como se ha indicado previamente (apartado 3), el modelo seleccionado ha sido el desarrollo de proyectos mediante trabajo en grupo

en colaboración. Los proyectos que los alumnos llevan a cabo son los siguientes:

1. Proyecto gestión documental: el proyecto debe definir las características, el ciclo de vida y el flujo de trabajo de un conjunto de documentos digitales, incluyendo control de versiones y de modificaciones, asignación de esquemas de metadatos, e indización del texto completo. Para este proyecto se usa *KnowledgeTree* (fig. 2).

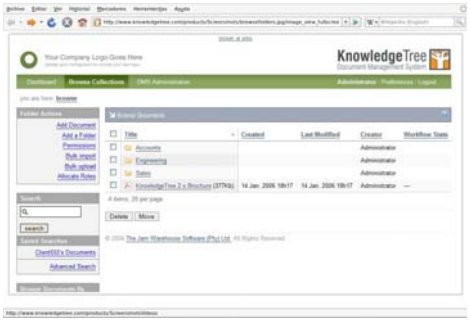


Figura 2. Servidor KnowledgeTree

2. Proyecto colección de documentos digitales: el proyecto debe crear una colección de documentos digitales, en diferentes formatos, incluyendo la indización a texto completo de su contenido, así como la asignación de metadatos y la publicación de la colección en internet para su aprovechamiento por usuarios finales. Para este proyecto se usa *Greenstone* (fig. 3).
3. Proyecto de gestión de contenidos: es el único proyecto que se lleva cabo de manera individual. Para desarrollar las habilidades básicas de gestión de contenidos en entornos digitales, el alumno debe crear, configurar y mantener un blog. A este fin, debe obtener y utilizar una cuenta en cualquiera de los servidores gratuitos de blog disponibles en internet (blogger, blogia, lacoctelera, etc.).

Figura 3. Servidor Greenstone

El planteamiento práctico de la asignatura, como puede apreciarse, busca que los estudiantes desarrollen las competencias y habilidades de la disciplina con herramientas que les obligan a desarrollar sus propios procesos de planificación, documentación e implementación. Los diferentes proyectos tienen como punto de partida un guión de desarrollo y una clase introductoria a la herramienta seleccionada, y a partir de este momento los procesos de documentación, aprendizaje, instalación y configuración de la herramienta quedan bajo la responsabilidad del alumno. Se ha preparado documentación de referencia especialmente orientada a este tipo de usuario [6]. Una vez operativa la aplicación, se inicia la implementación del proyecto de información y documentación digital en sí mismo. Durante todo el ciclo de vida del proyecto los grupos de trabajo disponen de tutorías de grupo y de soporte bajo demanda por parte de los docentes responsables de la asignatura.

6. Conclusiones

El cambio de modelo de desarrollo que ha supuesto el movimiento de software libre influye en cada vez más actividades, y plantea desafíos y soluciones de verdadero interés [7]. La planificación docente expuesta en los apartados anteriores ha sido puesta en marcha y validada durante el curso 2006-2007. Los resultados obtenidos permiten establecer las siguientes conclusiones:

- Las herramientas de software libre especializadas disponibles para la gestión de

información documental cumplen todos los requerimientos necesarios para su uso docente
- Estas herramientas cubren las competencias y habilidades prácticas que los estudiantes deben adquirir
- El uso de las mismas en proyectos refuerza las actitudes y aptitudes de autoaprendizaje deseables para los futuros profesionales de gestión de información y documentación.
- Los problemas encontrados en la aplicación del modelo expuesto se centran en la falta de documentación orientada al usuario final, y en la carencia de habilidades de autoaprendizaje por parte de los estudiantes.

La amplia varidad de herramientas disponibles pueden hacer posible extender la utilización del software libre a un campo más amplio de actuación. La creación de motores especializados de búsqueda para intranet/internet, o la recopilación de recursos de información digital de valor añadido, mediante directorios, o la creación de bases de datos bibliográfícas científicas y técnicas, son otras áreas en las que pueden integrarse perfectamente. Además, debe considerarse que en breve plazo las herramientas de software libre especialmente diseñadas para desarrollar el web semántico, en el cual predominan los métodos y técnicas de gestión de la información, potenciarán aún más las posibilidades de formación de especialistas en información y documentación digital, ofreciendo un mayor potencial que las opciones de software privativo.

Referencias

[1] Faber, B.D. "Educational models and open source: resisting the propietary university." *Proceedings of the 20th annual international conference on Computer documentation*, 2002, 31-38.

[2] Zaritski, R.M. "Using open source software for scientific simulations, data visualization, and publishing " *Journal of Computing Sciences in Colleges*, 19, 2, 2003, 218-222.

[3] Tramullas, J. "Software libre para gestión de recursos de información digital." *Encuentro Internacional sobre Conocimiento Libre, II Conferencia Internacional de Software Libre*, Badajoz, 2006, 346-356.

[4] Chawner, B. *Open Source Software and Libraries Bibliography*. 2006. (disponible en http://www.vuw.ac.nz/staff/brenda_chawner/biblio.html)

[5] Baldi, S., Heier, H., Mehler-Bicher, A. "Open Courseware and Open Source Software." *Communications of the ACM*, 2003, 46, 9, 105-107.

[6] Tramullas, J. Garrido, P. (coords.), *Software libre para servicios de información digital*. Madrid: Prentice Hall, 2006.

[7] Wolf, M.J., Bowyer, K., Gotterbam, D., Miller, K. "Open source software: intellectual challenges to the status quo." *Proceedings of the 33rd SIGCSE technical symposium on computer science education*, 2002, 317-318.

¿Aprendizaje Basado en Proyectos? ¡Pero si mi carrera no es técnica!

Francisco J. Gallego Durán, Faraón Llorens Largo
Dpto. de Ciencia de la Computación e Inteligencia Artificial
Universidad de Alicante
Campus de San Vicente, Aptdo. Correos 99
03080, San Vicente del Raspeig (Alicante)
{fgallego,faraon}@dccia.ua.es

Resumen

¿Es posible utilizar Aprendizaje Basado en Proyectos para enseñar informática a alumnos de carreras no-técnicas? ¿Se puede ver en acción los beneficios de esta metodología en un ambiente con alumnos completamente acostumbrados a las metodologías más clásicas? Estas dos preguntas iniciales y decenas de preguntas consecuentes nos llevaron a principios del curso 2005/2006 a replantearnos el diseño y la metodología docente dutilizada en la asignatura de Informática para Juristas. Esta asignatura, optativa en la Licenciatura en Derecho, había sido impartida hasta entonces en el formato magistral más clásico. Pensando en su reforma oteamos la posibilidad de encaminarla a despertar habilidades dormidas en los alumnos; habilidades dormidas para aquellos que no habían hecho uso de ellas, merced al tipo de exigencias de sus estudios. Sólo nos faltaba un ingrediente, ¿Cómo diseñar un proyecto que sirviera a alumnos no-técnicos para aprender informática? ¿Y por qué no combinarlo con Aprendizaje a Través de la Escritura? «¡Claro! ¡Eso es! ¡Una revista de informática!». Decidimos que ese año el proyecto de los alumnos consistiría en crear el primer ejemplar de la revista Informática para Juristas.

En este artículo describimos cómo combinamos las metodologías de Aprendizaje Basado en Proyectos y Aprendizaje a Través de la Escritura para guiar a nuestros alumnos no-técnicos en su periplo a través de la tecnología. Así mismo, especificamos en detalle el plan de la asignatura, los trabajos propuestos y las sesiones realizadas. Finalmente, mostramos los resultados de las encuestas de opinión realizadas y unas conclusiones que pueden ser de utilidad para quienes deseen repetir esta interesante experiencia, no sólo en Licenciatura en Derecho, sino también en cualquier carrera no-técnica.

1. Introducción

Se podría decir que la experiencia que presentamos en este artículo es el fruto del trabajo conjunto de muchos profesores durante innumerables horas de estudio, análisis, diseño metodológico y elaboración de materiales, que un año dieron lugar a una metodología muy estudiada que se puso en práctica con el mayor de los éxitos. No sería esto incierto si nos estuvieramos refiriendo a todos los docentes cuyas experiencias previas han forjado las ideas que hemos puesto en práctica. Sin embargo, lo que describiremos a continuación es el fruto de una idea, una simple idea que teníamos en mente, y un deseo: mejorar la calidad de la docencia y el nivel de satisfacción de nuestros estudiantes.

Muchos docentes piensan que las metodologías de Aprendizaje Basado en Proyectos (ABP, [1][2][3][4]) o Aprendizaje a Través de la Escritura (ATE, [5][6]) requieren muchísimo esfuerzo de preparación y tan sólo les son útiles a los intrépidos enseñadores compulsivos que se desviven por innovar y, tal vez, por estar a la última. Hay una única parte de verdad en esto: hace falta valor y arrojo, puesto que es necesario enfrentarse constantemente a situaciones imprevistas. Sin embargo, tal como mostramos en este artículo, no es necesario tener una planificación perfecta y absoluta que contemple hasta lo inesperado o, tal vez, lo incontemplable. Sólo hacen falta dos premisas: valor y flexibilidad.

Nuestra experiencia fue desarrollada durante el curso académico 2005/2006, cuando tomamos la decisión de abordar nuestro particular reto docente: *Informática para Juristas*, una asignatura impartida como optativa en los planes de la licenciatura en derecho. El atractivo fundamental de darle un nuevo aire a esta asignatura era muy sencillo. Ya conocíamos las metodología de ABP

y ATE, pero tan sólo las habíamos visto en práctica en cursos o asignaturas de contenido técnico. ¿Sería posible que sus beneficios fueran transferibles al ámbito de una carrera no-técnica?

Guiados por esta inquietud inicial y la intención de mejorar la experiencia de aprendizaje de los alumnos de Informática para Juristas, decidimos partir de cero, presentarles algo completamente nuevo y evaluar la calidad del mismo.

2. Nuevo planteamiento de la asignatura

Nuestra intención inicial era cambiar la metodología de la asignatura por ABP. Sin embargo, todas las referencias consultadas que hablaban de las maravillas de ABP y su aplicación se referían exclusivamente a carreras técnicas. El problema principal con estudiantes de otras titulaciones no técnicas era encontrar un proyecto en el que enmarcar los contenidos de la asignatura. Los proyectos técnicos no son útiles a este respecto, puesto que no se presupone ningún conocimiento a los alumnos. Es decir, no era posible pedirles que desarrollaran una aplicación informática o similares. Por otra parte, un proyecto de desarrollo o utilización de un sistema informático tampoco parecía que les fuera a aportar conocimientos útiles para ellos.

Tras pensar un poco recordamos una propuesta que nos podía ser extremadamente útil: ATE. ATE consiste en pedirles a los alumnos que escriban acerca de los temas de la asignatura, lo que les hará aprender sobre ellos. La principal ventaja de este método radica en las propias necesidades de la expresión escrita: cuando escribimos tenemos que aclarar y organizar nuestras ideas para que puedan ser comprensibles por el lector.

Enseguida acertamos a ver ATE como la piedra angular de la asignatura, puesto que se presumía muy interesante el uso de la escritura como herramienta de aprendizaje con futuros juristas. Sin embargo, esto no terminaba de resolver nuestro problema inicial: Seguíamos sin tener un proyecto para guiar el aprendizaje. ¿Cómo combinar ABP y escritura en un proyecto que enseñe los conocimientos informáticos necesarios a alumnos de derecho? Poco después de plantearnos la pregunta, vimos la solución encima de nuestro escritorio: ¡Claro! ¡Una revista!

Así pues, decidimos formar grupos de alumnos que iban a convertirse en eventuales editoriales, cuyo proyecto no era otro que producir el primer ejemplar de la revista *Informática para Juristas*. De esta forma les ibamos a poder hacer escribir artículos, reportajes, anuncios, secciones, etc. sobre cada uno de los temas de la asignatura. Esto les permitiría aprender sobre los temas desarrollando, además, habilidades de expresión escrita muy útiles para alumnos de derecho.

2.1. Nuestros objetivos

Antes de ponernos el mono y el casco y comenzar las reformas, redactamos nuestra lista de principios y objetivos básicos, muchos de los cuales provienen de las interesantes sugerencias encontradas en [7] y [8]:

1. Limitar al mínimo las clases magistrales, guiándolo todo a través del trabajo de los alumnos.
2. Plantear actividades que fomenten la curiosidad de los alumnos y su motivación.
3. Tener un sistema de evaluación continua.
4. El sistema de evaluación, además, debía estar muy claro desde un principio.
5. Los profesores debíamos implicarnos en los proyectos de los alumnos para guiarles lo mejor posible.
6. Así mismo, debíamos fomentar el pensamiento crítico, no imponiendo en ningún caso nuestras ideologías.
7. No olvidar que el principal objetivo del alumno siempre iba a ser el aprobado.
8. Valorar *todos* los trabajos que hicieran los alumnos.
9. Ser flexibles ante la respuesta de los alumnos, en todo momento.
10. Fomentar la investigación autónoma, progresivamente.

Con estos 10 principios básicos en mente, hicimos un esbozo rápido de la asignatura y nos lanzamos a la aventura.

2.2. Transformando la asignatura

Informática para Juristas era una asignatura en formato clásico absoluto: clases teóricas magistrales, clases prácticas con 3 trabajos prácticos a entregar y un examen final teórico tipo test (figura 1).

EVALUACIÓN CLÁSICA	
Teoría (50%)	**Prácticas (50%)**
Examen Final (Tipo test)	3 Prácticas en pareja (Word, Excel y Access)

Figura 1. Evaluación Clásica

Para empezar, comenzamos analizando los contenidos teóricos. Los bloques temáticos se muestran en la tabla 2. Decidimos respetar los contenidos de la asignatura por lo que respecta a los bloques temáticos, pero no así por lo que respecta a los contenidos de cada bloque, siguiendo nuestro 2º principio. Es decir, plantearíamos a los alumnos escribir sobre las cosas que más les interesasen de estos temas, de una forma progresiva.

Temario Informática para Juristas
1. La Informática en el ámbito jurídico
2. Descripción de un sistema informático
3. La Informática en la Oficina Jurídica
4. Bases de datos Jurídicas
5. Internet
6. Informática y derecho. Temas de actualidad.

Tabla 1. Bloques temáticos de la asignatura.

Disponíamos de 14 semanas efectivas de clase, con 1 hora en aula de teoría y 2 horas en aula de prácticas. Decidimos dedicar la hora en aula de teoría para presentar brevemente los contenidos y trabajos a realizar en el siguiente tema, y poner en común los trabajos realizados en el tema anterior. Por su parte, las 2 horas de prácticas servirían para que los grupos desarrollaran sus proyectos en aulas con ordenadores bajo la supervisión del profesor.

A este esquema básico faltaba añadirle la evaluación continua. Para esto consideramos la posibilidad de ir revisando los proyectos. Sin embargo, la carga de trabajo que podrían tener los profesores revisando proyectos trabajo por trabajo sería muy elevada, pudiendo desbordarnos y provocar retrasos o ineficiencia en el sistema. Por otra parte, no parece adecuado para los alumnos que toda su evaluación se base en revisiones cruzadas entre ellos. Así pues, decidimos hacer una mezcla. A lo largo del curso se realizarían 4 revisiones de la carpeta de proyecto de cada grupo, trabajo por trabajo. Las 3 primeras revisiones serían cruzadas cada 2 temas, es decir, realizadas por los miembros de otro grupo de alumnos. Por su parte, la revisión final la realizaría el profesor.

Con el fin de asegurar una adecuada calidad de las revisiones a realizar, sería el profesor el encargado de revisar las revisiones. Es decir, comprobar que los comentarios son útiles y respetuosos y las notas de los trabajos coherentes. Además, el profesor emitiría una evaluación de cada revisión en forma de una pequeña nota en el intervalo [-0.25, +0.25] que se sumaría a la nota del grupo que hubiera realizado dicha revisión.

Teniendo siempre en mente que estamos hablando de una asignatura optativa, y que la principal preocupación de los alumnos es el aprobado, establecimos una norma para garantizar la seguridad de los alumnos: el 50% de la nota se coseguiría diréctamente si la entrega final (la revista) incluía todos los trabajos obligatorios exigidos durante el curso. Por supuesto, todos debían ser trabajos propiamente dichos. No sería admisible introducir texto de relleno para simular un trabajo que no hubiera sido hecho.

Finalmente, para complementar todo lo anterior, decidimos otorgar unos premios especiales para incentivar el trabajo de los grupos. Por ejemplo, el grupo que mejores revisiones hiciera obtendría +0,25 sobre la nota final. El resto de extras los detallaremos más adelante.

De igual forma, y dado que los estatutos de la universidad nos obligarían a realizar una convocatoria de examen final, decidimos otorgar hasta +1,25 extra (en caso de sacar un 10) por hacer voluntariamente el examen final. Por supuesto, este examen final también servía para aprobar a aquellos que no pudieran seguir la dinámica en grupo de la asignatura, por motivos de trabajo o similares.

3. La metodología en detalle

En la sección anterior hemos explicado cómo surgieron las ideas y cómo dimos forma a un primer borrador de metodología a seguir. El resto del trabajo lo hicimos sobre la marcha, conforme la asignatura lo fue pidiendo. En esta sección analizaremos algunos pormenores interesantes de

la metodología tal como se desarrollaron en el aula. Comenzaremos por el análisis del sistema de evaluación tal cual se aplicó a los alumnos

Tal como se puede apreciar en la figura 2, la evaluación estaba dividida en bloques. Los bloques se encuentran detallados y explicados en la tabla 2.

Bloques de evaluación obligatorios	
Bloque	**% nota**
Mínimo exigido	50%
Revisiones de clase	20%
Revisión Final	30%
Extras de grupo	
Extra	**incentivo**
Revisar carpetas de otros grupos	[-0.25,0.25]
Exposicion al azar en clase	+0.10
Grupo que mejor revisa	+0.30
Grp. mejor califica exposiciones	+0.10
Trabajos optativos	[0.00,0.25]
Extras individuales	
Extra	**incentivo**
Exposicion volutaria en clase	+0.25
Competición "Knowledge Quiz"	[0.00,0.25]
Examen Final	[0.00,1.25]

Tabla 2. Detalle bloques de evaluación.

Nuestra política principal consiste en premiar siempre más que penalizar. La intención es la de incentivar al alumno de la misma forma que se incentiva a un profesional en cualquier ámbito: si se le pagan los extras aparte, el profesional trabaja más.

Con el sistema de evaluación ya detallado, pasamos a explicar detenidamente algunos de sus aspectos más interesantes, alguno que pudiera resultar negativo y, sobre todo, cómo se vivió la experiencia en el aula.

3.1. Revisiones cruzadas

Las revisiones de carpetas de proyecto son siempre un tema controvertido a la hora de aplicar metodologías como ABP. En general, la mayoría de los profesores se muestran reacios a dejar que los alumnos se evalúen entre ellos.

En la asignatura de Informática para Juristas, las revisiones de carpetas se realizaban como sigue:

- El grupo revisor recibía una carpeta con los últimos trabajos exigidos de otro grupo, una hoja con los criterios de valoración, varias copias de un formulario para puntuar cada trabajo y un formulario para puntuar la carpeta del proyecto.
- El grupo revisor debía revisar 2 veces cada uno de los trabajos. Cada una de estas 2 revisiones, por un miembro distinto del grupo, obviamente.
- Finalmente, debían poner en común las revisiones de todos los trabajos y reunirse para establecer una puntuación global de la carpeta de proyecto.
- Las revisiones debían hacerse puntuando individualmente cada trabajo según 5 apartados (de 0 a 5 puntos por apartado): Interés del contenido, corrección ortográfica y gramatical, legibilidad, información extra añadida y presentación.
- Además, para cada apartado a revisar, el revisor debía escribir un comentario sobre su revisión comentando los puntos fuertes y débiles del trabajo revisado, en cuanto a ese apartado.

GRUPO	ALUMNO	MÍNIMO EXIGIDO	REVISIONES CLASE						REVISIÓN FINAL	EXTRAS GRUPO	EXTRAS INDIVIDUAL	TOTAL CLASE	NOTA EXAMEN TEORÍA	NOTA FINAL	
			1ª REVISIÓN		2ª REVISIÓN		3ª REVISIÓN								
			TEMAS 1,2	REVISIÓN	TEMAS 3,4	REVISIÓN	TEMAS 5,6	REVISIÓN							
ACT10	AMPC	+5,00	+0,40	+0,10	+0,40	+0,05	+0,40	+0,15	+1,79	+0,40	+0,02	8,71	N.P.	8,71	N
	IC										+0,01	8,70	N.P.	8,70	N
	MCGA										+0,26	8,95	N.P.	8,95	SB
	MADG										+0,02	8,71	N.P.	8,71	N
Grupo AKI	MICV	+4,75	+0,53	-0,05	+0,40	-0,15	+0,27	+0,10	+2,09	+0,20		8,14	N.P.	8,14	N
	KKB											8,14	N.P.	8,14	N
	MAGA											8,14	N.P.	8,14	N
	MTRF											8,14	N.P.	8,14	N

Figura 2. Extracto de la hoja de cálculo real utilizada para la evaluación de los alumnos. Los nombres de los alumnos reales han sido enmascarados para mantener su anonimato.

Las revisiones, en general, resultaban de gran ayuda a los alumnos. Por un lado, tenían una retroalimentación continua que les indicaba cómo de bien progresaban sus trabajos. Por otra parte, revisar trabajos de otras personas sobre un tema sobre el que ya habían escrito previamente reforzaba y ampliaba sus conocimientos.

Tras cada revisión, los profesores de la asignatura supervisábamos los resultados de la misma. El objetivo no era ajustar las notas, sino comprobar que los comentarios eran apropiados y no ofensivos. A los grupos se les entregaban las hojas de revisión filtradas por los profesores. Además, se les entregaba también una hoja con comentarios acerca de su revisión, indicando cuán adecuada había sido y otorgando por ella una puntuación extra, como se indicaba en la tabla 2.

La experiencia vivida en clase fue muy gratificante. Los alumnos se implicaban en las revisiones y consultaban mucho al respecto. Se apreciaba que todos eran conscientes de la importancia de las revisiones.

3.2. Clases de teoría

Al comienzo de cada clase de teoría, los grupos recibían un folleto con los trabajos de teoría que se les exigían para el tema siguiente y un folleto con la programación pormenorizada de lo que iban a hacer en esa hora (ver figura 3). Las programaciones incluían reuniones de expertos, reuniones de grupo y exposiciones como pilares principales, siguiendo la técnica del puzzle [9]. Dependiendo del tema, se incluía, además, una actividad extra como entrevistar a un miembro de otro grupo o escribir un cuento teórico en cadena.

Cada tema se encontraba dividido en partes para que los integrantes del grupo pudieran repartirse el trabajo adecuadamente. En las reuniones de expertos, todos los alumnos que habían tratado la misma parte del tema se juntaban y ponían en común sus experiencias para enriquecerse sobre el tema. Posteriormente, en las reuniones de grupo, todos los miembros de un mismo grupo compartían lo que habían aprendido sobre cada parte del tema.

Finalmente, se escogía al azar un alumno de toda la clase, quien sería encargado de exponer en 5 minutos una de las partes del tema sobre la que no había trabajado directamente. El resto de grupos de la clase debían prestar atención a la exposición. Para asegurar esto último, se requería a los demás grupos que enviaran por correo electrónico una puntuación de la exposición junto con una justificación de dicha puntuación. Como incentivo, el grupo que más se aproximase a las medias de las puntuaciones durante el curso tendrían +0,1 en la nota final de grupo. De igual forma, si la exposición del alumno seleccionado al azar era aprobada por sus compañeros, su grupo obtenía también +0,1.

Horario de trabajo

Miércoles, 8 de marzo de 2006

Orden	Actividad	Tiempo
1	**Reunión de expertos** Todos *los redactores* que hayan escrito sobre el mismo tema deberán juntarse para charlar sobre el tema e intercambiar las nociones básicas que hayan adquirido sobre el mismo. Los integrantes de la reunión compartirán los datos más significativos que hayan obtenido en sus trabajos. Durante esta actividad, *El/La editor/a* se encargará de leer los artículos de sus redactores para realizar las críticas de calidad que considere oportunas.	10 mins
2	**Entrevista a un redactor de otro grupo** *El/La editor/a* de cada grupo deberá realizar una entrevista al redactor que le haya correspondido por sorteo. Se realizarán 4 preguntas sobre el tema que haya tratado el entrevistado. De esta actividad saldrá una pequeña columna, de entre 200 y 600 palabras, para añadir a los artículos de la revista. *Los redactores* que no son entrevistados aprovecharán estos 15 minutos para crear un boceto de anuncio imaginario de un programa de gestión para juristas. Este anuncio podrá ser desde una simple imagen o un dibujo, hasta una lista de las características del programa imaginario. La principal exigencia es que el anuncio contenga un eslogan publicitario corto y original. El anuncio ocupará como mínimo ¼ de página y como máximo ½ página.	15 mins
3	**Reunión de grupo** Todos los integrantes de cada grupo se reunirán para que cada uno de ellos cuente al resto toda la información relevante acerca de su propio trabajo, así como lo que haya obtenido de la reunión de expertos. Es especialmente relevante que todos los miembros del grupo se enteren de todas las tareas que se han realizado.	15 mins
4	**Exposición de un miembro al azar** De todos los grupos, se escogerá a un redactor al azar y se le pedirá que explique el trabajo realizado por alguno de sus compañeros redactores (en ningún caso podrá hablar de su propio trabajo). Si la explicación resulta convincente, el grupo entero obtendrá +0.1 puntos sobre la nota final del trabajo.	5 mins

Figura 3. Ejemplo de horario de clase

En las clases de teoría de los 4 primeros temas, los alumnos recibían documentos con la teoría a tratar a través de la página web de la asignatura. En el primer tema, toda la información que necesitaban saber sobre el mismo se encontraba en la documentación que les dábamos los profesores. En los siguientes temas fuimos, progresivamente, dando cada vez menos información y encaminándoles hacia las fuentes que podían consultar. En los últimos 2 temas, los alumnos ya no recibían documentación escrita y debían encontrarla por sus propios medios.

3.3. Dinamización y actividades

Para que el curso no resultase aburrido a los estudiantes, decidimos dinamizar las clases variando los tipos de trabajos exigidos y proponiendo la realización de actividades

especiales. Estas actividades especiales no se encontraban planificadas desde principio de curso, sino que iban siendo incorporadas conforme considerábamos que podía ser necesario o iba a resultar interesante para los alumnos.

En este sentido, una revista da mucho juego, ya que no está compuesta exclusivamente de artículos o capítulos, sino que incluye distintos tipos de trabajo: artículos cortos y largos, reportajes normales y de investigación, anuncios, talleres prácticos, cartas al director, debates, historias, entrevistas, coleccionables y cd-rom de extras. En cada tema se fueron proponiendo actividades de esta lista para añadir a la revista. A final de curso, se hubo propuesto al menos una actividad de cada tipo (ver figura 4).

Lista completa de trabajos mínimos exigidos

En esta lista aparecen reflejados todos los trabajos mínimos obligatorios para obtener 5 puntos asegurados en la asignatura. Repasad la lista con atención, pues **no debe faltar ninguno** de ellos.

TEMA 6
- **2 debates a elegir** (temas: patentes, hackers o compra por Internet)
- **1 debate sobre el Canon Digital** (que debe ser el artículo principal de la revista, y debe aparecer como lo más destacado en la portada)

TEMA 5
- **4 artículos largos** (3 en grupos de menos de 4 personas)
- **1 historia teórica por cada miembro del grupo.**

TEMA 4
- **1 reportaje de investigación**.
- **1 entrega** del Knowledge Quiz con **4 tarjetas.**
- **1 Presentación** Multimedia (PowerPoint) – Opcional --

TEMA 3
- **3 talleres prácticos.**
- **3 dudas** especializadas (2 en grupos menores de 4 personas)
- **1 Editorial** incluyendo el *Informe de la junta de editores*.

TEMA 2
- **1 reportaje.**
- **2 cartas al editor** respondidas.

TEMA 1
- **2 artículos breves** (3 si el grupo lo forman 4 personas).
- **1 entrevista** a un miembro de otro grupo.
- **1 anuncio publicitario** simple.

Figura 4. Lista de trabajos mínimos exigidos

Uno de los trabajos especiales más interesantes fue el denominado Knowledge Quiz. Este trabajo fue pensado especialmente para un día lectivo que caía en el calendario en medio de muchas fiestas. Obviamente, en un día así, no podía proponerse una clase para avanzar trabajos, pues era muy probable (y así ocurrió) que hubieran algunas ausencias. Así pues, pensamos en incentivar a los alumnos mediante un simulacro de concurso de preguntas en clase. Durante los primeros 15 minutos, cada alumno recibía unas fichas que debía rellenar con preguntas sobre los últimos temas y las respuestas correctas. Una vez tenían preparadas sus fichas, comenzaba el turno de preguntas. Los alumnos iban enviándose preguntas unos a otros, tratando de responder y respondiendo en rebote, acumulando con ello puntos y perdiéndolos en caso de fallar las preguntas. Al final del simulacro de concurso, el ganador obtenía +0,25 en la nota final y los demás participantes canjeaban sus puntos por +0,01 en la nota final. Este simulacro de concurso fue una de las formas de compartir conocimiento sobre el tema que más gustó a los alumnos, e incentivó a venir a cerca de la mitad a clase en una fecha donde la mayoría de aulas quedaron desiertas.

3.4. Apunte sobre el examen final

Dada la alta carga de trabajo, esfuerzo y dedicación que mostraron en general todos los alumnos asistentes a clase, el examen final fue diseñado con una dificultad elevada de forma intencionada. Algunos alumnos decidieron a principio de curso no someterse a la metodología de trabajo en grupo, ejerciendo su derecho a realizar un único examen final. Como es lógico, los profesores les advertimos que en ese examen exigiríamos la demostración de un nivel de esfuerzo similar al de sus compañeros asistentes a clase.

Es por estos motivos por los que decidimos otorgar la posibilidad de presentarse a examen final a todos los demás alumnos, permitiéndoles incrementar su nota si conseguían hacerlo bien. Los resultados del examen final fueron sorprendentes. Al contrario de lo que se espera de una metodología de ABP con respecto al rendimiento sobre exámenes finales, todos los alumnos de clase que decidieron presentarse (en total, 5) obtuvieron mejores resultados que los alumnos que se lo habían preparado por libre (en total, 2). Eso sí, dada la elevada dificultad del examen, la nota más alta fue un 5.

4. Resultados

Al final del curso, a todos los alumnos se les realizó una encuesta de valoración para conocer sus opiniones sobre la metodología y dejarles expresar libremente aquellos aspectos que pudieran haberles resultado menos agradables. En esta sección mostramos los resultados de esta encuesta.

La encuesta realizada constaba de 4 bloques diferenciados. El primer bloque contenía 10 preguntas a nivel personal del alumno: horas dedicadas, asistencia a clase, opinión personal, posibilidad de recomendar la asignatura, ... El segundo bloque constaba de 6 preguntas exclusivas sobre su opinión acerca del trabajo en equipo. El tercer bloque contenía 28 afirmaciones puntuables de 1 a 5 sobre la metodología concreta de la asignatura. Finalmente, el cuarto bloque instaba a los alumnos a expresar su opinión personal por escrito.

Comentaremos brevemente aspectos destacados de los 2 primeros bloques para centrarnos en el tercero, que es el más interesante por lo que respecta a la metodología en sí misma.

Es destacable que el 33,4% de los alumnos eligió la asignatura preferentemente para aprender informática, mientras que el 26,3% la escogió porque pensaba que era una asignatura fácil. Con estas prespectivas iniciales, resulta agradable ver los aspectos más valorados de la asignatura, por orden de importancia (tabla 3).

Aspecto a valorar según preferencia	% pnts
He desarrollado nuevas habilidades útiles	18,9%
Ahora veo más interesante el derecho informático	16,0%
He aprendido cosas sobre informática	15,6%
El trabajo en equipo	15,2%
El método de evaluación seguido	12,6%
La forma de llevar la clase del profesor	12,1%
La asignatura ha sido muy fácil	9,5%

Tabla 3. Aspectos mejor valorados.

Contraste curioso, eso sí, de esta primera parte con la 2ª parte, acerca del trabajo en equipo. Seis preguntas en positivo sobre la importancia que los alumnos consideran que tiene el trabajo en equipo para ellos, con 5 posibles respuestas (1 a 5, Nada a Mucho), todas ellas valoradas en media entre 3,63 y 4. Teniendo en cuenta que 3 correspondía en la encuesta a "Lo normal" y 4 a "Bastante", podemos establecer que, en general, los alumnos le dan bastante importancia al trabajo en equipo.

Para la parte más importante de la encuesta, la dedicada a la metodología de la asignatura, creemos conveniente exponer la tabla completa con todas las preguntas realizadas a los alumnos y sus resultados de forma pormenorizada (tabla 4).

Afirmación a valorar (de 0 a 5)	Media
1. La metodología me permite trabajar en grupo	3,96
2. Es difícil trabajar en grupo con tan poco tiempo	3,71
3. Esta forma de trabajar requiere más esfuerzo	3,96
4. He aprendido más cosas	3,17
5. He desarrollado habilidades más interesantes que sólo memorizar	3,83
6. He podido investigar cosas por mi cuenta	4,04
7. He mejorado resolviendo problemas por mi mismo	3,83
8. He podido conocer mejor mis limitaciones	3,29
9. Hubiese preferido un método tradicional (clase expositiva + examen)	2,46
10. Hubiere preferido más técnicas de trabajo no presenciales	2,75
11. Hubiese preferido trabajar de forma menos guiada	2,25
12. Es interesante depender del trabajo de otros compañeros	2,38
13. He podido aprender de mis compañeros además del profesor	3,25
14. Me he desmotivado porque otros compañeros trabajaban poco	3,25
15. Me he desmotivado porque otros compañeros trabajaban mucho	2,29
16. El profesor debería organizar más nuestro trabajo	2,92
17. Me ha defraudado la actitud de mis compañeros	2,50
18. Me hubiese gustado poder dedicarle más tiempo	2,46
19. El método de trabajo es bueno	3,54
20. Hay muchos aspectos de método de trabajo que mejoraría	3,25
21. El método de trabajo es malo	1,96
22. El desarrollo de la asignatura ha sido un fracaso	2,21
23. Las actividades de clase de teoría me han gustado	2,54
24. Me parecen interesantes las actividades en grupo guiadas	3,29
25. Me gusta que los trabajos formen parte de un único proyecto de grupo para toda la asignatura (la revista)	3,63
26. Las revisiones cruzadas me han ayudado a mejorar mis trabajos	3,08
27. Las revisiones cruzadas han motivado mi espíritu competitivo	3,21
28. Las revisiones cruzadas no me han servido de nada	2,67

Tabla 4. Resultados Bloque 3

Estos resultados, finalmente, arrojan opiniones bastante curiosas. Por ejemplo, queda patente que los alumnos en general piensan que este método les hace trabajar más (p.3), pero a cambio les lleva a aprender más cosas y, sobre todo, a desarrollar habilidades interesantes más allá de la simple y clásica memorización y a ser más capaces de resolver problemas por si mismos (p. 4, 5, 7).

Uno de los aspectos que valoraron muy bien, y no sólo a nivel de preguntas puntuables, sino también a nivel de opiniones personales, fue la posibilidad de investigar por su cuenta (p. 6), sobre todo en los temas finales de la asignatura, donde se les pidió que realizaran debates sobre temas de actualidad.

Pese a indicar claramente que el método de trabajo es bueno, si acaso, con puntos mejorables (p. 19, 20, 21), no están del todo convencidos de preferirlo ante el clásico "clase magistral + examen final". Teniendo en cuenta que tratamos con alumnos de derecho, quienes viven ese método en prácticamente la totalidad de sus asignaturas, creemos que el hecho de hacerles dudar, con tendencia positiva, ya es un gran avance.

En las últimas 5 preguntas (p. 24 a 28) muestran su interés por las actividades en grupo guiadas realizadas en clase, aplauden el hecho de realizar trabajos constructivos, es decir, la materialización progresiva de un proyecto, y muestran su opinión sobre las revisiones. Curioso punto este último donde, pese a reconocer que las revisiones les son útiles en general, les són aún más útiles para fomentar su espíritu competitivo. Y es que, lógicamente, uno de los efectos de las revisiones cruzadas es muy claro: han tenido la oportunidad de comparar directa y minuciosamente su trabajo con 3 trabajos de otros grupos, quedando patentes sus carencias y fomentando una necesidad por estar a la altura.

5. Conclusión final

Sin duda alguna, si nos preguntasen a los profesores si la experiencia nos ha resultado tan interesante como para volverla a repetir, la contestación sería clara: ya la estamos repitiendo. Pese al trabajo que conlleva tener que estar implicado en los proyectos de todos los alumnos, redactar criterios y normas, preparar actividades especiales de clase, elaborar la teoría para la metodología del puzzle y, en definitiva, enfrentarse a las cosas conforme ocurren, en vivo y en directo, pese a todo, la experiencia a nivel personal con los alumnos merece mucho la pena. Es gratificante que los alumnos se encuentren contigo casi 1 año después en la cafetería, se alegren de verte y recuerden y valoren, una vez pasado el sufrimiento, que contigo hicieron algo que realmente les mereció la pena.

Referencias

[1] Buck Institute for Education. *A overview of Project Based Learning*. 2002.

[2] Markham, T. *Project based learning handbook*. Buck IFE, 2003.

[3] Moursund, D. *Project based-learning: using information technology.* ISTE Publications 1999.

[4] Alcober, J., Ruiz, S., Valero, M. *Evaluación de la implantación del Aprendizaje Basado en Proyectos en la EPSC(2001-2003).* XI Congreso de Innovación Educativa en las Enseñanzas Técnicas. Vilanova i la Geltrú, julio 2003.

[5] Miró Julià, J. *Aprendizaje a través de la escritura.* V Jornadas de Enseñanza Universitaria de la Informática, Jenui'99. La Almunia de Doña Godina, Zaragoza 1999.

[6] Miró Julià, J. *Aprendizaje a través de la escritura. Experiencias.* VI Jornadas de nseñanza Universitaria de la Informática, Jenui'00. Universidad de Alcalá, Alcalá de Henares, pp. 451-456.

[7] A. Martí Campoy, J.C. Campelo Rivadulla, R. Ors Carot. *Experiencias en innovación Docente: aspectos positivos y negativos de un caso real.* X Jornadas de enseñanza universitaria de la informática. Jenui'04.

[8] Llorens, F., Satorre, R. *Decálogo del profesor novel.* X Jornadas de enseñanza universitaria de la informática. Jenui'04.

[9] Anguas, J., Díaz, L., Gallego, I., Lavado, C., Reyes, A., Rodriguez, E., Sanjeevan, K., Santamaría, E., Valero, M. *La técnica del Puzzle al servicio del aprendizaje de la programación de ordenadores.* XII Jornadas de enseñanza universitaria de la informática, Jenui'06.

Métodos pedagógicos innovadores

Mejora del sistema Moodle mediante personalización de contenido y generación evolutiva de actividades aleatorias

Antonio Javier Gallego, José Requena, Mar Pujol, Javier Montoyo
Dpto. de Ciencia de la Computación e Inteligencia Artificial
Universidad de Alicante
Ctra. San V requenaicente del Raspeig s/n, Alicante
{ajgallego, requena, mar, jmontoyo}@dccia.ua.es

Resumen

El presente trabajo se centra en el proceso de personalización y mejora de la herramienta de gestión del aprendizaje (LMS[1]) Moodle para su adaptación a los requisitos de dos asignaturas de la titulación de Ingeniería Informática. Al abordar este proceso se detectaron ciertas funcionalidades que la plataforma aún no brinda y que sería deseable incluir. Una de ellas es que la presentación de los contenidos se adapte a las capacidades del usuario, para lo cual se desarrolló un nuevo tipo de actividad en el que la complejidad depende de los resultados obtenidos. Además, los ejercicios (de tipo matemático) se generan de forma aleatoria, permitiendo elegir entre predefinidos, rangos de complejidades o tipo "evolutivo", que irá incrementando el nivel según la habilidad del alumno.

Otro aspecto es que Moodle ofrece tantas opciones que a veces puede llegar a confundir. Por lo que se decidió personalizar la interfaz, simplificando su contenido y facilitando su manejo. Además se realizaron otra serie de modificaciones como la integración de LaTeX, la inclusión de nuevos roles, un nuevo proceso de registro, etc.

Por último se muestran los resultados de las encuestas que se les pasaron a los alumnos y las conclusiones extraídas de la experiencia.

1. Introducción

Durante el curso académico 2005/06 se decidió renovar las webs de las asignaturas de Álgebra y Estadística de la titulación de Ingeniería Informática. Esto fue debido principalmente a que el contenido se había quedado anticuado, y, dado que todo el código fuente era estático, cualquier modificación costaba bastante de realizar. Además también se querían introducir una serie de mejoras en la web como ejercicios de autoevaluación, herramientas de comunicación, seguridad para poder poner todo el material, etc.

Esta propuesta también supone dar un primer paso para la adaptación de las asignaturas al Espacio Europeo de Enseñanza Superior (EEES) [3]. Entre otros cambios, este nuevo modelo propone la implantación del sistema europeo de créditos (ECTS), el cual pasará a medir las horas totales de trabajo del estudiante y no sólo las de docencia presencial. Para conseguir esto las programaciones y metodologías docentes se han de centrar en el aprendizaje de los estudiantes, y no exclusivamente en las horas lectivas.

En una primera etapa del proyecto se realizó el estudio de los diferentes entornos de e-learning que existen actualmente en el mercado, buscando, entre otras cosas, que fuesen "software libre" a fin de poder realizar modificaciones en caso de que fuese necesario. Tras ver sus características y probar diferentes entornos, decidimos utilizar Moodle (ver apartado 2), pues proporciona muchas ventajas y muy pocos inconvenientes (ver sección 2.1).

Actualmente muchas universidades españolas utilizan Moodle (Jaume I [5], Politècnica de Catalunya [13], País Vasco [12], Extremadura [4], Las Palmas [6], Castilla-La Mancha [2], Cádiz [1], Oviedo [11], etc.), por lo que en este artículo no nos centraremos en su uso, pues no es ninguna

[1] LMS (*Learning Management System*) es un Sistema de Gestión del Aprendizaje.

novedad, sino en las mejoras y modificaciones realizadas para su adaptación a las asignaturas.

Una vez seleccionada la herramienta se inició el proceso de adaptación. En esta fase se detectaron ciertas funcionalidades de las que la plataforma aún no dispone y que no nos permitían cubrir todos los objetivos que se habían marcado inicialmente. Debido a esto se decidió modificar la herramienta para crear estas nuevas funcionalidades y adaptarla así a las asignaturas.

Como ya se ha comentado, las asignaturas para las que se han desarrollado las páginas webs son las de Álgebra y Estadística, ambas troncales de 6 créditos y del primer año de Ingeniería Informática. En Álgebra se ven las operaciones con matrices, los sistemas de ecuaciones lineales y los espacios vectoriales. Y en Estadística se ven los conceptos y problemas básicos de la estadística y la probabilidad, como son: funciones de probabilidad, teorema de Bayes, distribuciones, etc. Por lo que básicamente los ejercicios que se van a incluir en las webs son de tipo matemático y estadístico.

A continuación se muestra el estado actual de la tecnología involucrada, así como sus ventajas e inconvenientes. En el tercer apartado se presentan las propuestas de mejoras y se describe el proceso seguido para la adaptación de Moodle a las asignaturas. Por último se comentan los resultados obtenidos de la evaluación de la experiencia y las conclusiones extraídas.

2. Estado actual de la tecnología involucrada: Moodle

Moodle (*Modular Object-Oriented Dynamic Learning Environment*, Entorno de Aprendizaje Dinámico Orientado a Objetos y Modular) es un paquete de software diseñado para ayudar a los educadores a crear cursos de calidad a través de Internet. Fue creado por Martin Dougiamas basándose en trabajos sobre el constructivismo social en pedagogía (colaboración, actividades, reflexión crítica, etc.), que afirman que el conocimiento se construye en la mente del estudiante en lugar de ser transmitido sin cambios a partir de libros o enseñanzas. Un profesor que opera desde este punto de vista crea un ambiente centrado en el estudiante que lo ayuda a aprender en base a sus habilidades y conocimientos propios, en lugar de simplemente publicar y transmitir la información.

Algunas de las características más interesantes de Moodle son la gestión de tareas, las herramientas de comunicación (chat, foros y correo electrónico), los cuestionarios (permite la realización de test y la importación de recursos como objetos SCORM[2], multimedia, etc.), la gestión de recursos (permite incluir código HTML, texto, archivos de cualquier tipo, etc.), la posibilidad de crear glosarios, talleres, wikis, gestión del calendario, etc. Otras características importantes son la organización de documentos, la gestión de los usuarios, la posibilidad de poder crear copias de seguridad, y el control de acceso mediante permisos y roles. El administrador dispone de un control total sobre el curso, pudiendo personalizar incluso su apariencia (los bloques de los que se compone, las columnas, las secciones, etc.). [9]

En términos de arquitectura, se trata de una aplicación web que se puede instalar en cualquier sistema operativo en el que se pueda ejecutar PHP (Windows, Linux y Mac OS X). Funciona con diferentes bases de datos tipo SQL, como por ejemplo MySQL y PostgreSQL. Se distribuye gratuitamente como Software Libre (*Open Source*) bajo la Licencia Pública GNU.

Es importante definir los términos “software libre” y “código abierto”, pues a menudo son utilizados indistintamente para el software que puede ser distribuido y descargado desde Internet gratuitamente. Sin embargo, el software gratuito tiene la definición específica de “gratuito”. El software libre no es “gratuito” en el sentido monetario, sino que los usuarios de este software tienen la libertad de ejecutar, modificar, adaptar, personalizar y compartir sin cargo. Gracias a esta condición ha sido posible la edición de esta herramienta para crear nuevas funcionalidades.

2.1. Ventajas e inconvenientes

A continuación se presentan algunas de las ventajas más importantes de la plataforma Moodle y del e-learning en general [14], que son las que nos hicieron decantarnos por este tipo de metodología.

[2] SCORM (*Sharable Content Object Reference Model*) es una especificación que permite crear objetos pedagógicos estructurados.

- Como ya se ha comentado en el apartado anterior, Moodle tiene la ventaja de ser Software Libre, lo cual nos ha permitido modificarlo y adaptarlo libremente a nuestras necesidades.
- Permite al docente y a los estudiantes conocer los resultados y trazar nuevas estrategias y estilos para aprender. También facilita la comunicación de los alumnos, tanto con los profesores como con el resto de compañeros.
- El e-learning potencia el trabajo independiente, las actividades no presenciales y el autoperfeccionamiento, ya que el alumno siente la necesidad de retroalimentar sus conocimientos y aprender a autoevaluarse.
- Crea una independencia respecto a los horarios (posibilitando la autogestión del tiempo), respecto a la ubicación física (las distancias geográficas dejan de ser una limitación), y respecto a la máquina y el sistema operativo (sólo requiere conexión a Internet).

Pero también existen una serie de inconvenientes, los cuales se tratarán de solventar con algunas de las mejoras propuestas en el siguiente apartado.

- Muestra los mismos contenidos a todos los alumnos. Es decir, no tiene manera de ir acompañando el proceso de aprendizaje de cada alumno, ni de presentar sólo las lecciones, actividades, etc., que más se adecuen a su perfil y capacidad.
- No incorpora algunas de las herramientas pedagógicas más utilizadas, como por ejemplo: juegos de roles, crucigramas o la generación de ejercicios matemáticos. Pero esta carencia se puede suplir mediante la incorporación de nuevos módulos.
- La herramienta dispone de tantas opciones que en algunas ocasiones puede llegar a confundir a usuarios y administradores no iniciados. Por lo cual requiere un periodo de adaptación.

3. Propuesta de mejoras

En este apartado se comentan los pasos seguidos para el desarrollo de las nuevas funcionalidades, y la personalización y mejora de Moodle. Algunas de estas mejoras son:

1. Personalizar el contenido de la web, simplificando la interfaz y adaptándola a las necesidades de las asignaturas.
2. Añadir seguridad mediante contraseña a todos los materiales de la Web.
3. Modificar algunas de las características de los actuales roles.
4. Integración de Moodle y LaTeX.
5. Generación aleatoria del contenido de los ejercicios.
6. Generación de actividades adaptadas a las capacidades del usuario.

A continuación se verán en detalle cada uno de estos puntos:

3.1. Personalización de la interfaz y del contenido

La interfaz ha sido diseñada teniendo en cuenta unos objetivos muy claros:

- En primer lugar se pretendía actualizar los contenidos de las webs antiguas e incorporarlos a la nueva plataforma. Para esta tarea sólo fue necesario el uso de los recursos disponibles en Moodle.
- Otro objetivo era obtener un producto sencillo y fácil de manejar. En este sentido Moodle proporciona tantas opciones que en algunos casos puede llegar a confundir al usuario. Por lo que se optó por personalizar la interfaz, simplificando su contenido y adaptándolo a nuestras necesidades. De forma que dependiendo del rol del usuario se ocultaron ciertas opciones de la herramienta, mostrando sólo aquellas que nos interesaban. El único usuario que tiene acceso a todas las opciones es el administrador.

Además, como paso previo se realizó un pequeño seminario para que los profesores aprendieran a utilizar la nueva web.

3.2. Control de acceso

El acceso a la Web ha sido protegido porque contiene todo el material del curso: ejercicios, teoría, etc. Esto además nos permite tener un registro de quien ha entrado, ver las operaciones que ha realizado y de esta forma obtener estadísticas de uso de cada una de las partes de la

aplicación. Otra ventaja es que se pueden controlar los ejercicios que realiza cada alumno y las notas que obtiene en cada uno de ellos.

En la página principal se dispone de un bloque para poder identificarse como profesor o alumno mediante el nombre de usuario y la contraseña. En caso de que el usuario no se identifique se ha permitido que consulte el contenido de la web, sólo se solicitará la contraseña en el caso de acceso a alguno de los materiales protegidos (como son los recursos de las asignaturas, los ejercicios de autoevaluación, etc.). Moodle sólo permitía solicitar la contraseña de forma general para toda la web, sin posibilidad de proteger los recursos individualmente.

Hay dos posibles formas de registrarse en la Web: una es haciéndolo directamente y esperando a que un profesor valide el acceso (registro con confirmación), y la otra es solicitando a un profesor que nos dé de alta. El profesor tiene la posibilidad de crear usuarios de forma individual o múltiple, mediante la importación de una lista de usuarios.

3.3. Gestión de roles

En primer lugar sólo se diferenció entre los roles (tipos de usuarios) de administrador, profesor y alumno. Se realizaron modificaciones sobre los permisos de cada uno para que sólo se visualizaran las opciones de la herramienta que nos interesaban. De esta forma se simplifica mucho la interfaz y se facilita el uso de la aplicación. El único usuario que tiene acceso a todas las opciones es el administrador. Básicamente cada uno de estos roles tiene las mismas opciones que tenía previamente, pero con algunas modificaciones (sólo se muestran las más significativas).

Rol de alumno:

- Puede visualizar cualquiera de los recursos de la web.
- Puede matricularse en uno de los cursos y realizar los ejercicios de autoevaluación.
- Puede ver los resultados de los ejercicios ya realizados.

Rol de profesor, para cada uno de sus cursos:

- Gestiona a los alumnos.
- Puede consultar las calificaciones de los alumnos.
- Gestiona las actividades de autoevaluación.
- Gestiona los recursos.
- Puede ver el registro de la actividad.

Rol de administrador:

- Puede realizar todo lo que realiza un profesor.
- Gestiona administradores y profesores.
- Visualiza todas las opciones

3.4. Integración de Moodle y LaTeX

Al introducir el contenido de los ejercicios de autoevaluación de las asignaturas de Álgebra y Estadística nos dimos cuenta de que Moodle no permite por sí solo visualizar correctamente fórmulas matemáticas. Para conseguir esto se incorporó a Moodle la posibilidad de trabajar con LaTeX mediante la utilidad de software libre MimeTeX [8].

El código LaTeX se ha de introducir en los mismos campos que proporciona Moodle para el contenido normal del resto de los ejercicios, pero marcado con los limitadores “$$”. En el proceso de visualización el sistema llama a MimeTeX para transformar el código LaTeX en una imagen que muestre correctamente todos los símbolos matemáticos.

Nos encontramos con el problema de que al escribir código LaTeX utilizando el editor HTML que incorpora Moodle y guardarlo, la próxima vez que se visualizaba se habían eliminado ciertos caracteres que impedían que las fórmulas se mostraran correctamente. Para solucionar esto se optó por eliminar el cuadro de escritura con formato, permitiendo solamente introducir texto plano.

3.5. Generación aleatoria del contenido de las actividades

Una de las partes más costosas de la renovación de la web fue la elaboración de cuestionarios y problemas, ya que había que introducir manualmente cada una de las preguntas y además se corría el riesgo de equivocarse al escribir los datos o la solución. Por lo que se nos ocurrió la posibilidad de desarrollar un nuevo tipo de curso que resolviera esta tarea de forma automática, generando aleatoriamente el contenido de los ejercicios. Esto tiene la ventaja añadida de que de esta forma se proporciona al alumno un número

ilimitado de actividades para que pueda practicar y consultar las soluciones.

Este tipo de actividades pueden ser configuradas de dos modos. El profesor puede elegir previamente la operación, el rango de tamaños y otra serie de características dentro de las cuales se generarán los ejercicios aleatoriamente. El otro modo permite al alumno seleccionar él mismo dichas características. De forma que cuando accede a una de estas actividades, primero tiene que elegir la operación o conjunto de operaciones que desea practicar (suma, diferencia, multiplicación, trasposición, inversa, etc. de matrices) y a continuación el rango de tamaños permitidos. Con estos datos el sistema generará una lista de ejercicios de forma aleatoria y posteriormente comprobará si los datos introducidos por el usuario son los correctos. Además, los ejercicios se generan en formato LaTeX y hacen uso de la utilidad MimeTeX (ver apartado 3.4) para que las fórmulas matemáticas se muestren correctamente.

La programación de los ejercicios aleatorios no fue complicada, pues sólo había que tener en cuenta que se cumpliesen las propiedades exigidas por el tipo de operación. La parte más costosa fue la programación de la resolución, para poder comprobar si el resultado introducido por el alumno es el correcto. Inicialmente se empezó por la programación de una operación básica (suma de matrices) y una vez resuelto se fue ampliando el conjunto de operaciones (resta, multiplicación, división, inversa, traspuesta, trabajo con propiedades de las matrices, etc.). Actualmente estamos añadiendo nuevas operaciones para permitir el cálculo estadístico.

3.6. Generación de actividades adaptadas al nivel del usuario

En un sistema de educación a distancia es deseable adaptar la complejidad de los temas a tratar de acuerdo a la capacidad de aprendizaje del estudiante. Esto es necesario porque se considera que los procesos de aprendizaje varían en función de las capacidades de cada alumno. Es decir, un alumno con un buen perfil requiere de un proceso de aprendizaje diferente de aquel que tiene un perfil inferior. Por lo tanto, los contenidos a utilizar por uno deberían ser diferentes de los utilizados por el otro. Es a esto a lo que se denomina “personalización de contenidos”, para lo cual se requiere de la aplicación de técnicas que permitan adaptar el material a los diferentes perfiles de los alumnos [7].

Para conseguir este objetivo se realizó un primer paso al incorporar el “nivel evolutivo” a la generación de ejercicios comentada en el apartado 3.5. Esta modalidad es una opción que debe activar el usuario al comenzar la realización de una actividad. De esta forma la complejidad se irá adaptando a la nota media obtenida. Para esto el sistema se basa en el historial de notas del alumno para ese tema y materia, y modificará el nivel de dificultad dependiendo de dichos resultados. Para aumentar la complejidad simplemente se aumenta el rango de variación, el tamaño de los números o de la operación a realizar. En este caso es muy sencillo controlar el nivel ya que todos los ejercicios son de tipo matemático, pero se podría aplicar a cualquier otro tipo de actividad. En este sentido, éste sólo es un paso inicial para el control de la complejidad, que depende de la operación en concreto a tratar. Actualmente, además de estar incorporando nuevos tipos de ejercicios, se está trabajando en mejorar el control del nivel evolutivo y en que el aumento de la complejidad sea más eficiente.

4. Evaluación de la experiencia

En la primera parte de esta sección se presentan los resultados obtenidos de la evaluación del proyecto, y en la segunda parte se ha incluido un pequeño resumen del proceso de planificación y desarrollo seguido durante la implementación de la nueva herramienta.

4.1. Resultados obtenidos

Para valorar la opinión de los alumnos se les pasó una encuesta sobre el interés y satisfacción de la nueva web, y sobre las expectativas, intereses y dedicación estimada a la asignatura. En total se realizaron 356 encuestas a 71 alumnos, una para cada uno de los cinco temas a tratar en las asignaturas. Para realizar estas encuestas también se utilizó la herramienta Moodle, pues permite confeccionar el cuestionario fácilmente y además proporciona los datos de manera sencilla (en una hoja de cálculo) para poder ser tratados estadísticamente. A continuación se recogen los resultados de dichas encuestas, adjuntando un análisis preliminar (dado que estamos en una fase

inicial de implantación del proyecto) que evidentemente habrá que evaluar de forma más detallada y contrastar con las sucesivas encuestas que vayamos haciendo durante el curso.

Los alumnos han valorado que la dificultad media de las asignaturas es similar al resto de las asignaturas del curso (el 79% considera que la dificultad es similar o inferior, mientras que el 19% considera que es más difícil), por lo que en general han dedicado el mismo tiempo de estudio. La mayoría considera que este tiempo ha sido suficiente, en torno a las 10 horas por tema (fuera del aula). Es importante destacar que el 90% ha asistido a casi todas las clases.

En general, los alumnos han contestado que el grado de utilización del actual Campus Virtual de la universidad ha sido poco y en algunos casos casi nulo. El 34% considera que le da un uso normal, el 52% afirma haberlo utilizado poco o casi nada, y sólo el 14% lo ha usado con regularidad para la consulta del material. En comparación sí que han valorado positivamente la nueva web (el 86%), sobre todo las herramientas de comunicación (como los foros) y los ejercicios de autoevaluación. Sólo el 7% afirma no encontrar de gran utilidad la herramienta. La mayoría comenta que les han parecido muy interesantes los nuevos tipos de ejercicios (aleatorios y evolutivos) y que en general (el 78%) lo han usado para estudiar para las prácticas y el examen. Principalmente han destacado las siguientes ventajas: el hecho de poder practicar ejercicios y corregirlos de forma automática, y el hecho de tener infinidad de problemas distintos, configurables y adaptables a su nivel.

4.2. Planificación y desarrollo del proyecto

El proceso de desarrollo se dividió en varias etapas bien diferenciadas, que cubrían desde el estudio inicial y planificación del proyecto, hasta la implantación final en las asignaturas. El tiempo de ejecución fue de aproximadamente un semestre, con un becario contratado a tiempo parcial dedicado exclusivamente a la programación (ver sección de Agradecimientos), junto con la colaboración de los profesores implicados de ambas asignaturas. A continuación se detallan a grandes rasgos los cometidos de cada una de las fases de desarrollo:

Planificación y estudio inicial: esta fue la etapa más importante del proyecto, se concretaron todos los objetivos a desarrollar (aunque en etapas posteriores se incluyeron algunos más), se decidió la herramienta a utilizar, y se inició el proceso de adaptación. Fue en este último paso en el que se detectó la imposibilidad de cumplir todos los objetivos a partir de la funcionalidad básica de la aplicación. Por esta razón se decidió iniciar una nueva fase de desarrollo.

Desarrollo y pruebas: esta etapa fue bastante iterativa, y por ende la más larga y costosa de todo el proyecto. Hay que tener en cuenta que se requiere un tiempo de adaptación al código y modo de desarrollo de Moodle. Esta fase se subdividió a su vez en el desarrollo de forma separada de cada uno de los objetivos, empezando por los más simples hasta los más complejos. En primer lugar se realizaba el proceso de implementación, y posteriormente se procedía a las demostraciones y pruebas ante los profesores. De esta forma se pudo seguir en todo momento el desarrollo de las nuevas funcionalidades y del cumplimiento de los objetivos.

Implantación de la herramienta: una vez terminada la segunda fase, se procedió a la instalación del código en un servidor del departamento y a realización de un seminario para todos los profesores de las asignaturas. Por último se impartió un pequeño curso de presentación a los alumnos de cada uno de los grupos de prácticas.

5. Conclusiones y trabajos futuros

En este artículo se han descrito las modificaciones realizadas sobre la plataforma Moodle para su adaptación a las asignaturas de Álgebra y Estadística de la titulación de Ingeniería Informática. Durante el desarrollo del proyecto se detectaron algunas funcionalidades que no estaban disponibles, por lo que se decidió modificar la herramienta para desarrollar estas nuevas características.

Algunas de las mejoras más importantes que se realizaron fueron la generación de ejercicios de forma automática y aleatoria, y la adaptación de la complejidad de las actividades al nivel del usuario. Estas dos características son altamente deseables en los sistemas de e-learning, pues permiten ofrecer al alumno un número ilimitado de actividades personalizadas y adecuadas a su nivel. La plataforma Moodle posibilita la

incorporación de estas técnicas mediante el agregado de módulos, lo que (junto al hecho de ser software libre) nos ha permitido crear estos nuevos tipos de actividades y mejoras. Estas características hacen de Moodle una opción innovadora entre los sistemas de e-learning.

Además se han incluido otra serie de características como la posibilidad de escribir los ejercicios en LaTeX para poder visualizar las fórmulas matemáticas correctamente, y la personalización del contenido de la web dependiendo del rol del usuario.

Las actividades de autoevaluación propuestas han permitido al estudiante mejorar sus conocimientos al estimularlo a enmendar sus errores. De igual modo ha ocurrido con el docente, que ha podido retroalimentarse acerca de la marcha del proceso de enseñanza-aprendizaje de sus estudiantes, ayudándole así a perfeccionar sus métodos y procedimientos de trabajo.

El sistema actualmente está en versión de pruebas, es el primer año que se utiliza en las asignaturas y queremos detectar todos los errores. También se está ampliando la herramienta para incluir otro tipo de operaciones de estadística, lógica, etc. Como trabajo futuro se pretende conectar la herramienta con Matlab para evaluar los resultados de las operaciones, de esta forma se ampliaría enormemente el conjunto de ejercicios y sería mucho más rápido su desarrollo. Además también se está estudiando como mejorar el sistema de ejercicios evolutivos, tanto en el análisis del perfil del alumno, como en el control del aumento de la complejidad.

Una vez que se hayan incorporado todas estas mejoras y se hayan solucionado todos los posibles fallos, se pretende colaborar con Moodle para que toda la comunidad se pueda beneficiar del trabajo realizado. Para ello se ha de adaptar el código según sus guías de desarrollo y normativa [10].

Agradecimientos

Este trabajo ha sido realizado durante el curso 2005/06 gracias al soporte del proyecto “Proyecto de implantación ECTS en las titulaciones de informática (Escuela Politécnica Superior)” dentro del programa de ayudas financiado por el Vicerrectorado de Calidad y Armonización Europea de la Universidad de Alicante y la Conselleria de Empresa, Universidad y Ciencia, en la Escuela Politécnica Superior.

Además de los firmantes del artículo, también ha colaborado en el proyecto el profesor Joan Josep Climent, al que deseamos agradecer su trabajo sin el cual no hubiera sido posible esta experiencia.

Referencias

[1] Cádiz. http://moodle.uca.es/login/index.php
[2] Castilla La Mancha. https://moodle.uclm.es/
[3] European Ministers of Education (1999). The European Higher Education Area Bologna Declaration, Bologna.
[4] Extremadura. http://campusvirtual.unex.es/zonauex/avuex/
[5] Jaume I. http://aulavirtual.uji.es/
[6] Las Palmas. http://telepresencial.ulpgc.es
[7] Marcelo Karanik, José Pérez, Jorge Roa, Sergio Gramajo, Rodrigo Vigil, Rosina Ramirez. Mejora de la Plataforma de e-learning Moodle Utilizando Redes Neuronales. Jornadas de Educación en Informática y TICS en Argentina. JEITICS 2005.
[8] MimeTeX. http://www.forkosh.com/mimetex.html
[9] Moodle. http://moodle.org
[10] Normativa Moodle. http://docs.moodle.org/es (Documentación para desarrolladores - Cómo puede colaborar)
[11] Oviedo. https://www.aulanet.uniovi.es
[12] País Vasco http://moodle.ehu.es/moodle/
[13] Politécnica de Catalunya. https://atenea.upc.edu/moodle
[14] R. Alonso Reyes, N. Cabrera Cabrera, O. Estévez, G. Jiménez, G. Limaya, M. Barba. La evaluación del aprendizaje usando las actividades de Moodle. 3rd Internacional Conference on Multimedia and Information and Communication Technologies in Education. m-ICTE2005.

Aplicación del portafolio como estrategia de evaluación formativa

María A. Pinar Sepúlveda, Joaquín Gracia Morán
Dpto. de Informática de Sistemas y Computadores
Escuela Técnica Superior de Informática Aplicada
Universidad Politécnica de Valencia
Camino de Vera s/n, 46022 Valencia
{mapinar, jgracia}@disca.upv.es

Resumen

Con la llegada del Espacio Europeo de Educación Superior (EEES), las estrategias didácticas deben cambiar para centrarse en el aprendizaje del estudiante, convirtiendo al alumno en un elemento activo dentro de su aprendizaje, incentivando su participación, de tal manera que se sienta parte activa del proceso de enseñanza–aprendizaje.

Otro cambio importante es la incorporación de metodologías activas dentro de este proceso de enseñanza–aprendizaje. Una de las técnicas que implica a los alumnos en su propio aprendizaje es el portafolio. Con este método, el alumno puede reflexionar acerca de sus metas, progresos, dificultades, etc., consiguiéndose además una evaluación individualizada y participativa.

En este trabajo se va a presentar la experiencia que se ha llevado a cabo durante el primer cuatrimestre del curso 2006/07, donde en una asignatura obligatoria de redes de la Universidad Politécnica de Valencia se ha ofrecido a los alumnos la opción de realizar la evaluación mediante portafolio, frente a la tradicional de un único examen final.

1. Motivación

Uno de los cambios más significativos que rigen la implantación del EEES es la formación en competencias [7], permitiendo el desarrollo integral de los alumnos, potenciando de esta manera su madurez y su compromiso social y ético. Es decir, con el EEES los estudiantes deben asumir un papel protagonista en su formación. Con este fin, y teniendo en cuenta el futuro EEES, cada vez es más común la utilización de metodologías activas dentro del aula [1][3]. Una de las técnicas es el *portafolio* .

El término original inglés *portfolio* [8] hace referencia a una carpeta grande y estrecha en la cuál el artista guardaba sus pinturas y dibujos. También se conoce con esta denominación el conjunto de materiales representativos del trabajo realizado por una persona, especialmente por un artista.

En nuestro entorno docente, podemos definir "el portafolio del estudiante" como una selección de trabajos organizada por el alumno con el objetivo de documentar, de manera reflexiva, el proceso y la consolidación de su aprendizaje [1]. Permite que los alumnos se impliquen en su propio aprendizaje. De esta manera, a través del análisis de las producciones cotidianas, el alumno puede tomar conciencia de su evolución, organizándose ellos mismos el tiempo, desarrollando el trabajo de forma procesual. De esta forma, la evaluación no se limita a una prueba final, sino que se realiza a lo largo del curso de forma continua, individualizada y participativa. Algunas ventajas de la evaluación por portafolio son [6]:

- Involucra al estudiante en un aprendizaje centrado en su realidad profesional y personal (aprendizaje centrado en quien aprende).
- Capacita a la persona para el aprendizaje a lo largo de la vida, así como para el desarrollo de habilidades de autoaprendizaje (aprendizaje autónomo).
- Implica al estudiante en su propio proceso de aprendizaje (aprendizaje responsable).
- Muestra hasta donde ha llegado el estudiante y el camino recorrido, eliminando el riesgo de valoraciones basadas en datos simples.

- Además de la función evaluativa, el portafolio presenta un gran valor educativo.
- Permite al alumno controlar su proceso de aprendizaje.

Por otra parte, otro de los objetivos que se pretendían alcanzar en esta experiencia era la realización de una *evaluación basada en competencias*. Este tipo de evaluación no se interesa solamente en conocer cuánto sabe el estudiante, sino los resultados que se reflejan en un desempeño concreto; se caracteriza por estar orientada a valorar el desempeño real del alumno, el cual sintetiza los conocimientos, habilidades, destrezas, actitudes y valores involucrados en la realización de una función o actividad.

El presente trabajo se ha organizado de la siguiente forma. La sección 2 presenta el contexto de la asignatura. La sección 3 describe la implantación del portafolio en la asignatura, mientras que la sección 4 realiza un estudio de la acogida y la participación de los alumnos en esta experiencia. En la sección 5 se presentan los resultados académicos obtenidos y para finalizar, la sección 6 presenta las principales conclusiones de este trabajo.

2. Contexto de la asignatura

El presente trabajo se ha realizado en la asignatura de Fundamentos de Redes de Computadores (FRC), asignatura obligatoria dentro de la titulación de Ingeniero en Informática de la Facultad de Informática de la Universidad Politécnica de Valencia [5]. Durante el curso pasado se matricularon 174 alumnos. Esta asignatura se cursa en el primer semestre del tercer curso de la titulación. Como particularidad podemos decir, además, que es la primera asignatura en la que los alumnos se relacionan con el mundo de las comunicaciones y las redes de computadores , siendo sus objetivos básicos:

- Estudio básico de las comunicaciones entre sistemas informáticos.
- Presentación de la arquitectura de comunicaciones TCP/IP.
- Comprensión del esquema de comunicaciones Cliente/Servidor.
- Estudio de las principales aplicaciones que actualmente funcionan en Internet, así como de su interfaz con el resto de la pila de protocolos.

El temario resumido de FRC es el siguiente:

- TEMA 1. INTRODUCCIÓN.
- TEMA 2. CONFIGURACIÓN AUTOMATICA DE DIRECCIONES (DHCP) Y SERVICIO DE NOMBRES DE DOMINIO (DNS).
- TEMA 3. LA APLICACIÓN DE WORLD WIDE WEB.
- TEMA 4. EL CORREO ELECTRÓNICO.
- TEMA 5. OTRAS APLICACIONES EN RED.
- TEMA 6. INTERFACES DE PROGRAMACIÓN.
- TEMA 7. SEGURIDAD EN LA RED.

Complementa este temario teórico la realización de cinco sesiones prácticas de dos horas que se organizan en cuatro prácticas:

- PRACTICA 1 (1 sesión). Conexión a otras máquinas.
- PRACTICA 2 (1 sesión). El Servidor Web Apache.
- PRACTICA 3 (1 sesión). Estudio de diferentes protocolos mediante un analizador de red: *Ethereal*.
- PRACTICA 4 (2 sesiones). Programación de Sockets en C.

La realización de las prácticas siempre han tenido un carácter voluntario, así como la asistencia a las clases de teoría.

Como se puede ver, los contenidos de la asignatura están relacionados con aplicaciones de uso común entre los estudiantes de Ingeniería Informática (como es el correo electrónico o Internet), lo que hace que se presente al alumno como una materia bastante interesante. A pesar de esto, cabe destacar la baja asistencia a las clases, tanto teóricas como prácticas, por parte de los alumnos.

La media de alumnos matriculados en los últimos cursos ha sido de unos 170, que se distribuyen en tres grupos de teoría y siete de prácticas. A esto le corresponderían una media de 57 alumnos por grupo de teoría y 24 alumnos por grupo de prácticas, mientras que la asistencia real se acerca a un poco más de la mitad de lo asignado.

Otra peculiaridad de la asignatura es que parece que los alumnos no la consideran una

asignatura difícil, ya que a pesar de la baja asistencia a las clases, los alumnos no tienen miedo a presentarse al examen, tal como se puede ver en la Figura 1. En ella se puede observar que el porcentaje de alumnos presentados suele ser del 90% respecto a los alumnos matriculados.

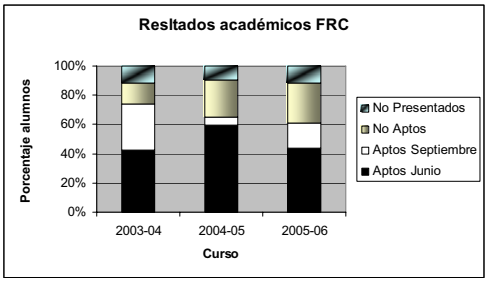


Figura 1. Resultados académicos FRC

La evaluación de la asignatura se ha realizado tradicionalmente mediante una única prueba final en la que se decidía la nota del alumno. Este examen ha sido normalmente una combinación de problemas, cuestiones cortas y preguntas de tipo test.

3. Puesta en marcha del portafolio

Durante el curso 2006/07, y con el fin de empezar a adaptarnos al EEES, los profesores que impartimos la asignatura nos planteamos la incorporación a la misma de la herramienta del portafolio del alumno. Nos planteamos varias formas de incorporarlo.

En primer lugar, pensamos plantearlo a los alumnos de forma obligatoria para todos. El problema que planteaba esta opción, recogido de experiencias anteriores, es que a los alumnos no les gustan demasiado los cambios bruscos y preferíamos que la acogida a nuestra propuesta fuera más favorable.

En segundo lugar, nos planteamos que los alumnos pudieran optar a la evaluación tradicional o a la evaluación por portafolio. Pero esta decisión la tenían que tomar al comienzo del curso y no podían cambiar de opinión más adelante. No adoptamos esta postura ya que pensamos que los alumnos suelen ser más bien conservadores ante lo desconocido y no tendríamos suficiente número de alumnos implicados en el proyecto.

Y, por último, decidimos plantear la participación en el portafolio de forma opcional, y, además, que los alumnos que optaban por esta forma de evaluación pudieran abandonarla en cualquier momento, incluso conociendo ya la nota obtenida. La única condición es que el abandono lo tenían que decidir siempre antes de la realización del examen final.

Esta última opción ayudó a que la acogida entre el alumnado fuera buena, sin ningún tipo de desconfianzas, ya que como ellos mismos nos transmitían "no tenían nada que perder por probarlo".

El portafolio que propusimos al alumnado consistió en la realización de las siguientes tareas:

- Resolución individual de una serie de ejercicios prácticos y/o teóricos relacionados con los contenidos de la asignatura (5 entregas a lo largo del curso).
- Entrega de una memoria para cada una de las prácticas que se realizaban en grupos de dos personas. En esta memoria se daba especial importancia a que el alumno fuera capaz de relacionar los conceptos teóricos con lo trabajado en la sesión de prácticas (4 entregas).
- Un trabajo realizado en grupos de 4 ó 5 personas en el que se planteaba a los alumnos el diseño de una práctica de laboratorio destinada a los alumnos de su clase. En este trabajo se les proponía la utilización de la herramienta que ellos mismos habían trabajado en la Práctica 3: el analizador de protocolos Ethereal.
- Se han propuesto, a lo largo del curso, una serie de debates sobre temas de la asignatura, que los alumnos tenían que preparar previamente.
- Realización de dos exámenes parciales, que cubrían, respectivamente, el primer y segundo tercio de la asignatura.
- Un examen final que abarcaba todos los contenidos de la asignatura, tanto los teóricos como los prácticos.

Todo esto se ha combinado, por supuesto, con un esquema de tutorías que ha permitido al alumno en todo momento participar activamente en su proceso de evaluación, así como ser más consciente de su proceso de aprendizaje.

Mediante estas actividades se han pretendido trabajar una serie de competencias y habilidades [7] que presentamos a continuación:

- Habilidades de análisis y síntesis de conocimientos.
- Priorización de conocimientos.
- Trabajo autónomo.
- Trabajo en equipo, interacción social, cooperación.
- Habilidades interpersonales.
- Gestión del tiempo.
- Capacidad de aprender.
- Capacidad de adaptarse a nuevas situaciones.
- Capacidad docente.
- Comunicación oral y escrita.
- Creatividad.

4. Acogida y participación de los alumnos en el proyecto de portafolio.

En principio, la acogida a la propuesta presentada a los alumnos fue bastante buena, de forma que el 46% de los alumnos matriculados decidieron participar en el proyecto de portafolio.

A medida que iba avanzando el curso han ido abandonando hasta quedar en un 26% del total de alumnos matriculados en la asignatura. El porcentaje más alto de abandono se produjo en el primer mes del curso, y el motivo que aducían principalmente los alumnos era que no podían llevar adelante el volumen de trabajo que les proponíamos. Preferían realizar un único esfuerzo final para el examen de la asignatura.

El día del examen final se les pasó una encuesta a los alumnos que habían participado en esta experiencia de portafolio. Con ello pretendíamos conocer la opinión del alumnado que había seguido hasta el final esta opción.

Reflejamos a continuación algunos de los resultados más significativos.

- En primer lugar preguntamos a los alumnos acerca de las diferentes actividades que habían formado el portafolio del estudiante, y les pedimos que nos indicaran cuál había sido la actividad que más les había aportado o enriquecido a la hora de preparar la asignatura.
  Las dos actividades que más han valorado los alumnos han sido los ejercicios prácticos que se iban planteando a lo largo del curso y los dos exámenes parciales.
  Los alumnos opinan que los ejercicios planteados les ayudaban a ir afianzando y practicando los conocimientos vistos en clase, mientras que los exámenes parciales les motivaban a estudiar y llevar la asignatura al día sin dejarla para el final.
  El resto de actividades propuestas han obtenido una valoración similar y considerablemente inferior a las dos primeras.

- En segundo lugar se preguntaba a los alumnos si consideraban una buena opción que para el próximo curso la evaluación se realizase solamente mediante portafolio.
  Nos ha sorprendido el resultado ya que aunque todos han valorado muy positivamente la experiencia de portafolio, solamente un 15% de los alumnos que lo habían seguido adoptarían esta postura. En general, la mayoría dicen preferir tener la posibilidad de elegir, en un momento dado, trabajar la asignatura de forma tradicional.
  A pesar de esto, al 100% de los alumnos les parece bien que exista la opción del portafolio.

- Se preguntó a los alumnos, también, sobre la posibilidad de que todas las asignaturas permitieran una evaluación por portafolio y la respuesta fue muy positiva.
  Un 74% del alumnado veía muy favorable la incorporación de esta metodología en la titulación completa, aunque reconocían que el volumen de trabajo que les supondría sería muy alto. Muchos comentan la necesidad de una buena coordinación entre los responsables de las diferentes materias para que no se saturara al alumno con entregas continuas de trabajos propuestos.
  De hecho, para el resto de alumnos que no veían adecuada esta opción, el motivo principal que aducían era el excesivo trabajo que podía suponer para el alumno. No lo veían viable.

- A continuación se les preguntaba a los alumnos qué les había aportado la evaluación por portafolio frente a la evaluación

tradicional y reflejamos a continuación las respuestas que nos parecen más significativas:

- "Una formación más extensa y práctica que la normal"
- "Obliga a indagar más sobre ciertos aspectos de la asignatura que en una evaluación tradicional"
- "Se valora mucho más el esfuerzo diario"
- "Un seguimiento continuo de la asignatura durante el cuatrimestre"
- "Me ha aportado el centrarme más en la asignatura y asimilar mejor los conceptos"
- "Se implica más al alumno en la asignatura y en su evaluación"
- "Mayor interacción con el resto de compañeros y con el profesor"
- "Creo que el alumno aprende más y de una forma más fácil"
- "Tranquilidad, ya que no te lo juegas todo en un día"
- "Poder ir corrigiendo los errores que vas teniendo"
- "Enlazar mejor todos los conceptos de todos los temas"
- "Una forma nueva y mejor de preparar una asignatura"
- "He asimilado mucho más el contenido de la asignatura"
- "Mayor estudio y mejor aprendizaje"
- "Mejor relación con el profesor en las tutorías, que me ha ayudado a ir mejorando mis conocimientos"

Como se puede comprobar parece que hay un reconocimiento positivo general de los alumnos hacia la experiencia del portafolio frente a la evaluación tradicional.

- Y, por último, se invitó a los alumnos a que reflejaran las deficiencias que ellos veían en esta nueva metodología. Estas son las respuestas más significativas que nos hemos encontrado:

  - "Es demasiado trabajo, sería imposible trabajar todas las asignaturas con esta intensidad"
  - "La evaluación resulta demasiado subjetiva en algunos aspectos"
  - "Los grupos deberían ser menos numerosos para que el profesor pudiera conocer mejor a cada alumno"
  - "Al final del todo y después de haber trabajado tanto durante el curso, no se debería hacer un examen final con toda la materia"
  - "Aglomeración excesiva de trabajos que quitaban tiempo para dedicarlo a otras asignaturas"
  - "Los parciales deberían eliminar materia del examen final"
  - "La carga de trabajo del portafolio en algunos momentos del cuatrimestre ha sido excesiva"
  - "Más flexibilidad en los plazos de entrega de los trabajos"

Como podemos ver, la mayor parte de las sugerencias van en la línea de reducir el volumen de trabajo que hemos ido proponiendo a los alumnos, así como una mayor valoración de los esfuerzos realizados. También aparecen algunas reflexiones sobre una mejor adaptación de la asignatura a esta nueva metodología: menos alumnos por grupo, coordinación con otras asignaturas….

5. Resultados académicos

Uno de los datos más reveladores, en cuanto al buen funcionamiento de esta nueva metodología docente, lo constituyen los resultados académicos obtenidos.

Vamos a mostar en el presente artículo datos relativos a la primera convocatoria de la asignatura, ya que son los únicos de los que disponemos en estos momentos. Para poder hacer una valoración completa, así como para realizar una comparativa con otros años, necesitaríamos tener los datos de las dos convocatorias que componen la evaluación del presente curso.

En este caso, ya que la experiencia de portafolio se ha realizado de forma voluntaria, resulta especialmente interesante hacer una comparativa entre los resultados obtenidos por los alumnos que han optado por la evaluación mediante portafolio y los alumnos que optaron por la evaluación tradicional.

- De los alumnos que siguieron la evaluación por portafolio, un 87% de ellos consiguieron aprobarla frente a un 13% que no la superaron. La nota media obtenida por este colectivo ha sido de 5,8 puntos.
- Los resultados obtenidos por los alumnos que optaron al examen final únicamente son: el 29% de ellos aprobaron frente a un 71% de suspendidos. La nota media obtenida por este grupo de alumnos es de 4,1 puntos.

Como se puede comprobar los resultados son bastante reveladores, reflejando que entre el grupo de alumnos de portafolio el número de aprobados es prácticamente 3 veces superior al número de aprobados entre los que se evaluaron mediante la evaluación tradicional.

Y, aunque consideramos haber obtenido unos resultados muy positivos de la experiencia realizada, deberíamos tener en cuenta una serie de factores que han podido desvirtuar un poco estos resultados:

- En primer lugar, debemos tener en cuenta que el tipo de alumno que ha participado desde el principio en este proyecto de portafolio, es un alumno que asiste habitualmente a clase. Este tipo de alumnado es el que suele obtener unos mejores resultados académicos ya en la evaluación tradicional, debido a que suele llevar la asignatura más al día.
- Por otra parte, entre el colectivo de alumnos que decidió participar inicialmente en el portafolio y que ha ido cambiando de idea, el principal motivo ha sido que no querían o no podían dedicarle tanto tiempo a la asignatura. Esto, necesariamente, ha de verse reflejado en los resultados.
- Entre este grupo de alumnos se encuentran también aquellos que han abandonado la opción del portafolio pocos días antes del examen final, porque las calificaciones que habían ido obteniendo durante el curso no eran muy elevadas.
- Por último, señalar que tradicionalmente Fundamentos de Redes de Computadores es una asignatura con un alto índice de presentados al examen final (una media de un 90% de los alumnos matriculados).
- Hay una gran cantidad de alumnos que sin haber preparado demasiado la asignatura, se presenta al examen final para probar suerte. Por supuesto, en este grupo tenemos un alto porcentaje de suspendidos.

A pesar de estos factores que pueden desvirtuar los resultados obtenidos, consideramos que la experiencia de la evaluación por portafolio ha sido altamente satisfactoria.

6. Conclusiones

Este trabajo presenta una experiencia de introducción de una nueva metodología docente: el portafolio del alumno, como intento de adaptación al nuevo Espacio Europeo de Educación Superior.

Sabemos que los alumnos son especialmente propensos a mostrar ansiedad cuando se introducen nuevos métodos docentes y especialmente si estos afectan a la evaluación, como es el caso del portafolio [4].

Esta esperable disconformidad se ha intentado paliar dotando a esta experiencia de un carácter de no obligatoriedad, así como facilitando al alumno una explicación clara sobre la finalidad educativa de cada uno de los ejercicios planteados, la dinámica que se iba a seguir y sobre cómo se iba a llevar a cabo la evaluación.

Esto ha motivado una respuesta bastante favorable por parte del alumnado, que desde el principio se ha involucrado con bastante ilusión en el proyecto.

Nos parece importante señalar que aunque los alumnos han visto en esta opción de portafolio una posibilidad para superar con más éxito la asignatura, a posteriori han valorado muy positivamente todo el trabajo realizado, desde el punto de vista formativo. Incluso en aquellos casos en los que la puntuación obtenida por el alumno no era la esperada, el alumno ha transmitido una valoración positiva de la experiencia.

Desde el punto de vista de los profesores que impartimos la asignatura, también reconocemos en el portafolio del alumno una muy buena herramienta para conseguir la consolidación de una serie de conocimientos, habilidades y competencias en el alumnado.

Por todo esto, nos parece interesante seguir trabajando e investigando en este proyecto,

aunque somos conscientes de que para una implantación del portafolio más consolidada, tanto en esta asignatura como en otras de la titulación, serían necesarias realizar una serie de adaptaciones:

- En primer lugar, sería necesaria una mayor formación en el tema por parte del profesorado, con el fin de conseguir una mayor efectividad, así como una evaluación lo más objetiva y fiable posible.
- Sería necesario, también, redimensionar el volumen de trabajo que los alumnos han tenido que realizar, con el fin de compatibilizar este tipo de experiencias con la buena marcha del alumno en otras asignaturas.
- También desde el punto de vista del profesor el volumen de trabajo se duplica fácilmente, lo que hace bastante difícil poder llevar todas las asignaturas mediante esta metodología docente.
- Es muy complicado llevar a cabo este tipo de proyectos en asignaturas troncales y/o obligatorias debido al elevado número de alumnos por grupo que suelen tener.
- Una de las ventajas que ofrecen este tipo de metodologías es que se basan en una mayor interacción del alumno con el profesor, y esto resulta excesivamente complicado cuando el número de alumnos es tan alto.
- Si quisiéramos conseguir una implantación generalizada en la titulación sería necesaria una gran tarea de coordinación por parte del Centro y con el resto de profesores de otras asignaturas.

Referencias

[1] Barberà Gregori, E. *Avaluació alternativa del estudiant mitjançant portfolios.* ICE de la Universitat de Girona. Girona, 2005 .

[2] Blanc Clavero, S. et al. *Metodologías activas para facilitar el aprendizaje en un curso básico de Tecnología de Computadores.* VII Congreso en Tecnologías Aplicadas a la Enseñanza de la Electrónica, Madrid, 2006.

[3] Fernández March, A. et al. *Metodologías activas para la formación de competencias.* Material de trabajo. Instituto de Ciencias de la Educación, Universidad Politécnica de Valencia, 2006.

[4] Friedman Ben D. et al. *Portfolios as a method of student assessment.* AMEE Medical Educational Guide Nº 24. 2001.

[5] http://www.fiv.upv.es.

[6] Pietroni R, Heath I, Burrows P, Savage R, Sowden D, Millard L. *Portfolio-based learning in general practice. Report of working group on higher professional education.* London: Royal College of General Practitioners , 1993(Occasional paper 63.)

[7] Tuning Educational Structures in Europe project, "Approaches to teaching, learning and assessment in competence based degree programmes", 2005. http://www.unideusto.org/tuning.

[8] Valero Marcet, M. *Introducción del portafolio del estudiante y del portafolio docente en la Facultad de Ciencias de la Salud y de la Vida de la Universidad Pompeu Fabra.*Tesis Doctoral realizada en el Departamento de Ciencias Experimentales y de la Salud, Universidad Pompeu Fabra. Barcelona, 2006.

Aplicación del método del caso a una parte de la asignatura Redes de Computadores

Guillermo Azuara Guillén
Dpto. de Ingeniería Electrónica y Comunicaciones
Universidad de Zaragoza
Pedro Cerbuna 12, 50009 Zaragoza
gazuara@unizar.es

Resumen

En este artículo se presenta la experiencia de aplicación del método del caso a un tema de la asignatura "Redes de Computadores", que se imparte en el segundo cuatrimestre de segundo curso de la titulación de Ingeniería Técnica de Informática de Gestión.

A lo largo del artículo se presentan los fundamentos teóricos y una breve descripción de esta metodología docente. Luego tras analizar diversos aspectos relacionados con su preparación y desarrollo en el aula se presentan las conclusiones.

En general la experiencia ha sido valorada como muy positiva por parte de los alumnos y como positiva por parte del profesor.

1. Introducción

La sociedad en la que actualmente vivimos, se puede caracterizar por la gran cantidad de información que se genera, procesa e intercambia. Ya no es suficiente con que los profesionales tengan unos ciertos conocimientos, sino que la sociedad demanda de ellos unas ciertas capacidades. Aunque esta situación no es del todo novedosa, en los últimos tiempos el mundo universitario ha empezado a caminar en esta dirección, reorientando el proceso de aprendizaje clásico en el que el docente era el centro del proceso, a un modelo en el que el alumno es el principal actor del proceso de aprendizaje, siendo el docente la persona que le va guiando en el camino.

En este contexto se han ido generando diversos documentos, como el proyecto Tunning [2] o el libro blanco del título de grado en ingeniería informática [1], en los que se han plasmado las competencias deseables que debería tener un estudiante al completar una titulación.

Para propiciar en los alumnos la potenciación de las competencias descritas en los citados documentos existen diferentes metodologías docentes, entre las que se incluye el método del caso (o resolución de casos) que es la que se va a describir en el presente documento.

En el siguiente apartado se va a bosquejar el contexto de la asignatura, una breve descripción de la metodología y finalmente los objetivos perseguidos.

En el apartado tercero se presenta el proceso de diseño del caso y diversas reflexiones sobre este proceso.

El apartado cuarto aborda el desarrollo de la experiencia en el aula.

Y finalmente en los apartados quinto y sexto, se presentan los resultados y las conclusiones obtenidas, respectivamente.

2. Contexto y objetivos

La asignatura "Redes de Computadores" se imparte en el segundo cuatrimestre del segundo curso de la titulación de ingeniería técnica de informática de gestión de la Escuela Universitaria Politécnica de Teruel (Universidad de Zaragoza). Es una asignatura de 6 créditos LRU (3 teóricos y 3 prácticos), teniendo una docencia de 3 horas de pizarra a la semana y dos horas de prácticas quincenales.

La asignatura se divide en cinco módulos, de los cuales el último, la parte de seguridad en redes, es el que se ha desarrollado utilizando el método del caso.

El perfil de los alumnos corresponde a alumnos que no tienen ningún conocimiento sobre seguridad informática, por lo que se debe presentar la materia desde cero.

La metodología utilizada, la resolución de casos, surgió a principios del siglo pasado, en torno a 1920, en el seno de las escuelas de

negocios y derecho. Cabe citar que alguna de ellas, como *Harvard Business School*, después de más de 80 años de experiencia, basa más del 80 % de sus clases en esta metodología [4].

El método del caso consiste básicamente en diseñar un caso de estudio lo más detallado posible, para que el alumno se vea inmerso en un entorno de decisión aparentemente real, donde tenga disponible la misma información de que dispondría en su puesto de trabajo. El caso, además de ofrecer toda esta información, también debe proporcionar un contexto, con diferentes personajes y pistas que permitan al alumno realizar su proceso de aprendizaje en pos de la resolución del reto al que se le enfrenta.

Lógicamente este tipo de metodologías son más factibles de aplicar en entornos reducidos. Indicar que en el curso 2005-2006 cuando se realizó la experiencia, había matriculados 30 alumnos de los cuales 22 acudían regularmente a clase. Este reducido número posibilitó el desarrollo de la experiencia.

A continuación se enunciarán los objetivos perseguidos:

- Potenciar las capacidades y habilidades de trabajo en equipo, diálogo, reflexión, negociación y liderazgo.
- Potenciar la capacidad de expresión oral, tanto en pequeños grupos como en público.
- Aprender a trabajar en entornos inciertos, potenciando su tolerancia a la ambigüedad.
- Potenciar las capacidades de búsqueda autónoma de información y síntesis.
- Adquirir los mismos conocimientos que se exigían en años anteriores mediante la metodología de la clase magistral.

Además de los objetivos de la actividad, desde el punto de vista del docente se perseguían otros objetivos:

- Probar la utilidad de esta metodología en la asignatura.
- Valorar la carga de trabajo para el profesor y para los alumnos.
- Evaluar los resultados tanto desde el punto de vista del profesor como de los alumnos.
- Estudiar el uso de esta metodología en otras partes de la asignatura en función de los resultados.

3. El diseño del caso.

Para diseñar el caso se han seguido las pautas marcadas por Ernesto de los Reyes en [3], aunque adaptadas a las necesidades del autor. También señalar que justo antes de emprender esta tarea el autor había recibido un curso sobre esta metodología impartido por el Instituto de Ciencias de la Educación de la Universidad de Zaragoza, lo que permitió conocer más en profundidad la metodología así como algunas experiencias aunque relacionadas con otras disciplinas (principalmente psicología).

Previamente al diseño se debe valorar si se va a utilizar la metodología en toda la asignatura, o sólo en algunas partes. Desde esta experiencia se recomienda probar primero con la parte que parezca más sencilla la aplicación de esta metodología, y en función de los resultados plantearse metas más ambiciosas. Además, y en el caso de que la experiencia sea exitosa, se puede intentar motivar a otros compañeros para que vayan utilizando esta metodología en otras asignaturas.

El primer paso para diseñar un caso debe ser definir los objetivos, no sólo los relativos a competencias sino también los relacionados con los conceptos o procedimientos que deben comprender y manejar los alumnos.

El segundo paso es plantearse el perfil de los destinatarios del caso, para poder ajustar el nivel de dificultad y conseguir la máxima motivación posible. En este punto cabe señalar que como se indica en [4] citando la curva de De Treville, un reto de dificultad baja (caso trivial) no motivará la curiosidad de los alumnos y el aprendizaje será muy bajo; por el contrario, un nivel de dificultad muy elevado creará sensación de frustración y el aprendizaje también será prácticamente nulo. Por todo ello, se intentará buscar un nivel de dificultad en el que el reto sea lo suficientemente elevado que motive la curiosidad y el interés por aprender, pero que no llegue a sobrepasar el límite de lo inalcanzable para la mayoría del grupo. En la experiencia que nos ocupa, ya ha sido definido el perfil, alumnos de segundo cuatrimestre de segundo curso sin conocimientos previos de seguridad.

El siguiente paso es plantear el problema, teniendo en cuenta que implique todos los objetivos de conocimiento marcados, y además

permita la realización de todas las actividades asociadas.

Llegados a este punto, y alterando el orden propuesto en [3], se plantean las preguntas que deberán resolver los alumnos. Se deberá tener en cuenta que haya tanto preguntas abiertas (múltiples soluciones) como cerradas. Se comprobará además que se engloban todos los objetivos. Si es necesario se añadirán pistas.

Una vez que seguros de que se cubren todos los objetivos, se pensará una historia que sirva de contexto. Luego se definirán y desarrollarán unos personajes y finalmente se hará un primer borrador.

En este punto, y dependiendo de las capacidades literarias de cada uno, se puede "pulir" la historia intentando dotar al relato de mayor fuerza e intentando despertar el interés del lector.

Para acabar la fase de diseño, se plantearán otras actividades relacionadas con el caso (como debates o participación en foros) y se planificará cómo se va a desarrollar la implantación del caso en clase. Lo que implica la distribución de las horas de clase de que se va a disponer.

Antes de presentar cómo se desarrolló la implantación, señalar que es muy importante, si es posible, calibrar el tipo de alumnos que asisten a la asignatura, por ejemplo si son muy participativos o poco y no intentar conseguir resultados inalcanzables. Es decir, de un grupo poco participativo se puede esperar unos resultados mucho más modestos que de un grupo eminentemente activo. Se procurará fomentar la participación de todos, en la medida de lo posible, para intentar que poco a poco vayan entrando en estas dinámicas, aunque sea en futuras asignaturas.

4. Implantación en el aula

Como se ha indicado, durante el diseño deberá hacerse una planificación detallada, de qué actividades se desarrollarán en el aula, qué actividades deben preparar los alumnos fuera del aula y qué material deberán entregar al profesor para su evaluación.

En este caso la materia del bloque didáctico es un tema genérico de introducción, pero donde deben tratarse varios conceptos muy importantes. El tiempo también es limitado, porque se cuenta con menos de dos semanas de tiempo (5 horas de clase). En la primera de las clases dedicadas al método del caso se explica en qué consiste la metodología, se reparte el material (caso, bibliografía y recursos) y se forman los grupos.

A la hora de formar los grupos, es importante tener en cuenta que deben ser pequeños (en nuestro caso dos de cinco personas y dos de seis personas), y en la medida de lo posible deben ser homogéneos entre sí y equilibrados internamente. Esto no siempre es posible, pero en grupos pequeños y sobre todo al final del cuatrimestre sí que en ocasiones podemos acercarnos a modelos de este tipo, ya que se posee un cierto conocimiento de los alumnos.

Para fomentar el trabajo en grupo, en esta experiencia se propuso que los grupos se repartieran internamente las tareas, y luego dentro de cada grupo las pusieran en común. Esta estrategia también vino motivada en parte por la premura de tiempo y por las fechas, ya que estaba muy cerca el inicio del periodo de exámenes.

Para fomentar la capacidad de síntesis, y de diálogo se puso como norma de obligado cumplimiento un tamaño máximo para las respuestas (una página por pregunta).

Para fomentar la capacidad de diálogo, de negociación y de espíritu crítico, se les referenció el material de dos fuentes distintas, y ellos debían completar las preguntas utilizando lo mejor de cada fuente, y debían valorar cada una de las fuentes y recomendar justificadamente sólo una de ellas.

La cuestión del liderazgo se dejó libre, pero se observó que en seguida surgió un líder en cada grupo (si se repite la actividad, se puede designar cada vez un portavoz, que normalmente suele acaparar el rol de líder).

Finalmente, se les indicaba que entregarían una única respuesta entre todos los grupos, que debía ser consensuada. Aquí surgen dos problemas: el primero es cómo se evalúa a cada alumno y el segundo qué pasa si no se ponen de acuerdo. Respecto a la evaluación, como era una prueba piloto se optó por dar una valoración igual a todos los participantes (esto no se desveló hasta el final, porque se recogió el material por separado de cada grupo), a añadir sobre la nota final (0,1 puntos); en el examen final este tema entraba como uno más, y los alumnos respondían individualmente a sus preguntas y así quedó

matizada la nota; para que se pusieran de acuerdo en una sola respuesta fue fundamental el trabajo del profesor que encauzó el debate, y al final se consiguió un consenso relativamente amplio entre los alumnos.

También se habilitó un espacio en el campus virtual para que pudieran trabajar los grupos, pero en este caso su utilización fue prácticamente nula.

El caso consistía en una pequeña historia en la que se describía el entorno de trabajo de una empresa a la que el alumno llegaba porque acababan de despedir al administrador de la red por un robo de información "secreta" y él debía "descubrir" los problemas de seguridad que tenía la empresa y proponer soluciones. El texto incluía pistas, por ejemplo entrevistas a los trabajadores, y una descripción detallada de la red, además de diverso material como listados de acceso a servidores (dirección IP, nombre de usuario, hora de la entrada), datos de controles de acceso físico, etc… Con la información del texto, una vez estudiado el material complementario, se podían responder las preguntas cerradas. Señalar, que se ha constatado que el texto es un buen medio de introducción de otros temas más genéricos que se deseaban tratar, y en lugar de presentarlos como preguntas directas, se introducían en el texto como peticiones específicas de la dirección o de los departamentos (esto permitía introducir aspectos de confidencialidad, autentificación, integridad y no repudio).

Se dedicaron: una hora a la presentación, lectura del caso, explicación de las preguntas y formación de grupos; dos horas al trabajo de los alumnos consensuando las respuestas; una hora a debate entre los grupos y una hora de clase magistral para repasar, afianzar, resumir y clarificar todos los conceptos tratados. Los alumnos dicen que trabajaron una media de diez horas fuera del aula entre las dos semanas.

5. Resultados

Para obtener la opinión de los alumnos se diseñaron unas encuestas, que incluían la valoración numérica de diversos aspectos (valoración general, amenidad, utilidad, claridad del planteamiento de la actividad), así como preguntas abiertas como valorar brevemente la actividad, indicar lo que se considera lo más interesante y lo que menos, lo que se quitaría y lo que se añadiría.

De las respuestas numéricas, donde 5 es lo mejor y 1 lo peor, se obtuvieron los siguientes resultados, mostrados en la figura 1:

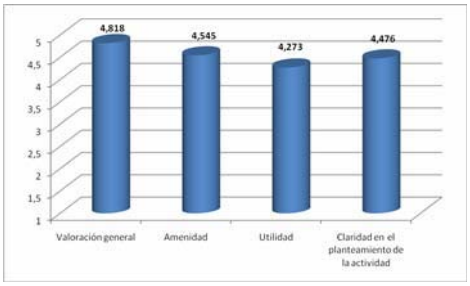


Figura 1. Resultados numéricos encuestas

Del análisis de los resultados numéricos destacar la elevada puntación obtenida en las 22 encuestas recogidas donde la valoración fue siempre y en todas las preguntas 4 o 5 (excepto en una encuesta que se calificaron con un 3 las dos últimas cuestiones, y otra que en utilidad estaba en blanco).

En cuanto a las cuestiones abiertas, una gran mayoría consideraba la actividad como "muy interesante", "muy útil" y la palabra que más apareció fue "novedosa". También afirmaban que habían ampliado mucho sus conocimientos y que al trabajarlo de esta manera se les "quedaba mejor". Algunos alumnos sugerían trabajar más módulos con esta metodología.

En la parte negativa, una gran mayoría señalaba el poco tiempo para realizar la actividad, y que les hubiera gustado profundizar más en el tema porque además muchos calificaban ese tema como "muy útil".

Por parte del profesor constatar la gran carga de trabajo que lleva este tipo de actividades, en este caso en torno a 30 horas, teniendo en cuenta que era el primero que se preparaba. Estas horas se reparten entre la búsqueda de material adecuado, en este caso se dio a los alumnos hasta tres fuentes, la preparación de las preguntas para obligar a buscar en las tres fuentes, la preparación en sí del caso y las tutorías en las que los alumnos buscaban un poco de ayuda.

La parte más compleja es sin duda planificar y diseñar el caso para que no se queden con lo evidente y vayan avanzando hasta buscar la solución y sobre todo el trabajo de dinamizador del debate pero sin intervenir activamente, pero intentando que participen todos los alumnos.

Lo más difícil de conseguir es la motivación. En este caso se recurrió a una historia de espionaje industrial con *hackers* de por medio que suscitó muchas preguntas y parece que motivó bastante.

Respecto a las calificaciones, los resultados fueron muy similares a los de años anteriores, con metodología "tradicional" (clase magistral).

6. Conclusión

En general la satisfacción de los alumnos parece alta, y la utilidad del método evidente, sobre todo en lo referente al desarrollo de habilidades (destaca de manera especial el trabajo en grupo).

Un tema a mejorar es el de la evaluación, y es que en condiciones normales podría plantearse la evaluación de esta parte a cada grupo, obteniendo una calificación y eliminando materia; o bien, en una parte del examen poner preguntas referentes al caso trabajado en clase, para los alumnos que hayan seguido este método.

También sería deseable repetir la actividad con más tiempo, y probar a dar menos bibliografía a los alumnos para fomentar su capacidad de búsqueda autónoma de información.

Para el próximo curso se ha planteado mejorar el caso de estudio (ampliar la historia, y generar listados más amplios, ampliar las transcripciones de las conversaciones, preparar correos electrónicos impresos en papel intentando crear una atmósfera lo más real posible, y cambiar un poco el problema central y la solución). Respecto a la evaluación, se pretende ampliar el cuestionario de opinión de la actividad para valorar tanto la dedicación del alumno como su opinión de diferentes apartados de la actividad.

Se descarta para el próximo curso el ampliar el método a otras partes de la asignatura, a expensas de estudiar los nuevos resultados que se obtengan con la implementación correspondiente a este curso académico (se realizará en Mayo).

Referencias

[1] *Libro Blanco. Título de grado en ingeniería informática*. Agencia Nacional de Evaluación de la Calidad y Acreditación, 2005. (Disponible on-line en http://www.aneca.es/modal_eval/docs/librobllanco_jun05_inform%E1tica.pdf)

[2] Página web del proyecto Tunning. *http://www.relint.deusto.es/TuningProject/index.htm*. Coordinada por la universidad de Desusto (España). Fecha última consulta: 30 de agosto de 2006. [on-line]

[3] Reyes, E. *Introducción al estudio de casos como método de enseñanza.* Instituto de Ciencias de la Educación. Universidad de Zaragoza, 2005.

[4] *The case method.* Disponible en http://www.hbs.edu/case. Harvard Business School. Fecha última consulta: enero 2007. [on-line]

La relevancia de los documentos en Google: Una actividad común de las asignaturas Programación y Computación Numérica

Assumpció Casanova, Juan Garayoa, Francisco Marqués, Vicente Vidal
Escuela Técnica Superior de Informática Aplicada
Dpto. de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
Camino de Vera s/n, 46022 Valencia
{casanova, jgarayoa, pmarques, vvidal}@dsic.upv.es

Resumen

En este artículo se presenta una experiencia docente llevada a cabo por un grupo de profesores de las asignaturas de Programación (PRG) y Computación Numérica (CNU) del primer curso de la Escuela Técnica Superior de Informática Aplicada (ETSIAp), de la Universidad Politécnica de Valencia (UPV).

La experiencia tenía como objetivo intentar hacer ver a los alumnos la relación entre asignaturas distintas, para evidenciar que las mismas no son necesariamente compartimentos estancos. Para ello se les planteó un problema en el que tenían que utilizar herramientas de ambas asignaturas, en concreto herramientas de programación en Java y algoritmos matriciales de computación numérica.

El problema escogido fue el cálculo de la relevancia o importancia de los documentos en la web, de acuerdo con el criterio que usa el buscador Google para presentar los resultados de una búsqueda. Este problema permitía trabajar aspectos significativos de ambas materias, al mismo tiempo que se consideraba de interés para el alumno, dada la actualidad de dicho buscador.

Para resolver el problema era imprescindible la colaboración entre distintos equipos que resolviesen cada una de las partes del mismo. De ahí la necesidad de definir unos requisitos para cada parte del trabajo y para realizar la tarea en equipo.

Aunque se reconoce que es materialmente imposible que todos los alumnos de los grupos implicados participen de la experiencia, sí que se juzga muy interesante la exposición del problema al conjunto de toda la clase y el planteamiento de los algoritmos necesarios para resolverlo.

1. Descripción de las asignaturas

La asignatura PRG es de primer curso troncal, en las titulaciones de ITIG/ITIS. Tiene 12 créditos y duración anual. Sus objetivos consisten en que el alumno sea capaz de diseñar, analizar, implementar, y validar soluciones algorítmicas eficientes para problemas específicos en el ámbito de la programación a pequeña escala.

Más concretamente, se pretende que el alumno sea capaz de resolver problemas sencillos (numéricos básicos, recorrido y búsqueda, ordenación,…) en un modelo imperativo, mediante iteración y recursión. Adicionalmente, el alumno debe saber usar arrays y definir y manipular clases de objetos, a nivel básico. Por último, se espera que el alumno sea capaz de analizar y medir el tiempo de ejecución de los algoritmos que diseña. El lenguaje de programación utilizado es Java.

La asignatura tiene su continuación, en segundo curso, en las asignaturas Estructuras de Datos y Algoritmos, y Metodología y Tecnología de la Programación, en donde se introducen las características más relevantes de la orientación a objetos, como la herencia, así como el diseño de la arquitectura de programas.

La asignatura CNU es de primer curso troncal, en las titulaciones de ITIG/ITIS. Tiene 4,5 créditos y se imparte en el segundo cuatrimestre.

El objetivo general de la asignatura es que el alumno conozca los métodos numéricos básicos que se utilizan habitualmente en la resolución de problemas científicos y de ingeniería, y sea capaz de entender, describir, analizar y programar eficientemente dichos métodos. Para ello, es necesario trabajar diversos aspectos matemáticos

(planteamiento del problema a resolver), algorítmicos (técnicas más comunes) y de implementación (programación eficiente). El análisis de los métodos se ha de realizar desde dos puntos de vista: por un lado, los errores producidos en los cálculos, debidos a la precisión finita del computador, y por otro, la complejidad temporal y espacial de los algoritmos. Además, se pretende que el alumno mejore sus habilidades en el desarrollo de programas, utilizando lenguajes específicos como MATLAB.

2. Motivación

Las asignaturas de primer curso se resienten en general del escaso hábito de trabajo de los estudiantes actuales, y de una tendencia al absentismo cada vez más instaurada.

Además, en el caso de PRG, para muchos alumnos es la primera vez que se enfrentan a la tarea de programar. En el caso de CNU, está ubicada en el segundo cuatrimestre, cursándose simultáneamente con otras asignaturas anuales consideradas difíciles por los alumnos, como PRG y Matemática Discreta y Álgebra, lo que, según los profesores, propicia el abandono.

Como consecuencia de todo ello, se observa un alto número de no presentados y suspensos, en mayor o menor grado, en ambas asignaturas.

En el curso 2004/05, los profesores de las dos asignaturas iniciaron una colaboración con el objetivo de coordinar materias, optimizar el esfuerzo del alumno, e intentar reducir el absentismo.

Existía el antecedente de alguna experiencia puntual de integración de conocimientos [6], realizada en el ámbito de un Proyecto de Innovación Educativa de la UPV, consistente en la impartición de una clase conjunta por profesores de ambas asignaturas.

En el curso 2004/05, las acciones llevadas a cabo consistieron en alguna reestructuración del temario que facilitara la coordinación, y el planteamiento de una práctica de laboratorio de PRG para programar en JAVA y medir los tiempos de ejecución de ciertas operaciones matriciales que se programan en MATLAB en una práctica de CNU, como refuerzo por redundancia.

Una de las conclusiones principales fue que la propia estructura y calendario de las asignaturas dejaban márgenes temporales demasiado apretados como para poder hacer la coordinación más efectiva.

Es por ello que en el curso siguiente 2005/06, se consideró la conveniencia de plantear alguna acción más flexible temporalmente, y que al mismo tiempo motivara más al alumno.

Para este propósito, se pensó en enfrentar al alumno a un problema real y de actualidad, convenientemente simplificado, con el incentivo de una puntuación adicional en una o en las dos asignaturas.

Se escogió un problema relacionado con el buscador Google, que se describe en el siguiente apartado, y que presenta aspectos relacionados con las materias de ambas asignaturas. Con ello se pretendía:

- Demostrar la posible interrelación entre asignaturas distintas cuando se trata de resolver un problema concreto.
- Disminuir el alto nivel de absentismo en clase.
- Aumentar el porcentaje de alumnos aprobados, así como la nota media de estos alumnos.

3. Descripción del trabajo

3.1. Objetivos docentes

Además de conseguir la motivación del alumnado, el problema planteado debía responder a diversos objetivos docentes. Algunos de índole general, como:

- Afrontar un problema de relativa complejidad, abordando su resolución de una manera sistemática.
- Realizar un trabajo de integración de conocimientos adquiridos en diversas materias y asignaturas.
- Sistematizar el trabajo en equipo.
- Dejar planteados problemas cuyo estudio se abordara en asignaturas más avanzadas.

Y objetivos específicos de las asignaturas, como:

- Realizar un trabajo que exigiera el uso de las herramientas básicas de programación, y en concreto del lenguaje Java, presentadas en el primer curso.

- Saber leer la especificación de un problema de un ámbito general o especializado, y la descripción de su resolución según los conocimientos propios de dicho ámbito.
- Saber aplicar las técnicas de cálculo numérico y de programación adecuadas que permitieran implementar dicha resolución.
- Saber aplicar estas técnicas para incluso mejorar o dar alternativas a algunos aspectos de las resoluciones propuestas desde ámbitos no informáticos.
- Saber construir librerías (clases) de operaciones generales, en concreto una de operaciones matriz-vector, que se utilicen en la resolución de distintos problemas.
- Conocer y utilizar distintas formas de almacenamiento de datos.
- Utilizar operaciones para matrices dispersas.

3.2. Enunciado del problema

Supóngase que se dispone de un programa buscador, que es capaz de buscar ciertos ítems en una red de páginas web, a requerimiento del usuario. Se plantea el problema de en qué orden presentar los resultados de la búsqueda, de una forma que pueda ser útil al usuario.

En [3] y [1] se presenta el modelo utilizado en la experiencia para abordar el problema, basado en el utilizado por Google.

Según este modelo, a cada documento se le asocia una importancia, que se considera función de los documentos que lo referencian, y de la importancia de cada uno de estos.

Más concretamente, la importancia de un documento es proporcional a la suma de las importancias de los documentos que lo referencian.

Si se conoce el grafo de referencias mutuas entre los documentos, el cálculo de las importancias aparece como la resolución de un sistema de ecuaciones lineales, que conduce a un cálculo de autovectores y autovalores.

Estas importancias constituyen el criterio de ordenación de los documentos obtenidos como resultado de una búsqueda.

Como simplificación del problema de búsqueda en la red, se elaboró una pequeña "red" local o "microrred", de documentos html fuertemente referenciados entre sí, objeto del problema.

De esta forma se evitaba el problema de manejar grandes volúmenes de información, dispersa y remota, aunque se creaba en el alumno la expectativa de considerar estos aspectos más adelante en su formación.

Para la elaboración de la microrred se contó con la ayuda de un becario que generó la base de datos documental (becario dotado por el Plan de Acciones para la Convergencia Europea, de la UPV).

Para explotar las posibilidades que presenta este problema al nivel de las asignaturas de primer curso, se planteó en forma de un proyecto en el que cabía distinguir las siguientes fases:

1. Obtención de la matriz de adyacencias asociada a la base de datos documental, que contuviera la información sobre las referencias entre los distintos documentos.
2. Cálculo del vector de importancias, que se reduce a la resolución de un problema de autovalores.
3. Evaluación del sistema de información. Esta evaluación consistía en la búsqueda de un ítem en las páginas de la microrred, y la presentación de los resultados ordenados por la importancia de las fuentes.

La resolución del problema se descomponía en cuatro tareas asociadas a las distintas fases del proyecto, y que podían resolverse de forma independiente por un alumno. Estas tareas, si se ajustaban a las especificaciones dadas, proporcionaban un conjunto de librerías y programas (clases) que resolvían el problema enunciado.

Las tareas planteadas a los alumnos fueron las siguientes:

1. Obtención del registro de referencias mutuas para el conjunto de páginas de la microrred: Búsqueda de las referencias en cada documento al resto de documentos. A partir de los resultados de esta búsqueda, se debía construir un fichero con la descripción del grafo de referencias en la microrred. Esta tarea resolvía la primera fase del proyecto.

2. Construcción de librerías de operaciones matriciales necesarias para la resolución del problema de autovalores: producto matriz-vector, producto escalar, norma, tanto en representación densa como dispersa. Esta tarea formaba parte de la segunda fase del proyecto.
3. Medida de la importancia de las páginas web de la microrred mediante la utilización de algoritmos de cálculo de autovalores, a partir del grafo de referencias, representado por la correspondiente matriz de adyacencias. Como resultado, se generaba un fichero con las importancias calculadas. Esta tarea, junto con la anterior completaban la segunda fase del proyecto.
4. Búsqueda de un ítem en las páginas de la microrred, y presentación de las citas obtenidas, ordenadas por la importancia de las fuentes. Esta tarea resolvía la tercera fase del proyecto.

Esta descomposición proporcionaba un conjunto de tareas, siendo cada una de ellas de dimensiones ajustadas a la carga deseada para el trabajo individual de un alumno. Además, concretaba los enunciados de forma que se revisaban los siguientes contenidos concretos de las asignaturas:

- Se recalcaba la importancia de las especificaciones de cada uno de los apartados para poder integrarlos de forma que se resolviera el problema en su globalidad.
- Se planteaban aplicaciones no triviales de algoritmos básicos de programación: búsqueda de palabras en un conjunto de textos, con estrategias eficientes; ordenación de ítems no elementales; algoritmos iterativos sobre arrays de más de una dimensión.
- Se planteaba la aplicación de un método numérico a la resolución de un problema aparecido en un ámbito eminentemente informático.
- Se proponía la utilización de matrices dispersas como solución al problema del almacenamiento de datos estructurados en matrices de grandes dimensiones.
- Se presentaban las especificaciones para la construcción de librerías (clases) de operaciones matriciales, tanto para matrices densas como para matrices dispersas.

El enunciado detallado de las tareas, junto con la microrred de prueba, se puede encontrar en [1].

3.3. Método de evaluación

El carácter del trabajo era voluntario, y dado el calendario de las asignaturas, se planteaba su realización mediado el segundo cuatrimestre.

El alumno podía escoger una tarea, y su desarrollo le podía suponer a lo sumo 1 punto adicional a la nota en una de las dos asignaturas, a su elección: 1 punto en CNU si había escogido entre las tareas 2 y 3, o 1 punto en PRG si había escogido cualquiera de las tareas.

Los alumnos podían integrarse en un grupo de 4, con el objetivo de completar todo el proyecto, en cuyo caso les supondría, una nota adicional en ambas asignaturas, de 1 punto como mucho en cada una.

4. Resultados

El número de alumnos que se involucraron en la experiencia fue relativamente bajo. Se detectaron 3 tipos de alumnos:

- Alumnos aventajados, movidos por su curiosidad o afición, y aspiración a una buena nota.
- Alumnos repetidores, que vieron en la experiencia una forma alternativa de repasar la asignatura pendiente.
- Alumnos con dificultades en la asignatura, que intentaron mejorar sus posibilidades de aprobar.

Los dos primeros grupos fueron en general los más numerosos y activos, constituyendo grupos que completaron el desarrollo del proyecto, y participaron en una presentación pública de resultados.

El último grupo, menos ambicioso en el desarrollo del proyecto, fue prácticamente residual.

Se constató la dificultad de proponer trabajos en grupo. Normalmente, se completó la fase de trabajo individual, pero se echó de menos un organizador que armonizase el trabajo del grupo. En futuras ediciones se debería cuidar más este aspecto.

En la asignatura PRG, se propuso el trabajo a tres grupos de teoría, de los que se presentaron al examen de junio un total de 118 alumnos. De ellos intervinieron 21 alumnos en la experiencia, de los cuales, 9 se incorporaron en algún grupo de los que completó la realización del proyecto, unificando su trabajo individual con el de sus compañeros y verificando el funcionamiento global.

En la asignatura CNU, se propuso el trabajo a dos grupos de teoría, de los que se presentaron en junio un total de 83 alumnos. De ellos participaron 21 alumnos en la realización de una parte individual del trabajo. Cuatro de los alumnos se integraron en un grupo.

En conjunto, se formaron un total de 3 grupos de trabajo, constituidos por alumnos matriculados en ambas asignaturas, o en una sola en el caso de algún repetidor.

Esta minoría de alumnos fue la que sacó mayor provecho de la experiencia, pues mostraron iniciativa para plantear problemas y cuestiones, que, trabajados en las tutorías, les permitió profundizar en algunos conocimientos.

Aunque el trabajo con estos pocos grupos de alumnos fue muy satisfactorio para los profesores y los propios alumnos, este método de trabajo es materialmente imposible generalizarlo, dados los ratios alumno/profesor típicos de una escuela como la ETSIAp.

Finalmente, los 3 grupos llevaron a cabo una exposición y discusión pública del trabajo desarrollado y los resultados obtenidos. Dicha exposición tuvo lugar fuera del horario lectivo debido a lo ajustado del calendario y programa de las asignaturas, que deben cumplir un estricto cronograma de coordinación entre teoría y prácticas. En algún grupo, el profesor tuvo la oportunidad de presentar y discutir el problema con cierto detalle, pudiendo incidir al menos en la aplicabilidad de algunos métodos y conceptos estudiados en las asignaturas.

5. Conclusiones

En los siguientes puntos se repasan algunas constataciones acerca de la experiencia:

- Se ha presentado a los alumnos de algunos grupos, un problema cuya resolución necesitaba integrar conocimientos de ambas asignaturas.
- Como se podía haber supuesto, solamente un grupo reducido se interesó en la realización de la experiencia.
- Se ha comprobado la imposibilidad de atender a grupos de alumnos más numerosos dadas las restricciones de tiempo.
- Se constata la dificultad de trabajar en equipo, la mayoría de alumnos han acabado resolviendo por completo sus problemas y no coordinando sus trabajos.
- La presentación de los resultados obtenidos se hizo fuera del horario lectivo, con lo que la mayoría de alumnos perdió la oportunidad de comprobar el resultado.
- No se logró reducir el porcentaje de suspendidos de forma significativa, ya que los alumnos que participaron en la experiencia eran los mejores.
- Sí que se logró subir la nota media de los alumnos aprobados, debido a la puntuación adicional conseguida por los alumnos participantes.

En conclusión, el trabajo propuesto parece adecuado para estimular el estudio en profundidad de los alumnos más aventajados, pero no parece conseguir el objetivo de interesar al alumno medio, ni de reducir el absentismo.

Si bien en cursos posteriores, el alumno dispone de recursos suficientes como para trabajar de forma más autónoma (véanse, por ejemplo, las experiencias descritas en [2], [4]), en el alumno de primer curso se detecta una falta de hábitos y de conocimientos básicos, por lo que debe ser más guiado. Esta conclusión parece desprenderse también de los resultados de un curso piloto de innovación docente en primero, que promovió la ETSIAp durante el curso 2005/06, y cuyos resultados para la asignatura de Programación se discuten en [5]. Dicha experiencia se basaba, entre otros puntos, en motivar al alumno, induciéndole a realizar parte del trabajo práctico en forma de un pequeño proyecto a desarrollar de forma incremental a lo largo del curso. Los resultados observados no fueron lo buenos que se esperaba en cuanto a la implicación de los alumnos en un trabajo constante, a pesar del considerable esfuerzo invertido por el profesor en largas sesiones de tutorías.

En consecuencia, una metodología de trabajo que suponga mayor proporción de docencia no

presencial, parece adecuada si se trabaja con grupos pequeños, y con alumnos más maduros que los de primer curso.

Aunque en esta experiencia el proyecto propuesto se basaba en un problema muy concreto, existen diversas aplicaciones de ingeniería, cuya programación requiere el uso de métodos numéricos: gráficos, recuperación de información, etcétera, y que no necesitan mayores conocimientos que los adquiridos en primer curso.

Esto sugiere a los autores el diseño de asignaturas de carácter práctico, que se podrían impartir desde el tercer semestre, y cuyos objetivos específicos fueran la integración de conocimientos, y la adquisición de destrezas transversales como el trabajo en grupo, y la exposición oral.

Agradecimientos

Los autores expresan su agradecimiento al becario Eloy Romero Alcalde, alumno de la Facultad de Informática de la UPV, por su estimable trabajo en la creación de la red de documentos html que sirvieron de prueba en este proyecto.

Igualmente, se desea agradecer a los alumnos participantes, su interés y entusiasmo.

Por último, se agradece a la ETSIAp su implicación y estímulo en estos proyectos de innovación docente.

Referencias

[1] Casanova, A., Garayoa, J., Marqués, F. V. Vidal, V. *Enunciado del trabajo conjunto Computación Numérica, Programación*. ETSIAp. Curso 2005/2006. http://www.dsic.upv.es/~casanova/prg-cnu07

[2] Díaz. A., y otros. *Experiencia educativa entre varias asignaturas*. Actas de las XI Jornadas de Enseñanza Universitaria de la Informática. Villaviciosa de Odón, 2005.

[3] Fernández, P. *El secreto de Google y el Álgebra lineal*. Bol. Soc. Esp. Mat. Apl. nº30 (2004), 115-141.

[4] Garrido, P.P., Migallón, H.F. *Cómo motivar al alumno entrelazando las asignaturas Programación Avanzada y Tecnología de Computadores*. Actas de las IX Jornadas de Enseñanza Universitaria de la Informática. Cádiz, 2003.

[5] Gómez, J. A. *Nuevo enfoque de Programación de primer curso*. Póster. Jornadas Nacionales de Intercambio de Experiencias Piloto de Implantación de Metodologías ECTS. Badajoz, 2006.

[6] Pérez, J.L., Rosso, P. *Integración de conocimientos de Programación y Cálculo Numérico en la Universidad Politécnica de Valencia*. Actas de las VII Jornadas de Enseñanza Universitaria de la Informática. Palma de Mallorca, 2001.

Hacia un Sistema Inteligente basado en Mapas Conceptuales Evolucionados para la Automatización de un Aprendizaje Significativo. Aplicación a la Enseñanza Universitaria de la Jerarquía de Memoria

L. Moreno, E.J. Gonzalez, J.D. Piñeiro, B. Popescu, A. Hamilton, J. Sigut, J. Torres, J. Toledo, J. Merino, C. González
Dpto. de Ingeniería de Sistemas y Automática y Arquitectura y Tecnología de Computadores
Universidad de La Laguna
Facultad de Física y Matemáticas, 38207 Tenerife
ejgonzal@ull.es

Resumen

El principal objetivo de esta ponencia consiste en plasmar la búsqueda de un sistema inteligente para apoyo a la docencia en un proceso de aprendizaje significativo [9]. Como dominio concreto, hemos elegido el campo de la jerarquía de memoria por su complejidad limitada pero que a la vez constituye un dominio lo suficientemente general para ser enseñado en diferentes asignaturas y grados de detalle. Además, los autores hemos desarrollado varios trabajos en esta área, incluyendo varios simuladores que han sido empleados como recursos docentes. Buscando la mejora continua de la docencia, se llevó a cabo un proceso de investigación-acción, dando como fruto la idea de elaborar el mencionado sistema inteligente, incluso para ser independiente del dominio de aplicación. Para esta aplicación, hemos desarrollado una variación de los tradicionales mapas conceptuales, a la que hemos denominado Grafo Instruccional Conceptual Pedagógico (*Pedagogical Concept Instructional Graph*, PCIG) que permite la planificación del aprendizaje y la enseñanza de un tema específico. Después de que el profesor haya diseñado el PCIG específico de la materia a enseñar, el sistema experto hará uso de un test adaptivo que generará una serie de preguntas siguiendo la estructura jerárquica descrita en el PCIG. De la respuesta obtenida por cada estudiante se puede obtener un mapa conceptual individualizado que describe lo que cada alumno conoce a priori de la materia a aprender.

1. Introducción

Se puede considerar el punto de partida de este trabajo como un proceso de investigación-acción. Este es un proceso bien documentado (la idea original se le atribuye a Kart Lewin en los años 40) de reflexión, análisis y discusión para la mejora de la enseñanza mediante actividades colaborativas que busquen soluciones para el día a día en el ámbito docente [1],[6],[7],[8].

En esta línea, los autores – profesores e investigadores – pretendieron en un principio aplicar las ventajas de un proceso de investigación-acción a las asignaturas que imparten, un amplio abanico que abarca desde la Arquitectura de Computadores hasta el Control Digital. A pesar de esta diversidad, varias de estas asignaturas ofrecen un solapamiento de temas, por ejemplo la cache y la memoria virtual, con lo que parece adecuado iniciar un proceso de investigación-acción en búsqueda de soluciones comunes. Para ello se celebraron numerosas reuniones en las que los participantes han examinado de forma sistemática su propia experiencia docente, empleando para ello técnicas propias del terreno de la investigación.

Uno de los primeros campos a los que hemos intentado aplicar este proceso es el de la jerarquía de memoria, un tema que aparece inevitablemente en la docencia de asignaturas como Sistemas Operativos, Estructura y Arquitectura de Computadores.

Tras dichas reuniones, surgió la idea del desarrollo de un sistema inteligente que pudiese ayudar a la docencia, averiguando lo que el

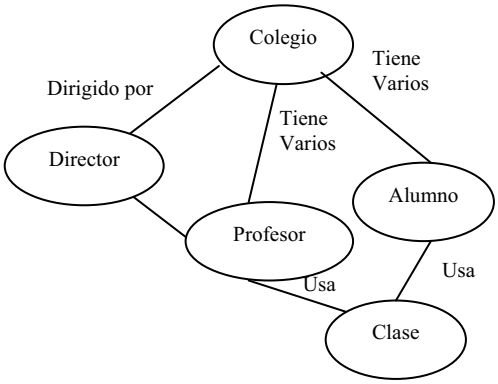


Figura 1. Ejemplo simple de mapa conceptual

alumno conoce del tema a enseñar y con qué profundidad en un momento determinado. A partir de esa información se diseñará un aprendizaje significativo particularizado para cada alumno. Emplearemos para este propósito mapas conceptuales, ya que permite dividir el tema en regiones más pequeñas, facilitando la clasificación del alumnado según su conocimiento previo, y con ello el aprendizaje significativo.

Los mapas conceptuales son unas herramientas gráficas para la organización y representación del conocimiento. Cada mapa conceptual se compone de conceptos, normalmente representados dentro de círculos o rectángulos, unidos jerárquicamente usando relaciones o preposiciones que pueden a su vez estar etiquetada mediante un enunciado descriptivo como "es parte de", "depende de", etc. [2],[3],[5],[11],[12]. En la Figura 1 se muestra un ejemplo simple de mapa conceptual.

Existen numerosas referencias bibliográficas que demuestran las ventajas del empleo de mapas conceptuales: mejoran la comprensión y retención de ideas, ayudan a la memorización de conceptos y relaciones, permiten personalizar el aprendizaje, la compartición de conocimiento y refuerzo de las habilidades de aprendizaje. Además, permiten su complementación con otros recursos docentes como son los Objetos de Aprendizaje (*Learning Objects- LOs*) o con otros tipos de actividades como chats, foros, enlaces a páginas web, etc.

Debido a su naturaleza, el empleo de mapas conceptuales se considera extremadamente útil para una docencia basada en aprendizaje significativo. Este tipo de aprendizaje se basa en el conocimiento previo del alumno, dentro del cual se asimilan los nuevos conceptos. De ahí que sea aconsejable estimar el estado de conocimiento actual del alumno antes de proceder a la docencia de nuevos conceptos.

Retornando al diseño del sistema inteligente propuesto en el proceso de investigación-acción, los autores presentamos una evolución de los tradicionales mapas conceptuales a los que hemos llamado Grafo Instruccional Conceptual Pedagógico (*Pedagogical Concept Instructional Graph*, PCIG), y que será explicado con profundidad más adelante. El profesor describiría el conocimiento de la materia mediante uno de estos grafos evolucionados. Tras esto, el sistema inteligente haría uso de un test adaptivo, generando varias preguntas sobre los conceptos incluidos en ese PCIG. De las respuestas que se obtengan, se puede construir un PCIG personalizado para cada alumno conteniendo la información de qué conceptos conoce a priori el alumno y, no menos importante, con qué profundidad. Comparando ambos modelos (el original con el personalizado), el sistema experto determinará un camino de aprendizaje (lo que en las referencias bibliográficas se conoce como *learning path*), basado en casos y personalizado para grupos de alumnos que presenten características similares. Estos caminos de aprendizaje involucran tanto a un conjunto de recursos como los LOs como al uso de teorías pedagógicas ampliamente avaladas por la experiencia, como es el caso del CSCL (*Computer Supported Collaborative Learning*).

Es importante remarcar que la metodología propuesta es totalmente independiente del dominio de la aplicación, aunque a lo largo de esta ponencia haremos referencia a su aplicación en la enseñanza universitaria de la jerarquía de memoria en la asignatura de Sistemas Operativos de la titulación de Ingeniería en Informática de Sistemas.

2. Proceso de investigación-acción y PCIGs

Tal como indicamos en la introducción, la metodología que presentamos en esta ponencia deriva de un proceso de investigación-acción. Este proceso lleva consigo una serie de pasos, descritos ampliamente en la literatura.

1. Identificación del dominio del problema, normalmente condensado en una pregunta a la que el proceso trata de dar respuesta. En el caso que nos ocupa, la pregunta ha sido "¿Qué serie de recursos didácticos podemos diseñar e implementar para obtener mejores resultados en la docencia del tema de jerarquía de memoria en nuestras asignaturas, siendo adaptables al nivel y contexto de cada asignatura?".
2. Recogida y organización de datos. En nuestro caso hemos recogido información principalmente mediante ejemplos del trabajo de estudiantes de las asignaturas involucradas, proyectos fin de carrera, actas de las reuniones y artículos en revistas del ámbito educativo (por ejemplo *IEEE Transactions on Education* y *Computers & Education*) o congresos/jornadas (principalmente actas de la JENUI de ediciones anteriores). Es de destacar que, al involucrar a alumnos de varias asignaturas, hemos conseguido una muestra rica en variedad.
3. Interpretación de los datos. Debido a la naturaleza de los datos recogidos, estos datos son principalmente cualitativos, por lo que su interpretación debe ser realizada con cuidado. Una de las conclusiones más sobresalientes es que una mayoría de los alumnos analizados echan en falta algún tipo de recurso informático que les guiase en el proceso de aprendizaje. No nos debe extrañar esta conclusión, ya que estamos analizando asignaturas de Informática, donde los alumnos están habituados a tratar con todo tipo de software.
4. Acción basada en los datos. A partir de la interpretación que indicamos en el punto 3 y tras el análisis de diversos artículos aparecidos en revistas de carácter educacional como las mencionadas en el punto 2, hemos elaborado un plan basado en el diseño e implementación de herramientas software donde se incluyan los conocidos 7 principios para la buena práctica educativa identificados por Chickering and Gamson [4] (promover el contacto entre el estudiante y su lugar de estudio, reciprocidad y cooperación entre los estudiantes, aprendizaje activo, realimentación adecuada, énfasis sobre el tiempo de las tareas realizadas, comunicación de altas expectativas en el aprendizaje y respeto de los diversos talentos y formas de aprendizaje).
5. Reflexión. El paso final consiste en la lógica evaluación de los resultados y considerar la posibilidad de refinar el proceso repitiendo los pasos descritos.

Como primer resultado del proceso, varios de los autores de esta ponencia hemos dirigido varios proyectos Fin de Carrera consistentes en el diseño e implementación de simuladores educativos para la enseñanza de la jerarquía de memoria y caché, uniendo de este modo el conocimiento teórico con la experiencia práctica. Entre otros, destacaremos dos simuladores, SIJEM [13] y su evolución MNEME [10]. En su diseño hemos intentado evitar uno de los problemas que suelen criticarse a este tipo de herramientas: ofrecer una versión demasiado simplificada del objeto de estudio, lo que puede llevar a nuevos errores de concepción en el modelo del estudiante, y con ello, fracasar en nuestro objetivo de un aprendizaje significativo. Aunque estos simuladores ofrecen una serie de interesantes características para la enseñanza universitaria de la jerarquía de memoria, el análisis de dichos simuladores cae fuera del objetivo de nuestra ponencia, por lo que animamos a aquel lector interesado para que visite sus respectivas páginas web donde pueden ser descargados de una manera gratuita.

El mero uso de simuladores por sí mismo no es suficiente para la obtención de mejores resultados en la docencia. Se hace necesario desarrollar otras actividades (discusiones de significado, wikis, ejercicios complementarios, presentaciones en clase) que permitan al alumno aumentar su conocimiento de la materia, comprender adecuadamente los conceptos teóricos y aplicar dichos conceptos al mundo real. En nuestro caso hemos incluido ese tipo de actividades en la conocida plataforma educativa Moodle.

Una segunda propuesta para la mejora de nuestra enseñanza consistió en el uso de mapas conceptuales relativos a la jerarquía de memoria. En una primera aproximación, empleamos la habitual actividad de pedir a los alumnos que plasmasen su propio mapa conceptual. Sin embargo, y tras largas reuniones entre los profesores involucrados, desarrollamos un único

mapa conceptual con todos los conceptos involucrados en las asignaturas relacionadas (Sistemas Operativos, Estructura de Computadores y Arquitectura e Ingeniería de Computadores…). De este modo conseguíamos además una mayor coordinación entre las asignaturas, ya que podíamos fijar una especie de fronteras entre los conceptos, evitando el solapamiento de materias (es decir, que a los alumnos se le repitiese varias veces el mismo concepto). A continuación, hemos solicitado a los alumnos que refinasen y completasen el mapa conceptual, incluyendo nuevos conceptos y asociando materiales educativos (teoría que han visto en clase, actividades con simuladores, etc.) a cada concepto que aparece en el mapa. Para esta actividad – desarrollada de forma colaborativa primero en grupos de 3 alumnos y después de forma intergrupal – los alumnos han empleado Compendium, una herramienta de código abierto. Una de las ventajas de esta herramienta es que se puede exportar el mapa conceptual a formato HTML, de tal forma que los alumnos pueden acceder al contenido asociado a cada concepto solamente pulsando el botón del ratón sobre dicho concepto. De este modo, el mapa conceptual les ayuda a estudiar la materia, organizar sus apuntes y preparar nuevo material didáctico que podremos emplear para los próximos cursos. Además, podemos incluir el código HTML generado en el Moodle (o en cualquier otra plataforma de e-learning).

En el contexto de un aprendizaje significativo, sería deseable una herramienta que nos informase de lo que cada alumno conoce en un momento determinado. Esta herramienta permitiría el diseño de un camino de aprendizaje personalizado para cada estudiante una vez que los conceptos-objetivo del proceso han sido definidos. Este camino de aprendizaje puede incluir aquellos recursos o actividades (por ejemplo los simuladores mencionados) que permitiesen a los alumnos llevar a cabo un autoaprendizaje adaptivo. Un buen punto de partida podría ser el mapa conceptual que hemos diseñado anteriormente. Sin embargo, podríamos ser más ambiciosos en nuestro objetivo y buscar una herramienta lo suficientemente general que sirva para un mapa conceptual genérico, no solamente para la jerarquía de memoria. Para que esta generalización sea posible, necesitamos establecer una serie de restricciones a la hora de construir el mapa conceptual. Este es el motivo por el que los autores proponemos un subconjunto de mapas conceptuales, a la que hemos llamado Grafo Instruccional Conceptual Pedagógico (Pedagogical Concept Instructional Graph, PCIG).

Un PCIG no ordena los conceptos según relaciones arbitrarias, tal como sucede en el caso de un mapa conceptual genérico, sino que los conceptos son colocados dependiendo del orden lógico en que deben ser presentados al alumno. Por tanto el PCIG se basará en relaciones del tipo "Concepto A debe ser enseñado antes del Concepto B porque A es necesario para entender adecuadamente B". Como ejemplo, en la Figura 2 mostramos el PCIG referido a la jerarquía de memoria en la asignatura de Sistemas Operativos. Este PCIG muestra, por ejemplo, que "Segmentación paginada" necesita que los conceptos "Segmentación" y "Paginación" sean enseñados con anterioridad. Otro ejemplo es el de "Paginación bajo demanda" que es necesario para el aprendizaje de "Archivos mapeados en memoria", "Reemplazamiento de página", "Asignación de marcos de página", "Copy-on-write" e "Hiperpaginación".

Tras el diseño del PCIG podemos emplearlo para explorar el conocimiento de cada estudiante. Para este propósito emplearemos un sistema experto, el cual describiremos en las siguientes secciones de la ponencia.

3. Determinación del conocimiento actual del estudiante mediante PCIGs

Los autores pretendemos implementar un sistema de enseñanza basado en el uso del ordenador. Para ello necesitamos testear el conocimiento del alumno para posteriormente presentarle contenidos relacionados con la materia a aprender (documentos, simuladores, enlaces, etc.). En el diseño de esta herramienta debemos distinguir dos funcionalidades distintas: la del estudiante y la del administrador de la herramienta.

El estudiante debe pasar los tests y visualizar los contenidos relacionados a algunos conceptos, mientras que el administrador se encarga de fijar los conceptos, las tareas a realizar y las relaciones entre ellos. En esta sección nos centraremos en los aspectos de diseño de esta herramienta.

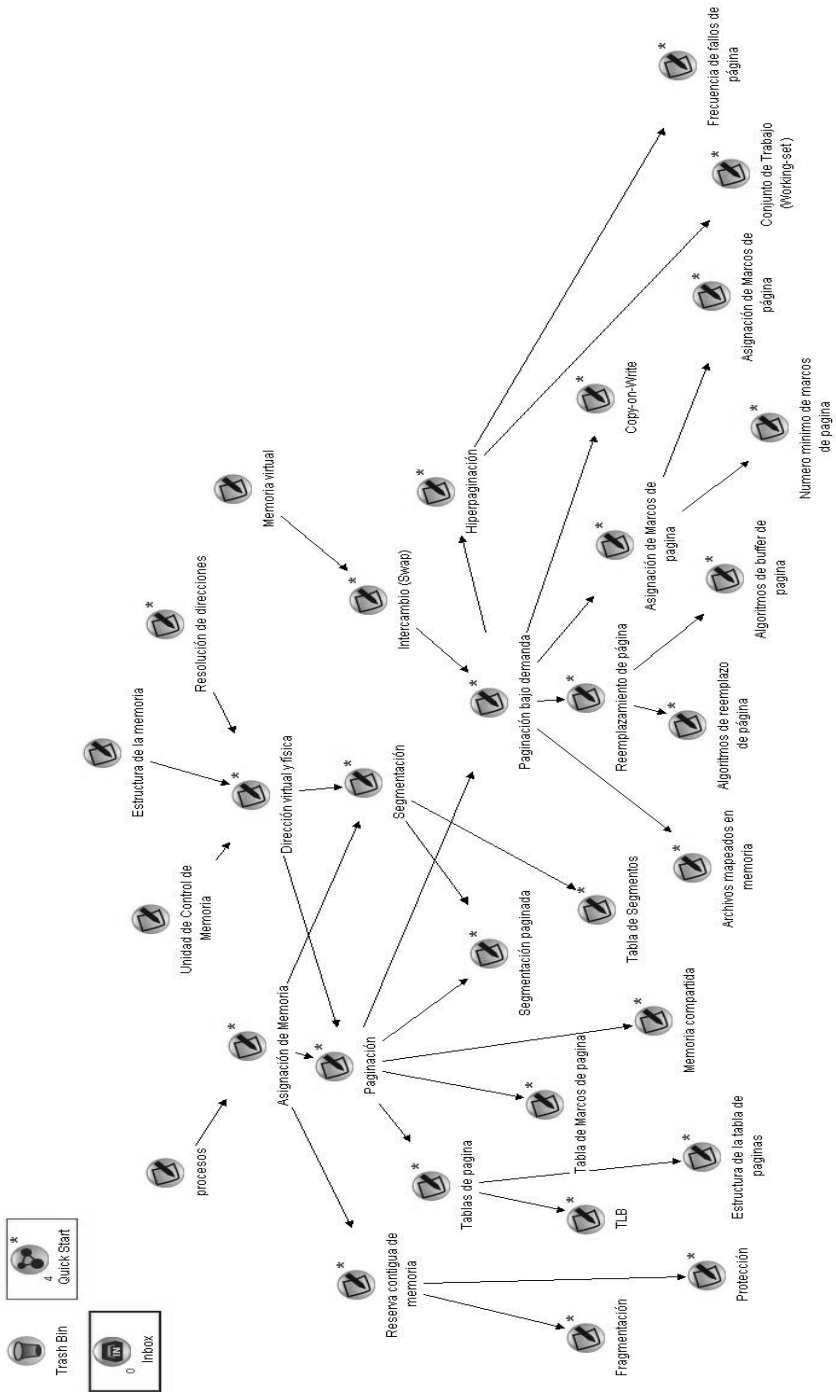


Figura 2. Ejemplo de PCIG aplicado a la jerarquía de memoria para la asignatura de Sistemas Operativos

3.1. Role de administrador

El administrador se encarga de las siguientes funciones:

- Crear el PCIG. Por notación cada enlace se designará como [concepto1, concepto2], que será interpretado como que el concepto1 depende del concepto2, por lo tanto un alumno no puede conocer el concepto1 sin haber visto antes el concepto2.
- Crear las preguntas del test adaptativo, con 4 posibles respuestas y definiendo una serie de factores como
  - Probabilidad a priori P de conocer el concepto.
  - Probabilidad de acertar la pregunta sabiendo el concepto P(p+|c+), lo que llamamos dificultad.
  - Probabilidad de acertar la pregunta no sabiendo el concepto P(p+|c-), lo que denominamos adivinanza.
  - Grado de relación de la pregunta con el concepto.
- Definir los conceptos previos del mapa, esto es, aquellos conceptos que el alumno debe conocer antes de enfrentarse al test (por ejemplo los conceptos que se supone que han visto en otras asignaturas) y los conceptos objetivo – los conceptos que el alumno debe llegar a conocer para aprobar el tema.
- Administración normal del sistema: crear nuevos usuarios, contraseñas, alias, permisos, etc.
- Seguir el estado de cada estudiante, visualizando los tests que cada alumno ha superado o no (la calificación obtenida, las respuestas que ha dado, etc.)
- Crear contenidos relacionados con cada concepto, del mismo modo que en el caso de la creación de las preguntas del test adaptivo, es decir, asignando un grado de relación a cada concepto.

3.2. Role de alumno

El role de alumno comprende las siguientes acciones:

- Modificar los datos de su cuenta (login, contraseña, etc.)
- Comprobar qué tests ha realizado.
- Escoger un concepto de los disponibles para ser visitado en el mapa (conceptos visitados anteriormente pero que no ha conseguido aprobar y que no han sido dados por finalizados por el administrador y conceptos no visitados a los que puede acceder por haber aprobado los conceptos necesarios).
- Realizar los tests en el tiempo prefijado por el administrador.
- Visualizar los resultados del test.

3.3. Arquitectura e implementación

El sistema resultante es una aplicación basada en web, escrita en Java, testeada para el servidor de aplicaciones Tomcat y empleando una base de datos MySQL. El algoritmo para el test adaptivo se basa en los siguientes principios:

- La siguiente pregunta del test se toma de la lista de las preguntas disponibles. Estas preguntas son las relacionadas con el concepto actual y que no estén relacionadas con conceptos que el alumno no haya aprobado.
- La puntuación obtenida al responder una pregunta depende de si el alumno ha respondido correctamente a la pregunta anterior del test (se atenúa de este modo el factor suerte a la hora de responder el test).

Actualmente se han implementado estos principios mediante un parser ANTLR y un algoritmo bayesiano.

- Tomando en cuenta la probabilidad de dar una respuesta acertada a la pregunta según la dificultad de la pregunta y la probabilidad de acertarla "por suerte", se puede definir la nueva probabilidad de conocer adecuadamente el concepto, aplicando el Teorema de Bayes, como

$$P_{a\ posteriori} = P(p+|c+)*P / (P(p+|c+)*P+(1-P)*P(p+|c-)) \quad (1)$$

- La siguiente pregunta se escoge de la lista de las posibles preguntas como aquella pregunta que maximiza una función de utilidad que

tiene en cuenta el resultado de la última cuestión planteada, la calificación obtenida por el alumno y las dificultades de las preguntas restantes relativas al concepto.
- La condición de parada del test se verifica cuando no existe un aumento significativo en la probabilidad de que el alumno conozca el concepto o bien ya no existen más preguntas en la base de datos.

3.4. Herramientas empleadas

Para la implementación del sistema inteligente se han empleado las siguientes herramientas:

- El Compendium Project para la generación del mapa conceptual (http://www.compendiuminstitute.org). La base de datos MySQL del Compendium se traduce a la base de datos del proyecto. Los nodos de tipo lista o mapa se sustituyen por nodos internos que no aparecen en la base de datos final.
- Hibernate framework para el acceso a la base datos para la capa de abstracción relacionada con el tipo de dicha base de datos (http://www.hibernate.org/80.html)
- Spring framework que implementa el patrón de DI (dependency injection) para la capa de abstracción mediante la definición del acceso a la base de datos, el algoritmo empleado por el test adaptivo, la forma de rellenar la tabla que contiene el mapa conceptual (actualmente implementada en Compendium) y la acción a tomar si el usuario no supera los conceptos previos del mapa en el test.
- Struts 2 un web framework implementado en Java que emplea la arquitectura Modelo-Vista-Controlador (MVC) (http://struts.apache.org/2.x/index.html).
- ANTLR (http://www.antlr.org) Another Tool for Language Recognition, una herramienta que proporciona soporte para construir reconocedores, compiladores y traductores de descripciones gramaticales. Se emplea para la implementación del algoritmo del test, con lo que se pueden poner las expresiones en un fichero externo leído por el parser.
- Google Web Toolkit, AJAX para la interfaz gráfica.

4. Conclusión

Hemos presentado en esta ponencia el proceso de diseño de un sistema inteligente para el apoyo de la docencia siguiendo un aprendizaje significativo. Como dominio de la aplicación hemos escogido la enseñanza universitaria de la jerarquía de memoria. El desarrollo comenzó con un proceso de investigación-acción para la mejora de la docencia. Esta investigación-acción conduce al desarrollo de un sistema inteligente, independiente incluso del dominio de la aplicación. Para este diseño, el profesor debe expresar este dominio de la aplicación en una nuestra propuesta de variación de mapas conceptuales a la que hemos denominado Grafo Instruccional Conceptual Pedagógico (Pedagogical Concept Instructional Graph, PCIG). Este PCIG no ordena los conceptos en relaciones arbitrarias, como es el caso de los mapas conceptuales, sino que son colocados según el orden lógico en que deben ser presentados al alumno.

En el diseño de este sistema podemos distinguir dos roles: el de administrador y el del estudiante. El sistema experto hace uso de un test adaptivo que presenta varias preguntas al alumno entre un conjunto elaborado por el profesor. De las respuestas de cada estudiante, se puede obtener un PCIG personalizado que describe lo que el alumno conoce a priori de la materia a estudiar.

Los siguientes puntos reflejan las líneas abiertas y futuras extensiones del sistema inteligente.

- Validación del sistema con usuarios reales. En la actualidad, el sistema (denominado *Student COncept MAp Explorer*, SCOMAX) está siendo empleado con éxito en la enseñanza del concepto de número en colaboración con una universidad brasileña y en el master DURIA, un título propio de la Universidad de La Laguna. Algunas pantallas de la herramienta implementada se pueden ver en la Figura 3.
- En la descripción del sistema, los autores proponemos emplear potencialmente el paradigma de las ontologías para la expresión del conocimiento, a partir del modelo del mapa conceptual generado por el profesor.
- A partir de los modelos original y personalizado, el sistema experto determinará una estrategia de aprendizaje basado en casos,

agrupando a los alumnos de similares características. Esta estrategia se compondrá de una serie de recursos: bibliografía, enlaces web, juegos o simuladores.

- Aplicación a otras áreas de conocimiento, en particular la enseñanza de Matemáticas en la educación secundaria, lo que permite una validación de la herramienta tanto en entornos universitarios como no universitarios.
- Facilitar que cada usuario pueda comprobar su estado de conocimiento en el contexto de un mapa conceptual (esto se puede conseguir empleando la posibilidad que ofrece la herramienta Compendium de exportar el mapa conceptual a HTML).
- Eliminar la dependencia del uso de la herramienta Compendium,
- Integración del sistema de estándares como SCORM.

Referencias

[1] Argyris, C.; Putnam, R.; Smith, D.McL. Action science: concepts, methods and skills for research and intervention. San Francisco, Ca.: Jossey-Bass, 1985

[2] Bareholz, H.; Tamir, P. A comprehensive use of concept mapping in design instruction and assessment. Research in Science and Technology Education. 1992, 10(1), 37-52.

[3] Bernard, R. M.; Naidu, S. Post-questioning, concept mapping and feedback: a distance education field experiment. British Journal of Educational Technology. 1990, 23(1), 48-60.

[4] Chickering, A. W.; Gamson, Z. F., "Applying the Seven Principles for Good Practice in Undergraduate Education" New Directions for Teaching and Learning, vol. 47, San Francisco: Jossey-Bass Inc., Publishers, 1991.

[5] Dabbagh N. Concept mapping as a mindtool for critical thinking. J Computing Teacher Educ. 2001, 17, 16–24.

[6] Heller, F.A. Group feedback analysis as a method of action research. In AW Clark, Experimenting with organisational life. New York: Plenum, 1976

[7] Hopkins, D. A teacher's guide to classroom research. Philadelphia: Open University Press, 1985

[8] Jonassen, D.H.; Peck, K.L.;Wilson, B.G. Learning with technology. Upper Saddle River, NJ: Merrill Publishing, 1999.

[9] Michael J. In pursuit of meaningful learning. Adv Physiol Educ. 2001, 25: 145–158.

[10] MNEME se puede descargar gratuitamente de la dirección http://www.isaatc.ull.es/portal/proyectos/mneme

[11] Novak, J. D.; Gowin, D.B. Aprendiendo a aprender. Ed. Martínez Rocca, España, 1988.

[12] Novak, J.D. Clarify with concept maps: A tool for students and teachers alike. The Science Teacher. 1991, 58(7), 45-49.

[13] SIJEM se puede descargar gratuitamente de la dirección http://www.isaatc.ull.es/portal/proyectos/sijem

Figura 3. Captura de pantallas de la herramienta SCOMAX

La revisión entre iguales como herramienta de aprendizaje y evaluación en la asignatura de sistemas operativos

Miguel Riesco Albizu, Marián Díaz Fondón
Dpto. de Informática
Universidad de Oviedo
c/ Calvo Sotelo, s/n 33007 Oviedo
{albizu,fondon}@uniovi.es

Resumen

La revisión entre iguales se lleva usando muchos años para medir la calidad de los trabajos científicos, siendo un método muy utilizado para considerar la aceptación de trabajos en publicaciones y congresos.

La misma idea se ha ido aplicando más recientemente como método de docencia en distintos entornos educativos. En este caso, en lugar de ser el profesor el responsable de la explicación de los contenidos, son los propios alumnos los que se explican la materia de la asignatura unos a otros. De la misma manera, también pueden ser los propios alumnos los que evalúen el trabajo de sus compañeros, pudiendo utilizarse así el trabajo entre iguales tanto para el aprendizaje como para la evaluación de los alumnos.

En la asignatura de Sistemas Operativos de segundo curso de la Escuela Universitaria de Ingeniería Técnica en Informática de Oviedo llevamos cuatro años utilizando esta técnica en la docencia y evaluación de algunas de las prácticas. En este artículo se presenta la manera en que lo llevamos a cabo, así como distintos resultados obtenidos respecto al aprendizaje y al trabajo que supone a los alumnos y al profesor.

1. El trabajo entre iguales

La revisión entre iguales o pares (*pair review*) se ha establecido de una manera sólida como la forma más utilizada para la evaluación de trabajos científicos [6], normalmente con el objetivo de ser publicados en una revista, aceptados para un congreso o para ser merecedores de una beca o premio.

Dentro del ámbito científico se ha venido utilizando de forma sistemática desde mediados del siglo XX y, si bien últimamente han surgido algunas críticas [5], en general se considera un método fiable y justo.

Su filosofía es simple: someter el trabajo de un autor al escrutinio de varios expertos de la misma área de la que versa el trabajo. Así, un autor evalúa el trabajo de sus colegas, normalmente de manera anónima.

La misma idea del *trabajo entre iguales* ha sido adoptada por distintas corrientes dentro del estudio de la pedagogía, y se ha intentado aplicar a distintos niveles:

- Aprendizaje entre iguales.
- Revisión entre iguales.
- Evaluación entre iguales.

1.1. El aprendizaje entre iguales

En el aprendizaje entre iguales (*peer instruction)* los actores del proceso educativo son fundamentalmente los alumnos. Así, el conocimiento no se transmite del profesor al alumno, sino que son los propios alumnos los que colaboran para aprender.

Este aprendizaje, según las teorías constructivistas del aprendizaje, puede darse a varios niveles [2]:

- Tutoría entre iguales. A pesar de ser sólo alumnos los implicados, unos desempeñan un papel de tutor y otros de alumnos. El papel no es fijo para cada alumno, sino que podrán actuar como tutores en unos temas y como alumnos en otros.
- Cooperación. Los alumnos tienen grados de habilidad similares, pero éstas son distintas y complementarias.
- Colaboración. Los alumnos tienen habilidades similares, desempeñando roles simétricos.

Independientemente del nivel en el que se aplique, lo fundamental en este caso es que el profesor pierde protagonismo en el proceso de aprendizaje, pasando casi la totalidad de éste hacia el alumno.

El profesor pasa a asumir un papel de guía en el proceso de aprendizaje.

1.2. La revisión entre iguales

Aplicado a entornos de aprendizaje, la revisión entre iguales funciona de manera similar a como lo hace en entornos científicos: el trabajo de un alumno es revisado por uno o varios compañeros, quienes ofrecerán su opinión sobre el mismo, indicando los problemas que presenta, sugerencias de mejora, etc.

1.3. Evaluación entre iguales

Además de utilizarse como instrumento de aprendizaje, el trabajo entre iguales también puede usarse para la evaluación. En la evaluación entre iguales (*peer assessment*) son los propios alumnos los que evalúan a sus compañeros, criticando su trabajo y, a diferencia del caso anterior, otorgándole una calificación.

Algunas de las ventajas que de este tipo de evaluación se han dado [1][4] son las siguientes:

- La evaluación pasa a ser parte del proceso de aprendizaje, pasando los errores a ser fuente de aprendizaje.
- Aumenta la motivación, al verse implicado en el proceso de evaluación.
- Se trabaja con competencias necesarias para el "aprendizaje a lo largo de la vida", como son las relativas a la evaluación, a la autonomía y responsabilidad, así como a fomentar el espíritu crítico.
- Favorece un aprendizaje más profundo de la materia, dado que para evaluar no basta con conocimientos superficiales.
- Disminuye la carga de trabajo del profesor, al verse liberado de la tarea de corrección.

Sin embargo, algunos autores [4] se muestran reticentes a utilizar este tipo de evaluación por distintos motivos:

- No se considera a los alumnos lo suficientemente expertos como para evaluar a sus compañeros.
- Se "sospecha" que los alumnos pueden ser excesivamente generosos en calificar a sus compañeros, no siendo esta calificación, por tanto, muy fiable.
- Algunos alumnos pueden rechazar este tipo de evaluación, al no confiar en sus compañeros para llevarla a cabo.

2. Entorno de aplicación de la experiencia

Dentro de la asignatura de Sistemas Operativos, de segundo curso de la Escuela Universitaria de Ingeniería Técnica en Informática de Oviedo, llevamos varios años intentando transformar la manera de impartir docencia según las directrices del Espacio Europeo de Educación Superior [3].

Dentro de este contexto, hace cuatro años comenzamos a aplicar la revisión y evaluación entre iguales en algunas de las prácticas de la asignatura.

Tradicionalmente las prácticas se evaluaban por parte del profesor, bien mediante la entrega de un trabajo práctico o bien mediante un examen (escrito o práctico). Esto daba como resultado una gran carga de trabajo para el profesor, dado que tenía que evaluar cinco prácticas de cerca de trescientos alumnos a lo largo del curso.

Además, los profesores tenían la impresión que los alumnos no estaban especialmente motivados con la realización de la práctica, la hacían con la intención de quitársela de delante cuanto antes y en cuanto la entregaban se olvidaban ya de ella.

Ante esta situación los profesores trataron de cambiar la dinámica de las prácticas, introduciendo el trabajo entre iguales en alguna de ellas. Básicamente lo que se cambió fue la evaluación de las prácticas, pasando a ser una evaluación entre iguales.

3. Objetivos buscados

Los objetivos que se buscaban al cambiar la forma de evaluar las prácticas fueron varios y de distintos tipos.

En primer lugar se pretendía mejorar la formación de los alumnos sobre la materia de estudio (*objetivos específicos*), fundamentalmente a través de un estudio más profundo y crítico. Para ello, se buscaba:

- Aumentar el periodo de trabajo con la materia, haciendo de la evaluación algo provechoso para el aprendizaje del alumno.
- Fomentar el estudio y comprensión en profundidad de la materia, en lugar de limitarse a un estudio superficial, como ocurría en muchas ocasiones.
- Ampliar el "campo de visión" del alumno sobre los problemas presentados, estudiando

planteamientos o soluciones alternativas al dado por él, diferentes maneras de abordarlo, etc.
- Proporcionar de manera rápida y eficiente realimentación sobre su trabajo.

Además de los anteriores, y de acuerdo con los planteamientos del EEES, se buscaba fomentar competencias transversales como las siguientes:
- Tomar un papel más activo en el aprendizaje.
- Capacidad crítica del alumno.
- Capacidad dialéctica, siendo capaz de argumentar y defender sus decisiones.
- Capacidad para asumir errores propios.
- Competitividad. No es lo mismo hacer un examen mal y que lo vea el profesor a que lo vea un colega.

Finalmente, también se buscaba disminuir la carga docente del profesorado asociado a la corrección de prácticas.

En nuestro caso se han llevado a cabo varias experiencias distintas relacionadas con el trabajo entre iguales. La que más provechosa nos ha parecido es la corrección de exámenes de respuesta corta, si bien también lo hemos aplicado a corrección y revisión de exámenes de tipo test y a revisión y evaluación de trabajos prácticos.

A continuación pasamos a describir por separado cada una de ellas, dado que, a pesar de estar basadas en la misma idea, su implantación y los resultados obtenidos difieren radicalmente de unas a otras.

4. Corrección de exámenes

Esta experiencia la aplicamos en la primera práctica (*Introducción al S.O. Unix*) que se evalúa mediante un examen práctico realizado frente al ordenador. El estudiante debe contestar una serie de preguntas respecto al funcionamiento de algunas órdenes del sistema, pudiendo utilizar el ordenador para probarlas o buscar ayuda. Hay que considerar que, dado que hay muchos grupos de prácticas y los ordenadores están muy cercanos unos de otros, cada alumno debe contestar un conjunto de preguntas distintas (no hay dos exámenes iguales).

Históricamente éste era un examen que entregaban los alumnos en papel y que corregía el profesor. De esta manera, el trabajo de los alumnos concluía al realizar el examen, y la corrección del mismo no resultaba provechosa para su formación, dado que sólo en la revisión de exámenes (los que acudían a ella) podían repasar los errores que habían cometido.

Para solventar en lo posible los problemas anteriormente expuestos se puso en marcha, en el curso 2003-04, un sistema de corrección de prácticas entre iguales.

4.1. Mecánica de funcionamiento

En este sistema la evaluación de la práctica se lleva a cabo en cinco fases:
1. El alumno (*examinando)* entrega su examen en algún formato electrónico. En nuestro caso a través de un formulario web.
2. Se distribuyen los exámenes entre todos los alumnos, de manera aleatoria y anónima, de forma que cada alumno (*corrector)* debe corregir el examen de un compañero.
3. Una vez corregido, se hace llegar la corrección al examinando. Podrá comprobar la calificación que su corrector le ha dado, los errores que ha cometido, la justificación de la corrección de cada pregunta, la respuesta correcta en cada caso, etc. Si no está de acuerdo con algún punto, podrá presentar una alegación a esa corrección, para que se le aclare mejor o para que se le modifique la nota.
4. Las alegaciones presentadas por el examinando se hacen llegar al corrector, que podrá a su vez aceptarlas o rebatirlas.

Los puntos 3 y 4 pueden repetirse el número de veces que se desee, llevando a cabo así un intercambio de opiniones generalmente fructífero entre corrector y corregido. En nuestro caso nos hemos limitado a realizarlo una sola vez.

La última fase del proceso (fase 5) será la revisión por parte del profesor de todo el trabajo de los alumnos.

La calificación final del alumno vendrá dada en función tanto por su labor como examinando (lo contestado en su examen) como por su trabajo como corrector (la corrección realizada al examen del compañero). Actualmente la nota final está constituida en un 70% por la nota obtenida como examinando y en un 30% por la nota obtenida como corrector. El cálculo de cada una de ellas se expone a continuación.

4.2. Calificación como examinando

La calificación obtenida por el alumno en su papel de examinando vendrá dada no sólo por lo que

contestó en el examen, sino que además se considerará su firmeza al defender sus respuestas correctas, así como su capacidad de crítica para asumir los errores cometidos.

La Tabla 1 muestra cómo se calcula la nota de cada pregunta para todos los posibles casos (la columna de Corrección "Bien" significa que está bien corregida y "Mal" que está mal corregida; el apartado de Alegación hace referencia a si se presentan alegaciones bien razonadas).

Respuesta	Corrección	Alegación	Valor
Bien	Bien	No	100%
Bien	Bien	Sí	50%
Bien	Mal	No	50%
Bien	Mal	Sí	100%
Mal	Bien	-	0
Mal	Mal	No	0
Mal	Mal	Sí	50%

Tabla 1. Cálculo de la nota del examinando

De entre todos los casos posibles es interesante reparar en los siguientes:

- Si la pregunta está bien contestada, bien corregida, pero se presentan alegaciones a la corrección: 50 % del valor de la pregunta. Se supone que el examinando no está seguro de la respuesta (a pesar de que estaba bien), y por eso alega. Por ese motivo se le penaliza.
- Si la pregunta está bien contestada, mal corregida (el corrector dice que está mal) y no se presentan alegaciones razonadas a la corrección: 50% del valor la pregunta. Al igual que en el caso anterior, el examinando no está seguro de la respuesta y no alega a pesar de haber contestado correctamente.
- Si la pregunta está mal contestada, mal corregida (se le ha dado como bien contestada) pero se presentan alegaciones razonadas a la corrección: 50% del valor la pregunta. A pesar de no haber contestado bien, al haberse equivocado también el corrector dándola como buena, si el examinando se ha preocupado de conocer la respuesta correcta y explicarla en la alegación, se le otorgará la mitad de los puntos de la pregunta.

Este esquema de puntuación a través de "premios y castigos" tiene por objeto que el alumno se preocupe de afianzar los conocimientos necesarios para responder cada pregunta, dado que puede ocurrir que una pregunta bien contestada en el examen le sea puntuada con la mitad de su valor por no saber defenderla adecuadamente.

De la misma forma, una pregunta mal contestada en el examen puede reportarle la mitad de su valor, en lugar de un 0, si se preocupa por conocer la respuesta posteriormente y defenderla adecuadamente.

4.3. Calificación como corrector

Además de la nota obtenida por el examinando en su examen, habrá que calcular también la nota en su papel de corrector.

La Tabla 2 resume las distintas posibilidades y el valor asignado al corrector para esa pregunta.

Corrección	Alegación	Se Acepta	Valor
Bien	Sí	No	100%
Bien	Sí	Sí	50%
Bien	No	-	100%
Mal	Sí	No	0
Mal	Sí	Sí	50%
Mal	No	-	0

Tabla 2. Cálculo de la nota del corrector

De manera similar al caso de la nota del examen, hay unas cuantas posibilidades merecedoras de ser destacadas:

- Si la pregunta está bien corregida, se presenta alegación y se acepta: 50%. A pesar de haber corregido bien al principio, no está muy seguro puesto que cambia su criterio ante una alegación.
- Si la pregunta está mal corregida, pero tras una alegación se corrige bien: 50% del valor del apartado. El corrector se había equivocado al principio, pero tras la alegación del examinando reconoce su error.
- Si la pregunta está mal corregida, se presenta alegación y no se acepta: 0. El corrector corrige mal y tras la alegación del examinando persiste en su error.

El corrector debe preocuparse de afianzar sus conocimientos para poder aceptar o rebatir las alegaciones que sobre su corrección haga el alumno examinando. De ello va a depender que su corrección, independientemente de que en principio estuviera bien o mal, le aporte puntos a la nota como corrector.

4.4. Puesta en marcha

Para poder llevar a cabo esta experiencia es necesario contar con el apoyo tecnológico adecuado, dado que las distintas fases de realización y corrección de los exámenes deben completarse e intercambiarse en formato electrónico. El intentar llevarlo a cabo con exámenes escritos en papel no sería operativo, a no ser que se contara con un número muy pequeño de alumnos, lo que daría lugar a que probablemente se perdiera el anonimato deseable en estas experiencias.

En nuestro caso llevamos realizando la experiencia desde el curso 2003/2004. Ese curso y el siguiente la realización de las distintas fases del examen se hacía editando un simple fichero de texto, llevándose a cabo las entregas e intercambios de exámenes mediante la ejecución de un sencillo programa en el servidor de la Escuela.

Este primer prototipo nos sirvió para darnos cuenta de los inconvenientes que presentaba, sobre todo para los profesores a la hora de revisar y calcular las notas de los alumnos.

A partir del curso 2005/2006 se utiliza un entorno web, implementado al igual que en el caso anterior por los profesores de la asignatura, donde las distintas fases (incluidas la corrección del profesor) se llevan a cabo a través de formularios web. Este entorno, mucho más amigable, ha permitido facilitar y acelerar el proceso de corrección del profesor de los exámenes de manera muy notable, como más adelante se muestra.

4.5. Beneficios e inconvenientes

Además de los beneficios e inconvenientes generales apuntados en el apartado 1.3, y de los objetivos expuestos en el apartado 3, tras llevar aplicando la experiencia expuesta cuatro cursos podemos comentar una serie de beneficios y de problemas que hemos encontrado en su aplicación.

Los beneficios que se obtienen con esta técnica son fundamentalmente para el alumno:

- Se logra que la fase de corrección sea productiva desde el punto de vista del aprendizaje. Se consigue un proceso de realimentación a través de los errores cometidos.
- El alumno tiene la oportunidad, aunque le haya salido mal el examen, de mejorar su nota haciendo una buena corrección del que se le asigne.
- El alumno desarrolla un aprendizaje más activo, lo que se traduce en un mayor dominio de la materia.

Entre los problemas que hemos detectado al aplicar esta técnica podemos citar los siguientes:

- Desde el punto de vista del profesor supone un incremento de trabajo, dado que tiene que, además de corregir la práctica, corregir la corrección. Sin embargo, como mostraremos posteriormente, este incremento es aparente, no real.
- Es fundamental tener un conjunto de herramientas informáticas adecuado para poder llevar a cabo el proceso de manera cómoda y segura. En nuestro caso, estas herramientas han tenido que ser desarrolladas por los profesores de la asignatura.

4.6. Impacto en el trabajo del profesor y del alumno

La realización de este tipo de exámenes va a suponer un aumento en el trabajo tanto del profesor como del alumno. En este apartado trataremos de cuantificar este aumento.

Respecto al alumno su trabajo se incrementa respecto al que realizaba en la misma práctica evaluada mediante un examen tradicional, al tener que realizar las fases de corrección, alegación y resolución de alegaciones.

En cualquier caso, el empleo de este tiempo podemos considerarlo positivo, dado que no es excesivo (por término medio se emplean unas dos horas en las tres fases) y es un tiempo productivo desde el punto de vista del aprendizaje del alumno. En términos relativos su carga de trabajo aumentaría un 20%, dado que la práctica se estima que dura 10 horas. Este porcentaje puede parecer alto pero viene dado por el hecho de que la práctica es relativamente corta.

En relación al trabajo del profesor podría parecer, dado que tiene que corregir además del examen el resto de las fases (corrección, alegación y resolución) que su trabajo iba a aumentar espectacularmente.

En los cuatro años que se lleva realizando la experiencia se ha computado el tiempo que se ha empleado en realizar las correcciones. Los resultados se muestran en la Tabla 3, ("Nº Al" es el número de exámenes, "Pr./Ex." indica el número de preguntas por examen, "T. C." es el tiempo

total de corrección del profesor y "T. C./Preg" es el tiempo de corrección por pregunta).

Curso	Nº Al.	Pr./Ex.	T. C.	T. C./P.
2003/2004	115	18	23 h.	40,0 sg
2004/2005	197	18	38 h.	38,6 sg
2005/2006	167	10	7,5 h.	16,1 sg
2006/2007	128	10	4,5 h.	12,7 sg

Tabla 3. Tiempos de corrección de los exámenes

Como se puede observar, el tiempo de corrección en los primeros años de la experiencia no es desmesurado. Los exámenes eran de preguntas cortas, y el tiempo que se muestra podría ser equiparable al de corrección de un examen típico realizado en papel. En este caso, dado que la entrega era electrónica se realizaron una serie de programas que facilitaban la visualización de las preguntas y sus respectivas correcciones, y permitía calificarlos mediante teclas previamente programadas, con lo que se aceleraba el proceso con respecto a un examen escrito en papel.

El gran salto cuantitativo ocurre en el curso 2005/2006, donde, por la experiencia tenida de los dos años anteriores, se mejoraron mucho los programas de apoyo a la corrección. Además de mejorar el interfaz de visualización de las distintas fases y el de introducción de notas, que aceleraban algo el proceso, la principal novedad radicó en que el sistema seleccionaba para la corrección todas las preguntas con el mismo enunciado (recuérdese que todos los exámenes eran distintos) de entre todas las preguntas de todos los exámenes, pudiendo así el profesor leer sólo el enunciado de la primera pregunta a corregir, pensar cuál es la respuesta correcta para ese enunciado y centrarse en corregir la respuesta que para esa pregunta han dado los alumnos a los que le fue asignada.

Como se ve en la tabla, esta sencilla idea prácticamente dividió a algo menos de la mitad el tiempo requerido de corrección. La experiencia acumulada y una mejora adicional en el interfaz de corrección para minimizar los desplazamientos del ratón han permitido disminuir un poco más los tiempos este año.

4.7. Resultados obtenidos

Un primer resultado cuantitativo que podemos mostrar es ver cómo ha variado la nota de los alumnos en el examen realizado en comparación con la nota que hubieran tenido si la corrección la hubiera realizado únicamente el profesor. La Tabla 4 muestra, el número y porcentaje de alumnos que han mejorado su nota, los que la han bajado y la media, el máximo y el mínimo en cada caso de los puntos que han subido o bajado.

	Nº	%	Media	Max.	Min.
Suben	80	62,50%	0,49	2,35	0,15
Bajan	37	28,91%	-0,19	-3,32	-0,07
Igual	11	8,59%			
Total	128		0,29	2,35	-3,32

Tabla 4. Comparación de las notas obtenidas

Se puede ver en la tabla anterior cómo en la mayoría de los casos han subido sus notas, lo que sin duda ayuda en la motivación del alumnado a realizar este tipo de experiencias. Se puede ver también cómo la mejora puede llegar a ser considerable, mientras que en el caso medio sube algo la nota pero no demasiado.

También hay alumnos cuya nota es inferior a la que hubieran obtenido con una corrección tradicional, al no haber llevado a cabo las distintas fases adecuadamente. Esto también puede servir para darse cuenta que hay que tomarse la corrección y el resto de las fases en serio.

Otro indicador de la implicación del alumnado en la experiencia ha sido que, de las 1.280 preguntas realizadas, 170 (un 13,3%) han sido alegadas.

Además de estos resultados se ha sondeado, mediante una encuesta, la opinión de los alumnos con relación a la experiencia y su influencia en el aprendizaje de la materia. Al 71% de los alumnos les pareció interesante o muy interesante la experiencia, mientras que el 84% consideraron que la experiencia era positiva para su formación. Conversaciones informales mantenidas con un gran número de alumnos han confirmado esta opinión.

Finalmente, los profesores de la asignatura también estamos muy satisfechos de los resultados de la experiencia, lo que nos ha llevado a incorporar de manera definitiva este tipo de exámenes a la dinámica normal de docencia de la asignatura.

5. Otras experiencias realizadas

Ante el éxito que tuvo la experiencia descrita en el punto anterior, se intentó aplicar el trabajo entre iguales en otras partes de la asignatura, si bien el resultado no fue tan bueno.

5.1. Corrección de exámenes de test

En el curso 2003/2004 se llevó a cabo una experiencia similar a la expuesta en el punto 3, pero aplicado a un parcial de teoría, que se evaluó mediante un examen de tipo test.

La idea era que los alumnos debatieran cada respuesta, al igual que en el examen práctico, hasta llegar a una solución de consenso entre ellos.

La mecánica también era similar: tras la realización del examen de tipo test (escrito esta vez) y una vez leídos por la *lectora óptica de marcas* se distribuían los exámenes entre los correctores, ya en formato electrónico.

El resultado no fue igual de bueno que en la experiencia anterior. Al ser un examen de tipo test las respuestas estaban bien o mal, no había apenas margen para la discusión. Por lo tanto, los correctores se limitaban a corregir el examen asignado según una plantilla que pronto "apareció" entre los alumnos.

Fue la primera y única vez que intentamos aplicar esta técnica a exámenes de tipo test, puesto que el resultado no fue en absoluto comparable al obtenido con exámenes de respuesta corta.

5.2. Corrección de trabajos prácticos

La misma filosofía de trabajo entre iguales, el curso 2005/2006 se aplicó a la revisión y corrección de trabajos prácticos. En concreto se aplicó a la corrección de la práctica tercera de Sistemas Operativos, consistente en la realización de un *shell* de Unix y un conjunto de programas de utilidad.

Tradicionalmente los alumnos entregaban esos programas al profesor, que los corregía en función de unos criterios conocidos de antemano, y publicaba las notas resultantes de la corrección. Ese año fueron los propios alumnos los que tuvieron que corregir las prácticas de sus compañeros.

La forma concreta de cómo se llevó a cabo fue la siguiente:

- La práctica se realizó por parejas. En total se entregaron 70 prácticas, con lo que el número de alumnos presentados fue de 140.
- Cada alumno, de manera individual, tuvo que evaluar dos prácticas, utilizando un baremo muy estricto de los puntos a considerar.
- Así, para cada práctica se obtuvieron 4 correcciones distintas. No se implantaron las fases de alegaciones y resolución de las mismas, al tener 4 correcciones independientes.
- Finalmente el profesor repasó todas las prácticas presentadas, generando sus propias calificaciones. La nota como correctores en este caso se han calculado comparándolas con las del profesor.

La experiencia demostró que esta última fase prácticamente se podría haber eliminado, dado que, al estar el baremo muy detallado, quedaba poco margen a la subjetividad, y, de hecho, la mayor parte de las correcciones diferían muy poco con la realizada por el profesor. Para experiencias posteriores es probable que baste con revisar sólo aquellas prácticas donde las notas de los distintos correctores varíen significativamente y unas cuantas prácticas al azar.

Desde el punto de vista del profesor esta experiencia tuvo un éxito moderado:

- Por una parte, se demostró que los alumnos son correctores estrictos, con lo que el trabajo de corrección de prácticas por parte del profesor puede disminuir.
- Al examinar el trabajo de otros alumnos, el corrector se da cuenta de lo problemático que resulta revisar código ajeno, concienciándose de la necesidad de llevar a cabo buenas prácticas de programación.
- La ausencia de las fases de alegación y resolución hace que la corrección sea menos formativa que si las hubiera, dado que el alumno debería poder modificar su trabajo en función de los comentarios de los correctores, corrigiendo errores, modificando su estilo de programación, etc.

La buena impresión que el profesorado obtuvo con la experiencia no fue refrendada por el alumnado. El aumento de trabajo que supuso, lo estricto del baremo de calificación, que convertía la corrección en algo mecánico, y el que los alumnos no apreciaran claramente que fuera bueno para su formación (muchos llegaron a decir que sólo servía para "hacer el trabajo del profesor") hizo que la experiencia fuera fuertemente rechazada por parte de los alumnos.

Esta contestación por parte del alumnado ha hecho que actualmente nos estemos replanteando el enfoque de esta experiencia. En este curso propondremos a cada grupo autoevaluar su práctica a partir de los criterios de evaluación, que serán explícitamente indicados. El objetivo es hacer que

el alumno encuentre sus errores y los corrija, asegurando así que han modificado su aprendizaje. Posteriormente haremos una revisión por pares (sin evaluación) para intentar lograr los objetivos antes señalados, pero permitiendo a los autores modificar su práctica antes de entregarla en función de las opiniones de sus revisores. La evaluación final la llevará a cabo el profesor.

6. Conclusiones

La evaluación entre iguales introduce en el proceso de aprendizaje una fase de retroalimentación consistente en identificar errores cometidos y, por tanto, corregir y mejorar el aprendizaje. Además, éste es un aprendizaje activo y, por tanto, de gran calidad.

El hecho de delegar la evaluación en otros implica que el profesorado deba definir con gran precisión y claridad los criterios de evaluación del trabajo del alumno.

Hemos podido constatar que el alumno mejora notablemente su rendimiento cuando conoce estos criterios, puesto que puede utilizarlos para su propia autoevaluación.

En este artículo se ha presentado la experiencia de uso de corrección entre iguales en la asignatura de Sistemas Operativos a distintos niveles. Se ha aplicado a la corrección de un examen de respuesta corta, de preguntas de test y de un trabajo práctico, siendo la más provechosa la primera de ellas.

Centrándonos por ello en la corrección de exámenes de pregunta corta, podemos concluir que el grado motivación de los alumnos en el proceso ha sido gratamente sorprendente. La implicación de los alumnos en el proceso se ha plasmado incluso en largas e interesantes discusiones técnicas en foros de debate, lo que demuestra su interés en la experiencia. Además, sondeada su opinión a través de encuestas, hemos podido constatar que:

- La gran mayoría de los alumnos (el 84 %) consideran la experiencia muy positiva para su aprendizaje.
- El aprendizaje de la materia en cuestión se concentra especialmente en las fases de corrección y posteriores del proceso, por encima, incluso, de la explicación y el estudio previo al examen.

Adicionalmente, esta experiencia contribuye al desarrollo de competencias transversales como espíritu crítico, capacidad de asumir errores, capacidad dialéctica y competitividad.

Por otro lado, se ha comprobado que, una vez asimilada la mecánica de funcionamiento, el trabajo del profesor no tiene por qué aumentar en relación con una evaluación tradicional.

Si bien la realización de este tipo de experiencias requiere de un apoyo tecnológico elevado (es necesario que todos los alumnos tengan a su disposición ordenadores para poder realizar todo el proceso, además de contar con las herramientas informáticas adecuadas para hacerlo) consideramos que, con la planificación adecuada, puede llevarse a cabo en cualquier centro universitario. En nuestro caso se ha podido realizar con sólo 26 ordenadores para un total de cerca de 200 alumnos.

Referencias

[1] Brown, S., Rust, C. and Gibbs, G. *Involving students in the assessment process, in Strategies for Diversifying Assessments in Higher Education*. Oxford Centre for Staff Development, and at DeLiberations. 1994

[2] Damon, W, Phelps, E. *Critical distinctions among three approaches to peer education*. International Journal of Educational Research, nº58, vol. 2,pp. 9-19. 1989.

[3] Díaz Fondón, M.; Riesco Albizu, M.; Martínez Prieto A. B. *Hacia el aprendizaje activo: una caso práctico en la docencia de Sistemas Operativos*. Revista Novática, pp. 54-58. Marzo-Abril 2005

[4] Falchikov, N. *Product comparisons and process benefits of peer group and self assessments*. Assessment and Evaluation in Higher Education, vol 11, pp. 146-166. 1986.

[5] Gérvas J, Pérez Fernández M. *La revisión por pares en las revistas científicas*. Atención Primaria 2001.

[6] Tracey Brown (Editor). *Peer Review And The Acceptance Of New Scientific Ideas*. Sense About Science, Octubre 2004.

El Aprendizaje Basado en Proyectos como modelo docente. Experiencia interdisciplinar y herramientas Groupware.

Juan R Reverte Bernabeu, Antonio Javier Gallego Sánchez,
Rafael Molina Carmona, Rosana Satorre Cuerda.
Dpto. de Ciencia de la Computación e Inteligencia Artificial
Universidad de Alicante
Ctra. San Vicente del Raspeig s/n, Alicante
{jreverte, ajgallego, rmolina, rosana}@dccia.ua.es

Resumen

En este artículo presentamos una experiencia desarrollada en un programa docente que utiliza el *Aprendizaje Basado en Proyectos* (ABP) para impartir de forma conjunta cuatro asignaturas de la titulación de Ingeniería Informática.

Para ello, en primer lugar se propuso la realización de un videojuego como proyecto conjunto. La elección de esta temática permitía cubrir los objetivos de todas las asignaturas. De esta forma todas aportan algo y reciben aportaciones de las demás, reforzando el carácter interdisciplinar de la experiencia.

Una docencia basada en proyectos que se desarrollan en grupo tiene unos altos requerimientos de comunicación y de intercambio de información. Proponemos el uso de una herramienta de trabajo colaborativo (Groupware) que hemos denominado *ABP-Forja*.

Para evaluar tanto la metodología como las herramientas tecnológicas realizamos un cuestionario cuantitativo y cualitativo que nos permitiera conocer las opiniones de los alumnos. Los alumnos han valorado positivamente la metodología, destacando el trabajo en grupo y la forma de evaluación, aunque consideran que se incrementa el tiempo de dedicación. Por otro lado, consideran que el uso de la *ABP-Forja* es muy apropiado y valoran, especialmente, la gestión de tareas, el control de versiones y el gestor de documentos.

Esta experiencia nace con el propósito de extenderse en el futuro a otras asignaturas. Se pretende, además, mejorar en los cursos sucesivos la herramienta de trabajo en grupo.

1. Introducción

Durante el curso académico 2006/07 un grupo de profesores de Ingeniería Informática de la Universidad de Alicante hemos puesto en marcha un nuevo tipo de metodología docente que hace uso del *Aprendizaje Basado en Proyectos* (ABP) [6] para impartir de forma colaborativa cuatro de las asignaturas optativas de la titulación de Ingeniería Informática.

La idea de modificar el método docente surge al observar el contexto de la sociedad actual. Hoy en día los cambios tecnológicos se suceden cada vez con mayor rapidez. Peter Scholtes [7] señala que en la actualidad un joven de 18 años será testigo a lo largo de su vida de al menos 4 o 5 renovaciones tecnológicas. Esto hace que cada vez tengan menos importancia los contenidos, y sea mayor la necesidad de enseñar a aprender y los métodos utilizados para ello. Se trata de preparar a nuestros jóvenes para el aprendizaje continuo.

La docencia tradicional se basa en cuatro principios fundamentales: la fragmentación del conocimiento en diferentes asignaturas, la clase expositiva como medio para la transmisión del conocimiento, el estudio individual y la evaluación a través de exámenes [4]. Esta metodología tiene la ventaja de permitir que el profesor transmita la información de manera rápida a muchos alumnos a la vez. Sin embargo tiene grandes limitaciones: no favorece la interdisciplinariedad, no promueve el pensamiento creativo, la discusión o la toma de decisiones, no facilita la aplicación del conocimiento a casos reales, etc. En definitiva, es un modo de docencia que tiene sentido en un contexto de masificación, pero no es adecuado para formar profesionales que trabajen en equipo. Concretamente, en las titulaciones de informática, la mayor parte de las veces se forma a los alumnos mediante métodos que sólo les preparan para trabajar de forma individualizada. Pero realmente el tipo de actividad que desempeñarán en el futuro requerirá de la colaboración y organización de grandes equipos.

Unido a todo esto, actualmente la universidad se encuentra inmersa en un proyecto de adaptación al Espacio Europeo de Enseñanza Superior (EEES) [2]. Entre otros cambios, este nuevo contexto va a suponer la implantación del sistema europeo de créditos (ECTS), el cual pasará a medir las horas totales de trabajo del estudiante (no sólo las horas de docencia presencial). Esta medida comporta un nuevo modelo educativo que ha de orientar las programaciones y metodologías docentes centrándolas en el aprendizaje de los estudiantes, y no exclusivamente en las horas lectivas.

Ante esta situación han ido apareciendo en los últimos años algunas nuevas formas de entender el proceso de enseñanza-aprendizaje. Entre estas propuestas destaca el Aprendizaje Basado en Proyectos (ABP), que intenta minimizar las limitaciones de la docencia tradicional, centrar la docencia en el aprendizaje del alumno y a la vez crear una situación mucho más aproximada a la real (la del futuro trabajo del alumno).

El propósito del proyecto que aquí se plantea ha sido modificar el modelo de docencia tradicional en cuatro asignaturas de Ingeniería Informática y adaptarlo al Aprendizaje Basado en Proyectos. Para esto se planteó a los alumnos un proyecto conjunto que, cubriendo los objetivos de todas las asignaturas, les ha permitido iniciarse en el desarrollo de grandes aplicaciones, aprender a trabajar en grupo, conocer herramientas de trabajo colaborativo (muy utilizadas en el sector de la informática), y practicar los contenidos de las asignaturas (conociendo así su parte más útil y aplicable).

En la siguiente sección se explica con más detalle la metodología ABP. El apartado 3 muestra los pasos seguidos para la implantación del modelo: adaptación del método docente y de las distintas asignaturas, e incorporación de nuevas herramientas de trabajo. En la sección 4 se presentan los resultados obtenidos del estudio cuantitativo y cualitativo de la experiencia. Por último, la sección 5 está dedicada a las conclusiones y a las futuras líneas de trabajo.

2. Aprendizaje basado en proyectos

El Aprendizaje Basado en Proyectos es una metodología didáctica en la que los alumnos, organizados por grupos, aprenden a través de su investigación los conceptos de la asignatura. Dichos conocimientos son aplicados a la realización de un proyecto o la resolución de un problema. Este proyecto sólo estará adecuadamente diseñado cuando para concluir de manera exitosa sea necesario adquirir todos los conocimientos que el profesor desee transmitir. Por lo tanto, el papel del profesor se modifica, pasando a ser un mediador o guía. Su labor se centra en encaminar al estudiante para que encuentre la mejor solución al problema.

El ABP ha ido incorporándose a la vida universitaria durante los últimos 50 años, especialmente en el mundo anglosajón. Las primeras experiencias se deben a la McMaster University en Ontario [1], la cual hizo uso (y en la actualidad continua haciéndolo) de este tipo de metodología para la enseñanza de la carrera de medicina. Algunas universidades del centro y el norte de Europa también han aplicado estos modelos en sus estudios: Aalborg o Roskilde en Dinamarca, Maastricht y Twente en Holanda o Linköpings en Suecia. En España también existen experiencias de este tipo, algunas de las cuales usan también herramientas de trabajo colaborativo [5].

Diversos estudios muestran que el ABP fomenta habilidades tan importantes como son: el trabajo en grupo, el aprendizaje autónomo, la capacidad de autoevaluación, la planificación del tiempo, el trabajo por proyectos o la capacidad de expresión oral y escrita. Además mejora la motivación del alumno, lo que se traduce en un mejor rendimiento académico y una mayor persistencia en el estudio. No obstante este modelo de aprendizaje plantea inicialmente dificultades de adaptación tanto para profesores como para alumnos, pues cambia los roles tradicionales.

En el siguiente apartado se muestran los pasos seguidos para la adaptación de las distintas asignaturas que forman el proyecto a la metodología ABP.

3. Nuestra propuesta

La implantación del proyecto se realizó a varios niveles. En primer lugar se modificó la docencia realizada en las asignaturas que comprendían el proyecto, adaptándolas a la nueva metodología docente y al proyecto global. En segundo lugar, se

introdujo una herramienta de trabajo colaborativo para facilitar la comunicación entre los alumnos.

3.1. Contexto de las asignaturas

A continuación se muestran las características de las cuatro asignaturas que han intervenido en el proyecto. Todas son optativas de 6 créditos (3 créditos prácticos y 3 créditos teóricos) de la carrera de Ingeniería Informática.

1. *Modelos de Fabricación Asistidos por Computador (MFAC).* Muestra las aplicaciones de los modelos matemáticos más usuales en investigación operativa. Estos modelos luego son aplicados al análisis y modelado de sistemas, organizaciones y sistemas de producción. La parte más práctica está orientada al estudio y resolución de problemas de naturaleza combinatoria, problemas a nivel estratégico y a nivel táctico.
2. *Razonamiento (RAZ).* Su principal cometido es la construcción de un sistema de inteligencia artificial mediante la aplicación de diferentes tipos de algoritmos, como son: los sistemas multi-agente, las máquinas de estados finitos o la lógica difusa.
3. *Gráficos Avanzados y Animación (GAA).* El objetivo de esta asignatura es dar al alumno una visión global de todo lo relacionado con la generación de imágenes sintéticas. Para esto se tratan temas como: las proyecciones, el tratamiento de modelos tridimensionales, así como diferentes técnicas de visualización en 3D (rendering) y de animación por computador.
4. *Juegos y Realidad Virtual (JRV).* Esta asignatura muestra al alumno las principales técnicas y algoritmos que se utilizan actualmente en la mayoría de los juegos por ordenador del mercado y en las aplicaciones de Realidad Virtual. Posteriormente todos estos conceptos se deberán aplicar al desarrollo de un proyecto que tiene como finalidad la creación de un juego.

Las dos primeras asignaturas se estudian en el primer cuatrimestre, y las dos siguientes en el segundo. Este factor ayuda a que el alumno no sienta la presión de tener que cursarlas todas en el mismo cuatrimestre.

Para poder participar de la experiencia el alumno debería matricularse de las 4 asignaturas. Esto en algunos casos puede ser complicado, ya que es posible que hayan cursado una o varias de las asignaturas durante otros años, o simplemente que no puedan cogerlas por motivos de horario o incompatibilidades. Por ello, para poder llegar a la mayor cantidad de alumnos y a la vez adaptarnos a la situación administrativa actual, se optó por dividir el proyecto en módulos. Es decir, el proyecto conjunto de las asignaturas está compuesto por 4 proyectos más pequeños, de forma que cada asignatura se encarga de un módulo concreto y delimitado del proyecto global. Esta solución tiene además dos ventajas:

- Los alumnos podrán cursar asignaturas por separado y participar de la experiencia, aun en el caso de que no hagan el proyecto completo. La única diferencia que tendrán respecto a los que sí lo hagan será que los módulos que implementen no formarán parte de un proyecto global. Su módulo será evaluado y será su nota de la asignatura.
- Los alumnos que realicen todo el proyecto aprovecharán la interdisciplinariedad de éste para desarrollar módulos de forma separada, pero que al final compondrán las partes de una aplicación mayor, que será el proyecto final. La calificación de cada asignatura se obtendrá de la evaluación del proyecto correspondiente.

Una vez entendido el contexto en el que se enmarcan las asignaturas implicadas, se pasa a explicar los cambios acometidos en la metodología docente de cada una de ellas.

3.2. Objetivos del proyecto común.

El objetivo del proyecto global es que los alumnos diseñen y desarrollen un videojuego. Este videojuego incorporará los diferentes módulos desarrollados en cada una de las asignaturas. Con esta decisión se persiguen varios propósitos:

- En primer lugar, conseguir una simbiosis entre las distintas asignaturas, es decir, enmarcarlas en un contexto donde todas aportan algo y reciben aportaciones de las demás. En nuestro caso los proyectos de MFAC y RAZ se encargan de la IA del juego, el de GAA de los gráficos, y el de JRV de la arquitectura

general del juego y de la unión de todos los módulos para formar el proyecto final.

- En segundo lugar, proporcionar un contexto para probar la metodología ABP en un proyecto de dimensiones mayores que las de una práctica tradicional. Es decir, un contexto más cercano al mercado laboral.
- Por último, utilizar la temática de los videojuegos para fomentar la motivación del alumnado. Esta motivación interviene en dos momentos: a la hora de escoger la asignatura para matricularse y a la hora de dedicarle más tiempo cuando la estén cursando.

3.3. Docencia del proyecto

Durante los primeros días de cada una de las asignaturas se realiza una pequeña introducción donde se explica a los alumnos el cambio de metodología, además de todo lo relacionado con el proyecto y el software de trabajo en grupo. Para la realización del proyecto se forman grupos reducidos de 3 a 4 personas. El trabajo se realiza en el aula, por lo que las clases de teoría y de prácticas se fusionan. Para adaptarnos a la organización actual de la Universidad se siguen impartiendo en el mismo horario y aula, o en el laboratorio de prácticas correspondiente a cada una.

El principal cambio a realizar en la docencia es la desaparición de la lección magistral. En su lugar se imparten breves seminarios sobre temas de interés, en los que sólo se proporciona la información más importante sobre la materia. El propio alumno será el encargado de obtener el resto de información complementaria en base a las directrices del profesor y a las necesidades derivadas de la realización del proyecto. La labor de investigación está guiada por una serie de problemas que va planteando el tutor, y que están encaminados a la consecución del proyecto. De esta manera la labor es compartida: el profesor tutoriza el avance de la investigación y el estudiante se encarga de buscar la bibliografía.

Cada cierto periodo de tiempo se irán impartiendo seminarios sobre temas avanzados o sobre aspectos concretos que el profesor considere que están presentando una mayor dificultad.

3.4. Evaluación del proyecto

La evaluación también ha sido modificada. Desaparece el examen. La calificación de la asignatura se obtiene evaluando el proyecto realizado por el alumno y su labor de investigación. Este es uno de los factores más importantes de la metodología ABP y, como se verá más adelante, uno de los más valorados por los alumnos.

En las primeras sesiones se les proporciona material bibliográfico para que vayan preparando el terreno. Con todo esto el alumno debe hacerse una idea de la complejidad del proyecto, el tiempo que va a necesitar o los posibles problemas puntuales con los que podría encontrarse.

En el momento en que los alumnos ya tienen la perspectiva adecuada, se les indica que dividan el trabajo en partes y realicen una planificación a seguir hasta la fecha de entrega final. Esta fecha es la misma para todos los grupos y coincide con el final del cuatrimestre. Cada grupo debe marcar en su planificación una serie de hitos intermedios que correspondan a cada una de las entregas de las diferentes partes del proyecto. La planificación de los hitos es muy importante, ya que cada grupo se hace responsable de su carga de trabajo. De esta forma pueden elegir las fechas que más les convengan y adecuarlas a sus horarios de exámenes o de trabajo, quitando peso de unas entregas y sobrecargando otras para ajustarlo a su agenda.

Para asemejar el proceso a un contexto empresarial, los grupos parten de un presupuesto. Dicho presupuesto equivale a la nota final, un 10 en total. Los alumnos asignan a cada hito una parte del presupuesto, es decir, un porcentaje de la nota final. Para cada hito deben indicar claramente los entregables que se van a presentar y la valoración que se le va a dar a cada uno de ellos. Una vez elaborada la planificación y el presupuesto, se le entrega al profesor el informe. El profesor adquiere en este momento el rol de una empresa y puede aprobar o no el presupuesto. Si no lo aprueba, les indica a los alumnos las cosas a cambiar para que lo vuelvan a elaborar. Este ciclo debe realizarse para evitar problemas durante el desarrollo y para que el presupuesto sea realista.

3.5. Elección e implantación de una herramienta de trabajo colaborativo.

En el ámbito del Aprendizaje Basado en Proyectos se hace necesaria la correcta utilización de las Tecnologías de la Información y la Comunicación

(TIC). El ABP tiene unos grandes requerimientos en cuanto a comunicación e intercambio y gestión de la información. Por lo que el uso de la herramienta adecuada nos puede ser de gran ayuda a la hora de gestionar todo el proyecto y de mejorar la comunicación entre los miembros del grupo. A estas aplicaciones se las conoce como *groupware* (software para el trabajo en grupo) [9] y aportan soluciones a las diversas situaciones que se pueden producir durante el desarrollo de proyectos (de cualquier tamaño) por un grupo de usuarios.

Para la realización del proyecto que aquí se plantea se analizaron varias de estas herramientas, buscando que facilitasen la coordinación de las diferentes asignaturas y que aportasen utilidades para la gestión de proyectos y la comunicación.

La herramienta escogida para la realización del proyecto ha sido *GForge* [3], las principales características que motivaron esta elección fueron su versatilidad y potencia. *GForge* incorpora una amplia variedad de utilidades que facilitan la gestión de la información entre los miembros del grupo, entre ellas destacamos:

- Encuestas: Permite crear preguntas concretas para que los demás las respondan o valoren.
- Foros: Maneja y almacena las discusiones creadas por los miembros del grupo.
- Listas de correo: Se encargan de transmitir los mensajes de correo a las personas indicadas en la lista.
- Noticias: Avisos sobre eventos concretos (fechas de entrega, días de reunión, etc.).
- Gestor de Tareas: Su función es asignar y gestionar las tareas entre los integrantes del grupo.
- Registro: Realiza un seguimiento de los bugs y parches realizados en el código.
- Gestor de Documentos: Permite colgar ficheros para compartirlos.
- *Subversion*: Control de versiones

Un factor clave que motivó esta elección frente a otro tipo de herramientas, fue que *GForge* cuenta con *SubVersion* [8] como control de versiones integrado en la aplicación. *Subversion* es una herramienta que permite que varias personas puedan acceder al mismo proyecto y trabajar a la vez con los mismos ficheros. Esta función resulta especialmente útil cuando se trabaja en grupo, porque no es necesario esperar a que otras personas hayan acabado de utilizar un fichero. Además guarda una versión de cada uno de los cambios realizados, por lo que siempre se mantiene un registro y en cualquier momento se puede volver a un estado anterior.

Una vez escogida la herramienta a utilizar se procedió a su instalación en un servidor del grupo de investigación. El aspecto de las páginas fue modificado para darle una apariencia similar a las demás Webs del grupo. Se decidió ponerle el nombre *ABP-Forja* para que los alumnos asocien ambos conceptos y la identifiquen como una herramienta para estas asignaturas. A continuación se proporcionó a los alumnos las instrucciones necesarias para que se pudieran registrar y formar grupos, así como los manuales para que aprendieran a utilizar las diferentes funcionalidades de la herramienta.

A partir de este momento cada grupo se gestionó de forma autónoma. Los propios alumnos crearon los nuevos proyectos y registraron al resto de miembros del grupo, asignándoles los roles y permisos que ellos consideraron oportunos. También decidieron la manera en que utilizarían cada una de las funcionalidades de *ABP-Forja* para gestionar su proyecto.

Los roles permiten definir los permisos que tendrán los miembros del grupo a la hora de ver y modificar cada uno de los aspectos del proyecto. Por ejemplo, habrá usuarios que sólo tendrán permiso para consultar algunos apartados del proyecto, pero no para modificarlos. Y otros usuarios, con rol de “administrador”, que tendrán control total sobre el proyecto. Cada grupo ha configurado estos permisos según ha creído conveniente. Dentro de los roles existe una cierta jerarquía por la cual los de nivel más alto pueden modificar los permisos a los de nivel más bajo. Además existe un rol especial, que es el de “profesor” o “tutor”, el cual puede observar en todo momento los avances realizados en cada proyecto.

Otra modificación realizada, respecto a la forma de trabajo normal de *GForge*, ha sido el uso de grupos privados. Por defecto, cualquier miembro de la forja (aunque sea de otro proyecto) puede utilizar el trabajo realizado por el resto de la comunidad de usuarios. Esta característica del sistema proviene de la filosofía de desarrollo del software libre, ya que el propio *GForge* lo es.

Pero esto no tiene sentido en nuestro caso, ya que se trata de trabajos de alumnos que deben ser genuinos y no permitir la copia. Por este motivo, en nuestro servidor todos los proyectos son privados y solamente pueden acceder los miembros autorizados.

4. Evaluación de la experiencia

Para evaluar los resultados de la experiencia se han utilizado los datos estadísticos sobre el uso de la aplicación de trabajo en grupo *ABP-Forja*, los resultados de las evaluaciones, así como las opiniones de los alumnos obtenidas a partir de las encuestas que se les pasó al final del primer cuatrimestre, por lo que corresponden a las dos primeras asignaturas (MFAC y RAZ). Las encuestas fueron contestadas por 25 alumnos.

Una primera parte del cuestionario pedía a los alumnos que valoraran de forma cuantitativa diferentes aspectos, tanto de actividades de teoría como de prácticas, según los siguientes criterios: la dificultad estimada (1 = poca dificultad; 5 = dificultad máxima), el interés que la actividad le había despertado (1 = ningún interés, 5 = mucho interés) y la satisfacción con los resultados obtenidos (1 = nada satisfecho, 5 = muy satisfecho).

Los resultados han permitido obtener una visión general del desarrollo de la metodología, conocer sus puntos fuertes y débiles, y proponer posibles mejoras para cursos sucesivos. También se han podido valorar otros aspectos como las dificultades, intereses o satisfacciones con respecto al software de trabajo en grupo y al resto de temas y actividades.

En la figura 1 se pueden observar los resultados de las encuestas en lo referente a la metodología ABP utilizada. Un primer análisis nos permite observar que la metodología no ha resultado difícil para los alumnos. Además, si la comparan con otras, su dificultad les ha parecido similar pero les ha causado más interés y satisfacción.

La dificultad de la interdisciplinariedad les ha parecido más alta que el interés y la satisfacción que les ha causado. Esto es debido a que se trataba en su mayoría de alumnos que no tenían escogidas las 4 asignaturas, por lo que la relación entre ellas no les ha importado demasiado.

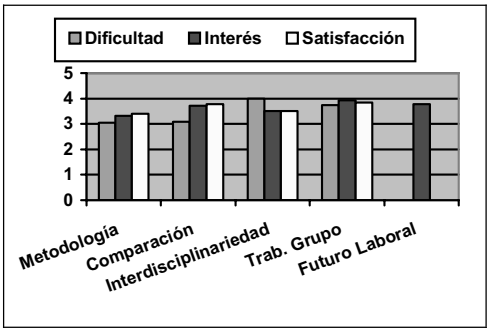


Figura 1. Valoración de la Metodología.

El trabajo en grupo ha sido el que más interés y satisfacción ha generado. Por último, se les preguntó a los alumnos el grado de interés que les generaba la metodología de cara a un futuro laboral, los alumnos la valoraron positivamente.

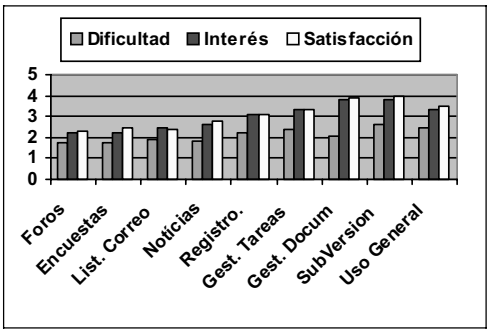


Figura 2. Valoración de ABP-Forja.

La figura 2 muestra las opiniones de los alumnos sobre la aplicación de trabajo en grupo *ABP-Forja.* En este último gráfico, las 8 primeras variables hacen referencia a las herramientas concretas que la integran, y la última se centra en el gestor de proyectos completo. Se han ordenado las herramientas de menor a mayor uso. Desde un primer momento llama la atención que el interés y satisfacción generados por cada herramienta ha sido superior a su dificultad. Las herramientas peor valoradas han sido las que menos uso han tenido, concretamente las 4 primeras (foros, encuestas, listas de correo y noticias) han tenido una utilización muy escasa.

El interés y satisfacción del gestor de tareas y de los registros ha sido valorado positivamente por parte de los alumnos. El uso de estas dos herramientas ha sido curioso, ya que los grupos

los han usado muy poco o lo han hecho masivamente, no ha habido término medio. Las dos herramientas más valoradas y utilizadas han sido el gestor de documentos y el *SubVersion.* Además, han sido también las que más dificultad han presentado, seguramente porque han sido en las que más esfuerzo pusieron los alumnos por sacar el máximo partido. Finalmente la valoración de la dificultad del uso general de *ABP-Forja* ha sido ligeramente superior a la media de las herramientas por separado. Este factor es debido a que, en ocasiones, la interfaz web de *GForge* resulta poco intuitiva.

Por último se realizó un cuestionario cualitativo, en el que se dejaba que los alumnos opinaran de forma abierta sobre los diferentes aspectos de la metodología. A continuación se muestran algunos de los resultados más significativos.

La mayoría de ellos ha considerado la búsqueda del material como una experiencia positiva, aunque en algunos momentos les ha causado un poco de incertidumbre. Respecto a la forma de evaluación muchos alumnos han valorado muy positivamente que no exista examen. Sin embargo, una pequeña parte ha comentado que no le parece justo que la nota sea la misma para todo el grupo, ya que no todos los integrantes han trabajado lo mismo.

La valoración de la herramienta *ABP-Forja* ha tenido respuestas muy variadas. Por sus respuestas podemos deducir que 4 alumnos no la han usado en absoluto y 17 sí le han sacado provecho. De estos últimos, 10 comentan que lo que más han usado ha sido el *SubVersion* y el gestor de documentos. Esto coincide con la encuesta cualitativa, en la que estas dos herramientas eran las más valoradas. Un alumno llegó a comentar que no sabía como podía haber vivido hasta ahora sin haber usado *SubVersion.*

La configuración de los grupos ha condicionado mucho el uso que se le ha dado a *ABP-Forja*, ya que la mayoría de integrantes eran amigos con anterioridad o coincidían a menudo. Solamente 2 personas comentaron que no conocían de nada a los miembros de su grupo hasta el momento de formarlo. Respecto a la comunicación entre los integrantes, 16 personas indicaron que se reunían periódicamente para realizar las tareas y para coordinarse. El mail y las herramientas de comunicación instantánea fueron los métodos elegido por otros 7 a la hora de comunicarse.

El trabajo en grupo ha sido valorado positivamente. Algunos alumnos manifestaron que habían tenido dificultades, debidas a las grandes diferencias de aptitudes y objetivos de los integrantes del equipo, pero que habían sabido solucionarlas internamente. Un par de grupos tuvieron quejas sobre componentes que no habían trabajado lo suficiente. Como anécdota cabe destacar que, durante el curso, un grupo tuvo que ser dividido en dos por el tutor. Esta decisión fue debida a que sus continuas discusiones hacían peligrar su trabajo y su calificación.

Por otro lado, según la mayoría de los encuestados, tener un videojuego como temática del proyecto común ha contribuido a aumentar su dedicación en horas a las asignaturas implicadas.

Por último, los alumnos valoraron muy positivamente la experiencia, solamente un alumno indicó que no le había gustado y que no la recomendaría a otros compañeros, frente a 24 que sí lo harían. Las características más valoradas han sido: tener libertad de elegir el camino a seguir, el trabajo en grupo, el método de evaluación y realizar un proyecto más cercano a la realidad. Por el contrario, lo que menos les ha gustado ha sido el incremento de trabajo y la incertidumbre del tener que buscar ellos mismos la información.

5. Conclusiones

La docencia tradicional es un modo de docencia que sólo tiene sentido en un contexto de masificación, pero no es adecuado para formar profesionales que trabajen en equipo, pues no permite la toma de decisiones o responsabilidades, no fomenta el desarrollo de la imaginación ni permite que se expresen adecuadamente. El Aprendizaje Basado en Proyectos (ABP) intenta minimizar las limitaciones de la docencia tradicional y fomentar habilidades tan importantes como son: el trabajo en grupo, el aprendizaje autónomo, la planificación del tiempo, las capacidades de expresión oral y escrita, las habilidades intelectuales de alto nivel y la capacidad de auto evaluación. Esta primera experiencia de desarrollo del proyecto nos ha servido como banco de pruebas para determinar el beneficio de la implantación del ABP sobre el modelo tradicional, y a la vez nos ha permitido

identificar algunas de las herramientas más útiles para este tipo de proyectos.

Desde nuestro punto de vista hemos conseguido inculcar la metodología de trabajo en grupo, con todo lo que ello implica (división de tareas, informar a los demás de los avances/problemas, documentación del proyecto, etc.), además de algunas pautas que el alumno se encontrará en el mercado laboral (proyectos de gran tamaño, planificación de hitos, presupuestos, etc.).

El cambio de metodología ha sido valorado positivamente por los alumnos, especialmente el trabajo en grupo y la forma de evaluación. Sin embargo, no les has gustado el incremento de trabajo que supone la búsqueda de información. Este incremento de trabajo también es compartido por el profesorado, que tiene que estar más pendiente de la evolución de los grupos y de sus problemas concretos. Además de la dificultad propia del cambio de metodología, la tarea de coordinar las 4 asignaturas para conseguir la interdisciplinariedad ha resultado en algunos momentos costosa para el profesorado. Sobre todo en lo que a comunicación y tiempo invertido se refiere (reuniones, correos, documentación, etc.).

Este incremento de trabajo se ha traducido, en la mayoría de los casos, en muy buenos resultados, ya que se han realizado proyectos de mucha calidad, que no hubieran sido posibles en asignaturas individuales.

Respecto a la utilización de la herramienta de trabajo colaborativo *ABP-Forja*, podemos decir que ha tenido un uso intensivo y ha ayudado mucho en los problemas de comunicación, especialmente en los grupos que no podían reunirse con frecuencia. De entre sus características, los alumnos han destacado la gestión de tareas, el control de versiones y el gestor de documentos.

A corto plazo el proyecto va a continuar durante el segundo cuatrimestre, donde se unirán los trabajos individuales de las 4 asignaturas y se efectuará un análisis más profundo de la nueva experiencia. A medio plazo se pretende mejorar las herramientas de trabajo colaborativo utilizadas e implantar la metodología ABP en otras asignaturas y proyectos.

Agradecimientos

Este trabajo cuenta con una ayuda para Proyectos de Innovación Tecnológico-Educativa 2006 del Vicerrectorado de Tecnología e Innovación Educativa de la Universidad de Alicante, proyecto "Nuevas metodologías docentes ante el EEES: Aprendizaje Basado en Proyectos y su implementación con tecnologías para el trabajo colaborativo". Además de los firmantes del artículo, forman parte de dicho proyecto los profesores Fidel Aznar, Francisco José Gallego, Faraón Llorens, Francisco Mora, Mar Pujol, Mireia Sempere y Carlos Villagrá, a los que deseamos agradecer su trabajo sin el cual no hubiera sido posible esta experiencia.

Referencias

[1] D.R. Woods, *Problem-based Learning: How to Gain the most from PB*. Donald R.Woods, McMaster University,1994

[2] European Ministers of Education (1999). The European Higher Education Area Bologna Declaration, Bologna.

[3] Gforge. http://en.wikipedia.org/wiki/GForge

[4] J. Bará, M Valero. *Taller de formación: Aprendizaje Basado en proyectos*. Universitat Politècnica de Catalunya. 2006

[5] J.E. Labra, D. Fernández, J. Calvo, A. Cernuda, *Una experiencia de aprendizaje basado en proyectos utilizando herramientas colaborativas de desarrollo de software libre.* XII Jornadas de Enseñanza Universitaria de la Informática (JENUI), 2006.

[6] Markham T. *Project based-learning hand-book*. Buck IFE, 2003.

[7] Scholtes P.R. *The Leader's Handbook, A guide to inspiring your people and managing the daily workflow*. McGraw Hill, 1998.

[8] SubVersion. http://es.wikipedia.org/wiki/Subversion

[9] Tom Brinck. "Groupware". 1998. http://www.usabilityfirst.com/groupware/intro.txl

Una experiencia docente orientada a incrementar el trabajo personal del estudiante

Beatriz Otero Calviño[1], Jaume Martí-Farré[2], Ernest Garriga Valle[2], Arantxa Alonso Maleta[3] y Lluis Prat Viñas[4]

[1]Dpto. de Arquitectura de Computadores,
e-mail: botero@ac.upc.edu
[2]Dpto. de Matemática Aplicada IV,
e-mail: {jaumem@ma4.upc.edu, egarriga@ma4.upc.edu}
[3]Dpto. de Física Aplicada,
e-mail: arantxa@fa.upc.edu
[4]Dpto. de Ingeniería Electrónica
e-mail: prat@eel.upc.edu
Universidad Politécnica de Cataluña
Campus Nord, 08034 Barcelona

Resumen

Este artículo describe las diferentes metodologías docentes aplicadas a un grupo de estudiantes de la *ETSETB-UPC*. Particularmente, el grupo de estudiantes a los que nos referimos tienen matriculadas las cinco asignaturas troncales del primer cuatrimestre (1A) de la fase selectiva.

El objetivo principal de esta experiencia docente es mejorar el rendimiento académico del grupo, utilizando metodologías docentes más activas. Estas metodologías contribuyen a aumentar el trabajo personal que el estudiante debe realizar fuera de las horas de clase. Este trabajo muestra los cambios introducidos, relacionados con la organización, la metodología y la evaluación de cada asignatura.

Como referencia, la cantidad de horas semanales de trabajo personal del estudiante debe coincidir con el número total de horas de clase semanales de cada asignatura. Las actividades, individuales o por grupos, son planificadas y diseñadas considerando estas horas.

Para realizar una valoración de la experiencia, utilizamos como métrica de comparación la nota final obtenida en cada asignatura. Adicionalmente, utilizamos una segunda métrica de comparación, el porcentaje de estudiantes que obtuvieron una nota mayor o igual a 4.0 puntos sobre 10 en el examen final. En general, los resultados muestran que el rendimiento académico del grupo experimental supera, en media porcentual, el rendimiento medio obtenido en cada asignatura, tanto en la nota final como en la nota del examen final.

Concluimos el trabajo comentando algunas mejoras que introduciremos en las metodologías aplicadas y que se pondrán en práctica para el próximo cuatrimestre.

1. Motivación

Actualmente, son muchas las experiencias docentes que se están poniendo en práctica para mejorar el rendimiento académico de los estudiantes en los primeros cursos de las carreras de ingeniería. Muchas de éstas experiencias proponen estrategias de aprendizaje [1] [2] [6], otras introducen metodologías docentes innovadoras al impartir las asignaturas de sus cursos [3] [4].

Durante el cuatrimestre de otoño 2006-2007 y como iniciativa de la dirección de la Escuela Técnica Superior de Ingeniería de Telecomunicación de Barcelona (*ETSETB-UPC)*, se decidió realizar una experiencia docente para estudiar el efecto que causaría, en el rendimiento académico, incrementar las horas de dedicación del estudiante en cada una de las asignaturas del curso 1A. Para realizar esto se contó con la colaboración de un grupo de profesores que impartían docencia en estas asignaturas.

Esta iniciativa surge a partir de la necesidad que hay de prepararnos para los nuevos planes de estudio del *EEES* (*Espacio Europeo de Educación Superior*) y la puesta en marcha del sistema de créditos *ECTS* (*European Credit Transfer*

System). A partir de ahora, el protagonista del paradigma educativo será el alumno, no el profesor [5]. Además, la falta de interés y el poco esfuerzo que realizan los estudiantes a la hora de llevar al día las asignaturas, son factores que motivaron la experiencia. Pensamos que si existe un seguimiento más cercano, por parte del profesor, de lo que diariamente realiza el estudiante resultará más fácil corregir su trabajo y motivarle para que continúe mejorando.

Los objetivos principales de la experiencia fueron:

- Motivar el trabajo personal y en grupo, diseñando actividades que resulten atractivas/entretenidas para el estudiante.
- Incrementar las horas de estudio que el estudiante dedica a cada asignatura fuera de las horas de clase.
- Ofrecer un *feedback* personalizado para mejorar el aprendizaje del estudiante.

En general, queremos trabajar para evitar la actitud pasiva de los estudiantes, actitud ésta que ha sido observada en ediciones anteriores del curso. Pensamos que si existen actividades que fuercen a que el estudiante se dedique, ésto motivará su trabajo a lo largo del curso. Además, el *feedback* implica progreso, con lo cual si un estudiante percibe cierta mejora en su aprendizaje lo motivará a que continúe trabajando.

Además, actualmente es muy común encontrarnos con alumnos que ingresan a los primeros cursos de la Universidad, como sucede en el curso 1A, que no están acostumbrados a dedicar demasiadas horas a estudiar. La idea es promover el trabajo personal del estudiante después de clase, ya sea en casa, o en el Campus universitario, para crear hábitos de estudio.

2. Organización

El curso 1A esta formado por 5 asignaturas troncales. Estas asignaturas son: Física I, Introducción a los Ordenadores (IO), Circuitos y Sistemas Electrónicos I (CISE I), Álgebra y Cálculo.

Debido a la cantidad de estudiantes que hay en los primeros cursos de la fase selectiva, las asignaturas se organizan en grupos. Cada grupo se divide a su vez en subgrupos, de manera que es posible organizar el trabajo en el laboratorio. Particularmente, la experiencia docente la aplicamos a los estudiantes del grupo 30, formado por 30 estudiantes. Este grupo se dividió en dos subgrupos.

Para comenzar a planificar el trabajo del estudiante, consideramos primordial estimar la cantidad de horas semanales que el estudiante debía dedicar a cada asignatura. De esta forma, determinamos que por cada hora de clase, el estudiante tenía que hacer una hora de trabajo personal. Además, debía dedicar 20 horas adicionales para preparar el examen final de cada asignatura. En base a este cálculo, los profesores planificamos y diseñamos actividades que no excedieran la dedicación en horas que cada estudiante debía dedicar a cada asignatura. Esto es difícil de estimar y pensamos que en la medida en que trabajemos podremos acotar estos tiempos.

Por otra parte, la experiencia no pretende influir en el estilo que cada profesor tiene de hacer sus clases. La experiencia mantiene el estilo y la metodología que cada profesor ha elegido para impartir sus clases. Cada uno ha utilizado la metodología docente que considera más apropiada para impartir docencia en su asignatura.

Sin embargo, a pesar de esta autonomía en esta experiencia docente hubo una integración entre todos los profesores. Para ésto, se organizaron reuniones en donde se explicaba la actividad que desarrollaba cada profesor con el grupo. El objetivo principal de estas reuniones era coordinar la actividad que se realizaba en cada asignatura de manera que no perjudicara la ejecución de las actividades propuestas en otras asignaturas. Por ejemplo, para evitar que los estudiantes se desbordaran de trabajo planificamos las fechas en las que se realizarían los controles de cada asignatura y las fechas de entrega de las actividades. Además, los profesores aportábamos sugerencias para mejorar la experiencia tanto en la propia asignatura como en el resto.

3. Metodología aplicada en cada asignatura

A continuación comentaremos los cambios introducidos en cada una de las asignaturas del

curso 1A para el cuatrimestre de otoño 2006-2007. Estas modificaciones pueden hacer referencia a cambios relacionados con la organización, la metodología y/o la evaluación.

Para hacer la planificación del curso y organizar el trabajo del estudiante, se introduce *Moodle.* En la *ETSETB-UPC* esta herramienta está montada en el Campus Digital llamado *ATENEA* [7].

3.1. La asignatura de Física I

En esta asignatura se diseñaron tareas obligatorias que el estudiante debía realizar en casa para entregar en clase. Esta actividad se realizaba por parejas. Entre las actividades propuestas se encuentran la realización de *tests* y la solución de problemas. En total, se prepararon 10 *tests* formados por 10-25 preguntas y se propusieron 4 problemas.

Con respecto a las clases de teoría, se mantuvo la metodología de clases expositivas para explicar los contenidos teóricos y realizar problemas modelo.

ATENEA se utilizó como un entorno virtual de trabajo que permitía intercambiar con los estudiantes, enunciados de tareas/problemas, respuestas de ejercicios, solución de *tests* y enunciados de exámenes finales resueltos. Fundamentalmente, se ha tratado de presentar la información del curso de una forma ordenada, siguiendo la estructura por temas y por semanas de clase. De esta forma, se pretendía guiar al estudiante organizándole el trabajo que tenía que realizar.

La nota final de la asignatura para todos los grupos es calculada utilizando la siguiente fórmula:

$$Nota_{Fisica} = Max(Nota_{Final_1}, Nota_{Final_2})$$

donde:

$$Nota_{Final_1} = 20\% Nota_{Teoria} + 20\% Nota_{Laboratorio} + 60\% Nota_{Examen_Final}$$

$$Nota_{Final_2} = 20\% Nota_{Laboratorio} + 80\% Nota_{Examen_Final}$$

En el caso del grupo 30, la nota de teoría es la nota media obtenida en dos controles y en las 14 tareas.

3.2. La asignatura de IO

Esta asignatura organiza el temario en cinco unidades de trabajo. Al finalizar cada unidad de trabajo, el estudiante debía entregar de forma individual un ejercicio previamente especificado por el profesor. Además de las entregas de los ejercicios individuales, el estudiante realiza en horas de clase y de forma individual dos controles. Como herramienta de trabajo, se utiliza *ATENEA* para organizar los contenidos, entregar los ejercicios, evaluar las tareas, atender consultas y determinar las próximas actividades a realizar.

Por otra parte, en las clases de teoría, se reduce en un 75% el uso de las clases expositivas para explicar los contenidos de la asignatura. Los apuntes de clase se presentan digitalizados y organizados por unidades de trabajo. El estudiante va a las clases con este material y utiliza la teoría para hacer ejercicios prácticos.

En el resto de las clases de teoría, el profesor realizó un resumen de los contenidos teóricos de cada unidad de trabajo, y resolvió ejercicios de exámenes finales de cuatrimestres anteriores. Además, durante las 5 últimas semanas del cuatrimestre se utilizaron las dos horas a la semana de clase de teoría para hacer ejercicios prácticos en el laboratorio. Estas actividades fueron preparadas previamente por el profesor.

En esta edición, el profesor organizó un taller de preparación para el examen final. Este taller se organizó considerando dos sesiones. En la primera sesión, se aclararon las dudas de los estudiantes y se comentaron los errores más frecuentes que suelen cometer los estudiantes al resolver los problemas. En la segunda sesión del taller, se realizó un simulacro de examen final, utilizando ejercicios de exámenes finales anteriores. Al finalizar la sesión, se presentaron las soluciones para cada pregunta del examen final.

Con respecto al laboratorio, se cambia la filosofía de trabajo. Se pasa de la realización de un gran proyecto a lo largo del cuatrimestre, al desarrollo de pequeños proyectos asociados a cada unidad de trabajo.

La nota final de la asignatura, para todos los grupos, está determinada por la siguiente fórmula:

$$Nota_{IO} = Max(Nota_{Final_1}, Nota_{Final_2})$$

donde:

$$Nota_{Final_1} = 25\% Nota_{Teoria} + 25\% Nota_{Laboratorio} + 50\% Nota_{Examen_Final}$$

$$Nota_{Final_2} = 25\% Nota_{Laboratorio} + 75\% Nota_{Examen_Final}$$

Particularmente, la nota de teoría del grupo 30 ha sido calculada como:

$$Max\left(Nota_{Teoria_1}, Nota_{Teoria_2}\right)$$

con

$$Nota_{Teoria_1} = (40\% Nota_{Control_1} + 60\% Nota_{Control_2})$$

$$Nota_{Teoria_2} = 85\%(40\% Nota_{Control_1} + 60\% Nota_{Control_2}) + 15\% Ejercicios_{Unidad}$$

Básicamente, se pretende motivar el trabajo continuado del estudiante y utilizarlo como herramienta para mejorar las notas de los controles y del examen final.

3.3. La asignatura de CISE I

Esta asignatura está organizada en cinco módulos. Al finalizar cada módulo, el estudiante realiza un control.

La metodología de trabajo de esta asignatura requiere un esfuerzo previo a la clase, por parte del estudiante. Este esfuerzo exige que el alumno dedique horas de estudio para entender los contenidos teóricos de la asignatura. Esto pretende que el estudiante se habitúe a aprender por el mismo, cuestión que consideramos imprescindible al realizar estudios universitarios.

De esta forma, las clases de teoría se dedican a resolver dudas y aplicar los conceptos/ideas trabajadas en problemas/ejercicios concretos.

Antes de finalizar cada clase de teoría, el profesor indicaba los apartados del libro de texto que el estudiante debía preparar, y el cuestionario que debía entregar al iniciar la próxima clase. Adicionalmente, el estudiante debía resolver un ejercicio de la colección de problemas, que sería revisado por el profesor en la siguiente clase.

Para finalizar la clase de teoría, cada estudiante debía resolver y entregar un ejercicio propuesto por el profesor. Este ejercicio se devolvía corregido y formaba parte de la evaluación del módulo.

La nota final de la asignatura, para todos los grupos, esta determinada por:

$$Nota_{CISE_I} = Max(Nota_{Final_1}, Nota_{Final_2})$$

donde:

$$Nota_{Final_1} = (40\% Nota_{Teoria} + 60\% Nota_{Examen_Final})$$

$$Nota_{Final_2} = 100\% Nota_{Examen_Final}$$

En el grupo 30, la nota de teoría es la media de las notas obtenidas en cada módulo. La nota de un módulo es calculada utilizando la siguiente fórmula:

$$Nota_{Módulo} = 50\% Nota_{control} + 50\% Nota_{Ejercicios}$$

Antes de finalizar el cuatrimestre, se realizó una modificación sobre el plan de trabajo inicial, que consistió, principalmente, en eliminar la obligatoriedad de entregar los cuestionarios.

Además, los ejercicios que el estudiante debía preparar para resolverlos al iniciar la clase también fueron omitidos. Esta modificación se realizó a petición de los estudiantes, debido a que se encontraban desbordados de trabajo y esto les impedía realizar el trabajo propuesto en el resto de asignaturas.

3.4. Las asignaturas de Álgebra y Cálculo

La organización de la docencia en las asignaturas de Álgebra y de Cálculo para esta experiencia se ha modificado, respetando las horas semanales de clase obligatorias (5 horas). Básicamente, las sesiones de clase se organizan en dos grupos de trabajo. El primero, formado por el grupo completo de estudiantes, y el segundo por subgrupos. De esta forma, el grupo completo semanalmente recibía 3 horas de clase de teoría, en vez de 4 horas 30 minutos, y cada subgrupo recibía 2 horas en vez de 30 minutos. Para el profesor las horas de clase a la semana se incrementaron en 1 hora.

En las sesiones con el grupo completo (sesiones teóricas) el profesor presenta la teoría y la ilustra con ejercicios y problemas ejemplares. Además, en estas clases el profesor distribuía y comentaba entre los estudiantes un guión que resumía los contenidos de la clase. Para ésto, fue necesaria la incorporación en el aula, de un proyector y un portátil. De esta forma, el

estudiante se concentraba en las explicaciones del profesor y no se preocupaba de tomar apuntes.

En los subgrupos, el alumno trabaja diferentes aspectos de la materia explicada en las sesiones teóricas. Estas sesiones son guiadas y tutorizadas por el profesor y se realizaban por parejas. Además, se introduce el uso de herramientas *software* que permiten la manipulación de operadores simbólicos, indispensables para realizar numerosos ejercicios gráficos y de cálculo. Estas sesiones se pueden realizar bien en el aula normal de clase o bien en aulas de laboratorio provistas de ordenadores y de herramientas software. Los ejercicios podían realizarse en papel o bien, utilizando el ordenador. La primera modalidad de trabajo permitía que el estudiante comprendiera los procedimientos, mientras que la segunda modalidad simplificaba los cálculos.

Para reforzar el trabajo que el estudiante dedica a la asignatura fuera de las horas de clase, el profesor sugiere realizar ejercicios específicos. Estos ejercicios son evaluados y forman parte de la nota de laboratorio.

La nota final de estas asignaturas, para todos los grupos, se calcula utilizando la siguiente fórmula:

$$Nota_{Calculo_Algebra} = Max(Nota_{Final_1}, Nota_{Final_2})$$

donde:

$$Nota_{Final_1} = 30\% Nota_{Teoria} + 10\% Nota_{Laboratorio} + 60\% Nota_{Examen_Final}$$

$$Nota_{Final_2} = 10\% Nota_{Laboratorio} + 90\% Nota_{Examen_Final}$$

La nota de teoría del grupo 30 se calcula como la media de la nota obtenida al realizar dos controles. La nota de laboratorio es el promedio de las notas obtenidas por el estudiante en las 13 sesiones de laboratorio. Esta nota evalúa el esfuerzo y el trabajo que los estudiantes han dedicado durante todo el cuatrimestre a la asignatura.

4. Resultados

Para evaluar el rendimiento de los estudiantes en el grupo 30 hemos utilizado como métrica de comparación, el porcentaje medio de estudiantes aprobados con respecto a los estudiantes matriculados en cada asignatura.

Para que esta valoración sea más objetiva hemos excluido de este estudio a los estudiantes con menos de 5 asignaturas. Esto nos permitirá definir una población de estudio más homogénea. Además, no consideramos a los estudiantes que cursaban estudios en la modalidad de *vía lenta*, ni a los estudiantes que hacían dos carreras en paralelo (*CFIS*). De esta forma, la cantidad total de estudiantes que participan en el estudio son 216. Estos estudiantes se encuentran distribuidos en 7 grupos. La figura 1 muestra el porcentaje de estudiantes aprobados en cada grupo y en cada asignatura.

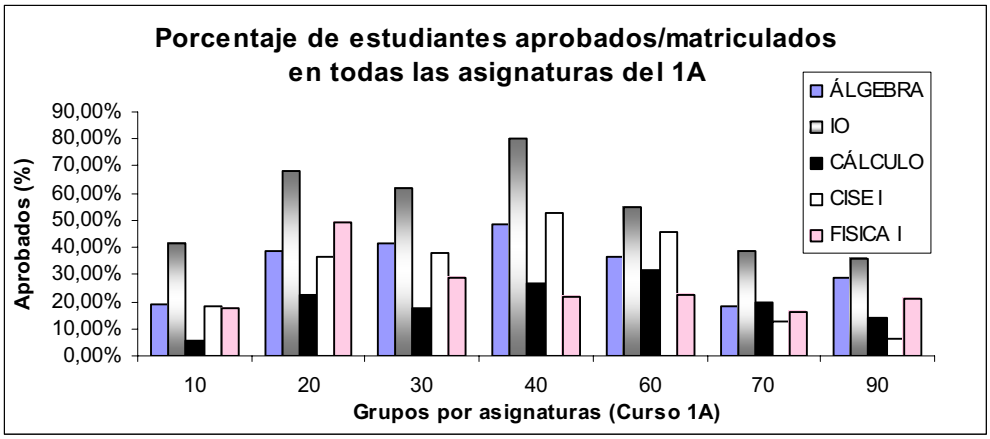


Figura 1. Porcentaje de estudiantes aprobados/matriculados por asignatura

Los resultados obtenidos no son tan buenos como esperábamos. Sin embargo, y a pesar de que no somos el grupo líder en todas las asignaturas, podemos decir que para todas las asignaturas, nuestros porcentajes de aprobados se encuentran por encima de los porcentajes medios obtenidos (nota final) en cada asignatura (todos los grupos).

En general, el cálculo de la nota final es variable entre asignaturas y entre grupos, ya que la nota de teoría puede ser determinada utilizando diferentes fórmulas de cálculo. Sin embargo, lo que sí es común para todos los grupos y para todas las asignaturas es el examen final. La nota obtenida en este examen tiene una contribución del 50% al 60% en la nota final, dependiendo de la asignatura. Este examen es realizado y evaluado por todos los profesores que imparten docencia en los diferentes grupos de una misma asignatura. Por esta razón, para valorar la experiencia decidimos incluir un estudio que muestre, por grupos y para cada asignatura, el porcentaje de estudiantes cuya nota en el examen final es superior o igual a 4.0 puntos sobre 10.

La figura 2 muestra los resultados obtenidos de este estudio. Como podemos observar en este caso, el porcentaje de estudiantes del grupo experimental con nota mayor o igual a 4.0 puntos es superior al porcentaje medio obtenido por los estudiantes en cada asignatura.

A la vista de estos resultados pensamos que debemos seguir trabajando. La idea es buena pero debemos conseguir que de forma espontánea el estudiante dedique horas de estudio para cada asignatura fuera del aula. Queremos mejorar la participación de los alumnos, sobre todo en las asignaturas donde se emplean clases expositivas. Pensamos que este compromiso generaría que los estudiantes se interesen más por cada asignatura.

5. Valoración de la experiencia por parte del estudiante

Para valorar la opinión del estudiante con respecto a la experiencia docente decidimos elaborar, entre todos los profesores, una encuesta que recogiera información, para cada asignatura.

La información obtenida de esta encuesta valoraba en una escala del 1 al 5:

- el trabajo en grupos reducidos (subgrupos),
- la utilización del Campus digital como medio para obtener documentos, acceder a las notas y realizar consultas,
- la(s) asignatura(s) a la(s) que dedicaron más tiempo,
- la forma de impartir las clases de teoría,
- las explicaciones del profesor,
- los cambios introducidos en la metodología docente: cuestionarios, *tests*, ordenador (álgebra y cálculo), solución de problemas tipo examen final,
- la frecuencia y la obligatoriedad de las actividades,
- el promedio de horas semanales de trabajo personal dedicado a cada asignatura.

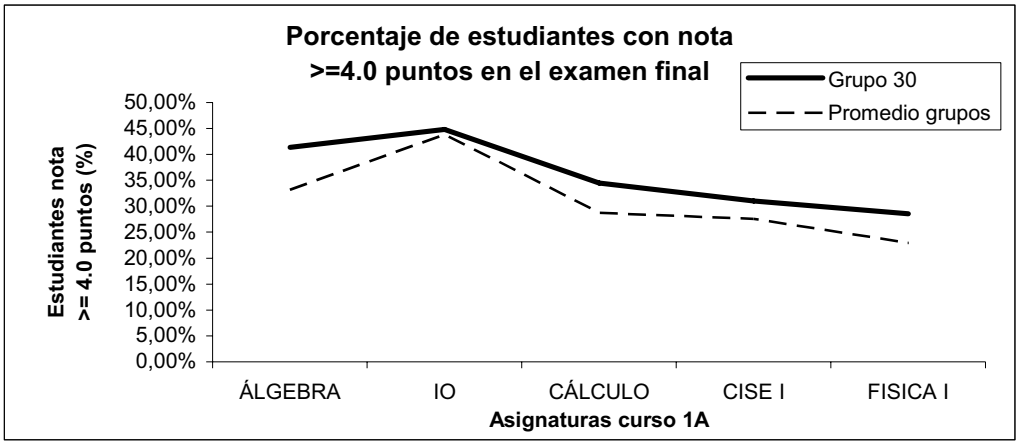


Figura 2. Porcentaje de estudiantes con nota ≥ 4.0 puntos en el examen final

En general, el estudiante ha valorado positivamente el hecho de que se le asigne un trabajo que le ayude a estudiar y a organizar la asignatura fuera de las horas de clase. Sin embargo, en algunas asignaturas manifestaron que no podían cumplir con todas las actividades propuestas por falta de tiempo. Esta reflexión hizo que pensáramos que nos habíamos excedido estimando la duración de las actividades. Sin embargo, al preguntarles las horas de trabajo personal que dedicaban semanalmente, observamos que, para algunas asignaturas, no dedicaban ni el 75% de las horas. La tabla 1 muestra la cantidad de horas promedio semanales que los estudiantes dedicaron a cada asignatura y la cantidad de horas reales de trabajo que exigía cada asignatura.

Asignatura	Horas dedicadas	Horas reales
Física I	2 h 30 min.	5 h
IO	3 h 30 min.	4 h
CISE I	3 h	4 h
Álgebra	3 h 15 min.	5 h
Cálculo	2 h 30 min.	5 h
Total	**14 h 45 min.**	**23 h**

Tabla 1. Promedio de horas semanales de trabajo personal por asignatura

Nuestra sensación es que para ellos es muy difícil pasar del método tradicional (estudiar el día antes del examen), que les ha funcionado hasta ahora, a considerar un método de trabajo continuo para llevar cada asignatura al día. Todos sabemos que adquirir este ritmo de trabajo al ingresar a la Universidad no resulta sencillo. Continuaremos trabajando para que la adaptación al ritmo de trabajo exigido por cada asignatura sea cada vez más rápida.

6. Plan de trabajo: Cuatrimestre de primavera 2007

El cuatrimestre de primavera del curso 1A tiene la particularidad de que todos los estudiantes son repetidores. Esto significa que:

1. Un gran porcentaje de los estudiantes tienen algún conocimiento de la asignatura, por lo que no será necesario repetir la teoría suponiendo que la desconocen completamente.
2. Los estudiantes tienen prisa por cursar otras asignaturas, ya que cuentan únicamente con dos años para aprobar todas las asignaturas de la fase selectiva.

Si consideramos estos factores, en el aula de clase se presentan dos situaciones:

Caso 1. El estudiante que asiste a clase.

Caso 2. El estudiante que no viene a clase.

Ante este panorama, pensamos que es necesario plantear estrategias que vayan dirigidas a ambos grupos. En este sentido, y para los estudiantes que desisten de venir a clase, ofreceremos la posibilidad, en algunas asignaturas, de tener nota de teoría (evaluación continuada) si cumplen con un plan de trabajo. Esto garantizará que trabajen regularmente las asignaturas durante todo el cuatrimestre, aunque no vengan a clase. El plan de trabajo propuesto consistirá en la entrega, para fechas pautadas, de ejercicios/cuestionarios resueltos indicados por el profesor. Sin embargo, nuestro primer objetivo será convencerles de que el trabajo en el aula es provechoso, por lo que la asistencia a las clases teóricas es realmente importante. Por otra parte, y para todos los estudiantes:

- Continuaremos trabajando para conseguir que los alumnos dediquen más tiempo al estudio personal de cada asignatura. Para garantizar esto, las actividades propuestas en cada asignatura tendrán carácter obligatorio.
- Seguiremos con el trabajo en grupos reducidos, ya que hemos mejorado la relación entre el alumno y el profesor.
- Ajustaremos los tiempos de cada actividad a la dedicación semanal que el estudiante tiene para cada asignatura.
- Realizaremos una planificación horaria para realizar el trabajo fuera de las horas de clase. De esta forma, le indicaremos ¿qué día? y ¿cuántas horas? dedicará a cada asignatura.

- Continuaremos utilizando los cuestionarios y los *tests*. Así como también, continuaremos resolviendo problemas tipo examen en horas de teoría.

7. Conclusión

La experiencia docente realizada incrementa las horas de trabajo personal del estudiante, pero no tanto como esperábamos. Sin embargo, los resultados obtenidos por el grupo experimental no son desalentadores. Los porcentajes de estudiantes aprobados en todas las asignaturas se encuentran por encima de los valores medios. También se obtienen buenos resultados en el examen final.

Sin embargo, la experiencia presenta algunos fallos que corregiremos en futuras ediciones del curso. Lo importante, es continuar trabajando para mejorar la calidad de nuestra docencia y el rendimiento académico de nuestros estudiantes.

Agradecimientos

Agradecemos la colaboración de la *ETSETB-UPC* y del Subdirector de Profesorado y Planificación Estratégica, al proporcionarnos las notas de los estudiantes matriculados en todas las asignaturas del curso 1A (cuatrimestre otoño 2006-2007). Agradecemos además, la colaboración prestada por los coordinadores de todas las asignaturas.
Este trabajo ha sido apoyado por el Ministerio de Educación y Ciencia, bajo proyecto TIN 2004-07739-C02-01.

Referencias

[1] Bofill P., Otero B., Toribio E., Aroca J. M., Breitman M., Garcias P. y Sancho J. M. *ÁNCORA: Aprendizaje organizado por tareas*. Actas JENUI 2005, 71-77, 2005.

[2] Fortuna Santos J. y Peña Pitarch E. *Experiencias de innovación educativa mediante la evaluación entre iguales en las asignaturas técnicas.* Actas XII Congreso Universitario de Innovación Educativa en las Enseñanzas Técnicas, 2004.

[3] Otero J. A, Ruiz Ma., Barbón Arsenio y Barbón Nicolás. *Aprendizaje en ingeniería: Nuevas aportaciones*. Póster en XII CUIEET, 2004.

[4] Torralba Martínez J. M., Coltell O. y Torralba López J. M. *Innovación en la enseñanza en grupos numerosos*. Actas CUIEET, 2004.

[5] Sánchez F. *¿Cómo serán las asignaturas del EEES?* Actas JENUI 2005, 147-154, 2005.

[6] Vivaracho Pascual C. E., Simón Hurtado A. y Martínez Monés A. *Aplicación de técnicas de aprendizaje cooperativo en la parte de teoría de una asignatura de primero, con aulas masificadas*. Actas JENUI 2005, 37-44, 2005.

[7] Web *ATENEA*. http://atenea.upc.edu

Uso de herramientas colaborativas que reducen la carga de gestión en la docencia.

Juan José Escribano Otero,
Enrique Puertas Sanz,
César Alberto Escribano Otero
Dpto. de Sistemas Informáticos
Universidad Europea de Madrid
{juanjose.escribano|enrique.puertas|cesaralberto.escribano}@uem.es

Resumen

Además de los cambios organizativos de planes de estudio, el Espacio Europeo de Educación Superior (EEES) propone al profesor universitario nuevos retos a la hora de organizar la docencia de sus asignaturas. Y la mayoría de las veces la asunción de cada uno de ellos viene aparejado a una mayor carga de trabajo para el mismo. Este hecho, junto con la normal resistencia al cambio de cualquier profesional, puede poner en peligro el éxito de la empresa.

Desde una perspectiva positiva, el grupo de profesores que intervienen en este artículo decidió utilizar el espíritu de las nuevas prácticas docentes junto con la proliferación de herramientas telemáticas de comunicación y cooperación, para reestructurar la docencia de una asignatura de primero de Informática sin que ello supusiera un mayor esfuerzo por su parte. Así, se pensó en utilizar una serie de herramientas pedagógicas que permitieran al alumno avanzar en su aprendizaje sin aumentar el tiempo de dedicación de su profesor. El resultado ha sido una asignatura bien valorada por lo alumnos y por los profesores, donde el tiempo dedicado a la gestión docente ha disminuido en lugar de haber aumentado. Este artículo pretende contar la experiencia.

.

1. Introducción

La declaración firmada en Bolonia[3] basándose en el informe firmado por Guy Haug y Jette Kirstein [10], supone el punto de partida de un cambio profundo en la Educación Superior.

En la declaración se fijan seis objetivos estratégicos que afectan fundamentalmente a la estructura de las titulaciones, y a la movilidad de pre y post graduados. Como requisito indispensable para ello, hay que unificar la forma de medir la carga lectiva que lleva aparejada una titulación. La propia declaración señala la necesidad de establecer un sistema de crédito común. Esta nueva métrica obliga a reestructurar ciertos aspectos de la ordenación de los estudios universitarios [6].

Es evidente que la nueva métrica y la limitación por curso de créditos ECTS, tienen consecuencias importantes en la tarea del profesor. Ahora, el acento se pone en el trabajo del estudiante y no sólo en las horas de clase que recibe. Lo importante es, por tanto, que "el alumno aprenda" y no que "el profesor enseñe". El profesor pasa de ser un transmisor y evaluador de conocimientos, a ser la persona que "conduce y asiste" el aprendizaje del alumno. Es necesario, por tanto, cambiar la dinámica de las clases, de forma que éstas no sean meros actos de transmisión/recepción de información, convirtiéndolas en actos en los que el alumno tiene un protagonismo activo.

Pero este cambio suele conllevar, además, una mayor dedicación por parte del profesor. El adaptar las clases y las actividades para que se amolden a las características personales del alumno (y potenciar así sus *competencias*), el trasladar al alumno la batuta del ritmo de su aprendizaje sin eliminar materia del temario, obliga al docente a un mayor seguimiento individual del alumno.

En la Escuela de Superior Politécnica de la Universidad Europea de Madrid, hemos considerado que la asignatura *Introducción a la Informática en Red*, por sus especiales características, constituye un banco de pruebas perfecto para trabajar en experiencias innovadoras, que nos permitan reorientar nuestra

práctica docente y adaptarla a las nuevas exigencias emanadas del compromiso de Bolonia. De hecho, ya en el pasado en estas mismas Jornadas, se presentaron algunos de los resultados de dichas experiencias, como la inclusión del juego llamado *cybergymkhana* [2] y la estructura general de la asignatura [11]. En la tabla 1 se presenta de forma sintética la asignatura.

Nombre de la asignatura:	Introducción a la Informática en Red		
Créditos:	6	ECTS:	5.1
		Carácter:	Obligatorio
Duración:			
	Primer curso - Primer parcial (octubre – Enero)		
	Días / Semana		2
	Horas /Dia		2
	Sesiones totales		30

Tabla 1. Cuadro de la asignatura

En esta ocasión, el objetivo fue conseguir un nuevo paso en la dirección de la convergencia europea sin que la inclusión de nuevas herramientas y actividades supusiera una sobrecarga para el docente. Pero antes de exponer dicha iniciativa, es necesario centrar la asignatura y explicar sus objetivos básicos.

Como se explica en [11], los alumnos que acceden al primer curso de una de las titulaciones de informática suelen aportar una visión *extraña* de lo que supone o debe suponer el estudio de una ingeniería. Probablemente el hecho de que la misma herramienta (el ordenador) que usan para sus juegos, sus comunicaciones y su búsqueda de información coincida con la herramienta de la Informática por antonomasia produce una distorsión en el nuevo universitario sobre la profundidad y rigor de la computación como disciplina.

Por otra parte el primer curso de cualquier ingeniería es, probablemente, el que encierra mayor carga de contenidos teóricos en casi todas las titulaciones. Esta circunstancia no está exenta de lógica ya que dichos conocimientos deben ser los pilares sobre los que se sustenten las posteriores prácticas.

Sin embargo esta situación provoca, en primer lugar, que el alumno se encuentre con mucha carga teórica en su primer año, justo cuando más motivado esta para experimentar con ordenadores. Además, el alumno de informática ve cómo los compañeros que cursan otras carreras utilizan el ordenador más que él y por consiguiente *"saben más informática que él"*.

Esto podría ayudar a explicar (junto con otros factores, naturalmente) la elevada tasa de abandono del primer año. Además, el alumno suele suplir, a menudo, esta *carencia* mediante las asignaturas de Libre elección cercanas a la informática de usuario. Pero estas asignaturas deberían cumplir un papel complementario de corte más humanista en la educación del universitario y no un complemento en sus conocimientos técnicos.

La asignatura sobre la que versa este artículo se incluyó en la revisión del plan de estudios del año 2000 [2] para paliar estos efectos, en las tres titulaciones ofertadas en la UEM (ingeniero en Informática, Ingeniero Técnico en Informática de Gestión e Ingeniero Técnico en Informática de Sistemas). Con esta inclusión, el número de créditos impartidos en el laboratorio en este primer curso representa un 24% sobre el total, frente al 14% anterior.

Los objetivos que se persiguen en esta asignatura se pueden resumir en:

1. Familiarizar al alumno con el trabajo en red, introduciendo los conceptos generales sobre redes y muy particularmente sobre Internet
2. Enseñar al alumno a utilizar los servicios de Internet para extraer información
3. Permitir el desarrollo de ciertas competencias personales en el alumno [6]. Dichas competencias representan un subconjunto de las propuestas por la UEM en su informe sobre el particular [7]: Iniciativa, trabajo en equipo, innovación y creatividad, confianza en sí mismo, habilidades comunicativas, responsabilidad, flexibilidad, conciencia de los valores éticos, planificación.

Para cumplir dichos objetivos, la dinámica de la asignatura incluía una nutrida colección de actividades variadas que permitieran al docente mantener un seguimiento constante de la evolución del alumno (evaluación continua). Dicha dinámica podría resumirse en el siguiente esquema:

- Primera semana: presentación de la asignatura; test inicial de conocimientos y

objetivos del alumno: *¿qué sabe ya? ¿qué cree saber? ¿qué quiere saber?*

- Semanas 2-8: cada sesión (100 minutos) se divide en tres periodos. En el primero (25 minutos), se corrige la práctica del día anterior (una práctica diaria). En el segundo (50 minutos), se desarrolla el temario (exposición del profesor, de un alumno, de un invitado externo, etc). En l tercer periodo (25 minutos), se plantea la siguiente práctica, explicando el enunciado y discutiendo su alcance.
- Semana 9: parte del temario expuesto en las semanas precedentes, se deja abierto para la realización de un trabajo en grupo por equipos de tres alumnos. Durante esta undécima semana, representan dichos trabajos y se discuten en clase para que cada grupo escoja el suyo.
- Semanas 9-13: se desarrollan los trabajos
- Semanas 14-15: se presentan en público los trabajos y se evalúan por todos los asistentes (evaluación entre iguales). Se realiza el examen final (preguntas de opción múltiple).

En el siguiente epígrafe de este mismo artículo, se intenta explicar cómo se ha gestionado la información sobre el progreso del alumno a lo largo de la asignatura.

2. Metodología y gestión de la docencia

Como ya se ha explicado en la introducción, la asignatura tiene cuatro horas de clase a la semana, agrupadas en bloques de dos horas (dos horas *docentes* = 100 minutos). La dinámica utilizada para impartir las clases ha consistido en dividir el tiempo de clase en tres bloques:

- Comentario de los trabajos hechos en la clase anterior.
- Desarrollo del temario.
- Trabajo práctico.

Como puede verse, cada día los alumnos deben realizar un trabajo práctico relacionado con los temas que se les ha contado en el bloque de explicación teórica. La entrega de esos trabajos forma parte de la calificación final que obtendrán en la asignatura.

Con este planteamiento de entregas diarias, es fácil darse cuenta de la enorme carga de trabajo de gestión docente que acarrea esta dinámica si no se planifica bien.

Para reducir esa carga de trabajo, los profesores de la asignatura decidimos utilizar algunas herramientas colaborativas en la red, junto con el servicio de ftp y el correo electrónico, que facilitasen algunos de los aspectos de gestión derivados de esta dinámica: blogs, wiki.

Al comienzo de la asignatura se pide a cada alumno que cree un blog en el que deberá ir publicando los resultados de los trabajos que va realizando en el transcurso de la asignatura. Esta práctica descarga al profesor de la tarea de recibir y almacenar los trabajos realizados en cada clase y le permite además hacer un seguimiento de las entregas de forma sencilla.

La otra herramienta utilizada ha sido un wiki de la asignatura. En el wiki se introdujeron los enlaces a los blogs de los alumnos y éstos ponían a su vez información del trabajo realizado en cada una de las clases: Cada día un alumno se encarga de redactar un acta de la clase, en la que se reflejaba los alumnos que habían asistido, un resumen de lo contado en el bloque de teoría, y en que consistía el trabajo mandado para esa clase. Al estar organizado de forma cronológica, el wiki acaba siendo un cuaderno de bitácora de la marcha de la asignatura durante el curso. De esta forma, el profesor puede consultar en cualquier momento quienes asistieron a clase un determinado día o que actividades se realizaron. Además, la confección de las actas de clase por parte de los alumnos, les pone en contacto además de con la propia herramienta con la necesidad de generar información estructurada mediante una plantilla del trabajo realizado, ya que el acta de una clase toma forma del acta de una reunión de trabajo.

Por otra parte, la existencia de la lista de alumnos y los enlaces a sus blog, facilita la comparación de los mismos, ayudando al docente a comparar el trabajo de varios alumnos, incluso si pertenecen a grupos distintos (se construyó un solo wiki para los tres grupos de la asignatura) pudiendo detectar copias o proponer a los alumnos ejemplos de buenas prácticas realizadas por compañeros de otros grupos.

3. Método de evaluación

El método utilizado para evaluar la asignatura ha tenido en cuenta cuatro elementos:

- Trabajos periódicos publicados en el blog
- Trabajo final con presentación oral
- Prueba teórica de opción múltiple
- Participaciones en clase, recogidas en las actas de las mismas (wiki confeccionado por los alumnos)

Gracias al blog el profesor puede realizar un seguimiento de las entregas realizadas a lo largo de la asignatura, controlando qué trabajos se han entregado y en qué fecha de forma mucho más cómoda, por lo que se reduce considerablemente el esfuerzo necesario para evaluar este apartado del sistema de evaluación de la asignatura.

Aparte de este seguimiento de trabajos realizados en clase, los alumnos tienen que preparar un trabajo por grupos de temas relacionados con la asignatura y que tendrán que presentar en clase ante el resto de sus compañeros. Esta actividad pretende reforzar las competencias de trabajo en equipo y capacidad para hablar en público. Se les pide además que todo el proceso de búsqueda de información y las referencias utilizadas para la elaboración del trabajo quedan reflejadas en una entrada en el blog.

Por último se usa una prueba evaluatoria de opción múltiple para evaluar los conocimientos teóricos adquiridos en la asignatura. Esta prueba consiste en una batería de 30 preguntas tipo test sacadas a partir de un conjunto de preguntas elaboradas por los propios alumnos. Antes del examen, cada alumno tuvo que idear entre 15 y 20 preguntas de opción múltiple sobre el temario visto en la asignatura y mandarlas a su profesor por correo electrónico, además de colgarlas en su blog. Se juntaron las preguntas obtenidas de todos los grupos. Ese conjunto de preguntas fue revisado por parte del grupo de profesores, eliminando preguntas inexactas o inadecuadas y se formó un paquete de varios cientos de preguntas (385, para ser exactos) que se colgó en el servicio de ftp de la asignatura una semana antes de la prueba, indicando a los alumnos que el examen estaría formado por preguntas extraídas de ese conjunto.

El objetivo de esta iniciativa era doble. Por un lado, se implicaba al alumno en su propio proceso de evaluación. Pro otra parte, se forzaba al alumno a dar *dos vueltas* al temario, una para redactar las preguntas, respuestas correctas y *distractores* (aumentando además su capacidad de síntesis y su espíritu crítico) y otra después al leer la batería completa de preguntas.

4. Valoraciones

Como se ha venido diciendo en varios sitios a lo largo de este artículo, el objeto fundamental de lo presentado es la utilización de herramientas colaborativas con un objetivo principal, reducir la carga asociada a la gestión y evaluación del aprendizaje del alumno. Así pues, las impresiones y valoraciones de los profesores involucrados en la experiencia dan reflejo del grado de éxito alcanzado.

La valoración por parte de los docentes, comparando con la experiencia de años anteriores ha sido muy positiva. Una de las mayores críticas que se han hecho del proceso de adaptación de las asignaturas al Espacio de Educación Superior ha sido la enorme carga de gestión docente que conlleva realizar todo el seguimiento del aprendizaje del alumno. La sensación del grupo de profesores que han impartido la asignatura de Introducción a la Informática en Red es que el hecho de utilizar estas herramientas colaborativas ha facilitado mucho la labor de seguimiento, reduciendo la sensación de trabajo.

Por ejemplo, en años anteriores los alumnos de esta asignatura, mandaban por correo electrónico las prácticas diarias a su profesor. Esto obligaba al docente a descargar todos los correos de sus alumnos, llegados en distintos momentos del día y entremezclados con otros correos no relacionados con al entrega de la práctica, guardar los documentos adjuntos en un directorio de su disco duro (a menudo renombrando el fichero para evitar colisiones con otros ya existentes) y revisar luego el contenido de dicho directorio abriendo uno a uno los documentos. Esta tarea, dos veces por semana, resultaba tediosa por parte del profesor y una ligera estimación imputaba unos 2 minutos por alumno y práctica dedicado solo a la parte gestora de la entrega (sin entrar en evaluaciones del contenido). El uso del blog como

herramienta de entrega, simplificaba sobremanera el proceso reduciendo la gestión, ya que solo hay que *abrir el blog*. El ahorro se estima en algo más de un minuto por alumno y práctica.

Además, el blog permite una mayor versatilidad en el *feedback* asociado. Además de una respuesta privada a través de correo electrónico, ahora es posible una respuesta pública en forma de comentario en el blog, que puede ser revisado por todos los alumnos y utilizado en clase como elemento enriquecedor.

Otro ejemplo interesante es el control de la asistencia del alumno, asociado ahora al acta redactada por uno de ellos en un servicio de creación de documentos colaborativos: el wiki[13] de la asignatura, en lugar de mediante hojas de firmas responsabilidad del profesor. Además, si un alumno falta un día a clase, la revisión del acta de lo hecho en la misma puede servirle para hacerse una idea de lo explicado en clase y de la práctica mandada, ayudándole por lo tanto a mantener el contacto con la misma.

Pero para ser fieles al espíritu del EEES, es imprescindible conocer la opinión del alumno en cada una de las acciones que se realicen para potenciar su aprendizaje. Para ello, al finalizar la asignatura se les pasó a los alumnos un cuestionario para que expresaran su valoración sobre el método utilizado para impartir la asignatura. La sensación general fue positiva. La mayoría de los alumnos expresaron que el sistema utilizado les resultaba más ameno que el usado en otras asignaturas (debido fundamentalmente a su papel de actores activos) aunque algunos comentaron que el hecho de tener que escribir las prácticas como posts en los blogs les resultaba una tarea monótona cuando ya llevaban varias prácticas. A pesar de esto la calificación global que le dieron fue de notable alto.

La crítica vertida por algunos alumnos ha sido tomada en consideración por el cuadro docente y significará una revisión de las prácticas diarias para hacerlas más versátiles en años venideros.

5. Análisis y conclusiones

Para realizar un estudio de la experiencia y poder sacar conclusiones exportables a otras asignaturas y mejorar la propia en el futuro, se escogieron dos de los tres grupos de la asignatura (descartando el más dispar en número de alumnos) y sobre ellos se realizaron estudios estadísticos sobre diversas variables, así como experimentos diseñados expresamente para evaluar el impacto de las herramientas colaborativas en el aprendizaje. Así quedaron dos grupos de 18 y 19 alumnos matriculados cada uno. En cada uno de los dos grupos, hubo 5 alumnos que si bien aparecían matriculados, nunca acudieron al aula, por lo que en todos los datos presentados en este articulo, dichos alumnos se consideran *inexistentes.* Con esta observación, solo dos alumnos de entre los que comenzaron la asignatura abandonaron o suspendieron o abandonaron la convocatoria.

En la Tabla 2 se recogen los datos sobre la participación de los alumnos en el blog de ambos grupos. Se puede apreciar una alta participación y asistencia (superior en media al 89% de entre los participantes).

		Valor
Nº de post	Moda	12
	Mediana	12
	Media	11,7407
	Desviación Típica	2,5957
		Porcentaje
Asistencia	Moda	100,00%
	Mediana	88,89%
	Media	89,49%
	Desviación Típica	0,1372

Tabla 2. Datos de asistencia y utilización del blog

Un resultado importante, a juicio de los profesores participantes, era descubrir si el alumno percibía como útil o atractivo el uso de estas herramientas como parte de su formación universitaria Para ello se diseñó un experimento oculto para el alumno.

Para realizar el experimento, se decidió utilizar un grupo de la asignatura como grupo control en el que el profesor no fomentaba el uso de las herramientas más allá de la publicación de los trabajos, y otro grupo en el que el profesor fomentó la participación y uso de estas tecnologías en clase.

Como resultado muchos de los alumnos en el grupo del experimento utilizaron los blogs para

escribir sobre temas relacionados con la asignatura pero que no se habían encargado como trabajos de clase o para incluir comentarios en el blog de algún compañero de clase, mientras que en el grupo control ninguno se salió de lo estrictamente mandado como obligatorio. Sin embargo, en la encuesta final donde preguntábamos por la participación en los blog, el grupo del experimento (que había trabajado más con el blog) se mostró menos crítico que el grupo control. Los datos de dicho experimento se recogen en la Figura 1.

Una posible conclusión a esta *proactividad inducida* que no se refleja cuando se deja la iniciativa en el alumno podría ser, a juicio de los profesores participantes, la fuerte influencia que las sugerencias de un profesor de primero de universidad tiene sobre el comportamiento del alumno. Tal vez esta influencia disminuya a medida que va creciendo el alumno. No obstante, de confirmarse dicha influencia, es importante tenerla en cuenta, máxime cuando la convergencia europea hace hincapié en el papel de guía y facilitador del profesor en el proceso de aprendizaje.

En cuanto a la utilización del wiki como herramienta para levantar acta de la actividad de de clase, la experiencia ha sido valorada muy positivamente por los profesores, ya que tener las actas de cada clase escritas por los alumnos y los trabajos publicados en los blogs facilitan las tareas de evaluación y seguimiento al permitir tener toda la información centralizada y siempre disponible, tanto para el profesor como para el alumno, lo que facilita el acceso a la información cuando la necesitemos, y además evita problemas debidos a posibles fallos informáticos o errores en el ordenador del profesor. No obstante, este uso no ha tenido una repercusión apreciable en los alumnos. En las preguntas abiertas del final de la asignatura, no ha habido más que un vago comentario sobre esta herramienta. Queda, pues, como trabajo futuro, diseñar experimentos para extraer conclusiones sobre el uso del wiki.

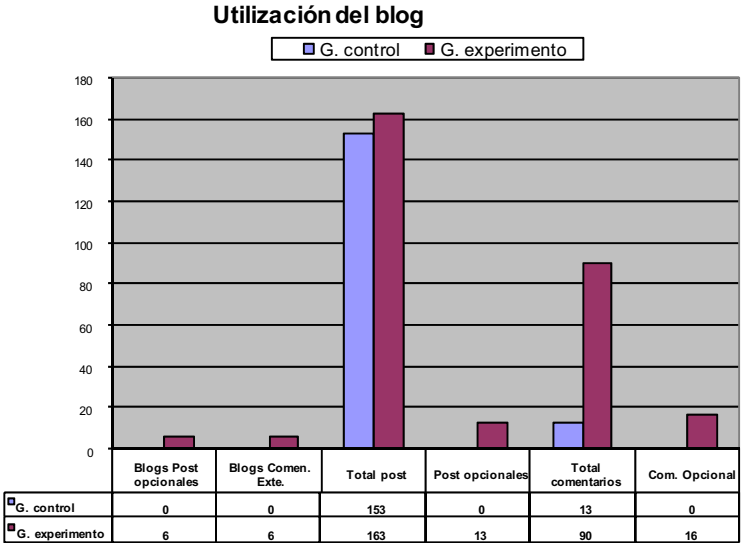


Figura 1: Utilización del blog en los dos grupos

Referencias

[1] http://www.esi.uem.es/asignaturas/IIR.htm *Documento Web de la asignatura*

[2] *http://www.esi.uem.es Documento Web de la Escuela Superior de Informática de la UEM*

[3] The Bologna declaration on the European space for higher education: an explanation. http://europa.eu.int/comm/education/socrates/erasmus/bologna.pdf

[4] Brandt Scott, Constructivism: Teaching for Understanding of the Internet. Communications of the ACM. Octubre 1997.

[5] Bricall , Joseph M. *Universidad 2 mil*. Conferencia de Rectores de las Universidades Españolas(CRUE). 2000.

[6] Docampo Amoedo, Domingo, et all. *El crédito Europeo y el sistema educativo Español*. http://www.crue.org/espaeuro/encuentros/credito-vigo2002_.pdf

[7] Iñigo Alvarez, et als. *Plan de desarrollo de competencias en el alumnado de la Universidad Europea – CEES*, documento interno, 2002.

[8] Pedro José Lara et als. *Nuevas técnicas de aprendizaje: Cybergymkhana*, JENUI 2002

[9] Peter J. Denning and Robert Dunham.The Profession of IT COMMUNICATIONS OF THE ACM November 2001/Vol.44, No.11.

[10] Haugh, G. y Kirstein, J. Trends in Learning Structures in Higher Education. http://www.rektorkollegiet.dk/sider/publikationer/english/edutrends.htm

[11] Juan José Escribano Otero et als. *El proceso de aprendizaje: herramienta para el desarrollo de competencias profesionales en primero de informática*, JENUI 2003

[12] ftp://aurora.esp.uem.es/Asignaturas/ servicio ftp de la ESP de la UEM

[13] http://es.wikipedia.org/wiki/Wiki

Organización curricular y planes de estudio

¿Podemos darle la vuelta a la enseñanza del desarrollo del software?

Josep M. Marco-Simó, Isabel Guitart Hormigo, M. Jesús Marco-Galindo, Àngels Rius Gavidia, M.Elena Rodríguez González
Estudios de Informática, Multimedia y Telecomunicación
Universitat Oberta de Catalunya
Rambla del Poblenou, 156. 08018 Barcelona
{jmarco, iguitarth, mmarcog, mriusg, mrodriguezgo}@uoc.edu

Resumen

Se describe el trabajo realizado por un grupo pluridisciplinar de once profesores del área de desarrollo del software (programación, bases de datos, ingeniería del software y sistemas de información) de la Universitat Oberta de Catalunya (UOC) en relación a la organización de la enseñanza de esta área. El trabajo ha estudiado las posibilidades de reordenar e incorporar nuevos contenidos, además de detectar solapamientos y contenidos que se alejan de las necesidades actuales de formación a nivel universitario.

En este artículo se expone por un lado el proceso de reflexión y por otro el resultado obtenido en forma de mapa del área y la justificación, tanto de los cambios propuestos, como de las decisiones de mantenimiento de algunos aspectos más clásicos.

Las conclusiones y las propuestas presentadas más que ser definitivas, pretenden estimular el desarrollo de reflexiones similares en el contexto de cambio que implica el EEES.

1. Motivación

Tradicionalmente, la docencia de los aspectos de desarrollo del software (DS) se ha impartido siguiendo una visión ascendente, es decir, empezando por la programación (P) hasta culminar con las tareas más complejas de la ingeniería del software (IS). El origen de este enfoque posiblemente hay que buscarlo en el propio desarrollo histórico del software. Sin embargo, en la actualidad este enfoque ascendente es precisamente inverso al proceso del DS y de los ciclos de vida indiscutiblemente aceptados [5].

En otras áreas de la informática, como por ejemplo la de redes de computadores, la docencia también había seguido este enfoque ascendente, a pesar de que el interés profesional de un estudiante fuera justo el contrario: mucho más centrado en la utilización de las capas OSI medias y altas que en el movimiento de los bits por la red. Hoy por hoy, ya existen propuestas para el área de redes de computadores [3] que abandonan el enfoque tradicional planteando una nueva visión descendente, visión que se está implantando en muchos currículos universitarios y que supone, en ocasiones, la pérdida de relevancia de algunos contenidos y/o la aparición de otros nuevos.

Por otro lado, parece claro que las discusiones actuales sobre: a) hasta qué punto un ingeniero en informática debe formarse en programación [4]; b) cómo adaptar los currículos de las diferentes áreas de la informática al EEES; c) cuáles son las competencias diferenciales de un ingeniero versus otros profesionales del sector; favorecían una reflexión acerca de la docencia en el área del DS, intentando descubrir aspectos a descartar y nuevas necesidades de formación, así como plantear nuevos itinerarios formativos de adquisición de las competencias dentro del currículo.

Así pues, con el objetivo final de plantear la docencia del DS desde lo general a lo particular, esto es, de manera descendente, un grupo de profesores de la UOC ha realizado a lo largo de 2006 una serie de sesiones de trabajo y discusión. El proceso de reflexión así como los resultados obtenidos se exponen en este artículo. Así, la sección 2 explica la situación actual de la docencia en el ámbito del DS en la UOC; la sección 3 resume el proceso de reflexión seguido y propone un nuevo mapa del área. Y la sección 4 expone las conclusiones y el trabajo futuro.

2. Punto de partida

En general, y también en el caso de la UOC como puede verse en la Figura 1, la docencia en el

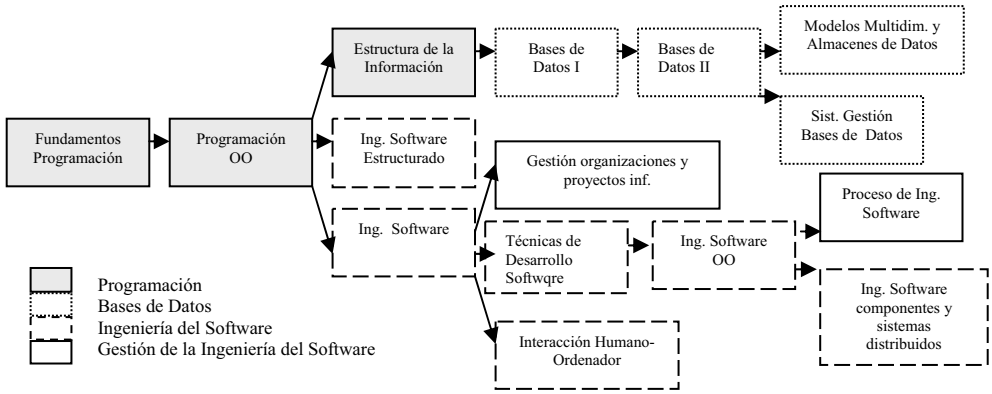


Figura 1. Organización curricular actual de DS en la UOC

ámbito del DS agrupa las competencias en dos grandes grupos: por un lado, las relativas a P, y por otro las que se sitúan en la IS, entendida en un sentido amplio (es decir, incluyendo los aspectos de bases de datos (BD) y de gestión del proceso de desarrollo del software (G)).

Los objetivos *propios* de aprendizaje de P se podrían sintetizar en los siguientes:

- PO1. Conocer y dominar las herramientas básicas de programación que permitan la concepción de algoritmos correctos.
- PO2. Saber definir las estructuras de datos necesarias para dichos algoritmos.
- PO3. Conocer y saber llevar a cabo el proceso de codificación-compilación-prueba de programas.
- PO4. Tener conocimiento de los paradigmas de P (estructurado, orientado a objetos, distribuido) para escoger el más adecuado, dado un entorno de programación y un problema concreto a resolver.

Por su parte, los objetivos *propios* de la IS podrían resumirse en:

- ISO1. Saber analizar requisitos de cualquier sistema de información.
- ISO2. Saber diseñar soluciones, tanto desde el punto de vista del proceso (IS), como desde el punto de vista de los datos (BD).
- ISO3. Conocer los paradigmas relacionados, y saber adaptarlos a cada caso concreto.
- ISO4. Ser capaz de llevar a cabo la planificación y control de proyectos (G).

Asimismo, el estudiante de DS debe desarrollar una serie de *habilidades genéricas* a P y a IS como son la capacidad de abstracción, la capacidad lógica y de análisis, y la capacidad para seguir métodos y aplicar técnicas generales a casos concretos. Por supuesto, estas habilidades, aún siendo comunes a P e IS, se aplican a problemas de diferente magnitud y ámbito. También puede observarse que la adquisición de estas capacidades genéricases objetivo de asignaturas de otras áreas del currículo (lógica, matemática discreta y/o informática teórica) que posiblemente el estudiante no haya cursado completamente cuando afronta las asignaturas de DS.

Por otro lado, intrínsecos al área de DS, los paradigmas de desarrollo (objetivos PO4 y ISO3), también son comunes a las dos áreas y, consecuentemente, en ambas se realizan esfuerzos para introducirlos. Ocurre, sin embargo, que estos esfuerzos no están siempre alineados.

Otra observación necesaria es que en los primeros cursos de P el estudiante además de aprender los conocimientos básicos, debe trabajar simultáneamente aspectos relativamente alejados: por un lado los objetivos más metodológicos (PO1, PO2 y PO4), y por otro el objetivo de habilidad práctica (PO3).

Todas estas consideraciones, junto con una organización curricular que prima la introducción rápida de PO3, y la distribución de objetivos que requieren maduración y practica como son PO1, PO2 y PO4 en asignaturas de alrededor de trece semanas útiles de docencia, pueden explicar en parte la dificultad para lograr todos los objetivos

de aprendizaje así como el bajo rendimiento de estas materias en los primeros cursos.

Tras constatar esta visión, las primeras preguntas que se plantearon fueron las siguientes:

- ¿Podríamos darle la vuelta al aprendizaje del DS empezando por la parte común (habilidades de análisis, diseño y paradigmas) y posponiendo la introducción de PO3?
- Esta inversión del orden de aprendizaje, ¿evitaría repeticiones y solapamientos de objetivos y contenidos entre distintas asignaturas, y, por tanto, permitiría disponer de más tiempo para profundizar en el aprendizaje de ciertas competencias?
- ¿Existen otras asignaturas en los currículos que, rediseñadas, puedan jugar un papel relevante en este replanteamiento?

3. El proceso de reflexión

Como esquema de trabajo para iniciar una reflexión más profunda que permitiera responder a las cuestiones anteriores (y a otras que pudieran aparecer posteriormente), se planificaron las siguientes fases que, mediante un proceso de aproximaciones sucesivas, nos debían llevar a nuevas propuestas y a posibles resultados:

- F1. Buscar experiencias innovadoras de ingeniería curricular en relación al DS.
- F2. Profundizar más allá de la visión tradicional del área de DS, buscando un cierto consenso sobre las debilidades y oportunidades de la docencia actual.
- F3. Plantear cambios y reorganizar los objetivos, con el fin de esbozar una posible ordenación curricular.
- F4. Establecer un mapa de contenido que facilite su inclusión en la organización que resulte de la aplicación del EEES.

El equipo de profesores implicados incluía personas de los cuatro ámbitos indicados: dos de P, dos de BD, tres de IS y dos de G, así como dos más de asignaturas de síntesis. El trabajo se realizó en seis sesiones conjuntas, cada una de tres horas, para las que previamente cada profesor, individualmente o en grupo, había realizado un trabajo/análisis previo según las directrices que se marcaban. Durante las sesiones, se ponían en común las reflexiones de cada profesor o pareja (mediante un sistema de tarjetas *colgables* en una pared o pizarra) y se sometían a discusión con el resto de profesores. Aunque, evidentemente, las discusiones detalladas sobre objetivos y contenidos a menudo no podían producirse entre profesores de las distintas subáreas, sí que permitían emerger ciertas ideas preconcebidas, a la vez que fluían otras más novedosas, fruto del "*brainstorming* controlado" que surgía de la interacción entre profesores de diferentes especialidades y sensibilidades.

A continuación se resumen los aspectos abordados en cada fase. No se entra, por motivos de espacio, ni en el detalle de cada sesión, ni en su dinámica, a veces diferente de la prevista.

3.1. F1: Buscando otras experiencias

Consultada la organización curricular de universidades que nos son cercanas (UPC, UPM, UPV, UCIII), es importante destacar que no encontramos experiencias reales de implantación que abordaran este tema desde una perspectiva similar a la que nos planteábamos.

Como era de esperar, sí que se encontraron algunas experiencias expuestas en anteriores ediciones de las JENUI que planteaban, bien desde otra perspectiva, bien parcialmente, temáticas parecidas [1].

En el contexto internacional la fuente más clara es el currículo sobre ingeniería del software propuesto por la ACM [2]. En una de sus estructuras curriculares propone empezar por IS. Sin embargo, el análisis de su propuesta permite constatar que dentro de IS incluye P, así que, de hecho, realmente empieza por P.

3.2. F2: Más allá de la visión clásica

Para intentar superar los tópicos más habituales del DS se intentó que dos cuestiones nos acompañaran durante toda la reflexión, a pesar de que, entre nosotros, tampoco existía un consenso claro sobre ellas.

- ¿Es necesario retrasar la codificación? ¿Es ventajoso o contraproducente "codificar-compilar-probar" desde el primer día? ¿Entenderían los estudiantes no hacerlo así?
- ¿Es posible empezar directamente introduciendo en genérico el análisis de

problemas y el diseño de soluciones sin conocer ninguna de las *palabras nucleares clave* de P e IS (como *variable, parámetro, tipo, procedimiento, compilar, depurar, entorno, descomponer, llamar, abstraer…*)?

Empezamos, pues, revisando conjuntamente y de forma plenaria todos los contenidos que impartíamos en el área del DS, con el fin de confirmar o no la visión más clásica y, sobre todo, de descubrir otros problemas de los que no éramos conscientes. Y surgieron los siguientes aspectos:

- La aparición explícita de todo el ciclo de vida de IS no aparece hasta la primera asignatura de IS, esto es, hasta claramente avanzada la carrera.
- El estudio sucesivo de UML, desde diferentes perspectivas o usos. Esto es, el riesgo de repetición de contenidos y, de nuevo, la falta de visión unitaria.
- La aparición reiterada de los conceptos de eficiencia y calidad en P, IS, BD, G.
- La repetición del modelado de datos tanto en BD como en IS, con el problema añadido de explicarse con métodos diferentes.
- La repetición de algunos contenidos en BD e IS, debido a la historia de desarrollo y puesta en marcha de algunas asignaturas.
- La falta de conexión entre IS y P. En IS se llega hasta el diseño que es el punto de partida de P, pero no se establece el enlace entre las dos áreas de manara natural.
- El exceso de contenido con poco tiempo para ser asumido, sobre todo en P. No se trata tanto de recortar el contenido, como de disponer de más tiempo para su asimilación.
- La aparición de temas de implementación muy *tecnológicos* (tanto en BD como en P) como, por ejemplo, aspectos internos de SGBD o de compiladores.
- La no inclusión de conceptos sobre diseño y programación distribuida hasta prácticamente las asignaturas de síntesis final.
- La incorporación de interacción de persona-ordenador como un aspecto colateral, no integrado en todo el ciclo de vida.
- La confirmación de la necesidad de disponer de asignaturas de síntesis que cubran todo el ciclo de vida del DS.

3.3. F3: Planteando los cambios

Teniendo en cuenta todos los aspectos surgidos en las dos fases previas, se planteó, para los cuatro ámbitos (P, IS, BD, G), una discusión liderada por los profesores especialistas de cada uno y confrontada con una actitud crítica del resto de profesores. El objetivo era proponer y debatir los cambios de cada subárea.

Para IS:

- El principal cambio que se plantea, consiste en limitar el estudio del paradigma estructurado, con el fin de maximizar la dedicación al paradigma orientado a objetos. Aunque este mayor énfasis en las metodologías de orientación a objetos ya se da en nuestra organización curricular actual, el remanente histórico de dedicación a las metodologías estructuradas aún es alto.
- Se propone un primer bloque curricular que tenga por objetivo la introducción de los conceptos de la IS y la importancia de garantizar la *calidad* en todo el proceso de DS, siempre *centrado en el usuario* – requisitos e interacción-, para continuar después con el análisis del sistema basado en UML.
- En un segundo bloque se trabajarían los objetivos de diseño para continuar con la IS distribuida (centrada en aspectos de diseño, pues los de análisis ya se han estudiado y sólo sería necesaria su revisión).

Para BD:

- En esta área la propuesta de cambio consiste más en reorganizar objetivos que en incluir nuevos.
- Los objetivos fundamentales incluyen el estudio del modelo de datos relacional y el álgebra relacional, los SGBD, aspectos de concurrencia desde un punto de vista de diseño, SQL (interactivo y programado), creación de BD partiendo de UML, bases de datos distribuidas, desnormalización y almacenes de datos. Como objetivos fundamentales de nueva incorporación se incluyen datos semiestructurados y XML.

- Como objetivos de aprendizaje optativos se proponen conocer los aspectos relacionados con el control de concurrencia y recuperación en BD, optimización y *tunning*. Como objetivos de nueva incorporación, conocer alternativas a las BD relacionales.

Para P:

- Las habilidades en programación, entendida como la técnica de codificación-ejecución-prueba-depuración que se implementa con diferentes lenguajes, deberían tener su propio lugar. Este lugar, en ocasiones llamado *tecnología de la programación*, debería ser independiente del otro lugar propio de P, el de la *algorítmica*. De esta forma, se podría intentar desactivar la discusión clásica sobre qué lenguajes utilizar en cada asignatura (Java, C, Pascal, ensamblador).
- En este lugar reservado a la técnica de la programación, se debería insistir en la búsqueda de la calidad del software, a partir de la importancia de la representatividad de las pruebas a realizar, la depuración y las cuestiones de eficiencia.
- Para facilitar la comprensión de estructuras de datos complejas, se confirma la necesidad de reintroducir conceptos de recursividad.
- La aceptación de esta área de *tecnología de la programación* separada de los contenidos sobre *algorítmica*, facilitará la reformulación y reorganización de las competencias objetivo de P
- Entre las ventajas de empezar por IS en lugar de por P, están las de incluir en el currículo una sola vez el estudio de las características de la OO, para luego ir desarrollándolas en amplitud y profundidad según las necesidades particulares de P y de IS. Adicionalmente, otra ventaja sería que el estudiante ya comprendería un análisis UML, en el momento en que tuviera que afrontar el aprendizaje del diseño e implementación de la solución.
- Por otra parte, parecería natural tender hacia una dedicación de dos cursos académicos completos para asimilar los conceptos de algorítmica, y de tecnología de la programación.

Para G:

- En la organización curricular actual los aspectos de calidad ya se abordan aunque en las asignaturas finales. Por eso sería necesario asegurar que se introducen antes, una vez conocido el ciclo de vida así como generalizarlos.
- Tradicionalmente, los contenidos propios de esta subárea se hayan mezclados con otros más cercanos a la dirección de sistemas y tecnologías de la información (STI). Se deben considerar temas relacionados, aunque no incluidos propiamente en el desarrollo del software. La razón de que tiendan a mezclarse está en que a menudo en el entorno profesional, un jefe de de departamento de STI acostumbra a haber sido, o a ser a la vez, responsable de proyectos de desarrollo de software.

3.4. F4: Dibujando un nuevo mapa

Del debate de F3 sobre las cuatro subáreas, parecía que se podía esbozar alguna respuesta a las preguntas inicialmente planteadas (ver final sección 2).

Así, si aceptamos la necesidad de disponer de más tiempo para aprender las habilidades expuestas para P, posponer la enseñanza de P hasta haber introducido IS puede afectar a la planificación curricular de otras áreas de conocimiento. En consecuencia, además de las dudas que plantea iniciar la docencia con una visión descendente, parece claro que hay otros condicionantes, como la estructura curricular, las expectativas de los estudiantes, y la necesidad de compartir las competencias de programación con otras asignaturas (p.e. redes o sistemas operativos). Posiblemente, pues, la reordenación debería pasar por realizar en paralelo la docencia de IS y P, en lugar de invertirla.

Adicionalmente, sin embargo, sí que podría ser posible dar una visión global de todo el DS, introduciendo los conceptos fundamentales y creando un marco de referencia general en la primera asignatura de DS. Este marco de referencia sería el hilo conductor al que el estudiante acudiría durante la profundización en las cuatro subáreas y en sus materias asociadas,

ayudándole a tener siempre presente esta visión descendente del proceso de DS.

Para dibujar el mapa final, se decidió aplicar las siguientes simplificaciones:

- La idea inicial era hacer un mapa genérico de competencias del área con independencia de en qué oferta formativa se incluyeran (ingeniería, ingeniería técnica, grado, máster). Sin embargo, resultaba muy difícil no tener en cuenta los objetivos/competencias diferenciales de cada oferta formativa. Así pues se optó por plantear un mapa para un hipotético currículo de *Grado en Informática*.
- Otra simplificación fue describir el mapa en forma de contenidos como se había venido haciendo hasta ahora y no de competencias. La experiencia en la descripción por competencias era desigual entre el grupo de profesores y desviaba la atención del objetivo principal del trabajo. Más adelante ya se plantearía una descripción por competencias en consonancia con las directrices del EEES.
- Calcularlo en créditos tradicionales, no ECTS. La experiencia en el cálculo de ECTS era también demasiado heterogénea en el grupo de profesores y podía dificultar la comprensión del resultado global.
- Establecer unidades de aprendizaje mínimas (a fin de no pensar en asignaturas) que facilitaran su ensamblaje posterior en la organización temporal que finalmente resultara del grado EEES. Estas unidades las definimos como "el conjunto máximo de contenidos mínimos (indivisibles) desde el punto de vista docente en un periodo lectivo y para una titulación de Grado".

Finalmente, el mapa resultante se presenta en la Figura 2 (unidades mínimas con indicación de su peso en créditos convencionales, así como su interrelación) y en la Figura 3 (contenidos asociados a cada unidad de aprendizaje mínima).

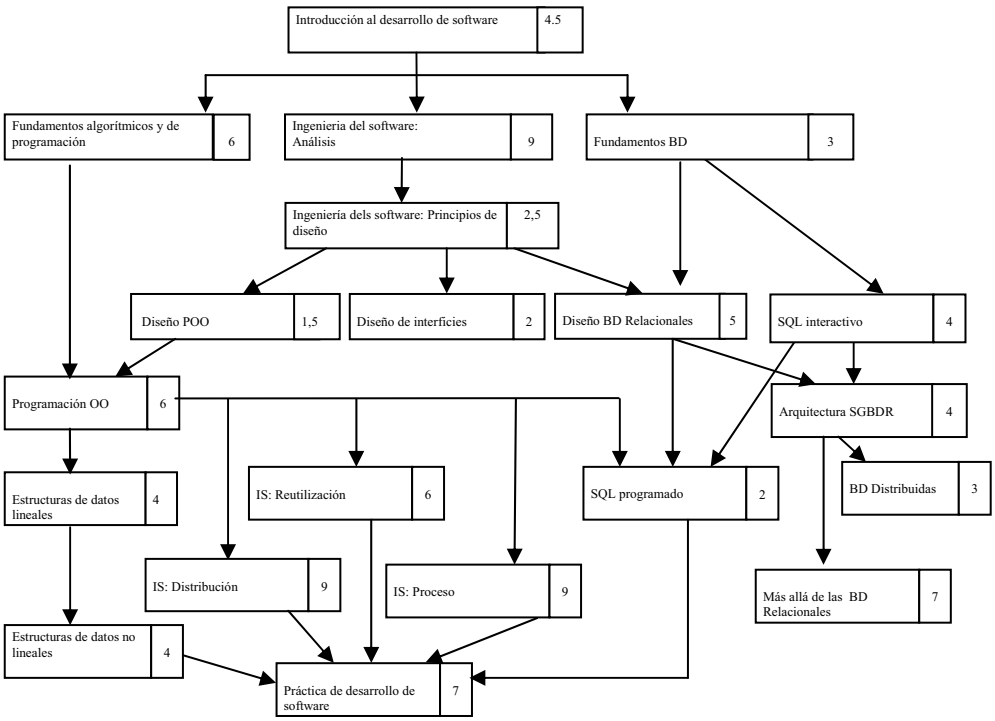


Figura 2. Mapa de contenidos propuesto

Introducción al desarrollo del software	- Etapas del ciclo de vida - Metodologías de desarrollo de software (estructurada, orientada a objetos, distribuida) - Algorítmica básica (control de flujo, tipos elementales, entrada/salida) - Tecnología de la programación básica (Codificación, depuración, test en C y pseudocódigo)
Fundamentos algorítmicos y de programación	- Estructuras de datos: Tablas y tuplas - Secuencias: Recorrido y búsqueda - E/S: Ficheros - Eficiencia temporal y espacial
Ingeniería del software: análisis	- Características OO - Metodología (Unified Process) - Captura de requisitos y análisis (UML y OCL) - Testing funcional - Análisis de la persistencia (Diseño conceptual UML. Modelo de datos)
Ingeniería del software: principios de diseño	- Diseño de casos de uso - Diseño arquitectónico - UML asociado
Fundamentos de BD	- Principios de la gestión de la información y evolución histórica - Funcionalidades de un sistema gestor de BD - Modelo de datos relacional y álgebra relacional.
Diseño POO	- Consolidación diseño (adaptación al lenguaje de programación)
Programación OO	- Justificación de la programación OO - Codificación POO (traducción del diseño a código OO). Herramientas Java. - Pruebas unitarias y de integración - Interfaces
Diseño de Interfaces	- Diseño interfaz de usuario en el contexto de la ingeniería del software
Diseño BD relacionales	- Diseño lógico - Normalización y desnormalización - Diseño físico (índices, tunning, esquema interno)
SQL interactivo	- Creación BD y tablas, consultas y cambios de la BD (inserción, borrado, modificación) - Vistas, disparadores, procedimientos almacenados, transacciones, autorizaciones y roles
SQL programado	- JDBC y SQL-J
Estructuras de datos lineales	- Introducción a los TAD - Revisión de eficiencia - Secuencias: pilas, colas y listas - Colas prioritarias - Librería de TAD en Java
Estructuras de datos no lineales	- Recursividad - Árboles, funciones y conjuntos, relaciones y grafos - Diseño de TADs complejos - Librería de TADs de Java
Arquitectura sistemas gestores de BD relacionales	- Gestión de vistas - Gestión de la seguridad - Optimización de consultas - Control de concurrencia y recuperación BD
Más allá de las BD relacionales	- BD OO y BD Object-relational - Almacenes de datos (multidimensionales, FIC, OLAP) - Datos semiestructurados y XML
BD distribuidas	- Acceso a BD distribuidas - Arquitectura SGBD distribuidas
IS: Distribución	- Especificación (distribución en general: RM-ODP) - Arquitecturas distribuidas - Ingeniería de componentes (UML asociado) - Diseño distribuido OO (UML asociado) - Tecnologías de implementación distribuidas
IS: Reutilización	- Principios y objetivos de la reutilización - Patrones y componentes - Otras técnicas de reutilización
IS: Proceso	- Calidad - Gestión de la configuración - Mantenimiento - Métricas
Práctica de desarrollo del software	- Aplicación práctica global de desarrollo de software

Figura 3. Relación de contenidos

4. Conclusiones

Para finalizar, las conclusiones más relevantes sobre la organización de los contenidos del DS podrían resumirse en:

- La necesidad de dar una visión global del proceso descendente del DS, se puede cubrir mediante la creación de un contenido inicial marco que sirva de referencia al estudiante.
- Sin embargo, diferentes motivos desaconsejan la inversión drástica del orden de docencia (empezar por IS y seguir por P).
- Ahora bien, aunque no se produzca esta inversión en el orden, sí que puede realizarse en paralelo la introducción tanto de IS como de P. Esto permite reforzar e insistir en los aspectos que son comunes y fundamentales, minimizar las repeticiones de contenido y facilitar la incorporación como un continuo de aspectos hasta ahora tratados no integradamente como la calidad del software o el diseño centrado en el usuario.
- Es necesario reducir el tiempo dedicado a los temas que ya se consideran no útiles para el profesional actual (metodologías de IS estructuradas, aspectos excesivamente teóricos de algorítmica, etc.).

En cuanto al detalle concreto de contenidos finales, no nos es posible justificarlo pormenorizadamente por problemas de espacio, por lo que remitimos a las figuras 2 y 3 que representan el mapa resultante.

En cuanto al proceso seguido para esta experiencia, las mayores dificultades que nos hemos encontrados podríamos resumirlas en:

- La ausencia de experiencias similares obliga a superar reticencias al cambio en la acción docente.
- El conocimiento de las dificultades del estudiante para afrontar temas abstractos no ayuda a ser más atrevidos en el nuevo diseño curricular.
- Otras asignaturas deben contribuir a aportar las bases para la enseñanza del DS. Esta discusión, que implica reconsiderar contenidos de otras áreas de conocimiento, requiere una estrategia compleja para que sea operativa.

En cuanto al trabajo de futuro pendiente es evidente la necesidad de realizar la conversión del mapa de contenidos en uno de competencias, así como transformar los créditos convencionales a créditos ECTS. En la actualidad, la UOC ha desarrollado unas indicaciones y herramientas propias que permitirán afrontar estas conversiones una vez conocidos los marcos de referencia EEES.

Por supuesto, no pretendemos haber planteado una propuesta definitiva, sino sólo volver a promover reflexiones similares entre los foros de profesorado. Y las revisiones que implica el EEES son otra ocasión para reconsiderar el papel de los currícula en este fundamental ámbito del DS.

Agradecimientos

Los autores quieren agradecer las indicaciones de los revisores de las JENUI que les han facilitado mejorar y releer este trabajo desde otra perspectiva. Asimismo quieren mencionar a los profesores de la UOC Joan Arnedo, Jordi Cabot, Santi Caballé, Daniel Riera, Atanasi Daradoumis y Enric Mor, quienes formaban parte, activa y fundamental, del grupo de trabajo implicado en la experiencia aquí descrita.

Referencias

[1] Cachero, C.; López, O. & Durá, M.J. *Del Diseño a la Implementación del Software: Una metodología de cohesión.* En *X Jornadas de Enseñanza Universitaria de la Informática. Actas del congreso.* Thomson, 2004.

[2] IEEE & ACM, *Software Engineering 2004 Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering A Volume of the Computing Curricula*, 2004.

[3] Kurose, J.F & Ross, K.W. *Computer Networking: A Top-Down Approach Featuring the Internet, 2/e* Addison Wesley, 2005.

[4] McBride, N., *The death of computing (Member view)* British Computer Society, 2007.

[5] Sommerville, I. *Software Engineering* Harlow: Addison-Wesley, 2005.

La influencia de la imagen social de la profesión en las titulaciones universitarias de informática

Luis Fernández[1]
[1]Dpto. de Sistemas Informáticos
Universidad Europea de Madrid
C/Tajo s/n, 28670 Villaviciosa de Odón (Madrid)
luis.fernandez@ uem.es

Resumen

A diferencia de otras ingenierías, la ingeniería informática tiene una imagen social demasiado condicionada por el concepto difuso de "informático". Esta circunstancia se inscribe en una situación tanto de falta de vocaciones expresadas en la matrícula universitaria como de peculiaridades de las expectativas laborales y de regulación de la profesión. También se refleja en los perfiles de egresado que se incorporan a los planes de estudio de las titulaciones de informática. En este trabajo se presenta un análisis de la imagen social de la profesión así como de algunas implicaciones en el desarrollo y definición de la enseñanza universitaria de las titulaciones relacionadas con la ingeniería informática.

1. Introducción

Uno de los objetivos de la adaptación al Espacio Europeo de Educación Superior (EEES) es la orientación de las titulaciones hacia la empleabilidad de los egresados. De hecho, la propia declaración de Bolonia [1] establece como uno de los objetivos fundamentales de la convergencia europea "Adoptar un sistema de títulos comprensibles y comparables a fin de promover la capacidad de inserción laboral de los ciudadanos europeos."En esta línea, la declaración de Bergen [2] indica que:
"El Espacio Europeo de Educación Superior se estructura en tres ciclos, donde cada nivel tiene simultáneamente las funciones de preparar al estudiante para el mercado laboral, de proporcionarle mayores competencias y de formarle para una ciudadanía activa. El marco general de las cualificaciones, el conjunto de directrices y estándares europeos comunes para el aseguramiento de la calidad y el reconocimiento de los títulos y períodos de estudios son también características clave de la estructura del EEES."

El problema en las titulaciones de informática es que existen dos factores que están influyendo en su evolución y, por tanto, en la adaptación al EEES. Por una parte, existe un descenso especialmente acusado en la matriculación en estas titulaciones no sólo en España sino en otros países de Europa y del mundo. Por otra, la profesión relacionada con la informática tiene serios problemas de definición e imagen social. Lógicamente, estos problemas van a tener una clara influencia en la adaptación de las enseñanzas al EEES.

En los siguientes apartados vamos a presentar los problemas relacionados con la matriculación y la imagen profesional para finalmente aportar algunas guías obtenidas de los informes RENTIC, elaborados bajo la dirección del autor desde 1998.

2. Matriculación universitaria en informática

Una de las preocupaciones de los responsables de las titulaciones de informática en España ha sido el descenso de la matriculación de estudiantes en las mismas. Por ejemplo, en el ámbito de la CODDI (Conferencia de Decanos y Directores de Informática) se ha comentado en diversas ocasiones esta situación. Incluso, durante un tiempo, algunos centros no querían reconocer dicho descenso pues creían que no era generalizado y les parecía que si les ocurría a ellos era porque tenían problemas especiales que "ahuyentaban" a los alumnos[1]. En un momento dado, a la vista de los datos, la CODDI en conjunto asumió que el descenso de matriculaciones era un fenómeno generalizado.

[1] Aunque ya los medios de comunicación comentaban el problema [3] con el expresivo título: "Estudiantes de informática: ¿especie en extinción?".

Incluso en la edición de JENUI de 2003 se comentaba la posible situación de descenso y algunos rápidamente proclamaban que eso les pasaba a otros y no a ellos. Analizaremos a continuación diversos elementos de este fenómeno para entenderlo mejor.

El primer elemento que influye es de tipo general: por la evolución demográfica heredada de las bajas tasas de natalidad experimentadas en los ochenta y noventa, existe . Las proyecciones demográficas según los datos de de INEBASE (alimentados por el MEC)[2] marcan una tendencia imparable hasta 2014 de descenso en la población de 18 años. Por tanto, todas las titulaciones y universidades se verán afectadas. La población inmigrante no es probable que influya ya que su contribución demográfica se produce actualmente en la educación primaria (como ya se está notando) y tardará en incorporarse a la universitaria (aparte de que su situación social puede que no favorezca las condiciones para llegar a la educación superior). Además, Hasta 2005, desde 2002, el descenso es del 7,23%, hasta 2010 del 12,13% y hasta 2015 del 14,93%.

El segundo elemento de influencia es que, a efectos de matriculación, este descenso no es homogéneo en todas las ramas de secundaria. El número de alumnos de educación secundaria que accede a la universidad por la rama científico-técnica es cada vez menor. Según los datos de INEBASE (alimentados por el MEC), mientras el descenso general de estudiantes que acceden a la universidad es de un 14,87% desde el curso 2000-01 hasta el 2004-05, en la rama científica-técnica es del 28,6%. Este es un fenómeno general en España y en Europa donde la anterior comisaria de Educación Viviane Reding ya resaltó esta tendencia e impulsó algunas iniciativas para paliar la falta de vocación técnica y científica de los jóvenes. En España, la rama de ciencias sociales la matriculación ha descendido en el mismo período sólo un 9,05% y las de artes ha crecido un 40,7%. Lo único que se constata es que el porcentaje general de aprobados en las pruebas de acceso a la universidad ha crecido desde el 80,8% de 2000-01 al 84,28% de 2004-05 (la distribución en las distintas ramas es equivalente). No sabemos si esto se ha producido por una mejor preparación de los estudiantes o por un nivel de exigencia menor en las pruebas.

El tercer elemento que influye en las enseñanzas técnicas en general es su tradicional bajo porcentaje de mujeres en el alumnado y que puede influir en los anteriores números. Podría pensarse que, con la evolución de las costumbres y mentalidad en la Sociedad, la tendencia debería ser al incremento en el porcentaje de mujeres matriculadas en ingeniería y ciencia. También que aumentaría en general el porcentaje de mujeres en la universidad. Sin embargo, los números no indican nada de esto. El porcentaje de matriculación en secundaria desciende levemente en general (del 59,2% al 58,7%) pero en cuanto a aprobados en el acceso a la universidad el porcentaje se mantiene estable pero el descenso experimentado en los totales desde 2000-01 a 2004-05 es del 15,6%. De nuevo, las ciencias sociales descienden menos y las artes crecen. En la rama científico-técnico, el descenso de matriculadas en acceso a la universidad desde 2000-01 es del 38,3% y el porcentaje de mujeres desciende del 27,7% al 26%.

Lógicamente, el resultado de esta situación en educación secundaria incide en la matriculación en enseñanzas técnicas. Así, con los datos del MEC en INEBASE, podemos ver la variación entre el curso 2000-01 y el 2005-06. En su conjunto, las enseñanzas técnicas superiores (Arquitectura e Ingenierías) han perdido un 1,71% de alumnado, si bien las universidades públicas han perdido un 3,7% mientras que las privadas han ganado un 22,3%. En el conjunto de las técnicas, el descenso es de un 5,85% con un 10,78% para las públicas y un incremento del 54,5% para las privadas. En conjunto, los descensos dada la caída de natalidad no es tan grave tomando estos datos de total de matriculados porque aún quedan por concluir sus estudios una masa importante de estudiantes a pesar de que hay menos alumnos de nuevo ingreso. Así, el descenso en las titulaciones superiores desde el curso 2000-01 al 2004-05 en conjunto es sólo del 0,05% (con descenso en las públicas e incremento en las privadas); en el caso de las titulaciones técnicas, el descenso conjunto es del 8,84% con descensos tanto en públicas como en privadas.

[2] http://www.ine.es/inebase/index.html

Sin embargo, estos datos pueden ser algo engañosos ya que la evolución de las distintas carreras no es ni mucho menos igual. Lo primero que hay que resaltar (quizás consecuencia del éxito del sector inmobiliario y de construcción) es que las titulaciones de arquitectura e ingeniería civil o de caminos no han descendido sino aumentado ingreso y matrícula con amplitud. Entre las ingenierías con mayor peso de alumnos, los resultados son distintos como se ve en la Tabla 1.

Titulación	% matrícula vs 00-01	% ingreso vs 00-01
Telecomunicación Global	+15,43%	-11,2%
Superior	-5,99%	-12,43%
Técnica	+18,08%	-10,19%
Industrial Global	+6,8%	+2,4%
Superior	+ 10,03%	-1,32%
Técnica	-4,60%	-2,90%
Informática Global	-5,64%	-10%
Superior	-14,08%	+20,63%
Gestión	- 1,35%	-18,73%
Sistemas	+ 0,62%	-18,52%

Tabla 1. Descenso en matrícula total y nuevo ingreso en ingenierías en el curso 04/05 respecto del curso 00/01

Frente a los datos de la Tabla 1 se resalta que:

- Los grupos de Arquitectura y Caminos suben respectivamente un 4,78% y un 4,23% en matrícula; 15,4% y 23,8% en ingreso.
- El avance de datos de matrícula del curso 05-06 limitaría la caída del grupo de informática al -1,5%. Los demás grupos mejoran excepto telecomunicación que se sitúa en +11,42%.
- En cuanto a las mujeres matriculadas, ha aumentado su número en informática en un 3,55% y en telecomunicación un 15,46% mientras que descendió en el grupo de industrial (-4,69%) y en caminos (-0,73%).

En la Tabla 2, se indican los porcentajes de mujeres sobre el total de matriculados donde pe puede ver que informática e industriales son las titulaciones que menor porcentaje femenino en la matriculación. Esta baja representación femenina se ha observado tanto en la educación como en el mercado laboral y ha provocado distintos análisis y estudios [4][5].

Titulación	% matrícula 04-05	% matrícula 00-01
Total	26,82%	27,16%
Arquitectura	38,96%	43,10%
Industrial	19,72%	18,79%
Informática	19,28%	19,97%
Caminos	27,45%	27,25%
Telecom.	26,33%	30,40%

Tabla 2. Porcentaje de mujeres matriculadas en distintas enseñanzas técnicas en el curso 04/05 y el curso 00/01

En definitiva, cada vez son menos los alumnos que, desde la educación secundaria, eligen las enseñanzas técnicas y son aún menos los que eligen informática sin que colectivos infrarrepresentados como las mujeres compensen la situación e incrementen su presencia en las titulaciones de informática. Lógicamente esta evolución se debe, entre otras causas, a la percepción de la profesión y de la ingeniería informática que no resulta atractiva a los posibles candidatos.

Lejos de afectar sólo a España, existen datos que revelan que esta tendencia también ocurre en países con mayor desarrollo tecnológico como EE.UU. o Reino Unido. De hecho, la ACM (Association for Computer Machiney) ha previsto el reparto de 250.000 folletos[3] sobre la profesión informática principalmente en institutos para alumnos, profesores y padres para incentivar las vocaciones y evitar la previsible carencia de especialistas para los próximos años. En Reino Unido, la British Computer Scoiety (BCS) se hace eco de la falta de vocaciones[4]. El IEEE (Institute of Electrical and Electronic Engineers) ha promovido un reality show sobre ingeniería donde los participantes deben realizar en equipo, vigilados por cámaras 24 horas, tareas de construcción o ingeniería[5]. Seguramente esta iniciativa televisiva tendrá éxito ya que en España se comprobó como series de TV sobre médicos, periodistas o abogados generaron un importante incremento de

[3] www.acm.org/membership/CareersBrochure.pdf
[4] www.bcs.org/server.php?show=ConWebDoc.9662
[5] http://bmsmail3.ieee.org/u/4938/50505050

matriculaciones en las correspondientes facultades en el siguiente curso académico. Quizás sea el momento de volver a crear una serie como la de McGyver (por cierto, de nuevo emitida en España ante la proliferación de cadenas de TV y la falta de contenidos).

3. Profesión en informática

Aunque a veces se comenta que no es necesario lograr una titulación universitaria para obtener un empleo, lo cierto es que a mayor nivel educativo mejor situación laboral, en promedio, se obtiene. Así, la OCDE [6] establece como media en 2004 que el 82-85% de los titulados en edad activa (25-64 años) tiene empleo frente al 73% de quienes tienen estudios secundarios (incluye la formación profesional de España); en España, la relación es del 81-82% al 73-74%. Es decir, no existe una razón basada en el mercado laboral para no estudiar una titulación si así se desea; porque el 73,3% de los alumnos elige fundamentalmente sus estudios por vocación [7] y son pocos los que tienen en cuenta desde el inicio sólo las salidas laborales. Sin embargo, esto no es siempre así para los padres y familias. En la Universidad Europea de Madrid, donde todos los alumnos que ingresan pasan por una entrevista, hemos visto como en 2003, en los años más negativos del sector de tecnología de la información y de las comunicaciones, había alumnos que deseaban cursar informática y sus padres les "obligaban" a cursar ingeniería industrial porque era una apuesta más segura.

En muchos casos, el prestigio social de la profesión de la informática ha sido influida por la imagen que ha llegado a la Sociedad a través de los medios de comunicación. No son extrañas las noticias que relacionan la informática con delitos o que emplean la palabra "informático" sin distinción para referirse tanto a un titulado universitario de informática como a cualquiera que usa un ordenador con cierta pericia. Así, la detención de una banda de falsificadores motiva que se diga que:

> "…este ingeniero que descifró los códigos secretos de los bancos es "un genio de la informática y de la matemática", por lo que su potencial delictivo era muy alto" [7]

Por otra parte, podemos encontrar una noticia en la que, además de incluir una relación con el delito, la palabra "informático" se asigna a un chico de 15 años [8]. Por último, vemos que a efectos laborales hay anuncios en prensa como el mostrado en la figura 1. También resulta sorprendente la oferta de empleo difundida recientemente por la Fundación Española para la Ciencia y la Tecnología (FECYT) en la que se solicita "Astronomer or Software Engineer" para "Development and Implementation of Virtual Observatory tools for the ESO Science Archive" y las titulaciones aceptadas son Físico, Ingeniero Informático e ¿ingeniero de telecomunicación?

IMPORTANTE EMPRESA NECESITA
INFORMÁTICO/ALMACENERO
PENSAMOS EN UN JOVEN QUE TENGA AMPLIOS CONOCIMIENTOS DE INFORMÁTICA Y QUE COMPATIBILICE LA JORNADA DE TRABAJO EN EL ORDENADOR (75%) Y EN ALMACÉN (25%).
LAS CONDICIONES SERÁN:
CONTRATO LABORAL, SEG. SOCIAL, JORNADA PARTIDA, INCORPORACIÓN INMEDIATA.
[redacted]

Figura 1. Anuncio de informático almacenero

De hecho, un estudio encargado por la Universidad Europea de Madrid (UEM), a una empresa de investigación sociológica y de mercados, sobre posicionamiento y percepción social de las titulaciones era muy revelador sobre la diferencia entre ingenieros informáticos y otras ingenierías. Se concluía respecto de las titulaciones de informática lo siguiente:

- No ha conseguido investirse de un "valor científico superior" por arrastrar un pasado de no cualificación (personal con cursillos tipo academia) y porque no ha sido aceptada como una ingeniería de verdad (con una "superioridad académica").
- Tiene problemas de reconocimiento en la empresa por el recuerdo de un pasado reciente resuelto por profesionales no titulados, ausencia de linaje (quizás sería mejor lobby de presión) como otras ingenierías, muchas veces físicamente ubicado lejos de los centros de poder (en centros de cálculo en sótanos, etc.) y con

escasa habilidad para venderse y endiosarse[6].

- Expectativas positivas de empleo, con un efecto positivo adicional por la creciente ocupación de titulados en los puestos clave que reducirán el intrusismo.
- Formación académica correcta pero con escasez de habilidades personales ("personas peculiares por no decir raras"), quizás en la línea de [10], con una buena imagen como solucionadores de acciones diarias pero sin el "brillo" o creatividad de otras áreas.

En efecto, las expectativas laborales son realmente buenas. Así, un reciente estudio del canal CNN.Money[7] menciona que el trabajo número 1 en los próximos años será el de ingeniero de software y el séptimo para "Computer/IT analyst". En España, ya los medios generales (por ejemplo, [11] y [12]) informan de que las empresas comienzan a sentir la escasez de personal cualificado y tratan de aplicar medidas para paliar la situación. En algún caso se resalta que "los bajos sueldos y la falta de prestigio social desaniman a los profesionales" [11]. Se ha tomado conciencia de que un déficit de especialistas en informática incluso amenaza la recuperación del sector [13].

Es curioso pero, al menos por el autor, no menos esperado el ver cómo la mala imagen percibida socialmente sobre el empleo y las perspectivas laborales tras la crisis de las punto.com (principalmente desde 2001 a 2004) haya dejado paso de nuevo a noticias como éstas. En estos años, se fomentó una imagen de crisis mayor que la real puesto que, a diferencia de los empleos en ingeniería de telecomunicación muy centrados en el propio sector electrónico y de comunicaciones, el empleo en informática es más transversal y permite trabajar en muchos más sectores que cuentan con sus departamentos de informática o requieren inversiones y modernización de sus inafrestructuras y aplicaciones; es decir, el crecimiento, por ejemplo, de las empresas de construcción o inmobiliarias aporta oportunidades de crecimiento de empleo informáticos para su soporte de información. Lo cierto es que en estos años hubo numerosísimas ocasiones en las que los medios reflejaron declaraciones catastrofistas (o presentadas como tales por los medios) por parte de, por ejemplo, las primeras responsables en España de una consultora de recursos humanos o de una gran compañía multinacional del sector. La mala imagen y el desaliento creado fue tal que hizo incluso posible que una broma como la de la figura 2 tuviera bastante crédito al lanzarse[8].

Figura 2. Una "hoax" sobre el desempleo informático

También resulta sorprendente que se vuelvan a emitir estudios e informes más o menos formales (de hecho, por la misma compañía de análisis de mercado: IDC) donde se demuestra la inminente carencia de especialistas Por ejemplo, en 2001 IDC preparó para CISCO un estudio que revelaba la carencia de cientos de miles de especialistas en redes en Europa [14] para 2004; recientemente en 2006 se ha difundido otro informe (concluido en 2005) financiado también por CISCO que alerta de la falta, de nuevo, de cientos de miles de profesionales de redes en Europa para 2008 [15]. Algo similar ocurrió con Microsoft con un informe de IDC muy optimista en 2000[9] [16] (muy utilizado en su momento para apoyar la matriculación en toda clase de estudios de informática no sólo en niveles universitarios) y

[6] Por el contrario, se señala que los titulados en telecomunicación han debido bajar de su endiosamiento sólo tras la crisis de las punto.com

[7] money.cnn.com/magazines/moneymag/bestjobs/

[8] www.pcmag.com/article2/0,1759,1590109,00.asp

[9] En 2003, se veía claro que dichas cifras optimistas no eran lógicas [14].

algunas iniciativas recientes de MS para combatir el fenómeno de "IT-Skills Shortage".

Por supuesto, existen muchos más aspectos de la imagen social y profesional para los titulados en informática que podrían tratarse y mencionarse (algunos de los cuales, en cierto modo, fueron contemplados en el mencionado estudio encargado por la UEM): imagen personal, "frikis", combate a prácticas comerciales de software o de patentes y propiedad intelectual (software libre, etc.), lucha por reconocimiento de atribuciones profesionales, la gran variedad de cualificaciones comerciales de las empresas de tecnología (Microsoft, CISCO, Oracle, Sun, etc.), bajo porcentaje de mujeres, etc..

4. ¿Qué imagen llega desde las universidades?

A través de diversos proyectos de investigación educativa se ha procedido a analizar la imagen profesional que las universidades que imparten titulaciones de informática transmiten a través de los perfiles de egreso declarados por las universidades que tienen aprobado un plan de estudios según el Registro Nacional de Universidad, Centros y Enseñanzas. Esto totaliza un gran número de titulaciones como se refleja en la Tabla 3.

Titulación	*Públicas*	*Privadas*	*Iglesia*
I.T.Inf. Gestión	40	12	4
I.T.Inf. Sistemas	37	11	3
Ing. Informática	41	12	4

Tabla 3. Distribución de universidades que imparten cada una las titulaciones

Los resultados del análisis son variados y amplios [17] pero se pueden resumir mediante el número de puestos mencionados (Tabla 4).

Perfil profesional	*Perfiles de egreso*
Programador	31
Técnico de sistemas	25
Analista programador	20
Comercial	17
Director de informática	17
Especialista	17
Formador/profesor	17
Analista	14
Técnico de comunicaciones/redes	13
Consultor informático	12
Responsable de proyectos/desarrollo	12
Informático	11
Administrador de base de datos	9

Tabla 4. Número de perfiles pcada puesto

Cuando comparamos estos resultados con la demanda real de cada puesto/perfil profesional según se contabiliza en la actualización de los estudios RENTIC [18], obtenemos los resultados mostrados en la Tabla 5.

Perfil profesional	*Perfiles egreso*	*Rank ofertas*	*% de ofertas*
Programador	31	1	10,7%
Técnico de sistemas	25	6	3,6%
Analista programador	20	2	8,2%
Comercial[10]	17	44	0,4%
Director de informática	17	10	2,43%
Especialista	17	26	0,81%
Formador o profesor	17	13	1,82%
Analista	14	5	5,26%
Técnico de redes	13	31	0,71%
Consultor informát.	12	3	7,2%
Responsable de proyectos/desarrollo	12	24	0,91%
Informático	11	36	0,61%
Administrador de base de datos	9	11	2,13%

Tabla 5. Perfiles de egreso y el % de ofertas que los demandan

Algunos perfiles contemplados en la comparativa llaman la atención como es la inclusión del perfil de docente que no tiene una gran demanda en el estudio (si bien es cierto que los puestos tradicionales de profesorado en universidades o en educación secundaria no se publicitan en prensa claramente y, por lo tanto,

[10] En los estudios RENTIC no se contemplan todos los puestos de tipo comercial sino sólo aquellos que requieren titulados en informática.

no están bien representados en RENTIC). También podemos mencionar que el puesto de programador es el más mencionado.

También resulta curiosa la denominación de informáticos que no contribuye a aportar información y permite la confusión poco deseable entre personal no cualificado ni titulado con los ingenieros de la informática. Además puede contribuir a que se asimile a los titulados en informática con personas no acreditadas que realizan intrusismo en la profesión.

En cualquier caso, podemos resaltar que se mencionan puestos bastante conocidos y frecuentes en las ofertas si bien no sabemos si esta elección es precisamente motivada por esto (es decir, que "suene" a los alumnos y padres que consulten la información) o porque realmente el diseño formativo está orientado a formar en estos puestos.

Además de los resultados obtenidos, podemos mencionar distintas observaciones obtenidas durante el proceso de análisis de los textos de perfiles de egreso:

- La extensión del texto del perfil va desde 27 (el más corto) a 370 palabras (el más largo): una excesiva variedad que parece indicar que no es muy clara la idea de la profesión.
- No se utilizan matizaciones ni anglicismos (*manager*, *senior*, *junior*, etc.).
- Hay cinco universidades que muestran el mismo párrafo, se cree que obtenido de la misma fuente pero sin citarla.
- En general, se detectan fallos relacionados con la redacción (esperamos que por falta de dedicación más que por otros problemas) cuando se afirma que "El Ingeniero en Informática [se pone esto en las tres titulaciones cambiando el nombre de la titulación] está capacitado para desarrollar las labores tradicionalmente desempeñadas por informáticos". También se encuentra que "el título de informática de gestión capacita a las competencias que le son propias". O alguna universidad donde la primera salida indicada es la docencia.
- También se encuentran algunas incoherencias como que los perfiles de las ingenierías técnicas se detallen mucho en una universidad mientras que la ingeniería informática se limite a decir que "sirve para empresas de informática, administración, docencia e investigación" o el caso de una universidad donde los perfiles de gestión y sistemas son prácticamente idénticos y parecidos al del grado superior.
- Por último, parece que no es una buena idea la finalidad de frases como la siguiente "El título de ingeniero técnico de sistemas es el que te permite acceder a los estudios de segundo ciclo o bien a Organización Industrial y es ésta última la que te permite trabajar como Director de informática, programador, etc…" (no creemos necesario comentar en detalle).

En definitiva, si los alumnos mirasen estas declaraciones seguramente no llegarían a una idea razonable de la profesión a la que aspirar con sus estudios de ingeniería en informática.

Por último, cabe señalar que los últimos datos de los estudios RENTIC señalan los siguientes resultados (ver Tablas 6 y 7) sobre titulaciones requeridas para los puestos y las áreas de trabajo más habituales.

Área funcional	% de titulados informática
Desarrollo	18,94%
Sistemas	18,97%
Consultoría	12,31%
Dirección	23,17%
Técnicos	22,59%

Tabla 6. Porcentaje de perfiles que requieren titulados de informática por área

Puesto	% de titulados informática
Programador	20,8%
Analista	
Analista programador	
Técnico de sistemas	
Ingeniero de software	

Tabla 7. Porcentaje de puestos que requieren titulados de informática por área

Podemos ver que ciertamente los puestos de dirección tienden a solicitar específicamente más titulados de informática. Hay que aclarar que hay otras denominaciones de titulaciones que incluyen a los titulados de informática no contempladas en estas tablas (por ejemplo, si se solicitan "titulados superiores" en general).

5. Conclusiones

Hemos visto cómo el panorama laboral para los titulados de informática es favorable pero, sin embargo, su reconocimiento social y académico no es el coherente con sus perspectivas. Esta imagen está influyendo en la caída de vocaciones para los estudios de informática precisamente cuando la situación de empleo está en un buen momento, lo que parece augurar un nuevo desajuste laboral que puede provocar efectos no deseados. Lamentablemente la mala imagen sigue presente en los medios de comunicación pero las universidades, al menos con sus perfiles de egreso, no están contribuyendo a aclarar o mejorar la idea sobre la profesión. En definitiva, creemos que debería realizarse un gran esfuerzo en este aspecto para corregir los desequilibrios inminentes que se prevén tanto en las titulaciones de informática como en el mercado laboral. Para ello, por ejemplo, se deberían realizar estudios sectoriales propios de informática (sin intromisión de las telecomunicaciones) con participación de empresas e instituciones y que la información conseguida se incluya en las titulaciones. Esto, además, ayudaría a cerrar la tradicional brecha entre universidad y empresa, que ha llegado a generar muchas críticas a veces tan exageradas como la expresada en [19] sobre universidades en EE.UU. o más suaves como la de [20].

Referencias

[1] The European Higher Education Area. Joint declaration of the European Ministers of Education, Bologna, 19th of June 1999. http://www.sc.ehu.es/siwebso/Bolonia/textos/AEES_EHEA/Bologna_declaration.pdf

[2] The European Higher Education Area. Communication of the Conference of European Ministers Responsible for Higher Education, Bergen, 19-20 May 2005. www.us.es/eees/formacion/html/bergen_declaracion.htm

[3] M.A.Criado "Aulas casi vacías", El Mundo.Ariadna, 1-6-03, p. 1

[4] G. Valenduc et al.,*Widening Women's Work in Information and Communication Technology*, European Commission, 2004.

[5] J. Millar y N. Jagger, "Women in ITEC Courses and Careers", University of Sussex, noviembre, 2001.

[6] OCDE, *Education at a Glance*, 2006, en http://www.oecd.org/edu/eag2006.

[7] Accenture, *Expectativas de los universitarios españoles*, Accenture, 2001.

[8] P. Arenós, "Cae una banda que descifró los códigos bancarios para crear tarjetas sin originales", La Vanguardia, 11-7-06, p. 36.

[9] El País, "Un informático de 15 años, cerebro de una red que estafó 200.000 euros", El País ed. Madrid, 25-4-04, p. 1.

[10] G. R. Djavanshir and W. W. Agresti, Habilidades sociales, "IT Consulting: Communication Skills Are Ke"y, IT Professional, Enero-Febrero 2007, p.46-50

[11] El Periódico, "Las empresas no logran cubrir las plazas de informáticos", El Periódico, 21-1-07, p. 21.

[12] L. Sánchez, "La falta de personal cualificado amenaza la recuperación del sector TIC", ABC, 15-10-06, p.3

[13] A. Milroy, "Networking Skills Shortages in EMEA", IDC Bulletin, mayo, 2001.

[14] M. Kolding y V. Kroa "Networking Skills in Europe: Will an Increasing Shortage Hamper. Competitiveness in the Global Market?", IDC, septiembre, 2005.

[15] A. Milroy y P. Rajah, "Europe's Growing IT Skills Crisis", IDC, 2000.

[16] A. Eisner, "Europe's I.T. Skills Shortage Evaporates", Web Host Industry Review, , www.thewhir.com/print.cfm?page=/features/euro-skills.cfm.

[17] Luis Fernández, María José García, María Cruz Gaya, Verónica Égido, Mario Mata y Nourdine Aliane, "Análisis de la empleabilidad y perfiles de egresado en las titulaciones de informática en España", Actas de las XII Jornadas de Enseñanza Univ. de Informática 2006, pp.231-236.

[18] Luis Fernández Sanz, "Requisitos para el empleo en Nuevas Tecnologías de la Información: el estudio RENTIC", Revista Novática, nº 161, Enero-Febrero 2003.

[19] S. Reisman, "Slaying the Academia Beast", IT Professional, Enero-Feb., 2007, p.63-64.

[20] Expansión, "El sector de la banca considera que la formación universitaria es insuficiente", Expansión, 24 -5-04

La Informática Médica en el Espacio Europeo de Educación Superior

J. Martín Moreno, V.J. Berenguer Miralles, D. Ruiz Fernández, A. Soriano Payá, D. Gil Méndez
Dpto. de Tecnología Informática y Computación
Universidad de Alicante
Carretera San Vicente s/n, 03690 San Vicente (Alicante)
{jmartin,vjberenguer,druiz,soriano,dgil}@dtic.ua.es

Resumen

La informática médica es una disciplina que ha evolucionado notoriamente en los últimos años desembocando en varias líneas de interés y acercando áreas, tan aparentemente distantes, como las tecnologías de la información y la medicina. Buena muestra de ello es el nacimiento de diferentes grupos de investigación y formación, así como eventos de difusión con el único objetivo de hacer valer cada vez más una asociación interdisciplinar de varias áreas para conseguir una mejora en la calidad de vida de los ciudadanos. En este marco de evolución, se quiere plantear la incorporación de una asignatura optativa relacionada con la bioingeniería en los planes de estudios de Informática. Teniendo en cuenta las directrices marcadas por el Espacio Europeo de Educación Superior (EEES) dicha asignatura sería cuatrimestral y con una duración de 108 horas equivalente a 4 créditos ECTS.

1. Introducción

La bioingeniería es la ciencia que estudia y busca la aplicación de principios y técnicas de la ingeniería para solucionar problemas de la biología y la medicina [23]. Podemos dividir esta en tres áreas según estableció en 1997 el Comité de Educación del *Group of Engineering in Medicine and Biology del Institute of Electrical and Electronics Engineers (IEEE)*. Estas áreas son la ingeniería biológica, la cual se encarga de descubrir y clasificar los fenómenos biológicos refiriéndose a la biología como un todo; la ingeniería biomédica, encargada de estudiar la forma de combatir las enfermedades; y la ingeniería clínica, dedicada a dar solución a los problemas asistenciales de los hospitales. A su vez, cada una de estas áreas se nutre de muy diferentes disciplinas, cuya clasificación se podría hacer en ciencias básicas generales y ciencias básicas interdisciplinarias. En el primer grupo podemos destacar la Biología, Fisiología, Química, Física, Matemáticas, Electrónica y Computación; y en el segundo la Biofísica, la Biomatemática y la Bioquímica.

Si tomamos una definición más laxa del concepto de bioingeniería la podemos tratar como la ciencia que agrupa la bioinformática (estudio del genoma esencialmente), el diagnóstico distribuido y cuidado sanitario en el hogar, el modelado y simulación de sistemas biológicos, la bioinstrumentación, la gestión hospitalaria, los sistemas de ayuda al diagnóstico, etc. El conjunto de todas estas actividades es lo que aquí entenderemos por informática médica.

En este artículo presentamos una propuesta de asignatura optativa relacionada con la bioingeniería en los planes de estudios de las Ingenierías en Informática. Se detallan aspectos como las competencias que se pretenden fomentar con dicha asignatura, la interrelación con otras asignaturas ya presentes actualmente y los contenidos que se proponen, atendiendo especialmente a su adaptación al EEES.

2. Relevancia de la Informática Médica en Europa

La informática médica en la sociedad actual (tanto en el marco nacional como europeo) ha evolucionado ramificándose en diferentes líneas de trabajo. Todas estas líneas se han ido desarrollando durante los últimos años en grupos de investigación de distintas universidades europeas, obteniendo una entidad suficiente como para proponer congresos y simposiums específicos. Este desarrollo científico-técnico ha facilitado la oferta docente de asignaturas, cursos,

masters y programas de doctorado en los planes de estudios de titulaciones con carácter más técnico como ingenierías en electrónica, en telecomunicaciones, en informática, etc.

La investigación en biomedicina y ciencias de la salud ha evolucionado en los tres últimos años, gracias en gran parte a los programas sobre investigación científica llevados a cabo por el Ministerio de Sanidad y Consumo, el de Educación y Ciencia y las entidades equivalentes de las diferentes comunidades autónomas. Esta línea de actuación se enmarca dentro de las directrices de la Unión Europea, que a través del 6º y 7º Programa Marco de Investigación, ha promocionado la integración de centros de investigación y sanitarios de gran prestigio, capaces de coordinarse y aumentar sus capacidades.

Nuestro país ha acercado, de forma novedosa y eficiente, su estructura de investigación biomédica a este modelo europeo. De esta forma se tiende a una clara convergencia con Europa y se facilita la aproximación y trabajo en equipo entre grupos interdisciplinares y centros españoles de prestigio. Entre dichos centros encontramos, por ejemplo, el Parque de Investigación Biomédica de Barcelona [21], cuyas líneas de investigación se centran en descifrar la base genética y molecular de la vida y ayudar a solucionar los problemas de salud emergentes aplicando tecnologías informáticas. Otro grupo de investigación destacado dentro del panorama nacional es el de Biotecnología, Electrónica y Telemedicina de la Universidad Politécnica de Valencia [8]. Dentro de este grupo existen varias líneas de investigación, entre las cuales están: bioingeniería, tecnologías para la salud y el bienestar social e informática biomédica. En el panorama nacional también destacamos el grupo de Biotecnología y Telemedicina de la Universidad Politécnica de Madrid [7]. Sus líneas de investigación se dividen en la telemedicina domiciliaria (aplicaciones multimedia en medicina, sistemas distribuidos en atención primaria, telemedicina en países en desarrollo, etc.) y en las imágenes médicas, resaltando la imagen funcional, la imagen microscópica, radioterapia, etc. Podemos mencionar también el grupo de Ingeniería Bioinspirada e Informática para la Salud de la Universidad de Alicante [9]. Este centra actualmente sus líneas de investigación en el diagnóstico distribuido y cuidado sanitario en el hogar (D_2H_2), la cibernética y la biónica, la biometría, la informática médica y la bioinstrumentación. Dentro de estas líneas tiene cabida el desarrollo de aplicaciones de modelado y simulación de sistemas biológicos, el desarrollo de dispositivos biomédicos, aplicaciones de gestión médica, telemedicina y computación ubicua, etc.

Si nos fijamos en la actividad de investigación en biomedicina en Europa encontramos que existe un mayor desarrollo en países como Alemania, Reino Unido, Dinamarca, Holanda, etc. Por ejemplo, en la facultad de ciencias de la vida de la Universidad de Manchester [6] encontramos un grupo dedicado a la investigación en genómica funcional, evolución molecular, análisis de secuencias, bioinformática estructural, etc. Otra universidad donde podemos encontrar un grupo de investigación relacionado con la bioinformática es la de Bergen en Noruega [4]. Este pertenece al departamento de Informática y se dedica esencialmente al análisis de secuencias y estructuras de proteínas y de la actividad de los genes.

Los resultados de la investigación llevada a cabo en todos los grupos y centros citados anteriormente y en muchos otros se difunden en congresos y simposiums. Existe una gran variedad de estos eventos si nos centramos en un marco europeo, pero el abanico se extiende si hablamos de un ámbito internacional. Por ejemplo, el *2nd IEEE International Conference on e-Science and Grid Computing* [10], congreso internacional celebrado en Ámsterdam en diciembre de 2006, tenía como objetivo acercar las posturas de los desarrolladores y los usuarios de aplicaciones biomédicas, y dar a conocer los principales desarrollos en Tecnologías de la Información aplicadas a la e-Ciencia. También podemos destacar también el 4th *International Conference on Information & Communications Technology* [11], el cual en el año 2006 se centró en la Informática Médica.

Hemos podido observar que existe un desarrollo notable de todas las facetas comprendidas en la bioingeniería. Y que se requiere, cada vez más, de la aplicación de los principios y los métodos de la ingeniería y tecnología para la comprensión y resolución de los problemas biológicos y médicos. Como

consecuencia de ello surge la necesidad implacable de incluir la informática médica en las titulaciones universitarias, masters y programas de doctorado, con el fin de cubrir la formación de la gran cantidad de profesionales que se demandan en este campo. Entre las ofertas docentes que podemos encontrar en el panorama nacional tenemos que en el plan de estudios de la Ingeniería en Telecomunicaciones de la Universidad Politécnica de Madrid se propone una intensificación en bioingeniería [19]. Dicha oferta se conforma de una asignatura de primer ciclo: Tecnologías de la Información y las Comunicaciones en Medicina; y cuatro más en el segundo: Fundamentos de Bioingeniería, Señales e Imágenes Médicas, Laboratorio de Telemedicina y Laboratorio de Señales e Imágenes Médicas. Además, existe en esta escuela un programa de doctorado sobre Ingeniería Biomédica en el cual se tratan las líneas de bioinstrumentación y bioelectromagnetismo, informática médica y telemedicina, imágenes biomédicas y simulación médica y biomedicina y biomateriales. En la Universidad Pública de Navarra el Departamento de Ingeniería Eléctrica y Electrónica oferta un título propio de ingeniería médica [20]. Este tiene una duración de 40 horas y va destinado a alumnos de ingenierías industriales, en telecomunicaciones e informáticas. La organización de este título es tal que los alumnos pueden obtenerlo cursando seis asignaturas de libre elección a lo largo del primer y/o segundo ciclo de sus estudios. Dichas asignaturas son: Introducción a la Ingeniería Médica, Fisiopatología fundamental, Telemedicina, Gestión de servicios sanitarios, Tecnologías biomédicas específicas y El espacio sociosanitario. También existe en las titulaciones de Ingeniería en Telecomunicaciones y en Electrónica de las universidades de Vigo, Málaga, Alcalá, La Laguna (Tenerife), Politécnica de Cataluña y Politécnica de Valencia asignaturas optativas relacionadas con la bioingeniería. En estas se busca proporcionar al estudiante una introducción al campo de la ingeniería biomédica y las aplicaciones clínicas relacionadas, la electrofisiología, el análisis de problemas biológicos y médicos, bioelectrónica, señales y sistemas biomédicos, telemedicina, etc. Además, también existen diferentes ofertas de masters como el de la Universidad de Barcelona [16], la Pontificia de Salamanca [17] o la Complutense de Madrid [18]. En estos tres últimos casos estas ofertas de postgrado son propuestas por las facultades de informática y persiguen la especialización del alumno en alguna o varias de las áreas de la bioingeniería, acercándose más al concepto de informática bioinspirada.

Los autores de este texto quieren resaltar que existen en los planes de Ingenierías Informáticas de varias universidades asignaturas designadas como Bioinformática. Sin embargo, dichas ofertas docentes tienen un carácter complementario a otras asignaturas del área de la inteligencia artificial y están dedicadas al estudio de tecnologías surgidas en el contexto de la computación natural; no centrándose en gran medida en el concepto de bioinformática que aquí hemos considerado.

Hay que destacar también la creciente demanda por parte de la empresa pública y privada de profesionales cualificados que cubran los nuevos puestos de trabajo en proyectos y desarrollos vinculados al ámbito de la informática médica.

3. Peculiaridades de la Informática Médica

Hemos visto hasta este punto el desarrollo que existe de la bioingeniería tanto en el panorama nacional como europeo, así como la instauración de la oferta docente relacionada con sus diferentes áreas en los planes de estudio de titulaciones con un perfil más ingenieril. Observamos que tanto la electrónica como la computación juegan un papel fundamental dentro del marco interdisciplinar que define la bioingeniería. Estas materias armonizan con otras disciplinas como pueda ser la biología, la fisiología o la medicina para conformar las diferentes áreas que se definen en la bioingeniería. Si hablamos del papel de la electrónica como ciencia básica general de la que se nutre la bioingeniería encontramos aplicaciones en el desarrollo de sistemas de medida y actuación asociados a los sistemas biológicos, el estudio de los efectos biológicos asociados con la electricidad, el desarrollo de sistemas de adquisición de variables biológicas y biomédicas como pueden ser las imágenes o las bioseñales, etc. La formación en estos aspectos dentro de los planes de estudio de las Ingenierías en Electrónica

y en Telecomunicaciones queda cubierta con intensificaciones y asignaturas optativas como las comentadas en el apartado anterior. Si hablamos del papel que juega la computación o la informática en general dentro de las diferentes áreas que conforman la bioingeniería encontramos gran cantidad de aplicaciones. Entre ellas, las relacionadas con la bioinformática como pueden ser el análisis y descifrado de secuencias de material genético, la gestión del conocimiento o el reconocimiento de patrones. Por otra parte, las relacionadas con la ingeniería médica: clasificación y análisis de señales bioeléctricas, análisis de imágenes de detalles anatómicos y funciones fisiológicas, diseño y desarrollo de sistemas de ayuda al diagnóstico, desarrollo de aplicaciones orientadas a mejorar la gestión médica, tanto en aspectos de accesibilidad como en seguridad, la telemedicina, la computación ubicua, etc. También podemos destacar el diagnóstico distribuido y cuidado sanitario en el hogar, el modelado y simulación de sistemas biológicos, la robótica aplicada a la medicina, etc. Todas estas aplicaciones precisan la intervención de diferentes áreas de la informática entre las cuales destaca la inteligencia artificial, la ingeniería del software, el diseño de bases de datos, las arquitecturas de computadores, la física y electrónica aplicadas, la ingeniería telemática, etc. Además, la mayoría de estas áreas requerirían de una adaptación que las enfocase a las funcionalidades específicas de la informática médica. Por ejemplo, en las aplicaciones bioinformáticas, se precisaría de entornos de computación masivamente distribuidos como las arquitecturas Grid o los grandes supercomputadores para cubrir las crecientes necesidades de procesamiento que se requiere para el descifrado del genoma humano. Otra muestra de adaptación la encontramos en la gestión del conocimiento y los aspectos de seguridad que este lleva asociados, ya que, por ejemplo, los sistemas de registro de información de una gran red sanitaria deben cumplir una serie de políticas de seguridad que cualquier otro sistema no precisa. También los sistemas de ayuda al diagnóstico son cada vez más complicados debido a la enorme casuística de síntomas y predicción de enfermedades o patologías. Con el fin de cubrir la integración y adaptación de las diferentes áreas de la informática anteriormente citadas, se cree necesaria la inclusión de una oferta docente relacionada con la informática médica en los planes de estudio de las Ingenierías en Informática.

4. Propuesta según las directrices del EEES

Como base para cursar la asignatura es recomendable haber adquirido conocimientos relacionados con arquitecturas de computadores, bases de datos, análisis de sistemas, redes e inteligencia artificial.

4.1 Objetivos y competencias

La asignatura se orienta a que el alumno adquiera el mayor número de aptitudes y capacidades en la materia que le permitan desenvolverse sin problemas en un futuro dentro de la informática médica. Entre los objetivos de conocimientos tenemos los siguientes:

- El alumno deberá adquirir técnicas y métodos de trabajo para llevar a cabo o formar parte de un desarrollo destinado a resolver problemas biomédicos.
- El alumno comprenderá la importancia de la información a tratar y de la seguridad que se ha de aplicar en el acceso a datos.
- El alumno deberá comprender el proceso de análisis seguido en el Proyecto Genoma Humano y las conclusiones derivadas de éste.
- El alumno adquirirá habilidades en el desarrollo de sistemas de análisis, aprendizaje y de ayuda al diagnóstico en D_2H_2.

Finalmente, destacan los siguientes objetivos de competencias, orientados a que el alumno llegue a:

- Conocer y utilizar de modo adecuado diversas herramientas hardware y software para solventar diferentes problemas vinculados con la materia.
- Adquirir hábito de trabajo continuado en equipos multidisciplinares.
- Aplicar sus conocimientos para desarrollar hábitos de análisis, autocorrección y mejora de los resultados obtenidos.

4.2 Metodología

Durante las clases teóricas se introducirán los conceptos básicos de cada tema para que los alumnos, a través de su participación en las sesiones (debatiendo, comentando, aportando ideas o rectificaciones) y su posterior estudio, alcancen conocimientos más avanzados por sí mismos.

Las sesiones prácticas empezarán cada bloque con una explicación básica que introducirá al alumno en la temática a tratar. El alumno abordará la propuesta hasta alcanzar los objetivos marcados satisfactoriamente, entregando una memoria con los métodos aplicados, resultados, conclusiones y problemas encontrados durante la realización de la práctica.

La metodología que se seguiría consta de tres partes bien diferenciadas:

- Conocimientos mínimos. Parte eminentemente teórica, en la que se explican los conocimientos básicos que debe adquirir el alumno a lo largo de la asignatura. Este punto considerará las horas docentes teóricas y una porción de horas dedicadas por el alumno fuera del aula en la revisión de conceptos explicados, ampliación de conocimientos y para solventar dudas de forma personalizada en tutorías.
- Prácticas. Durante el desarrollo de la asignatura se realizarán cinco prácticas, haciendo corresponder la temática con el bloque que se esté impartiendo en teoría. Consistirá en una práctica guiada por bloque, que tendrá además un componente de investigación y desarrollo por parte del alumno. Las prácticas serán llevadas a cabo durante las horas de docencia en laboratorio.
- Proyectos. El alumno deberá realizar un proyecto teórico-práctico que se centre en alguno de los puntos estudiados o analizados en la parte teórica o práctica de la asignatura. Deberá entregarse una completa documentación del trabajo, así como los códigos o desarrollos alcanzados en el proyecto. El alumno realizará estos proyectos dentro de las horas de trabajo personal previstas en el marco del Espacio Europeo de Educación Superior. Como ayuda a la realización del proyecto se proponen horas dedicadas a un seguimiento semanal personalizado por parte del profesorado de la asignatura.

4.3 Contenidos

El contenido de la asignatura ha sido dividido de forma que en su desarrollo se aborden los conocimientos y campos de investigación básicos sobre la materia. Para esto, el temario se ha estructurado en cuatro temas o bloques principales, que englobarán el grueso de la asignatura, precedidos de un primer tema de introducción. Esta distribución se basa en los temarios existentes en las diferentes ofertas docentes que se expusieron en el segundo punto de este texto (Relevancia de la Informática Médica en Europa). Se han seleccionado las materias en las que mayor incidencia consideramos que puede tener la ingeniería informática.

Dada la amplitud y novedad del temario, no es posible su tratamiento y estudio a partir de un único libro de texto, se proponen como punto de partida las referencias [2], [3], [5], [12] y [14].

Tema 1. Introducción

En este tema se presentarán al alumno conocimientos iniciales acerca de la informática médica, como pueden ser historia, definiciones, relación transversal entre medicina e informática, interdisciplinariedad de la materia, así como proyectos reales o casos de uso de la informática aplicada a la medicina.

En la práctica se propondrá una sesión en la que el alumno, durante la primera mitad del tiempo, buscará información sobre informática médica. En la segunda mitad de la sesión se abrirá un debate participativo sobre las competencias de los ingenieros informáticos en el sector médico.

Tema 2. Bioinformática

En el segundo bloque se estudiará la bioinformática y su definición, explicando las dos posibles visiones que ha adoptado; qué puede aportar la informática a la biología y qué necesita la biología de la informática. Se estudiarán distintos puntos como son algoritmos, análisis de secuencias de ADN (proyecto del Genoma Humano), análisis y predicción de estructuras de

proteínas, etc. Se revisarán los problemas computacionales actuales y las soluciones que se proponen desde los distintos grupos de investigación. Finalmente se estudiará la fuerte relación que guarda la biología con las arquitecturas de computadores para el análisis y solución eficiente de esos problemas de computación.

En prácticas se propondrá un trabajo sobre bases de datos del proyecto Genoma, para el alineamiento múltiple y análisis de familias de proteínas, predicción computacional de genes mediante marcos abiertos de lectura (ORFs) o de genómica comparada.

Tema 3. Sistemas de ayuda al diagnóstico

En la ayuda al diagnóstico se estudiarán distintas técnicas de resolución de problemas y modelos de aprendizaje, dividiendo el tema en cuatro partes. En primer lugar se introducirán los modelos ocultos de Markov y su aplicación para el reconocimiento de patrones. En la segunda parte se estudiará el aprendizaje mediante un perceptrón aplicando la regla delta y perceptrón multicapa, con la aplicación de *backpropagation*. En la tercera parte se estudiarán los mapas auto-organizativos aplicados a la toma de decisiones de diagnóstico. Finalmente se estudiará el modelo de árboles de decisión o ID3 como apoyo a la toma de decisiones basado en reglas.

En las prácticas de este tema se propondrá la implementación de alguno de los modelos estudiados en teoría. Con la implementación se pretende que el programa analice un conjunto de datos previamente y que los clasifique para que, partiendo de esa información, sea capaz de asignar a una clase en concreto nuevos datos o información que le sean introducidos. Para esto se podrá implementar, por ejemplo, un perceptrón multicapa en lenguaje C para la clasificación de disfunciones. Se proporcionará una base implementada de los algoritmos, con la idea de que el alumno pueda avanzar más rápidamente y profundizar así mejor en los conceptos.

Tema 4. Modelado y simulación de sistemas biológicos

En el cuarto tema se abordará el modelado funcional mediante técnicas matemáticas, sistemas de modelado CAD/CAM, simulación de movimientos físicos, representación del comportamiento físico del cuerpo humano mediante reglas físicas y matemática aplicada, necesidades computacionales para la renderización y simulación del sistema, etc. También se tratarán temas relacionados con técnicas paralelas y sistemas distribuidos orientados a la simulación de sistemas biológicos.

Se realizará una práctica en la que se utilizará un software de simulación para recrear el comportamiento de distintos órganos o funciones del cuerpo humano.

Tema 5. Gestión hospitalaria

Este tema incluirá tres apartados claramente diferenciados. En primer lugar se hablará de la normativa para el tratamiento de datos privados, introduciendo aspectos legales de la protección de datos y reglamentos de seguridad. En el segundo apartado se tratarán los sistemas de información médicos, incluyendo el tratamiento de la información dentro de bases de datos médicas, el acceso a la información y su protección frente a accesos fraudulentos. Finalmente se abordará D_2H_2, viendo las bases de este nuevo tipo de redes, consideradas como una extensión de la red de datos de un hospital hacia el domicilio de los pacientes, incluyendo los protocolos de comunicación y dispositivos actuales y futuros que utilizarán.

Como práctica se propondrá la implementación de un sistema de envío y recepción de información (mediante sockets, RMI o WebServices), utilizando alguno de los protocolos explicados en teoría, analizando las ventajas e inconvenientes del protocolo escogido frente a otros. También en este caso se proporcionará una base implementada de los algoritmos, con la idea de que el alumno pueda avanzar más rápidamente y profundizar así mejor en los conceptos.

4.4 Planificación

La extensión de la asignatura será de un cuatrimestre. Se le han asignado un total de cuatro créditos ECTS divididos a partes iguales entre teoría y práctica. La asignación horaria será de dos horas semanales de teoría, dos de prácticas y otras dos horas que el alumno deberá administrar libremente, haciendo un total de ciento ocho horas lectivas. De esta forma el compendio final de

créditos y horas lectivas será de veintisiete, quedando dentro de lo estipulado por el EEES (entre veinticinco y treinta por crédito) [24]. Dentro de cada una de las dieciocho semanas contabilizadas se ha distribuido la materia de forma que se impartirán tanto clases de teoría como de prácticas semanalmente. Cada sesión tendrá una duración de dos horas, haciendo un total de cuatro horas semanales lectivas, además del tiempo que el alumno dedique para asistir a tutorías. En cuanto a las horas no lectivas, el alumno debería dedicar fuera del aula otras dos horas más semanalmente. Con este planteamiento, la asignatura podría ser comprendida en conjunto, habiendo tiempo suficiente también para ahondar en las materias por las que el alumno sienta especial interés dentro del marco de la asignatura [22]. Una vez finalizado el periodo docente se realizará una prueba de conocimientos de corta duración (alrededor de una hora).

En primer lugar se impartirá una sesión que introduzca al alumno en el marco de la asignatura, ofreciendo un primer contacto con los contenidos, definiciones y estado de la informática médica. A continuación se abordarán los tres bloques principales de la asignatura, hablando de bioinformática, sistemas de ayuda al diagnóstico y modelado y simulación de sistemas biológicos, que ocuparán la práctica totalidad de la asignatura, haciendo mayor hincapié en los temas tres y cuatro por ser los más directamente relacionados con las capacitaciones de un ingeniero. Finalmente se tratará el tema de gestión hospitalaria y normativa, al que se dedicarán las dos sesiones finales. Esta distribución horaria se puede apreciar en la siguiente tabla.

	Teoría (2 ECTS)	**Práctica (2 ECTS)**
Tema 1	1 sesión (2h)	1 sesión (2h)
Tema 2	5 sesiones (10h)	5 sesiones (10h)
Tema 3	5 sesiones (10h)	5 sesiones (10h)
Tema 4	5 sesiones (10h)	5 sesiones (10h)
Tema 5	2 sesiones (4h)	2 sesiones (4h)

Tabla 1. Distribución de horas lectivas para el temario de la asignatura

4.5 Evaluación

La evaluación se realizará aportando un 20% de la nota mediante un examen de conocimientos básicos (N_{examen}), un 40% corresponderá a las prácticas en laboratorio (P_1, P_2, P_3, P_4 y P_5) y otro 40% de la nota al proyecto teórico-práctico ($N_{proyecto}$). Tanto los proyectos como las prácticas serán controlados semanalmente por el profesorado en tutorías o durante las sesiones en laboratorio.

Las prácticas se evaluarán de forma independiente y se ponderarán en función de su presencia en la asignatura. Pudiendo aprobar las prácticas sin haberlas superado todas, siempre y cuando se obtenga una nota final igual o superior a cinco.

En cuanto a la parte teórica, no será necesario aprobar ambas partes por separado (proyecto y examen), la nota será ponderada y sumada independientemente de la calificación obtenida en cada parte. En la evaluación de cada parte por separado, la puntuación estará entre cero y diez puntos. Estas evaluaciones independientes serán ponderadas posteriormente.

Las calificaciones de la asignatura quedarán como sigue:

$$N_P = P_1 \cdot 0.05 + P_2 \cdot 0.25 + P_3 \cdot 0.30 + P_4 \cdot 0.30 + P_5 \cdot 0.10$$
$$N_F = N_{examen} \cdot 0.2 + N_{proyecto} \cdot 0.4 + N_P \cdot 0.4$$
$$Si\ N_P \geq 5\ y\ N_{examen} + N_{proyecto} \geq 5$$

5. Conclusiones

Como hemos visto a lo largo del texto existen y siguen apareciendo grupos de investigación, eventos y ofertas docentes dedicadas al desarrollo y estudio de las distintas disciplinas que componen la informática médica. Esto genera la conveniencia de incluir una asignatura de iniciación en el campo de la medicina dentro de los nuevos planes docentes de las ingenierías informáticas en el marco definido por el EEES. Esta asignatura introducirá al alumno en las distintas áreas de la informática médica, aportándole los conocimientos necesarios que le permitan abordar nuevos desarrollos o proyectos.

Referencias

[1] Agencia Nacional de Evaluación de la Calidad y Acreditación. Libro blanco del Título de Grado en Ingeniería Informática. *http://www.aneca.es*

[2] J.D. Broncito. *The Biomedical Engineering Handbook*. 2nd Edition. CRC Press – IEEE Press 2006.
[3] J.J. Carr, J.M. Brown. *Introduction to Biomedical Equipment Technology*. 4th Edition. Prentice Hall 2000.
[4] Department of Informatics, University of Bergen. *http://www.ii.uib.no*
[5] J. Enderle, S. Blanchard, J. Bronzino. *Introduction to biomedical engineering*. Academic Press 1999.
[6] Faculty of Life Sciences, University of Manchester. *http://www.ls.manchester.ac.uk*
[7] Grupo de Bioingeniería y Telemedicina de la Universidad Politécnica de Madrid. *http://www.gbt.tfo.upm.es*
[8] Grupo de Bioingeniería, Electrónica y Telemedicina de la Universidad Politécnica de Valencia. *http://bet.upv.es*
[9] Grupo Ingeniería Bioinspirada e Informática para la Salud, Universidad de Alicante. *http://www.dtic.ua.es/ibis*
[10] 2nd IEEE International Conference on e-Science and Grid Computing. *http://www.escience-meeting.org/eScience2006/index.html*
[11] 4th International Conference on Information & Communications Technology. *http://icict.gov.eg/ICICT-2006/ICICT2006.html*
[12] R. S.H. Istepanian, S. Laxminarayan, C.S. Pattichis. *M-Health*, Springer 2006.
[13] I.S. Kohane. *Bioinformatics and clinical informatics: the imperative to collaborate*. J Am Med Inform Assoc 2000; 7: 512-6.
[14] F. Martín, V. Maojo, G. López. *Integrating genomics into health information systems.* Challenges for Medical Informatics as an Academic Discipline 2001.
[15] M.A. Martínez, V. Carrasco. *Investigar en diseño curricular*. Vol. I, Marfil 2005.
[16] Máster en Biomedicina, Universidad de Barcelona. *http://www.ub.es/biomed/cas/index.htm*
[17] Máster en Gestión de la Información y el Conocimiento en Ciencias de la Salud, Universidad Pontificia de Salamanca. *http://informatica.upsam.net/postgrado/index.php?Mod=Estudios&Section=Mostrar&IdEstudio=76&Lang=es*
[18] Máster en Investigación en Informática, Universidad Complutense de Madrid. *http://www.fdi.ucm.es/posgrado/*
[19] Oferta docente (intensificación en bioingeniería), Universidad Politécnica de Madrid. *http://www.gbt.tfo.upm.es/web-gbt/docencia/94.asp*
[20] Oferta docente (título propio), Universidad Pública de Navarra. *http://www.unavarra.es/organiza/etsiit/cas/estudiantes/ingmed/page_05.htm*
[21] Parque de Investigación Biomédica de Barcelona. *http://www.prbb.org*
[22] Real Decreto 1125/2003. BOE núm. 224. *http://www.boe.es*
[23] F. Sanz, C. Díaz, F. Martin, J. Bonis. *European support to Biomedical Informatics development: in pursue of genomic medicine.* IEEE Computer Society, 2004.
[24] ECTS – European Credit Transfer and Accumulation System. *http://ec.europa.eu./education/programmes/socrates/ects/index_en.html*

Programación, algoritmos y estructuras de datos

Metodología para la enseñanza aprendizaje de la lógica de la programación orientada a objetos

Leobardo López Román
Dpto. de Ingeniería Industrial y de Sistemas
Universidad de Sonora
Hermosillo, Sonora, 83000, México
llopez@industrial.uson.mx

Resumen

A la comunidad académica le llevó muchos años consolidar la forma correcta como los estudiantes deben aprender a programar computadoras. Primero, deben aprender la metodología de la programación, usando técnicas de diseño algorítmicas o pseudolenguajes; y después, deben aprender como implementarla usando un lenguaje de programación.

Con el desarrollo del lenguaje Java y la penetración que ha tenido como el primer lenguaje que muchos estudiantes están aprendiendo; y debido a la falta de una metodología apropiada, se está cayendo en el error de enseñar a programar directamente con el lenguaje Java, dejando de lado el desarrollo de la lógica; y se están formando programadores sin lógica.

Este autor tiene la convicción de que en la actualidad los estudiantes deben aprender un lenguaje orientado a objetos como Java; pero antes, deben desarrollar la lógica básica de la programación orientada a objetos. Es por ello que he desarrollado una metodología de la programación, usando pseudocódigo en forma similar como se usaba con la programación estructurada; pero ahora integrando la lógica básica de la programación con los conceptos y estructuras de la programación orientada a objetos.

En esta ponencia se presenta un resumen de la metodología original, la cual esta escrita con todo detalle en el libro [12] *Metodología de la programación orientada a objetos*; publicado por la editorial Alfaomega (www.alfaomega.com.mx), México, 2006; ISBN 970-15-1173-5. Y es distribuido en los países de habla hispana.

1. Motivación

En la actualidad muchos estudiantes y profesionales de la programación de computadoras están aprendiendo Java, que es un lenguaje orientado a objetos; sin embargo, muchos de ellos, no están aprendiendo a programar orientado a objetos; porque se les está enseñando prácticamente en forma directa con el lenguaje Java; y no se les está enseñando a "pensar"; es decir, no están desarrollando la lógica de la programación orientada a objetos.

La idea de este autor es que lo fundamental al aprender a programar computadoras es desarrollar la lógica necesaria para solucionar problemas en forma algorítmica, independientemente de algún lenguaje de programación; esto es, aprender a diseñar programas (algoritmos) usando un pseudolenguaje; y no hacerlo directamente con un lenguaje.

Metodología de la programación orientada a objetos, es un desarrollo que viene a coadyuvar en la solución de una necesidad largamente experimentada por la comunidad académica de la programación de computadoras; contar con un método que permita conducir la enseñanza-aprendizaje de la programación, mediante el uso de un pseudolenguaje de diseño de programas (algoritmos) orientados a objetos.

La metodología contiene en forma natural los conceptos, estructuras y filosofía que se han generado hasta estos tiempos en que la programación orientada a objetos y el lenguaje Java marcan la pauta de la programación de computadoras.

Esta metodología es el resultado de la integración y adaptación de varias técnicas, como son; los conceptos y estructuras de la programación orientada a objetos: objetos, clases, encapsulación, herencia y polimorfismo; con el diagrama de clases de UML (Unified Modeling Language); con la arquitectura modelo-vista-controlador; con algunos conceptos introducidos por el lenguaje Java; y con los conceptos y bases lógicas de la programación estructurada en pseudocódigo.

Dicha metodología permite diseñar programas o algoritmos orientados a objetos; y prepara a los estudiantes para que puedan aprender y comprender cualquier lenguaje orientado a objetos como Java.

2. Estado del arte

La actividad de programar computadoras ha tenido varias crisis, provocadas por el permanente aumento en la complejidad de las aplicaciones que deben enfrentarse, lo que provoca que las técnicas y estructuras que resultan adecuadas en un momento, con el paso del tiempo se vuelvan inadecuadas. Esta problemática ha dado origen a dos revoluciones: La primera, a la que se le llamó Programación Estructurada; permitió evolucionar desde programar de una forma "tradicional", casi artesanal; a programar de una mejor forma, que aportó las bases para sustentar la segunda revolución en la evolución de los paradigmas de programación; a lo que hoy se conoce como programación orientada a objetos.

2.1. Programación tradicional

En la década de los 60´s y principios de los 70´s se programaba en forma "tradicional", en esos tiempos sólo existían las estructuras lógicas: Secuenciación, If y For (que se conocía como Do en Fortran); y se utilizaban los diagramas de flujo como técnica de diseño de programas (algoritmos). Al aumentar la complejidad de las aplicaciones que se enfrentaban, esa forma de programar tuvo una severa crisis.

2.2. Programación estructurada

A principios de la década de los 70´s, surge un movimiento llamado programación estructurada que vino a añadir nuevas estructuras, técnicas y conceptos a la programación: Se añadieron las estructuras lógicas DO-UNTIL, DOWHILE y se formalizaron el IF-THEN, IF-THEN-ELSE y CASE. Se inventó el módulo, la función y el concepto de parámetros. Se desarrollaron nuevas técnicas de diseño de programas (algoritmos): Pseudocódigo, diagramas Warnier, diagramas Chapin, Jackson, Diseño estructurado de Yourdon, Top Down Design (Diseño descendente), entre otras; que vinieron a desplazar a la tradicional técnica de diagramas de flujo.

Fueron apareciendo nuevos lenguajes: Pascal, C, Cobol estructurado, Basic estructurado. Se estableció que se debe aprender a programar utilizando un pseudolenguaje, es decir, no enseñar directamente con un lenguaje. Y se estableció que se debe usar un estilo de programación que haga más entendible el algoritmo y el programa.

Nuevamente al aumentar la complejidad de las aplicaciones que se enfrentaban, esa forma de programar tuvo una severa crisis. Esto llevó a que siguiera evolucionando y se generaron los conceptos de programación modular, y luego el concepto de abstracción de datos, para dar paso al desarrollo de la programación orientada a objetos.

2.3. Programación orientada a objetos

Aunque la programación orientada a objetos (POO) aparece muchos años antes, es a mediados de los 90´s cuando se generaliza su uso. La POO añade a la programación una nueva estructura: el Objeto, con sus conceptos; objetos, clases, encapsulación, herencia y polimorfismo. Aparecen nuevas técnicas de diseño: Booch, Rumbaugh, Jacobson, Yourdon, UML (Unified Modeling Language), etc. Se desarrollan nuevos lenguajes: C++, Java, C#, etc.

3. Problemática de la enseñanza aprendizaje de la programación orientada a objetos

En los últimos años se ha insistido y ejercido una gran presión para que Java sea el primer y único lenguaje que los estudiantes deben aprender. En consecuencia, muchas instituciones educativas, están enseñando Java desde la fase introductoria a la programación de computadoras; eliminando una formación previa que permita el desarrollo de la lógica básica de la programación. Alguna gente dice que al estudiar el lenguaje Java va implícito el desarrollo de la lógica; y que la programación es mucho más fácil, rápida, agradable y avanzada

en Java que lo que anteriormente era la programación.

Alguna gente dice que cualquier persona que no sepa nada de programación, puede entender fácilmente los conceptos de la programación orientada a objetos; y estoy de acuerdo en parte, porque en un nivel abstracto cualquiera puede comprenderlos. Pero en el momento en que se debe implementar los objetos en instrucciones en un lenguaje de programación como Java, es donde se dan cuanta que "algo" falta. Porque un programa orientado a objetos, se compone por un conjunto de objetos; y cada objeto, por un conjunto de métodos que implementan las funciones del objeto; a algunos de esos métodos hay que enviarles datos a través de parámetros, para que establezcan y accedan los datos; y otros métodos realizan cálculos. De manera que ese "algo" que falta es la lógica básica de la programación; que consiste en: Tipos de datos; entero, real, cadena, arreglos, etcétera; Estructuras de control; secuenciación, if-then, if-then-else, switch, do-while, for, while; métodos (módulos y funciones definidas por el usuario); parámetros por valor y por referencia. Es por ello que digo que esos elementos y estructuras son la base de la programación orientada a objetos; y que una persona que no desarrolle esas bases, jamás podrá comprender cómo implementar los métodos de objetos que procesan datos.

Por lo anterior, pienso que la programación orientada a objetos no se está enseñando adecuadamente en las instituciones de educación. ¿Cuál es la causa? Que en la bibliografía existente sobre programación orientada a objetos; esta ausente la metodología de la programación orientada a objetos enfocada a estudiantes principiantes. En otras palabras, existen muchos libros sobre POO y UML, pero no están enfocados para niveles básicos de aprendizaje. Los libros que están enfocados para niveles básicos de aprendizaje son los libros de Java, que son excelentes manuales del lenguaje Java; pero no conducen el aprendizaje de la lógica básica de la programación inmersa en la programación orientada a objetos. Llevando a que los estudiantes "aprendan a programar" sin desarrollar la lógica. Esto significa que los estudiantes realmente están aprendiendo a codificar usando el lenguaje Java, que es un lenguaje orientado a objetos; pero, no están aprendiendo a programar orientado a objetos usando el lenguaje Java, que sería lo correcto. Porque programar es un proceso que implica diseñar el programa antes de codificarlo. Además, sabemos que si un estudiante aprende a programar directamente con el lenguaje que esta de moda, su mente queda "casada" con ese lenguaje; y cuando el lenguaje que esta de moda cambie, la formación que se le dio con el anterior lenguaje se convierte en deformación [12].

4. Metodología de la programación orientada a objetos

Con el objetivo de llenar ese vacío que ha dejado la bibliografía y con el propósito de coadyuvar en el mejoramiento de la enseñanza-aprendizaje de la programación de computadoras, he desarrollado esta metodología de la programación orientada a objetos.

La metodología se divide en dos partes; en la *primera parte*, que abarca del capítulo uno al nueve, se estudia la técnica pseudocódigo y su uso en el diseño de algoritmos pequeños que tienen una sola tarea o función, por tanto, se establece el uso de una clase y dentro de la clase el método principal, donde se plasma la lógica que soluciona el problema. En esta primera parte se da énfasis al desarrollo de la lógica básica de la programación usando pseudocódigo. Se estudian los tipos de datos, identificadores, operaciones de entrada, cálculo y salida.

Las estructuras de control: La secuenciación; la selección simple (IF THEN), doble (IF THEN ELSE) y múltiple (SWITCH); la repetición DO…WHILE, la repetición FOR y la repetición WHILE. Los arreglos unidimensionales, bidimensionales, tridimensionales y tetradimensionales. Y por último de esta primera parte, se estudia cómo usar más de un método en la clase, en problemas que involucran a más de una tarea o función, métodos que no regresan valor (equivalente a módulos en la programación estructurada), métodos que regresan valor (equivalentes a funciones definidas por el usuario en la programación estructurada), parámetros por valor y por referencia.

Este autor tiene la convicción de que el estudiante debe desarrollar las bases lógicas de la programación; es por ello, que esta primera parte es lo que se estudia o estudiaba en un primer curso de lógica de programación con técnicas estructuradas, pero enfocando la estructura del algoritmo en forma apropiada a la programación orientada a objetos; usando una clase y dentro de la clase el método principal, entrenando a los estudiantes para que desarrollen las bases lógicas de la programación de computadoras.

A continuación se presenta un ejemplo, para mostrar una idea general de cómo se usa la primera parte de la metodología en la solución de una aplicación.

Problema 1:

Elaborar un algoritmo que permita leer un vector de diez números en un arreglo de 10 elementos; que lo imprima e imprima la media. Leer el vector en un método; calcular la media en otro método; e imprimir el vector en otro método. Utilizando parámetros.

```
Algoritmo MEDIA CON METODOS
 Clase MediaConMetodos
  1. Método principal
     a. Declaraciones
        Variables
           vector: Arreglo[10] Real
           promedio: Real
     b. Llamar leerVector(vector)
     c. promedio = calcularMedia(vector)
     d. Llamar imprimirVector(vector)
     e. Imprimir promedio
     f. Fin Método principal

  2. Método leerVector(Ref vec: Arreglo[10] Real)
     a. Declaraciones
        Variables
           n: Entero
     b. FOR n=0; n<=9; n++
           1. Solicitar elemento vec[n]
           2. Leer vec[n]
     c. ENDFOR
     d. Fin Método leerVector

  3. Método calcularMedia(Ref v: Arreglo[10]
                                  Real): Real
     a. Declaraciones
        Variables
           sumatoria, prom: Real
           i: Entero
     b. sumatoria = 0
     c. FOR i=0; i<=9; i++
           1. sumatoria = sumatoria + v[i]
     d. ENDFOR
     e. prom = sumatoria / i
     f. return prom
     g. Fin Método calcularMedia

  4. Método imprimirVector(Ref vect: Arreglo[10]
                                  Real)
     a. Declaraciones
        Variables
           x: Entero
     b. FOR x=0; x<=9; x++
           1. Imprimir vect[x]
     c. ENDFOR
     d. Fin Método imprimirVector
 Fin Clase MediaConMetodos
Fin
```

Explicación:

El algoritmo tiene una clase; en la cual se tienen cuatro métodos. El método principal, en el que se define la variable vector, como un arreglo de 10 elementos; y la variable promedio. Enseguida llama al método leerVector(vector), enviando vector como parámetro y conectándolo con el parámetro por referencia vec; en el cual lee los 10 números.

Luego llama al método calcularMedia(vector) enviando vector como parámetro y conectándolo con el parámetro v; del cual calcula la media y la devuelve para colocarla en promedio.

A continuación llama imprimirVector(vector) enviando vector como parámetro y conectándolo con el parámetro vect; el cual imprime. Y finalmente imprime el promedio.

En la *segunda parte* de la metodología, que abarca del capítulo diez al dieciséis, es donde se estudian de lleno los conceptos de la programación orientada a objetos, integrándolos con el concepto de diagrama de clases de UML (Unified Modeling Language), con la arquitectura modelo-vista-

controlador, con las estructuras estudiadas en los primeros nueve capítulos y la incorporación de los conceptos de la programación orientada a objetos en la técnica pseudocódigo, logrando una metodología de la programación que permite diseñar algoritmos orientados a objetos.

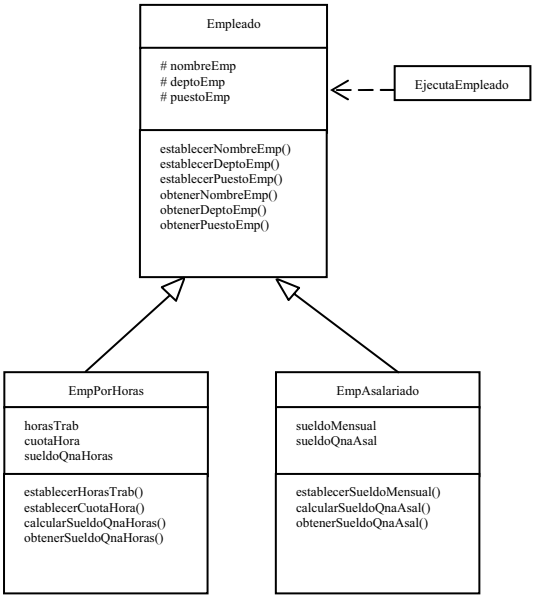
Lo relevante del método es que enseña a programar computadoras utilizando un pseudolenguaje (pseudocódigo), es decir, sin utilizar la computadora directamente. Esto permite desarrollar las capacidades mentales lógicas que una persona debe tener para programar computadoras y sienta las bases de disciplina y buena estructura. Este enfoque se le dificulta a mucha gente, sin embargo, hay que enfrentarlo, porque siendo la programación una actividad intelectual que requiere mucha creatividad, capacidades de abstracción, de análisis, y de síntesis; éstas no se pueden desarrollar operando un lenguaje en la computadora, sino ejercitando la mente con una metodología apropiada. A continuación se presenta un ejemplo, para mostrar una idea general de cómo se usa la segunda parte de la metodología en la solución de una aplicación.

Problema 2:

En cierta empresa se tienen empleados; los empleados se dividen en dos tipos: empleados por horas, a los que se les paga de acuerdo al número de horas trabajadas y a una cuota que se les paga por hora. El otro tipo son los empleados asalariados, a quienes se les paga de acuerdo a un sueldo fijo mensual. Por cada empleado se tienen los datos: Nombre, Departamento, Puesto; si es empleado por horas, el número de horas que trabajó y la cuota que se le paga por hora; si es empleado asalariado, el sueldo mensual que se le paga. Elaborar un algoritmo que permita leer los datos de los empleados e imprimir el nombre y el sueldo quincenal de cada empleado.

A continuación se tiene la solución en dos partes: En la primera, se diseña el diagrama de clases, que contiene la estructura general del programa (algoritmo); y después, en la segunda parte, se diseña el algoritmo que contiene la lógica que soluciona el problema usando pseudocódigo.

Diagrama de clases



Algoritmo CALCULA SUELDOS DE EMPLEADOS

Clase Empleado

1. Declaraciones
   Datos
   # nombreEmp: Cadena
   # deptoEmp: Cadena
   # puestoEmp: Cadena
2. Método establecerNombreEmp(nom: Cadena)
   a. nombreEmp = nom
   b. Fin Método establecerNombreEmp
3. Método establecerDeptoEmp(dep: Cadena)
   a. deptoEmp = dep
   b. Fin Método establecerDeptoEmp
4. Método establecerPuestoEmp(pue: Cadena)
   a. puestoEmp = pue
   b. Fin Método establecerPuestoEmp
5. Método obtenerNombreEmp(): Cadena
   a. return nombreEmp
   b. Fin Método obtenerNombreEmp
6. Método obtenerDeptoEmp(): Cadena
   a. return deptoEmp
   b. Fin Método obtenerDeptoEmp

```
 7. Método obtenerPuestoEmp(): Cadena
    a. return puestoEmp
    b. Fin Método obtenerPuestoEmp
Fin Clase Empleado

Clase EmpPorHoras hereda de Empleado
 1. Declaraciones
    Datos
        horasTrab: Entero
        cuotaHora: Real
        sueldoQnaHoras: Real
 2. Método establecerHorasTrab(horasTr: Entero)
    a. horasTrab = horasTr
    b. Fin Método establecerHorasTrab
 3. Método establecerCuotaHora(cuotaHr: Real)
    a. cuotaHora = cuotaHr
    b. Fin Método establecerCuotaHora
 4. Método calcularSueldoQnaHoras()
    a. sueldoQnaHoras = horasTrab * cuotaHora
    b. Fin Método calcularSueldoQnaHoras
 5. Método obtenerSueldoQnaHoras(): Real
    a. return sueldoQnaHoras
    b. Fin Método obtenerSueldoQnaHoras
Fin Clase EmpPorHoras

Clase EmpAsalariado hereda de Empleado
 1. Declaraciones
    Datos
        sueldoMensual: Real
        sueldoQnaAsal: Real
 2. Método establecerSueldoMensual(sdo: Real)
    a. sueldoMensual = sdo
    b. Fin Método establecerSueldoMensual
 3. Método calcularSueldoQnaAsal()
    a. sueldoQnaAsal = sueldoMensual / 2
    b. Fin Método calcularSueldoQnaAsal
 4. Método obtenerSueldoQnaAsal(): Real
    a. return sueldoQnaAsal
    b. Fin Método obtenerSueldoQnaAsal
Fin Clase EmpAsalariado

Clase EjecutaEmpleado
 1. Método principal
   a. Declaraciones
       Variables
           nomEmp, depto, puesto: Cadena
           hrsTra, tipoEmp: Entero
           cuoHr, sdoMen: Real
           desea: Carácter
   b. DO
 1. Imprimir Menu y solicitar tipo de
                empleado
            Tipos de empleado
          1. Empleado por horas
          2. Empleado asalariado
            Teclee tipo:
 2. Leer tipoEmp
 3. Solicitar Nombre, departamento y puesto
 4. Leer nomEmp, depto, puesto
 5. IF tipoEmp = 1 THEN
    a. Crear objeto
        EmpPorHoras objEmp = new
                              EmpPorHoras()
    b. Solicitar número de horas trabajadas
                y cuota por hora
    c. Leer hrsTra, cuoHr
    d. Establecer
  objEmp.establecerNombreEmp(nomEmp)
  objEmp.establecerDeptoEmp(depto)
  objEmp.establecerPuestoEmp(puesto)
  objEmp.establecerHorasTrab(hrsTra)
  objEmp.establecerCuotaHora(cuoHr)
    e. Calcular
        objEmp.calcularSueldoQnaHoras()
    f. Imprimir
            objEmp.obtenerNombreEmp()
            objEmp.obtenerDeptoEmp()
            objEmp.obtenerPuestoEmp()
            objEmp.obtenerSueldoQnaHoras()
 6. ELSE
    a. Crear objeto
       EmpAsalariado objEmp = new
                              EmpAsalariado()
    b. Solicitar sueldo mensual
    c. Leer sdoMen
    d. Establecer
objEmp.establecerNombreEmp(nomEmp)
objEmp.establecerDeptoEmp(depto)
objEmp.establecerPuestoEmp(puesto)
objEmp.establecerSueldoMensual(sdoMen)
    e. Calcular
        objEmp.calcularSueldoQnaAsal()
    f. Imprimir
            objEmp.obtenerNombreEmp()
            objEmp.obtenerDeptoEmp()
            objEmp.obtenerPuestoEmp()
            objEmp.obtenerSueldoQnaAsal()
 7. ENDIF
 8. Preguntar "¿Desea procesar otro
                              empleado(S/N)?"
 9. Leer desea
```

c. WHILE desea = “S”
d. Fin Método principal
Fin Clase EjecutaEmpleado
Fin

Explicación:

En el diagrama de clases, se esquematiza la estructura general de la solución. Se tiene la clase controlador EjecutaEmpleado, la cual utiliza al modelo, que está formado por tres clases jerarquizadas: La clase Empleado que es la superclase o clase principal; La clase EmpPorHoras que es una subclase que se deriva de la superclase Empleado; y La clase EmpAsalariado que es una subclase que se deriva de la superclase Empleado.

En el algoritmo se diseña la lógica de cada una de las clases usando pseudocódigo. En la clase Empleado se declaran los datos: nombreEmp, deptoEmp y puestoEmp; y los métodos establecerNombreEmp(), establecerDeptoEmp(), establecerPuestoEmp(), obtenerNombreEmp(), obtenerDeptoEmp() y obtenerPuestoEmp(); para establecer y obtener cada uno de los datos respectivamente. Empleado es la superclase que se usa para derivar subclases, a través del mecanismo de herencia; es por ello que a sus datos se le antepuso el símbolo # el cual indica que el dato es protegido (protected); los datos deben ser protegidos para que se puedan heredar.

En la clase EmpPorHoras que es una subclase que se deriva de Empleado, se declaran los datos: horasTrab, cuotaHora y suedoQnaHoras; y los métodos establecerHorasTrab(), establecerCuotaHora(), calcularSueldoQnaHoras() y obtenerSueldoQnaHoras(); para establecer las horas trabajadas, establecer la cuota por hora, calcular el sueldo quincenal del empleado por horas y obtener el valor del sueldo quincenal respectivamente. Al derivarse de Empleado, EmpPorHoras hereda los datos y métodos de Empleado a través del mecanismo de herencia.

En la clase EmpAsalariado que es una subclase que se deriva de Empleado, se declaran los datos: sueldoMensual y suedoQnaAsal; y los métodos establecerSueldoMensual(), calcularSueldoQnaAsal() y obtenerSueldoQnaAsal(); para establecer el sueldo mensual, calcular el sueldo quincenal del empleado asalariado y para obtener el sueldo quincenal respectivamente. Al derivarse de Empleado, EmpAsalariado hereda los datos y métodos de Empleado a través del mecanismo de herencia.

En la clase EjecutaEmpleado que es la clase controladora, es donde se establece la lógica que soluciona el problema utilizando las otras clases. Se declaran las variables necesarias para dar entrada a los datos. Se establece un ciclo DO…WHILE que permitirá procesar varios empleados. En el proceso de cada empleado, se debe indicar el tipo de empleado que es; y con el uso de un IF-THEN-ELSE se procesa de la forma que corresponda. Enseguida se genera el objeto correspondiente, luego se interactúa con el operador para que introduzca los datos, los cuales se leen y se llevan al objeto a través de los métodos setters; luego se calcula el sueldo llamando al método correspondiente; y finalmente se obtienen del objeto los datos que se van a imprimir, a través de los métodos getters.

Para conocer la metodología en detalle, se recomienda la lectura del libro [12].

5. Conclusión

Actualmente hay una tendencia a utilizar Java como primer lenguaje directamente con el concepto orientado a objetos, sin profundizar en el desarrollo de las bases lógicas de la programación, esto puede resultar muy dañino, porque vamos a generar programadores buenos para codificar usando lenguajes, pero sin bases lógicas; es decir, programadores que no saben programar.

Todo estudiante de sistemas, computación o informática debe aprender a programar orientado a objetos en lenguaje Java; pero para lograrlo, primero debe desarrollar las habilidades mentales lógicas necesarias; porque la programación es lógica y debe ser independiente de algún lenguaje de programación.

Ahora, la comunidad académica tiene a su disposición un libro [12] donde se presenta una metodología enfocada al desarrollo de las bases lógicas de la programación; esto es, para aprender a diseñar programas (algoritmos) orientados a objetos usando un pseudolenguaje (pseudocódigo). Dicha metodología, permite preparar a los estudiantes para que puedan aprender y comprender cualquier lenguaje orientado a objetos como Java, UML, etcétera.

Referencias

[1] Bell, D. y Parr, M., *Java para estudiantes Tercera edición*, México, Prentice Hall, 2003.

[2] Booch, G., *Análisis y diseño orientado a objetos con aplicaciones Segunda edición*, USA, México, Addison-Wesley/Díaz de Santos, 1996.

[3] Booch, G., Rumbaugh, J. y Jacobson, I., UML *El lenguaje unificado de modelado*, España, Addison Wesley, 1999.

[4] Ceballos, F.J., *Java 2 Curso de programación*, México, Alfaomega-Rama, 2000.

[5] Deitel, H.M. y Deitel, P.J., *Como programar en Java Quinta edición*, México, Pearson Prentice Hall, 2004.

[6] Horatman, C. S y Cornell, G,. *Core Java 2 Volume I Fundamentals 5th edition*. USA. The Sun Microsiystem Press Prentice Hall, 2000.

[7] Horton, I., *Beginning Java 2*, USA, Wrox Press Inc., 2000.

[8] Jacobson, I., Booch, G. y Rumbaugh, J., *UML El proceso unificado de desarrollo de software*, España, Addison Wesley, 2000.

[9] Joyanes, A.L., *Programación orientada a objetos Segunda edición*, España, Osborne Mc Graw Hill, 1998.

[10] Joyanes, A.L. y Fernández, A.M., *Java 2 Manual de programación*, España, Mc Graw Hill, 2001.

[11] Lemay, L. y Cadenhead, R., *Aprendiendo Java en 21 días*, México, Pearson Prentice Hall, 1999.

[12] López, R. L. *Metodología de la programación orientada a objetos,* Alfaomega (www.alfaomega.com.mx), México, 2006. ISBN 970-15-1173-5.

[13] Meyer, B., *Construcción de software orientado a objetos Segunda edición*, España, Prentice Hall, 1999.

[14] Rumbaugh, J., Jacobson, I. y Booch, G., *UML El lenguaje unificado de modelado. Manual de referencia*, España, Addison Wesley, 2000.

[15] Schildt, Herbert, *Fundamentos de programación en Java 2*, Colombia, Osborne Mc Graw Hill, 2002.

[16] Schildt, Herbert, *Java 2 Manual de referencia*, España, Osborne Mc Graw Hill, 2001.

[17] Wu, T.C.*, Introducción a la programación orientada a objetos en Java*, España, Mc Graw Hill, 2001.

[18] http://java.sun.com

Nota de referencia:

Una ponencia similar se presentó como Artículo Invitado en la 5ta. Conferencia Iberoamericana en Sistemas, Cibernética e Informática CISCI 2006, organizada por el International Institute of Informatics and Systemics, en Orlando, Florida, EE.UU.; y se está presentando esta ponencia en JENUI 2007, porque es para la difusión del contenido del libro referenciado.

Adoptando el Paradigma de la Programación Orientada a Atributos

Raúl Marticorena Sánchez, Carlos López Nozal, Carlos Pardo Aguilar
Área de Lenguajes y Sistemas Informáticos
EPS, Edificio C. C/ Francisco de Vitoria, s/n. CP 09006, Burgos
Universidad de Burgos
email: {rmartico, clopezno, cpardo}@ubu.es

Resumen

A la sombra del paradigma de la programación orientada a objetos, surgen en la actualidad nuevos paradigmas, relacionados en mayor o menor medida. Dentro de ellos se pueden enmarcar la programación orientada a aspectos y la programación orientada a atributos. Dada la actual imposición curricular del paradigma orientado a objetos, resulta difícil escapar a la inclusión de alguno de estos nuevos paradigmas en el campo docente.

En particular este trabajo muestra los primeros resultados de la introducción del concepto de programación orientada a atributos: desde una visión superficial, en la que se usan marcos de trabajo simples, a un nivel profundo con el desarrollo de bibliotecas basadas en dicho concepto, y finalmente con su empleo en entornos más avanzados.

El trabajo pretende mostrar las primeras experiencias realizadas en este sentido, y los resultados observados, para ir mejorando en su progresiva incorporación, y ayudando a futuras planificaciones de su integración y uso en asignaturas.

1. Planteamiento del Problema

Desde hace ya muchos años se ha estado impartiendo en asignaturas de programación el paradigma de orientación a objetos (POO). En nuestro caso particular, con diferentes lenguajes como C++, Eiffel o Java, pero básicamente centrándonos en las características generales: abstracción, encapsulación, herencia, polimofismo, etc [1].

Durante varios años, esta situación se ha mantenido estable, incluso con la incorporación del concepto de genericidad en lenguajes muy extendidos (Java) o pujantes (lenguajes de la familia .NET).

Sin embargo, el propio paradigma ha originado problemas que han motivado nuevas soluciones. El paradigma de la orientación a aspectos surge como necesidad a la captura de partes de un sistema que los modelos de programación habituales obligan a repartir en distintos módulos del sistema. Estos fragmentos, que afectan a distintos módulos, son llamados aspectos, y los problemas que solucionan se denominan problemas cruzados (crosscutting concerns).

Por otro lado se observa que algunos aspectos semánticos pueden ser resueltos por medio de herramientas, aprovechando ciertas declaraciones en el código. Esta opción, menos compleja, resuelve problemas similares a la programación orientada a aspectos, y recibe el nombre de programación orientada a atributos (POA) [2][3][4].

Los programadores pueden señalar o marcar elementos de sus programas (clases, variables de instancia, métodos, etc.) con metadatos, para indicar que mantienen una semántica específica de la aplicación o del dominio. Otras herramientas procesan dichos atributos para entrelazar la lógica de negocio con las cuestiones semánticas marcadas.

Como ejemplo, en esta línea de trabajo, surgieron soluciones como la incorporación del diseño por contrato con etiquetas, siguiendo reglas de documentación (JML [5]), o la generación de código intermedio (XDoclet para EJB [6]). Estas soluciones siguen la idea previa de usar preprocesadores: tomando cierta información introducida por el programador (comentarios especiales) se genera código adicional.

Dicha solución ha sido adoptada por las dos plataformas de desarrollo dominantes, .NET y Java, incorporando la idea de atributo en las correspondientes especificaciones de sus lenguajes.

Este concepto también permite llevar a cabo la implementación directa de un elemento de UML [7] tan usado como son los estereotipos y valores etiquetados. El actual vacío que existía entre la solución de diseño e implementación viene a ser cubierto de una manera natural.

En la medida de nuestro conocimiento, no existen experiencias docentes en este campo. Por lo tanto, no cabe sino afrontar el reto de introducir dicho paradigma como una pieza más de aquellas asignaturas en las que se ha venido utilizando la POO y el modelado UML como solución instrumental.

A continuación este trabajo presenta el contexto docente particular en la Sec. 2, identificando el problema de partida. En la Sec. 3, se expone el uso de atributos dentro de un marco de trabajo simple, basado en pruebas. En la Sec. 4 se definen, desarrollan y usan bibliotecas semánticas para la identificación de patrones de diseño y generación de código. Como última experiencia, en la Sec. 5, se utilizan atributos para el desarrollo avanzado de componentes distribuidos. Se finaliza en la Sec. 6 con las conclusiones obtenidas y líneas de trabajo futuro.

2. Contexto Particular

El paradigma de POO se introduce en segundo curso de la titulación técnica utilizando Java como lenguaje de soporte. A partir de ese momento, se ve reforzado su uso en asignaturas relacionadas con el desarrollo de estructuras de datos (2º curso), programación avanzada usando patrones de diseño y la web (3er curso), y en trabajos final de carrera. En paralelo se imparte en segundo curso UML como lenguaje de modelado y se utiliza intensivamente en asignaturas de cursos posteriores.

En el segundo ciclo de la ingeniería se plantean prácticas en C, C++ y Java, tomando como base los lenguajes más utilizados en el ciclo anterior. Tanto en primer como segundo ciclo se muestra una cierta evolución a implantar .NET, particularmente en trabajos final de carrera.

La experiencia actual es positiva y la demanda por parte de las empresas circundantes confirma la adecuación de dichos contenidos. Pero debido precisamente a la elección de Java y el posible cambio hacia .NET, junto con el hecho de que ambos en sus últimas especificaciones incluyan atributos como parte inherente del lenguaje, disparan la necesidad de su estudio.

Sin embargo, en la actualidad surge el problema, desde un punto de vista docente, de cómo y cuándo incluir esta nueva solución. Básicamente motivado por una adaptación a los nuevos paradigmas, entornos de desarrollo y por la posibilidad de utilizar este nuevo concepto de manera natural.

3. Primera Experiencia: Uso de Atributos

Durante los últimos años se ha venido incorporando la inclusión de pruebas [8] [9] de forma obligatoria en asignaturas de programación de nivel II.

Dichas prácticas se han venido realizando utilizando un framework de pruebas como JUnit [1] [10] que se ha convertido en un estándar de facto.

Las versiones utilizadas hasta la fecha se basaban en un diseño basado en patrones [13] y la utilización de reflexión e introspección para su correcto uso. A los alumnos se les presentaba dicho concepto brevemente, explicando el patrón *Compuesto* y las reglas de nombres a utilizar para que la reflexión basada en dichas normas funcionase correctamente.

Como ejemplo, el Algoritmo 1 muestra un breve extracto de código escrito siguiendo dichas reglas. El método `setUp` se ejecuta antes de cada test para realizar las inicializaciones pertinentes. Los métodos cuyo nombre empieza por la palabra `test`, contienen un test individual e independiente. En particular el último test comprueba que una excepción es lanzada en tiempo de ejecución siguiendo las especificaciones dadas.

Debido a la complejidad y nivel de la resolución interna de dichos conceptos, simplemente se realizaba una presentación superficial, dando una breve guía de programación.

Curiosamente, la evolución actual de este framework de pruebas, ha sufrido un cambio radical. Se abandona la solución reflexiva, los patrones de diseño quedan ocultos, y todo su funcionamiento se basa en el uso de atributos (anotaciones en Java [11]). Se ha optado por permitir compatibilidad hacia atrás y hacia delante

en los tests construidos, aunque se recomienda cambiar a la nueva solución.

```
public void setUp(){
}

public void testComprobar(){
}

public void testExcepcion(){
   try{
   // código que genera excepción
   fail(); // provocar fallo
   }
   catch(ExcepcionEsperada es){
       // ok
   }
}
```

Algoritmo 1. Codificación en JUnit 3.8.1

A continuación se muestra un breve extracto del código con atributos (ver Algoritmo 2), equivalente al mostrado previamente en el Algoritmo 1. La convención de nombres se abandona y se sustituyen por anotaciones precedidas con el símbolo @.

```
@Before
public void inicializar(){
}

@Test
public void comprobarPrecio(){
}

@Test(
expected=ExcepcionEsperada.class)
public void exception(){
// código que genera excepción
}
```

Algoritmo 2. Codificación en JUnit 4.2

Aunque la solución parece intuitiva, el cambio de paradigma puede desorientar inicialmente. De hecho, en algunos casos, el planteamiento es completamente distinto, como se puede deducir del último test, donde toda la semántica ha sido sustituida básicamente por una anotación con valores asociados.

A la hora de afrontar el cambio en la docencia de dichos contenidos se introduce el concepto, las anotaciones y su semántica asociada, ilustrando su uso sobre una colección de tests de ejemplo. Se proporciona a los alumnos una plantilla base a partir de la cual construir nuevos tests, y se les aportan ejemplos de utilización.

La complejidad de la ejecución dentro del framework, así como el procesado de dichos atributos por el propio entorno de ejecución, se omiten, incidiendo en la semántica qué implica el atributo, pero no el cómo se resuelve internamente el problema de su procesado.

3.1. Práctica Propuesta

Se plantea una práctica obligatoria donde se propone un módulo a probar, utilizando pruebas de caja negra. Para ello los alumnos deben implementar el correspondiente módulo de prueba con JUnit, siguiendo las directrices marcadas anteriormente, utilizando obligatoriamente la nueva implementación del framework.

3.2. Evaluación

Se solicita a los alumnos que rellenen una encuesta en relación con dicha práctica, centrándose en aquellos alumnos repetidores que realizaron la práctica en años anteriores con la versión previa, frente a la solución presentada en este curso.

Las preguntas realizadas y resultados obtenidos, recogen la opinión sobre 38 alumnos (un 50% aproximado de la población total de alumnos que realizaron la práctica). En las Tabla 1 y Tabla 2 se presentan las preguntas realizadas y la frecuencia de las puntuaciones recibidas.

De estos resultados se pueden obtener algunas conclusiones previas. En relación a la primera pregunta, se observa que existe una cierta tendencia a solicitar un número de horas en el temario para explicar con mayor detalle el concepto de atributo. Inicialmente el concepto se usa, pero no parece estar del todo bien definido y aclarado por parte del docente.

Curiosamente, a partir de la segunda pregunta, se desprende que no existe esa misma tendencia en el alumnado en cuanto a comprender las ventajas del nuevo concepto y su uso. El interés del aprendizaje de usar atributos en el código, no ha quedado plasmado con esta primera experiencia.

Desde el punto de vista de alumnos que han cursado la asignatura en años previos, y que

realizaron la práctica con la antigua versión del framework, se les pregunta sobre el grado de dificultad ante la nueva versión y su preferencia al utilizar una u otra (ver Tabla 2).

Pregunta \ Puntuación	1	2	3	4	5
Es necesario dedicar alguna hora de teoría a explicar el concepto de anotaciones en Java y su uso	0	4	7	18	9
Es conveniente aprender el concepto de programación orientada a atributos usando anotaciones	0	2	20	10	5

Tabla 1. Evaluación de anotaciones

En este caso las puntuaciones reflejan el grado de dificultad, siendo una puntuación de 1 muy fácil y 5 muy difícil:

Pregunta \ Puntuación	1	2	3	4	5
Considera que el cambio realizado en la nueva versión de JUnit, con el uso de anotaciones, hace que la práctica respecto a otros años, en dificultad es:	1	8	6	5	1
Versión JUnit elegida	3.8.1		4.2		
Si se dejase la posibilidad de realizar la práctica con una de las dos versiones (3.8.1 o 4.2) ¿cuál elegirías?:	9		12		

Tabla 2. Valoración de dificultad y selección

Se observa una tendencia general a considerar más fácil el uso de la nueva versión, aunque existe un cierto número de respuestas que indican todavía dificultades y reticencias a aceptarla.

Es más curioso todavía, si se observa que casi existe un empate ante la posibilidad de realizar las prácticas con la antigua versión, incluso aunque se apoye en conceptos avanzados como la introspección y patrones de diseño, conceptos que los alumnos de estos cursos todavía no han visto.

4. Segunda Experiencia: Definición de Atributos

Esta experiencia ha sido llevada a cabo en la asignatura de Diseño y Mantenimiento del Software II, impartida como materia troncal en quinto curso de Ingeniería en Informática. El contenido temático de esta asignatura, se basa en el estudio de los conceptos y relaciones existentes entre atributos del software, métricas de producto, patrones de diseño, pruebas y refactorizaciones.

En la asignatura se ha llevado a la práctica dos experiencias relacionadas con programación orientada a atributos. Por un lado, definir los atributos necesarios que permitan reflejar la semántica de aplicación de patrones de diseño, en un sistema software. Por otro lado, aprender a traducir a código los conceptos de modelado UML: valores etiquetados y estereotipos.

4.1. Práctica Propuesta: Semántica de Patrones de Diseño

Antes de proponer la práctica a los alumnos, se parte de una sesión de prácticas guiadas donde se muestra a los alumnos la importancia que tiene el conocimiento previo de los patrones de diseño en la fase de mantenimiento. En esta sesión de entrenamiento, los alumnos se enfrentan de manera supervisada al conocimiento del diseño de tres componentes software.

El problema que ocurre en la mayoría de los casos, es que parte de los componentes no mantienen la documentación de diseño. Únicamente se proporciona código ejecutable o código fuente. Se les muestra que la identificación de patrones puede hacerse de manera automática, a través de herramientas, o a partir de la documentación de diseño.

Para este cometido, en la misma sesión se enseña a manejar la herramienta WOP [14]. Esta herramienta permite identificar patrones de diseño a partir de un repositorio de definición de éstos, basado en su estructura.

El componente JUnit lleva asociada documentación de diseño basada en patrones. Por tanto, es un candidato perfecto para probar la bondad de la herramienta en su funcionalidad de identificación. En la Tabla 3 se muestra el resultado de la comparativa, la primera columna indica los patrones que son aplicados. Esta información ha sido extraída de la documentación

de diseño del componente. En la segunda columna los patrones disponibles en la herramienta para su identificación. Y en la columna tercera los patrones que se logran identificar con la herramienta.

PD en documentación de diseño JUnit	WOP Repositorio	WOP Identificación
Comando	NO	NO
Método Plantilla	SI	NO
Recolección de parámetros	NO	NO
Adaptador (clase)	SI	NO
Compuesto	SI	SI

Tabla 3. Evaluación de WOP

Una vez vista la dificultad de la herramienta para extraer la información de diseño a partir de la definición estructural externa de un patrón, se propone una nueva alternativa de trabajo: incluir la semántica asociada a la aplicación de los patrones de diseño en el propio código. Se toma como base la notación definida por uno de los autores de GoF y recogida en [15]. En esta propuesta, se toma la representación de los patrones de diseño basada en estereotipos y valores etiquetados UML [7], en lugar de colaboraciones. La razón para usar esta alternativa es el mayor contenido semántico que puede aportar. En la Figura 1 y Figura 2 se puede observar las diferentes representaciones UML de una aplicación del patrón *Singleton* [13] sobre la clase `GestorImpresion`.

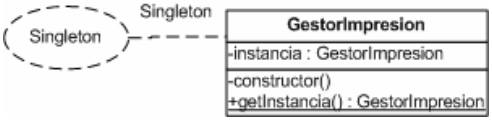


Figura 1. Representación de patrones con colaboraciones

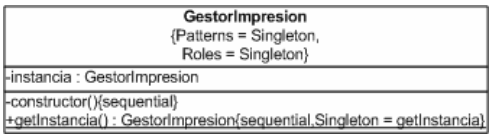


Figura 2. Representación de patrones con valores etiquetados y estereotipos

A partir de la semántica expuesta, se propone definir una anotación Java [11] que la represente. En el Algoritmo 3 se muestra el código de una posible anotación java para representar la semántica de aplicación de patrones de diseño. El tipo `PatternName` se corresponde con una enumeración de cadenas con los nombres de los patrones tratados.

```
@Documented
@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.TYPE)
public @interface
 PatternAnnotation {
   PatternName[] patternName();
   String participantName();

}
```

Algoritmo 3. Anotación Java para patrones

4.2. Práctica Propuesta: Traductor UML a Java

La segunda experiencia se basa en añadir una nueva funcionalidad a un software existente, en concreto a una herramienta CASE UML desarrollada en varios proyectos final de carrera. Se pide construir un traductor de UML a código java que permita tratar los conceptos de estereotipos y valores etiquetados. En el desarrollo se deben aplicar patrones de diseño, refactorizaciones, pruebas y métricas de producto.

A modo de ejemplo, en la Figura 3 se muestra una clase generada con la herramienta (`PrinterPool`) donde se aplica el patrón de diseño *Singleton*. La funcionalidad pedida consiste en leer la información de las instancias del metamodelo UML y transformarlas a código java. En el Algoritmo 4 se puede observar el resultado de salida esperada.

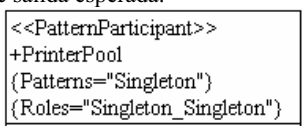


Figura 3. Entrada caso de prueba: mapeo estereotipo y valores etiquetados

```
import java.lang.annotation.*;

@Documented
@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.TYPE)
public @interface
 PatternParticipant {
```

```
  String[] Patterns();
  String[] Roles();
}

@PatternParticipant (
 Patterns={"Singleton"},
 Roles={"Singleton_Singleton"})
public class PrinterPool{
}
```

Algoritmo 4. Salida caso de prueba: mapeo estereotipo y valores etiquetados

4.3. Evaluación

La encuesta fue realizada a 14 alumnos sobre un total de 31 alumnos matriculados en la asignatura. En la Tabla 4 se observan un resumen de los resultados.

Pregunta \ Puntuación	1	2	3	4	5
Piensas que la representación de patrones de diseño mediante anotaciones java o valores etiquetados clarifica tus diseños	0	2	5	7	0
La identificación de PD mejora la comprensión preliminar de un diseño	1	4	9	1	4

Tabla 4. Evaluación de patrones de diseño y anotaciones

Como conclusión, los alumnos muestran un interés elevado en añadir contenido semántico a su código para incorporar información explícita sobre sus diseños. Indirectamente, ven la mejor adecuación de la programación orientada a atributos para poder conseguir este propósito. Aunque como contradicción, no queda tan clara la utilidad de la semántica en la identificación de patrones.

5. Tercera Experiencia: Uso Avanzado y Evaluación de Solución con Atributos

En la asignatura optativa de quinto curso de Sistemas Distribuidos [16][17][18], se proponen prácticas utilizando componentes distribuidos Enterprise JavaBeans. Mientras que en cursos anteriores se ha venido utilizando una herramienta que sigue la especificación EJB 2.1 [19], basada en interfaces y ensamblado basado en ficheros de despliegue (utilizando XML), en el presente curso se ha avanzado hacia la siguiente especificación EJB 3.0 [20], dónde desaparece la necesidad de ficheros intermedios utilizando atributos.

Como ejemplo, mientras en la especificación 2.1 es necesario trabajar con un número de 3 a 5 clases/interfaces Java junto con 1 o 2 ficheros de despliegue XML, en la nueva especificación se puede ver reducido a una o dos clases/interfaces Java. En el Algoritmo 5 se puede ver un esquema de declaración de un EJB de mensaje con la especificación 2.1. Se omite el código de acceso a la cola de mensajes, por simplificación, y los ficheros adicionales para el despliegue del EJB.

```
public class MdbBean
       implements
   MessageDrivenBean,
   MessageListener {
   // lógica de negocio de acceso
   // a la cola de mensajes
}
```

Algoritmo 5. Esquema de EJB MDB 2.1

```
@MessageDriven(activationConfig =
{
@ActivationConfigProperty(propert
yName = "destination",
propertyValue = "ColaEjemplo"),

@ActivationConfigProperty(propert
yName = "destinationType",
propertyValue =
"javax.jms.Queue")
    }
)
public   class   MessageDrivenBean
implements MessageListener {
   // cuerpo de la clase
}
```

Algoritmo 6. Esquema de EJB MDB 3.0

Por otro lado, en la especificación EJB 3.0 [20], el uso de atributos (anotaciones) sustituye al código que enlaza el EJB con la cola de mensajes (inyección de recursos) y declaraciones adicionales. Se elimina la necesidad de ficheros externos XML y se simplifica el código ejecutable del EJB a partir de una solución declarativa que recoge dicha semántica. En el Algoritmo 6, se

muestra el código equivalente, utilizando atributos.

Dado que en este caso, el alumno sí dispone de una fuerte base, al tratarse de alumnos de último curso de la ingeniería, se pide una implementación siguiendo ambas especificaciones, pero además solicitando una evaluación comparativa entre ambas.

5.1. Enunciado de la Práctica

La práctica plantea el desarrollo de un Enterprise JavaBean orientado a mensajes (Message Driven Bean - MDB) para captar mensajes asíncronos. Los mensajes se envían a un tema de suscripción conteniendo un fichero a replicar en varios suscriptores, guardando un registro de la operación en base de datos.

El MDB debe ser desarrollado siguiendo ambas especificaciones:

- EJB 2.1 sobre JOnAS
- EJB 3.0 sobre EasyBeans (como plugin de JOnAS)

Se debe analizar y evaluar las correspondientes facilidades y dificultades asociadas a cada una de las soluciones. Para el desarrollo de ambos EJBs se les facilita código de ejemplo, que puede ser utilizado como plantilla.

5.2. Evaluación

Junto con el propio ejercicio práctico se ha realizado una evaluación docente de la asignatura donde se les ha planteado preguntas similares a las planteadas en primeros cursos. Al tratarse de una optativa el número de encuestados es mucho menor, recogiendo los resultados de aquellos alumnos que han seguido la asignatura con asiduidad y que han presentado las prácticas obligatorias. En la Tabla 5 se muestran los resultados recogidos.

Aunque el número de encuestados es pequeño, sí que permite recoger algunas conclusiones parciales. Al igual que ocurría en primeros cursos, los alumnos creen adecuado que se introduzca el concepto de atributos previamente.

Por otro lado un alto porcentaje ve conveniente aprender dicho concepto. En este caso particular los alumnos sí que han realizado el mismo desarrollo con ambas soluciones y la comparativa es objetiva.

Respecto a la última pregunta, en el caso particular del desarrollo de EJB con la especificación 2.1 o 3.0, parece que existe una tendencia generalizada a admitir que el uso de atributos les ha facilitado el desarrollo de la misma, pese a lo novedoso de la solución.

Pregunta \ Puntuación	1	2	3	4	5
Es necesario dedicar alguna hora de teoría a explicar el concepto de anotaciones en Java y su uso en esta asignatura	1	4	2	2	0
Es conveniente aprender el concepto de programación orientada a atributos usando anotaciones	1	2	3	3	0
	Muy fácil	Fácil	Igual	Difícil	Muy difícil
Consideras que las prácticas con anotaciones son:	0	6	2	1	0

Tabla 5. Evaluación docente en sistemas distribuidos

6. Conclusiones y Líneas de Trabajo Futura

Aunque se han mostrado conclusiones particulares sobre cada una de las experiencias, se da a continuación una visión global de nuestra primera experiencia con la inclusión de la programación orientada a atributos, durante el presente curso.

Se puede establecer la necesidad inicial de incluir en los temarios de programación orientada a objetos de primeros cursos, alguna sesión de teoría y práctica al concepto de atributo. La simple utilización del concepto, deja a los alumnos con una cierta indecisión y dudas al respecto de su uso y funcionamiento. Para apoyar esta idea se cree adecuado coordinar la inclusión del concepto en asignaturas de ingeniería del software.

También, a la vista de los resultados, parece que el realizar prácticas comparativas entre soluciones orientadas y no orientadas a atributos acaba aclarando el papel concreto de dicho paradigma para el alumnado.

En este punto, puede ser importante el resaltar la importancia de su uso práctico en la actualidad, y su posible entronque con el concepto de la

programación orientada a aspectos en futuros trabajos.

Esto permitiría en posteriores cursos donde se utiliza dicho paradigma, el poder focalizar sobre la semántica particular que cada conjunto de atributos recoge, y las particularidades dentro de cada framework o plataforma de desarrollo.

En próximos cursos se podrá comprobar el efecto de haber introducido el paradigma en asignaturas previas, permitiendo agilizar el proceso de aprendizaje o profundizar sobre cuestiones más avanzadas.

Como reto futuro, parece obvio el comenzar a introducir en un modo similar el concepto de aspecto, y su relación con el paradigma de programación orientada a objetos y a atributos. A partir de estos primeros resultados, se puede empezar a prever los problemas y soluciones que habrá que plantear y resolver ante la introducción de nuevos paradigmas.

Referencias

[1] Meyer, B. Construcción de Software Orientado a Objetos 2ª Edición. Prentice Hall, 1998.

[2] Pawlak, R. Spoon: Annotation –Driven Program Transformation – The AOP Case. 1st International Middleware Workshop on Aspect Oriented Middleware Development (AOMD) ser. AICPS. Vol. 118.Grenoble, France ACM, Nov. 2005. pp. 1—6

[3] Wada, H. and Suzuki, J. Modeling Turnpike Frontend System: A Model-Driven Development Framework Leveraging UML Metamodeling and Attribute-Oriented Programming. In 8th International Conference on Model Driven Engineering Language and Systems (MoDELS), ser. LNCS, vol. 3713. Montego Bay, Jamaica: Springer, Oct. 2005, pp. 584—600

[4] Eichberg, M., Schäfer, T., and Mezini, M. Using Annotations oto Check Structural Properties of Classes. In 8th International Conference on Fundamental Approaches to Software Engineering (FASE). ser. LNCS, nº 3442. Edinburgh UK: Springer, Apr. 2005, pp. 237—252.

[5] Gary T. Leavens, Albert L. Baker, and Clyde Ruby. Preliminary Design of JML: A Behavioral Interface Specification Language for Java. ACM SIGSOFT Software Engineering Notes, 31(3):1-38, March 2006.

[6] Walls, C. & Richards,_N. XDoclet in Action. Manning Publications. 2003.

[7] OMG Object Management Group. Unified Modeling Language: Superstructure version 2.0. Revised Final Adopted Specification (ptc/04-10-02) October 8, 2004. Disponible en http://www.uml.org/.

[8] Myers, G.J. El Arte de Probar el Software. El Atenero, 1984.

[9] Binder, R.V. Testing Object-Oriented Systems. Models, patterns, and tools. Addison Wesley, 2000.

[10] Link, J. et al. Unit Testing in Java. How Tests Drive the Code. Morgan Kaufmann, 2003.

[11] McLaughlin, B. & Flanagan D. (2004). Java 1.5 Tiger: A Developer´s Notebok. Ed O'REILLY.

[12] JUnit. Testing Resources for Extreme Programming. http://www.junit.org. Última visita 26 de enero de 2004.

[13] Gamma, E. Helm, R., Jonhson R., y Vlissides, J. Patrones de Diseño. Addison Wesley, 2003.

[14] WOP Web Of Patterns. Ontology documents homepage. http://www-ist.massey.ac.nz/wop/

[15] Dong, J. (2002) UML Extensions for Design Pattern Compositions. Journal of Object Technology, vol.1, nº5, Noviembre-Diciembre de 2002, páginas 151-163. Disponible en http://www.jot.fm/issues/issue_2002_11/article3

[16] Coulouris, G. , Dollimore, J. y Kindberg, T. *Sistemas Distribuidos. Conceptos y Diseño.* Ed. Addison-Wesley. 3ªEdición, 2001.

[17] Liu, M.L. *Computación Distribuida. Fundamentos y Aplicaciones.* Ed. Addison-Wesley, 2004.

[18] Tanenbaum, A. & Van Steen, M. *Distributed Systems.* 1st Edition. Prentice-Hall. 2002.

[19] DeMichiel, L.G. Enterprise JavaBeansTM Specification, Version 2.1. Sun Microsystems. FCS. November, 2003.

[20] DeMichiel, L.G. & Keith, M. JSR 220: Enterprise JavaBeansTM,Version 3.0. EJB Core Contracts and Requirements. Sun Microsystems. May, 2006.

La práctica monitorizada: una herramienta válida en el aprendizaje activo de la asignatura Estructuras de la Información

Gabriel Fiol Roig, Catalina Fiol Roig, Margarita Miró Julià
Dpto. de Ciencias Matemáticas e Informática
Universidad de las Islas Baleares
Ctra. De Valldemossa, km 7,5, Ed. A. Turmeda, 07122 Palma de Mallorca
biel.fiol@uib.es, cfiolroig@hotmail.com, margaret.miro@uib.es

Resumen

La motivación de los alumnos es la base para su implicación en un proceso de aprendizaje activo y continuo, el cual a la vez constituye un estímulo de la misma, en un ciclo que conduce inexorablemente al éxito de la actividad docente.

En este artículo se presenta el diseño e implantación durante el curso 2005-06 de una práctica monitorizada, cimentada en el enfoque del *aprendizaje basado en proyectos –ABP-*, como herramienta de aprendizaje activo y continuo de la asignatura *Estructuras de la Información*, de las Ingenierías Técnicas en Informática de Gestión e Informática de Sistemas, cuya duración se extiende a lo largo del transcurso de la asignatura.

La organización de la práctica se basa en una serie de etapas cuyo desarrollo permite al alumno, por una parte, lograr las competencias específicas propias de la materia, tales como aclarar, consolidar e implantar los contenidos de la asignatura, y por otra promover una serie de competencias transversales de forma que se alcancen los objetivos formativos, todo ello a través de la ejercitación de un diversificado conjunto de actividades de aprendizaje convenientemente monitorizadas.

Los resultados de la implantación de la práctica monitorizada en el contexto de adaptación de la asignatura a las directrices del EEES han sido muy satisfactorios, tanto por lo que respecta al alto grado de éxito escolar como al nivel de conocimientos alcanzado.

1. Motivación

El EEES ofrece un nuevo escenario docente cuyo eje central gira en torno a la figura del estudiante, a diferencia del escenario actual, donde el protagonista es el profesor.

Dos de los aspectos del nuevo marco que más interés suscitan son el cambio de metodología docente, dirigida hacia un modelo de enseñanza centrado en el aprendizaje activo del alumno [7], y la medida con la que se valorará la actividad académica, centrada en el esfuerzo de los estudiantes para superar las diferentes materias. Ambas razones conducen ineludiblemente hacia una mejora de la calidad de la enseñanza.

En este contexto, un grupo de profesores universitarios decidimos emprender la tarea de mejora de la calidad de la enseñanza en el ámbito de la asignatura Estructuras de la Información –*E.I.*-. Dicha tarea se llevó a cabo a través de un proyecto [5] de mejora de la calidad de la docencia mediante la adaptación de la asignatura al sistema de créditos europeo –ECTS-.

El presente artículo describe parte del plan de trabajo del mencionado proyecto así como los resultados de la correspondiente implantación, centrándose en las cuestiones que favorecen la implicación activa del estudiante en su proceso de formación. En particular, se presenta el diseño y los resultados de la implantación de una práctica monitorizada, como una herramienta de aprendizaje activo y continuo a lo largo del transcurso de la asignatura, cuyos resultados han mostrado su eficacia respecto a la motivación de los alumnos y consecuentemente su éxito escolar, y al nivel de calidad del aprendizaje alcanzado por aquellos que han superado el proceso.

El artículo se organiza como sigue: en la sección 2 se ofrece una descripción de la asignatura E.I.; en la sección 3 se presentan la filosofía de la práctica, cimentada en el aprendizaje basado en proyectos, y el diseño de la

misma a partir de la diversificación de un grupo de actividades de aprendizaje y los mecanismos de monitorización para llevarlas a cabo; en la sección 4 se muestran los resultados de la implantación del proyecto, que afectan a los alumnos y también al profesor, finalmente se sacan las conclusiones más relevantes sobre la experiencia.

2. La asignatura Estructuras de la Información

E.I. es una asignatura troncal de las Ingenierías Técnicas en Informática de Gestión e Informática de Sistemas, con una carga lectiva de 60 horas y un esfuerzo global por parte del alumno de 7,24 créditos ECTS [5], impartiéndose en el segundo cuatrimestre a razón de cuatro horas a la semana, estructuradas en dos clases de dos horas cada una.

La tabla 1 sintetiza el marco docente en el que se imparte la asignatura.

Asignatura	Estructuras de la Información
Estudios	Ingenierias Técnicas en Informática de Gestión e Informática de Sistemas
Centro	Escuela Politécnica Superior
Curso	2º
Duración	Cuatrimestral -2º cuatrimestre-
Carácter	Troncal
Créditos ECTS	7.24 -3,51 teóricos + 3,73 prácticos-
Nº de alumnos	195
Nº de grupos de teoría	2
Descriptores BOE	Tipos abstractos de datos. Estructuras de datos y algoritmos de manipulación.

Tabla 1. Marco docente de la asignatura

El cómputo de los créditos ECTS que aparece en la tabla forma parte del resultado de adaptación de la asignatura a las directrices del EEES. Se ha considerado que un crédito ECTS corresponde a 25 horas de dedicación del estudiante.

El programa de la asignatura está formado por cinco temas y una práctica de tamaño medio-grande que se desarrollará durante todo el transcurso de la asignatura. La tabla 2 muestra los descriptores de los contenidos relativos a cada uno de los temas.

La exposición temporal de los cinco temas se realiza de acuerdo con el orden en que aparecen listados en la tabla 2. El inicio de las tareas de desarrollo de la práctica tiene lugar en la segunda mitad de la exposición del tema 3, y se prolonga durante el resto del curso, con especial insistencia durante el transcurso del tema 4, que constituye, de hecho, un tema de soporte a la misma.

Tema 1	Mecanismos de agregación de datos
Tema 2	Programación con tipos abstractos de datos
Tema 3	Tipos abstractos de datos lineales
Tema 4	Diseño de tipos abstractos de datos a gran escala
Tema 5	Tipos abstractos de datos no lineales
Práctica	Diseño modular guiado por tipos abstractos de datos

Tabla 2. Programa de la asignatura

La evaluación de la asignatura es una de las tareas más importantes y a la vez delicadas del proceso docente, pues entre otros aspectos, constituye un importante factor motivador/ desmotivador del interés de los estudiantes. Para fomentar la motivación a través de la evaluación hay que tener presentes dos cuestiones importantes: primero, la evaluación debe ser objetiva, esto es, debe ajustarse a los méritos realizados por los estudiantes; segundo, la evaluación debe considerar el conjunto de los méritos de manera proporcional al esfuerzo desarrollado por los estudiantes en cada una de las actividades en su intento por alcanzar los objetivos de formación.

En este contexto, se decidió un plan de evaluación acorde con los resultados obtenidos en [5], donde, entre otros aspectos, se llevó a cabo el cómputo del esfuerzo de las diferentes actividades de aprendizaje de la asignatura. El resultado final mostro la necesidad de implantar E.I. a través de un plan de trabajo de 7,24 créditos, de los cuales 3,73 corresponden a actividades de aprendizaje relacionadas con la práctica monitorizada y el resto, 3,51, al resto de actividades.

La equidad entre los créditos prácticos y los teóricos condujo al diseño de un proyecto de evaluación equilibrado: un examen teórico conceptual y el resultado de la elaboración de la práctica, con una proporcionalidad de un 50% en cada una de las partes.

3. La práctica monitorizada

Tradicionalmente, las prácticas en la asignatura E.I. han sido consideradas como un simple complemento de los contenidos teóricos del temario, cuyo valor en el peso de la evaluación final de la asignatura se reducía –aspecto todavía vigente en numerosos programas de asignaturas- a la simple obligatoriedad de su desarrollo, sin computar en la nota final, o con un cómputo muy bajo en relación al peso del examen –son típicas las situaciones donde el peso del examen y el de la práctica oscilan aproximadamente entre el 80% y el 20% respectivamente.

La incorporación del concepto de *tipo abstracto de datos –TAD-* en el ámbito de la programación de ordenadores, y más en concreto, en el área de las estructuras de datos, dio un gran impulso al desarrollo de potentes metodologías de diseño de programas a partir de organizaciones abstractas de datos *–Data Driven Methodologies-*. A partir de ahí, el concepto de estructura de datos fue más allá del de las simples organizaciones de datos, como las listas, los árboles,… abarcando estructuras tan complejas como se requiera, cuyo diseño exige unas pautas metodológicas bien definidas.

La implantación de una estructura de datos compleja en términos de un tipo abstracto constituye un proceso arduo, en el que se integran conocimientos adquiridos con conocimientos nuevos; que abarca, por una parte, cuestiones metodológicas de diseño, y por otra, cuestiones técnicas específicas de la materia, todo lo cual supone el desarrollo y la puesta en práctica de un ámplio abanico de habilidades de aprendizaje en grupo e individuales, tales como el trabajo cooperativo o colaborativo, el trabajo autónomo, el autoaprendizaje, la autocrítica, la autoevaluación y actividades intelectuales de nivel alto según la taxonomía de Bloom [1], de manera especial la abstracción y la resolución de problemas por aplicación de diferentes técnicas.

3.1. El aprendizaje basado en proyectos

El aprendizaje basado en proyectos –ABP- [6] constituye una propuesta metodológica de enseñanza-aprendizaje que permite explotar una gran variedad de habilidades, tanto en grupo como individuales, en la que la formación del alumno se basa en una actitud permanentemente activa.

Las competencias generales fomentadas por el ABP encajan claramente con las que persigue el desarrollo de la práctica, a destacar:

- Desarrollo de la capacidad para resolver problemas a través de la combinación de diferentes técnicas, con especial ejercicio de la abstracción conceptual. Se trata del desarrollo de habilidades intelectuales de nivel alto según la taxonomía de Bloom [1].
- Consolidar e integrar conocimientos de campos diferentes. Cabe matizar en este sentido que la práctica que se propone se ajusta más al desarrollo de un problema concreto de tamaño medio-grande, en el que deben integrarse conocimientos de diferentes campos del área de la programación, que de un proyecto multidisciplinar. Por lo demás, el reto formativo que persigue el ABP es perfectamente extrapolable al que se pretende con el desarrollo de la práctica.
- Desarrollar la capacidad de autoaprendizaje, donde el alumno debe reforzar su actitud autodidacta en la búsqueda de soluciones, quedando relegado el profesor a una fuente más de conocimiento. El autoaprendizaje constituye un excelente ingrediente en el logro de diversas completencias generales.
- Desarrollar la capacidad de razonamiento crítico, adquiriendo conciencia de las limitaciones de la solución encontrada.
- Capacidad para planificar y organizar el estudio a través del trabajo en equipo. La organización y la planificación resultan imprescindibles para un desarrollo eficiente de la práctica, de manera especial cuando se lleva a cabo en grupo.
- Capacidad de comunicación oral y escrita. Tanto el desarrollo como la evaluación de la práctica exigen la ejercitación de habilidades de comunicación oral. Además, en la evaluación se exige la entrega de un informe escrito detallado, que actúa como un plano de la práctica que sirve de guía para su interpretación y posterior valoración.
- Capacidad de análisis y síntesis. El desarrollo de la práctica se caracteriza por la existencia de numerosas soluciones alternativas en la mayoría de sus etapas, en las que el alumno deberá ejercer, por una parte, su capacidad de análisis con objeto de determinar la bondad de la solución encontrada en cada etapa, y por

otra, su capacidad de síntesis que le permita vislumbrar las propiedades derivadas del ensamblaje de las soluciones de las diferentes etapas.

3.2. Diseño de la práctica

El diseño de la práctica se cimienta en la filosofía del aprendizaje basado en proyectos, adaptado a las características del contexto por lo que respecta a la integración de conocimientos, tal como se ha mencionado en la sección 3.1, cuyo contenido se basa en el desarrollo de un proyecto de programación de tamaño medio-grande centrado en torno al diseño e implementación de una compleja organización de datos.

Etapa I	Descripción de las características generales de las prácticas –tipo de problema, proceso de desarrollo, características de la memória, …-
Etapa II	Discusión y análisis del enunciado del problema objeto de la práctica.
Etapa III	Diseño y especificación formal del modelo y las operaciones del tipo abstracto de datos –TAD- del problema.
Etapa IV	Implementación abstracta de los requerimientos del problema mediante las operaciones diseñadas en la especificación formal.
Etapa V	Implementación del modelo y las operaciones de la especificación formal. Deben desarrollarse dos implementaciones diferentes de los elementos de la especificación formal.
Etapa VI	Desarrollo del programa fuente.

Tabla 3. Etapas del desarrollo de la práctica

La realización del mencionado proyecto se lleva a cabo en grupos de dos personas, a través de seis etapas bien definidas, en el orden en que se ilustran en la tabla 3, cuya actividad se lleva a cabo en paralelo con el desarrollo del resto de los aspectos conceptuales de la asignatura, lo cual exige una meticulosa coordinación.

3.3. Actividades de aprendizaje para el desarrollo de la práctica

En realidad, la tabla 3 constituye un auténtico plan de trabajo donde el alumno se ve necesariamente implicado de forma activa y en cuyo desarrollo se ejercitan diversas actividades de aprendizaje.

En la tabla 4 se esquematiza el conjunto de actividades de aprendizaje para el desarrollo de la práctica, junto con las características del grupo en el que se llevarán a cabo, el esfuerzo de cada actividad para cada alumno, los objetivos específicos de cada etapa y la necesidad de coordinación con los temas del programa de la asignatura ilustrados en la tabla 2.

- Las *clases magistrales*. Se usan para exponer aquellos aspectos, teóricos y prácticos, de carácter general sobre la práctica, en los que, por una parte, se pretende que el alumno tome conciencia del carácter de la metodología de programación dirigida por datos, y por otra se le instruye en el proceso de diseño de tipos abstractos a gran escala. Es importante despertar el espíritu de autocrítica del alumno a través de las tareas de análisis y síntesis, pues el abanico de soluciones es amplio y variado, y la solución que va a encontrar tendrá sus limitaciones. Al tratarse de proyectos de tamaño razonable, debe insistirse en la necesidad de una organización y planificación del trabajo y en la importancia de adquirir buenas habilidades de trabajo en equipo. En la evaluación de cada fase de la práctica se considera, por un lado, la parte técnica, y por otro se exige una rigurosa y completa memoria de la misma. En este sentido se instruye a los alumnos sobre las características de la memoria, lo cual exige la ejercitación de las habilidades de comunicación escrita.
- Los *seminarios*. Se desarrollan en grupos de tamaño medio –unas 35 personas a lo sumo-, participando únicamente aquellos grupos que deben realizar la misma práctica, siendo su objetivo la discusión del enunciado de la misma con objeto de aclarar cualquier duda y delimitar su alcance. Además, en el seminario se incide especialmente en cuestiones de organización y planificación de la práctica y en el desarrollo del trabajo en equipo. El seminario constituye el paso previo al inicio de las auténticas etapas de desarrollo.
- El *trabajo autónomo, el trabajo cooperativo* y la *tutorización + evaluación*. El trabajo autónomo y el trabajo cooperativo constituyen la principal actividad de desarrollo de la práctica, realizándose a través de las etapas III, IV, V y VI. En el trabajo cooperativo los

miembros del grupo deben, en una primera etapa, organizar y planificar las tareas con objeto de determinar su distribución en aras a un eficiente desarrollo autónomo, pasando posteriormente a la discusión de los resultados individuales y a su ensamblaje. Durante el desarrollo de las etapas III, IV y V, el trabajo autónomo se combina con las actividades de tutorización, cuyo objetivo es doble: por una parte se establece un control sobre el trabajo realizado en la correspondiente etapa, lo cual obliga a los alumnos a poner en práctica sus

Etapas de la práctica	Actividades de aprendizaje	Duración del trabajo del alumno	Objetivos	Coordinación con el temario
Etapa I	Clase magistral -Teoria- Grupo grande	1 h.	-Describir las características generales del desarrollo de un proyecto de programación dirigida por datos a gran escala.	Tema 2
		3/4h	-Exponer las características específicas para el desarrollo de la práctica. Especificar las directrices para llevar a cabo la memoria de la práctica	Tema 4
Etapa II	Seminario -Problemas y Prácticas- Grupo mediano	1h	-Discusión y comentarios sobre el enunciado de cada uno de los problemas planteados para la práctica.	Tema 4
Etapa III	Clase magistral -Prácticas- Grupo grande	8h	-Diseño y especificación formal de tipos abstractos de datos a gran escala.	Tema 4
Etapa III	Tutorización + Evaluación Grupo pequeño (2 p.)	1h 15 min [Tiempo promedio]	-Diseño, especificación y evaluación del TAD diseñado.	Temas 3, 4 y 5
Etapa IV			-Implementación abstracta de los requerimientos del problema en términos del TAD desarrollado.	
Etapa V			-Diseño y evaluación de cada una de las implementaciones del TAD especificado.	
Etapa III	Trabajo autónomo, trabajo cooperativo y autoevaluación -Desarrollo- Grupo pequeño (2 p.)	11,94h [Tiempo promedio]	-Desarrollo del modulo de la especificación del TAD.	Temas 3, 4 y 5
Etapa IV	Trabajo autónomo, trabajo cooperativo y autoevaluación -Desarrollo- Grupo pequeño (2 p.)		-Desarrollo del módulo principal del programa, independiente de la implementación del TAD.	
Etapa V	Trabajo autónomo, trabajo cooperativo y autoevaluación -Desarrollo- Grupo pequeño (2 p.)	69,19h [Tiempo promedio]	-Desarrollo de los módulos de implementación.	Temas 3, 4 y 5
Etapa VI	Trabajo autónomo, trabajo cooperativo y autoevaluación -Desarrollo- Grupo pequeño		-Codificación y ensamblaje de todos los módulos en lenguaje fuente.	
Etapa VI	Evaluación Grupo pequeño (2 p.)	15 min. [Tiempo promedio]	-Corrección y evaluación personalizada del programa final.	

Tabla 4. Características del las etapas de desarrollo de la práctica

habilidades de análisis, autocrítica y decisión; por otra parte, los alumnos pueden solicitar cualquier sugerencia o aclarar cualquier duda que les permita seguir trabajando con éxito. Con objeto de garantizar unos estándares de calidad equilibrados para todos los grupos, en cada una de las mencionadas etapas se exige un número mínimo de tutorizaciones, que puede oscilar entre una y dos. Las tutorizaciones se usan además para dar el visto bueno al desarrollo del trabajo de cada etapa, con lo que su función se extiende también a la de evaluación. Todos los grupos están obligados a respetar el calendario de trabajo establecido para cada una de las etapas, lo cual les impone una actividad constante a lo largo del curso, habiéndose mostrado como una medida muy efectiva del rendimiento global. Hemos constatado como el trabajo cooperativo y autónomo convenientemente monitorizados y asistidos a través de las tutorizaciones, además de implicar activamente a los alumnos, resultan muy eficaces en el desarrollo de su capacidad de autoaprendizaje.

- La *autoevaluación*, la *evaluación* y la *evaluación final*. La autoevaluación constituye una de las principales virtudes del aprendizaje activo. Por autoevaluación se entiende la capacidad de tomar conciencia por sí mismo sobre las cualidades y el progreso del trabajo que se está desarrollando. El ejercicio de autoevaluación puede llevarse a cabo a diferentes niveles y en diferentes momentos del desarrollo de una tarea. Por lo que a la práctica respecta, la autoevaluación se ejercita a dos niveles y en cada una de las etapas de la misma, de la siguiente manera: el primer nivel de autoevaluación de una etapa se lleva a cabo a través del análisis y la autocrítica del estudiante respecto al desarrollo de la tarea de la etapa, además de la posible ayuda que puedan proporcionarle otras fuentes de conocimiento, como libros, el propio profesor, etc…; el segundo nivel de autoevaluación de una etapa se lleva a cabo a través del desarrollo de la etapa siguiente, pues cada etapa depende de la anterior y ninguna puede ser desarrollada satisfactóriamente sin que el éxito de la anterior haya sido constatado. Por tanto, el ejercicio de autoevaluación estimula el autoaprendizaje a la vez que conduce inexorablemente al éxito de la práctica. En este sentido, hay que matizar que cada etapa de desarrollo es evaluada por el profesor de manera presencial, siendo la evaluación final el resultado de ponderar las evaluaciones positivas de cada etapa.

3.4. Cálculo del esfuerzo de los alumnos

El esfuerzo necesario para que un alumno lleve a cabo cada una de las etapas del desarrollo de la práctica se refleja en la tercera columna de la tabla 4. Obsérvese como el tiempo correspondiente a algunas actividades está etiquetado como «tiempo promedio». Ello se debe a que el esfuerzo en relación a dichas actividades no es uniforme para todos los alumnos, pudiendo sufrir sensibles variaciones dependiendo de las necesidades de la persona o del grupo del que forma parte. Por ejemplo, en las actividades de tutorización de las etapas III, IV y V, el esfuerzo del trabajo es, de promedio, de 1 hora y 15 minutos para cada alumno, variando sensiblemente de unos alumnos a otros.

El cálculo del tiempo promedio de las actividades de aprendizaje no presenciales se llevó a cabo durante el transcurso de cada una de ellas, para lo cual se solicitó la colaboración de los estudiantes. El procedimiento se basó en la elaboración de unas plantillas apropiadas que eran completadas por los alumnos. Con objeto de obtener una valoración lo más rigurosa y objetiva posible, se les exigió el compromiso de seguir escrupulosamente una serie de pautas establecidas.

Por lo que respecta a las actividades presenciales de tutorización y evaluación de la práctica, el esfuerzo era controlado por los profesores mediante las correspondientes plantillas en las que se anotaban los aspectos destacables de cada consulta, entre ellos el tiempo dedicado a la atención de los alumnos.

En relación al esfuerzo correspondiente al resto de las actividades de la tabla 4, al tratarse de tareas cuyo agente activo era el profesor, éste era quien lo fijaba, siendo uniforme para cada estudiante.

Conclusiones sobre el esfuerzo de los alumnos

De la tercera columna de la tabla 4 se concluye que el esfuerzo de los alumnos relacionado con el

desarrollo de la práctica es de 93,38 horas, lo cual equivale a 3,73 créditos ECTS.

El número total de créditos ECTS obtenido mediante el desarrollo del proyecto de adaptación de la asignatura a las directrices del EEES es de 7,24 [5], por tanto, el número de créditos para el desarrollo del resto de actividades es de 3,51. Nos encontramos pues en una situación en la que el peso de las actividades relacionadas con el desarrollo de la práctica es similar al del resto de actividades: 3,73 frente a 3,51.

4. Resultados de la implantación del proyecto

Los principales resultados de la implantación del proyecto de mejora de la calidad docente [5] se extraen de la experiencia de la práctica. En este sentido cabe matizar:

- Una proporción ligeramente superior a las tres cuartas partes de los alumnos -149 de 195- siguieron el proceso de adaptación metodológica, cuya opinión obtenida a través de los correspondientes cuestionarios fue considerada muy positiva. Los aspectos que más les incitaron a una participación activa en el desarrollo de la asignatura fueron: la disponibilidad de los medios adecuados, un progreso continuo del cual eran conscientes, y sobretodo la garantía de que sacarian partido de su trabajo.
- De los 149 alumnos monitorizados, 123 superaron la práctica, lo que supone un 82,5%.
- De los 123 alumnos que superaron la práctica, 114 superaron también el examen y así la asignatura, lo que en términos porcentuales supone el 92,68%.
- El costo medio de la tutorización de la práctica depende de cada grupo, así, mientras hay grupos que con una única tutorización por etapa –de unos 30 minutos de duración- tienen suficiente, otros han precisado una asistencia del profesor mucho mayor –del orden de las cinco horas-.
- La disponibilidad de recursos didácticos adecuados orientados al desarrollo de ta tarea docente [2], [3], junto con una apropiada coordinación con el resto de las actividades de aprendizaje, supuso un extraordinario soporte tanto para los alumnos como para el profesor, pues entre otros aspectos, permitió reducir sensiblemente el tiempo dedicado a las tutorizaciones.
- En relación a la calidad del software desarrollado por los alumnos que superaron la práctica cabe matizar:
  La práctica contiene, de media, entre 3000 y 4000 líneas de código.
  Los parámetros considerados en relación a la evaluación de las diferentes etapas de la práctica incluyen: la rigurosidad del diseño del modelo de datos, la eficiéncia del modelo de datos, la eficiencia de cada una de las dos implementaciones diferentes del modelo de datos, la interface gráfica del programa y los resultados de la ejecución del mismo.
- El uso de la metodología facilita la implementación del software en cualquiera de los modernos lenguajes de cuarta generación –Java, C++, ADA,…-.
- En cuanto al promedio del esfuerzo realizado por el profesor, el conjunto de todas las actividades para llevar a cabo la asignatura requiere una dedicación de 125 horas para un grupo 80 alumnos –tamaño considerado razonable-, de las cuales 75,75 corresponden a actividades relacionadas con las prácticas y 49,25 al resto de actividades. Esta sensible diferencia surge a raiz de la atención personalizada y en grupos reducidos que el profesor dedica a los alumnos en el desarrollo de la práctica.

5. Conclusión

La adaptación de una asignatura a las directrices del EEES requiere un notable esfuerzo por parte del profesor, que no siempre dispone de los recursos necesarios para llevarlo a cabo. Además, éste no es el único sujeto implicado activamente en la actividad, los alumnos también juegan un papel esencial sin el cual cualquier iniciativa carecerá de la profundidad y seriedad suficientes.

El mantenimiento de una actitud permanentemente activa del alumno en el desarrollo de la tarea docente es una de las principales bazas del éxito escolar. En este sentido, se ha presentado un proyecto de práctica monitorizada a partir de la idea del ABP, mostrando una gran efectividad en los resultados finales. Así, el éxito escolar ha sido muy elevado, pues el 82,5% de los alumnos presentados superó

la práctica, y de éstos, el 92,68% superó el examen, con lo que la práctica además ha demostrado ser un elemento muy eficaz en la superación del examen. La motivación de los estudiantes también ha redundado en la calidad del software diseñado, con un nivel medio altamente aceptable. A ello hay que añadir la buena acogida que la iniciativa ha tenido por la mayoría de los estudiantes, sobretodo por el hecho de garantizarles unos resultados acordes con el esfuerzo desarrollado

La atención personalizada a través de la tarea de tutorización por parte del profesor ha jugado un papel importante en los mencionados logros y consideramos que constituye una de las bazas a considerar en el diseño de cualquier proyecto docente de estas características.

Por lo que respecta al esfuerzo del profesor, éste se ve duplicado en relación al actual sistema de créditos. A ello hay que añadir la pobre consideración y soporte que el sistema educativo actual ofrece a los docentes, lo cual no incentiva precisamente sus pasiones respecto al desarrollo de proyectos de estas características.

Para evitarlos riesgos en la implantación de una experiencia de este tipo, es importante haber valorado préviamente y de manera objetiva una serie de factores, entre los que destacan el esfuerzo del alumno, una adecuada monitorización por fases del proyecto y las posibilidades que ofrece el proyecto de desarrollar la capacidad de autoaprendizaje.

La adaptación de las asignaturas a las nuevas directrices del EEES sólo supone el primer paso en el diseño de los futuros planes de estudio. Posteriormente debe establecerse la coordinación horizontal entre las materias de cada curso con objeto, entre otros aspectos, de distribuir adecuadamente la carga de trabajo de los estudiantes. En este sentido, en [5] se presentan los resultados de la coordinación entre las asignaturas *Tecnología de la Programación* y *Estructuras de la Información*, que componen el conjunto de materias relacionadas con el aprendizaje de la programación de segundo nivel. La coordinación con el resto de asignaturas del mismo curso constituye nuestro próximo objetivo para alcanzar la plena adaptación.

Agradecimientos

Este trabajo ha sido realizado en el marco de la convocatoria anual de ayudas a la mejora de la calidad de la docencia para el año académico 2005-06, de la Universidad de las Islas Baleares. Su desarrollo ha sido posible gracias a la colaboración e implicación de los estudiantes. Por ello queremos agradecer a aquellos que cursaron la asignatura *Estructuras de la Información* durante el curso 2005-06 su complicidad en la tarea, de la cual otros muchos también podrán beneficiarse.

Referencias

[1] Bloom, B.S.; Hastings, J.T.; Manaus, G.F. *Taxonomía de los objetivos de la Educación* . Marfil, 1973.

[2] Fiol, G.; Barceló, M. J. *Estructures de la Informació. Part 2: Programació amb Tipus Abstractes de Dades*. Col.lecció Materials didàctics, 88, Universitat de les Illes Balears, 2001.

[3] Fiol, G.; Barceló, M. J. *Estructures de la Informació. Part I: Agregats de Dades*. Col.lecció Materials didàctics, 67, Universitat de les Illes Balears, 2000.

[4] Fiol, G.; Fiol, C.; Miró, M. *Programación dirigida por datos: una experiencia docente adaptada a las directrices del EEES*. Actas del 14º Congreso Universitario de Innovación Educativa en las Enseñanzas Técnicas, CUIEET 2006.

[5] Fiol, G.; Fiol, C.; Miró, M. *Propuesta de coordinación de las asignaturas Estructuras de la Información y Tecnología de la Programación y su adaptación a las exigencias actuales del EEES: diversificación de las actividades de aprendizaje de los alumnos*. Proyecto de mejora de la calidad de la docencia, Universidad de las Islas Baleares, 2005-06.

[6] Markham, T. *Project Based Learning Hand-Book* . IFE, 2003.

[7] Silberman, M. *Active Learning: 101 strategies to teach any subject*. Allyn & Bacon, 1996.

Recursos docentes

Herramientas para la creación de tutoriales inteligentes en un entorno de teledocencia utilizando redes bayesianas

R. Laza, R. Pavón, D. Glez-Peña, J. R. Méndez, F. Fdez-Riverola
ESEI: Escuela Superior de Ingeniería Informática
Universidad de Vigo
Campus Universitario As Lagoas s/n, 32004, Ourense
{rlaza, pavon, dgpena, moncho.mendez, riverola}@uvigo.es

Resumen

Este trabajo presenta tres herramientas complementarias: (*i*) una herramienta de autor para la creación de tutoriales en formato XML, (*ii*) SITUA un sistema de edición de tutoriales inteligentes vía web y (*iii*) una utilidad para la creación automática de la red bayesiana que representa el tutorial. Entre otras características, SITUA es capaz de guiar al alumno a través de los conceptos que conforman el tutorial, mostrando únicamente aquellos cuyo estudio es viable en cada momento. Para ello, se ha hecho uso de las redes bayesianas que han permitido dotar al sistema del mecanismo de razonamiento necesario para guiar al alumno de forma inteligente. SITUA se enmarca dentro de un proyecto más amplio denominado Ariadna, un sistema integral de teledocencia desarrollado por miembros del Departamento de Informática de la Universidad de Vigo.

1. Introducción y Motivación

En la actualidad, se hace cada vez más notable la incorporación de las TIC como complemento para el proceso de enseñanza/aprendizaje tradicional. La inclusión de elementos tecnológicos permite incrementar la eficiencia a la hora de transmitir y adquirir nuevo conocimiento, aumentar su disponibilidad, así como de posibilitar el manejo de un gran volumen de información y facilitar su actualización.

Entre los complementos tecnológicos más conocidos aplicados al proceso de enseñanza tradicional, se encuentran las aplicaciones informáticas educativas. Este tipo de aplicaciones ha ido creciendo en popularidad de forma paralela al auge de las nuevas tecnologías e Internet en la sociedad. Prueba de ello, es el hecho de que en la propia red se pueden encontrar numerosos ejemplos de aplicaciones formativas que incorporan distintos niveles de funcionalidad.

Los tipos de aplicaciones educativas existentes son numerosos y variados, e incluyen desde simples recopilaciones de documentos indexados y enlazados mediante hipertexto (como la mayoría de los tutoriales que se pueden encontrar en Internet), hasta vistosas aplicaciones que se apoyan en capacidades multimedia y en diversos recursos gráficos para mostrar los contenidos al estudiante.

La diferencia existente entre los distintos enfoques radica principalmente en la vistosidad con la que se muestran los contenidos, aunque en ambos casos, el procedimiento de enseñanza del material docente sigue un esquema pasivo y lineal muy parecido a lo que sería leer un libro. Además, aunque la presencia de la animación multimedia pueda sorprender inicialmente, puede llegar a cansar, provocando la desmotivación y el abandono por parte del estudiante.

Actualmente, son ya numerosos los expertos en pedagogía e informática que se encuentran investigando en el desarrollo de sistemas tutoriales inteligentes (ITS, *Intelligent Tutoring Systems*) [6] como modelos flexibles, interactivos y adaptativos que utilizan técnicas estadísticas y de inteligencia artificial como el razonamiento basado en casos, redes bayesianas, teoría de la probabilidad, factores de certidumbre o lógica difusa para la mejora del proceso de enseñanza/aprendizaje asistido por ordenador.

También en la actualidad, está teniendo mucho auge la enseñanza a distancia o teleformación, debido principalmente a la escasez de tiempo de los profesionales y a las restricciones innatas de la enseñanza tradicional. En este sentido, la teleformación proporciona un método de enseñanza sin restricciones de tiempo ni de espacio. Los alumnos pueden aprender por ellos

mismos a su ritmo, y a la vez, trabajar y colaborar con otras personas de diferentes lugares.

Ariadna [1] es un sistema de gestión de cursos on-line (CMS, *Course Management System*) desarrollado en el marco de un proyecto de investigación (FIT-150500-2003-287) cuyo objetivo principal es el de crear una herramienta para el aprendizaje cooperativo, mediante la cual los alumnos no se encuentren aislados, pudiéndose ayudar unos a otros. Dentro de los cursos on-line proporcionados por Ariadna se encuentran los tutoriales, los cuales disponen de un sistema de enseñanza personalizada e inteligente.

El presente trabajo presenta SITUA (*Sistema de Tutoriales para Ariadna*) como soporte a la edición de tutoriales inteligentes vía web. Dichos tutoriales están disponibles para todos los alumnos de Ariadna, que podrán cursarlos siendo guiados en todo momento por SITUA, que les monitorizará su avance en función del grado de conocimiento que vayan adquiriendo.

SITUA se enmarca, por tanto, dentro de los ITS basados en la creación de un modelo de usuario, que representa los progresos que realiza el alumno en el proceso de adquisición de conocimientos en cada tutorial. Su principal característica es la adaptabilidad al usuario. Esto requiere un proceso de razonamiento, pudiéndose optar por alguna de las diferentes representaciones del conocimiento antes citadas. Concretamente, el modelo de representación utilizado por SITUA es una red bayesiana [4]. Esta técnica combina el riguroso formalismo probabilístico con una representación gráfica adecuada y eficientes mecanismos de inferencia [5].

Los tutoriales que maneja SITUA se guardan en formato XML conforme a una DTD que se ha establecido durante la creación del sistema. Los contenidos del tutorial se guardan en formato HTML y éste es embebido en los documentos XML que almacenan la estructura del tutorial. Este formato de almacenamiento hace muy complicada la creación de contenidos para el sistema de forma manual. Además no es posible incluir elementos multimedia tales como imágenes o animaciones Flash en los contenidos de los tutoriales, lo que limita en gran medida la capacidad expresiva de los mismos. Por ello, SITUA se complementa con una herramienta de autor que permite a los creadores de tutoriales olvidarse de los detalles de cómo se representa la información internamente y centrarse únicamente en el contenido de los mismos.

Además, los autores de los tutoriales también tienen a su disposición otra herramienta que crea automáticamente una red bayesiana básica a partir de un tutorial dado. Posteriormente, se permite la edición de la red para su adecuación a las necesidades del itinerario que se desea definir y se comprueba si la red modificada sigue siendo válida.

El resto del trabajo se estructura como sigue. La sección 2 describe la arquitectura del sistema que permite la creación de tutoriales en SITUA. La sección 3 presenta la estructura interna que se emplea para representar el contenido de los tutoriales. La sección 4 detalla cómo se han aplicado las redes bayesianas para dotar a SITUA de un mecanismo de razonamiento. A continuación, la sección 5 resume los aspectos más destacables de la interfaz de usuario de SITUA y de las herramientas de autor. Finalmente, la sección 6 comenta el impacto y la implantación de SITUA en un entorno real.

2. Herramienta de Autor para la Creación de Tutoriales en SITUA

El sistema ha sido diseñado para ser ejecutado en un entorno distribuido accediendo a una base de datos centralizada. La implementación se ha llevado a cabo siguiendo el modelo cliente/servidor (ver Figura 1), que para una aplicación web supone que el cliente será quien solicite la información y el servidor quien responda a esas solicitudes.

La misión del servidor es escuchar las peticiones de los clientes. Cuando llega una petición, primero verifica que el cliente está autorizado, luego procesa la petición (que en la mayoría de los casos incluirá una conexión con una base de datos para recuperar o actualizar información), y finalmente devuelve la respuesta al cliente, que será una página HTML generada dinámicamente a partir de los datos contenidos en un documento XML.

En el servidor se almacenan las páginas XML y las plantillas XSLT necesarias para transformar el contenido de dichas páginas en código HTML entendible por el navegador cliente. Esta transformación se lleva a cabo mediante el procesador XSLT Apache Xalan. Se ha elegido

este procesador porque está escrito totalmente en Java y por su facilidad de integración con Apache Tomcat. El acceso a las distintas bases de datos se realiza a través de un puente JDBC. Como servidor web y contenedor de servlets se ha utilizado Apache Tomcat 5.5.9, ya que proporciona un excelente rendimiento en entornos multiusuarios

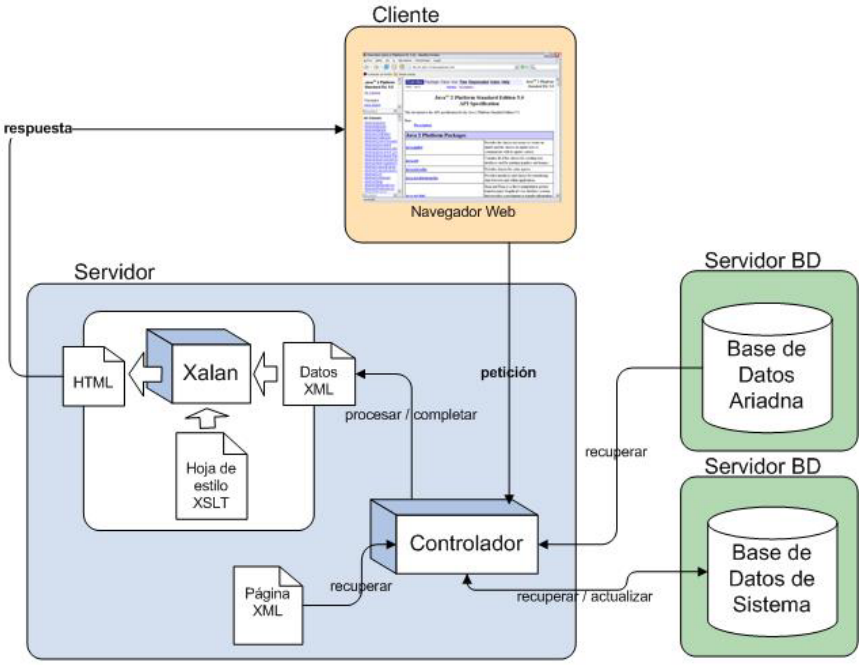


Figura 1. Arquitectura del sistema desarrollado

3. Estructura de un tutorial SITUA

El componente principal del sistema SITUA son los tutoriales, cuya estructura se muestra en la Figura 2. Concretamente, un tutorial SITUA está compuesto por las siguientes partes:

- En un nivel jerárquico superior se encuentra el elemento tutorial, que engloba y representa la totalidad de los contenidos.
- En un nivel jerárquico inmediatamente inferior se encuentran los temas, que realizan una primera división general de los contenidos del tutorial.
- Posteriormente se encuentran los conceptos, que al igual que en los sistemas Bits [2] y Smodel [7], son consideradas las unidades principales de conocimiento. El alumno se encontrará en SITUA tutoriales divididos en conceptos, aunque éstos se engloben en temas debido a la similitud de los contenidos que incluyen. A medida que el alumno vaya dominando los conceptos, irá aumentando su nivel de conocimiento en el tema en el cual están incluidos dichos conceptos, e incrementará los conocimientos sobre el tutorial.
- Como último nivel en la jerarquía, se encuentran las explicaciones, que son unidades en las que se dividen los contenidos de un concepto y son mostradas de forma secuencial.

Adicionalmente, y como se puede ver en el esquema de la Figura 2, se incluyen elementos destinados a obtener información sobre el nivel de conocimientos adquirido por el alumno en determinadas partes del tutorial. Estos elementos son las preguntas cortas de selección y los exámenes.

Por un lado, las preguntas cortas de selección están asociadas a cada explicación de un concepto. De esta manera, cuando un alumno ha estudiado los contenidos incluidos en una explicación, se le realizan una serie de preguntas para medir el nivel de conocimientos adquiridos. Este tipo de preguntas tienen una estructura de selección, donde se propone un enunciado y una serie de respuestas posibles, permitiendo obtener información a un nivel detallado del conocimiento adquirido por el alumno.

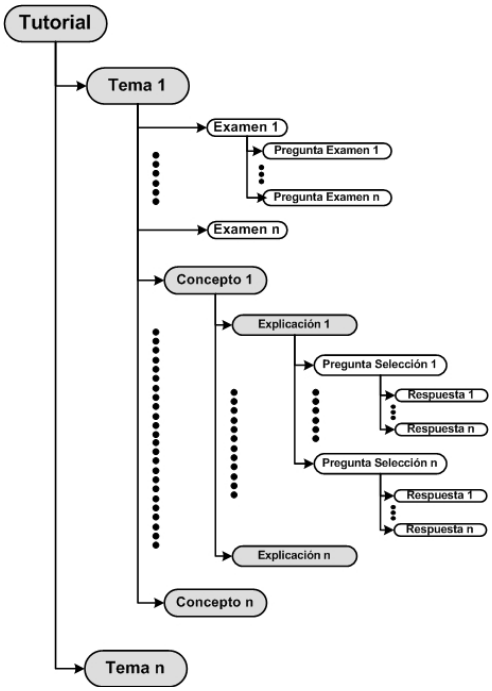


Figura 2. Estructura de un tutorial SITUA

Por otro lado, los exámenes están asociados a los temas del tutorial. Cuando un alumno conoce todos los conceptos de un tema determinado, podrá realizar el examen correspondiente. De este modo se obtiene una información más general del nivel de conocimientos del alumno. En este tipo de preguntas, el alumno tendrá que desarrollar en mayor o menor medida las respuestas, por lo tanto, será el profesor el encargado de evaluar y corregir las cuestiones del alumno en cada examen.

4. Redes Bayesianas en SITUA

En el desarrollo de ITS como Bits [2] o la aproximación propuesta por Gamboa y Fred [3] se utilizan de manera satisfactoria las redes bayesianas para representar el progreso de un alumno en un tutorial. Las características típicas de este tipo de técnicas se adaptan perfectamente a la división del conocimiento en conceptos con relaciones de dependencia, permitiendo representar fácilmente su jerarquía en la red. Al mismo tiempo, las redes bayesianas proporcionan un mecanismo sólido de inferencia que se puede utilizar para el proceso de diagnóstico sobre los conocimientos del alumno.

SITUA sigue la aproximación de las redes bayesianas facilitando la realización de procesos de diagnóstico para, por ejemplo, aconsejar al alumno en determinados aspectos del tutorial. Una aplicación práctica de esta funcionalidad es la de bloquear el acceso a la explicación de conceptos que no son abordables por falta de conocimientos previos.

En la estructura de la red bayesiana mostrada en la Figura 3, se pueden distinguir fácilmente los niveles jerárquicos de agregación que se corresponden con los elementos en los que se divide la estructura del tutorial, tal y como se había visto en la Figura 2.

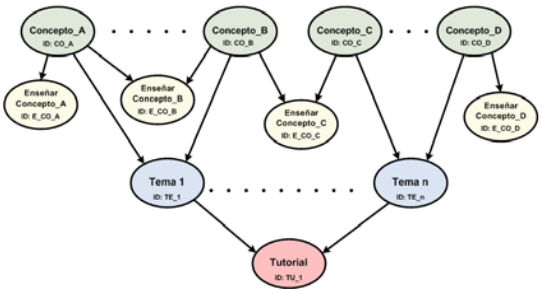


Figura 3. Modelo de red bayesiana utilizada en SITUA

Como se puede observar, existen nodos concepto, nodos tema y un nodo tutorial. Cada uno de ellos tiene un estado llamado *conocido* y otro estado llamado *desconocido*, que indican la probabilidad (complementaria) de que el alumno conozca o desconozca respectivamente un concepto, tema o tutorial determinado.

Además, existen unos nodos especiales denominados enseñar. Poseen los estados *viable* e

inviable que reflejan la conveniencia de que un concepto pueda ser presentado o no al alumno. En la red, cada nodo enseñar estará relacionado con el nodo concepto al que hace referencia. Sin embargo, un nodo enseñar puede relacionarse además con otros nodos concepto, lo que semánticamente implicaría que esos conceptos deben ser dominados previamente, antes de poder comenzar a estudiar el concepto correspondiente al nodo enseñar.

Cada vez que el alumno realiza las preguntas de selección de un concepto, SITUA estima de nuevo la probabilidad de que el alumno conozca este concepto y, aplicando el mecanismo de propagación de probabilidades de las redes bayesianas, recalculará todas aquellas probabilidades dependientes del resto de la red.

Haciendo uso de la red bayesiana, el sistema SITUA realiza las siguientes decisiones:

- Es viable el aprendizaje de un concepto cuando son conocidos todos los conceptos de los que depende y su estado es desconocido.
- Un concepto es conocido cuando el valor de su estado Conocido es superior a 0.7.
- Un tema es conocido cuando el valor de su estado Conocido es superior a 0.7 y todos sus conceptos son conocidos.
- Un tutorial es conocido cuando el valor de su estado Conocido es superior a 0.7 y todos sus temas son conocidos.

5. Trabajando con SITUA

Los usuarios de SITUA son tanto alumnos cursando los tutoriales, como profesores creándolos. Los primeros utilizan una aplicación web (integrada en el entorno Ariadna) que es capaz de "ejecutar" los tutoriales y realizar el seguimiento del progreso del alumno. Los profesores utilizan previamente la herramienta de autor para la creación de contenidos en sus tutoriales.

El objetivo fundamental de la herramienta de autor es facilitar la creación de tutoriales para el sistema SITUA. Esta herramienta se encuentra accesible desde el entorno Ariadna y permite exportar e importar tutoriales en formato compatible con SITUA.

Esta herramienta de autor permite crear los tutoriales con sus explicaciones, preguntas de selección y exámenes, además de recibir a través de ella las respuestas a las pruebas que realizan los alumnos. Para la creación de contenidos se proporciona un editor de texto enriquecido WYSIWYG que permite crear cualquier tipo de contenido HTML (incluyendo elementos multimedia como imágenes, Flash, etc.), tal y como se muestra en la Figura 4.

Existe además una herramienta que facilita la creación automática de la red bayesiana asociada a un tutorial. Concretamente, es capaz de crear una red inicial a partir de los contenidos elaborados por el profesor, los cuales se encuentran reflejados en el documento XML que representa el tutorial. Posteriormente, el docente puede establecer relaciones de dependencia adicionales indicando, por ejemplo, que un concepto depende del conocimiento previo de otros o incluso editando las tablas de probabilidades condicionales de la red bayesiana creada. Permite además el cálculo de los posibles itinerarios que el alumno podría seguir para conocer todos los temas del tutorial. En este sentido, mientras qua Figura 5 muestra el aspecto de la herramienta de edición de la red bayesiana, la Figura 6 presenta el aspecto de un tutorial visualizado por un alumno en SITUA.

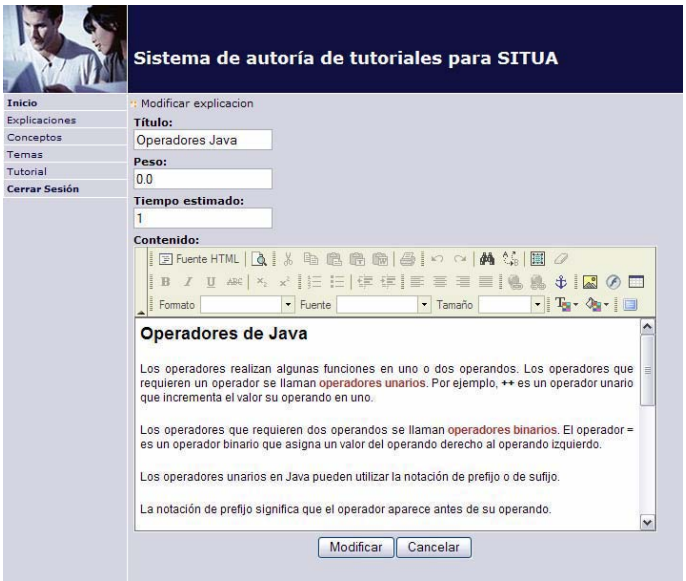


Figura 4. Edición de un concepto mediante el editor de texto enriquecido

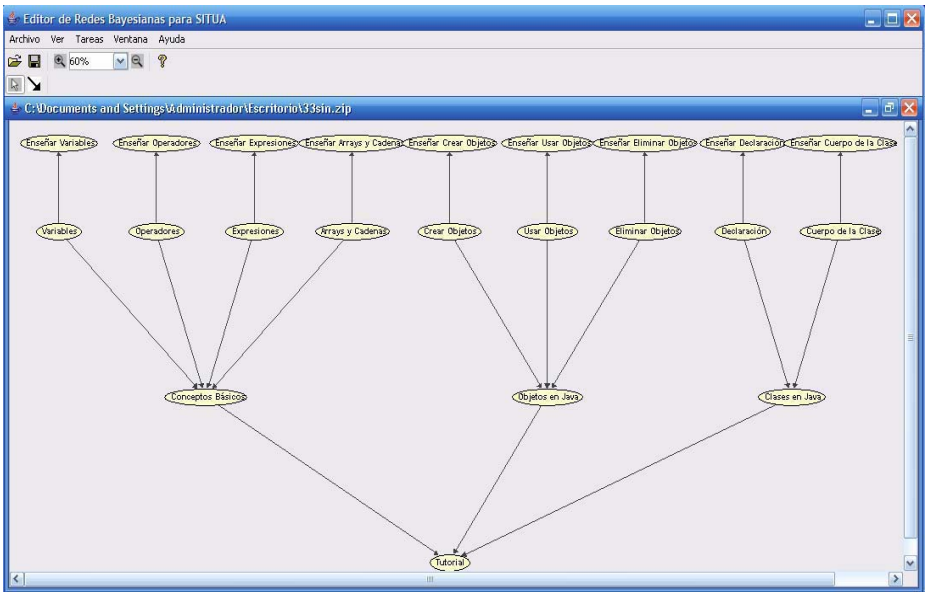


Figura 5. Herramienta de edición de la red bayesiana

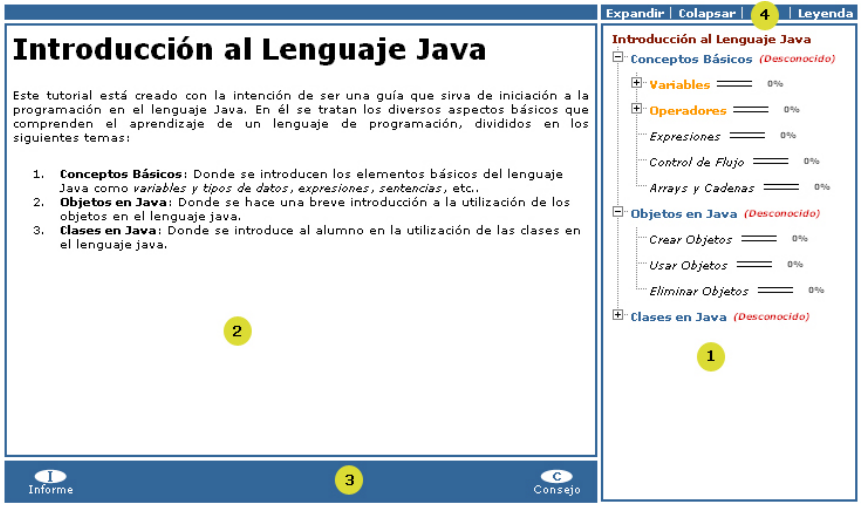


Figura 6. Visualización de un tutorial en SITUA

Los componentes principales de la interfaz mostrada en la Figura 6 son:

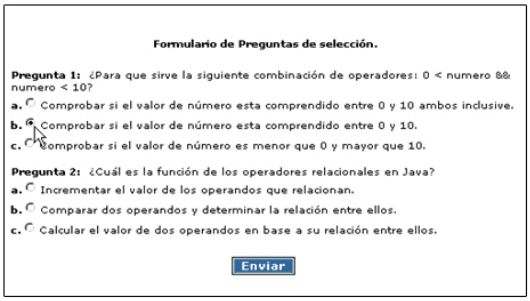
- **Índice de Contenidos**: muestra un árbol con la jerarquía de los elementos que componen el tutorial al que ha accedido el alumno (nº 1 en Figura 6). El índice de contenidos dispone de una pequeña barra de opciones en su parte superior (nº 4 en Figura 6).
- **Área de Contenidos**: es la zona central de la página (nº 2 en Figura 6), donde se le muestran al alumno los contenidos pertenecientes al elemento que seleccione en el índice.
- **Barra de Navegación**: dispone una serie de botones que proporcionan al alumno diversas funcionalidades a lo largo de su proceso de estudio del tutorial (nº 3 en Figura 6).

Los elementos del índice de contenidos siguen un código de colores intuitivo que se detalla en la Tabla 1.

Color	Concepto	Tema
Negro	*Inviable*: el alumno todavía no está preparado para abordarlo	--
Naranja	*Viable*: el alumno puede, si quiere, comenzar a estudiar el concepto	*Conocido*: el alumno conoce ya el tema, pero todavía no ha realizado el examen
Rojo	*Desconocido*: el alumno ha abordado el concepto, pero todavía no lo conoce en su totalidad	*Desconocido*: el alumno todavía no conoce el tema completamente
Verde	*Conocido*: el alumno ha logrado conocer el concepto	*Completo*: el alumno conoce el tema y ha superado el examen

Tabla 1. Código de colores del índice de contenidos

Como se ha comentado anteriormente, cada vez que el alumno termina la explicación de un concepto, deberá responder a una serie de preguntas cortas de selección y, una vez terminados todos los temas de un tutorial, deberá además realizar un examen, sólo evaluable por el profesor. El aspecto visual que presenta este tipo de actividades se puede ver en la Figura 7.



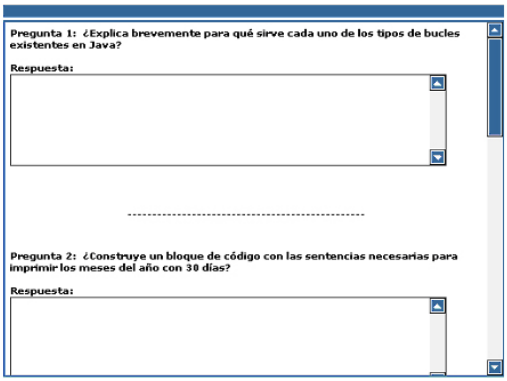


Figura 7. Preguntas de selección y exámenes en SITUA

6. Ventajas e Impacto de SITUA

SITUA se encuentra incluida en la plataforma de teledocencia Ariadna, la cual se utiliza con éxito desde hace varios años en diferentes materias de las titulaciones de Ingeniería Técnica en Informática de Gestión e Ingeniería Informática en la Escuela Superior de Ingeniería Informática de la Universidad de Vigo.

El uso del sistema de tutoriales SITUA está siendo probado como experiencia piloto este curso en algunas disciplinas como Estructuras de Datos y de la Información y Metodología y Tecnología de la Programación, no teniendo todavía la perspectiva suficiente como para sacar conclusiones definitivas sobre el impacto de la herramienta sobre el proceso de enseñanza/aprendizaje.

Sin embargo, el proyecto ha tenido gran aceptación entre el profesorado, que se ha animado a utilizar la herramienta confiando en que puede resultar de gran utilidad para, por un lado, mejorar el material didáctico que se le proporcionaba a los alumnos hasta la fecha (basado mayoritariamente en ficheros PDF y referencias bibliográficas) y por otro, para que los alumnos puedan seguir y estudiar los contenidos de la materia de un modo mucho más interactivo, organizado y guiado.

Referencias

[1] *ARIADNA. Herramienta de teleformación.* http://dellgwai.ei.uvigo.es/~socrates/pagina_inicial/index.php

[2] Butz, C.J.; Hua, S.; Maguire, R.B. *Bits: a Bayesian Intelligent Tutoring System for Computer Programming.* Department of Computer Science, University of Regina. Regina, SK, Canada.

[3] Gamboa, H.; Fred, A. *Designing Intelligent Tutoring Systems: A Bayesian Approach.* Instituto de Telecomunicaçoes, Instituto Superior Técnico, Lisboa Portugal. Escola Superior de Tecnología de Setúbal, Campo do IPS, Estefanilha, Setubal Portugal.

[4] González E., Laza R., Pavon R., *Sistema de Tutorización basado en Agentes: SITUA.* I Taller en Desarrollo de Sistemas Multiagente (DESMA 2004).

[5] Jensen, F. *An Introduction to Bayesian Networks.* Springer N.Y., 1996.

[6] Polson, M. C.; Richardson, J. J. *Foundations of Intelligent Tutoring Systems.* Ed. Lawrence Erlbaum Associates, Inc, 1988.

[7] Zapata-Rivera, J.D.; Greer, J. *SMODEL Server: Student Modelling in Distributed Multi-Agent Tutoring Systems.* International Conference on Artificial Intelligence in Education AIED'01, 2001.

ALT: Algorithm Learning Tool

R. Laza, D. Glez-Peña, J. R. Méndez, F. Fdez-Riverola, J. Baltasar García, M. Reboiro
ESEI: Escuela Superior de Ingeniería Informática
Universidad de Vigo
Campus Universitario As Lagoas s/n, 32004 Ourense
{rlaza, dgpena, moncho.mendez, riverola, jgarcia}@uvigo.es, mrjato@correo.ei.uvigo.es

Resumen

Este trabajo presenta ALT, una herramienta on-line para la visualización gráfica, intuitiva y paso a paso de algoritmos computacionales escritos en Java. ALT es una herramienta extensible, que permite la inclusión de forma sencilla de nuevos algoritmos que se puedan representar gráficamente. Esta herramienta, desarrollada como Proyecto Fin de Carrera en el Departamento de Informática de la Universidad de Vigo, brinda un nuevo y útil recurso docente a los profesores de asignaturas de Informática donde se impartan materias relacionadas con la algoritmia.

1. Introducción y Motivación

En los planes de estudio actuales, por lo general la mayor parte de las horas lectivas se dedican a clases teóricas presenciales, que dejan menos tiempo del que profesor y alumno desearían para la realización de prácticas y/o ejercicios sobre la teoría. Además, es habitual encontrarse con que el alumno no realiza los ejercicios que se le proponen o que, teniendo dudas, no acude a tutorías para resolverlas.

Para solucionar estos problemas, los profesores de la materia MTP (*Metodología y Tecnología de la Programación*) impartida en la Escuela Superior de Ingeniería Informática (ESEI) de la Universidad de Vigo, propusieron la realización de un nuevo portal Web para la asignatura. El objetivo inicial del proyecto era el de servir de apoyo a la materia e implementar un lugar de trabajo común en el cual se pudiesen proponer distintos ejercicios, con sus correspondientes soluciones, para que el alumno los pudiese realizar o consultar de una manera más cómoda.

Partiendo de esta proposición inicial, se avanzó hasta concretar dos tipos de ejercicios que deberían estar presentes: (*i*) los cuestionarios de preguntas cortas y (*ii*) los ejercicios tipo test. Ambos tipos de ejercicios proporcionaban una solución genérica para la realización de ejercicios sobre los temas en los que no se pudieran proponer actividades más concretas.

Sin embargo, y ya desde un punto de vista más cercano a la materia MTP, es preciso destacar que los algoritmos tratados habitualmente son un elemento de difícil comprensión para los alumnos (cursando el primer año de Ingeniería Informática). Tras repasarlos en profundidad, se llegó a la conclusión de que los algoritmos de búsqueda y ordenación impartidos podían ser representados y "animados" de un modo gráfico, hecho que facilitaría en gran medida el estudio y comprensión de los mismos.

La intención de buscar nuevas formas de facilitar la enseñanza de algoritmos no es nueva [1,4,5]. En este sentido, el recurso más habitual se basa en utilizar precisamente animaciones de la ejecución de los mismos. Las diferencias existentes entre las diversas alternativas suelen estribar en el nivel de control de las animaciones (si permiten o no detener el algoritmo, avanzar paso a paso, etc.), la posibilidad de visualizar el valor de las variables o la vistosidad de la interfaz.

No obstante, la carencia principal de los animadores de algoritmos existentes radica en la extensibilidad, es decir, la posibilidad de modificar o incluir nuevos esquemas por parte de los alumnos. Este aspecto se consideró como una característica clave, ya que el poder hacer modificaciones puntuales sobre los algoritmos sería de gran ayuda para facilitar su comprensión.

Este artículo se presenta la herramienta ALT (*Algorithm Learning Tool*), como resultado final de la implementación real de esta idea. ALT constituye un sistema complejo, que permite tanto el control de la ejecución de algoritmos como su creación, modificación y posterior visualización por el alumno.

Concretamente, ALT se encuentra integrada y disponible dentro del nuevo portal Web de la

asignatura MTP (ver apartado de conclusiones), cuyos profesores planean poner en explotación en el próximo curso académico.

Mientras que la primera sección de este trabajo introduce la motivación existente de cara a la implementación de ALT, el resto del artículo se estructura como sigue. En primer lugar se detalla el funcionamiento y manejo de ALT durante la animación de los distintos algoritmos existentes. A continuación se describe cómo los alumnos pueden incluir nuevos algoritmos. Posteriormente se comentan brevemente algunos detalles de implementación y, finalmente, se exponen las conclusiones y trabajo futuro.

2. Animación de algoritmos con ALT

Esta sección describe el funcionamiento de ALT comentando, por un lado, los aspectos más destacables en la visualización y control de la animación de los algoritmos implementados: ordenación por burbuja, inserción, inserción binaria, selección, shell, MergeSort y QuickSort, además de búsqueda líneal, binaria, binaria recursiva y mediante un array índice.

2.1. Visión general

La Figura 1 muestra el aspecto de ALT durante la ejecución de un algoritmo de ordenación (ordenación por burbuja).

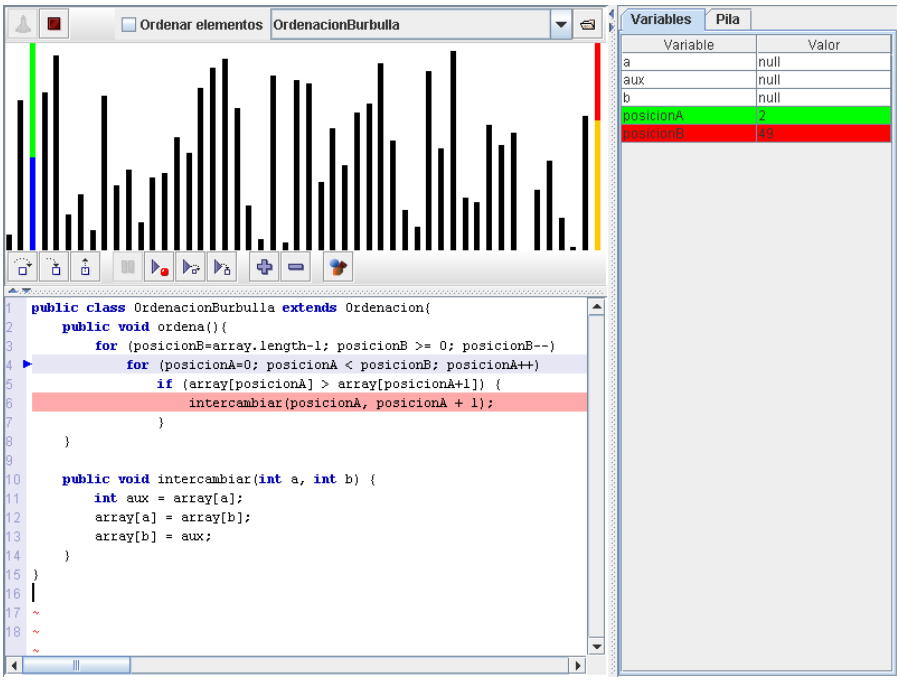


Figura 1. Aspecto general de ALT

La aplicación se puede dividir en cuatro zonas claramente diferenciadas:

- **Las barras de herramientas.** Se sitúan rodeando el visor del vector. Contienen los controles para el manejo de la animación.
- **Visor del vector**. Se sitúa en la parte superior izquierda y se encarga de mostrar el estado del vector durante la ejecución de la animación.
- **Visor del código fuente.** Se sitúa en la parte inferior izquierda y muestra el código fuente del algoritmo. Permite establecer puntos de ruptura en la ejecución y, durante la misma, resaltará la línea por donde se vaya ejecutando el algoritmo.
- **Panel de variables.** Se sitúa en la parte derecha y presenta dos pestañas: “Variables” y

"Pila". Permite visualizar los valores de las variables empleadas en el algoritmo y los registros de activación de las llamadas a funciones en la pila.

2.2. Control de la ejecución

La ejecución (o "animación") del algoritmo se maneja a través de la barra de herramientas, que contiene una serie de controles similares a los de los depuradores de los entornos de desarrollo o IDE (*Integrated Development Environment*).

Figura 2. Barra de herramientas de control

Tal y como muestra la Figura 2, el alumno puede efectuar las siguientes operaciones:

- *Step over, step into* y *step out* . Permiten saltar sobre una línea de código, entrar en un método o salir del mismo, respectivamente. Estos controles son de uso común en los IDE.
- *Pausa* . Detiene la ejecución.
- *Animar hasta breakpoint* . Continúa la ejecución hasta encontrar un punto de ruptura previamente establecido.
- *Continuar con step over* . Continúa la ejecución saltando línea a línea, pero con un pequeño intervalo de tiempo entre cada paso.
- *Continuar con step into* . Igual que el anterior, pero entrando en las llamadas a métodos.
- *Regular la velocidad* . Permite ajustar la velocidad de la animación, es decir, el intervalo de tiempo entre cada paso.

2.3. Visor del vector

El estado actual del vector sobre el que trabaja el algoritmo se puede ver en todo momento a través de un visor especializado, tal y como se muestra en la Figura 3. Dicho visor representa el vector en forma de gráfico de barras, donde el eje de abscisas representa las posiciones y el eje de ordenadas el valor del contenido en cada posición del vector.

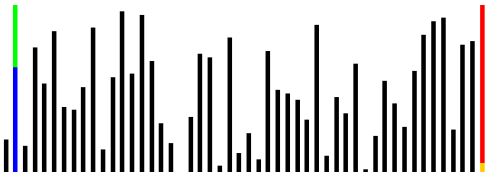


Figura 3. Panel de visualización del vector

Como se puede observar en la Figura 3, el visor permite además resaltar variables que almacenan posiciones del vector (índices) mediante dos colores. Uno de ellos cubre hasta la longitud de la barra, mientras que el otro alcanza la cima del panel con un color asociado a la propia variable que permite su identificación.

En ALT se pueden distinguir tres tipos de variables en función de su visualización:

- **Observadas.** Aparecen en el panel de variables con su valor.
- **Gráficas.** Son variables Observadas que además se les asocia un color y pueden ser vistas en el visor del vector *sólo* si se hace clic sobre ellas en el panel de variables.
- **Visibles.** Son variables Gráficas que se muestran *siempre* en el visor del vector.

Con el fin de mejorar la visualización de ciertos aspectos relevantes en la ejecución de algoritmos ALT permite la definición de **zonas.** Una zona representa una región del vector comprendida entre dos variables Visibles, que será representada en el visor del vector mediante un color de fondo diferente, tal y como muestra la Figura 4.

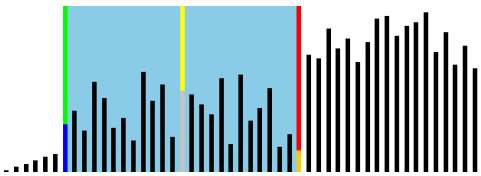


Figura 4. Zona señalada en el visor del vector

Las zonas han sido pensadas para algoritmos recursivos, donde muchas veces existen dos variables que delimitan la región del vector que se está tratando (por ejemplo en la ordenación QuickSort).

2.4. Visor de código fuente

Este visor muestra el código fuente de los algoritmos. Permite crear y eliminar *breakpoints*

en las distintas líneas que provocarán que la ejecución se detenga cuando sean alcanzadas.

La Figura 5 muestra el aspecto del visor de código fuente de un algoritmo a punto de ejecutar la línea 4 y donde se ha establecido un punto de ruptura en la línea 6.

```
1  public class OrdenacionBurbulla extends Ordenacion{
2      public void ordena(){
3          for (posicionB=array.length-1; posicionB >= 0; posicionB--)
4          for (posicionA=0; posicionA < posicionB; posicionA++)
5              if (array[posicionA] > array[posicionA+1]) {
6                  intercambiar(posicionA, posicionA + 1);
7              }
8      }
9
10     public void intercambiar(int a, int b) {
11         int aux = array[a];
12         array[a] = array[b];
13         array[b] = aux;
14     }
15 }
16
17
18
```

Figura 5. Visor del código fuente

2.5. Panel de variables y de pila

Este panel contiene a su vez dos componentes: una solapa para mostrar el valor de las variables y otra para visualizar la pila de ejecución.

La primera de ellas presenta una lista de variables junto con su valor y, en caso de ser índices del vector, su color asociado, tal y como muestra la Figura 6. Permite además realizar una serie de operaciones como son la selección de variables gráficas para que sean señaladas en el visor del vector, además de añadir o eliminarlas de la lista en durante la ejecución del algoritmo.

Variables | Pila

Variable	Valor
a	null
aux	null
b	null
posicionA	2
posicionB	49

Figura 6. Panel de variables

La segunda de las solapas muestra un árbol que presenta un resumen del estado actual de la pila. El elemento principal son las llamadas a métodos, que a su vez contienen todas las variables visibles dentro de ese método. Esta vista es muy útil para comprender el funcionamiento de los algoritmos recursivos, donde se suceden llamadas al mismo método de forma continuada. La Figura 7 muestra el aspecto de esta solapa durante la ejecución del algoritmo recursivo QuickSort.

Variables | Pila

- Pila
  - main
    - OrdenacionQuickSort.quickSort: 40
      - this
      - int[]A = int[50]
      - int i = 0
      - int j = 8
      - int indicePivote = 1
      - int pivote = 6
    - OrdenacionQuickSort.quickSort: 42
    - OrdenacionQuickSort.quickSort: 42
    - OrdenacionQuickSort.quickSort: 42
    - OrdenacionQuickSort.quickSort: 42
    - OrdenacionQuickSort.ordena: 5

Figura 7. Panel de la pila

3. Inclusión de Nuevos Algoritmos

Una de las características principales de ALT de cara al aprendizaje, es la posibilidad de que los alumnos puedan simular la ejecución de sus propios algoritmos. Gracias a esta funcionalidad, se pueden llevar a cabo modificaciones de los algoritmos impartidos en clase y comprobar cómo afectan los cambios de una forma inmediata.

La inclusión de un nuevo algoritmo se realiza a través de la página Web de la asignatura donde se integra el sistema ALT. El alumno deberá rellenar una plantilla como la que se muestra en la Figura 8.

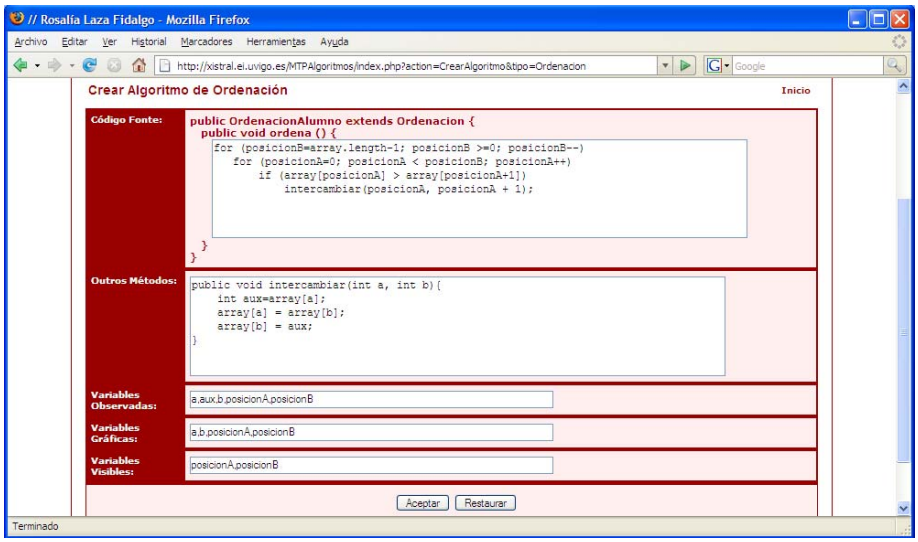


Figura 8. Formulario para la creación de nuevos algoritmos en ALT

Los campos a cubrir son los siguientes:

- **Código fuente**. Es el código del método de ordenación o búsqueda. Si el algoritmo es de búsqueda se deberá devolver un entero, mientras que si es de ordenación no se devolverá nada. Es necesario saber que el array a ordenar se encuentra en una variable denominada precisamente "array".
- **Otros métodos.** En muchas ocasiones es necesario, o más didáctico, el uso de métodos auxiliares. Para ello se proporciona otro campo donde se pueden introducir uno o varios métodos adicionales.
- **Variables observadas.** Una lista separada por comas de las variables que se desea que aparezcan visualizadas en el panel de variables.
- **Variables gráficas.** Un subconjunto de variables observadas que almacenan posiciones del array, y que se desea que sean señaladas en el visor del vector si se hace clic sobre ellas.
- **Variables visibles.** Un subconjunto de las variables gráficas que se desea que permanezcan señaladas durante toda la ejecución del algoritmo. Es preciso destacar que existen dos variables predefinidas que se pueden usar en el código fuente de forma global: "posicionA" y "posicionB". Se debe hacer uso de ellas si se quiere que estén señaladas siempre en el visor del vector, independientemente de si la ejecución se encuentra en el método principal o en métodos auxiliares.

Una vez cubierto el formulario se pasará a ejecutar ALT y, entre los algoritmos por defecto, aparecerá también el nuevo algoritmo del alumno.

4. Implementación

Esta sección describe brevemente algunos de los aspectos técnicos sobre la implementación llevada a cabo para ALT.

Tal y como se muestra en la Figura 9, los alumnos acceden a la herramienta a través de la página Web de la asignatura MTP, ya que ALT ha sido implementado como un Applet Java [6].

Una vez que se descarga e inicia dicho Applet, éste se conecta de nuevo al servidor Web para descargar los algoritmos, junto con un fichero de configuración que detalla cómo han de ser visualizados. Entre los algoritmos disponibles, se encontrarán tanto los que han sido creados por los profesores, como los que el alumno haya podido implementar durante su sesión de trabajo con la herramienta.

Una vez que el Applet, ejecutándose en la Máquina Virtual Java (JVM, *Java Virtual Machine*) del navegador cliente, disponga del

código fuente de los algoritmos, éstos serán compilados para luego pasar a ser ejecutados. Dicha ejecución es llevada a cabo por una JVM secundaria, controlada y monitorizada desde el Applet, que atenderá las peticiones del alumno (inicio, pausa, ejecución paso a paso, monitorización del estado del array y de las variables observadas, etc.).

Es preciso destacar que ALT ha sido implementado partiendo de otro proyecto anterior desarrollado en el Departamento de Informática, denominado *JavaTraceIt* [2,3]. Esta herramienta implementa un depurador y optimizador de aplicaciones Java y, entre otras funciones, permite la compilación de ficheros fuente y la ejecución paso a paso de los programas. Estas características están incluidas dentro de ALT y no han sido codificadas trabajando directamente con la librería Java destinada a tal fin (JPDA, *Java Platform Debugger Architecture*) [7], sino que se ha utilizado la API de JavaTraceIt, de más alto nivel y que ha facilitado en gran medida el trabajo realizado.

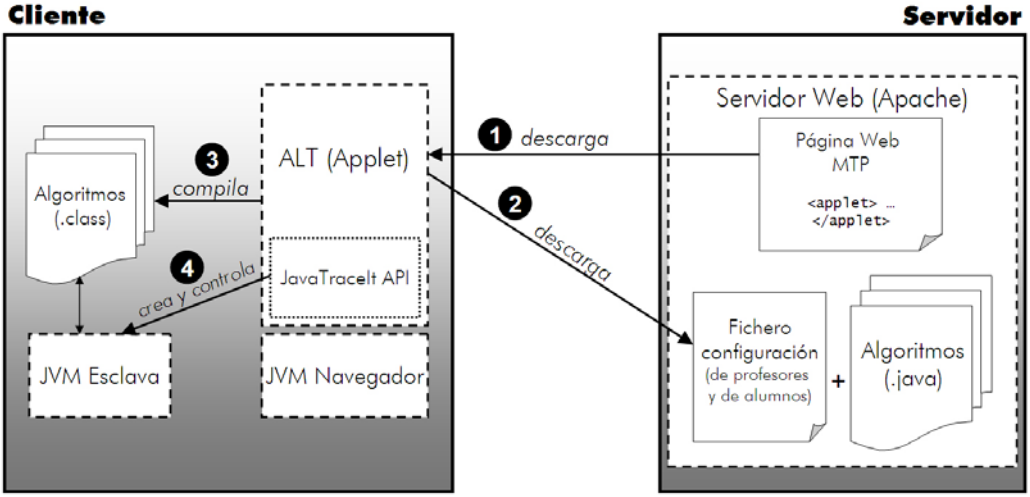


Figura 9. Arquitectura del sistema ALT

5. Conclusiones y Trabajo Futuro

Este artículo presenta una herramienta flexible y útil para la docencia de contenidos algorítmicos. Permite al alumno por un lado, ver los algoritmos de una manera gráfica y dinámica y, por otro, participar en la modificación de su código para ver cómo los cambios afectan a la "animación". Además está pensada para ser utilizada por el profesor en las clases presenciales como complemento a la pizarra.

ALT ha sido desarrollada recientemente y ya existe una versión totalmente funcional disponible en *http://trevinca.ei.uvigo.es/~rlaza/teoria.htm*. Se planea comenzar su utilización en el aula durante el próximo curso 2007-2008 en la asignatura MTP, por lo que todavía no se dispone de información acerca del nivel aceptación de la herramienta por parte de los alumnos. De todos modos, se confía en que será de gran ayuda porque ha surgido de una necesidad real detectada por el profesorado de la asignatura, que tiene varios años de experiencia impartiendo docencia de forma continuada.

El trabajo futuro se centrará principalmente en recoger las impresiones del alumnado durante el próximo curso, tratando de incorporar las aportaciones más interesantes. Sin embargo, se pueden anticipar varias mejoras interesantes como que el alumno pueda modificar el código fuente directamente sobre el visor y no tener que alterar el algoritmo mediante un formulario; la posibilidad de que se pudiesen visualizar no sólo vectores, sino árboles u otras estructuras; la capacidad para ocultar métodos auxiliares que sean superfluos para la animación o mejoras en la

interfaz, como por ejemplo el visor de pila que podría llegar a ser poco intuitivo para un alumno de primer curso de Informática.

Referencias

[1] Biliana, K.; Thiébaut, D. *Sorting Algorithms.* 1997. http://maven.smith.edu/~thiebaut/java/sort/demo.html.

[2] Glez-Peña, D.; Fdez-Riverola, F. *Understanding JPDA (debugging) & JVMTI (profiling) Java APIs within JavaTraceIt*. IADIS International Conference WWW/Internet 2006: ICWI, 2006.

[3] Glez-Peña, D.; Fdez-Riverola, F.; Méndez, J.R.; Díaz, F. *JavaTraceIt!: software didáctico de apoyo a la docencia en Java.* XI Jornadas de la Enseñanza Universitaria de la Informática: JENUI, 2005.

[4] Gosling, J.; Harrison, J.; Boritz, J. *Sorting Algorithms Demo*. 2001. http://www.cs.ubc.ca/~harrison/Java/sorting-demo.html

[5] Rosales, I. *Animador de Algoritmos*. Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Sevilla. http://www.lsi.us.es/docencia/asignaturas/ip1/trabajos/IndiceAnimador.htm.

[6] Sun Microsystems, Inc. *Applets*. Sun Developer Network. http://www.java.sun.com/applets.

[7] Sun Microsystems, Inc. *Java Platform Debugger Architecture (JPDA)*. 2002. http://java.sun.com/j2se/1.4.2/docs/guide/jpda/

Una herramienta para la enseñanza y aprendizaje de lenguajes formales y teoría de autómatas

Jose Jesus Castro-Schez, Ester del Castillo, Julian Hortolano
Dpto. Tecnologías y Sistemas de Información
Escuela Superior de Informática
Universidad de Castilla-La Mancha
Pº Universidad, 4 - 13071 Ciudad Real
JoseJesus.Castro@uclm.es Ester.Castillo@uclm.es

Resumen

El objetivo de este trabajo es presentar una experiencia de innovación docente basada en el uso de una herramienta software que ha sido desarrollada empleando tecnologías Web. Este software ha sido diseñado con el propósito de mejorar la calidad de la formación impartida, facilitando el aprendizaje de los principales conceptos de la asignatura Teoría de Autómatas y Lenguajes Formales. Esta herramienta comprende desde aspectos puramente docentes hasta aspectos de seguimiento del trabajo realizado por los alumnos durante el periodo de estudio de la asignatura.

1. Motivación

El empleo de software docente puede ser útil para la enseñanza y aprendizaje de una materia y una ayuda en la búsqueda de la excelencia y mejora de la calidad formativa de cualquier institución universitaria. En este trabajo se describe el diseño y desarrollo de una herramienta creada para mejorar la calidad de la formación impartida en la asignatura Teoría de Autómatas y Lenguajes Formales (TALF) de la Escuela Superior de Informática (ESI) de la Universidad de Castilla-La Mancha (UCLM).

La hipótesis de partida es que la calidad de la formación que se imparte en dicha asignatura se podría mejorar usando herramientas software, ya que en dicha asignatura, los ejercicios son muy importantes a la hora de asimilar conocimientos teóricos y estos pueden ser resueltos siguiendo una serie de pasos perfectamente definidos.

En la actualidad se pueden encontrar una serie de herramientas que se han propuesto, diseñado y usado con este objetivo, como ejemplo citar JFLAP[12], SEFALAS[5] o el proyecto SEPa![13].

Estas herramientas permiten la realización de ejercicios sobre o con autómatas y gramáticas que son dados como entrada al sistema.

En el diseño de la herramienta (SELFA) que se propone en este trabajo se ha intentado incorporar las características más interesantes de las herramientas antes mencionadas (entornos fáciles de usar, visuales o con capacidad gráfica, uso de un lenguaje para la definición de autómatas y gramáticas así como para la especificación de ejercicios sobre esos elementos). Además, se han añadido características útiles, como la posibilidad de ejecutarla sin necesidad de instalación local (utilizando una arquitectura cliente/servidor) y la creación e integración de una base de datos para recoger datos de uso con los que generar estadísticas que permitan analizar el trabajo realizado por los alumnos. Otro aspecto a destacar es el uso de variables para almacenar resultados intermedios de manera que se generen soluciones en un solo paso a diversos tipos de ejercicios tanto sobre gramáticas como autómatas indistintamente.

La solución al problema presentado se fundamenta en la propuesta de un lenguaje que permite definir autómatas, gramáticas y ejercicios sobre ambos, así como usar variables para guardar resultados intermedios. De este modo, el diseño de la herramienta se basa en un procesador para dicho lenguaje.

Este último aspecto permite que la herramienta pueda emplearse como ejemplo de aplicación de los conocimientos vistos en otra asignatura de los planes de estudio de la ingeniería informática, Procesadores de Lenguajes, en la solución a un problema real.

La idea de desarrollar esta aplicación tiene su origen al estudiar los resultados obtenidos por experiencias previas realizadas para otras asignaturas del plan de estudio de la Ingeniería Informática, concretamente por la herramienta ProleTool [9][10]. Esta herramienta se usa en la

asignatura Procesadores de Lenguajes y permite definir gramáticas y resolver ejercicios de aplicación de técnicas de análisis sintáctico descendente (LL1) y ascendentes (SLR1, LR1 y LALR1) sobre dichas gramáticas.

El resto del trabajo está estructurado de la siguiente manera: En la siguiente sección se presenta y analiza el problema que se pretende resolver. En la Sección 3 se estudiará como se ha diseñado y desarrollado la aplicación *SELFA*. A continuación, se mostrarán unos datos sobre el uso de la herramienta en el curso académico 05/06 y, para finalizar, se mostrarán las conclusiones y futuros trabajos.

2. Problemática existente

La asignatura de Teoría de Autómatas y Lenguajes Formales tiene una gran importancia en la formación de los titulados en Informática ya que proporciona los fundamentos teóricos de la disciplina, lo cual permite comprender mucho mejor la informática y sus orígenes (tanto históricos como matemáticos) y explorar sus problemas y posibilidades [4][6][7][8].

En la actualidad, parece que existe una preocupación por una enseñanza eminentemente práctica. Es por esto por lo que esta asignatura ha dejado de ser troncal en los planes de estudios y no tiene presencia en el Libro Blanco de la Ingeniería Informática, dentro de la propuesta para contenidos troncales del Título de Grado en Ingeniería [1]. No obstante, creemos o confiamos que las universidades, dentro de las asignaturas optativas, ofrecerán a los alumnos la posibilidad de elegir formarse en esta disciplina.

En la asignatura TALF se pretende que los alumnos adquieran conocimientos y desarrollen habilidades sobre autómatas, gramáticas y lenguajes formales que les permitan analizar, entender y solucionar problemas.

La asimilación y aprendizaje de los conocimientos y conceptos de la asignatura de una manera sólida requieren de la realización de un gran número de ejercicios por parte del alumno.

Sin embargo, en la mayoría de los casos, esto no es posible hacerlo de forma intensiva en clase, debido a la extensión del temario y al tiempo limitado de las mismas.

Por ello, se hace cada vez más necesario establecer medidas encaminadas a minimizar este problema. Una primera posible medida consiste en proporcionar material a los alumnos para que estos lo estudien antes de cubrir el tema en clase, de esta forma se podría dedicar ese tiempo a la realización de ejercicios. El principal inconveniente de esta medida es que los conceptos que los alumnos tienen que asimilar son muchas veces demasiados abstractos, lo cual puede dificultar su aprendizaje sin la explicación del profesor.

Otra posible medida implica diseñar y proporcionar a los alumnos una batería de ejercicios junto con sus soluciones de manera que les permita resolver cada uno de los ejercicios propuestos por su cuenta y comprobar la solución obtenida. El principal inconveniente de esta medida es la necesidad de mantener actualizada constantemente esta batería.

Como una alternativa a esta medida, en este trabajo se propone el diseño y desarrollo de una aplicación, a la que se ha nombrado SELFA, que permite a los alumnos la propuesta de ejercicios de la asignatura y la generación de las soluciones a los mismos, pudiendo comprobar como SELFA ha generado los resultados y si estos coinciden con la resolución hecha por el alumno. Esto puede motivar a los estudiantes en el estudio de la asignatura proponiendo ellos sus propios ejercicios, así como potenciar su capacidad creativa y de análisis.

Por otro lado se pretende que la herramienta también pueda ser empleada en las clases magistrales que imparte el profesor.

De este modo, se pretende mejorar la enseñanza y aprendizaje de la asignatura de TALF.

Además, el contexto en el que se mueve la Universidad Europea hacia el establecimiento de un Espacio Europeo de Educación Superior (EEES) y hacia el sistema de transferencia de créditos europeos (ECTS) [2] que pretende reflejar el esfuerzo real requerido y realizado por el estudiante para conseguir una serie de objetivos, hace necesario incorporar a SELFA un sistema de control de uso asociado a cada alumno que permita valorar el trabajo realizado por este en el estudio de la materia de TALF.

3. Diseño de SELFA

En el diseño de la aplicación SELFA[11] se han tenido en cuenta otras herramientas existentes, como JFLAP[12], SEFALAS[5], el proyecto SEPa![13] y ProleTool[10] así como los requisitos de los potenciales usuarios de la misma.

Se pretende que SELFA sea usada por dos tipos de usuarios: alumnos y profesores. Estos dos tipos tienen necesidades, deseos y prioridades comunes y diferentes, por lo que hay que tener en cuenta todo ello para realizar un diseño que satisfaga a ambos.

La principal funcionalidad que debe ofrecer la herramienta, desde el punto de vista del alumno y del profesor, es la de *aceptar ejercicios y calcular las soluciones a los mismos, mostrándolas de una forma clara y sencilla de manera que se permita comprender como se han alcanzado*.

A este objetivo común se deben añadir otras características deseables para ambos tipos de usuarios de cara a conseguir que la herramienta sea práctica y se use:

- Disponibilidad temporal. Debería estar disponible las 24 horas de todos los días.
- Disponibilidad geográfica. Sería interesante que la herramienta tuviera una instalación lo más sencilla posible para conseguir su uso en el ordenador del alumno, de los laboratorios, de las clases teóricas y en el ordenador del profesor.
- Actualizable. Esta herramienta surge con la idea de que esté en continua mejora. Un cambio en la aplicación debe ser fácil de trasladar a cada uno de los ordenadores donde esté instalada. Por ello, se ha de establecer un sistema de actualización que facilite dicha tarea.
- Intuitiva. Debe tener una interfaz amigable, no sólo a la hora de introducir ejercicios, si no en todas las opciones que pueda realizar el usuario con la herramienta.
- Facilitar la entrada de tantos ejercicios como se desee e incluso utilizar resultados intermedios para realizar operaciones sobre ellos.
- Correcta. Debe proporcionar una información precisa y libre de errores.
- Completa. Debe proporcionar toda la información necesaria para que se comprendan las soluciones obtenidas.
- Plataforma libre, se debe poder ejecutar en cualquier plataforma.

Desde el punto de vista de los profesores de la asignatura sería deseable que la aplicación tuviera las siguientes características generales:

- Ejemplar. La aplicación debe ser un ejemplo para futuros alumnos sobre como emplear los conocimientos teóricos y prácticos de la asignatura y en general de la carrera en la resolución de un problema real.
- Accesible. Debe tener un acceso cómodo y rápido para que se pueda usar fácilmente como herramienta de ayuda a la enseñanza en cualquier sitio.
- Funcional. De manera que se pueda acceder a las opciones más importantes y mostrar la información generada de la manera más adecuada dependiendo del ambiente en la que se use.
- Transparente. Debe permitir ver el trabajo que se realiza con la herramienta y quién lo realiza, si el usuario lo permite. Esta información podría ser empleada para valorar el trabajo realizado por el alumno a la hora de aprender los conceptos de la materia.
- Software Libre. Que esté desarrollado con licencia GPL, para así permitir que personas interesadas, como alumnos o personal docente, puedan mejorar la herramienta.

La herramienta ha sido diseñada de una manera incremental. En cada paso del diseño, se ha tenido un producto que los alumnos y profesores han podido probar. A cada producto obtenido se le ha incorporado un mecanismo de seguimiento de uso de la misma. Además, se le incorporaba un sistema para recoger comentarios y sugerencias de los usuarios que pudieran ser empleados para mejorar la herramienta.

Durante este proceso aparecieron nuevos requisitos que fueron considerados en el diseño y desarrollo final de la herramienta SELFA.

Las siguientes características aparecieron como nuevos requisitos durante este proceso:

- Lenguaje fácil de aprender y usar.
- Eficiente. Las soluciones a los ejercicios deben calcularse de forma rápida.
- Ayuda útil. Debe existir un sistema de consulta con todo lo referente a la herramienta que sirva como manual de usuario.

- Uniforme. Debe emplear en la medida de lo posible la misma notación, tanto para autómatas como gramáticas. Además esta debe ser lo más parecida posible a la empleada en clase de teoría.
- Gráfica. Las soluciones deben mostrarse de una manera gráfica para facilitar la comprensión de la misma.

El implicar a los usuarios en el desarrollo de la herramienta, considerando sus comentarios, ha servido para fomentar su uso.

En la siguiente sección se detallan las decisiones de diseño tomadas para satisfacer los requisitos de los usuarios

3.1. Interacción con el usuario

La primera decisión de diseño que se ha tenido que tomar es la de establecer como debe ser la entrada de ejercicios a la herramienta. Para ello, antes se ha definido con precisión cual va a ser la entrada.

Una entrada consiste en la definición de uno o varios autómatas, una o varias gramáticas y la especificación de una serie de operaciones sobre esos elementos definidos, que pueden o no devolver resultados intermedios sobre los que se van a poder volver a operar.

La definición de un autómata $A=(Q,\mathcal{A},\delta,q,F)$, implica la especificación del conjunto de estados (Q), del alfabeto de entrada $(\mathcal{A})$, de la función de transición (δ), del estado inicial (q) y del conjunto de estados finales (F).

La definición de una gramática, $G=(N,T,P,S)$, implica especificar el conjunto de símbolos No Terminales (N), el conjunto de Terminales (T), el conjunto de producciones (P) y el símbolo inicial (S).

La definición de un ejercicio implicará el uso de una palabra reservada asociada a dicho ejercicio y unos argumentos, entre los cuales se encontrará una gramática o un autómata previamente definido sobre el cual se va a aplicar la operación.

Para determinar como se va a proporcionar la entrada a SELFA, se han estudiado las distintas alternativas usadas en las herramientas existentes: vía formularios [3], gráficamente [12][13] o por medio de un lenguaje de entrada [5][10].

La alternativa gráfica es muy interesante para usuarios no especializados en el tema, pero por contra tiene como principales inconvenientes el ser lento a la hora de interactuar con la herramienta y obligar a tener que trabajar "on-line".

La entrada a través de formularios tiene la ventaja de ser un proceso guiado, lo cual es idóneo para usuarios noveles en el proceso pero tiene los mismos inconvenientes que la anterior.

Por estas razones, la forma elegida ha sido mediante un fichero de entrada escrito en un lenguaje en el que se especifiquen los autómatas, gramáticas y los ejercicios a resolver. Esta alternativa, aunque tiene el inconveniente de que obliga a los usuarios a aprender un lenguaje, tiene como principal ventaja el permitir trabajar "off-line" y proporcionar una forma rápida de introducir ejercicios. Para escribir ejercicios bastaría con tener un editor de texto e introducir el ejercicio o ejercicios que se quisiera solucionar. Además, el inconveniente se puede minimizar diseñando un lenguaje sencillo y fácil de aprender.

La solución que ofrece la herramienta será:

- Nuevos autómatas o gramáticas, más información adicional útil para comprender la solución y cómo se ha alcanzado, cuando la entrada es correcta.
- Una serie de mensajes útiles para detectar los errores que puedan aparecer en la entrada, cuando esta no es correcta.

La herramienta ofrecerá un mecanismo que permita navegar por la entrada y la información dada como salida.

3.2. Lenguaje de entrada a SELFA

El diseño del lenguaje de entrada es algo delicado, ya que es una de las partes más importantes de la herramienta y debe realizarse minuciosamente. El lenguaje diseñado tiene las siguientes características:

- Tiene cierta similitud con el lenguaje CUP, que se usa en clases prácticas de TALF, lo cual facilitará su aprendizaje.
- Permite la entrada en un solo archivo de todos los ejercicios que el usuario quiera y resolverlos en un solo paso. Esto es de utilidad cuando se tiene un tiempo de acceso limitado a Internet o se trabaja off-line.
- Los resultados intermedios de las operaciones realizadas pueden ser almacenados en variables sobre las que después se podrá volver a operar.

- Existe una única operación imprimir que actuará de una forma u otra dependiendo de lo que se le pase como argumento (un autómata o gramática).

La sintaxis en notación EBNF del lenguaje propuesto se muestra en la Tabla 1.

El símbolo `ID` corresponde al lexema identificador cuyo patrón de construcción es el siguiente: `ID=[A-Za-z][A-Za-z0-9]*.`

En este lenguaje existen unas restricciones que no pueden ser expresadas por medio de la sintaxis y que el procesador del lenguaje deberá detectar si no se cumplen, informando al usuario. Estas restricciones son:

- Toda gramática y/o autómata debe tener un identificador único.
- En la declaración de elementos de una gramática no se permite la asignación de dos tipos diferentes (terminal y nonterminal) a un mismo elemento.
- El símbolo inicial de una gramática ha de estar declarado anteriormente como un elemento nonterminal.
- La parte izquierda de una producción debe haber sido declarada como un elemento nonterminal.
- En la definición de un autómata los elementos que se definan como estados (states) no pueden definirse también como elementos del alfabeto (alphabet) y a la inversa.
- El elemento declarado como inicial debe haber sido previamente definido como un estado.
- Los elementos declarados como final deben haber sido declarados como estados.
- En las transiciones el primer identificador debe haber sido declarado como un estado, el segundo como un elemento del alfabeto y el resto de identificadores deben ser elementos del conjunto de estados.
- Las operaciones sobre autómatas (OPA) y sobre gramáticas (OPG) deben realizarse sobre autómatas o gramáticas previamente declarados o resultados de operaciones sobre estos.

```
PROGR   ::=  (DECL|OPNS){DECL|OPNS}
DECL    ::=  GRAM|AUTM
GRAM    ::=  grammar ID BGRAM
BGRAM   ::=  '{'DS SINI DP'}'
DS      ::=  T NT | NT T
T       ::=  terminal {ID','} ID';'
NT      ::=  nonterminal {ID','} ID';'
SINI    ::=  si ID';'
DP      ::=  productions '{'PROD{PROD}'}'
PROD    ::=  ID ':=' PDCHA;
PDCHA   ::=  ID{ID}{'|'ID{ID}}('|'$|$)
AUTM    ::=  automaton ID BAUTM
BAUTM   ::=  '{' EST ALF INI FIN TRANS '}'
EST     ::=  states {ID','} ID';'
ALF     ::=  alphabet {ID','} ID';'
INI     ::=  initial ID';'
FIN     ::=  final {ID','}ID';'
TRANS   ::=  transition '{'TRA{TRA}'}'
TRA     ::=  ID','ID '=' DEST
DEST    ::=  ID';'|'(' ID {','ID}')''';'
OPNS    ::=  OPG|OPA|PRIM
OPG     ::=  OG1|OGN|sim_null '(' ID ')' ';'
OG      ::=  ID '=' OPG1 '('ID')''';'
OPG1    ::=  inaccesibles|nodevterm|inutils|null|unitary|clean|fnc|fng
OGN     ::=  cyk '('ID','ID{ID}')''';'
OPA     ::=  OA1|OA2|OAN|equals'('ID','ID ')''';'
OA1     ::=  ID '=' OPA1 '('ID')''';'
OPA1    ::=  passND|passAFD|passAFDM|complement|inverse
OA2     ::=  ID '=' OPA2 '('ID','ID')''';'
OPA2    ::=  Intersection | union
OAN     ::=  recognize '('ID','ID{ID}')''';'
PRIM    ::=  print '('ID')''';'
```

Tabla 1. Notación EBNF de la sintaxis del lenguaje propuesto.

Como ejemplo de uso de SELFA, se muestra un ejercicio propuesto a los alumnos. El objetivo es, dado un Autómata Finito No Determinista con lambda transiciones, obtener un AFD mínimo equivalente y ejecutar este AFD sobre una palabra en la entrada. Para terminar, se propone al alumno que obtenga, sin usar SELFA, el AFD mínimo, aplicando los algoritmos vistos en teoría, y compare su resultado con el resultado de SELFA, usando la instrucción equals(A1,A2).

```
automaton prueba{
           states q0,q1,q2,q3,q4,q5;
           alphabet a,b;
           initial q0;
           final q5;
           transition{
                     q0,$=q1;
                     q0,$=q3;
                     q1,a=q1;
                     q1,b=q2;
                     q2,$=q5;
                     q3,a=q4;
                     q4,b=q4;
                     q4,$=q5;
                     q5,$=q0;   }
}
print(prueba);
pruebaAFDM=passAFDM(prueba);
print(pruebaAFDM);
recognize(pruebaAFDM, a b b a b b);
```

El resultado de esta ejecución provoca que SELFA muestre los resultados de los Autómatas intermedios que ha generado antes de obtener el AFD mínimo. Respecto al reconocimiento de la palabra, SELFA muestra la secuencia de transiciones aplicadas al ejecutar el autómata sobre la palabra en la entrada.

Usando este ejercicio, el alumno debería resolver el mismo problema y comprobar si lo ha hecho correctamente, usando la instrucción *equals (A, pruebaANDM)*, donde A será el AFD mínimo obtenido por él.

3.3. Tratamiento de la entrada de ejercicios

Para comprobar si el texto de entrada tiene una sintaxis correcta, verificar si se cumplen las restricciones antes mencionadas y extraer la información relevante de los ejercicios para poder solucionarlos es necesario que SELFA tenga un procesador de dicho lenguaje.

El procesador del lenguaje se ha diseñado por etapas y por fases. Primero, se ha prestado atención a la etapa de análisis, diseñando el analizador léxico, después el sintáctico y, por último, el analizador semántico. La integración de estos tres analizadores ha permitido construir el procesador buscado con las siguientes características:

- Portable. Se ha implementado en Java con lo que es multiplataforma.
- Extensible. Se ha diseñado de manera que se puedan añadir características nuevas si se desea, por ejemplo, nuevos algoritmos que solucionen más ejercicios.
- Integro. El procesador funciona como debe, obteniendo soluciones cuando la entrada es correcta e informando del mayor número de errores posible cuando la entrada no es correcta (posee un mecanismo de recuperación de error).

Una vez desarrollada la etapa de análisis, nos enfrentamos a la etapa de síntesis. En esta etapa se han implementado una serie de algoritmos sobre gramáticas: Eliminación de símbolos y producciones que deriven en la cadena vacía, eliminación de símbolos y producciones que no deriven en cadena de terminales, eliminación de símbolos inaccesibles y eliminación de producciones unitarias, así como pasar una gramática libre de contexto a Forma Normal de Chomsky (FNC) y a Forma Normal de Greibach (FNG) y algoritmo de Cocke-Younger-Kasami, (CYK) para decidir si una palabra pertenece al lenguaje definido por una gramática.

Respecto a las operaciones sobre autómatas, se implementa los algoritmos para pasar un autómata finito no determinístico con transiciones nulas a un autómata finito no determinista, pasar un autómata finito a un autómata finito determinista, pasar un autómata finito a un autómata determinístico mínimo, así como realizar la unión, la intersección, el inverso, el complemento autómatas. Comprobar si una cadena es reconocida por un autómata y comprobar si dos autómatas son iguales.

Por último, se ha integrado el procesador de lenguaje con los algoritmos que resuelven los ejercicios y la parte de la interfaz. La etapa de análisis del procesador de lenguaje obtiene, de las entradas correctas, la información relevante de cada ejercicio y se la comunica a la etapa de síntesis que se encarga de resolver los ejercicios mediante la llamada a los algoritmos que realizan las operaciones pertinentes. La integración del procesador de lenguaje con los algoritmos ha sido relativamente sencilla ya que ambas partes usaban un lenguaje común que era Java.

3.4. La salida de la herramienta

Una vez determinado como se introducen los ejercicios a la herramienta por medio de un lenguaje, como se analizan las entradas, o mejor aún los ejercicios, por medio de un procesador de dicho lenguaje y como se resuelven estos mediante la invocación de los algoritmos correspondientes, queda determinar cómo se van a presentar las soluciones al usuario de manera que se permita navegar por ellas y puedan ser mostradas en cualquier ordenador.

El formato interno en el que se generan las soluciones ha sido XML por ser un lenguaje fácil de generar y de utilizar por aplicaciones externas. Además, gracias a la tecnología XSL, estas soluciones en XML pueden ser formateadas pudiendo devolver la solución en formato HTML. Esto será de mucha utilidad al integrar esta parte con la de la interfaz externa de la herramienta.

3.5. Arquitectura de la herramienta

Uno de los requisitos importantes de la herramienta es que su instalación y actualización debía ser lo más sencilla posible, siendo lo deseable, incluso, que no necesite instalación. De este modo, se pretendía facilitar su uso en cualquier ordenador independientemente de su ubicación física. Además, otro requisito esencial era que debía registrar el uso que los usuarios, principalmente los alumnos, hacían de ella. Para cumplir estos objetivos se ha tomado la decisión de crear una aplicación centralizada usando una arquitectura cliente/servidor Web.

La arquitectura elegida para la herramienta ha sido una arquitectura multicapa Web, en la que cada capa ofrece sus servicios a la capa inmediatamente superior y recibe los servicios de la capa inferior. De este modo, se ha reducido el acoplamiento y la complejidad de la herramienta. La arquitectura Web tiene las siguientes ventajas:

- Toda la carga de trabajo cae sobre el servidor, que es en la que se ejecutará SELFA, haciendo la aplicación en el lado del usuario (el cliente) una aplicación muy ligera.
- No necesita instalación local, sólo un navegador conectado a Internet.
- Al ser una aplicación centralizada, se permite registrar el uso de la herramienta mediante una base de datos situada en el servidor.

De este modo, las actualizaciones de SELFA son inmediatas ya que, al estar la aplicación centralizada, basta con actualizar la parte del servidor para que los cambios sean visibles en todos los sitios donde se use.

La parte Web de la herramienta ha sido desarrollada usando el lenguaje de script PHP que tiene las características de ejecutarse en el servidor, ser mutiplataforma y soportar gran cantidad de usuarios. Además, la base de datos en la que se registrará todo el uso de la herramienta será MySQL. Este tipo de base de datos es muy potente, es gratuita, tiene una productividad elevada, un bajo consumo y un tiempo de respuesta bajo. Las soluciones son devueltas en formato HTML.

3.6. Capacidades incorporadas a SELFA

La herramienta SELFA, a diferencia de otras herramientas, permite:

- Gestión y administración de usuarios (alta, baja, modificación de datos).
- Administrar las opciones de configuración de la herramienta, qué se puede hacer, cuándo y por quién.
- Publicar noticias relacionadas con la herramienta, así como su envío a través del correo electrónico a los usuarios registrados.
- Registrar las estadísticas de uso referidas a la clase de ejercicios que ha realizado cada alumno y generar informes en forma tabular y gráfica.
- Gestionar un repositorio de ejercicios escritos en el lenguaje de entrada de SELFA.

4. Resultados Observados

En la etapa de evaluación de la herramienta los resultados han sido muy satisfactorios. SELFA está en funcionamiento desde el segundo cuatrimestre del Curso 05/06, momento en el que comenzaron las clases de la asignatura de TALF de la Ingeniería Técnica en Informática de Gestión. Los resultados de uso, de valoración por parte de los usuarios de este grupo y de utilidad pueden ser consultados en la Tabla 4.

Los datos muestran no sólo que los usuarios han utilizado y utilizan la herramienta, sino que también la consideran útil y fácil de usar.

En cuanto a la utilidad de la herramienta, tras un análisis de las calificaciones obtenidas en las

convocatorias de Junio y Septiembre por los alumnos que han usado la herramienta (25, ver Tabla 4), se ha comprobado que 24 de ellos han aprobado la asignatura, lo que supone el 75% del total de aprobados. Se puede concluir que SELFA ha ayudado en el aprendizaje de la asignatura de TALF. De los 18 aprobados en Junio, 14 habían usado SELFA. De los 10 aprobados de Septiembre, 10 habían usado SELFA y es por esta razón por lo que se pretende que siga en funcionamiento en los próximos cursos.

Variable	Curso 05/06
Alumnos registrados	45
Alumnos que la usaron	25
Archivos sometidos	1196
Ejercicios realizados	1449
Ejercicios de Gramáticas	
Elim. Produc. No Dev. Term.	134
Símbolos Inaccesibles	40 (2,76%)
Producciones Cadena Vacía	63 (4,35%)
Eliminación Cadena Vacía	60 (4,14%)
Eliminación Unitarias	104 (7,18%)
Forma Normal Chomsky	55 (3,8%)
Forma Normal Greibach	45 (3,11%)
Algoritmo CYK	3 (0,21%)
Ejercicios Autómatas	
Pasar a AFND	6 (0,41%)
Pasar a AFD	228 (15,73%)
Pasar a AFDM	144 (9,94%)
Complemento	44 (3,04%)
Igualdad	69 (4,76%)
Intersección	121 (8,35%)
Inverso	63 (4,35%)
Reconocimiento	232 (16,01%)
Unión	38 (2,62%)
Puntuaciones	
Media en Facilidad de Uso	4,389 sobre 5
Media en Valoración Total	4,417 sobre 5

Tabla 2. Resumen de las estadísticas de uso y valoración.

5. Conclusión

En este trabajo se ha mostrado el trabajo realizado en la ESI (UCLM) para mejorar la calidad docente de la asignatura de TALF.

Se ha presentado una herramienta cuyo diseño está basado en un procesador de lenguaje que se ejecuta en una arquitectura cliente/servidor que permite su uso a través de Internet, principal ventaja de *SELFA* frente a otras herramientas diseñadas con el mismo propósito. El principal objetivo de la herramienta es permitir la propuesta de ejercicios sobre diferentes temas de la asignatura y la resolución de los mismos.

Esta herramienta puede ser utilizada tanto por el profesor para su uso en las clases magistrales como por los alumnos para su uso en el estudio de los conceptos de la materia.

Respecto a trabajos futuros, se pretende ampliar esta herramienta para conseguir cubrir toda la materia de la asignatura, añadiendo expresiones regulares, autómatas a pila y máquinas de Turing.

Agradecimientos

A la ESI, a la UCLM y a los alumnos del curso 05/06 por sus sugerencias y apoyo a este proyecto.

Referencias

[1] http://www.aneca.es/modal_eval/docs/libroblanco_jun05_informática.pdf

[2] http://europa.eu.int/comm/education/programmes/Sócrates/ecos_en.html

[3] Diez, J.L., Diaz J. JavaPars. http://paginaspersonales.deusto.es/josuka/jparser/parser.html

[4] Hopcroft, J.E., Motwani, R., Ullman, J.D. *Introduction to automata theory, languages and computation*. Addison Wesley, 2001.

[5] Jodar, J.F., Revelles, J. SEFALAS, http://lsi.ugr.es/~pl/software.php

[6] Kelley, D. *Automata and Formal Languages: An Introduction*. Prentice Hall, 1998.

[7] Lewis, H.R., Papadimitriou, C.H. *Elements of theory of computation*. Prentice-Hall, 1997.

[8] Martin, J.C. *Introduction to languages and the theory of computation*. McGraw-Hill, 2003.

[9] Santos, P.A., Castro-Schez, J.J. Una herramienta para la enseñanza y aprendizaje de la asignatura Procesadores de Lenguajes. *XII Jornadas de la Enseñanza Universitaria de la Informática*, 499-506, Bilbao, 2006.

[10] Santos, P.A., Castro-Schez, J.J. PROLETOOL, http://oreto.inf-cr.uclm.es/proletool/

[11] Hortolano, J., Castro-Schez, J.J., Castillo, E., SELFA, http://apps.oreto.inf-cr.uclm.es/SELFA/

[12] Rodger, S., Finley, T. JFLAP, http://www.jflap.org

[13] Proyecto SEPa!

Herramienta para la gestión de software y enlaces de Investigación Operativa

Victoria Fernández, Ana Zelaia, Pablo Aramendia
Dpto. de Ciencia de la Computación e I A
Facultad de Informática
Universidad del País Vasco
victoria.fernandez@ehu.es, ana.zelaia@ehu.es

Resumen

En el proceso de adaptación al Espacio Europeo de Educación Superior será necesario modificar las metodologías utilizadas en la enseñanza para adecuar las asignaturas a la medida del crédito europeo, no solo en términos de la duración de las clases sino incluyendo también las horas de trabajo personal del alumnado. Para realizar este proceso es necesario elaborar materiales didácticos dirigidos a potenciar el trabajo guiado del alumnado, apoyándose en el uso de las nuevas tecnologías.

El objetivo de este trabajo ha sido doble. Por una parte, se ha realizado una revisión de software y enlaces relacionados con la investigación operativa con el objeto de recopilar aquellos que resulten más apropiados para la labor docente. Por otra parte, se ha desarrollado una aplicación que contiene una base de datos que permite gestionar el software y los enlaces de interés de forma personalizada, para que el docente saque el mayor partido posible de la información recogida. Esta propuesta puede ser fácilmente trasladada a otras materias para las que exista mucho software disponible.

1. Motivación

Para pasar de un sistema basado en la enseñanza del profesorado a otro basado en el aprendizaje del alumnado, una manera de potenciar el trabajo personal es proponer la realización de trabajos guiados que permitan lograr una mejor comprensión de los conceptos teóricos. Dependiendo de la naturaleza de cada asignatura, habrá que diseñar diferentes tipos de trabajos.

En algunas asignaturas, como en la Investigación Operativa, se estudian técnicas algorítmicas de resolución de problemas que fácilmente pueden ser abordadas a través de la realización de laboratorios y trabajos guiados mediante el uso de software específico. De hecho, la oferta de dicho software para Investigación Operativa y más concretamente para Programación Lineal es muy amplia. Sin embargo, puede resultar difícil seleccionar los programas más apropiados en cada caso, dado que no todos los paquetes resultan útiles para que el alumnado los utilice de forma autónoma con fines didácticos.

La principal motivación para la realización de este trabajo ha sido hacer una revisión del software existente para ayudar a el/la docente en la elección de los paquetes que resulten más apropiados para plantear laboratorios o trabajos guiados, al mismo tiempo que sirve de ayuda al alumnado para hacer sus propias elecciones. Por otra parte, se ha podido comprobar que la mayoría de los paquetes carecen del desarrollo de muchos de los aspectos didácticos importantes para facilitar el aprendizaje de la asignatura.

Una característica de la mayoría de las técnicas y algoritmos que se estudian en Programación Lineal es que pueden ser mostradas gráficamente para una visión más intuitiva. Si el modelo tiene sólo dos variables es posible mostrar gráficamente el conjunto de soluciones y la solución óptima, tanto de un problema lineal general como de uno de programación entera. Asimismo, se puede mostrar gráficamente la evolución de algunos algoritmos en la búsqueda de soluciones óptimas, así como el análisis de sensibilidad, etc.

En cuanto al nivel de interacción, el/la usuario/a puede llegar a una mejor comprensión de las técnicas algorítmicas si los programas permiten una participación más activa en el proceso de resolución, haciendo posible la selección manual de criterios, para poder comprobar qué ocurre cuando no son respetados.

De lo contrario, el/la usuario/a resulta un/a mero/a observador/a.

Para un mejor aprovechamiento del software existente y para identificar aspectos didácticos que están sin desarrollar, hemos creado la aplicación informática que presentamos en este artículo. De forma complementaria, se han añadido enlaces de interés sobre la asignatura.

2. Antecedentes

En los últimos años ha aumentado de forma notable el software para resolver problemas de optimización y, en particular para Programación Lineal. En general, los paquetes son fáciles de manejar y, en algunos casos, con una didáctica que ofrece la posibilidad de abordar personalmente el estudio y la comprensión de los métodos implementados. Sin embargo, hay diferencias entre los paquetes haciendo unos preferibles a otros dependiendo del contexto. Algunos antecedentes a este trabajo de revisión muestran aspectos destacables de los distintos paquetes.

En [1] se analizan cinco paquetes de software comercial centrándose en los siguientes aspectos: tamaño de los problemas, algoritmos, interfaz, conexión a otros programas y didáctica. Un estudio más exhaustivo [4] analiza más de cuarenta paquetes y más aspectos que el anterior, aunque pasa desapercibido si contiene algunos aspectos didácticos en los que queremos hacer hincapié. La lista de paquetes analizada en [5] es mucho más exhaustiva pero la información que detalla sobre paquetes es muy genérica, obligando a visitar las páginas oficiales de las empresas si se quiere conocer las posibilidades que ofrece el software.

Los trabajos mencionados anteriormente recogen en tablas los distintos aspectos del software analizado. La aplicación que presentamos en este artículo tiene muchas similitudes con dichos trabajos aunque presenta algunas diferencias. Las fundamentales son: (1) que se aporta una base de datos con la que poder gestionar y personalizar la información referente a los diferentes software y (2) que se hace un especial hincapié en analizar aspectos docentes. Además, se incluyen enlaces a páginas web relacionadas con la Investigación Operativa.

3. La Aplicación

El objetivo principal de esta aplicación es mostrar las diferencias, sobre todo las didácticas, entre los distintos paquetes para poder elegir de manera rápida y sin necesidad de analizarlos exhaustivamente. Un objetivo complementario es poder determinar las carencias didácticas del software analizado que pueden ser objeto de estudios posteriores.

La aplicación ha sido diseñada para gestionar una base de datos que contiene dos tipos de información: programas y enlaces y corresponde al proyecto fin de carrera de Ingeniería Técnica en Informática de Sistemas del alumno Pablo Aramendia. La implementación ha sido desarrollada en el entorno NetBeans 5.5 para los sistemas operativos windows 2000, windows XP y linux. Se ha creado el archivo Soft_IO.jar que se puede ejecutar con un intérprete de Java y que está disponible en la página web de la asignatura [2]

Figura 1: Pantalla Principal

La pantalla principal con la que se inicia la aplicación permite seleccionar el idioma entre euskara y castellano, tal como se puede observar en la Figura 1.

Una vez seleccionado el idioma aparece una portada explicativa del uso de la aplicación. En la parte superior de dicha portada se tiene la barra de herramientas que permite gestionar la base de datos (ver Figura 2).

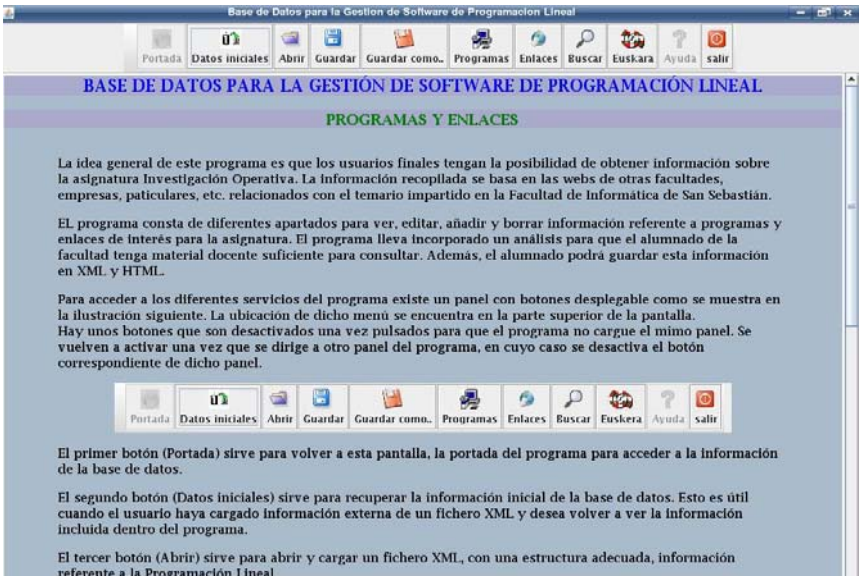


Figura 2: Portada explicativa y barra de herramientas

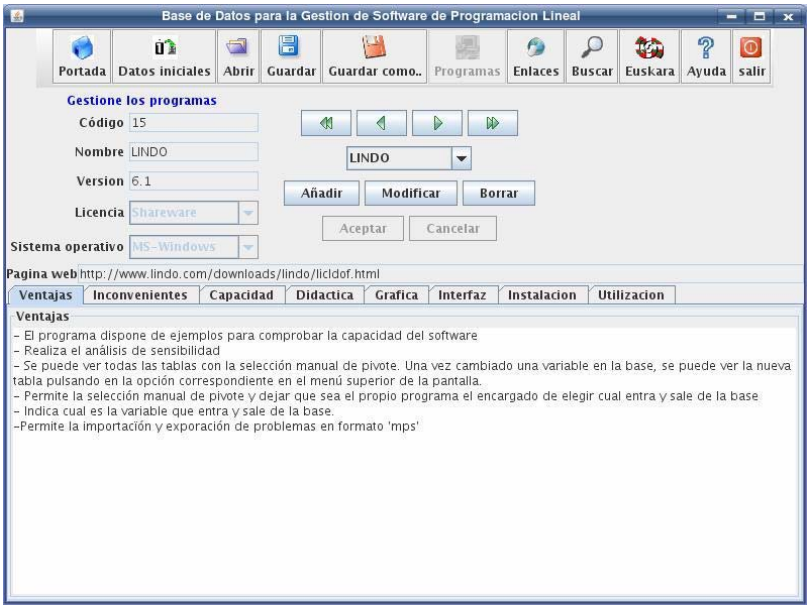


Figura 3: Pantalla que corresponde a la gestión de programas

Esta barra está visible en todo momento, y pulsando sobre los botones *programas* y *enlaces* se accede a la gestión de los mismos. Además, en la barra de herramientas hay disponibles otros botones que tienen utilidades como, por ejemplo, realizar búsquedas o acceder a la ayuda.

A continuación se analizan de forma más detallada los paneles con los que se accede a la gestión de programas y enlaces.

3.1 Gestionar programas

Al pulsar sobre el botón *programas* se accede al panel de gestión de software de la base de datos. Dicho panel permite consultar los distintos programas informáticos almacenados, y personalizar la base de datos eliminando/ insertado programas o modificando la información recogida sobre ellos, tal como se puede observar en la Figura 3. En la actualidad, la base de datos dispone de 26 programas.

En la parte superior de este panel aparecen una serie de botones que permiten avanzar y retroceder. De esta manera, se seleccionará un programa y toda la información correspondiente al mismo aparecerá en pantalla.

A la izquierda de los botones se tiene la información general: nombre del programa, sistema operativo sobre el que se puede ejecutar, dirección web en la que está disponible, etc.

En la parte inferior del panel se puede encontrar información adicional sobre los programas incluidos en la base de datos. Concretamente, los resultados de este análisis se han recogido en las siguientes ocho categorías: ventajas, desventajas, capacidad, didáctica, gráfica, interfaz, instalación y utilización

Estas características han sido seleccionadas por los/las autores/as de este artículo precisamente para analizar si el programa resulta adecuado desde el punto de vista docente y pueden ser observadas con solo pulsar en la pestaña correspondiente.

Sin embargo, la aplicación no está pensada para ser utilizada únicamente como herramienta de consulta de programas; en realidad, sirve como base para que las personas interesadas en software para Investigación Operativa puedan gestionar la información que crean interesante y puedan personalizarla con todo tipo de comentarios. Para ello se dispone de los botones de añadir, modificar y borrar, que permiten añadir nuevos programas, modificar la información existente o borrar el programa. El objetivo final es que el/la usuario/a pueda ordenar la información de la forma más adecuada posible para poder sacar partido al software existente realizando laboratorios, proponiendo prácticas y trabajos guiados al alumnado.

3.2 Gestionar enlaces

Al pulsar sobre el botón *enlaces* se accede al panel de gestión de enlaces de interés para la Investigación Operativa. Al igual que se ha procedido con la gestión de programas, este panel permite añadir, modificar y borrar enlaces que el/la usuario/a considera de interés. En la actualidad, la base de datos dispone de 27 enlaces. El aspecto del panel de enlaces es muy parecido al panel de programas, tal como se puede observar en la Figura 4.

En la parte superior se tienen los botones de avance y retroceso que permiten desplazarse a través de la base de datos para seleccionar el enlace deseado. A la izquierda de dichos botones se muestra el nombre y la dirección de la página web.

Dado que un texto explicativo sobre la información que alberga una página web podría ser poco explícito, se ha optado por mostrar la propia web directamente desde la aplicación. Para facilitar la consulta se ha implementado un navegador web sencillo que permite desplazarse con más facilidad. Las páginas web se cargan en un JEditorPane que sólo tiene compatibilidad garantizada con la versión 3.2 de la especificación de HTML. Algunas funcionalidades de la versión 4.01 de HTML funcionan como los frames. Sin embargo, otras componentes como JavaScript, ActiveX o applets no son reconocidas porque se incluyeron en especificaciones posteriores a la versión 3.2 de HTML [6][7][8].

Además de poder acceder y observar directamente la página web, se ha considerado conveniente incluir un apartado donde poder

escribir comentarios sobre la información de interés.

De la misma manera que se ha procedido con los programas, también se da la opción de gestionar los enlaces a través de botones para añadirlos a la base de datos, modificar algún dato sobre algún enlace o borrarlo. De esta manera, el/la usuario/a guardará sólo los enlaces que cree son de su interés.

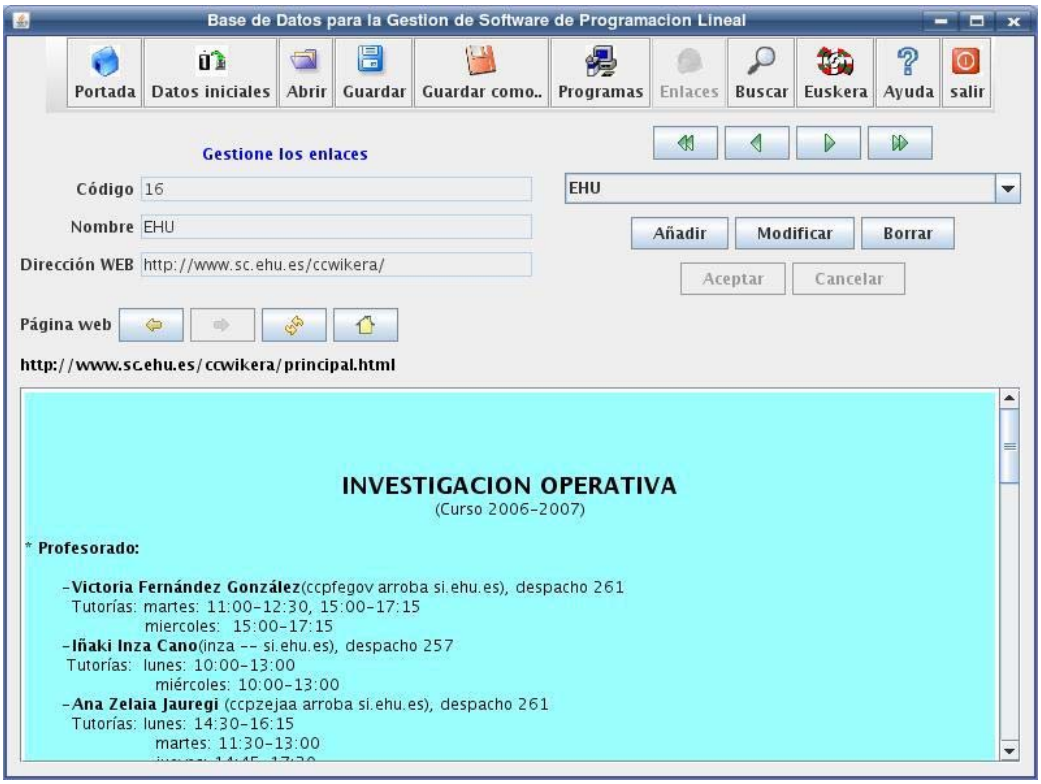


Figura 4: Pantalla que corresponde a la gestión de enlaces

4. Conclusión

La principal aportación del trabajo presentado en este artículo consiste en haber desarrollado una herramienta informática mediante la cual gestionar una base de datos creada específicamente para almacenar información relevante para Investigación Operativa. El enfoque de la información almacenada es básicamente docente. El objetivo que se pretende alcanzar es que la herramienta ayude a docentes y alumnado a encontrar el software que se adecua a sus necesidades. La herramienta queda a disponibilidad de los/as interesados/as a través de la página web de la asignatura [2].

Como conclusión, queremos destacar que, tras el exhaustivo análisis realizado, encontramos que se podría desarrollar un software con una didáctica más clara. Concretamente, creemos que

sería interesante que este software incluyera un desarrollo gráfico más amplio y ésta es nuestra línea actual de trabajo. Un ejemplo de ello es la aplicación *simplex.jar* que permite observar gráficamente la evolución del algoritmo simplex cuando el modelo tiene sólo dos variables. Esta aplicación fue presentada en JENUI'06 [3] y es accesible a través de la página web de la asignatura. Actualmente estamos abordando aplicaciones de este tipo para la programación entera, el algoritmo simplex dual y el análisis de sensibilidad.

En un futuro nos planteamos ampliar el nivel de interacción de las aplicaciones para que, aparte de la resolución automática de los problemas, permitan la elección de criterios en el proceso de resolución. Esto hará posible contrastar si las decisiones tomadas son las más apropiadas, y tener una actitud más participativa.

Referencias

[1] Canós Darós, M.J., Ventura Marco, M. *Una Evaluación de Software Comercial para la Optimización de Modelos Lineales en Economía* http://www.uv.es/asepuma/VI/14.PDF

[2] Fernández, V., Inza, I., Zelaia, A. Página web de la asignatura de Investigación Operativa de la Facultad de Informática de la Universidad del País Vasco. http://www.sc.ehu.es/ccwikera/principal.html

[3] Fernández, V., Urdangarin, I., Zelaia, A. *Herramienta gráfica para el aprendizaje del algoritmo simplex*. Actas de las XII Jornadas de Enseñanza Universitaria de la Informática, JENUI 2006, pág: 531-535

[4] Web: Linear Programming Survey. http://www.lionhrtpub.com/orms/surveys/LP/LP-survey.html

[5] Web: NEOS server for Optimization. http://www-fp.mcs.anl.gov/otc/Guide

[6] Web: Java API para procesar XML. http://java.sun.com/xml

[7] Web: Componentes gráficos de Java. http://java.sun.com/docs/books/tutorial/uiswing

[8] Web: http://www.w3schools.com

Simuladores de Jerarquía de Memoria en el Contexto de un Proceso de Investigación-Acción

L. Moreno, E.J. González, B. Popescu, J. Torres, J. Toledo, C. González
Dpto. de Ingeniería de Sistemas y Automática y Arquitectura y Tecnología de Computadores
Universidad de La Laguna
Facultad de Física y Matemáticas, 38207 Tenerife
ejgonzal@ull.es

Resumen

Presentamos en esta ponencia dos simuladores para la docencia de la jerarquía de memoria en varias asignaturas, productos de un proceso de investigación-acción en el cual también se ha originado un mapa conceptual para una mejor coordinación a los profesores de las asignaturas involucradas. En el diseño de ambos simuladores se ha perseguido abarcar todos los conceptos de estudio sin perder la facilidad de uso ni el objetivo pedagógico.

1. Motivación

Los profesores de las asignaturas de Estructura de Computadores, Arquitectura de Computadores y Sistemas Operativos de la titulación de Ingeniero en Informática hemos iniciado un proceso de diagnóstico de nuestras asignaturas que pretende formular estrategias para resolver los problemas en la impartición de algunos de los conceptos comunes a estas disciplinas. Se trata de establecer una adecuada relación entre la teoría y la práctica educativa, aplicando las aportaciones que provienen de las teorías educativas y del conocimiento práctico de los profesores.

Para llevar a cabo este proceso, hemos elegido inicialmente una materia recurrente en las tres disciplinas mencionadas: la Jerarquía de Memoria, considerada en su doble aspecto (Memoria Virtual y Memoria Cache). En este punto, debemos recordar que precisamente la enseñanza de la cache fue el Tema Estratégico de las JENUI celebradas en Palma de Mallorca en 2001 y objeto de estudios posteriores [2], lo que demuestra la importancia de esta materia en la enseñanza universitaria de la Informática.

Para el análisis de los conceptos involucrados en la Jerarquía de Memoria, hemos realizado un proceso de investigación–acción, es decir, un proceso secuenciado que parte de la búsqueda y diagnóstico de problemas, continúa con la formulación de estrategias para solucionarlos, los pone en práctica, y los evalúa. Con todo ello se plantea una nueva formulación y descripción del problema. En la primera fase del proceso, nos hemos reunidos todos los profesores involucrados, y hemos realizado el mapa conceptual de todos los conceptos que deberán ser impartidos en las tres asignaturas mencionadas

Los mapas conceptuales, creados por Novak en los años 60, son unas redes de conceptos, donde los nodos representan los conceptos y los enlaces las relaciones entre dichos conceptos. Es una herramienta que se constituye en un instrumento eficaz para ayudar a los alumnos a reflexionar sobre la estructura y el proceso de producción del conocimiento. Los mapas conceptuales representan una estructura que va de los conceptos más generales a los más específicos, facilitando la ordenación y secuenciación jerárquica de los contenidos de enseñanza, de forma que ofrecen estímulos adecuados al alumno [6]. Recordemos que en este sentido la motivación en Arquitectura de Computadores ya ha sido objeto de estudio en diversos trabajos como en [2].

En el tema de la Jerarquía de memoria, el uso del mapa conceptual (consultable en la dirección http://www.isaatc.ull.es/portal/proyectos/mneme/jerarquia.jpg) nos permitirá explorar estrategias de mejora en la enseñanza de la misma. En este proceso hemos analizado cuáles son los conceptos a abordar, llegando a las siguientes conclusiones:

- Surgen una gran cantidad de conceptos que definen a su vez una gran cantidad de parámetros a dimensionar: políticas locales/globales, procesos, colas de espera/ejecución, algoritmos de reemplazamiento, tamaño de los buses entre los diferentes niveles de cache, tamaño de las caches, tamaño de la página en memoria

virtual y de las líneas en las caches, write allocate/non write allocate, tipos de tablas de página, tamaño de la TLB, etc.
- Debemos contemplar todas las características que nos lleven a situaciones de máquinas existentes: niveles de cache, implementaciones de la tabla de página, algoritmos de reemplazamiento, etc.

El segundo paso en el proceso de investigación-acción ha sido la creación o utilización de simuladores que faciliten la comprensión de los conceptos. El uso de simuladores como recurso didáctico se encuentra muy extendido, ya que ofrece entre otras las siguientes ventajas [5]:

- El alto número de alumnos matriculados en determinadas asignaturas conlleva un aumento de carga docente en prácticas. Los simuladores contribuyen a atenuar este efecto sin que la calidad de la docencia disminuya.
- El desarrollo de plataformas educativas *on-line* puede facilitar su uso y su control mediante un programa de prácticas y tutoriales adecuado. En este caso emplearemos además el mencionado mapa conceptual, el cual los alumnos han colaborado a complementar asociando a cada concepto la teoría correspondiente y los recursos relacionados.

No vamos a entrar en la discusión sobre si deben emplearse simuladores sencillos o simuladores complejos [2]. Diremos que un simulador no puede hacer una simplificación muy grande de la realidad con objeto de hacerlo sencillo. Si la realidad es compleja con multitud de parámetros y con necesidad de mostrar el funcionamiento de una gran variedad de conceptos, entonces el simulador resultará complejo y habrá que dotarlo de mecanismos para hacerlo útil pedagógicamente.

Proponemos, como conclusión primera de este segundo paso, un simulador realizado en nuestra Universidad, que reúne en la Jerarquía de Memoria tanto la memoria virtual como la cache multinivel. Dicho simulador se denomina SIJEM, concebido como una aplicación de escritorio, realizado en C++. Asimismo y también como conclusión de esta etapa hemos decidido crear un nuevo simulador que permita a los alumnos el aprendizaje en situaciones más complejas como son las que se presentan en las asignaturas de Sistemas Operativos y Arquitectura de Computadores. Este simulador lo hemos denominado MNEME. Ambos recursos pueden ser descargados de manera gratuita de la página web de nuestro Departamento [4][7].

Como tercer paso del proceso de investigación-acción hemos sometido ambos simuladores a la validación de profesores y alumnos de 5º curso de la titulación de Ingeniero Informático. Fruto de esta evaluación surgió la cuarta fase de la estrategia consistente en una reunión conjunta de profesores y alumnos. Esta reunión dio lugar a un conjunto de ideas para mejorar las prestaciones del simulador MNEME, haciéndolo más accesible a los alumnos e incorporando los sistemas multiprocesador y multicore. Ello ha dado lugar a la definición de un nuevo simulador paramétrico y reconfigurable en función de los conocimientos de los alumnos, con lo que hemos llegado a una nueva formulación del problema investigado en el proceso de investigación-acción.

2. Características del Simulador SIJEM

SIJEM [3],[7] es un simulador de jerarquía de memoria cuyo objetivo es servir de apoyo al aprendizaje de los conceptos relacionados con memoria virtual y memorias cache. A través de una interfaz altamente visual SIJEM trata de ilustrar los conceptos de:

- Memoria Virtual: Traducción de direcciones virtuales a direcciones reales; Paginación; Uso de tabla de páginas y TLBs (Translation Lookaside Buffer); Estrategias de búsqueda, colocación y reemplazamiento (FIFO, LRU, Clock, LFU, NUR, etc.) en memoria principal.
- Niveles de memoria: Memoria secundaria; Memoria principal; Memorias caché multinivel (Nivel 3, Nivel 2 y Nivel 1 Conjunta o Separada en datos e instrucciones); Estrategias de colocación, reemplazamiento (FIFO, LRU, Clock, LFU, NUR, etc.) y coherencia entre los diferentes niveles.

El simulador posee una gran cantidad de parámetros configurables, tales como tamaños de las diferentes memorias, bloques y páginas,

tiempos de acceso, caches habilitadas o deshabilitadas, algoritmos de colocación, reemplazamiento y coherencia, etc., disponiendo de un asistente que guía al alumno en él. Este asistente esta dividido en tres fases:

1. *Configurar*, a través de una serie de pasos, guía al usuario en el proceso de configurar todos los parámetros del simulador.
2. *Cargar fichero*. Aquí el usuario elige el fichero de traza entre los múltiples ejemplos incluidos en el simulador o bien los propios ficheros creados por él mismo.
3. *Simular.* En este punto, el usuario elige el tipo de simulación a realizar.

Además se incluye el botón *Resumen configuración*, que muestra en cualquier momento un resumen de la configuración elegida durante el proceso de configuración.

2.1. Configurar

Para guiar al usuario en el proceso de configurar el simulador, los parámetros fueron agrupados según la siguiente secuencia:

1. Memoria virtual / secundaria.
2. Memoria principal.
3. Tamaño de página.
4. Mecanismo de traducción de direcciones.
5. Parámetros de paginación.
6. Tamaño del bloque de caché.
7. Caché de nivel 3 (Tamaño, organización, número de vías, política de reemplazamiento).
8. Caché de nivel 2 (Tamaño, organización, número de vías, política de reemplazamiento).
9. Caché de nivel 1 (Tamaño, organización, número de vías, política de reemplazamiento).

Otra facilidad del simulador en este proceso es la opción de cargar ficheros de configuración creados a mano por el usuario o incluidos como ejemplo, con todos los parámetros necesarios para la simulación. Los ficheros de configuración son ficheros de texto con extensión *.cfg*. Estos ficheros pueden cargarse al inicio del asistente, de manera que los parámetros de las diferentes pantallas son introducidos automáticamente, aunque el usuario tiene la posibilidad de cambiarlos.

2.2. Cargar Fichero

Para realizar una buena simulación es necesario tener ficheros de traza que se correspondan con programas reales. La solución adoptada fue obtenerlos a través del portal de distribución de trazas del PEL (*Performance Evaluation Laboratory*) de la Bringman Young University en Washington. Dado que se trataba de efectuar simulaciones que mostraran las características de diferentes tipos de programas se optó por incluir dentro del simulador las trazas correspondientes a 6 tests del benchmark SPEC2002.

2.3. Simulación

Una vez realizada la configuración y elegido el fichero de traza es el momento de comenzar la simulación. La simulación se divide en dos partes: Traducción de direcciones y Búsqueda de páginas.

Traducción de direcciones

Dependiendo de la configuración elegida, se pueden mostrar tres interfaces diferentes, uno para cada tipo de mecanismo de traducción: a) Traducción de direcciones por transformación directa; b) Traducción de direcciones por transformación asociativa-directa con TLB y c) Traducción de direcciones por transformación asociativa-directa con TLB dividida en datos e instrucciones

Todos ellos comparten los siguientes elementos: Dirección virtual (tanto en binario como en decimal, dividida en número de página y desplazamiento dentro de la página); Dirección real (tanto en binario como en decimal), Dirección base de la tabla de páginas y Tabla de páginas.

Búsqueda de páginas

Dependiendo de la configuración elegida puede llegar a tener los siguientes componentes: Memoria secundaria, Memoria principal, Caché de nivel 3, Caché de nivel 2, Caché de nivel 1 conjunta o separada en datos e instrucciones y CPU.

La simulación se divide en una serie de pasos que muestran el estado de la memoria en ese instante, destacando las posiciones afectadas, la acción realizada o el suceso ocurrido, y una serie de estadísticas. Una de las características

innovadoras de este simulador es que hemos incluido la posibilidad de ir hacia delante y hacia atrás en el proceso de la simulación. Este método permite que los usuarios, en el caso de que den un paso hacia adelante y no comprendan lo sucedido, puedan retroceder uno o varios pasos para tener una visión global del suceso.

Finalmente se han incluido colores como elemento de ayuda al alumno para facilitar la comprensión de lo que está ocurriendo en cada momento. El color verde indica "éxito" mientras que el color rojo indica "fallo" en la búsqueda de páginas en cada uno de los niveles de la jerarquía. Esta funcionalidad ha sido añadida a requerimiento de los alumnos que han evaluado la herramienta.

3. Simulador MNEME

Como dijimos anteriormente, para las asignaturas de Sistemas Operativos y Arquitectura de Computadores hemos diseñado un simulador que trata de superar las capacidades de SIMJE, permitiendo experimentar en todos y cada uno de los conceptos que dichas asignaturas estudian en lo referente a jerarquía de memoria. Este simulador se denomina MNEME [4] y se ha implementado en Java Swing. A continuación analizaremos cuáles son las prestaciones de este nuevo simulador en función de la asignatura a la que va destinado:

En Sistemas Operativos:

- Definición de procesos asociados a un número de unidades de tiempo de ejecución e implementación de las colas de espera y de ejecución.
- Algoritmos de reemplazamiento basados en políticas globales, haciendo referencia a todos los procesos: working set (page aging) y Page fault frequency (PFF), basado en dos parámetros max PFF y min PFF. Cuando un proceso es superior a maxPFF, entonces el mecanismo memory allocation toma, de forma aleatoria, una página de un proceso que tenga su page fault frequency más bajo que minPFF.
- Algoritmos de reeemplazamiento basados en políticas locales, haciendo referencia exclusivamente a páginas del propio proceso: LRU, FIFO, Clock, LFU, MIN u OPT, NUR, NFU, MRU.
- Tabla de página con las implementaciones Top-down, Bottom- up e inversa.

En Arquitectura de Computadores:

- Niveles de cache: L1, L2 y L3 independientes, con un tamaño de línea específico para cada nivel.
- Dimensión de los buses.
- Write Allocate/non write allocate.
- Paso del paso: procedimiento gráfico de descomposición en pasos más elementales del proceso de búsqueda de una instrucción/dato.

3.1. Prácticas con el simulador MNEME

En este punto será tratada una secuencia de prácticas tipo para la asignatura de Sistemas Operativos de Segundo Curso de Ingeniería Técnica en Informática utilizando las capacidades docentes que aporta el simulador MNEME. Las prácticas versarán sobre los distintos algoritmos de paginación que pueden ser utilizados en un sistema operativo, poniendo de manifiesto sus particularidades a través del simulador.

Familiarización con el simulador

La primera práctica básica tiene como objetivo principal que el alumno aprenda a utilizar el simulador MNEME. Para ello se planteará el caso más sencillo de paginación, donde el alumno empezará a utilizar el programa en un caso muy conocido para él, fácil de seguir y predecir. En este primer caso se planteará un sistema de paginación jerárquica de dos niveles y un único proceso que utilice esta memoria. El código del proceso consistirá en una serie de accesos a memoria, algunos produciendo fallos de página y otros accediendo a paginas ya disponibles en memoria principal.

Tras esta primera sesión de prácticas el alumno debería haber aprendido a manejar el simulador, entender todas las tablas y gráficos, además de ser capaz de seguir una traza de accesos a memoria desde el código fuente del proceso al funcionamiento del simulador.

Paginación Multiproceso

En la segunda práctica se presenta un sistema multiproceso con el mismo esquema jerárquico de

paginación, consiguiendo introducir al alumno en el simulador cuando entran en escena varios procesos accediendo simultáneamente a memoria. Para esto hay que definir el quantum de tiempo que tiene cada proceso y realizar la traza de los accesos a memoria de cada uno de los procesos. Para seguir esta traza hay que tener en cuenta que un proceso será expropiado de la CPU en el caso de que se produzca un fallo de página o se acabe su quantum de tiempo de ejecución.

Distintas estrategias de la gestión de páginas

A continuación, una vez que el simulador de memoria y la estructura de procesos sean bien conocidos, se buscará profundizar en los distintos algoritmos de gestión de páginas. Para ello, se verá cómo cambia el comportamiento del sistema con los distintos algoritmos de sustitución de página. En este cometido se utilizará un código con un gran número de fallos de página y se definirá un tamaño de memoria principal pequeño para forzar a que el simulador envíe a memoria secundaria las páginas seleccionadas. Se probarán tanto estrategias de intercambio locales como globales explotando las posibilidades del simulador.

Sistemas de paginación avanzada

La última práctica se centrará en algoritmos avanzados de paginación como la "tabla de páginas inversa", el envejecimiento de páginas y la tabla de páginas lineal en memoria virtual, facilitando al alumno la comprensión de estas técnicas de gestión de memoria que son nuevas para él. También se utilizará la TLB y se comprobará el efecto de la variación del tamaño de ésta para la eficiencia del sistema.

Por último y de manera meramente informativa se realizará una simulación utilizando todas las características del programa, haciendo el seguimiento del dato desde la memoria. Se comprobará la funcionalidad de los distintos niveles de cache de la CPU y el funcionamiento global del sistema, abarcando desde la visión del sistema operativo hasta la visión de la estructura de la máquina.

Esta sesión de prácticas se ha centrado en el estudio de la paginación, obviando otras funcionalidades que aporta el simulador (que serán utilizadas en otras asignaturas como el funcionamiento de las caches multinivel en Arquitectura de Computadores). Un simulador que permite la simulación de una amplia gama de técnicas permitirá aumentar la visión global del alumno en el funcionamiento de Arquitectura de Computadores desde el hardware hasta el sistema operativo.

3.2. Proceso de Validación

El Simulador SIMJE fue realizado en el año 2004, año en el que fue validado por los alumnos de ese curso académico, y se hicieron propuestas de mejora dirigidas fundamentalmente a la interfaz con el usuario. Desde entonces y una vez mejorado, ha sido utilizado ininterrumpidamente con resultados satisfactorios en los sucesivos cursos hasta la actualidad.

Sin embargo, en este apartado nos centraremos en el simulador MNEME. Para validarlo se dedicaron 4 grupos de alumnos de 5º curso de la Ingeniería Informática, con amplia experiencia en la utilización de simuladores y en programación. Realizaron una validación de tipo colaborativo por grupo, aunque cada miembro del grupo tuvo que completar una encuesta realizada de forma individual. En total hemos analizado un conjunto de 10 encuestas. Los resultados se comentan a continuación.

El 100% de los alumnos afirmó que al menos probaron el programa MNEME entre 3 y 6 horas antes de contestar el cuestionario. Dicho cuestionario constaba de una serie de preguntas referentes tanto a aspectos educativos y técnicos como de funcionalidad del software. También se les ofreció la oportunidad de que indicaran cuáles eran las mejoras que a su juicio deberían ser implementadas en el programa. Sobre este punto es importante destacar que el 90% afirmó que no conocían software alguno que fuera similar al nuestro.

En el aspecto educativo los alumnos afirmaron en un 90% que el software ayuda a comprender mejor los contenidos, mientras que el 70% dijeron que además es capaz de mantener el interés. Sin embargo también nos confirmaron que es necesario pulir la interfaz de usuario, pues el 100% de los usuarios consideró que el esquema de máquinas no se entendía fácilmente, a la vez que el 90% afirmó que la ayuda suministrada no era lo suficientemente clara. Es más, desde el punto de

vista de la funcionalidad del software, el 40% de los alumnos dijo que el contenido de la memoria no se muestra con suficiente detalle y claridad. Sin embargo es importante destacar que la inmensa mayoría de los encuestados consideraron que el programa tiene las opciones necesarias para ser lo suficientemente flexible de cara a diseñar nuevos ejercicios. En lo que respecta al aspecto técnico hay un consenso generalizado en que el programa funciona correctamente. Es decir, el 100% de los usuarios afirma que el programa no se cuelga y se ejecuta perfectamente en distintos ordenadores.

Finalmente se les pidió a los alumnos que indicaran aquellos aspectos que eran mejorables y propusieran algunas modificaciones. La mayor parte de los encuestados afirmaron que el software era interesante y que tiene un gran potencial de cara a su formación; al mismo tiempo que indicaron la necesidad de mejorar la experiencia del usuario. En este sentido los alumnos se inclinaron por mejorar la ayuda mediante la inclusión de un asistente que iniciara al usuario en la utilización del programa y la utilización de globos emergentes que informaran acerca de lo que está pasando en cada instante de la simulación.

Como consecuencia de esta evaluación realizamos una reunión conjunta los profesores implicados con el simulador y los alumnos que realizaron la encuesta. Fruto de esta reunión han surgido una serie de ideas innovadoras encaminadas a:

- mejorar la interfaz.
- crear un tutorial de utilización del simulador.
- crear una herramienta que permita de forma gráfica ensayar diferentes estrategias de caché multinivel, e incorporarlas al sistema como metadatos XML.
- incorporar en cada nivel de cache su inclusividad/exclusividad.
- incorporar al simulador la capacidad multiprocesadora y multicore.
- crear una herramienta asociada con el simulador que posibilite al profesor ocultar, dependiendo de los conocimientos del alumno, determinadas funcionalidades del programa.

Estas ideas ya se están abordando en estos momentos, con lo que se inicia nuevamente el proceso de investigación-acción.

4. Conclusión

A través de un proceso de investigación-acción hemos tratado de analizar rigurosamente cuáles son algunos de los problemas que nos encontramos en la impartición de las asignaturas de Estructura de Computadores, Sistemas Operativos y Arquitectura de Computadores, centrándonos en el tema de la Jerarquía de memoria. Hemos realizado una estrategia que ha consistido en 4 fases: análisis de los conceptos involucrados y creación de un mapa conceptual, una segunda fase de creación de un simulador MNEME que permita trabajar con todos los conceptos involucrados con la Jerarquía de memoria y que permita al alumno un aprendizaje constructivista. Hemos incorporado al proceso otro simulador sencillo y amigable, denominado SIMJE. El proceso se ha completado con dos fases más de validación del programa MNEME por parte de los alumnos a través de la realización de un cuestionario y una reunión mantenida entre profesores y alumnos que han permitido definir en el simulador nuevas características referidas a la interfaz, a la configuración de las caches multinivel, tanto en un medio monoprocesador, como multiprocesador y/o multicore.

Referencias

[1] Almisas, R.; Paz, R.; Linares, A.; Amaya, C. y Sevillano, J.L. “Un simulador de memorias cache multinivel”. Actas de las VII Jornadas de Enseñanza Universitaria de Informática, Jenui 2001, 429-433.

[2] Anguita, M.; Fernández, A..; Linares, A.F.; Díaz, A.; Cañas A. y Prieto, A. “Práctica de optimización para asignaturas de Escritura de los Computadores”. Actas de las X Jornadas de Enseñanza Universitaria de Informática, Jenui 2004.

[3] González C., Alesanco F., Castilla I., Moreno L. “SIJEM: Una herramienta didáctica para la enseñanza de la Jerarquía de Memoria”. 4ta. Conferencia Iberoamericana en Sistemas, Cibernética e Informática. CISCI 2005. Volumen III. Julio. 2005.

[4] MNEME puede descargarse de: http://www.isaatc.ull.es/portal/proyectos/MNEME.

[5] Molero, X.; Rodas, A.; Pont, A., Sahuquillo, J. y Valiente, J.M. "Diseño de experiencias prácticas sobre memoria cache" Actas de las VII Jornadas de Enseñanza Universitaria de Informática, Jenui 2001, 3-8.

[6] Novak, J.D. Clarify with concept maps: A tool for students and teachers alike. The Science Teacher. 1991, 58(7), 45-49.

[7] SIJEM puede descargarse de : http://www.isaatc.ull.es/portal/proyectos/sijem.

[8] Vega Rodríguez, M.A.; Gómez Pulido J.A. y Sánchez Pérez, J.M. "Enseñanza de arquitecturas de memorias caché mediante simuladores". Actas de las VII Jornadas de Enseñanza Universitaria de Informática, Jenui 2001, 9-14.

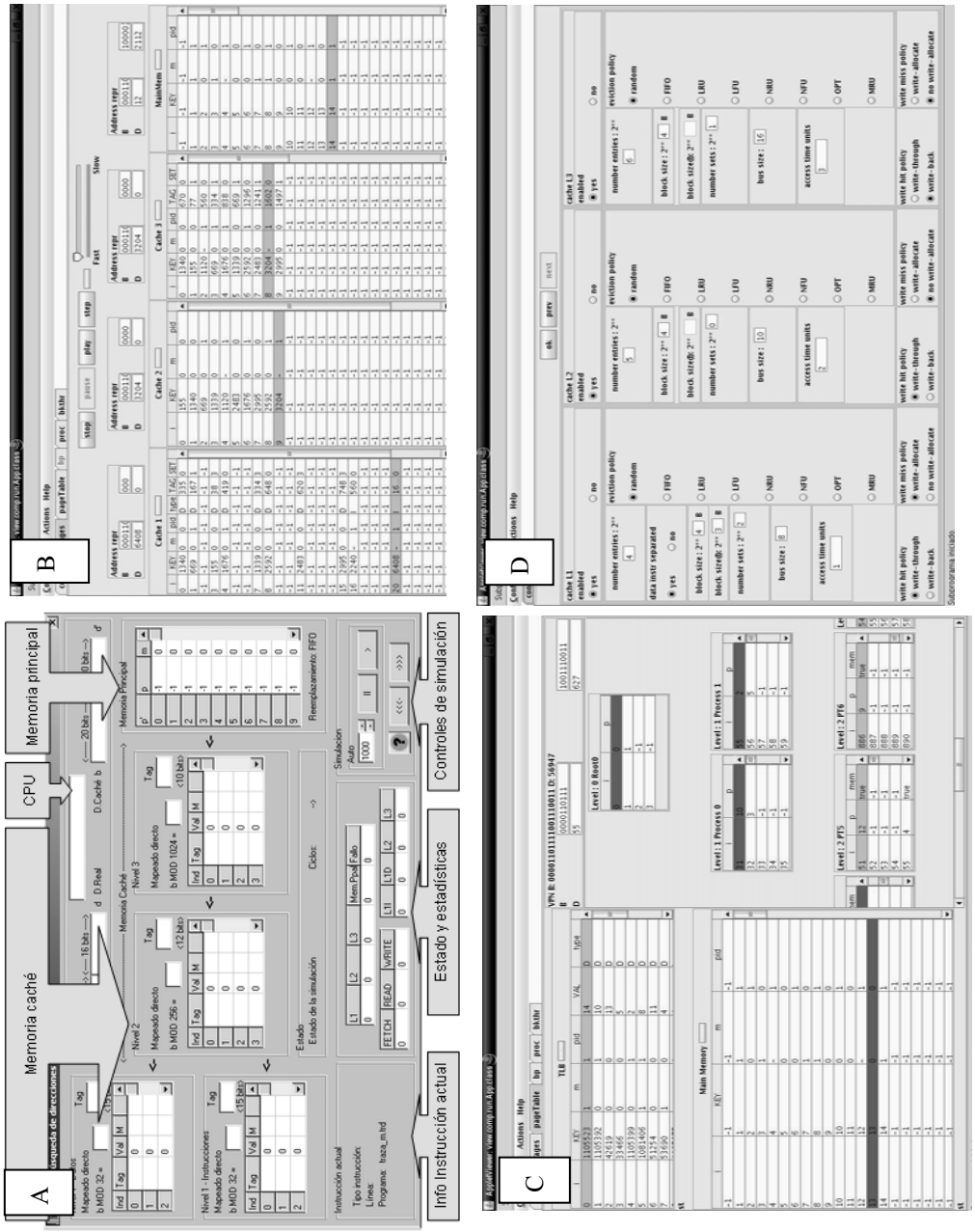


Figura 1. Diversas capturas de pantalla para los simuladores SIJEM (A) y MNEME: caches (B), tabla de páginas (C) y configuración de páginas (D)

PROIEKGES: Gestión y Planificación Estratégica de Proyectos Fin de Carrera

Jose Antonio Lozano Alonso[1] , Miren Bermejo Llopis[2], Julián Gutiérrez Serrano[2]
[1]Dpto. de Ciencias de la Computación e Inteligencia Artificial
[2]Dpto. de Lenguajes y Sistemas Informáticos
Facultad de Informática
Universidad del País Vasco-Euskal Herriko Unibertsitatea
{ja.lozano, miren.bermejo, julian.gutierrez}@ehu.es

Resumen

El objetivo de este artículo es presentar la herramienta Proiekges: aplicación para la gestión de Proyectos Fin de Carrera (PFC). Este sistema nace para resolver la complicada gestión de los PFC en las distintas titulaciones que ofrece la Facultad de Informática de San Sebastián (FISS). Los proyectos pasan por distintos estadios: oferta (pública, empresa y acordada), asignación de proyecto a los alumnos, composición y adjudicación de tribunales, defensa, informe de evaluación y publicación de notas, almacenamiento de memorias, informe final de PFCs, etc. Toda esta gestión consumía una gran cantidad de recursos, principalmente personales (horas de trabajo) pero también materiales (papel). La aplicación que aquí se presenta resuelve este problema de forma óptima. Al mismo tiempo, ésta aporta la información suficiente para mantener un diagnóstico constante de la situación de la asignatura en las diferentes titulaciones, ayudando a la planificación estratégica de la misma y de los últimos cursos. Este artículo describe cuáles han sido las decisiones de planificación que se han tomado gracias a Proiekges y hace balance del ahorro de recursos que supone disponer de una aplicación como ésta.

1. Motivación

Las titulaciones de Ingeniería en Informática e Ingeniera Técnica en Informática de Sistemas se implantaron en la Facultad de Informática de San Sebastián de la Universidad del País Vasco-Euskal Herriko Unibertsitatea en el año 1995 y 1997 respectivamente, y desde entonces han salido varias promociones de Ingenieros en Informática (II) e Ingenieros Técnicos en Informática de Sistemas (ITIS) que han tenido que defender un Proyecto Fin de Carrera antes de obtener el título [2]. En el caso de la II, este proyecto corresponde a la asignatura troncal del mismo nombre a impartir durante el primer y segundo cuatrimestre que lleva asociada una carga lectiva de 15 créditos. En la ITIS es una asignatura obligatoria de 6 créditos.

La implantación de los PFC en esta Facultad supuso un esfuerzo importante por varios motivos:

- Era una asignatura completamente nueva y sobre la cual no se tenía ninguna experiencia.
- A diferencia de otras asignaturas, que las imparte un solo departamento, en la docencia de ésta se tenían que implicar tres. Cada uno de ellos asumía la responsabilidad de dirigir parte de los PFC matriculados.
- El número de alumnos que se licenciaba anualmente era elevado: alrededor de 120. Esto obligaba que la cantidad de proyectos a ofertar fuese alta.
- Para gestionar cada proyecto eran necesarios una gran cantidad de recursos.

Todo esto llevó a crear una comisión que se encargaría de dar un marco organizativo para los PFC. Este equipo viene trabajando desde 1997 y recoge los trabajos realizados por una comisión anterior constituida en el departamento de LSI, que ya llevaba estudiando el tema desde 1995. Como resultado de su trabajo se desarrolló la Normativa de Proyectos que reglamenta el desarrollo y evaluación los mismos.

A pesar de todos los esfuerzos de la comisión y de disponer de un marco regulador, la organización de los PFC era lo suficientemente compleja como para que cada año surgieran nuevos problemas y retos. Pero la mayor carencia que se descubría era cómo hacer frente al coste de recursos que suponía toda la gestión de los PFC. Entonces se planteó la idea de desarrollar una aplicación que ayudara en dicho cometido:

Proiekges. Esta herramienta fue creada en 2003 y lleva en funcionamiento desde el curso 2004-2005.

En el apartado 2 se explican las peculiaridades de la asignatura PFC en la Facultad para a continuación, plantear los objetivos que se fijaron cuando se desarrolló la aplicación. En el punto 4 se detallan las funcionalidades principales de la herramienta, en el 5 se presentan las características del módulo de estadísticas y finalmente, se exponen las conclusiones obtenidas.

2. La asignatura PFC en la Facultad de Informática

Antes de introducir la aplicación en detalle describiremos las principales características de la gestión de los PFC dentro del centro.

Tal y como se ha explicado en el apartado anterior, en la actualidad se imparten dos titulaciones en la FISS y ambas tienen la asignatura de Proyectos de Fin de Carrera en sus planes de estudios. Indicar que de cara a la gestión, no se hacen diferencias entre los proyectos de una y otra titulación exceptuando el porcentaje de los mismos que cada departamento asume como parte de su docencia. Dicho reparto es el siguiente: para la Ingeniería Informática el 30% de los PFCs los dirigen profesores del departamento de Arquitectura y Tecnología de Computadores (ATC), otro 30% del de Ciencias de la Computación e Inteligencia Artificial (CCIA) y un 40 % del de Lenguajes y Sistemas Informáticos (LSI). En el caso de la Ingeniería Técnica un 40 % el departamento de ATC, un 30% el de CCIA y el 30% restante LSI. Por otro lado, el proyecto de fin de carrera se concibe como un trabajo que el alumno debe desarrollar de forma individual y en el que pone en práctica los conocimientos adquiridos durante los estudios.

Una de las características principales de la gestión de PFC es que es un sistema garantista, en el sentido de que trata de garantizar proyecto a todo alumno matriculado. De esta forma existe un proceso de asignación de proyectos que se realiza en dos etapas. En un primer plazo se realizan acuerdos de proyectos entre profesores y alumnos, y en un segundo plazo se realiza una oferta pública de proyectos que cubre a los alumnos sin proyecto asignado hasta dicho momento.

Otra característica importante es la evaluación del PFC. Este proceso ha ido evolucionando con la experiencia adquirida a lo largo de los años. Actualmente existen dos formas de evaluar: con defensa (presentación oral del PFC) y sin ella. En el primer caso el proyecto es presentado ante un tribunal formado por dos profesores que son los que asignan la calificación. En el segundo caso, en el que la evaluación se lleva a cabo sin realizar una defensa, el director del proyecto asigna la calificación y ésta no puede superar el 6. El proceso de evaluación comienza con el VºBº del proyecto por parte del director y con la emisión de un informe acerca del mismo (necesarios para que se admita a evaluación el proyecto). Asimismo el tribunal de un proyecto debe emitir un informe que justifique la calificación asignada. En ambos casos, el alumno debe presentar una memoria final que debe cumplir unos requisitos en cuanto a estructura y formato.

Otro aspecto a tener en cuenta es que los estudiantes pueden llevar a cabo el PFC en la Facultad (en cuyo caso utilizan un laboratorio reservado para ello), en otra universidad o en alguna empresa. En este último caso, además de los trámites habituales existen otros adicionales tales como firmar un convenio, que complican un poco más la gestión. Desde el comienzo, la posibilidad de desarrollar el PFC en la empresa ha sido una opción muy apreciada por el alumnado y cada vez es mayor el número de alumnos que optan por ella. De igual manera, las empresas han ido viendo las ventajas y el número de organizaciones interesadas en ofrecer proyectos ha ido aumentando significativamente (ver figura 1. etiqueta “externos”).

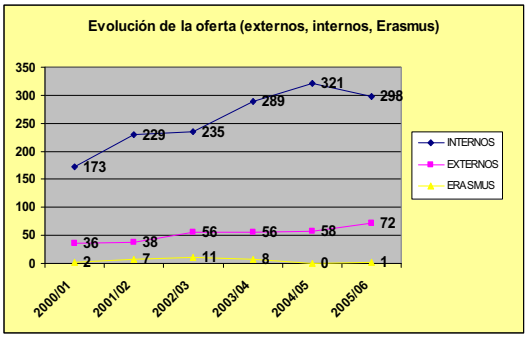


Figura 1. Evolución de PFCs desarrollados en la empresa

3. Objetivos de Proiekges.

Tras algunos cursos en los que se fue viendo la complejidad creciente de la gestión de los PFCs en la Facultad, se planteó la necesidad de automatizarla a través de una aplicación [3]. Está fue especificada y desarrollada en su mayor parte como el PFC de un alumno [4]. La implantación final de la aplicación fue llevada a cabo por una empresa de desarrollo de software.

Antes de construir la aplicación se plantearon los siguientes objetivos:

- Disminuir/eliminar los desplazamientos de los implicados a la secretaría del centro para la realización de gestiones relacionadas con el PFC mediante acceso Web.
- Simplificar el proceso de gestión, descentralizar la inserción de los datos en el sistema de gestión (antes exclusivamente secretarías) y distribuir determinados aspectos de la gestión a los usuarios apropiados.
- Guiar a los usuarios en las gestiones (eliminando o minimizando la posibilidad de error).
- Reducir la carga de trabajo del personal de secretaría.
- Reducir la carga de trabajo a la hora de cumplimentar un documento o realizar una gestión.
- Eliminar los picos de trabajo generados por dicha gestión en la secretaría de la Facultad durante los dos primeros meses del inicio de un nuevo curso.
- Optimizar la gestión de los PFC implicando directamente a los profesores, empresas y alumnos en la misma.
- Informatizar y automatizar los procesos de gestión.
- Estandarizar y mejorar los documentos y los listados necesarios para la correcta gestión de los PFCs.
- Eliminar el consumo excesivo de papel y la pérdida de espacio y tiempo para su clasificación.
- Informatizar y centralizar la información (lo que permite su análisis conjunto).
- Automatizar el proceso de asignación de proyectos de oferta pública.
- Mejorar el conocimiento que se tenía de los proyectos, su evolución y estado.
- Mejorar el análisis de los datos (estadísticas) y con ello la toma de decisiones.
- Permitir que el profesorado tuviera una visión global de la asignatura y no sólo de los proyectos que dirigía o de los que se dirigían en su departamento.
- Implicar la participación de las empresas en los PFC facilitando su relación con la Facultad a través de la aplicación [1]

4. Descripción y uso de PROIEKGES

La aplicación Web Proiekges funciona sobre un servidor instalado por la Facultad para tal fin. El interfaz de usuario es bilingüe, posibilitando el uso de cualquiera de los dos idiomas oficiales de la Comunidad Autónoma Vasca. Asimismo, toda la comunicación entre los usuarios del sistema se realiza vía electrónica. A continuación describimos con cierto detalle las funcionalidades de la aplicación asociadas a cada uno de los usuarios.

4.1. Profesores

La funcionalidad correspondiente al usuario *profesor* está dividida en tres bloques: operaciones genéricas (actualización de datos de la cuenta, acceso al tablón de anuncios etc.), acciones propias de un director de proyecto y también las propias de un miembro de tribunal (ver figura 2). A continuación se explican las correspondientes a la labor de director ya que las de miembro de tribunal tienen más adelante un apartado específico para ello.

Al comienzo del proceso lo habitual es que el profesorado tenga acordados algunos proyectos con los alumnos. Una vez que el alumno se ha dado de alta en la aplicación (cosa que pueden hacer cuando esté matriculado en la asignatura), el profesor da de alta el PFC desde su pantalla. Esto implica introducir datos tales como el título, el tipo de proyecto, asignar al estudiante o introducir los datos de la empresa en el caso de que se realice mediante esta opción. Una vez hecho esto, el alumno puede consultar los datos de su proyecto accediendo desde su cuenta.

Sin embargo, el acuerdo no es la única vía que tienen los alumnos de elegir proyecto tal y como se ha indicado anteriormente. Otra de las

funcionalidades implementadas para este tipo de usuario es la posibilidad de incluir sus proyectos en la oferta pública que hace la Facultad.

Una vez terminado el trabajo, el director da el VºBº a la defensa desde esa misma pantalla y completa un informe al que accede el tribunal asignado a ese proyecto. En caso de que la evaluación se realice sin tribunal, se activa esta posibilidad y el director adjudica la nota directamente. El sistema controla que no se ponga una calificación mayor que la máxima permitida (un seis).

Además de las funcionalidades comentadas, el profesorado puede consultar datos históricos de los proyectos dirigidos y calificados.

Figura 2. Pantalla del usuario profesor

4.2. Tribunales

Tal y como se puede ver en la figura 2, el acceso a las funcionalidades correspondientes a los usuarios como miembros de los tribunales de evaluación se realiza desde la pantalla de profesor. Con estas opciones, entre otras cosas, puede consultar qué proyecto tiene que evaluar y cuál es su cargo en el mismo (presidente o secretario). También es posible obtener un listado de todas las evaluaciones realizadas en los diferentes cursos. Además, la documentación necesaria para poner la nota (actas de tribunal, informe etc.) se encuentra accesible desde la aplicación de forma que no es necesario imprimir ningún papel.

4.3. Alumnos

El usuario *alumno* está dividido en dos bloques: las operaciones genéricas similares a las del profesor y las específicas de los alumnos (ver figura 3).

Al inicio del curso los alumnos utilizan la aplicación para acceder a la oferta de proyectos públicos pudiendo realizar una selección de los mismos.

Una vez asignado un proyecto (ya sea de oferta pública o acordado) el estudiante puede consultar los datos del mismo. Los documentos estándar necesarios para el desarrollo correcto de la memoria también se encuentran accesibles desde aquí, entre otros, la portada ya adaptada con el título de su proyecto o el documento con el VºBº del director. Además, puede consultar el tribunal que le ha sido asignado, o en caso de quererlo, solicitar ser evaluado sin tribunal. En este último caso, el director del proyecto recibe un aviso de que el alumno ha solicitado ese tipo de evaluación y queda desactivada la parte correspondiente al tribunal.

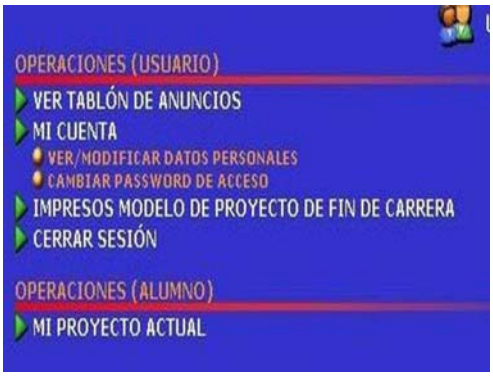


Figura 3. Pantalla del usuario alumno

4.4. Administración

El administrador de la aplicación es una persona de la Secretaría del Centro que se encarga de realizar las operaciones de actualización y puesta al día de los datos además de obtener los diferentes informes al finalizar el curso.

Las funcionalidades de este perfil están dividas en cuatro bloques: una correspondiente a operaciones genéricas como las del profesor o el alumno, otra con las opciones para gestionar los

usuarios, una tercera para configurar la aplicación y una última para gestionar los proyectos (ver figura 4). A continuación se explican con más detalle los apartados específicos del administrador.

La gestión de usuarios permite validar a los mismos (alumnos, profesores y tutores). Esto implica ir dando por buenos los datos una vez que los usuarios se han dado de alta. Aunque los profesores prácticamente no varían de un curso a otro, sí lo hacen los alumnos y los tutores de las empresas.

En la gestión de proyectos se encuentran la mayoría de las opciones de este usuario. Entre ellas, se pueden insertar nuevos proyectos (aunque normalmente se trate de una labor que realiza el director) o gestionar los mismos. También es posible introducir los tribunales asignados a cada proyecto, y llevar el control de los convenios firmados con empresas. Este último punto es de gran importancia ya que todos los alumnos que realizan proyectos en una empresa deben suscribir un seguro de responsabilidad civil obligatorio y además sus datos hay que enviarlos al Instituto Nacional de la Seguridad Social.

Finalmente, las funcionalidades asociadas a la configuración de la aplicación permiten gestionar los anuncios o modificar opciones de administración.

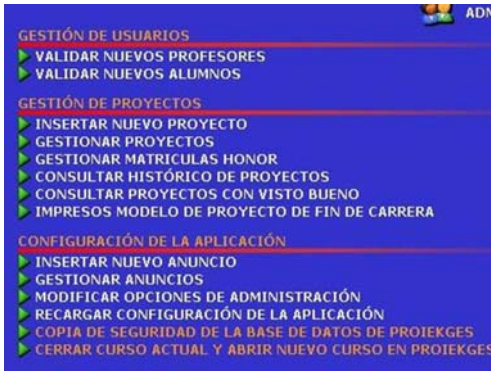


Figura 4. Pantalla del usuario administrador

4.5. Empresas

Las empresas a las que les interesa ofertar PFC se dan de alta en la aplicación y pueden realizar su oferta a través de la misma.

Los alumnos ven el listado de proyectos ofrecidos y se apuntan a los que les interesan. La empresa puede entonces obtener el currículo de los alumnos interesados en sus proyectos y realizar la selección.

Otro de los aspectos relacionados con los PFC desarrollados en la empresa es el convenio de colaboración que se rellena automáticamente con los datos de la misma y del estudiante. No queda más que completar los datos e imprimir el documento para su firma.

4.6. Otras tareas

Además de las tareas señaladas con anterioridad, la aplicación dispone de otras funcionalidades más generales tales como la asignación automática de PFC en la oferta pública o un módulo para obtener estadísticas y obtención de los créditos asignados a cada profesor por dirección de proyectos.
Esta última posibilidad es la que proporciona los datos, estudios y comparativas que permiten tomar decisiones de cara al siguiente curso. Debido a la importancia de este módulo, se dedica el apartado siguiente a dar una descripción más exhaustiva del mismo.

5. Proiekges: un observatorio de los proyectos fin de carrera

Como se ha señalado anteriormente, la aplicación permite realizar toda la gestión de los FPC vía web, facilitando el acceso a la información por parte de los usuarios y simplificando de forma importante las tareas de secretaría de centro. Pero, además de estos beneficios, la aplicación proporciona una cantidad de información y estadísticas que permite realizar una planificación estratégica y una toma de decisiones que traten de corregir los desequilibrios que pueden surgir en relación con la asignatura.

El módulo de estadísticas del Proiekges está realizado mediante un cubo OLAP, de esta forma se puede acceder a estadísticas tan variadas como las siguientes:

- PFC: Número de proyectos evaluados por curso académico. Duración de los proyectos. Idioma de cada proyecto.
- Evaluación: Tipo de evaluación (con, sin defensa)

- Notas de los alumnos.
- Alumnos: Número de alumnos matriculados, número de ellos sin proyecto asignado, número de PFC presentados y los que se quedan sin presentar.
- Profesores: Es posible conocer las notas medias de sus alumnos, las notas asignadas como tribunal, número de años de duración del proyecto de los alumnos dirigidos.

Los tres primeros factores, número de proyectos, tipo de evaluación, y notas de los alumnos pueden combinarse con cualquiera de los siguientes: convocatoria, titulación, departamento y tipo de proyecto (acordado, ofertado, en empresa) para obtener estadísticas más específicas. Por supuesto, cualquier estadística es posible obtenerla con un marco temporal de varios años.

En la figura 5 se puede ver un ejemplo de gráfica de datos obtenida a partir de módulo de estadísticas de la aplicación, En ella se ven las notas recibidas por los alumnos en los seis últimos cursos académicos.

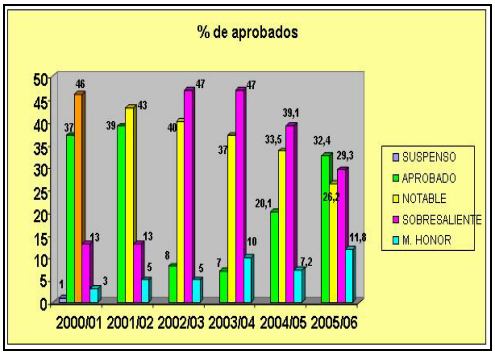


Figura 5. Ejemplo de gráfica obtenida a partir del módulo de estadísticas del ProiekGes

El observatorio proporcionado por Proiekges ha permitido tomar algunas decisiones para corregir algunos de los desequilibrios que se han ido detectando y prevenir los futuros. A continuación se describen con ejemplos algunas de las acciones llevadas a cabo.

5.1. Evaluación de proyectos

Una de las decisiones adoptadas ha sido la de permitir la evaluación del proyecto sin llevar a cabo una defensa del mismo. Esta decisión se tomó a la vista de los siguientes aspectos:

- El gran incremento en el fracaso en la asignatura mostrado en la figura 6 (PFC defendidos frente a PFC matriculados).

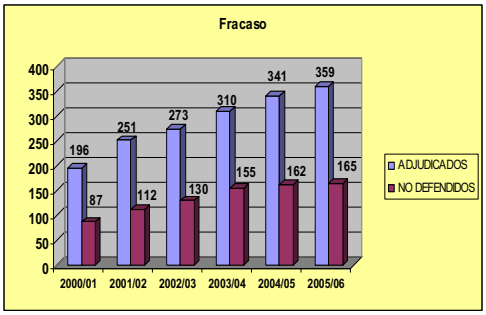


Figura 6. Evolución de proyectos defendidos frente a los adjudicados

- Las notas medias de los proyectos (estaba por encima de 8.0). Esta alta nota implicaba una gran exigencia en los proyectos. Es decir, un director de proyecto únicamente daba el VºBº al alumno cuando la calidad del proyecto era muy alta.
- La existencia de una bolsa de alumnos, que teniendo todos los créditos aprobados, y habiéndose incorporado al mercado laboral, se les hacía muy complicado realizar un proyecto con grandes requerimientos. Este grupo de alumnos permanecían muchos años matriculados en la asignatura pero sin presentar el proyecto.

Para solucionar este problema se planteó la posibilidad de evaluar el proyecto sin defensa. Esto ha permitido invertir la línea ascendente de fracaso en la asignatura. La relajación de los requerimientos para aprobar un proyecto, de tal forma que ahora no es necesario tener un trabajo de al menos notable para ser evaluado, ha animado a muchos alumnos a terminar su PFC. Este mecanismo, ha posibilitado finalmente la reducción de la bolsa de profesionales que se encontraban pendientes del PFC. Esta medida se puso en marcha en el curso 2005/2006, y es en este curso donde se invierte la tendencia

ascendente de los proyectos sin defender, bajando del 48% del curso anterior al 44% en dicho curso. Se espera que esta nueva tendencia descendiente continúe durante los próximos cursos.

5.2. Un sistema garantista

Otra particularidad del sistema de gestión es garantizar un PFC a cada alumno. Aunque es una característica que consideramos importante, el volumen de alumnos matriculados y sobre todo los datos esperados en los próximos cursos, han llevado a tomar medidas para no colapsar el proceso (en el curso pasado el número total de alumnos matriculados fue de 395, con el esfuerzo que esto supuso). Como consecuencia de una análisis exhaustivo de los datos proporcionados por Proiekges se observó que en el 85% de los PFC defendidos el alumno tenía como máximo 15 créditos matriculados a parte de los que supone el propio proyecto. Esto, llevó a tomar la decisión de ofertar únicamente proyecto a los alumnos que se encontrasen en dichas condiciones de matrícula. Así se evita que muchos alumnos se arriesguen a matricularse en el PFC, haciendo que la oferta sea mucho más ajustada a la realidad. Es lógico que si el alumno ve facilidades en la oferta, a pesar de que su carga crediticia en la matrícula sea grande, apueste por matricularse en el PFC. Precisamente es este factor de alta carga de trabajo el que llevaba a un fracaso anunciado.

6. Conclusiones

A Lo largo de este artículo se ha presentado la aplicación Proiekges para la gestión de PFC. La utilización de este sistema en la Facultad de Informática de San Sebastián ha supuesto notables mejoras en varios aspectos. En primer lugar ha reducido el tiempo y los recursos humanos utilizados en la gestión: aproximadamente un 70%.

En segundo lugar, se ha conseguido un importante ahorro de papel: aproximadamente un 85%. Por último hay que destacar que gracias a la gestión de datos y a la obtención de estadísticas que permite la aplicación se han podido realizar estudios que han llevado a la toma de decisiones de carácter fundamental en la mejora de la gestión y en la planificación estratégica de la asignatura de PFC y en la de los últimos cursos de la carrera.

Este sistema lleva en funcionamiento desde el curso 2004-2005 y la acogida que ha tenido entre los usuarios ha sido muy alta. Cada curso se gestionan a través de ella del orden de 400 proyectos, y acoge a 120 profesores, unos 350 alumnos y unas 70 empresas.
Cada año y como consecuencia de las sugerencias de los usuarios, y sobretodo de los cambios que se realizan en la normativa de gestión de proyectos, se realizan mejoras y actualizaciones del sistema, si bien, la aplicación presenta una gran robustez y utilidad.
En otro orden de cosas, decir que la aplicación ha sido considerada por otras escuelas técnicas de la Universidad del País Vasco para ser adaptada y utilizada.
Este sistema recibió en 2004 el Accesit en los premios ETI Euskadi para instituciones.

Referencias

[1] Cesar I. García, Carlos López, Belén Vaquerizo, Calos Prado. Experiencias de colaboración con empresas en la realización de Proyectos fin de carrera de la Ingeniería Informática de Gestión de la Universidad de Burgos. JENUI 2001.
[2] C.W. Dawson. El Proyecto Fin de Carrera en Ingeniería Informática. Una guía para el estudiante. Prentice Hall. 2002.
[3] http://proiekges.si.ehu.es/
[4] Proyecto Fin de Carrera del Proikeges

AIBench: framework de programación para la enseñanza de técnicas de IA en tercer ciclo

R. Domínguez, D. Glez-Peña, J. R. Méndez, F. Fdez-Riverola
ESEI: Escuela Superior de Ingeniería Informática
Universidad de Vigo
Campus Universitario As Lagoas s/n, 32004, Ourense
rubendc@sing.ei.uvigo.es, {dgpena, moncho.mendez, riverola}@uvigo.es
F. Díaz
Escuela Universitaria de Informática
Universidad de Valladolid
Plaza Santa Eulalia, 9-11, 40005, Segovia
fdiaz@infor.uva.es

Resumen

En este trabajo se presenta AIBench, un framework que agiliza la codificación, ejecución, prueba y optimización de técnicas de Inteligencia Artificial (IA). AIBench está pensado para su utilización por parte de alumnos de tercer ciclo, involucrados en programas de doctorado donde se imparten materias que implican la comprensión e implementación de distintas técnicas de IA. La finalidad del framework desarrollado es liberar al alumno de las labores de programación repetitivas y propensas a errores que están relacionadas, entre otras, con la entrada de datos, diálogos de configuración de parámetros o presentación de resultados, permitiéndole centrarse únicamente en los algoritmos objeto de estudio.

1. Introducción y Motivación

El objetivo principal de un programa de doctorado es la formación del alumno en distintas áreas de investigación y su capacitación final para llevar a cabo el desarrollo posterior de una tesis doctoral. En este contexto, la finalidad de las distintas materias (primer curso) y trabajos (segundo curso) que componen el itinerario de tercer ciclo, es la de proporcionar una rigurosa base científica y una visión global sobre un amplio espectro de técnicas y algoritmos de investigación aplicados a distintos campos.

En este sentido, el programa de doctorado TADSI (*Tecnologías Avanzadas para el Desarrollo de Software Inteligente*) impartido en el Departamento de Informática de la Universidad de Vigo, tiene como finalidad la especialización del alumno en técnicas de investigación relativas al desarrollo de software usando técnicas basadas en IA e ingeniería de software avanzada [1].

Formando parte del mencionado programa de doctorado, las líneas prioritarias que ofertan los miembros del grupo de investigación SING (*Sistemas Informáticos de Nueva Generación*) [2] engloban a diversos campos de la IA aplicando distintas técnicas para la resolución de problemas de ámbito diferente. En concreto, se puede mencionar la utilización de redes neuronales, lógica difusa, agentes inteligentes o sistemas de razonamiento basado en casos, aplicados a problemas tan diversos como la clasificación de correo spam, el análisis del genoma humano o el desarrollo de sistemas de detección de intrusos (IDS, *Intrusion Detection System*), entre otros.

En el caso concreto de las distintas materias existentes en el primer curso del bienio de doctorado, es habitual la asignación al alumno de tareas de implementación de técnicas concretas de IA que complementen el contenido más teórico de los cursos. Por lo tanto, la situación actual existente en el curso de doctorado y en el seno del grupo SING, es que como resultado de cada trabajo de investigación realizado por parte de los alumnos y/o investigadores, se han venido desarrollando un conjunto de aplicaciones software que han sido implementadas de forma independiente e inconexa.

Siguiendo este método de trabajo, además del esfuerzo requerido en cada caso para el estudio de las distintas técnicas, es necesario dedicar una cantidad considerable de tiempo extra para el

desarrollo "desde cero" de la herramienta software final de soporte y prueba. Considerando a mayores el hecho de que los alumnos del curso de doctorado posteriormente se integran en los distintos grupos de investigación existentes en el departamento, los beneficios esperados de la utilización de un entorno común de programación de carácter integrador para tercer ciclo y grupos de investigación, se vislumbran a todas luces enriquecedores.

Teniendo en cuenta lo comentado anteriormente, se ha realizado un análisis de las distintas herramientas software desarrolladas por el grupo SING, así como otras alternativas disponibles en Internet (p.ej. Weka [6]), con el objetivo de identificar un patrón común acerca de las funcionalidades que las distintas técnicas de IA demandan de forma repetitiva. Resultado de este proceso de análisis, se ha observado que en cualquiera de las herramientas existentes el flujo de actividades que se ejecutan reiteradamente es el siguiente: carga del conjunto de datos de prueba, preprocesamiento, configuración de parámetros, ejecución de técnicas y visualización de resultados.

En este contexto y como solución a la problemática existente nace AIBench [3] con un doble objetivo: (*i*) eliminar el trabajo repetitivo que supone el desarrollo de toda aplicación de soporte a la investigación y (*ii*) ofrecer al alumno y/o investigador una serie de facilidades o servicios de los que puede hacer uso en la codificación de su técnica, facilitando con ello la reutilización de componentes software.

En este sentido, AIBench se puede definir como un framework de programación abierto que proporciona los servicios comunes anteriormente citados, liberando al investigador de las tareas comunes de programación relacionadas con la interfaz de usuario, entrada y salida de datos, etc. El programador puede así centrarse únicamente en la codificación de la técnica o algoritmo objeto de estudio. Para las restantes tareas se hará uso de los servicios disponibles en el framework proporcionado. Entre dichos servicios se encuentran ya disponibles distintas utilidades para la importación y exportación de datos en diferentes formatos, paneles genéricos para llevar a cabo la visualización de resultados, formularios parametrizables para captura de datos de usuario, etc.

Una vez adquirido el domino necesario de la plataforma, las principales ventajas para los alumnos de tercer ciclo y los docentes del programa de doctorado se pueden resumir en los siguientes grandes apartados:

- El alumno solamente debe centrarse en los aspectos relativos a la programación de las técnicas objeto de estudio.
- Se genera la posibilidad natural de integración de distintas técnicas entre sí, lo que permite el tratamiento de problemas más complejos o el análisis del mismo problema desde una perspectiva más amplia.
- De forma paralela se lleva a cabo la creación de un repositorio de algoritmos especializados y documentados, que favorece la reutilización de componentes software específicos.
- Permite la existencia de un único punto de integración, ampliación y utilización para la interconexión con otras plataformas de desarrollo estándar.

Mientras que la primera sección ha presentado una breve introducción al trabajo realizado motivando su originalidad e interés, el resto del artículo se estructura como sigue. En primer lugar se presentan las abstracciones básicas empleadas en el diseño de AIBench, necesarias para poder extender y hacer uso de las funcionalidades que ofrece. Posteriormente se describe brevemente la forma en la que el programador de AIBench puede implementar nuevas operaciones o técnicas haciendo uso de los servicios básicos actualmente disponibles. A continuación se detalla la arquitectura interna del framework, haciendo especial hincapié en su comportamiento incremental basado en plugins. Finalmente, se exponen las conclusiones así como el trabajo presente y futuro, citando alguno de los proyectos asociados en los que se está trabajando actualmente.

2. Conceptos Básicos en AIBench

AIBench implementa un framework de programación ligero, no intrusivo y basado en MVC (Modelo Vista/Controlador) que posibilita el desarrollo de aplicaciones JAVA permitiendo:

- La conexión, ejecución e integración de operaciones con una entrada/salida bien definida.
- Diseño independiente del problema, es decir, AIBench no contiene clases relativas a conceptos de Inteligencia Artificial, como

pueden ser: *classifier, clusterer, feature selection, model, genetic search, cross validation,* etc.

- Diseño *default-driven*, es decir, AIBench toma una decisión por defecto para los posibles valores de configuración, buscando que el programador obtenga con el mínimo código posible una versión inicial, pero funcional, de su trabajo.

El framework desarrollado constituye una aplicación modular y fácilmente extensible compuesta por diferentes elementos, denominados plugins. Tanto los plugins que componen el núcleo de AIBench, como aquellos que pueden ser desarrollados por los programadores, se ejecutan bajo el motor de plugins Platonos [4]. A través de esta filosofía de desarrollo se pueden identificar dos roles de usuario relacionados con AIBench: (*i*) los desarrolladores del núcleo o *core* y (*ii*) los programadores de nuevas extensiones (plugins) del framework (habitualmente alumnos de tercer ciclo e investigadores).

En el ámbito de AIBench, bajo el concepto de plugin se engloban los siguientes componentes software:

- **Operaciones** que representan e implementan técnicas o algoritmos concretos.
- **Tipos de datos** que representan instancias del dominio del problema.
- **Vistas** asociadas a los distintos tipos de datos con los que trabaja el framework.

La Figura 1 muestra el aspecto visual de la herramienta donde se guarda un registro de las *operaciones* llevadas a cabo (*Session Tree*), un registro de las instancias de los tipos de datos generados (*Clipboard Tree,* ver punto 2.4), el diálogo generado automáticamente para la entrada de datos de una operación específica (*Generated Dialog*) y una *vista* personalizada para un *tipo de datos* concreto (*Custom View*) (conjunto de microarrays de ADN).

A continuación se exponen en detalle los distintos tipos de plugins existentes en AIBench.

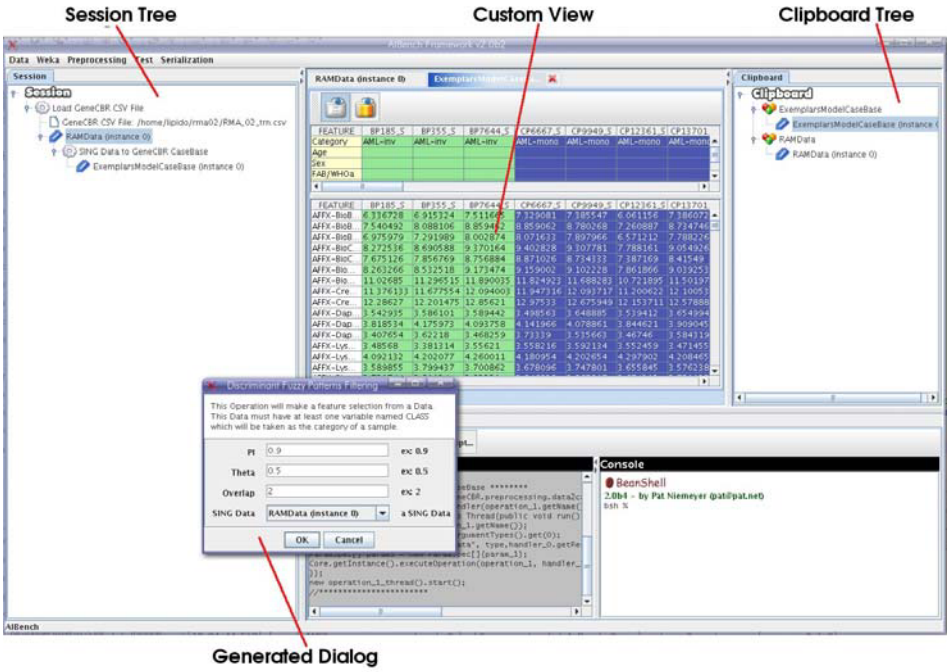


Figura 1. Interfaz de usuario de AIBench durante la ejecución de un experimento

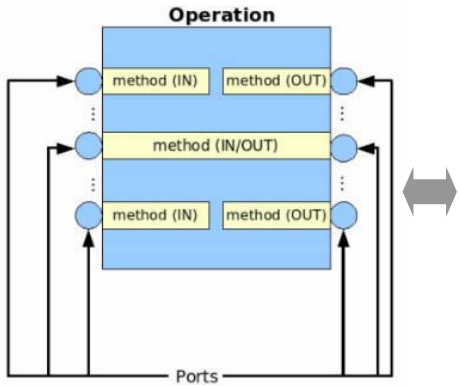
2.1. Operaciones

La extensión natural del framework de AIBench se realiza a través de la codificación de nuevos bloques de código fuente, denominados operaciones. Una operación corresponde con la implementación concreta de una técnica o algoritmo particular.

El modelo subyacente de programación de operaciones en AIBench presupone que:

- La lógica puede ser desacoplada totalmente de la interfaz de usuario.
- La interconexión de operaciones puede ser modelada utilizando el concepto de *experimentos*, que pueden ser ejecutados de forma repetitiva sobre distintos datos.
- El programador está obligado a *pensar antes de programar*, a través de la definición de distintas operaciones.

Para el programador de AIBench una operación es simplemente una clase Java con anotaciones en su código, que definen claramente sus entradas y salidas a través de los correspondientes *puertos* (puertos de entrada, de salida o de entrada/salida). El ejemplo mostrado en la Figura 2 define una sencilla operación que implementa la suma de dos números enteros.




```
@Operation(description="this operation adds two numbers")
public class Sum{
 private int x,y;

@Port(direction=Direction.INPUT, name="x param")
public void setX(int x){
 this.x = x;
}

@Port(direction=Direction.INPUT, name="y param")
public void setY(int y){
 this.y = y;
}

@Port(direction=Direction.OUTPUT)
public int sum(){
 return this.x + this.y;
}
}
```

Figura 2. Modelo abstracto de operaciones de AIBench basado en puertos de entrada, salida y entrada/salida

A través de la utilización de anotaciones, junto con un descriptor de la operación o plugin en formato XML, AIBench dispone de toda la información necesaria para:

- Colocar las operaciones en sus correspondientes menús.
- Generar los cuadros de diálogo necesarios para la introducción de datos.
- Almacenar los resultados obtenidos para presentarlos como posible entrada en posteriores ejecuciones de otras operaciones compatibles con el tipo de dato generado.

2.2. Tipos de datos

Con el objetivo de facilitar la extensión del framework, AIBench ofrece al programador la posibilidad de definir sus propios tipos de datos, que serán utilizados como entrada o salida en las operaciones disponibles. Dichos tipos de datos no heredan ni implementan ninguna clase o interfaz para evitar cualquier tipo de acoplamiento, únicamente serán representaciones en memoria de conceptos relacionados con el dominio del problema. En AIBench se diferencian dos tipos de datos en función de su complejidad y naturaleza:

- Tipos de datos simples, generalmente introducidos por el usuario.
- Estructuras de datos complejas, generalmente obtenidas como resultado de la ejecución de una operación.

2.3. Vistas

En el campo de la IA es muy común el uso de representaciones gráficas de los datos, ayudando al investigador en la evaluación de una determinada técnica. Estas representaciones utilizan distintos formatos que van desde una simple vista tabular, pasando por representaciones de funciones o gráficos de barras, hasta complejas representaciones para el análisis de secuencias del genoma humano. La Figura 3 muestra un ejemplo de este último tipo de vistas.

Figura 3. Vista de datos compleja en AIBench

En la distribución actual de AIBench (AIBench SDK 1.9.1) ya se incluyen vistas por defecto para los tipos de datos que maneja internamente el framework. De este modo, el programador siempre puede hacer uso de estas vistas para visualizar rápidamente los resultados obtenidos con sus técnicas. No obstante, se pueden implementar e incorporar nuevas vistas personalizadas y adecuadas a los tipos de datos que se manejen en cada momento.

2.4. Integración entre operaciones: *clipboard*

AIBench conoce en todo momento el conjunto de datos con los que trabaja el programador, generados desde operaciones. Esto se consigue con la implementación de un espacio de intercambio de datos al que se denomina portapapeles o *clipboard*.

La finalidad última del clipboard es posibilitar la integración entre las distintas operaciones a nivel de intercambio de datos. El clipboard de AIBench actúa del mismo modo que los portapapeles disponibles en la mayoría de los sistemas operativos existentes. Todos los resultados generados tras la ejecución de una operación se incorporan al clipboard, donde se clasifican y catalogan en función de la clase Java a la que pertenecen (tipo de datos que implementan). Con esta estructura de intercambio, los datos generados por unas operaciones estarán disponibles como entrada para posteriores ejecuciones de otras operaciones. La Figura 4 presenta un ejemplo de esta funcionalidad donde "Load CSV" y "Classifier Train" son dos operaciones conectadas a través del clipboard.

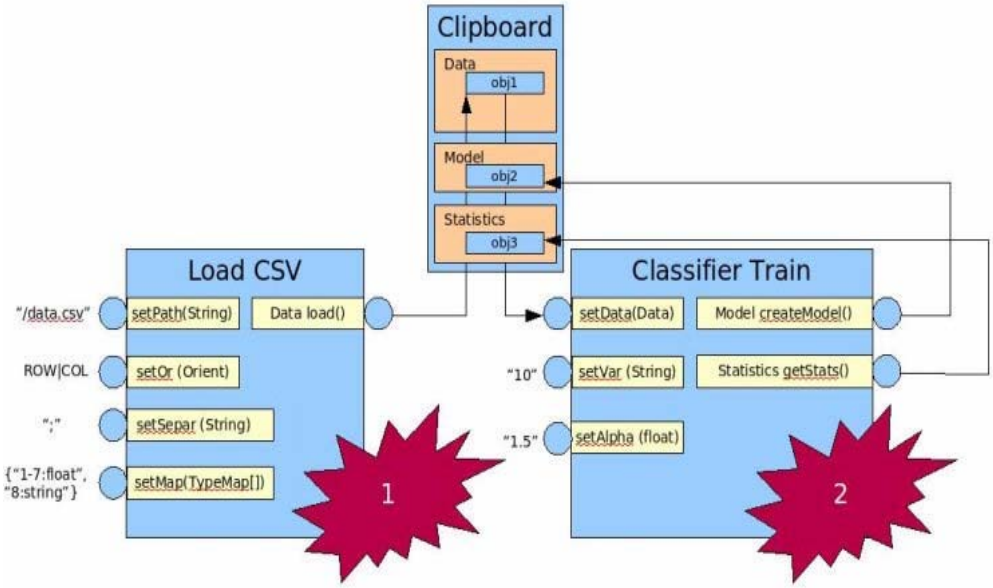


Figura 4. Integración de operaciones en AIBench utilizando el clipboard

3. Servicios de AIBench

3.1. Generación dinámica de la interfaz de usuario

El diseño e implementación de la interfaz de usuario de una aplicación, suele acarrear un esfuerzo considerable en relación a la totalidad del proyecto a desarrollar. Para reducir dicho esfuerzo, AIBench proporciona al programador un espacio de trabajo similar al presentado en la Figura 1, al que dinámicamente se incorporan las operaciones codificadas en sus correspondientes menús. Para conseguir esta funcionalidad sin necesidad de recompilar de nuevo la aplicación, AIBench hace uso de descriptores de operaciones utilizando ficheros con formato XML, así como anotaciones en las clases Java relacionadas con el mecanismo de introspección del que dispone el lenguaje JAVA.

Del mismo modo, los cuadros de diálogo que permiten al usuario indicar cuáles son los valores de los parámetros requeridos por las operaciones, también pueden ser generados dinámicamente por AIBench a partir de las anotaciones definidas en la implementación de la correspondiente operación. Como se comentaba anteriormente, los datos solicitados por las operaciones pueden ser de tipo simple o complejo. En el caso de tipos no simples, el usuario podrá seleccionar la instancia deseada de entre aquellas del mismo tipo existentes en el clipboard de AIBench.

La Figura 5 muestra un ejemplo de un cuadro de diálogo de entrada de datos, generado de forma automática por AIBench para la suma de números enteros.

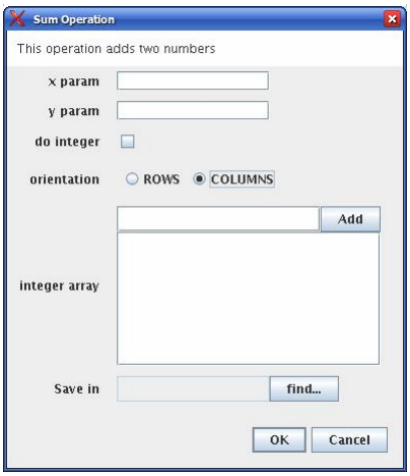


Figura 5. Cuadro de diálogo generado automáticamente por AIBench

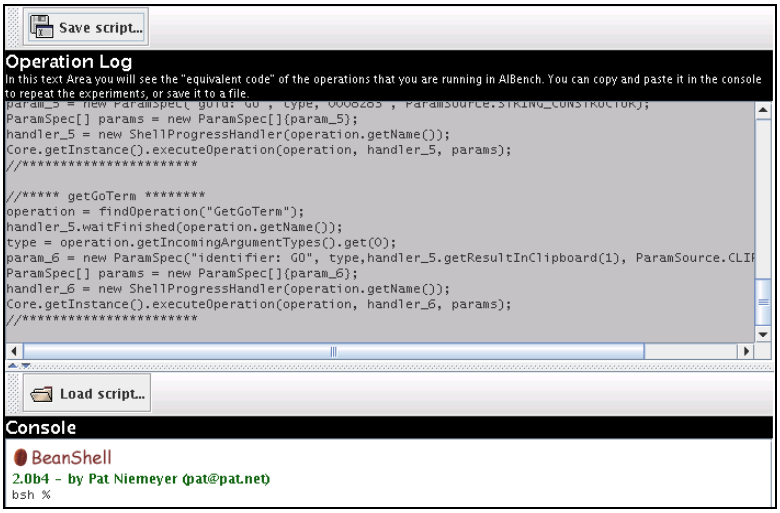


Figura 6. Shell scripting generado automáticamente para cada acción que el programador lleva a cabo en AIBench

3.2. Automatización de experimentos: *shell*

Una tarea común en la programación, prueba y validación de cualquier algoritmo, suele ser la ejecución reiterada de dicha técnica variando únicamente en cada ocasión algunos de los parámetros que determinan su comportamiento. Del mismo modo que tiene lugar la ejecución de procesos por lotes o scripts en un sistema

operativo, donde la interacción con el usuario es prácticamente nula una vez lanzada la tarea, sería de gran utilidad el poder disponer de algún mecanismo que diese soporte a la automatización de los experimentos en el marco de AIBench.

En este sentido, se ha integrado en AIBench un shell de libre distribución que permite dicha automatización [5]. Para cada una de las interacciones que tienen lugar entre el programador y AIBench, el framework genera dinámicamente el código fuente que define dicha secuencia de acciones (véase Figura 6). El *shell script* generado refleja las operaciones ejecutadas y las entradas de datos que el usuario ha realizado. Este código fuente se puede ejecutar posteriormente de nuevo a través de una consola incorporada en AIBench.

Con este servicio, el programador de AIBench puede guardar el código fuente generado a partir de un experimento inicial, cambiar ciertos parámetros en dicho código, y volver a ejecutarlo como si de un proceso por lotes se tratase.

4. Arquitectura Interna del Framework

Después de haber presentado los conceptos básicos que rodean AIBench, este apartado ofrece una visión integrada de su arquitectura modular reflejada en la Figura 7.

En la zona superior de la Figura 7 se encuentran los componentes que forman parte del núcleo (*core*) de AIBench: el clipboard, el historial de operaciones y el propio core. Además, existe un registro de las vistas implementadas que mantiene la asociación con los tipos de datos que pueden visualizar, y un *workbench* que se encarga de contener los elementos visuales que forman parte de la aplicación final desarrollada: menús, barras de herramientas, etc.

En la zona inferior de la Figura 7 se presenta, a modo de ejemplo, cómo se conectarían con el núcleo y la interfaz de usuario de AIBench un conjunto de operaciones concretas implementadas. En la Figura 7, los círculos intermedios representan las conexiones existentes entre los distintos componentes de la arquitectura. La conexión *plugins-core* es necesaria para que sea posible el registro y ejecución de las distintas operaciones implementadas. La conexión *plugins-workbench* permite que en los menús de la interfaz de usuario estén disponibles todas las operaciones implementadas.

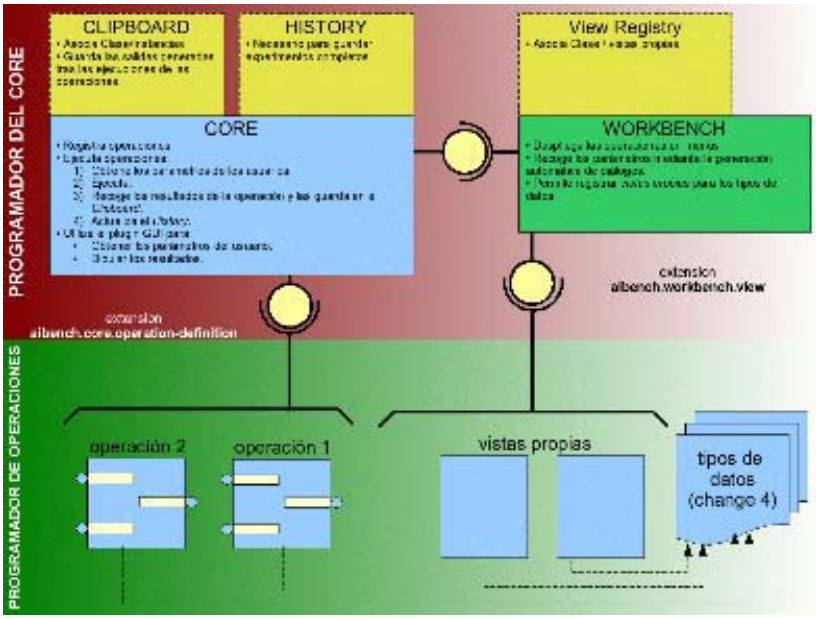


Figura 7. Arquitectura modular y extensible de AIBench

4.1. Comportamiento plug & play

Con anterioridad se ha comentado el hecho de que la arquitectura basada en plugins de AIBench permite la carga dinámica de nuevas operaciones implementadas. Esta funcionalidad se consigue empaquetando todos los componentes requeridos por cada plugin (ficheros binarios, iconos, etc.) en un fichero .JAR, junto con un descriptor XML que define unívocamente las características del plugin. Cada vez que se inicia AIBench, el framework lleva a cabo una búsqueda de todos los plugins disponibles, que se encuentran en un directorio concreto del sistema de ficheros. Para incorporar un nuevo plugin al sistema, únicamente se deberá codificar, crear su descriptor XML, empaquetarlo en un fichero .JAR y ponerlo accesible en el directorio de plugins de AIBench.

5. Conclusiones y Trabajo Futuro

En este artículo se propone AIBench como framework de programación para la enseñanza e interconexión de técnicas de IA en tercer ciclo, así como su posterior utilización por parte de los alumnos cuando son incorporadas a los grupos de investigación de los departamentos.

El objetivo final es facilitar al alumno la realización de las tareas asignadas en cada una de las materias del programa de doctorado. Su implantación real se está llevando a cabo en el primer curso del bienio de doctorado 2006-08 del programa de tercer ciclo TADSI perteneciente al Departamento de Informática de la Universidad de Vigo. En concreto, se ha planificado su utilización en las materias de "representación y análisis del conocimiento", "métodos de clasificación" y "sistemas de predicción".

A la vista de los resultados satisfactorios obtenidos dentro del grupo de investigación SING y de los proyectos asociados en los que está siendo utilizado, se espera un elevado nivel de satisfacción entre alumnos y docentes tras su implantación inicial en el programa en curso. A este respecto, actualmente están disponibles en el portal de AIBench los siguientes contenidos:

- La primera distribución pública del framework: *AIBench SDK 1.9.1*.
- Distintos plugins ya programados relacionados con el área de la bioinformática y la minería de datos.
- Un manual en línea para el programador de plugins, al que se remite al lector para profundizar en cuestiones concretas que caen fuera del alcance de este artículo.
- Un foro de discusión para que los usuarios puedan discutir sobre cualquier tema relacionado y resolver sus dudas.

Además, a través del portal de AIBench se anima a otros grupos de investigación a que envíen sus propios plugins a través del correo electrónico de contacto *contact@aibench.org*.

En cuanto al trabajo que se está llevando a cabo actualmente, destacar la creación de una nueva extensión del núcleo consistente en la codificación de un diseñador visual de experimentos. Se trata de un servicio íntimamente relacionado con la automatización de experimentos (apartado 3.2), pero que va un paso más allá. El objetivo es poner a disposición del programador un entorno en el que de modo gráfico (*drag & drop*), pueda diseñar la ejecución de sus operaciones. El diseño de un experimento consistirá en definir la secuencia de operaciones a ejecutar sobre un conjunto de datos dado, ofreciendo al usuario la posibilidad de guardar el experimento actual, recuperar los experimentos previamente almacenados, modificarlos, etc.

Agradecimientos

Los autores de este trabajo quieren agradecer y dejar constancia de la colaboración prestada por los proyectos asociados a AIBench. En concreto, al Departamento de Informática de la Universidad do Minho (Portugal), al Departamento de Informática de la Universidad de Coimbra (Portugal) y a la Fundación Biomédica del Complejo Hospitalario Universitario de Vigo (FICHUVI), sin cuya colaboración no sería posible la continua evolución y mejora de AIBench.

Referencias

[1] TADSI: *Tecnologías Avanzadas para el Desarrollo de Software Inteligente*. (2006). http://dellgwai.ei.uvigo.es/~postgrado/2005-07/Spanish/uno.html

[2] SING: *Sistemas Informáticos de Nueva Generación*. (2007). http://sing.ei.uvigo.es/

[3] AIBench: *the Artificial Intelligence workbench*. (2006). http://www.aibench.org/

[4] Platonos. (2004). http://platonos.sourceforge.net/

[5] BeanShell: *Lightweight Scripting for Java*. (2007). http://www.beanshell.org/

Invocación de Métodos Remotos: prácticas de laboratorio

Coromoto León Hernández, Gara Miranda Valladares
Dpto. de Estadística, Investigación Operativa y Computación
Universidad de La Laguna
c/Astrofísico Fco. Sánchez s/n - 38271 La Laguna – S/C de Tenerife
(cleon|gmiranda)@ull.es

Resumen

En este trabajo se presenta una propuesta de prácticas de laboratorio para el tema de Invocación de Métodos Remotos utilizando el lenguaje de programación Java, en el contexto de una asignatura de Programación de Sistemas Distribuidos. En primer lugar, se propone una práctica de iniciación al modo de uso y a continuación, la implementación de un servicio remoto que permita resolver problemas utilizando la técnica de Ramificación y Acotación, siempre y cuando el usuario especifique el problema utilizando un formato determinado.

1. Introducción

La asignatura optativa "Programación de Sistemas Distribuidos" de seis créditos se oferta en el tercer curso de la Ingeniería Técnica en Informática de Sistemas en la Universidad de La Laguna.

El contenido de la asignatura cuenta con temas de Comunicación entre Procesos, Paradigma Cliente/Servidor, Interfaces de Programación de Sockets (tanto orientados como no orientados a conexión) y Objetos Distribuidos (Llamadas a Procedimiento Remoto, Invocación de Métodos Remotos, CORBA) [1][6]. La Tabla 1 muestra el conjunto de prácticas de laboratorio que se han programado asociadas a estos temas.

Java [3] es el lenguaje de programación elegido para desarrollar los contenidos prácticos de la asignatura. Una de las principales razones de esta elección es la facilidad de aprendizaje del mismo si se conoce C++ y la gran cantidad de material docente disponible.

La programación imperativa precede a la programación orientada a objetos. En ella, un procedimiento o función es una estructura de control que proporciona la abstracción correspondiente a una acción. Una llamada a procedimiento convencional es una llamada a un procedimiento que reside en el mismo sistema que el que la invoca. En el modelo de llamada a procedimiento remoto, un proceso realiza una llamada a un procedimiento de otro proceso, que posiblemente reside en un sistema remoto. Desde su introducción a principios de los años 80, el modelo de llamadas a procedimiento remoto (*Remote Procedure Call* – RPC) ha sido muy utilizado en aplicaciones distribuidas. Sin embargo, puesto que el lenguaje elegido para realizar las prácticas de la asignatura es Java, en lugar de RPC se estudiará el paradigma de Invocación a Métodos Remotos (*Remote Method Invocation* – RMI) que es una implementación orientada a objetos del modelo RPC [4]. Java-RMI es una herramienta exclusiva para programas Java, aunque debido a su relativa simplicidad, constituye un buen comienzo para estudiantes que están empezando a utilizar objetos distribuidos.

En RMI, un *Servidor de Objetos* exporta un *Objeto Remoto* y lo registra en un *Servicio de Nombres*. El objeto proporciona métodos remotos que un *Cliente* puede invocar utilizando una sintaxis similar a las de invocación de métodos locales.

Práctica	Semanas
Creación de Hilos (*Threads*)	1
Sincronización de Hilos (*Threads*)	1
Direcciones IP y Nombres de dominio	1
Serialización de objetos	1
Comunicación no orientada a conexión	2
Comunicación orientada a conexión	2
Comunicación en grupos	1
Invocación de Métodos Remotos	2
CORBA	2
Servicio de resolución mediante Ramificación y Acotación	1
Total	14

Tabla 1. Propuesta de prácticas de laboratorio

El contenido del trabajo se estructura de la siguiente forma: en la sección 2 se describe el funcionamiento del paradigma de Invocación a Métodos Remotos, en la sección 3 se presenta el enunciado de las prácticas de laboratorio asociadas al tema y finalmente aparecen las conclusiones y los trabajos futuros.

2. Invocación a métodos remotos en Java

El mecanismo de *Invocación a Métodos Remotos* permite hacer algo que parece sencillo. Si se tiene acceso a un objeto en una máquina distinta, se podrá llamar a los métodos de ese objeto remoto. Esto implica que:

- De algún modo, los parámetros del método deben pasarse a la otra máquina.
- El objeto debe ser informado para que ejecute el método.
- El valor obtenido debe ser devuelto.

Las clases que proporciona Java-RMI permiten manipular de forma transparente todos estos detalles. Considérense las siguientes definiciones:

- *Objeto Cliente*: objeto cuyos métodos efectúan la llamada remota.
- *Objeto Remoto*: objeto localizado en el servidor de Objetos.

La definición de un objeto se utiliza para una única llamada a un método. Es posible invertir los papeles en cualquier punto del camino. Por tanto, el servidor de una llamada previa puede convertirse en cliente cuando llame a un método remoto de un objeto que reside en otra máquina.

Cuando un *Objeto Cliente* quiere invocar a un método de un objeto remoto, llama a un método ordinario que está encapsulado en un objeto denominado *registro* (*stub*). El objeto *stub* reside en la máquina del cliente, no en el servidor. El *stub* empaqueta como un bloque de bytes los parámetros utilizados en el método remoto (usando el mecanismo de serialización). El objetivo de esta operación es convertir los parámetros a un formato susceptible de ser enviado de una JVM (*Java Virtual Machine*) a otra. El objeto *stub* en el *Cliente* construye un bloque con la siguiente información y la envía al *Servidor*:

- Una descripción del método que será llamado.
- Los parámetros codificados.

En el *Servidor* el objeto receptor realiza las siguientes acciones por cada método remoto invocado:

- Decodifica los parámetros.
- Localiza al objeto que ha sido llamado.
- Encuentra el método requerido.
- Captura y ordena los valores a devolver, o la excepción de la llamada.
- Envía un paquete con los datos del resultado codificados al *stub* del Cliente.

El *stub* del *Cliente* decodifica el valor de retorno, o la excepción, procedente del *Servidor*. Este valor se convierte en el valor de retorno de la llamada al *stub*. Si el método remoto lanza una excepción, el *stub* la propaga en el ambiente en que tuvo lugar la llamada. Este proceso aunque complejo es completamente transparente para el programador. Además, los diseñadores de objetos remotos en Java trataron de darles el mismo aspecto y funcionamiento que los locales.

3. Prácticas de laboratorio

Las prácticas de laboratorio asociadas al tema son dos: una inicial para afianzar los conceptos y mostrar la metodología de desarrollo y otra de aplicación de los mismos.

3.1. Aplicación simple: suma de dos arrays

El ejercicio consiste en implementar una aplicación Cliente/Servidor usando Java-RMI en la cual el servicio que ofrece el Objeto Remoto es un método que suma los elementos de dos arrays.

En el programa *Servidor* se ha de tener una instancia de un *Objeto Remoto*. El programa *Cliente* ha de proporcionar dos arrays de enteros e invocar al método remoto para que realice la suma y devuelva el array resultado [5].

Se han de implementar las siguientes clases:

- `Arith.java` (definición de la interfaz)
- `ArithImpl.java` (implementación de la interfaz)
- `ArithServer.java` (implementación del servidor)
- `ArithClient.java` (implementación del cliente)

El programa cliente tiene que manipular ciertos objetos, pero no tiene copia de ellos puesto que residen en el servidor. Sin embargo, como tiene que saber cómo trabajar con ellos, su forma de funcionamiento la encuentra en una interfaz que es compartida por el cliente y el servidor y que reside en ambas máquinas.

Todas las interfaces de *Objetos Remotos* deben derivar de la interfaz `Remote` que está definida en el paquete `java.rmi`. Todos los métodos declarados en estas interfaces pueden lanzar una `RemoteException` puesto que siempre es posible que una llamada remota falle. Así pues, Java obliga a capturar una `RemoteException` en cada llamada a un método remoto, además de tener que especificar una acción adecuada en el caso de que dicha llamada no se produzca.

El servidor debe implementar la clase que indica cómo funcionan los métodos de la interfaz remota. Para que esta clase sea un servidor de objetos remotos hay que hacer que extienda a la clase `UnicastRemoteObject`, que es una clase concreta del paquete `java.rmi.server` que hace que los objetos sean accesibles remotamente.

Para acceder a un *Objeto Remoto* que se encuentra en un servidor, el cliente necesita un objeto `stub` local. ¿Cómo puede un cliente hacerse con un objeto de este tipo? La librería RMI de Sun ofrece un *Servicio de Registro de Nombres* para localizar al objeto servidor. El programa servidor registra objetos en ese servicio, y el cliente obtiene los `stub` a partir de esos objetos. Para registrar un objeto se debe suministrar al servicio de registro una referencia al objeto y un nombre (que debería ser único):

```
Naming.bind("nombre", obj);
```

El cliente obtiene un `stub` para acceder a un *Objeto Remoto* especificando el nombre del servidor y del objeto de la siguiente forma:

```
... =(TypeCast) Naming.lookup(
   "rmi://yourserver.com/nombre");
```

Los programas cliente que usen RMI deben cargar un administrador de seguridad para controlar las actividades de los `stubs` que se cargan dinámicamente.

`RMISecurityManager` es un administrador de seguridad que se instala con la siguiente instrucción:

```
System.setSecurityManager(
     new RMISecurityManager());
```

El programa cliente obtiene la referencia a un objeto `Arith` que se ha dado de alta en el *Registro de Nombres* RMI e invoca al método `add()` con dos arrays. Por defecto, `RMISecurityManager` prohíbe a todo el código del programa establecer conexiones de red. Pero son necesarias para acceder al registro RMI y para contactar con los objetos del servidor. Para ello se ha de proporcionar un archivo de política de seguridad e indicarle al administrador de seguridad que lea de ese archivo:

```
System.setProperty(
   "java.security.policy",
   "client.policy");
```

Los pasos a seguir para poner a funcionar la aplicación son los siguientes:

- Compilar los ficheros fuente de las clases interfaz, implementación de la interfaz, cliente y servidor:
  `javac Arith*.java`
- Compilar con `rmic` la clase de implementación para obtener el `stub`:
  `rmic -v1.2 ArithImpl`
- Inicializar el Registro de Nombre RMI con:
  `rmiregistry &`
- Ejecutar el servidor en una terminal:
  `java ArithServer`
- Ejecutar al cliente en otra terminal:
  `java ArithClient`

Asegurarse de que el fichero `client.policy` está en el directorio actual.

Además de la implementación anterior se proponen los siguientes ejercicios:

- Añada código al método `add()` de la clase `ArithImpl` de forma que haya un retardo de cinco segundos antes de que el método termine. Esto tiene el efecto de alargar de manera artificial la latencia de cada

invocación del método. Compile y arranque el servidor. En pantallas separadas arranque dos o más clientes. Observe la secuencia de eventos en las pantallas. ¿Se puede saber si el servidor de objetos ejecuta las llamadas a los métodos concurrente o iterativamente? Explíquelo.

- Implemente una aplicación Cliente/Servidor, utilizando el API de `Sockets`, en la cual el cliente envíe dos arrays de enteros al servidor, éste los sume y devuelva el array resultado al cliente. Utilizar la clase serializable `DataObj` como interfaz entre el cliente y el servidor. Implementar el servidor en una clase `ArithServer`. El servidor será un servidor simple que espera por una conexión. Cuando la conexión se establezca, leerá los objetos que le envía el cliente. Realizará la suma y devolverá el objeto resultado. Implementar el cliente en una clase `ArithClient`. El cliente será un cliente simple. Ha de definir dos objetos arrays con los que realizará su solicitud al servidor. Cuando recibe el resultado lo muestra por la pantalla.
- Utilice el método `currentTimeMillis()` de la clase `System` para establecer un cronómetro en los clientes de los ejercicios de Sockets y RMI. ¿Qué se puede decir acerca del tiempo que consume cada una de las implementaciones?

3.2. Aplicación compleja: servicio de resolución de problemas mediante ramificación y acotación

La técnica de Ramificación y Acotación es un método general que permite resolver un amplio rango de problemas de optimización combinatoria. Este tipo de paradigma pertenece a los denominados métodos enumerativos que se basan en la exploración del espacio de soluciones completo.

La técnica de Ramificación y Acotación consiste en explorar un grafo, normalmente acíclico o un árbol, buscando la solución óptima de un problema dado. En cada nodo, se calcula una cota del posible valor de aquellas soluciones que pudieran encontrarse más adelante en el grafo. Si la cota muestra que cualquiera de estas soluciones va a ser necesariamente peor que la mejor solución encontrada hasta el momento no se necesita seguir explorando esta parte del grafo. La Ramificación y Acotación utiliza cálculos auxiliares para decidir en cada momento qué nodo debe explorarse a continuación y una lista con prioridad para almacenar los nodos que se han generado y que se encuentran aún sin examinar. El orden en que se exploran los nodos puede variar según la implementación. Es típico usar una exploración en profundidad o una exploración en anchura, aunque también se puede realizar una exploración en la que se elija en primer lugar el nodo con mayor valor o realizar la decisión en base a otros criterios [2].

La implementación de un esquema genérico de resolución de problemas mediante esta técnica algorítmica conlleva el uso del siguiente conjunto de *interfaces* Java:

- `Problem`: define la interfaz mínima necesaria para definir un problema. Se debe sobrescribir el método `generateSubProblem()` que a partir del problema original debe generar el subproblema raíz del árbol de búsqueda.
- `SubProblem`: define la interfaz mínima necesaria para representar cada uno de los subproblemas en que se irá dividiendo el problema original. Los métodos que se han de implementar son:
  - `Solution solve(Problem p)`: devuelve la mejor solución encontrada para el subproblema actual.
  - `void branch(Problem sp, Solver s)`: genera cada uno de los subproblemas en los que el subproblema actual se ramifica, llamando al método `insert` del objeto s para cada uno de ellos.
  - `double lower_bound(Problem p)`: establece una cota inferior del valor de las soluciones que se conseguirán a partir del subproblema.
  - `double upper_bound(Problem p)`: establece una cota superior del valor de la soluciones que se conseguirán a partir del subproblema.
- `Solution`: esta interfaz debe ser implementada por la clase que representa la solución al problema. No presenta ningún método.

Se considerará como caso de estudio el Problema de Mochila 0-1. Su formulación es la siguiente:

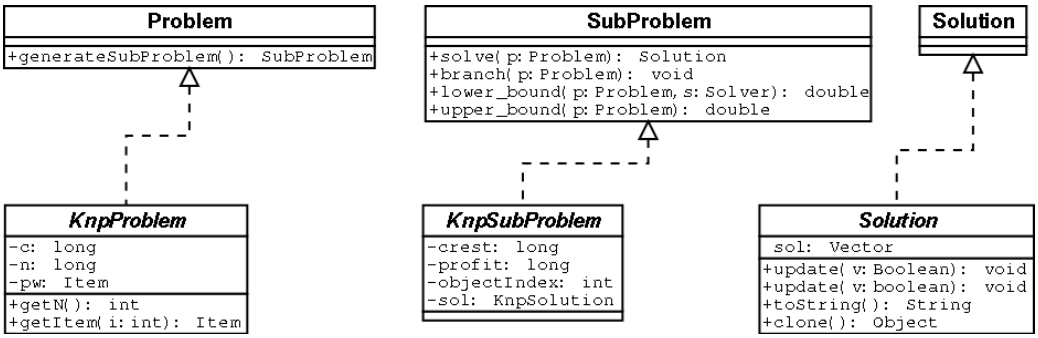


Figura 1. Diagrama UML para las clases que representan un problema

Dada una mochila de capacidad C y N objetos con pesos p_i y beneficios b_i para i = 1,...,N, se trata de guardar en la mochila aquellos objetos que maximicen el beneficio y no sobrepasen la capacidad. Se crean clases que implementan a las interfaces `Problem`, `SubProblem` y `Solution` para este problema concreto. La Figura 1 muestra el diagrama UML del conjunto de clases que componen el programa.

La clase `KnpProblem` contiene la definición de la capacidad, el número de objetos y el vector con los pesos y los beneficios. Es la que se muestra en la Figura 2.

La definición de la solución como un vector de booleanos que indican si se incluye o no un objeto se hace en la clase `KnpSolution` que aparece en la Figura 3.

```
public class KnpProblem
         implements Problem {
  private long c;
  private long n;
  private Item [] pw;
  ...
  public SubProblem
    generateSubProblem(){
    KnpSubProblem sp = new
    KnpSubProblem(c, 0, 0,
         new KnpSolution());
    return sp;
  }
}
```

Figura 2. Definición de la clase del problema

La clase `KnpSubProblem` contiene la definición de los subproblemas. Su implementación es la que aparece en al Figura 4. Se trabaja con: la capacidad restante, el beneficio acumulado, el objeto en estudio y la solución hasta el momento (esto es, los objetos que se han insertado o no hasta llegar al que está en estudio). Nótese que implementa todos los métodos de la interfaz `SubProblem`.

```
import java.util.*;

public class KnpSolution
 implements Solution, Cloneable {
  Vector sol;

  KnpSolution(){
    sol = new Vector();
  }

  public void update(Boolean v){
    sol.add(v);
  }

  public void update(boolean v){
    sol.add(new Boolean(v));
  }

  public String toString(){
  ...
  }

  public Object clone(){
  ...
  }
```

Figura 3. Definición de la clase solución

```
public class KnpSubProblem implements SubProblem {
  private long crest, profit; //Capacidad restante y beneficio actual
  private int objIndex;       //objeto en estudio
  private KnpSolution sol;    //solución
  ...
  public double upper_bound(Problem p){
    KnpProblem pm = (KnpProblem)(p);
    long weigth = 0, pft = profit, lastObject = pm.getN();
    for (int i = objIndex; i < lastObject; i++){
      if ((weigth + pm.getItem(i).getWeigth()) <= crest){
        weigth += pm.getItem(i).getWeigth(); pft += pm.getItem(i).getProfit();
      } else break;
    }
    if (i < lastObject){return (pft + (pm.getItem(i).getProfit()*
                                (crest - weigth)/pm.getItem(i).getWeigth()));
    }else return pft;
  }

  public double lower_bound(Problem p){
    KnpProblem pm = (KnpProblem)(p);
    long weigth = 0, pft = profit, lastObject = pm.getN();
    for (int i=objIndex; ((i < lastObject) && (weigth < crest)); i++){
      if ((weigth + pm.getItem(i).getWeigth()) <= crest){
        weigth += pm.getItem(i).getWeigth(); pft += pm.getItem(i).getProfit();
    } }
    return pft;
  }

  public Solution solve(Problem p){
    KnpProblem pm = (KnpProblem)(p);
    long weigth = 0, lastObject = pm.getN();
    KnpSolution sol2 = (KnpSolution)sol.clone();
    for (int i= objIndex; ((i < lastObject) && (weigth < crest)); i++){
      if ((weigth + pm.getItem(i).getWeigth()) <= crest){
        weigth += pm.getItem(i).getWeigth(); sol2.update(true);
      } else{ sol2.update(false);}
    }
    for (; i < lastObject; i++)sol2.update(false);
    return sol2;
  }

  public void branch(Problem pr, Solver s) {
    KnpProblem pm = (KnpProblem)(pr);
    if (objIndex < pm.getN()){
      KnpSubProblem spNo = new KnpSubProblem(crest, objIndex + 1,
                                   profit, (KnpSolution)sol.clone());
      s.insert(spNo); spNo.sol.update(false);
      long cap = pm.getItem(objIndex).getWeigth();
      if (cap <= crest){
        KnpSubProblem spYes = new KnpSubProblem(crest-cap, objIndex+1,
                              profit+pm.getItem(objIndex).getProfit(),
                              (KnpSolution)sol.clone());
        spYes.sol.update(true); s.insert(spYes);
    } }
  }
}
```

Figura 4. Definición de la clase del subproblema

```
 1  import java.util.*;
    ...
 2  Stack spList;
    ...
 3  public Solution run(Problem pbm) {
 4    double bestActual = -1;
 5    SubProblem actualsp;
 6    Solution sol = null;
 7
 8    spList.push(pbm.generateSubProblem());
 9
10    while(!spList.empty()){
11      double upper;
12      actualsp = (SubProblem)spList.pop();
13      if ((upper = actualsp.upper_bound(pbm)) > bestActual){
14        double lower;
15        if ((lower = actualsp.lower_bound(pbm)) > bestActual){
16          bestActual = lower;
17          sol = actualsp.solve(pbm);
18        }
19        if (upper != lower)
20          actualsp.branch(pbm, this);
21        }
22    }
23    return sol;
24  }
25  public void insert(SubProblem sp) {
26    spList.push(sp);
27  }
```

Figura 5. Esquema genérico de Ramificación y Acotación

La Figura 5 muestra un esquema genérico para resolver problemas de maximización mediante la técnica de Ramificación y acotación. Se utiliza la clase `java.util.Stack` para definir la estructura `spList` en la que se va a almacenar el recorrido del árbol de búsqueda de la solución (línea 2). El método `run()` es el encargado de realizar la búsqueda (líneas 3-24). En primer lugar se inserta en la lista de subproblemas, `spList`, el problema original (línea 8). En el bucle principal, mientras la lista no esté vacía (línea 10), se extrae un problema de la lista (línea 12), se calcula la cota superior del mismo (línea 13) y si es mejor que la mejor solución hasta el momento, se ramifica invocando al método `branch()` (línea 20). Además, se calcula la cota inferior (línea 15) y si es mejor que la mejor solución hasta el momento se sustituye por la que se acaba de calcular. El método `insert()` (líneas 25-27) es utilizado por el método `branch()` para insertar en `spList` los nuevos problemas que se generen. Estos métodos están implementados en la clase `Solver`.

El uso conjunto de todas las clases se implementa a través de una clase `Main` en la que se puede especificar el siguiente ejemplo:

Capacidad = 104;
Número de items = 8;
Pesos = { 25, 35, 45, 5, 25, 3, 2, 2};
Beneficios = {350, 400, 450, 20, 70, 8, 5, 5};
Solución = {1,3,4,5,7,8}
Beneficio máximo = 900.

Los ejercicios que se proponen al estudiante son:

- Compilar y ejecutar el ejemplo del Problema de la Mochila.
- Utilice el método `currentTimeMillis()` de la clase `System` para establecer un cronómetro en la clase `Main` que permita conocer el tiempo que se consume en la

resolución del problema. ¿Cuánto tiempo consume la implementación local?

- Implementar una aplicación Cliente/Servidor usando Java RMI en la cual el servicio que ofrece el objeto remoto es la resolución mediante la técnica de Ramificación y Acotación de un problema especificado. En el cliente se localizará y preguntará por el objeto remoto que resuelve un problema dado. Esto se implementará con las siguientes clases:
  - `Solver` (definición de la interfaz)
  - `SolverImpl` (implementación de la interfaz)
  - `SolverServer` (implementación del servidor)
  - `SolverClient` (implementación del cliente)
- Utilice el método `currentTimeMillis()` de la clase `System` para establecer un cronómetro en el cliente. ¿Cuánto tiempo consume la implementación distribuida?
- ¿Qué conclusión puede dar sobre los dos tipos de implementación?

4. Conclusión

Se ha presentado un conjunto de prácticas de laboratorio para cubrir el tema de Invocación a Métodos Remotos en una asignatura con contenidos de Programación de Sistemas distribuidos. La principal novedad es la propuesta de implementación de un servicio que proporciona la posibilidad de resolver problemas utilizando la técnica de Ramificación y Acotación, siempre que el usuario especifique el problema utilizando un formato determinado. Este esquema es genérico y se puede extender a otras técnicas algorítmicas como Divide y Vencerás o Programación Dinámica.

Para la evaluación de la propuesta se ha realizado una encuesta a los alumnos. El número total de respuesta fue 24. Una de las preguntas planteadas fue: ¿Considera la práctica final 'Servicio de resolución mediante Ramificación y Acotación' más compleja que las anteriores? Si / No. Los resultados de la encuesta muestran que el 96% de los alumnos no la consideran más compleja que las anteriores. Otra de las preguntas de la encuesta pedía que se valorara el grado de interés por cada práctica en una escala que iba de nada interesante a muy interesante. La Tabla 2 muestra los resultados obtenidos sumando como porcentaje de interés todos lo que estaban por encima del interesante por razones de espacio.

Práctica	%interés
Creación de Hilos (*Threads*)	100
Sincronización de Hilos (*Threads*)	96
Direcciones IP y Nombres de dominio	92
Serialización de objetos	87
Comunicación no orientada a conexión	96
Comunicación orientada a conexión	100
Comunicación en grupos	87
Invocación de métodos remotos (*Java – RMI*)	96
CORBA (*Java-IDL*)	100
Servicio de resolución mediante Ramificación y Acotación	92

Tabla 2. Resultados de la encuesta sobre interés de las prácticas de laboratorio

5. Agradecimientos

Este trabajo se ha realizado en el marco del plan Nacional de I+D+I bajo el proyecto con número de referencia TIN2005-08818-C04-04. El trabajo de Gara Miranda Valladares ha sido financiado con la beca FPU-AP2004-2290.

Referencias

[1] G. Coulouris, G., Dollimore, J., Kindberg, T. *Sistemas Distribuidos: Conceptos yDiseño*, Addison-Wesley, 3ra. Edicion, 2001.

[2] Dorta, I., Dorta, P., León, C., Rojas, A. *Utilización de Software en la Docencia de Técnicas Algorítmicas*. JENUI'2001. Páginas 190-195.

[3] Java 2 Standard Edition (J2SE) http://java.sun.com/j2se/

[4] Liu, M. L. *Computación Distribuida. Fundamentos y Aplicaciones.* Addison-Wesley, 1rd. Edición, 2004.

[5] Mahmoud, Qusay H. *Distributed Programming with Java*. Manning, 1999.

[6] Tanenbaum, A, van Steen, M. *Distributed Systems: Principles and Paradigms*, Prentice Hall, 2002.

Enseñanza asistida de teoría de autómatas y lenguajes formales mediante el uso de THOTH

César García Osorio, Andrés Arnáiz Moreno, Álvar Arnáiz González
Departamento de Ingeniería Civil
Universidad de Burgos
Avda. Cantabría s/n, 09006, Burgos
cgosorio@ubu.es, arnaizmoreno@hotmail.com, aagcaballeo@hotmail.com

Resumen

La teoría de autómatas y lenguajes formales suele ser percibida como árida y aburrida por los estudiantes. El uso de numerosos ejemplos es indispensable para facilitar la compresión de los conceptos. El problema es que estos suelen ser tediosos de escribir a mano o excesivamente simples. Thoth es una herramienta que pretende asistir en la docencia de los cursos de autómatas y lenguajes formales proporcionando al alumno una forma de interaccionar con conceptos y algoritmos. En su desarrollo se ha priorizado dos aspectos, la facilidad de utilización, y la reutilización de su código como applets que puedan integrarse en la web. En esta su primera versión se centra en los autómatas finitos, las expresiones regulares y las gramáticas independientes del contexto.

1. Introducción

Uno de los principales problemas de la enseñanza de las asignaturas de teoría de autómatas y lenguajes formales es la percepción que tienen los alumnos de la misma como demasiado compleja y teórica. Esta percepción se puede aliviar con la abundante resolución de ejercicios y el uso de ejemplos. Desafortunadamente estos son o muy simples o largos y tediosos de hacer a mano. Además los alumnos están acostumbrados a usar el ordenador en otras asignaturas. Aquí presentamos Thoth, una herramienta de apoyo a la docencia de teoría de autómatas y lenguajes formales. Thoth permite la simulación de los dispositivos de esta teoría y la ejecución paso a paso de sus algoritmos. Inicialmente estaba pensada para trabajar exclusivamente con autómatas finitos y expresiones y gramáticas regulares, pero en su versión actual también incluye algoritmos para gramáticas independientes del contexto, tales como la obtención de la forma normal de Chomksy, la generación de tablas de análisis LL(1) y el algoritmo de Cocke-Younger-Kasami. Existen otras herramientas que cubren un espectro más amplio de dispositivos y algoritmos, la ventaja de Thoth frente a éstas es que esta en castellano y que presenta una muy cuidada interfaz gráfica. La edición de un autómatas finito se ha simplificado enormemente, es posible dibujar y ubicar todos sus elementos, estados y transiciones, con simples operaciones de ratón sin necesidad de seleccionar previamente en un menú o botón el elemento a dibujar. Además, muestra mucha más información en pantalla sobre la estructura del autómata.

Antes de dar más detalles sobre la funcionalidad que Thoth proporciona, la sección que sigue da un rápido repaso a los principales conceptos de la teoría de lenguajes y autómatas. A continuación enumeramos y describiremos algunas otras herramientas que como Thoth asisten en la enseñanza de dichos conceptos. La siguiente sección detalla los algoritmos y operaciones con los que podemos interaccionar en Thoth. El artículo finaliza con una sección de conclusiones y de trabajo futuro.

2. Conceptos de la teoría de autómatas y lenguajes formales

Los tres conceptos fundamentales con los que se trabaja en THOTH son: gramática independiente del contexto, expresión regular y autómata finito. En esta sección explicamos brevemente en que consisten.

Las *gramáticas formales* permiten describir lenguajes. Desde un punto de vista formal, los *lenguajes* no son más que un conjunto de cadenas construidas utilizando un conjunto de símbolos de un alfabeto. Las gramáticas dan las reglas de cómo formar las cadenas que constituyen el

lenguaje. Formalmente una gramática esta constituida por los siguientes elementos, el alfabeto sobre el que construye las cadenas, un conjunto de símbolos auxiliares denominados *símbolos no terminales* (en contraste los elementos del alfabeto se llaman *símbolos terminales*), un conjunto de reglas de reescritura, llamadas *producciones* (indican como transformar unas cadenas en otras) y un símbolo especial dentro del conjunto de símbolos no terminales denominado *axioma* de la gramática, que inicia el proceso generativo. Así el lenguaje que genera una gramática esta constituido por las cadenas de símbolos del alfabeto que podemos derivar a partir del axioma aplicando las transformaciones definidas en las reglas o producciones. Cualquier subcadena que aparezca en la parte izquierda de una regla se puede sustituir por su parte derecha. Una gramática con las siguientes reglas { $S \rightarrow aSB$ (1), $S \rightarrow a$ (2), $aB \rightarrow a$ (3), $aB \rightarrow Ba$ (4), $B \rightarrow b$ (5)} (S es el axioma) permite generar todas las cadenas de aes y bes con un número de aes mayor que número de bes. Por ejemplo, la cadena *baaa*, se podría derivar del axioma con la secuencia $\underline{S} \Rightarrow^1 a\underline{S}BB \Rightarrow^1 aa\underline{S}BB \Rightarrow^2 aa\underline{aB}B \Rightarrow^4 a\underline{a}BaB \Rightarrow^4 \underline{a}BaaB \Rightarrow^4 \underline{B}aaaB \Rightarrow^3 baa\underline{aB} \Rightarrow^5 baaa$ (en cada paso se muestra subrayada la subcadena substituida, en regular la nueva cadena y el superíndice índica la regla aplicada). La anterior secuencia de cadenas se denomina *derivación*, cada una de las cadenas obtenidas *forma sentencial*, y la cadena finalmente obtenida, formada exclusivamente por símbolos terminales, *palabra* o *sentencia*.

Dependiendo de la estructura de las producciones se establece una jerarquía de gramáticas y lenguajes. En THOTH sólo se utilizan las *gramáticas independientes del contexto*, los lenguajes que generan (o describen) estas gramáticas se denominan *lenguajes independientes del contexto*. En las gramáticas, independientes del contexto, las partes izquierdas de las reglas sólo pueden tener un símbolo no terminal. Un caso especial de gramáticas independientes del contexto son las *gramáticas regulares*. En estas gramáticas además se restringe a que en la parte izquierda sólo pueda aparecer un símbolo terminal, o dos símbolos el primero de ellos terminal y el segundo no terminal. Las gramáticas regulares permiten generar los llamados *lenguajes regulares*.

Una forma alternativa de describir los lenguajes regulares son las *expresiones regulares*. Los lenguajes regulares se caracterizan porque se pueden definir aplicando las operaciones de unión, concatenación y cierre a lenguajes que sean regulares. Las operaciones de unión y concatenación no son más que la unión y producto cartesiano de conjuntos. El cierre no es más que una concatenación repetida. Las expresiones regulares definen una sintaxis para describir estas operaciones. La unión se representa con el operador '|' (algunos autores usan el carácter '+' para representar este operador), la concatenación no tiene un operados explícito (algunos autores utilizan el carácter '·'), y el cierre con un operador sufijo '*'. De este modo la expresión regular $(b|c)(aa)^*a$ representa el lenguaje de las cadenas que comienzan por *b* o *c* y van seguidas de un número impar de aes.

En teoría de lenguajes formales se suele hacer una división entre dispositivos reconocedores de lenguajes y dispositivos descriptores o generadores. Las gramáticas y expresiones regulares pertenecen a esta segunda categoría. En la primera tenemos las máquinas de Turing, los autómatas de pila y los autómatas finitos. Mientras que las máquinas de Turing pueden reconocer cualquier lenguaje que pueda ser descrito por una gramática, los autómatas de pila sólo pueden reconocer los lenguajes independientes del contexto, y los autómatas finitos sólo los lenguajes regulares. En la versión actual de THOTH no se implementan ni las máquinas de Turing ni los autómatas de pila.

Los *autómatas finitos* constan de un conjunto de *estados*, en cada momento, sólo uno de ellos puede ser el estado actual, y de unas *transiciones* etiquetadas que nos indican como cambiar el estado actual dependiendo de cual sea el símbolo en curso. El autómata se puede representar como un grafo, cuyos nodos representan estados y cuyos arcos representan transiciones. Si partiendo del estado inicial, y siguiendo los arcos etiquetados con los símbolos que constituyen una cadena, logramos llegar a un estado etiquetado como final, la cadena se considera reconocida por el autómata. El conjunto de todas las cadenas que el autómata reconoce constituye el *lenguaje reconocido por el autómata*.

Esta sección ha dado una descripción necesariamente somera de conceptos, los libros de

Kelley [1], Brookshear [2] y Alfonseca [3] son excelentes para obtener descripciones más amplias.

3. Otras herramientas

En esta sección presentamos otras herramientas que pueden ser usadas para facilitar el aprendizaje de la teoría de autómatas y lenguajes formales. El principal inconveniente de la mayoría de ellas es que no se encuentras disponible en castellano.

- *Grail+* (http://www.csd.uwo.ca/Research/ grail/ index.html): se trata de una colección de programas escritos en C++ para manipular autómata finitos y expresiones y lenguajes regulares. Estos programas se pueden utilizar como filtros en la línea de comandos o como una libraría cuya funcionalidad se puede incluir en otros programas C++. Por tanto carece de una interfaz gráfica de usuario propiamente dicha. En la actualidad es un proyecto abandonado, la última actualización de su página web se remonta al 2002.
- *Simple Java regular expression parser and recognized* (http://www.oursland.net/projects/regexp/): Se trata de un analizador de expresiones regulares en modo texto. Permite conocer si una cadena pertenece al lenguaje definido por una expresión regular. Además muestra por pantalla el autómata construido y el autómata mínimo resultante. No está orientado al aprendizaje de la materia. Al ser toda la información en modo texto no resulta demasiado atractivo de cara al usuario final.
- *The finite state machine explorer* (http://www.belgarath.org/java/fsme.html): Se trata de una aplicación que permite la simulación de un autómata finito. Disponible una versión de la aplicación, con capacidad para cargar y guardar en ficheros, y un applet para poder utilizarle en el navegador. Ofrece una pobre interfaz de usuario y dibujar el diagrama de transiciones para un autómata se hace muy pesado, pero es capaz de reconocer palabras iluminando los estados actuales en cada momento. Permite mover los estados y redibujarlos. No muestra apenas información por pantalla de las operaciones que realiza, está desarrollado para mostrar una solución al problema no para mostrar como se consigue. Es muy específico y no presenta grandes funcionalidades. Resulta muy lento y pesado el uso de su entorno gráfico, demasiados botones con funciones que pueden agruparse dentro del mismo botón.
- *Visual automaton simulator* (http://www.cs.usfca.edu/~jbovet/vas.html): Es un simulador de autómatas finitos y máquinas de Turing en modo gráfico. Permite guardar y salvar en disco los elementos creados. Es la primera herramienta encontrada con una interfaz de usuario algo compleja, muestra la construcción de las máquinas paso a paso. Permite validar cadenas tanto en autómatas finitos como máquinas de Turing. No implementa algoritmos importantes a la hora de trabajar con los autómatas, solo permite validar cadenas y eliminar el determinismo. Además para el reconocimiento de palabras es necesario especificar el alfabeto del autómata en lugar de calcularlo automáticamente. La introducción de nuevos estados (algo muy habitual) resulta muy costosa, esto hace que la aplicación resulte incómoda al usuario final. Los archivos guardados en disco resultan muy extensos, ocupando espacio innecesario para almacenar autómatas muy sencillos. Presenta un lienzo poco apto para autómatas complejos, debido a su escaso tamaño.
- *Java formal language and automata processor* (*JFLAP*) (http://www.jflap.org/): Se trata de una herramienta muy completa y bien documentada [4]. La cualidad más destacable de esta aplicación es la gran cantidad de operaciones que se pueden llevar a cabo. Puede trabajar con distinta tipos de máquinas: autómatas finitos, autómatas de pila y máquinas de Turing. En su última versión permite utilizar máquinas de Turing como elementos constructivos de otras máquinas de Turing, lo que ayuda mucho en el diseño de máquinas de Turing complejas. También es capaz de trabajar con gramáticas y expresiones regulares. Su interfaz gráfica es fácil e intuitiva y está bastante lograda, aunque puede ser mejorada considerablemente (sólo se pueden usar líneas rectas para las transiciones, necesitamos cambiar de herramienta de dibujo para completar el diseño del autómata, no es posible mover

grupos de estados). Al ser demasiado extenso no profundiza en todos los aspectos, sobre todo en gramáticas y expresiones regulares, y hay algoritmos que no implementa. No implementa el *algoritmo de Cocke-Younger-Kasami* (CYK) [5,6] ni el *método de las derivadas* [7].

- *Simulations in Automata Theory* (http://www.cs.iitm.ernet.in/tell/automata/Automata/): Es un conjunto de applets para la simulación de distintos dispositivos. Dispone de un simulador de autómatas finitos, de autómata de pila, máquina de Turing y el algoritmo CYK.
- *Anagra* (http://webdiis.unizar.es/~ezpeleta/COMPI/compiladoresI.htm): Mientras que las anteriores herramientas estaban orientadas en su mayor parte a la simulación de autómatas, esta aplicación está orientada a la simulación y análisis de gramáticas (independientes del contexto y regulares). Es una herramienta que permite el análisis sintáctico asociado a gramáticas. Es capaz de realizar tanto el análisis ascendente como el descendente. También permite transformar gramáticas de un tipo a otro según la jerarquía establecida por Chomsky. Anagra es una herramienta muy completa, implementa gran cantidad de transformaciones que podemos aplicar sobre nuestras gramáticas. Sin embargo, desde nuestro punto de vista, esta limitada por la escasez de información, ya que da por hecho que el usuario conoce toda la información y no muestra los pasos intermedios de las transformaciones realizadas. Podría resultar muy útil para comprobar si un ejercicio se ha resuelto correctamente, pero no para aprender su funcionamiento.

De todas estas herramientas las más completas son las dos últimas, y las que estamos tomando de inspiración en el desarrollo de Thoth. El punto fuerte de JFLAP es que cubre una amplia variedad de dispositivos de reconocimiento. Anagra la complementa con su amplia variedad de algoritmos de análisis para gramáticas independientes del contexto. Thoth aspira a integrar las funcionalidades de ambas, corrigiendo algunas de sus carencias y mejorando su interfaz gráfica.

En su versión actual la funcionalidad de nuestra aplicación es perfectamente competitiva con JFLAP en le referente a las operaciones relativas a autómatas, expresiones regulares y a gramáticas. La interfaz gráfica es mucho más conveniente y además incorpora funcionalidad adicional para la conversión entre expresiones regulares y autómatas finitos y el análisis de cadenas.

La internacionalización, Thoth permite elegir entre cuatro lenguajes: español, inglés, francés y alemán, es otro punto a favor de nuestra aplicación ya que la hace mucho más atractiva para usuarios de habla no inglesa, como por ejemplo en Latinoamérica o Europa.

4. Algoritmos implementados en Thoth

Uno de los aspectos de diseño de THOTH ha sido la facilidad de uso. THOTH presenta en su interfaz de usuario un montón de funcionalidades que no están presentes en otras herramientas similares. Es posible agrupar y mover partes del autómata para facilitar que usuario minimice los cruces. También permite distribuir los estados de un autómata de forma automática en el área de dibujo (en esta primera versión simplemente los distribuye en torno a un círculo). Muchas de las operaciones más habituales no necesitan cambiar de herramienta de dibujo. Los arcos no se limitan a ser líneas rectas como en otras aplicaciones, con lo que es más fácil evitar los cruces de arcos.

4.1. Trabajando con gramáticas

Thoth permite introducir una gramática utilizando la misma sintaxis que la utilizada por bison [8]. Una vez introducida la gramática, determina el tipo de la misma y los alfabetos de terminales y no terminales (Figura 1, izquierda), entonces podemos llevar a cabo varias operaciones sobre la gramática (Figura 1, derecha).

Existen tres tipos de operaciones que podemos hacer con una gramática. Transformarla en otra que represente el mismo lenguaje, pero en la que sus producciones sigan un tipo particular de estructura. Utilizarla para determinar si una cadena particular pertenece al lenguaje que representa. Si la gramática es regular, obtener el autómata finito que reconoce el lenguaje que la gramática representa. En THOTH las posibles transformaciones tienen que ver con la

eliminación de: *producciones unitarias* (producciones que únicamente transforman un no terminal en otro no terminal), *producciones epsilón* (producciones que generan la palabra vacía, hacen que un no terminal desaparezca de la forma sentencial), *símbolos no alcanzables* (símbolos que no van a aparecer en ninguna forma sentencia que se pueda derivar desde el axioma de la gramática), *símbolos no terminables* (símbolos que nunca podrán transformarse en una cadena de terminales), *recursión directa* e *indirecta* por la izquierda (las gramáticas con este tipo de recursión no son adecuadas para el análisis descendente). En cuanto al análisis de cadenas, en THOTH es posible utilizar un algoritmo general, el *algoritmo de Cocke-Younger-Kasami* (Figure 2, izquierda) y un análisis de tipo descendente (la parte derecha de la Figura 2 muestra la tabla de análisis sintáctico predictivo utilizada en este tipo de análisis).

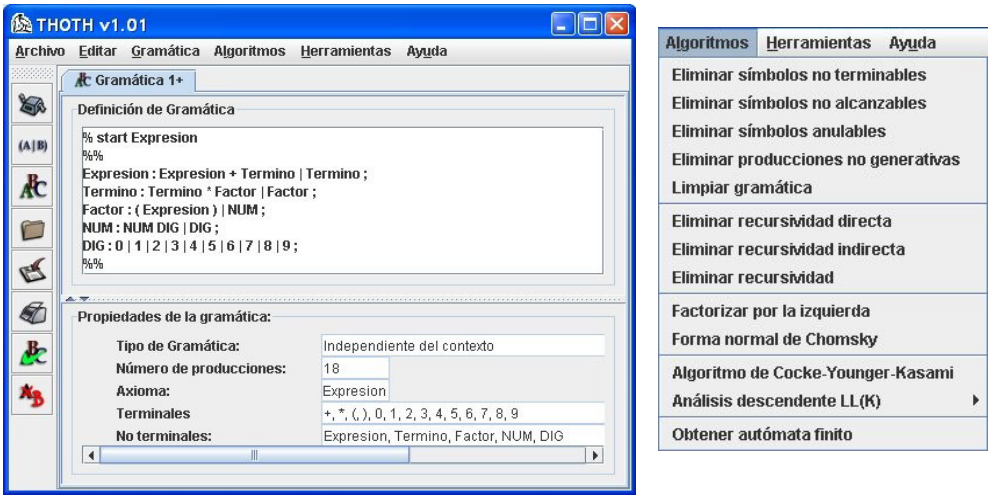


Figura 1. Izquierda: Entrada de gramática. Derecha: Algoritmos sobre gramáticas.

4.2. Trabajando con expresiones regulares

Cuando tratamos con expresiones regulares podemos tener en mente tres tipos de operaciones: obtener el autómata finito que reconoce el lenguaje representado por la expresión regular, determinar si una palabra en particular pertenece a dicho lenguaje, o determinar si dos expresiones regulares representan el mismo lenguaje. Estas dos últimas operaciones no las podemos llevar a cabo directamente en Thoth, aunque si de forma indirecta obteniendo primero el autómata finito que reconoce el lenguaje representado por la expresión regular, y entonces comprobando si la una cadena pertenece al lenguaje o si el autómata obtenido es equivalente al obtenido de otra expresión regular.

Thoth implementa tres algoritmos distintos para obtener el autómata finito equivalente a una expresión regular: el método de las derivadas [7], el método que Aho, Sethi y Ullman explican su su conocido "libro del dragón" [9] (algoritmo 3.5, que presenta una conexión muy estrecha con el método descubierto simultáneamente por Glushkov [10] y por McNaughton y Yamada [11]), y el método de Thompson's [12]. En la Figura 3 se puede ver el resultado de aplicar los primeros métodos a la expresión regular $a^*b|c^*$. Ninguno de estos dos métodos se ha encontrado implementado en las herramientas analizadas en la sección 3.

4.3. Trabajando con autómatas

Cuando se usa Thoth para trabajar con un autómata finito podemos o bien obtener primero un autómata a partir de una expresión regular, utilizando cualquiera de los métodos comentados en la sección previa, o bien dibujar directamente el diagrama de transiciones del autómata.

Dibujar un diagrama de transiciones con Thoth es un método muy directo. Se dispone de una herramienta para dibujar estados y otra para conectarlos mediante transiciones (en realidad el proceso se ve acelerado dado que no necesitamos cambiar la herramienta de dibujo en absoluto, es posible simplemente hacer doble click sobre el estado origen y arrastrar al estado de destino). Podemos mover estados individuales o agruparlos y moverlos todos juntos, de esta forma podemos minimizar el cruce de transiciones y presentar un diseño claro del autómata. Es posible incluso hacer que Thoth reubique los estados por nosotros. Todas estas características dan a Thoth una clara ventaja frente a otras herramientas para trabajar con autómatas finitos.

Una vez disponemos de un autómata podemos comprobar si reconoce una determinada palabra, podemos eliminar el no determinismo o minimizar el número de estados cuando esto es posible. Podemos también comprobar su equivalencia con otro autómata. Finalmente, podemos obtener la gramática o la expresión regular que representan el mismo lenguaje que el autómata reconoce. Thoth obtiene la expresión regular equivalente a un autómata usando el método de las ecuaciones características (también conocido como el método algebraico de Brzozowski [7]). Este es un método que no se ha encontrado implementado en otras herramientas. En la Figura 4 podemos ver el diagrama de transiciones de un autómata y el menú de algoritmos disponibles para trabajar con autómatas.

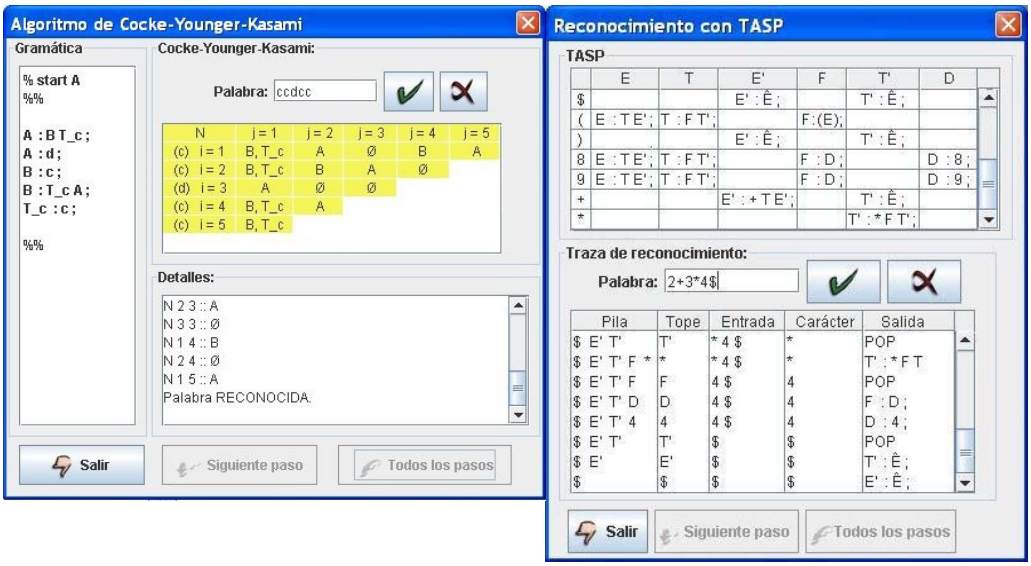


Figura 2. Izquierda: Algoritmo de Cocke-Younger-Kasami. Derecha: Tabla de análisis sintáctico predictivo.

5. Conclusiones y trabajo futuro

Hemos presentado una herramienta que pretende facilitar al alumno la interacción con algoritmos que siempre es más visual y menos pesada que las trazas con lápiz y papel, permitiendo la experimentación y observando la evolución paso a paso de los algoritmos.

La principal ventaja de Thoth frente a otras herramientas es su interfaz de usuario, que permite el fácil y rápido diseño de autómatas dibujando estados y conectándolos con transiciones. La evolución que presenta de los algoritmos esta orientada al aprendizaje de los mismos y no se limita a ser una simple herramienta de corrección de ejercicios. Además proporciona de forma integrada algoritmos no presentes en otras herramientas.

Sin embargo Thoth es una herramienta joven y con numerosas mejoras aún pendientes. Actualmente estamos trabajando en la

incorporación de los autómatas de pila y las máquinas de Turing. Además planeamos añadir los algoritmos de análisis ascendente.

También hemos identificado mejoras en su interfaz, en su versión actual no hay posibilidad de deshacer acciones, esperamos que en futuras versiones este sea otro de los puntos fuertes de su interfaz. Por último esperamos refinar la granularidad con la que se muestran los pasos de ejecución de los algoritmos para proporcionar al alumno la mayor información posible sobre los mismos.

Lamentablemente todavía no se dispone de una evaluación formal por parte de los alumnos, la asignatura de "Autómatas y Lenguajes Formales" en Burgos es una asignatura del primer cuatrimestre y la versión estable de la aplicación no estuvo disponible a tiempo como para poder haber sido utilizada intensivamente. Este es otro trabajo pendiente, junto con el de las mejoras previamente comentadas. La versión comentada en este artículo se encuentra disponible para su descarga en la dirección: http://pisuerga.inf.ubu.es/cgosorio/THOTH/ y cualquier comentario sobre la misma será bienvenido.

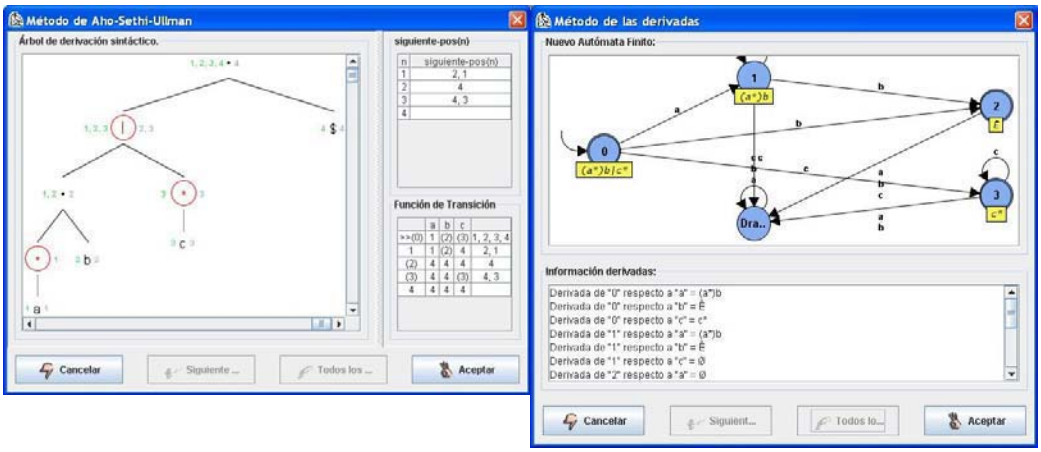


Figura 3. Izquierda: Método de Aho-Sethi-Ullman. Derecha: Método de las derivadas.

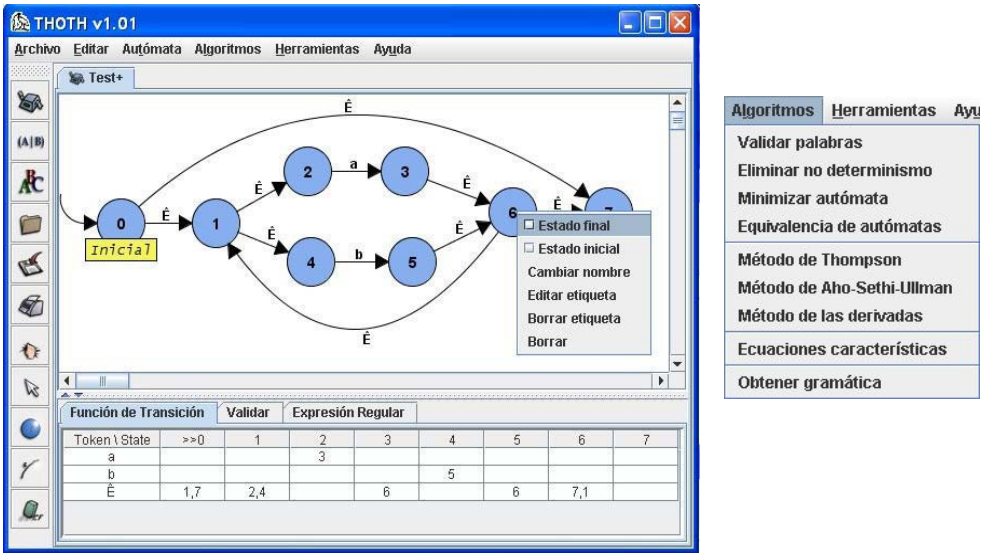


Figura 4. Izquierda: Entrada de autómata. Derecha: Algoritmos sobre autómatas.

Agradecimientos

La realización de este trabajo ha sido posible gracias al proyecto BU004B06 de la Consejería de Educación de la Junta de Castilla y León.

Referencias

[1] Dean Kelley. *Automata and Formal Languages: An Introduction.* Prentice Hall, 1998.

[2] Dean Kelley. *Automata and Formal Languages: An Introduction.* Prentice Hall, 1998.

[3] J. Glenn Brookshear. *Theory of Computation: Formal Languages, Automata, and Complexity.* Benjamin-Cummings Publishing Company, 1989.

[4] Manuel Alfonseca, Enrique Alfonseca y Roberto Moriyón. *Teoría de Autómatas y Lenguajes Formales*, McGraw Hill, 2007.

[5] Susan H. Rodger and Thomas W. Finley. *JFLAP: An Interactive Formal Languages and Automata Package.* Jones & Bartlett Publishers, 2006.

[6] D. H. Younger. Recognition and parsing of context-free languages in time n^3. *Information and Control*, 10(2):189-208, 1967.

[7] T. Kasami. An eficient recognition and syntax algorithm for context-free languages. *Scientific Report* AFCRL-65-758, Air Force Cambridge Research Lab., Bedford, Massachussetts, 1965.

[8] Janusz A. Brzozowski. Derivatives of Regular Expressions, in *Journal of the ACM*, 11(4): 481-494, 1964.

[9] Charles Donnelly and Richard Stallman. *Bison, The yacc-compatible parser generator.* Free Software Foundation, 2006.

[10] Alfred V. Aho, Ravi Sethi y Jeffrey D. Ullman. *Compiladores: Principios, técnicas y herramientas*. Addison Wesley, 1990.

[11] V. M. Glushkov. "The abstract theory of automata", Russian Mathematical Surveys, 16:1-53, 1961.

[12] R. McNaughton y H. Yamada. "Regular expressions and state graphs for automata", IEEE *Transactions on electronic computers* 9(1):39-47, 1960.

[13] K. Thompson. Regular expression search algorithms, *Communication of ACM* 11(6): 419-422, 1968.

DES: un recurso para el aprendizaje de bases de datos deductivas

Fernando Sáenz Pérez
Dept. Ingeniería del Software e Inteligencia Artificial
Universidad Complutense de Madrid
Facultad de Informática, c/ Profesor José García Santesmases, s/n, 28040 Madrid
fernan@sip.ucm.es

Resumen

En este artículo se presenta DES (Datalog Educational System), un sistema con el que se pueden transmitir los conceptos fundamentales de las bases de datos deductivas con Datalog como lenguaje de acceso a datos. Además, dado que está implementado en Prolog, los estudiantes que conozcan este lenguaje pueden estudiar su implementación en cursos más avanzados. El motivo de su desarrollo pareció evidente en su momento al no existir un sistema de uso fácil y multiplataforma. El sistema incorpora recursión no lineal (a diferencia de SQL) y negación estratificada. Se puede usar tanto desde un intérprete Prolog (por lo tanto, sobre cualquier plataforma hardware/software que admita tal intérprete) como desde ejecutables para Windows, Linux y Unix. El sistema se distribuye desde Sourceforge bajo licencia GPL, y su código es abierto y gratuito. Ha suscitado interés en la comunidad docente e investigadora internacional desde la publicación de su primera versión en el año 2003, y se publican nuevas versiones anualmente.

1. Introducción

La intersección de las bases de datos, la lógica y la inteligencia artificial ha dado lugar a las bases de datos deductivas. Una base de datos deductiva es un sistema que incluye mecanismos para definir reglas (deductivas) que pueden inferir o deducir información adicional de los hechos almacenados en la base de datos. Las reglas se especifican en un lenguaje declarativo y un motor de inferencia (el mecanismo de deducción) es capaz de deducir nuevos hechos de la base de datos. El modelo usado para las bases de datos deductivas está estrechamente relacionado con el modelo de datos relacional y en particular con el formalismo de cálculo relacional de dominios. Está también relacionado con el lenguaje Prolog y, principalmente, con el lenguaje Datalog, un subconjunto de Prolog que evita el uso de estructuras potencialmente infinitas.

El álgebra relacional se ha mostrado insuficiente en la expresión de consultas a bases de datos. Un defecto importante es la ausencia de recursión, que impide expresar consultas sencillas como es el cierre transitivo de un grafo. Este tipo de consultas pueden expresarse en la lógica de predicados de primer orden. No obstante, a partir del estándar SQL:1999 para SQL (el lenguaje *de facto* de bases de datos relacionales) se incluyó una forma limitada de recursión: la recursión lineal.

El área de las bases de datos deductivas y en particular el procesamiento recursivo de consultas se reveló muy activo en 1984 con el inicio de tres proyectos importantes, dos en Estados Unidos y otro en Europa. Los proyectos Nail!, en Stanford, LDL, en Austin y EKS de ECRC han producido contribuciones importantes y la construcción de prototipos simples. En la Universidad de Wisconsin se desarrolló Coral [11] a finales de los 80 (véase también el apartado 2).

En este artículo se presenta un sistema que se puede usar como recurso docente para la enseñanza de las bases de datos deductivas que disponen de Datalog como lenguaje de consulta. Estas enseñanzas están orientadas a las asignaturas de planes de estudios relativas a bases de datos avanzadas e ingeniería del conocimiento. En el siguiente apartado se comentan los sistemas relacionados. En el apartado 3 se listan sus características y limitaciones. El sistema se describe en el apartado 4 y se plantean algunos ejemplos de uso en la enseñanza en el 5. En el apartado 6 se comenta su aplicabilidad y utilidad en el contexto internacional y, finalmente, el apartado 7 recoge las conclusiones y posible trabajo futuro.

2. Sistemas relacionados

La comunidad científica ha dedicado un enorme esfuerzo en la investigación de bases de datos deductivas [13] (suscitando muchos talleres y conferencias) y ha dado lugar a varios sistemas. Sin embargo, ninguno de ellos proporciona un entorno fácil, amigable y multiplataforma cuyo objeto sea la introducción de este tema de estudio a los alumnos. No obstante, se pueden comentar algunos de los sistemas de bases de datos deductivas más representativos.

El sistema de bases de datos deductivas LOLA [23] está basado en un lenguaje de cláusulas declarativas que da soporte a la recursión, los términos compuestos, negación, agregación y predicados predefinidos. El motor deductivo está basado en el paradigma de evaluación ascendente con operaciones relacionales. Está disponible mediante una interfaz para clientes Web.

El proyecto LDL en MCC originó el prototipo LDL++ [21], un sistema de bases de datos deductivas con características como negación tanto estratificada como no estratificada, términos de tipo conjunto y agregados. Actualmente se puede usar via Internet con un cliente con capacidades Java.

XSB [12] (http://xsb.sourceforge.net/) es un sistema Prolog extendido que se puede usar para aplicaciones de bases de datos deductivas. Se caracteriza por implementar una semántica bien fundada para reglas con literales negativos en su cuerpo e implementa mecanismos de tablas de extensión. Funciona sobre los sistemas operativos Unix/Linux y Windows. Datalog++ [16] es una fachada para el sistema de bases de datos deductivas XSB.

Coral [11] es un sistema deductivo con un lenguaje de consulta declarativo que da soporte a cláusulas generales de Horn aumentadas con términos compuestos, agrupación por conjuntos, agregación, negación y relaciones con tuplas que contienen variables cuantificadas universalmente. Sólo funciona sobre plataformas Unix. Existe también una versión que permite características de la programación orientada a objetos: Coral++ [15].

FLORID (F-LOgic Reasoning In Databases) [8] es un sistema de bases de datos deductivas orientadas a objetos que emplea F-Logic como lenguaje de consulta y de definición de datos. Con el interés creciente por los datos semiestructurados, FLORID se ha extendido para tratar este tipo de datos en el contexto de la integración de datos en la Web semántica.

El proyecto NAIL! proporcionó un prototipo con negación estratificada, negación bien fundada y negación estratificada modularmente. Posteriormente se añadió el lenguaje Glue, que consiste esencialmente en reglas lógicas con instrucciones SQL incorporadas en un lenguaje imperativo convencional [10,7]. El enfoque de la combinación resulta similar a Coral, que usa C++. No funciona sobre plataformas Windows.

Otro sistema de bases de datos deductivas que sigue esta combinación de lenguajes declarativos e imperativos es Rock&Roll [2].

ADITI 2 [19] es la versión actual de un sistema de bases de datos deductivas que usa el lenguaje de programación lógico-funcional Mercury. No funciona sobre plataformas Windows y se ha detenido su desarrollo.

El único sistema de bases de datos deductivas comercial ha sido SDS (Smart Data System) y su lenguaje de consulta declarativo DECLARE (Declarative Reasoning) [9], con soporte para la negación estratificada y para los conjuntos.

No obstante, el mayor impacto de las bases de datos deductivas en el ámbito comercial quizás haya sido la incorporación de la recursión lineal en la versión SQL del sistema relacional DB2 de IBM con técnicas de implementación basadas en los conjuntos mágicos [3].

3. Características del sistema DES

Como ventajas del sistema se pueden enumerar:

- Sistema multiplataforma. Se ejecuta en la mayoría de los sistemas operativos actuales, bien sea a partir de su código fuente Prolog o mediante los ejecutables proporcionados.
- Recursión no lineal. Admite múltiples llamadas recursivas en el cuerpo de una regla, incluyendo llamadas recursivas con negación (aunque la negación en sí no puede aparecer en un ciclo; véase el siguiente punto).
- Negación estratificada [18]. La negación estratificada significa *grosso modo* que un objetivo negado no puede llamarse a sí mismo en ningún camino de cómputo con objeto de asegurar la corrección de la respuesta y así evitar

situaciones como la conocida paradoja de Russell. La negación puede aparecer en uno o varios literales negados del cuerpo de las reglas. El sistema implementa un algoritmo ascendente guiado por el objetivo y con cómputos descendentes que calcula de forma segura las respuestas.

- Vistas transitorias. Permiten emitir consultas conjuntivas al vuelo, es decir, desde el inductor de comandos y sin necesidad de crear una relación *ex profeso* para ello (véase el apartado 4.4).
- Implementación con técnicas de tablas de extensión [6,17] que mejoran significativamente el rendimiento del sistema con respecto a técnicas ascendentes ingenuas.

Algunas de las limitaciones del sistema que se pueden enumerar incluyen:

- Agregados. El sistema no dispone de funciones de agregación, que se demuestran muy útiles en la práctica. Su implementación no es sencilla, aunque está prevista.
- Aritmética. Aún no se ha implementado la posibilidad de incluir primitivas aritméticas, aunque es una labor sencilla e incluida en el trabajo futuro.
- La interfaz de usuario del sistema es un intérprete de comandos que acepta tanto consultas Datalog como comandos del sistema. Sin embargo, se debería ampliar esta interfaz a una interfaz gráfica de usuario (véase también el apartado de conclusiones).
- Depuración. Actualmente no hay herramientas para la depuración de programas Datalog, pero se están investigando actualmente [4].
- Incapacidad de representar términos compuestos. Los términos compuestos pueden introducir problemas de no terminación. La terminación es una característica que se exige *de facto* a los sistemas de bases de datos, y se podría asegurar limitando la profundidad de los términos a una cota finita. Una correspondencia entre constantes y un conjunto finito de términos construidos podría dar lugar a una implementación sencilla.

4. Descripción del sistema DES

Dado que Datalog se derivó de Prolog, se han adoptado prácticamente todos los convenios sintácticos de Prolog para la escritura de programas Datalog. Sin embargo, los comandos son diferentes a los que están acostumbrados los programadores de Prolog (véase el apartado 4.2).

4.1. Instalación e inicio

El sistema educativo DES se encuentra en su página de distribución (http://des.sourceforge.net). El sistema se puede usar a partir de su código fuente Prolog sobre cualquiera de los siguientes sistemas Prolog: GNU Prolog [5], Ciao Prolog [1], Sicstus Prolog [14] y SWI Prolog [20]. No obstante, el método recomendado es instalar uno de los ejecutables para Windows, Linux o Unix. En esta página de distribución se incluye el manual de usuario con instrucciones más detalladas sobre la instalación del sistema. No obstante, elegir el método recomendado supone, para la instalación, descomprimir simplemente el archivo de distribución y, para su inicio, ejecutar un fichero.

4.2. Uso del sistema

Al iniciar el sistema se muestra lo ilustrado en la figura 1. La última línea (**DES>**) es el inductor de Datalog, en el que se pueden escribir comandos o consultas Datalog. Si el sistema se ejecuta desde un intérprete Prolog y en algún momento se sale del sistema por algún error se puede escribir **des.** en el inductor de Prolog para continuar.

```
*********************************************************
*                                                       *
* DES: Datalog Educational System v.1.2.0               *
*                                                       *
* - Stratified Negation                                 *
* - Full recursion                                      *
* - Noncompound terms                                   *
*                                                       *
* Type "/help" for commands                             *
* Type "des." if you get out of DES                     *
*   from a Prolog interpreter                           *
*                                                       *
*                          Fernando Sáenz (c) 2004-2007 *
*                                                DISIA UCM *
*            Please send comments, questions, etc. to: *
*                                   fernan@sip.ucm.es *
*                                Visit the Web site at: *
*                           http://des.sourceforge.net/ *
*********************************************************

DES>
```

Figura 1. El sistema DES

El modo habitual de uso del sistema es crear mediante cualquier editor de texto en el sistema operativo un fichero con extensión **.dl** para guardar el programa Datalog que se desarrolle. Una vez se haya escrito un programa Datalog se puede cargar con el comando:

/consult *NombreArchivo*

Donde ***NombreArchivo*** es el nombre de archivo, como **relop.dl** (la extensión predeterminada **.dl** se puede omitir) que en este caso es un programa de ejemplo de la distribución y que muestra la versión Datalog de las operaciones relacionales básicas. Obsérvese que los comandos comienzan por una barra de división y la especificación de argumentos es similar a los intérpretes de comandos de un sistema operativo (con espacios separadores en lugar de paréntesis y comas).

Este archivo de la distribución alojado en la carpeta de ejemplos contiene un programa que reproduce las operaciones relacionales básicas mediante reglas Datalog. Contiene tres relaciones (**a**, **b** y **c**) que se usan como argumentos de las operaciones relacionales.

```
% Relations
a(a1).      b(b1).      c(a1,b2).
a(a2).      b(b2).      c(a1,a1).
a(a3).      b(a1).      c(a2,b2).

% Relational Operations

% pi(X)(c(X,Y))
projection(X) :- c(X,Y).

% sigma(X=a2)(a)
selection(X) :- a(X), X=a2.

% a X b
cartesian(X,Y) :- a(X), b(Y).

% a |x| c
join(X) :- a(X), c(X,Y).

% a U b
union(X) :- a(X).
union(X) :- b(X).

% a - b
difference(X) :- a(X), not(b(X)).
```

Los programas Datalog pueden incluir comentarios (indicados por el símbolo **%** y se extienden hasta el final de la línea).

Si este archivo está en el directorio de inicio del sistema, se podría escribir:

/consult relop

Una vez cargado el programa se puede emitir una consulta Datalog, como:

```
DES> projection(X)
```

que da como resultado:

```
{
  projection(a1),
  projection(a2)
}
```

El resultado de la evaluación de una consulta es el conjunto de hechos derivado de la consulta en el contexto del programa de forma tanto intensional (con las reglas que definen nuevas relaciones a partir de otras relaciones) como extensional (con las reglas especificadas como hechos). En este ejemplo, **projection(X)** se corresponde con la operación relacional proyección sobre el primer argumento de la relación **c**.

Para que el alumno comprenda el diferente modo de evaluar y entender las consultas Datalog con respecto a Prolog, se puede usar el comando:

```
DES> /prolog projection(X)
projection(a1)
? (type ; for more solutions, <Intro>
to continue) ;
projection(a1)
? (type ; for more solutions, <Intro>
to continue) ;
projection(a2)
? (type ; for more solutions, <Intro>
to continue) ;
no
```

Obsérvese la diferencia en la ejecución Datalog y la ejecución Prolog. El primero aporta el significado de la relación como un conjunto y el segundo busca soluciones, una a una, que satisfagan un objetivo; por ello se obtienen tres respuestas (con **projection(a1)** repetido debido a la definición de **c**).

Dado que DES está implementado con técnicas de tablas de extensión, se puede consultar el significado de las relaciones calculadas con el comando **/list_et**, que muestra todas las que se han calculado hasta el momento (incluyendo consultas anteriores). Además, muestra las últimas llamadas que se han producido en la última iteración de punto fijo, que puede ser útil si se desea estudiar su implementación:

```
Answers:
{
  projection(a1),
  projection(a2),
  c(a1,a1),
```

```
  c(a1,b2),
  c(a2,b2)
}
Calls:
{
  projection(_10771),
  c(_10766,_10767)
}
```

A menos que se emita una vista transitoria (véase el apartado 4.4) o el comando **/clear_et** (que borra el contenido de la tabla de extensión), se conservarán las respuestas calculadas.

4.3. Consultas

Una consulta es el nombre de una relación del programa con tantos argumentos como la aridad de la relación y que se escribe en el inductor DES. Cada uno de ellos puede ser una variable o una constante, pero no un término compuesto. Obsérvese que no se pueden escribir consultas conjuntivas, como **a(X), b(X)** para calcular la intersección de ambas relaciones. Sin embargo, esta situación se puede abordar desde dos puntos de vista: (1) escribir en el programa una regla **r(X) :- a(X), b(X)** (o insertarla con el programa cargado mediante el comando **/assert**) y emitir la consulta **r(X)**, y (2) usar vistas transitorias (véase el apartado 4.4).

La respuesta a una consulta con variables para todos sus argumentos resulta en el conjunto completo de hechos que definen su significado. Si una consulta contiene una constante en uno o más de sus argumentos, en el resultado se descartan los hechos que definen su significado pero que no casan con el patrón de la consulta. Ésta es una operación análoga a la selección del álgebra relacional, como, por ejemplo, **a(a3)** en el ejemplo anterior.

4.4. Vistas transitorias

En este sistema se ha introducido un concepto nuevo en los sistemas de consulta a bases de datos deductivas, las vistas transitorias, que permiten escribir consultas conjuntivas al vuelo. Una vista transitoria es una regla que se añade a la base de datos y su cabeza (parte izquierda) se considera como una consulta que se ejecuta. Después de su evaluación, la regla se elimina de la base de datos. Las vistas transitorias son útiles para emitir rápidamente consultas conjuntivas, como, por ejemplo, en:

```
DES> d(X) :- a(X), not(b(X))
```

que calcula la diferencia de conjuntos entre las relaciones **a** y **b**.

Obsérvese que la vista se calcula en el contexto del programa, por lo que si se tienen definidas más reglas para la misma relación de consulta (expresada en la cabeza de la vista), su evaluación devolverá el significado completo de esta relación considerando el resto de reglas del programa. Por ejemplo:

```
DES> a(X) :- b(X)
```

calcula la unión de conjuntos de las relaciones **a** y **b** en el contexto del programa **relop.dl**, dando como resultado:

```
{
  a(a1),
  a(a2),
  a(a3),
  a(b1),
  a(b2)
}
```

4.5. Reglas

Las reglas Datalog son similares a las cláusulas Prolog con las mismas limitaciones indicadas para las consultas. Estas reglas tienen la forma **cabeza :- cuerpo**, o simplemente **cabeza**, terminando ambas en un punto. Una cabeza es un término compuesto, y un cuerpo es una secuencia de objetivos separados por comas que pueden contener operadores predefinidos. Los objetivos pueden aparecer negados como **not(objetivo)**.

4.6. Programas

Los programas consisten en un conjunto de reglas en el que su orden no es importante. Pueden contener comentarios (**%**).

4.7. Detalles de la implementación

A diferencia de muchas implementaciones de Datalog, que usan una evaluación ascendente (*bottom-up*), este intérprete usa una evaluación ascendente guiada por una descendente (*top-down*). En lugar de calcular el significado del programa completo, calcula el significado de la consulta emitida usando tablas de extensión [6] como técnica de optimización.

5. Ejemplos

En este apartado se muestran algunos ejemplos ilustrativos del uso de DES en la enseñanza de Datalog y que ilustran algunas de sus características más relevantes (entre paréntesis se indica el nombre del fichero que se pueden encontrar en la carpeta **examples** de la distribución).

5.1. Caminos de un grafo (fichero paths.dl)

Este programa introduce el uso de la recursión en DES definiendo el grafo de la figura 2 y el conjunto de tuplas <origen, destino> tal que existe un camino desde el origen hasta el destino

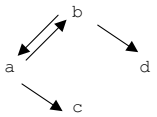


Figura 2. Caminos de un grafo

```
% Paths in a Graph
edge(a,b).      edge(a,c).
edge(b,a).      edge(b,d).

path(X,Y) :- path(X,Z), edge(Z,Y).
path(X,Y) :- edge(X,Y).
```

La consulta path(X,Y) devuelve:

```
{
  path(a,a),
  path(a,b),
  path(a,c),
  path(a,d),
  path(b,a),
  path(b,b),
  path(b,c),
  path(b,d)
}
```

que se interpreta como el cierre transitivo de la relación **path**.

5.2. Negación estratificada (fichero negation.dl)

DES asegura que la información negativa obtenida a partir de un programa con objetivos negados es correcta siempre que el programa admita una estratificación. El siguiente programa lo ilustra:

```
a :- not(b).
b :- c,d.
c :- b.
c.
```

La consulta **a** tiene éxito con significado **{a}**. Obsérvese que el significado de la consulta **not(a)** es el conjunto vacío.

Si se estudia la resolución de consultas Datalog en presencia de negación, los siguientes comandos resultan de utilidad:

```
DES> /pdg
Nodes: [d/0,a/0,b/0,c/0]
Arcs : [a/0-b/0,c/0+b/0,
        b/0+d/0,b/0+c/0]

DES> /strata
[(d/0,1),(a/0,2),(b/0,1),(c/0,1)]
```

El primer comando muestra el grafo de dependencias de predicados (relaciones) (véase, e.g., [22]) para el programa cargado. En primer lugar se listan los nodos en una lista cuyos elementos R son predicados con sus aridades de la forma relación/aridad. A continuación se muestran los arcos como una lista cuyos elementos son P+Q o P-Q, donde P y Q son nodos del grafo. Un arco P+Q significa que existe una regla tal que P es la relación para su cabeza y Q es la relación de uno de sus literales. Si el literal está negado, el arco es negativo, lo cual se expresa por P-Q. El grafo de este programa se ilustra en la figura 3.

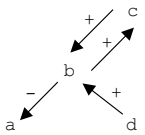


Figura 3. Grafo de dependencias de predicados (relaciones) para **negation.dl**

El segundo comando muestra el estrato asignado a cada relación. Los estratos se muestran como una lista de pares (R,S), donde R es una relación y S es su estrato. En este ejemplo todas las relaciones se encuentran en el estrato 1 salvo **a**, que se encuentra en el estrato 2. Esto significa que si se calcula el significado de **a**, es necesario tener calculado completamente el o los estratos inferiores (en este caso sólo el estrato 1) y sólo para las relaciones que intervengan para calcular el significado de **a**.

Para ilustrar que sólo se calculan las relaciones necesarias, considérese la consulta **b** para el mismo programa. DES calcula el subgrafo de

dependencias para **b**, i.e., el grafo en el que aparecen todas las relaciones alcanzables a partir de **b**; después se calcula de nuevo una estratificación. Obsérvese que el sistema informa de los estratos calculados (siempre que la salida informativa esté activada, que es el comportamiento predeterminado y que se puede modificar con los comandos **/verbose** y **/noverbose**):

```
DES> a
Info: Computing by stratum of [b].
{
  a
}
DES> b
{
}
```

Para la consulta **a**, el sistema informa de que **b** se calcula previamente (usando la técnica de optimización basada en tablas de extensión), mientras que no es necesario para la consulta **b** recurrir al algoritmo de cómputo por estratos.

5.3. Operadores

Todos los operadores son infijos salvo la negación. Los infijos se deben entender como tests en lugar de relaciones para asegurar finitud en la respuesta. Esto significa que la lectura declarativa de las reglas con estos operadores no es clara. Por ejemplo, una vez cargado el programa **relop.dl** y al añadir la siguiente regla (con el comando **/assert**):

```
less(X,Y) :- X<Y, c(X,Y).
```

la evaluación de la consulta **less(X,Y)** devuelve un significado vacío, pero si se permuta el orden de los objetivos en la regla, se obtiene:

```
{
  less(a1, b2),
  less(a2, b2)
}
```

Por lo tanto, no se aseguran respuestas correctas para los programas que usen primitivas sin sus argumentos cerrados (i.e., sin variables) dado que en general el orden de los objetivos que incluyan estos operadores afecta a la semántica. Es una consecuencia de las reglas inseguras [22], las cuales se pueden caracterizar sintácticamente aunque la versión actual del sistema no advierte de su presencia.

6. Utilidad del sistema

Según lo que conocemos, DES ha sido reconocido de utilidad en la comunidad internacional y aplicado en diferentes cursos de bases de datos y sistemas basados en conocimiento (EE.UU. y Europa). También se ha evaluado en un proyecto europeo y aplicado en el ámbito de la investigación. Está listado como un producto de interés por el grupo SIGMOD (Special Interest Group on Management of Data) de la sociedad ACM. Se ha demostrado como un sistema muy demandado e incluido en la lista de los diez sistemas más activos de Prolog en sourceforge.net, el sitio de referencia para la distribución de aplicaciones de código abierto, donde se han contabilizado en el último año más de 180 descargas mensuales de nuestro sistema.

7. Conclusiones

Nuestra experiencia al usar este recurso en la asignatura "Bases de datos y sistemas de información", junto con la realimentación por parte de usuarios (alumnos, profesores, investigadores y profesionales) de la comunidad internacional, ha permitido ajustarlo y mejorarlo para su aplicación en el entorno docente. Los resultados han sido muy satisfactorios (corroborado por colegas internacionales). No obstante, aún quedan aspectos importantes y pendientes de su implementación, tales como la integración de aritmética, los agregados, un entorno gráfico de desarrollo y herramientas de depuración. Markus Triska, moderador del grupo comp.lang.prolog, ha desarrollado un entorno Emacs para DES. Actualmente estamos desarrollando un entorno de desarrollo configurable (para DES en particular) denominado ACIDE (véase sourceforge). Finalmente, investigamos la depuración declarativa de programas Datalog [4].

Agradecimientos

El autor desea agradecer al grupo Clip por proporcionar su sistema Ciao Prolog gratuito y en particular a F. Bueno por su ayuda en la adaptación de DES. También a Jan Wielemaker y Daniel Diaz por proporcionar sus sistemas Prolog gratuitos, así como a Mats Carlsson por resolver rápidamente los aspectos de implementación en Sicstus, y a todos ellos por permitir usar sus listas de distribución como medio de difusión del sistema. Final-

mente agradecer la realimentación de todos los usuarios de DES y a los revisores de este trabajo.

Referencias

[1] F. Bueno, D. Cabeza, M. Carro, M. Hermenegildo, P. López–García, and G. Puebla. “The Ciao Prolog system. Reference manual”, School of Computer Science, Technical University of Madrid (UPM), 1997.

[2] M.L. Barja, N.W. Paton, A. Fernandes, M.H. Williams, A. Dinn, “An Effective Deductive Object–Oriented Database Through Language Integration”, In Proc. of the 20th VLDB Conference, 1994.

[3] C. Beeri and R. Ramakrishnan, "On the Power of Magic", Journal of Logic Programming, 10(3,4):255-299, 1991.

[4] R. Caballero, Y. García-Ruiz, and F. Sáenz-Pérez, “Towards a Set Oriented Calculus for Logic Programming”, PROLE 2006, pp. 41-50, Barcelona, 2006.

[5] D. Diaz, GNU Prolog, http://www.gnu.org/software/prolog.

[6] S.W. Dietrich, “Extension Tables: Memo Relations in Logic Programming”, IV IEEE Symposium on Logic Programming, 1987.

[7] M. Derr, S. Morishita, and G. Phipps, “Design and Implementation of the Glue–NAIL Database System”, In Proc. of the ACM SIGMOD International Conference on Management of Data, pp. 147–167, 1993.

[8] M. Kifer, G. Lausen, J. Wu, "Logical Foundations of Object Oriented and Frame Based Languages", Journal of the ACM, vol. 42, p. 741-843, 1995.

[9] W. Kiessling, H. Schmidt, W. Strauss, and G. Dünzinger, “DECLARE and SDS: Early Efforts to Commercialize Deductive Database Technology”, VLDB Journal, 3, 1994.

[10] G. Phipps, M. A. Derr, and K.A. Ross, “Glue–NAIL!: A Deductive Database System”. In Proc. of the ACM SIGMOD Conf. on Management of Data, pp. 308–317, 1991.

[11] R. Ramakrishnan, D. Srivastava, S. Sudarshan, and P. Seshadri, “Implementation of the Coral Database System”, In Proc. of the ACM SIGMOD Conference on Management of Data, 1993.

[12] P. Rao, Konstantinos F. Sagonas, Terrance Swift, David Scott Warren, and Juliana Freire, “XSB: A System for Efficiently Computing WFS", Logic Programming and Non–monotonic Reasoning, 1997.

[13] R. Ramakrishnan and J.D Ullman, “A Survey of Research on Deductive Database Systems”, Journal of Logic Programming, 23(2): 125–149, 1995.

[14] SICS, Sicstus Prolog, http://www.sics.se/sicstus.

[15] D. Srivastava, R. Ramakrishnan, S. Sudarshan, and P. Seshadri, “Coral++: Adding Object–Orientation to a Logic Database Language”, Proceedings of the International Conference on Very Large Databases, 1993.

[16] Z. Tang, "Datalog++: An Object-Oriented Front-End For The Xsb Deductive Database Management System", http://citeseer.ist.psu.edu/tang99datalog.html.

[17] H. Tamaki and T. Sato, “OLD Resolution with Tabulation”, Proceedings of ICLP’86, Lecture Notes on Computer Science 225, Springer–Verlag, 1986.

[18] J.D. Ullman. Database and Knowledge-Base Systems, Computer Science Press, 1995.

[19] J. Vaghani, K. Ramamohanarao, D.B. Kemp, Z. Somogyi, and P.J. Stuckey, “Design Overview of the Aditi Deductive Database System”, In Proc. of the 7th Intl. Conf. on Data Engineering, pp. 240–247, 1991.

[20] J. Wielemaker, SWI Prolog http://www.swi-prolog.org.

[21] C. Zaniolo, N. Arni, and K. Ong, “Negation and Aggregates in Recursive Rules: the LDL++ Approach”, In Proc. Intl. Conf. on Deductive and Object Oriented DBs, 1993.

[22] C. Zaniolo, S. Ceri, C. Faloutsos, T.T. Snodgrass, V.S. Subrahmanian, and R. Zicari, "Advanced Database Systems", Morgan Kauffmann Publishers, 1997.

[23] U. Zukowski and B. Freitag, “The Deductive Database System LOLA”, In: J. Dix and U. Furbach and A. Nerode (Eds.). Logic Programming and Nonmonotonic Reasoning. LNAI 1265, pp. 375–386. Springer, 1997.

Generador-corrector de problemas de la Unidad de Control

Antonio J. de Vicente[1], Manuel Prieto[2], Rafael Fernández
Dpto. de Automática
Universidad de Alcalá
Campus Universitario. 28871 Alcalá de Henares (Madrid)
[1]avicente@aut.uah.es, [2]mpm@aut.uah.es

Resumen

Este artículo describe un recurso docente orientado a la imparticion del tema de la Unidad de Control de las asignaturas de Arquitectura de Computadores de I. T. Informática de Gestión y la de Arquitectura de Computadores I de I. T. Informática de Sistemas.

El recurso permite generar diferentes computadores elementales, parametrizados por el usuario, y mostrar como serían las fases de ejecución de una instrucción máquina, para distintas instrucciones con diversos modos de direccionamiento. El usuario puede emplear el recurso como un tutorial, o como un generador de problemas para que resuelva, siendo corregidos automáticamente por la aplicación.

1. Introducción

En este artículo, presentamos la aplicación PREVIUC (**PR**esentación **VI**sual de la **U**nidad de **C**ontrol) orientada al aprendizaje de la unidad de control de un computador. Con ella, el estudiante puede resolver diferentes problemas propuestos por la aplicación, o simplemente emplearla como un tutorial.

La estructura del artículo es la siguiente:

- En el apartado 2 se resume el estado del arte de otros recursos docentes similares.
- El apartado 3 expone la motivación que llevó al desarrollo de la aplicación.
- En el apartado 4 se describen los contenidos del tema de la Unidad de Control de la asignatura de Arquitectura de Computadores dentro del plan de estudios de I. T. Informática de Gestión / Sistemas.
- En el apartado 5 se presentan las diferentes funcionalidades de la aplicación y cómo es su funcionamiento
- El apartado 6 recoge las sugerencias docentes para que el recurso sea empleado.
- En el apartado 7 arroja las conclusiones del desarrollo y empleo del recurso.

2. Estado del arte

El recurso docente que se propone no es nuevo, en la práctica existen una serie de recursos docentes que trabajan el tema de la Unidad de Control.

Entre los recursos que existen se encuentran:

- Spim [1] que permite ver como evoluciona el contenido de la memoria y el de los registros pero que no muestra gráficamente el camino de datos durante la ejecución de la instrucción.
- DLX [2] y su extensión para Windows WinDLX, que si permiten ver de manera gráfica la ejecución de instrucciones sobre el camino de datos.
- SimuProc [3], es un simulador de un Procesador Hipotético con el cual se pueden aprender las nociones básicas para empezar a programar en lenguaje ensamblador. Muestra todo el proceso interno de ejecución del programa a través de cada ciclo del procesador.
- Simula3MS [4], es un simulador que muestra de manera gráfica la evolución de todos los componentes ciclo a ciclo (incluso paso a paso en el caso del procesador multiciclo) y tener también la opción de ejecutar conjuntamente todas las instrucciones y observar únicamente el efecto que produce la ejecución completa.
- MSX88 [5], representa de manera gráfica la ejecución de instrucciones en el Intel 8088. Simula un subconjunto del repertorio de instrucciones de dicho procesador.

3. Motivación

Desde hace varios años, para completar la impartición del tema de la Unidad de Control, hemos sugerido a los alumnos el empleo del programa MSX88. No obstante, su funcionalidad

se quedaba corta, principalmente por la interacción con el usuario, en la que el estudiante era un receptor pasivo del comportamiento de un i8088 simplificado.

Se observó, también, que los alumnos tenían problemas a la hora de desarrollar los cronogramas de ejecución de las instrucciones que se les proponían, y además, el número de ejercicios corregidos les resultaban escasos.

Es por ello, por lo que se desarrolló la aplicación Previuc, en la que el alumno podría parametrizar el computador elemental, entre varias opciones, y ejercitarse en el cálculo de los cronogramas de ejecución para diferentes instrucciones con distintos modos de direccionamiento; sabiendo, que la aplicación le indicaría los errores que pudiera cometer.

4. Descripción de las asignaturas

La asignatura de Arquitectura de Computadores, de la titulación de I. T. Informática de Gestión y de la de Arquitectura de Computadores I de I. T. Informática de Ssstemas de la Universidad de Alcalá, se recogen en el B.O.E. del 24 de mayo de 1995 [6].

A partir del descriptor de la asignatura, el tema de la unidad de control ocupa 8 horas de clase, y en él se ven los siguientes conceptos:

1. Operaciones elementales.
2. Estructura de un computador elemental.
3. Señales de control.
4. Temporización de las señales de control.
5. Ejecución de instrucciones.
6. Diseño de la unidad de control.
7. Unidad de control cableada.
8. Unidad de control microprogramada.
9. Microprogramación y nanoprogramación.

La aplicación Previuc abarca los puntos 1 a 5.

5. Previuc

La aplicación Previuc presenta de manera visual la ejecución de instrucciones elegidas por el alumno, en un computador elemental, también diseñado por el alumno. Además, permite al estudiante resolver los distintos ejercicios de manera interactiva, indicándole la aplicación si ha cometido algún error o si es correcto.

5.1. El asistente para la creación de máquinas

La aplicación cuenta con un asistente para la creación de diferentes computadores elementales con el que el usuario puede decidir que elementos quiere y cuales no en el diseño final de su arquitectura.

La aplicación comienza con dicho asistente, tal y como se muestra en la figura 1.

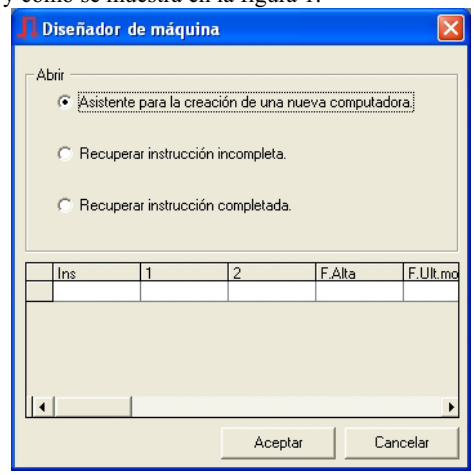


Figura 1. Asistente para la creación del computador.

Cuando se desea crear una máquina nueva sobre la que ejecutar instrucciones, se nos presentan diferentes opciones, recogidas en las figuras siguientes, desde la elección de si el bus es multiplexado para datos y direcciones, o se cuenta con un bus diferente para estos (figura 2)

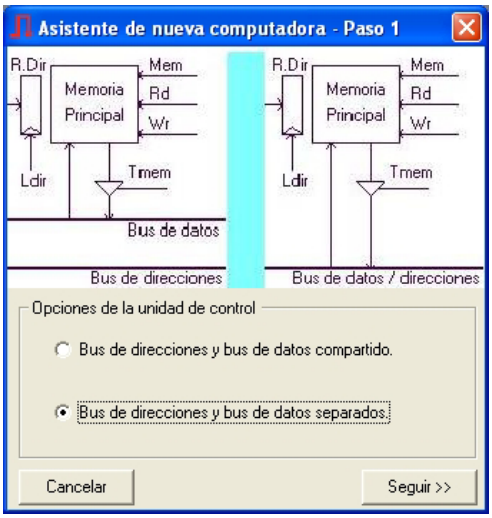


Figura 2. Elección del tipo de bus.

Posteriormente se permite al usuario la elección de si el banco de registros contará con un puerto de entrada y uno de salida o con uno de entrada y dos de salida (figura 3)

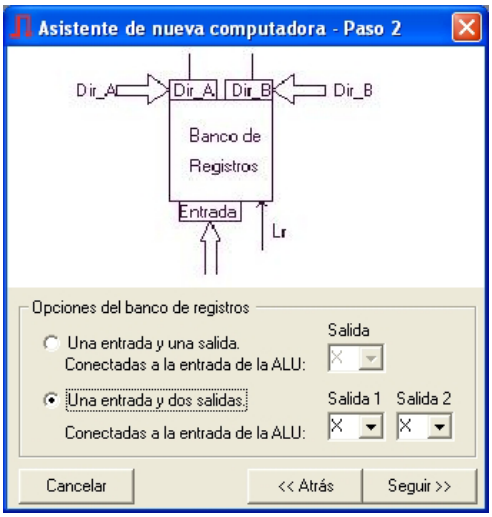


Figura 3. Elección del banco de registros.

El siguiente paso es decidir si la unidad de direccionamiento contará con la posibilidad de autoincremento o se deberá pasar por la ALU para actualizar el valor del contador de programa. Dicha elección se muestra en la figura 4.

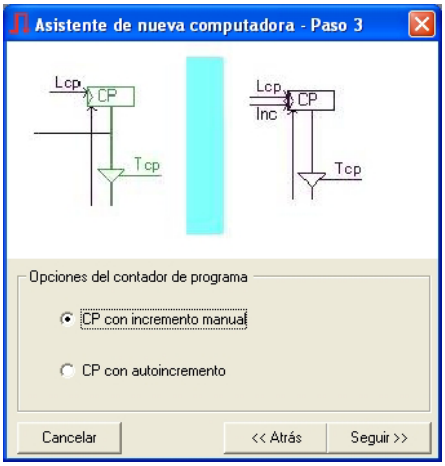


Figura 4. Elección de la unidad de direccionamiento.

Finalmente, se presentan una serie de opciones adicionales para ver si los buses se comunican, o a qué entrada de la ALU se puede conectar el contador de programa (figura 5)

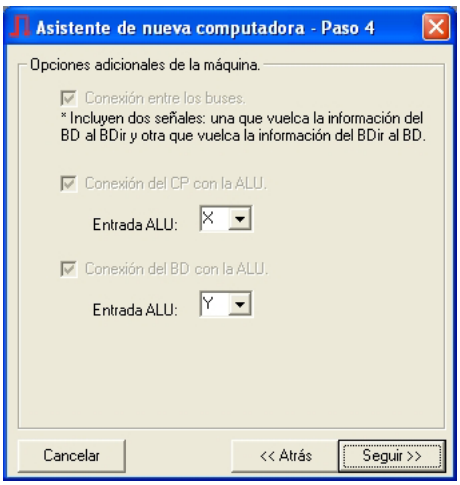


Figura 5. Opciones adicionales.

Si en los 4 pasos anteriores, el usuario ha creado una maquina incoherente, la aplicación generará el computador sin errores que imposibiliten la ejecución de instrucciones

.

Un ejemplo del computador elemental generado con las opciones anteriores se muestra en la figura siguiente

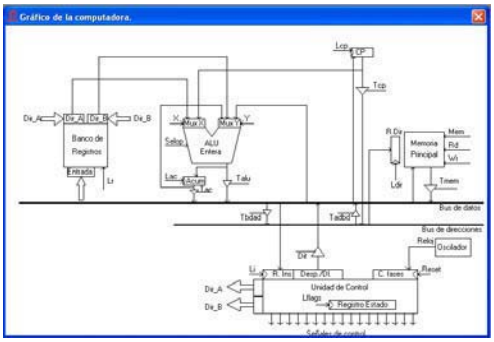


Figura 6. Computador elemental generado.

Una vez generada la máquina elemental, el programa pregunta al usuario por la instrucción que desea ejecutar (figura 7)

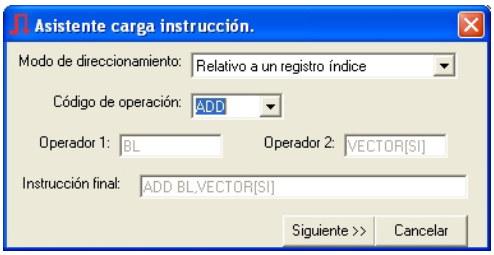


Figura 7. Solicitud de la instrucción a ejecutar.

Con la máquina ya terminada y con la instrucción y el modo de direccionamiento elegidos por el usuario, ya permitiría al usuario resolver el cronograma de ejecución (figura 8)

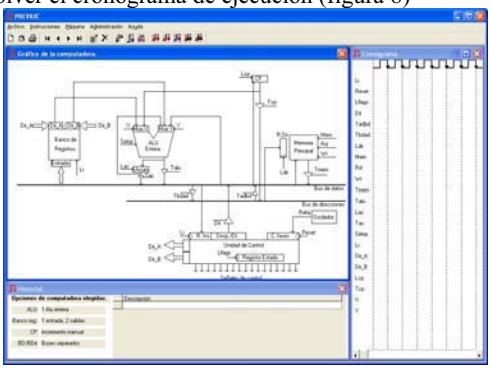


Figura 8. Resolución del cronograma.

La manera de indicar la solución del ejercicio es ir introduciendo las señales y los valores de éstas en cada uno de los periodos de ejecución de la instrucción (figuras 9 y 10)

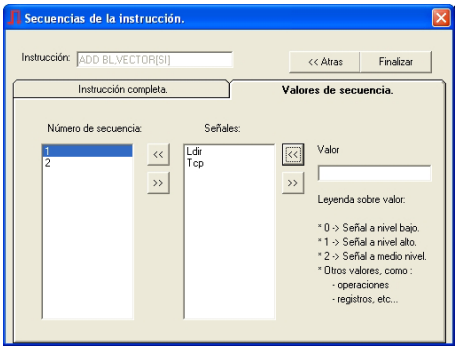


Figura 9. Introducción de los periodos.

Figura 10. Introducción de las señales de cada periodo de ejecución.

La descripción de lo que hace cada periodo se muestra en la ventana inferior, la del historial, tal y como muestra la figura 11.

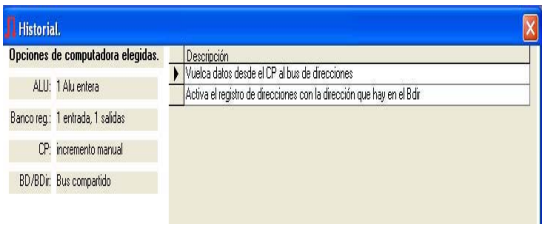


Figura 11. Historial de la ejecución de la instrucción.

El estudiante puede pedir ver la solución entera, o la solución de un determinado periodo y comprobar si lo ha resuelto, o no, correctamente. Los errores se muestran en rojo y la solución de alumno en verde. En la figura 12, los errores se han remarcado con un círculo.

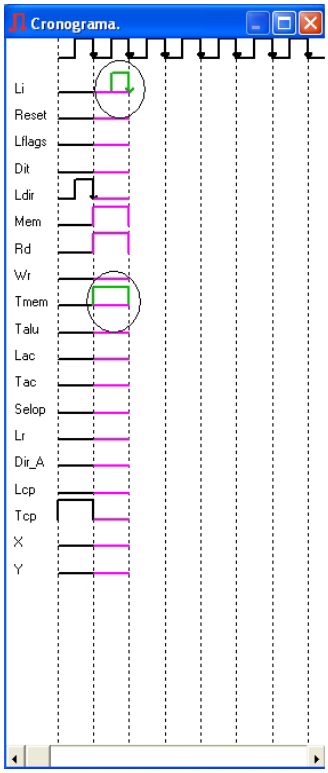


Figura 12. Errores en la solución del alumno.

El alumno puede solicitar en cualquier momento la solución de un periodo o la solución completa y la aplicación se la daría. El cronograma de la solución de la instrucción del ejemplo: **ADD BL, Vector [SI[** se muestra en la figura 13.

La aplicación también permite dejar la resolución del ejercicio para más adelante, con lo que el alumno tiene que guardar el estado actual del mismo. De hecho, guardaría la arquitectura generada y las fases que ha resuelto hasta el momento. De esa manera, al iniciar la aplicación nuevamente, en la pantalla del asistente, el usuario deberá elegir la opción de "Recuperar instrucción incompleta" (figura 14)

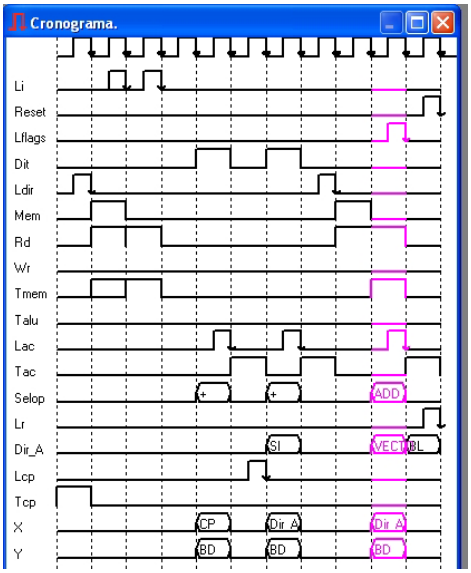


Figura 13. Solución del programa Previuc.

Al recuperar una instrucción incompleta, el programa le mostrará las instrucciones que ha dejado a medias en otras sesiones. Entre ellas el estudiante podrá elegir con cual continuar, simplemente, marcándola de la lista.

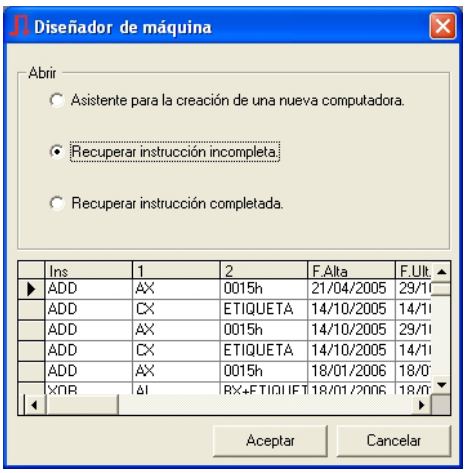


Figura 14. Recuperación de instrucciones incompletas.

Finalmente, para el modo tutorial, la opción de la aplicación es desde el asistente escoger "Recuperar instrucción completada" (figura 15)

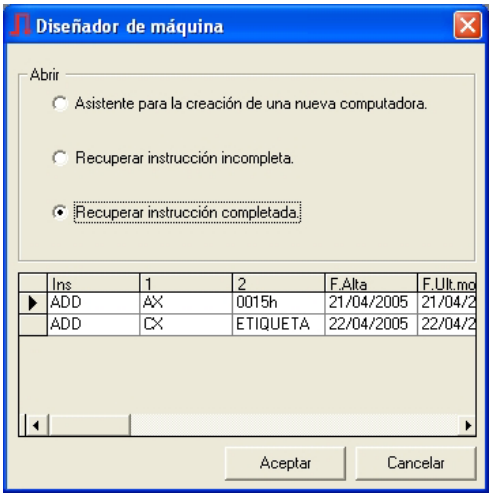


Figura 15. Recuperación de instrucciones completadas.

Una vez que se recupera una instrucción completada, se pueden ver los periodos de los que constó su ejecución, las señales generadas, su cronograma y la explicación del mismo en la ventana del historial.

6. Sugerencias docentes

Es aconsejable que al estudiante se le entregue el programa Previuc junto con una guía metodológica para su aplicación.

En dicha guía se le debe indicar cómo se construye uno de los posibles computadores elementales y las opciones que va a tener disponibles.

También se le debe solicitar la ejecución y entrega impresa de los cronogramas de dos o tres instrucciones con diferentes modos de direccionamiento (entre los disponibles) para que el profesor evalúe si está empleando el recurso docente de manera correcta o no.

Finalmente, y una vez comprobado que el estudiante no tiene problemas con la aplicación, se le puede dejar sólo resolviendo tantos ejercicios como él desee de manera totalmente autónonoma.

7. Conclusiones

Se ha desarrollado una aplicación que permite al estudiante suplir la falta de ejercicios resueltos del tema de la Unidad de Control. Por lo que se espera que en el próximo cuatrimestre la puedan emplear en sus estudios.

No obstante, los autores consideramos que en un futuro se podría desarrollar la misma aplicación, pero para la Web, de manera que sea accesible a todos los estudiantes, incluso, desde sus domicilios y puestos de trabajo.

Referencias

[1] **Britton, Robert L.** MIPS Assembly Languaje Programming. Prentice Hall, 2004.

[2] Boehm, Erich. DLX Distribution Homepage. http://www.wuwien.ac.at/usr/h93/h9301726/dlx.html, 1996.

[3] Simuproc. http://simuproc.softonic.com/

[4] **Concheiro, Raquel ; Loureiro, Marta; Amor, Margarita y González, Patricia.** Simula3MS: simulador pedagógico de un procesador. JENUI'05

[5] **De Diego Martínez, Rubén.** MSX88: Una Herramienta para la Enseñanza de la Estructura y Funcionamiento de los Ordenadores. URSI'94

[6] B.O.E. del 24 de mayo de 1995.

[7] **Patterson, D. A., Patterson J. L**. *Estructura y diseño de computadores. Reverté 2000*

El Control de Versiones en el aprendizaje de la Ingeniería Informática: Un enfoque práctico

Fran J. Ruiz-Bertol, Francisco Javier Zarazaga-Soria
Departamento de Informática e Ingeniería de Sistemas
Universidad de Zaragoza
c/ María de Luna 1, 50018 Zaragoza
franjr@unizar.es, javy@unizar.es

Resumen

En la formación necesaria para el desempeño de las tareas de un Ingeniero en Informática es necesario conocer, aprender y usar herramientas para la gestión de proyectos. Esto debe llevarse a cabo haciendo uso de los recursos disponibles en el marco de la enseñanza universitaria, y sin olvidar el trasfondo de aprendizaje que hay más allá de la herramienta específica que se utilice. Una aproximación muy válida es la basada en el uso de recursos *open source* de éxito por dos razones: el alumno va a poder contar con ellos en su vida profesional (cosa que no ocurre con los productos comerciales); y suelen ser el resultado de la amalgama de buenas prácticas y experiencias de grupos muy heterogéneos. Uno de los aspectos fundamentales del proceso de aprendizaje en las habilidades de gestión de proyectos es tener un control sobre los cambios que se producen durante un proyecto, para facilitar la gestión, revisión y auditoria de tanto los productos o servicios software producidos como el proceso de cambios realizado. En este artículo se presenta el aprendizaje de un sistema de gestión de configuraciones desde un enfoque práctico, apoyado en una herramienta *open source* con dos objetivos primordiales: (i) conocer y aprender a usar una herramienta para la gestión de configuraciones y (ii) afianzar los conceptos de control de cambios, versiones y auditoria de una manera práctica.

1. Introducción

La sociedad demanda cada día más a profesionales universitarios que sean capaces de incorporarse al mercado laboral con plenas garantías de que su formación durante su carrera ha sido adecuada tanto a nivel teórico como práctico. Concretamente, para un Ingeniero en Informática se requiere una buena base teórica de aspectos algorítmicos, de diseño, de arquitecturas, capacidad de resolver problemas, capacidad de análisis, entre otros. Todo este conjunto de aspectos teóricos se cumplen de manera adecuada.

Desde un punto de vista práctico, un Ingeniero Informático debe ser capaz de desenvolverse y adaptarse a las situaciones o problemas que surgen durante su desarrollo profesional, así como aplicar técnicas, herramientas y habilidades aprendidas durante sus estudios. Esto implica que los futuros profesionales informáticos, no sólo deben tener los conocimientos teóricos de cómo resolver ciertas situaciones que pueden darse en el mundo profesional, sino que también deben estar dotados de un conocimiento práctico y experiencias que le permitan seleccionar en cada momento el conjunto de herramientas necesarias para la toma de decisiones.

En el libro blanco del título de grado en Ingeniería Informática [1], una de las competencias específicas fundamentales que es común para los tres perfiles sugeridos (Desarrollo Software, Sistemas, y Gestión y Explotación de las TI) es la Dirección, Planificación y Gestión de Proyectos. Para capacitar al alumno en esta competencia, es necesario desarrollar su aprendizaje desde un nivel eminentemente práctico durante el estudio de la carrera como competencia transversal, y formalizarla en los últimos cursos de la titulación para que ésta sea efectiva en su proceso de aprendizaje como Ingeniero. El actual plan de estudios de Ingeniería en Informática de la Universidad de Zaragoza (BOE de 1 de febrero de 1995) estructura la materia troncal de Ingeniería del Software en tres asignaturas: Ingeniería del Software I, Ingeniería del Software II y Proyectos. Para esta última asignatura se han reservado los contenidos directamente relacionados con la gestión de

proyectos. Una asignatura de estas características es parte del plan de estudios troncal común a todas las Universidades Españolas (BOE de 26 de octubre de 1990), y también corresponderá a una materia fundamental para el Espacio Europeo de Enseñanza Superior en el grado de Ingeniería Informática (BOE de 21 de enero de 2005). De manera genérica, la gestión de proyectos puede definirse como "*la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades del proyecto para cumplir los requisitos*" [2].

Como los resultados principales del desarrollo de un proyecto informático son fundamentalmente productos o servicios software (código fuente, aplicaciones, manuales, documentos de trabajo, etc.) es imprescindible mantener una coherencia e integridad de dichos entregables. Este fin se alcanzará utilizando, por una parte, los conocimientos y habilidades necesarios para avanzar en el cumplimiento de requisitos, y, por otra, utilizando las herramientas y técnicas necesarias para que dichos productos de trabajo guarden una coherencia e integridad adecuada. Por ello, es necesario que durante su periodo formativo, los alumnos sean capaces de adquirir los conocimientos y puedan desarrollar las habilidades para el desempeño de la profesión, pero que también que sepan utilizar las herramientas necesarias que faciliten el desempeño de su tarea. A la hora de plantear una práctica en este contexto, ha resultado complejo dar con otros ejemplos que nos guíen. Esto ha llevado a la necesidad de transmitir nuestra experiencia para el aprovechamiento de otros.

En este artículo, presentamos la implantación de un sistema de control de versiones *open source* denominado Subversion en la asignatura Proyectos de la Universidad de Zaragoza. La implantación de este software supone, no sólo instalar los programas y servicios necesarios para que el software funcione, sino que además se encuentra el objetivo de que los alumnos hagan uso de forma guiada de una herramienta que sirva para el control de configuraciones, mantenimiento de versiones y revisiones, aprendizaje de los problemas que pueden surgir en un entorno de acceso compartido, y la introducción de los procesos de registro y calidad en el desempeño de la profesión.

En la sección 2 se presenta una guía general del proceso de implantación de Subversion y su cliente gráfico Tortoise SVN, detallando el conjunto de funcionalidades que permite este software y su aplicación lógica en el entorno de un proyecto software. La sección 3 detalla la metodología de trabajo desde el enfoque de los objetivos de aprendizaje y verificación de los mismos que se espera de los alumnos durante el desarrollo de la práctica. Este trabajo termina con una sección de conclusiones.

2. Subversión y Tortoise SVN

La gestión de configuración del software (SCM) se define como "*el conjunto de actividades diseñadas para el control de cambios identificando los productos de trabajo susceptibles de cambiar, estableciendo las relaciones entre ellos, definiendo mecanismos para la gestión de diferentes versiones de esos productos de trabajo, y controlando, auditando y transmitiendo los cambios realizados*" [3]. Hasta hace pocos años, el software SCM por excelencia era *Concurrent Versioning System* (CVS), creado por Grune en los años 80 [4], y que implementaba bajo una arquitectura cliente/servidor el conjunto de funcionalidades necesarias para el control de configuraciones. CVS trabaja principalmente bajo entornos Unix, pudiendo conectarse a dicho servidor cualquier cliente que implementara el protocolo de comunicación establecido por CVS [5].

Sin embargo, en los últimos años, CVS ha quedado relegado a un segundo plano tras la aparición de Subversion [6], debido, principalmente, a que este último soluciona problemas que existían en CVS, como el mantenimiento de versiones tras renombrar o mover un archivo/directorio, una difícil integración de los conjuntos de caracteres de internacionalización, o una compleja estructura del árbol de directorios del sistema. En esta sección se proporciona una breve guía sobre Subversion, así como los pasos necesarios para la instalación, configuración e implantación de la arquitectura cliente/servidor a través del cliente Tortoise SVN [7].

2.1. Subversion

Subversion (SVN) es un sistema de control de configuraciones *open-source*. Esta herramienta de control de versiones también puede considerarse como un repositorio que guarda la información de archivos y directorios, incluyendo la información asociada a las modificaciones realizadas. Los clientes pueden acceder a SVN para solicitar archivos, actualizarlos, modificarlos e incluso crear nuevos ficheros y disponerlos para los demás usuarios en el repositorio. Pero la funcionalidad de SVN no se basa en ser un software que actúa de servidor de archivos, sino que incorpora las características necesarias para ser un SCM. Entre estas características se encuentra el mantenimiento de versiones y revisiones antiguas, la información asociada a cada una de las versiones o revisiones, la auditoría, la posibilidad de que varios usuarios estén trabajando sobre un mismo archivo y se contemplen sin bloqueos las modificaciones realizadas por cada uno de los usuarios, comparar las diferencias entre archivos de distintas versiones, y disponer de un almacén de datos que contenga la información de un proyecto dada actualizada. Este último hecho se convierte en fundamental para muchas empresas de desarrollo software, equipos virtuales y esfuerzos para el desarrollo de software libre, ya que en muchas ocasiones los implicados en el desarrollo deben disponer en un momento dado de archivos compartidos (y posiblemente sujetos a cambios de manera continua y concurrente) y poder realizar modificaciones que no afecten a los demás usuarios. Por ejemplo, supongamos un equipo virtual, distribuido en diferentes localizaciones mundiales y con distintos husos horarios. Si un usuario accede para modificar una función del código fuente de un proyecto software, y a su vez, otro usuario está consultando y refinando otro conjunto de funciones del mismo código fuente, a través de SVN ambos usuarios podrán confirmar correctamente sus modificaciones a través del comando *merge* sin afectar a las modificaciones realizadas por el otro usuario.

SVN está disponible para la mayoría de plataformas (Windows, Linux/Unix, Mac y Solaris) [10]. A efectos prácticos, en este artículo se desarrollarán las instalación, configuración y la utilización de las funciones más importantes en una plataforma Windows, ya que éste será el sistema operativo a utilizar en las prácticas asociadas a la asignatura de Proyectos.

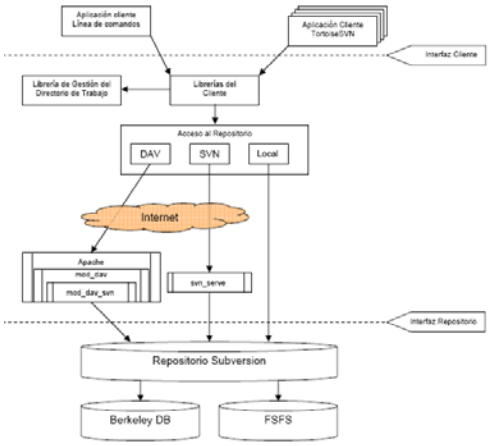


Figura 1. Arquitectura y Estructura de los módulos de Subversion obtenida de [8].

Primero es necesario obtener los archivos de instalación más adecuados para nuestros propósitos. SVN se ofrece tanto como código fuente (siempre bajo licencia *open-soruce*) como en binario o compilado para cada una de las plataformas. Una vez descargado, la instalación es completamente guiada, y dejará instalado SVN en el directorio seleccionado. Después, será necesario arrancar el servicio correspondiente para poder trabajar con repositorios. Sin embargo, esto implica que es necesario configurar adecuadamente el software, que dependerá de la arquitectura que deseemos para la utilización de SVN.

Desde el punto de vista de la conexión, SVN proporciona tres arquitecturas: local, usando el módulo `web_dav_svn` en Apache, y a través del servicio `svnserve` (ver Figura 1).

Para crear un repositorio SVN tanto en modo local como remoto, se utiliza el siguiente comando desde la consola: `svnadmin create "file:///Ruta/Repositorio"`, siendo este último parámetro el directorio donde vamos a crear el repositorio. Si dicho repositorio quiere exponerse para que actúe de manera remota, se instala como servicio dentro de Apache

incluyendo el módulo `web_dav_svn` en la configuración del servidor Apache, e indicando la ruta del repositorio recién creado [9]. Si disponemos de un servidor limitado, o no deseamos configurar el servidor Apache, SVN proporciona el servicio autónomo `svnserve`, que permite utilizar el repositorio de la misma manera que con el servidor.

Desde el punto de vista de almacenamiento de la información, los repositorios SVN pueden crearse sobre dos tipos de almacenes de datos: Berkeley DB [10] y FSFS [11]. Berkeley DB proporciona un almacén de datos escalable y un acceso rápido, soportando transacciones reales, copias de seguridad y coherencia en los datos. FSFS proporciona otro enfoque en el manejo de ficheros en el repositorio, aunque proporciona todas las características de Berkeley DB pero sin ser en el sentido estricto una base datos. FSFS, además, mejora las transacciones incluyendo en un único fichero de transacción todas las modificaciones de un usuario, reduce el tamaño del repositorio (en torno a un 20%), y es independiente de la plataforma [12].

Una vez creado el repositorio SVN, para utilizar dicho repositorio tanto de manera local como remota, es necesario una copia de trabajo usando el siguiente comando `svn checkout` repositorio. Este comando sirve tanto para crear una copia local de los archivos que hay en el repositorio, pero también para reemplazar y actualizar los ficheros existentes en la copia de trabajo.

Ya esa en modo local como remoto, es necesario establecer una autenticación de los usuarios que les permita acceder al repositorio SVN, y una autorización que permita a dichos usuarios acceder en modo lectura, lectura/escritura o permitir el acceso de dichos usuarios a ciertos recursos del repositorio SVN [8]. Esto se realiza a través de los ficheros de configuración que se encuentran en el directorio `/conf` del repositorio. Para establecer las políticas de autorización y autenticación, SVN dispone del fichero svnserve.conf, que determina los métodos montados sobre el repositorio para determinar los archivos de autenticación y autorización, proporcionando incluso métodos de configuración a través de SSH.

Entre las operaciones disponibles en SVN se destacan las siguientes:

- `create`. El comando `svn create directorio` crea un nuevo repositorio. Una vez este se crea, habrá que arrancar los servicios remotos adecuados (si es necesario) y establecer la configuración para la autenticación y autorización de los usuarios.
- `import`. El comando `svn import [origen] repositorio` permite realizar una importación inicial o volcado al repositorio de forma recursiva.
- `list`. El comando `svn list` permite listar el contenido del repositorio.
- `update`. Si no estamos seguros de si estamos trabajando con los archivos en su última versión en nuestra copia privada, es necesario que actualizar la copia de trabajo usando el comando `svn update repositorio`. Este comando comprueba si el contenido del repositorio y de la copia de trabajo son coherentes, y en caso contrario, transmite los ficheros contenidos en el repositorio a la copia de trabajo.
- `commit`. El comando `svn commit` confirma en el repositorio los cambios realizados en la copia de trabajo. Si existe conflicto de datos, nos avisa de ello.
- `add`. El comando `svn add` añade nuevos ficheros/directorios a la copia de trabajo. Sin embargo, estos cambios no son efectivos hasta que los ficheros o directorios son confirmados a través del comando `commit`.
- `delete`. El comando `svn delete` elimina de la copia de trabajo los archivos o directorios especificados como parámetro. El efecto del comando será efectivo cuando se realice la operación `commit`.
- `copy`. El comando `svn copy origen destino` copia el fichero o directorio origen al fichero o directorio destino a la copia de trabajo actual. En el repositorio se comprometerán estos cambios tras la operación `commit`.
- `move`. El comando `svn move origen destino` renombra el fichero o directorio origen al nombre especificado por destino. Los cambios serán efectivos en el repositorio tras la confirmación.
- `status`. El comando `svn status` determina el estado de la copia de trabajo en comparación con el contenido del repositorio.

El estado se determina a través de un carácter, que puede ser 'A' (*add*), 'D' (*delete*), 'M' (*modified*), R (replaced) ó '?' (*non versined*).

- `diff`. El comando `svn diff` compara y detalla por pantalla las diferencias entre las distintas versiones de los archivos de la copia de trabajo y del repositorio.
- `revert`. El comando `svn revert archivo` establece en la copia de trabajo la versión inmediatamente anterior del archivo especificado.

2.2. Tortoise SVN

Tortoise SVN [7] es una aplicación basada las bibliotecas SVN que actúa de cliente gráfico para el acceso al repositorio SVN, tanto si éste es remoto como si es local. Este cliente gráfico ha sido diseñado para su integración contextual en el Explorador de Windows proporcionando el acceso a la mayoría de funciones que proporciona SVN.

Desde Tortoise SVN se pueden realizar todas aquellas operaciones relacionadas con la administración del directorio de trabajo (añadir, eliminar, modificar, mover), comunicación con el repositorio SVN (checkout, commit, actualizar, revertir, bloquear), resolución de conflictos (ramas, fusionar, exportar, relocalizar) y la creación de repositorios locales SVN. Un ejemplo de este menú contextual puede observarse en la Figura 2.

Con Tortoise SVN se puede observar en cada momento de manera icono-gráfica el estado de los archivos (nuevo, actualizado, modificado, eliminado, movido, renombrado, fuera de versión), así como de la copia de trabajo. De esta manera, se permite trabajar de una manera transparente con los archivos contenidos en un directorio de trabajo, pero también ser consciente de que si ha habido modificaciones, nuevos datos, borrado o conflictos con los datos actuales, éstos se marcan para que el usuario sea consciente de las distintas operaciones que debe realizar para transmitir estos cambio al repositorio SVN mediante las operaciones correspondientes.

Este cliente para SVN también contiene las ventanas de diálogo asociadas a las operaciones para poder determinar las opciones asociadas a cada una de las operaciones, así como los avisos al usuario y notas de versión que proporciona Subversion.

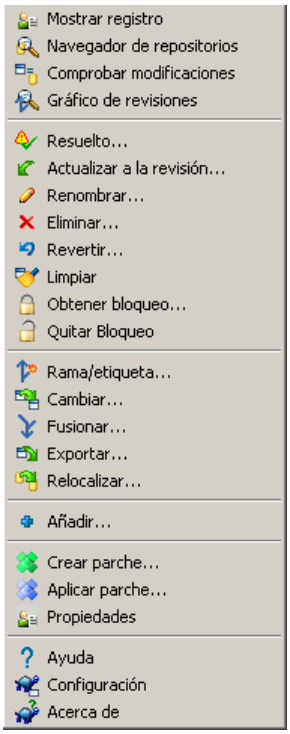


Figura 2. Menú contextual de Tortoise SVN integrado en el shell de Windows

3. Metodología de trabajo

El objetivo de conocer y utilizar este software viene dado a que la formación necesaria de los Ingenieros en Informática en las herramientas que facilitan la gestión de proyectos. De manera secundaria, a través de este software queremos que los alumnos aprendan el control de configuraciones, la gestión de calidad (a través de la anotación y seguimiento del proyecto durante su ciclo de vida) y conozcan el término auditoria tan importante para el desempeño profesional en entidades que poseen algún tipo de certificación de calidad (ISO 9001, Spice, CMMI, etc.).

El trabajo práctico se lleva a cabo en un entorno cliente-servidor sobre el sistema operativo Windows, donde primero se creará un repositorio en modo local, y luego se conectará a un servidor Solaris para probar las características de autorización y autenticación en modo remoto.

Como primer paso se procede a instalar por parte del profesorado Subversion en un servidor remoto y crear allí un repositorio SVN. En dicho repositorio se dejarán documentación y código fuente para que los alumnos puedan trabajar. Por otra parte, en cada uno de los equipos que vayan a utilizar los grupos de trabajo se instala el cliente Tortoise SVN, que les permita acceder a todas las funcionalidades para trabajar con un SCM.

Posteriormente, se definirán las reglas de autenticación y autorización de los grupos correspondientes con las cuentas de dominio asociadas, estableciendo permisos de lectura y escritura sobre los directorios que hemos dejado en el servidor, teniendo para cada grupo el acceso a un directorio propio, a un directorio compartido para todos, y varias combinaciones en modo lectura, y lectura/escritura para que puedan observar de una manera práctica las distintas operaciones que pueden realizarse con SVN. La documentación asociada a la realización de esta experiencia se encuentra publicada y disponible en [13]. Las operaciones que se desea que realicen los alumnos son las siguientes:

- Preparación del trabajo para un repositorio. Antes de crear los repositorios y las copias de trabajo, se debe realizar una preparación de la estructura del directorio de trabajo. Esta estructuración es principalmente crear una línea base de trabajo (`/trunk`), y los directorios para las ramas y etiquetas (`/branches` y `/tags`).
- Creación de un repositorio local. Se solicitará al alumno que cree un repositorio local usando el cliente Tortoise SVN, y se le proporcionará la documentación necesaria para que el alumno aprenda la arquitectura de un repositorio SVN, y cómo está estructurado éste para poder actuar de repositorio.
- Importación inicial. Una vez que están preparados el repositorio y el trabajo a importar, se realiza la importación a través del comando Importar de SVN.
- Exploración del repositorio. Una de las funciones más interesantes es comprobar el contenido del repositorio, ya que desde la ventana del navegador del repositorio nos proporciona un conjunto de operaciones de copiar, mover, eliminar y añadir nuevos ficheros, entre otras.
- Creación de una copia de trabajo. Se solicitará al alumno que cree un nuevo directorio de trabajo que se asocie al repositorio SVN recién creado a través del checkout. Esto dará lugar a la copia de trabajo local. Con esta tarea el alumno debe ser capaz de comprender la diferencia entre un repositorio, un directorio de trabajo y una copia de trabajo.
- Modificación de ficheros. Se ha proporcionado al alumno un sencillo código fuente en Java para la conversión de euros a libras. Para esta tarea será necesario que los alumnos trabajen sobre su copia de trabajo en la acción. Para comprobar las diferencias entre el archivo anterior y el que han tocado, podrán utilizar el comando diff.
- Comprobación de modificaciones. Es interesante que el alumno conozca qué modificaciones hay entre la copia de trabajo y el repositorio.
- Creación de nuevos ficheros. Se solicitará al alumno que cree en su copia privada nuevos archivos en la copia privada. Para ello, la manera más sencilla de crear nuevos archivos es solicitarles compilar (`javac`) y generar la documentación (`javadoc`) para el código fuente proporcionado.
- Actualización y comprometer los cambios. Estas dos operaciones son fundamentales para que se puedan confirmar los cambios realizados por los alumnos sobre el repositorio local, así como para que incluyan cada vez que comprometan estos cambios, las notas y mensajes asociados a la versión o revisión correspondiente.
- Resolución de conflictos. Se pondrá al alumno frente a un conflicto de datos, utilizando para ello un directorio compartido remoto donde se creó el repositorio. Su principal función será ser capaces de resolver este conflicto usando para ello las distintas copias existentes del archivo (revisión de donde se obtuvieron los datos, revisión actual del repositorio y copia local).
- Modificación concurrente de ficheros (opciones merge). Se solicitará a distintos grupos trabajen sobre el mismo fichero, modificando distintas partes del código fuente de un archivo. Posteriormente se solicitará que cada uno de ellos comprometa los datos, y

trabajen con la operación `merge` para hacer efectivos los cambios de ambos grupos.

- Auditar los cambios realizados sobre todo el repositorio. Los alumnos deben ser capaces de auditar los cambios realizados sobre el repositorio SVN, y observar el proceso que se ha seguido para obtener la configuración ó versión actual.
- Conexión remota y trabajo concurrente. Se solicitará al alumno que trabaje sobre un repositorio remoto donde tenemos arrancado un servidor SVN, y que realice operaciones de consulta y modificación.
- Bloqueo de archivos. Finalmente, se observará la efectividad de la operación de bloqueos, y cuándo debe utilizarse ésta.

De esta manera, cada uno de los alumnos conocerá las opciones básicas de la utilización de un SCM, y que son comunes a todos los sistemas de control de versiones. Su aprendizaje es fundamental no sólo como contenido exclusivo de la asignatura Proyectos, sino que, una vez adquirido el conocimiento de cómo utilizar esta herramienta, también han aprendido a utilizar una materia básica para poder gestionar los proyectos, adquiriendo diferentes roles: administrador, director, programador, gestor de proyectos, etc.

Esta experiencia docente tiene el objetivo de que los alumnos puedan aplicar lo aprendido durante el desarrollo de la práctica al desarrollo de su proyecto fin de carrera, y posteriormente a su carrera profesional, proporcionando así un valor añadido al finalizar su periodo de formación.

4. Experiencia práctica

A la hora de remitir la versión definitiva de este artículo, los alumnos han desarrollado esta experiencia docente de una manera eficaz, destacando en su gran mayoría la utilidad de aplicar esta experiencia de aquí en adelante, tanto en lo restante de sus estudios, como en su desarrollo profesional.

Los alumnos que han desarrollado esta práctica han sido conscientes de los problemas que surgen al trabajar con un sistema de gestión de configuraciones, pero cómo este hecho facilita la coordinación de los equipos de trabajo y la seguridad que proporciona disponer de un repositorio centralizado para los archivos de trabajo. De hecho, muchos alumnos han evidenciado que durante sus estudios han borrado alguna vez de manera accidental parte de sus prácticas, sin opción de poder recuperarlas.

Durante el desarrollo de la práctica, lo más complicado ha sido familiarizarse con las operaciones de Subversion y Tortoise SVN, ya que una gran minoría había trabajado anteriormente con sistemas de control de versiones. Tras comprender la documentación entregada y utilizarla de manera práctica, han adquirido dicho conocimiento.

5. Conclusiones

En este artículo se ha querido exponer una experiencia práctica de cómo aplicar la gestión de configuraciones durante la formación final de un Ingeniero en Informática. Esta experiencia ha surgido de la necesidad del compromiso de que la Universidad forme de manera adecuada a los alumnos, estando supeditado dicho aprendizaje a su asociación a cada materia docente impartida. En este caso concreto, la asignatura Proyectos debe formar en las capacidades de dirección, gestión, control y auditoria sobre los proyectos software.

Sin embargo, nos hemos visto en el problema de que en muchas Universidades Españolas no se encuentran experiencias docentes de este tipo o no han sido publicadas. Creemos fehacientemente que dentro de las competencias que deben adquirir los futuros Ingenieros en Informática es precisamente conocer y saber controlar las versiones de todos los productos/servicios que generen, independientemente de que a nivel corporativo se tenga previsto un proceso de copia de seguridad. Pensamos que esta experiencia resulta interesante de compartir con otros compañeros que se hayan fijado un objetivo docente similar al nuestro.

Así mismo, está a disposición de aquellas personas que lo soliciten el contenido de la práctica realizada [13]. Para ello, se recomienda que contacten con los autores de este artículo.

Referencias

[1] ANECA. *Libro Blanco: Título de Grado en Ingeniería Informática*. ANECA, 2005.

[2] Project Management Institute. *A guide to the Project Management Body of Knowledge: PMBOK guide*. 3rd. Ed. Project Management Institute, 2004.

[3] Presman, RS. *Software Engineering: A Practicioner's Approach*. 6th Ed. McGraw-Hill, 2005.

[4] Grune D. *Concurrent Versions System, a method for independent cooperation*. Internal Report 113, Vrije Universiteit, Amsterdam, pp. 9, 1986.

[5] Fogel, K. Bar, M. *Open Source Development with CVS*. 3rd Ed. Paraglyp, 2003.

[6] Tigris.org. *Subversion.* Electronic Source: http://subversion.tigris.org/ [Último acceso: 13/02/2007]

[7] Tigris.org. *Tortoise SVN*. Electronic Source: http://tortoisesvn.tigris.org/ [Último acceso: 13/02/2007]

[8] Collins-Sussman, B. Fitzpatrick, BW. Pilato, CM. *Version Control with Subversion*. O'Reilly, 2004.

[9] Clemm G. *et al*. *RFC 3253: Versioning Extensions to WebDAV (Web Distributed Authoring and Versioning)*. WebDAV Standards Track, Proposed Standard. 2002.

[10] Oracle Berkeley DB. *Berkeley DB*: Oracle Embeded Database. Electronic source: http://www.oracle.com/database/berkeley-db/db/index.html [Último acceso: 13/02/2007]

[11] FSFS. *The Fast Secure File System*. Electronic Source: http://fsfs.sourceforge.net/ [Último acceso: 13/02/2007]

[12] FSFS. *Subversion FSFS implementation: FSFS filesystem type repository*. Electronic Source: http://svn.collab.net/repos/svn/trunk/notes/fsfs [Último Acceso: 13/02/2007]

[13] Ruiz-Bertol, FJ; Zarazaga, FJ. *Control de Configuraciones. Control de Versiones con Subversion*. Educational Report ER-04-07. Dpto. Inf. e Ing. Sistemas. Univ. de Zaragoza, 2007.

ECTS y Moodle: Guía docente y aplicación práctica

Pascual Timor Hermano[1], Pablo Moreno Meseguer[2], Jordi Forcada Cerdà[3]
[1]Dpto. de Proyectos
Escola Superior de Cerámica de Manises
Ceramista Alfons Blat s/n, 46940, Manises
pascual@eees.eu

[2] Dpto. de Ciencias Aplicadas y Tecnología
Escola D'Art i Superior de Disseny de Castelló
Fadrell 1, 12002, Castellón
pablo@eees.eu

[3]Dpto. de Ciencias Aplicadas y Tecnología
Escola D'Art i Superior de Disseny d'Alcoi
Alzamora 1, 03801, Alcoy
jordi@eees.eu

Resumen

Este recurso docente complementa la ponencia "El espíritu de los créditos ECTS y el pragmatismo de Moodle" y de la misma manera también consta de dos partes diferenciadas pero complementarias.

En la primera parte se aborda la Guía docente de la asignatura *informática aplicada al diseño gráfico (Adobe Photoshop®)* bajo el prisma de los créditos ECTS (European Credit Transfer System).

En la segunda se describe cómo implementar la guía docente de la asignatura mediante el paquete de software Moodle (Modular Object-Oriented Dynamic Learning Environment) con los requisitos que demanda la Educación Superior en el espacio común europeo.

1. Introducción

El recurso docente que presentamos plantea una propuesta de aplicación a una asignatura basada en el sistema de créditos ECTS utilizando un software libre que sirve a los propósitos y el espíritu de los mismos en la Educación Superior en el Espacio Común Europeo.

Esta propuesta está basada en la experiencia previa acumulada durante dos cursos en los que hemos utilizado el entorno Moodle en los planes de estudio actuales en los que desarrollamos nuestra actividad docente. Por tanto esta propuesta no ha sido implantada todavía tal como se plantea aquí, es decir, bajo el prisma de los créditos ECTS.

Nuestra experiencia docente en la utilización de este entorno se ha llevado a cabo con un alumnado que ha probado distintas herramientas de las que ofrece el sistema (documentación, foros, exámenes, notas, comentarios,...)

La experiencia ha sido gratificante, ya que la adaptación de los alumnos ha sido rápida y sin problemas relevantes. Incluso los alumnos más reticentes han comprendido las ventajas de este nuevo sistema, tales como:

- Disponibilidad permanente de recursos y de información.
- Libertad para organizar el tiempo.
- Retroalimentación en cuanto a notas de trabajos y exámenes.
- Acceso a comentarios del profesor.
- Intercambio de información con los participantes del curso.

Por todo ello pensamos que este recurso docente podría conjugar varios factores: por un lado, la experiencia acumulada en la materia, por otro, el grado de madurez alcanzado en el uso de Moodle para desarrollar actividades en el aula y finalmente las exigencias planteadas por la Educación Superior en el Espacio Común Europeo.

Creemos que hemos evolucionado desde el uso de Moodle como mero organizador de contenidos y plazos hacia una concepción más constructivista del curso. En el estadio actual incorporamos más actividades de reflexión crítica en el trabajo individual y colectivo y sobre todo definimos el modelo de actividad que debe de realizar el alumno.

El planteamiento favorece los requisitos que demandan los créditos ECTS:

- La carga de trabajo esta razonablemente distribuida.
- No existe ambigüedad en el volumen de trabajo que debe realizar el alumno.
- Favorece la movilidad.
- Los resultados del aprendizaje son comparables y asequibles.
- Contacto permanente con el profesor
- Facilita la labor del profesor (programar, dirigir, acompañar al alumno en logro de los objetivos).

Para que se puedan visualizar con claridad estas premisas hemos dividido en dos apartados diferenciados el recurso: por una parte la Guía Docente de la asignatura Photoshop Aplicado al Diseño y por otra un ejemplo acotado en el tiempo del uso de Moodle.

2. Guía docente

2.1. Descripción de la asignatura

Nombre	Photoshop Aplicado al Diseño
Titulación	-
Código	-
Tipo	Troncal
Ciclo (nivel de la asignatura)	1º
Curso	1º
Carácter	Cuatrimestral (1º)
Nº de créditos	9
Prerrequisitos	*Básicos de Sistemas Operativos*
Profesores responsables	Jordi Forcada Cerdá Pablo Moreno Meseguer Pascual Timor Hermano

Tabla 1. Descripción de la asignatura

2.2. Objetivos

Objetivo general

Desarrollo de proyectos de diseño mediante el uso de programas informáticos aplicados a distintos sectores productivos.

Objetivos específicos

- Conocer los fundamentos de los archivos gráficos.
- Aprender a utilizar el escáner.
- Controlar la resolución del documento.
- Aprender a organizar un proyecto.
- Aprender a utilizar las herramientas de trazado.
- Convertir trazados en selección.
- Interconectar programas.
- Ejercer un control sobre las secuencias de acciones realizadas utilizando los comandos "Volver" e "Historia".
- Aprender a retocar y mejorar la calidad de una imagen escaneada.
- Aprender a trabajar con el "modo" de imagen más adecuado.
- Aprender a utilizar canales.
- Aprender a ajustar el color de las imágenes.

Para conseguir estos objetivos la materia ha sido estructurada en los siguientes bloques temáticos:

1. Fundamentos de los archivos gráficos.
2. Ajustes del documento y planificación del proyecto.
3. Descripción del programa Adobe Photoshop®. Usos y utilidades.
4. Descripción de la paleta de herramientas.
5. Retoques de imágenes.
6. Utilización de los canales.
7. Ajustes de color.

2.3. Competencias y destrezas

Competencias trasversales o genéricas (instrumentales, personales, sistémicas o de otros tipos)

- Capacidad de análisis y síntesis.
- Capacidad para aplicar los conocimientos teóricos en la práctica.
- Sensibilidad hacia la cultura objetual.
- Trabajo en un equipo de carácter interdisciplinar.
- Motivación por la calidad.
- Aprendizaje autónomo.
- Capacidad de organizar y planificar.
- Capacidad para comunicarse con expertos de otras áreas.
- Compromiso ético.
- Comunicación oral y escrita.
- Desarrollo de la capacidad de resolución de problemas y toma de decisiones.
- Razonamiento crítico.

Competencias específicas

- Elaboración y ejecución de proyectos basados en imágenes digitales.
- Capacidad de consideración multidisciplinar de un problema de diseño.
- Conocimientos generales básicos.
- Gestión de proyectos.

- Capacidad de interpretación cualitativa de datos.

2.4. Desarrollo ECTS

Considerando un tiempo de trabajo diario del estudiante de 8 horas, 5 días a la semana y 40 semanas de clase, se exigirían 1.600 horas de trabajo por curso académico. Dado que se ha fijado el número de créditos por curso en 60, cada crédito exigiría una carga de 26,7 horas de trabajo.

El Real Decreto 1125/2003, de 5 de septiembre que establece el sistema europeo de créditos en las titulaciones universitarias españolas, ha establecido el número mínimo de horas por crédito en 25 y el máximo en 30.

9 créditos por 25 horas/crédito = 225 horas /asignatura.

	Clase[1]	Presenciales fuera aula[1]	Factor trabajo estudiante	Trabajo personal[1]	Totales[1]	Créditos ECTS
Teoría	30		1	30	60	2,40
Seminarios	15		2	30	45	1,80
Prácticas	22,5		3	67,5	90	3,60
Tutorías		24			24	0,96
Búsqueda red				6	6	0,24
Total	67,5	24		133,5	225	9

Tabla 2. Desarrollo ECTS de la asignatura

2.5. Componentes de la programación

Programa de los contenidos teóricos

1. Fundamentos de los archivos gráficos:
1.1 Tipos de archivo: imágenes de mapa de bits y archivos de trazado vectorial.
1.2 Formatos de archivos: archivos de formato propio y archivos de formato de intercambio.
1.3 Concepto de resolución en las imágenes de mapa de bits.
1.4 Las curvas de "Bezier" en los archivos de trazado vectorial: nodos y manejadores.
1.5 Modos de color en las imágenes de mapa de bits.
2. Ajustes del documento y planificación del proyecto:
2.1 Formatos y unidades.
2.2 Ajustes de resolución.
2.3 Ajustes de impresión.
3. Descripción del programa Adobe PhotoShop®. Usos y utilidades:
3.1 Ventana del documento. barra de menús y ventanas.
3.2 El comando preferencias.
3.3 El comando ajustes de color.
3.4 Descripción de las herramientas de trazado en Adobe PhotoShop®.
4. Descripción de la paleta de herramientas:
4.1 Herramientas de selección.
4.2 Herramientas de recorte.
4.3 Herramientas de texto, dibujo, pintura y enfoque.
4.4 Herramientas de visualización y desplazamiento.
4.5 Control de color frontal y color de fondo.
4.6 Control de visualización en la ventana del documento.
4.7 Edición de máscara.
5. Retoques de imágenes:
5.1 Descripción del menú "Imagen".
5.2 Descripción del comando "Ajustar".
5.3 Descripción del comando "Modo".
5.4 Descripción del menú "Ver".
6. Utilización de los canales:
6.1 Activar y visualizar canales.
6.2 Comando de la paleta canales.
7. Ajustes de color
7.1 El comando "Tono / Saturación".
7.2 El comando "Niveles".

Programa de Seminarios

- Recursos en línea para Photoshop.
- Cámaras fotográficas reflex digitales para profesionales.
- Adobe Photoshop CS y el formato de archivo raw (*DNG Digital Negative Specification*).
- Programas de de gestión de *workflow* para fotografía profesional: *Aperture, Adobe Lightroom* y *Nikon Capture*.
- El estudio fotográfico.

Programa de los contenidos prácticos

Prácticas a realizar sobre imágenes digitales

1. El comando explorar: gestión de imágenes, cambio de nombre por lotes, el IPTC, el *photomerge*.
2. Trabajar con el contraste. Ajustar niveles.
3. Trabajar con el color. Ajustar curvas. Posterizar imágenes
4. Extraer y fusionar imágenes.
5. Fotomontajes.
6. El trabajo con canales.

[1] En horas.

7. Realización de un proyecto definido por el alumno.

Trabajos teóricos

Los trabajos teóricos representan la base cultural del proyecto. El alumno lo desarrollará a partir de los temas propuestos:

1. ¿Qué es la gamma?
2. La resolución.
3. El histograma
4. La interpolación.

El alumno deberá:

- Elegir el tema
- Documentar el tema
- Realizar el guión
- Elaborar el trabajo escrito.

El trabajo se realizará en tres etapas:

1. Elaboración del listado de referencias bibliográficas y online y elaboración del guión del trabajo.
2. Elaboración del borrador completo del trabajo.
3. Trabajo definitivo, que se presentará en formato digital, constará al menos de una memoria (en la que se usará la plantilla) y una presentación en PowerPoint, la memoria constará de:

- Índice
- Introducción justificativa
- Contenido (desarrollo del tema)
- Bibliografía

2.6. Bibliografía

Sobre papel

- CÓRDOBA, E.; GONZÁLEZ, C.; CÓRDOBA, C.: *Photoshop para fotógrafos y artistas*, Madrid: Ra-Ma Editorial, 2006.
- MCCLELLAND, DEKE; ULRICH, LAURIE: *La Biblia de Photoshop CS2*, Madrid: Ediciones Anaya Multimedia, S.A., 2006.
- AAVV: Adobe *Photoshop 6.0 para Windows y Macintosh*, San José (California, USA), 2000.
- ESCUDERO, SOFÍA: *Photoshop 6 Curso de inici ción*, Barcelona, Inforbook's, S.L., 2000.
- WILLMORE, BEN: *Adobe Photoshop 5 avanzado para Windows*, Madrid: Ediciones Anaya Multimedia, S.A., 1999.

Recursos On-line

- <http://www.adobe.com/es/>
- <http://www.dpreview.com/>
- <http://www.gusgsm.com/>
- <http://www.quesabesde.com/>

2.7. Métodos docentes

1. Explicaciones teóricas de los temas incluidos en el programa
2. Realización de los supuestos prácticos
3. Ponencias y debates sobre las temas de los seminarios
4. Fomento de la participación del alumno en las sesiones
5. Utilización de nuevas tecnologías en las sesiones de clase y en la preparación de las mismas y de los trabajos del alumno

2.8. Evaluación

El sistema de evaluación se estructura en los siguientes componentes:

Participación en la calificación	(%)
Trabajo teórico	20
Seminario	20
Prácticas y proyecto	60

Tabla 3. Obtención de la calificación

2.9. Idioma en que se imparte

Castellano

3. Creación de la asignatura en Moodle

3.1. Antes del inicio de las clases

El diseño de las actividades del curso y la pauta temporal son especialmente importantes en la concepción ECTS del desarrollo de la docencia.

De las posibilidades que ofrece Moodle para estructurar los contenidos de los cursos (semanal, social, por temas), se ha elegido el formato semanal. La razón es sencilla: con un simple vistazo el alumno ve la planificación temporal del curso.

También existe el módulo de calendario para tener una vista mensual de los eventos futuros del curso.

Considerando un tiempo de trabajo diario del estudiante de 8 horas, 5 días a la semana y 40 semanas de clase, se han distribuido los contenidos de manera que cada semana haya que realizar, al menos, un par de tareas distintas. El desarrollo teórico y las prácticas

se van desarrollando a lo largo del curso, introduciendo cada 6/8 semanas un seminario de los propuestos. A su vez, el alumno debe elegir un trabajo teórico de los propuestos e ir desarrollándolo a lo largo del curso. El trabajo se ha dividido en fases y en cada una de ellas el alumno debe entregar lo que se pide (aproximadamente cada 6 semanas se debe realizar una entrega.)

Por lo tanto, antes de empezar el curso ya se tiene diseñada, distribuida temporalmente y explicitada la marcha que debe seguir el alumno para alcanzar los objetivos deseados.

3.2. Desarrollo del curso

El curso se desarrolla a grandes rasgos como sigue:

- El alumno recibe información por parte del profesor u obtiene información por su cuenta.
- El alumno procesa y aplica esa información mediante los trabajos o tareas propuestos por el profesor.
- El alumno crea y entrega el resultado de las dos fases anteriores.

El profesor evaluará tanto el resultado como el proceso desarrollado por el alumno.

Herramientas de Moodle utilizadas

De todas las posibilidades que ofrece Moodle se han escogido:

- El formato semanal, como ya se ha indicado antes.
- Etiquetas para organizar los contenidos.
- Enlaces a ficheros o web para la documentación.
- Tareas para las prácticas y entregables de los alumnos
- Cuestionarios para la autoevaluación de los alumnos.
- Encuestas sobre el curso y el profesorado.
- Foros para la tutoría y la información general.

La estructura y etiquetado de los diferentes ítems se ha planteado de forma que sea intuitiva y no necesite ninguna explicación adicional.

Seguimiento del alumno y retroalimentación

Tres son los métodos principales que se tienen en el curso para la comunicación y el seguimiento del alumno:

1. Actividades del curso que se desarrollen de forma presencial.
2. Las entregas de tareas de los alumnos.
3. El desempeño del alumno en la tutoría (en el siguiente apartado).

De estos tres ámbitos Moodle da soporte a los dos últimos. Cuando se califica una tarea, el profesor puede escribir los comentarios oportunos y quedan registrados y asociados a dicha tarea.

Tutoría

La tutoría se ha implementado mediante un foro, calificable, ya que la tutoría se entiende como una faceta más de la actividad del curso donde alumnos y profesores ponen en común sobre temas relacionados con el curso. Algo más que la tradicional resolución de dudas individuales.

Evaluación de los alumnos

Si bien la distribución temporal que presenta el curso impone un ritmo, considerado ideal para alcanzar los objetivos en el tiempo destinado a la asignatura, cada alumno puede avanzar a su ritmo según se configuren las tareas de Moodle.

Alcanzar los objetivos en el tiempo previsto es otro factor a determinar en la calificación final del alumno.

De forma general las tareas se califican de 0 a 100 y se agrupan jerárquicamente según este esquema:

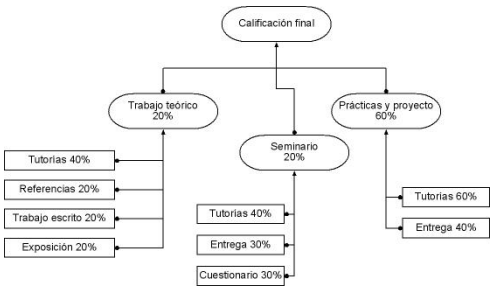


Figura 1. Esquema de calificación.

Moodle permite la calificación con letra. Se ha utilizado la siguiente correspondencia.

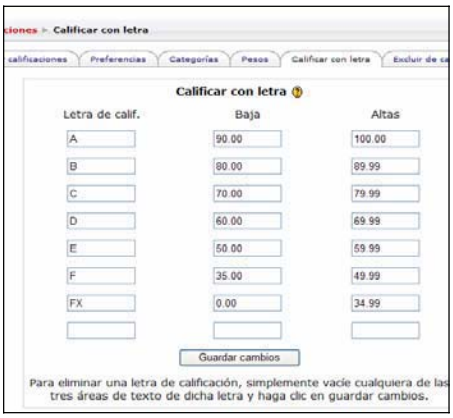


Figura 2. Calificación con letras

A modo de ejemplo de ejemplo las calificaciones funcionan así:

Figura 3. Calificaciones en Moodle

Lo único que Moodle no nos ofrece en este momento es un cálculo de porcentajes por letra, que hemos suplido mediante una hoja de cálculo que los obtiene a partir del pegado de los datos de calificaciones desde Moodle.

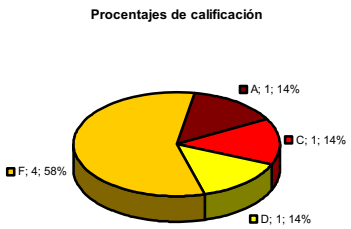


Figura 4. Porcentajes de calificación

Evaluación del curso

La evaluación de la actividad docente puede abordase en Moodle desde dos puntos de vista:

1. Utilizar un foro exclusivo para profesores donde recoger las experiencias positivas o negativas de la actividad docente, de forma que para el siguiente curso se puedan reforzar o corregir.
2. Utilizar la actividad de "encuesta" para conocer la opinión del "público" sobre el desarrollo del curso, la adecuación de materiales e instalaciones, el trabajo de los profesores, y cuantos aspectos del curso sea interesante conocer la valoración por parte de los alumnos.

4. Conclusiones

En el recurso que hemos planteado se propone un modelo de trabajo basado en la utilización de una herramienta informática (Moodle) para gestionar la guía docente de una asignatura (Photoshop aplicado al Diseño) programada bajo el prisma de los créditos ECTS.

El modelo de trabajo evidencia una serie ventajas para el estudiante:

- Le permite conocer desde el inicio del curso como está distribuida la carga de trabajo, es decir le facilita la realización de las actividades y la movilidad.
- Le facilita el acceso a los contenidos y al intercambio de información con todos los participantes.
- Le garantiza el acceso a las calificaciones.

Este modelo da soporte también al nuevo rol del profesor:

- Le facilita el trabajo de programar y definir los contenidos y las actividades de la asignatura.
- Le permite seguir y guiar la actividad del estudiante en el aprendizaje.

Por tanto, creemos que Moodle es una herramienta eficaz para dar cobertura al desarrollo de la actividad docente bajo el espíritu del sistema ECTS.

Tutorial gráfico interactivo de estructuras de datos

Mariano J. Cabrero Canosa, Javier Blanco Gutiérrez
Departamento de Computación, Universidade da Coruña
Campus de Elvira s/n
15071 A Coruña
cicanosa@udc.es

Resumen

En este artículo se presenta un sistema tutor interactivo de estructuras de datos. El objetivo es mostrar mediante animaciones gráficas el comportamiento de las operaciones básicas sobre las estructuras de datos clásicas, proporcionando al usuario un control total sobre la ejecución de los algoritmos, no sólo mostrándole paso a paso el efecto de las distintas operaciones sobre la estructura, sino permitiéndole cambiar las condiciones de ejecución. Por otro lado, proveerá de diversos controles sobre la reproducción de la animación, permitiendo la ejecución paso a paso, el avance, el retroceso o la parada. Para facilitar el acceso a la herramienta y ampliar el rango de usuarios potenciales se ha implementado de manera independiente a la plataforma hardware y software, siendo accesible vía web a través de un sencillo applet. En su desarrollo se ha utilizado un sistema de animación basado en Java, JAWAA, que proporciona un lenguaje simple de comandos para la creación de animaciones y su visualización a través de un navegador.

1. Introducción

La animación de algoritmos es una forma de visualización de software que consiste en utilizar elementos como el movimiento, el color, la forma y el sonido con el propósito de generar representaciones gráficas dinámicas asociadas a conceptos abstractos que definen el comportamiento de un algoritmo dado [1].

Supongamos la implementación de un conjunto prefijado de operaciones para manipular una estructura de datos. La animación permite que, a través de una sencilla representación gráfica de la estructura, sea posible mostrar de una forma didáctica y dinámica los cambios ocurridos como resultado de la ejecución de alguna de dichas operaciones. Permite ver, por tanto, la ejecución paso a paso de la operación, explorando los detalles subyacentes del algoritmo involucrado para obtener una mejor comprensión de su lógica.

Desde el punto de vista docente, la animación de algoritmos supone un apoyo importante en el aprendizaje de los conceptos básicos de la programación, puesto que permite al estudiante interactuar con el código y verificar de manera gráfica el resultado de la ejecución. Este aspecto es muy importante si tratamos de reorientar el estilo clásico de enseñanza centrado en el profesor a un estilo centrado en el alumno.

La utilización de la animación en la enseñanza de algoritmos no es nueva; de hecho comienza hace dos décadas, cuando el profesor Ronald Baecker de la Universidad de Toronto creó el video *Sorting Out Sorting*. Los primeros sistemas de animación surgen entre 1980 y 1990: Estos sistemas fueron denominados BALSA-I [2] (Brow ALgorithm Simulator and Animator) y TANGO [3] (Transition-based ANimation GeneratiOn). El primero de ellos, soportaba múltiples vistas simultáneas de las estructuras de datos y permitía visualizar la ejecución simultánea de distintos algoritmos. Sirvió como acicate a otros investigadores para participar en el desarrollo de otros sistemas. El segundo de ellos, desarrollado por John Stasko en la Universidad de Brown, supuso la introducción del paradigma "path-transition" en el diseño de la animación: se abstraen las instrucciones de dibujo para permitir un modelado más cómodo de las animaciones de los algoritmos. Su aportación más importante fue el establecimiento de un marco conceptual para los sistemas de animación de algoritmos, desde entonces adoptado por muchos como arquitectura fundamental.

A BALSA-I le siguió, BALSA-II, un sistema que manipula imágenes con múltiples vistas y provee una facilidad de "scripting" para la creación de animaciones. TANGO, por otro lado, tiene numerosos sistemas descendientes, como POLKA y su "front-end", Samba [4]. POLKA es un sistema multipropósito especialmente diseñado para construir animaciones concurrentes de programas en paralelo. Inicialmente en 2D orientado a objetos, ha sido extendido a 3D, Samba es un intérprete interactivo que lee

comandos ASCII y lleva a cabo las correspondientes acciones de animación. Completan la nómina de programas otras contribuciones como Zeus [5], primer sistema de visualización interactivo que permite animar programas paralelos, y Leonardo [6], entorno integrado para el desarrollo y animación de programas en C, que proporciona un extenso control del usuario sobre la animación.

Frente a los anteriores, con la llegada de las nuevas tecnologías, la popularidad de Internet y la evolución del lenguaje de programación Java, los desarrolladores han pasado a construir sistemas de animación de algoritmos online, independientes de la plataforma, y de código abierto, no limitados en su uso al aprendizaje en aulas o laboratorios sino extensivos al aprendizaje a distancia. Catai [7] es un sistema de animación que anima programas en C++ y permite a varios usuarios compartir la misma animación a través de la abstracción de un aula virtual. Mocha [8] es un modelo distribuido con una arquitectura cliente-servidor que particiona de un modo óptimo los componentes software de un típico sistema de animación de algoritmos. Sólo se exporta a la máquina del usuario el código para la interfaz gráfica, mientras que el algoritmo se ejecuta en el servidor. Más reciente y completo es EDApplets [9], un conjunto de Applets Java que realizan animaciones clásicas de estructuras de datos y algoritmos, efectuando trazas y ejecuciones paso a paso sobre el código, y permitiendo interactuar al alumno.

La herramienta que se presenta en este artículo integra bajo un mismo Applet distintas animaciones de estructuras de datos abstractas como las listas, pilas, colas y árboles binarios, todos ellos con implementación dinámica. El alumno tiene a su disposición un conjunto completo de operaciones para el manejo de la estructura: creación y destrucción e inserción, eliminación y búsqueda de datos. Puede interactuar totalmente con las operaciones, modificando su comportamiento a través de los datos de entrada. En caso de error, la aplicación informa del motivo, bien por introducir incorrectamente alguno de los parámetros, bien por aplicar incorrectamente la operación. El diseño de la herramienta permite, a través de un lenguaje simple, crear y animar fácilmente elementos gráficos y sincronizarlos con la sentencia de código que se ejecuta, permitiendo de esta forma su extensibilidad.

Los estudios empíricos llevados a cabo sobre el hipotético beneficio pedagógico de la utilización de la animación de algoritmos en la docencia han arrojado resultados favorables, si bien en todos se destaca la necesidad de combinarla con otros materiales de aprendizaje. A modo de resumen, éstas serían las conclusiones más sobresalientes:

- Brown y Sedgewick [2]: la animación como complemento a la clase tradicional reduce el tiempo necesario para la comprensión.
- Lawrence [10]: permitir a los estudiantes controlar e interactuar con la animación y disponer de ella fuera de las aulas, de manera que utilicen sus propios conjuntos de datos para el algoritmo en lugar de observar conjuntos de datos preparados, contribuye a un mayor entendimiento.
- Kehoe [11]: las animaciones de algoritmos adquieren mayor efectividad si se usan en situaciones de aprendizaje abiertas e interactivas (como los ejercicios de clase). La comprensión del algoritmo se hace más accesible y menos intimidante.

En resumen, la literatura parece apoyar la utilización de estas técnicas como apoyo docente en lo que respecta al aprendizaje de algoritmos y estructuras de datos.

2. Contexto

La materia Estructura de Datos y de la Información (EDI) es una asignatura troncal cuatrimestral del plan de estudios de las Ingenierías Técnicas en Informática de Gestión y Sistemas y de la Ingeniería Informática de la Facultad de Informática de A Coruña. La asignatura se imparte en el primer curso del segundo cuatrimestre y tiene asignados un total de 6 créditos, 4.5 de teoría y 1.5 de prácticas, lo que se traduce en 3 horas teóricas y 1 hora práctica semanal. Los descriptores propios son:

- Estructura de datos y algoritmo de manipulación.
- Tipos abstractos de datos

Aunque no existe ningún requisito previo normativo para poder cursar esta materia, se recomienda tener conocimientos básicos de la

asignatura Programación, cursada en el cuatrimestre previo.

Actualmente, las técnicas en el campo de las estructura de datos van en la línea de diferenciar las "especificaciones" de una estructura de datos, es decir, su comportamiento, de sus "implementaciones", en el ámbito de las disponibilidades de un lenguaje de programación específico. La clave, tanto en el diseño de estructuras como de algoritmos que las manipulen es el concepto de "abstracción". Utilizando el concepto de abstracción, podemos definir las diferentes estructuras "olvidando" los detalles de implementación de los objetos, tales como su representación en la memoria o los algoritmos de manejo, concentrándonos en especificar sus valores y sus operaciones permitidas, que es lo que se conoce como "Tipos Abstractos de Datos".

Dentro del plan de estudios mencionado, la asignatura se podría englobar en el bloque temático de asignaturas relacionadas con la Ingeniería del Software, dentro del cual esta asignatura constituye una disciplina totalmente básica. En este grupo, las relaciones más estrechas se establecen con Bases de Datos I, Algoritmos, Metodología de la Programación y Programación Orientada a Objetos.

2.1. Objetivos generales de EDI

- Conocer los mecanismos de abstracción y valorar el papel importante que juega la abstracción en general, y la de datos en particular, como herramienta fundamental en la metodología de construcción de programas.
- Comprender el concepto de Tipo Abstracto de Datos (TAD).
- Comprender la necesidad de separación entre los niveles de especificación, implementación y uso de un TAD.
- Conocer, desde el punto de vista de la abstracción de datos, las estructuras de datos básicas que se utilizan en programación, sus realizaciones más comunes, sus algoritmos de manipulación y su utilidad en aplicaciones básicas realistas.
- Familiarizarse con el manejo de estructuras de datos básicas y desarrollar la capacidad de crear otras más complejas usando las primeras como referencia.
- Analizar la estructura de datos apropiada para la resolución de un problema y valorar la solución en cuanto a eficiencia y coste.
- Desarrollar la habilidad para definir y analizar alternativas de representación.
- Desarrollar implementaciones de las estructuras estudiadas utilizando un lenguaje de programación imperativo de alto nivel.
- Valorar como buenas prácticas de programación el desarrollo de código estructurado, modular y bien documentado.

2.2. Contenidos teóricos

El temario de teoría que se imparte en la asignatura es el siguiente:

1. Gestión dinámica de la memoria
2. Recursividad
3. Introducción a los TAD
4. Estructuras de datos lineales: listas, pilas y colas
5. Estructuras de datos no lineales: árboles binarios y árboles de búsqueda

Para el desarrollo de las clases teóricas se sigue un patrón expositivo magistral de descripción de contenidos temáticos, implementando en un lenguaje de programación como PASCAL las operaciones básicas de manejo de las distintas estructuras de datos. En estas clases presenciales el profesor involucra en el desarrollo práctico a los alumnos, que colaboran por grupos en el proceso de solución. Además, se plantean distintos ejercicios de aplicación de estructuras de datos a problemas reales. En suma, la orientación de la asignatura es fundamentalmente práctica, lo cual también se refleja en el proceso de evaluación del alumno.

2.3. Contenidos prácticos

Para evaluar la capacidad del alumno en la aplicación práctica de tipos abstractos de datos a problemas reales se plantean dos supuestos prácticos obligatorios que los alumnos deben resolver por parejas. El profesor supervisará el trabajo durante las distintas sesiones, solucionando dudas y corrigiendo errores de interpretación, malos hábitos de programación, etc. Los objetivos docentes a alcanzar a través del programa de prácticas se resumen en:

1. Aumentar el dominio técnico en la tarea de programación.

2. Consolidación de los conceptos de abstracción a través de la resolución práctica de supuestos reales.
3. Mejorar la capacidad analítica para seleccionar la estructura de datos adecuada en la solución software adoptada, valorando y criticando las distintas aproximaciones.
4. Desarrollar y/o potenciar aptitudes de trabajo en equipo.

2.4. Evaluación

La evaluación de alumno tiene en cuenta la valoración de las prácticas obligatorias y la realización de un examen final escrito al término del cuatrimestre, de carácter eminentemente práctico para que pueda demostrar que ha adquirido los conocimientos necesarios de abstracción y diseño de TADs y se ha entrenado lo suficiente como para poseer las habilidades precisas para resolver supuestos. La nota final tiene en cuenta en un 90% la nota del examen escrito y en un 10% la nota de las prácticas obligatorias.

3. Motivación

Cualquier plan de estudios de informática contempla asignaturas relacionadas con estructuras de datos y algoritmos de manipulación. El estudiantado novel debe adquirir en poco tiempo capacidades conceptuales de algoritmos y estructuras de datos, capacidades procedimentales para especificar, diseñar y programar estructuras de datos y ser capaces de desarrollar el pensamiento creativo para resolver problemas reales. Así pues, el alumno se encuentra con un proceso abstracto y dinámico, como es un algoritmo, difícil de comprender aún cuando posean el seudocódigo que describa la lógica interna y consulten la bibliografía seleccionada.

El proceso instructivo tradicional ha sido siempre unidireccional: el docente utiliza distintos recursos (pizarra, transparencias,…) para presentar la materia y la única interacción con el estudiante se limita a discusiones verbales. En muchos casos, el profesor se limita a escribir fragmentos de código e ilustrar gráficamente el modo en que dichos fragmentos se comportan, pero no es posible reflejar el comportamiento de las distintas operaciones y los cambios que se producen en las estructuras de datos en virtud de su ejecución. Esto lo proporciona la animación de algoritmos, una combinación de texto y gráficos con la cual el estudiante puede experimentar la lógica interna del algoritmo en el laboratorio y por extensión fuera del aula, fijándose su propio tiempo de estudio. Con este tipo de recurso docente se pueden alcanzar valores pedagógicos óptimos, siempre y cuando el sistema de animación posea las siguientes características:

1. Usabilidad, referida a la capacidad del sistema para ser comprendido, aprendido, usado y ser atractivo para el usuario de modo que la curva de aprendizaje sea mínima.
2. Multipropósito, en el sentido de no limitarse exclusivamente a la animación de algoritmos y estructuras de datos.
3. Acceso abierto al sistema en el centro educativo, o a través de Internet.
4. Independencia de la plataforma.
5. Resaltado de sintaxis del código fuente, destacando de alguna forma las líneas del algoritmo que se ejecutan a la vez que la transcurre la animación.
6. Control de la animación, avanzando paso a paso, deshaciendo, retrocediendo o avanzando rápidamente hacia cualquier estado deseado.
7. Interactividad entre los usuarios y el sistema.
8. Historia para habilitar el acceso a estados previos de la animación.
9. Retroalimentación de los estudiantes para evaluar la efectividad del sistema con el fin de mejorarlo.

4. Requisitos funcionales del sistema

La siguiente lista muestra los requisitos funcionales que la herramienta debería de satisfacer:

- Mantener información acerca de la codificación de las estructuras y de las operaciones sobre las estructuras de datos (codificación)
- Detectar condiciones de error en operaciones
- Procesar las operaciones y generar la animación de la estructura correspondiente
- Controlar la velocidad de la animación
- Resaltar líneas de los algoritmos durante la animación
- Animar los nodos de las estructuras
- Mantener apuntadores gráficos entre los nodos

- Disponer variables auxiliares usadas en los algoritmos en la pantalla
- Mantener el estado interno de las estructuras de datos que se animan
- Detectar entradas de datos anómalos
- Soportar internacionalización

5. Arquitectura del sistema

En 1990, John Stasko publicó un informe en el que introdujo un nuevo modelo para desarrollar los sistemas de animación de algoritmos [3]. "La animación de algoritmos abarca la animación del programa y la estructura de datos que interviene, lo que implica típicamente una relación uno a uno entre los datos del programa y las imágenes de la animación…Con frecuencia, las imágenes de la animación no tienen ninguna correspondencia directa a los datos del programa o a las unidades de la ejecución, sino que por el contrario representan las abstracciones diseñadas para aclarar la semántica del programa…Los diseñadores de la animación no deberían tener que crear una rutina distinta de animación para cada configuración posible del programa. Por lo tanto, la animación de algoritmos requiere modelos que soporten la creación de rutinas de animación que puedan adaptarse a varios estados del programa." De este modo, su informe manifiesta la imperiosa necesidad de destruir el acoplamiento entre las rutinas que modelan la animación y el programa que se quiere animar.

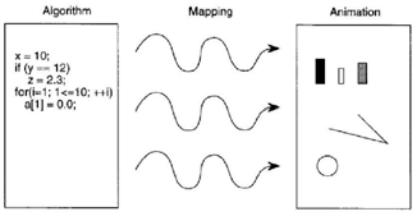


Figura 1. Marco conceptual definido por Stasko

El objetivo del marco conceptual es desarrollar un sistema limpio, de gran alcance y flexible. La arquitectura propuesta consta de tres componentes como se muestra en la Figura 1: algoritmo, animación y "mappings". El componente algoritmo sirve para conducir o activar la animación usando un enfoque conducido por eventos: se identifican las posiciones en el código que son importantes en la semántica del programa, y se modelan como llamadas a funciones con parámetros; a medida que se van alcanzando dichos puntos durante la ejecución del programa, su nombre y parámetros se pasan al componente de animación. El componente de animación contiene los objetos gráficos que cambian la posición, el tamaño, y el color en la animación, y las operaciones que controlan los cambios para simular una acción. El componente "mappings" contiene la traducción del programa y de su ejecución en la animación. Una porción de este componente implica la construcción de un mecanismo general de almacenamiento y otro de recuperación de objetos, que permite a los diseñadores conectar un objeto de datos que participa en la animación con un conjunto de parámetros recibidos del programa.

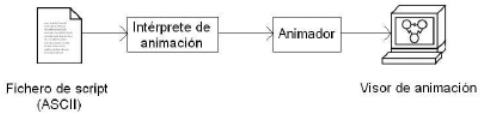


Figura 2. Arquitectura de los sistemas de animación de algoritmos

Desde la publicación del citado informe, muchos sistemas de animación de algoritmos han adoptado el marco conceptual propuesto por Stasko. La arquitectura genérica constaría de tres componentes; un animador, un intérprete de animación y un visor de la animación (Figura 2). El animador contiene librerías con funciones de representación gráfica y animación de los objetos, en los dispositivos gráficos. El visor de la animación provee un entorno gráfico para mostrar la animación a los usuarios. El intérprete de animación actúa como un canal de comunicación entre el sistema de animación y el usuario final. Las animaciones se modelan a través de comandos de script, insertados en los puntos clave del programa que se desea animar. El intérprete traduce estos comandos y pasa los parámetros de control del objeto que se quiere animar al animador. Finalmente el animador dibuja el objeto animado de acuerdo a los parámetros de control en el visor de animación. Estos parámetros son las coordenadas x e y donde aparecerá el objeto, su color, forma, etc.

Esta arquitectura fue adaptada y ampliada para construir el sistema de animación interactivo. A

partir de sistema de animación existente como JAWAA [12], se planteó un modelo como el de la Figura 3, compuesto fundamentalmente por un animador de objetos complejos, un generador de animaciones de algoritmos, una interfaz gráfica de usuario y el subsistema de animación JAWAA.

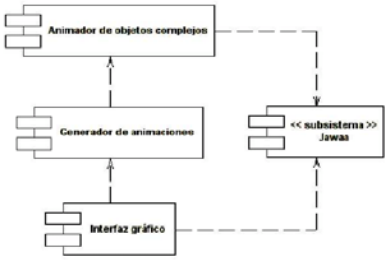


Figura 3. Arquitectura del sistema tutor

5.1. Animador de objetos complejos

El animador de objetos complejos contiene básicamente dos tipos de componentes: los objetos gráficos que intervienen en la animación y la coreografía de la misma.

Los objetos gráficos son aquellos que experimentan cambios en la localización, tamaño, color, etc. a través de los "frames" de la animación. Así pues, se han desarrollado en este componente los objetos que representan los nodos de las estructuras, los enlaces entre ellos y el algoritmo que se intenta animar (operación de manejo de la estructura de datos). Todos ellos están compuestos a su vez de otros objetos gráficos más simples que ofrece el subsistema JAWAA. Diversas operaciones estándar permiten cambiar el color, el tamaño, la posición, de los objetos gráficos, etc. A mayores se diseñaron operaciones para establecer apuntadores entre nodos, destacar la línea de código del algoritmo que se ejecuta, etc.

La coreografía guarda los cambios que se deben producir en los atributos de los objetos gráficos del subsistema JAWAA entre un "frame" y otro, información acerca de las pausas en la animación y de la agrupación de efectos animados sobre los objetos gráficos que se deben realizar en cada paso de animación.

5.2. Generador de animaciones

El generador de animaciones está compuesto por un conjunto de clases Coreógrafo que permiten modelar las animaciones correspondientes a la ejecución de los algoritmos sobre las estructuras de datos. Antes de modelar la animación se encarga de detectar condiciones de error debido a la introducción incorrecta de parámetros o a la violación de las precondiciones del algoritmo. Si la ejecución es exitosa añade los objetos gráficos a la primera secuencia de animación e invoca operaciones sobre ellos para conducir y transformar los objetos gráficos a través de las secuencias de animación posteriores. Además necesita mantener el estado de las estructuras de datos para lograr que las animaciones sean dinámicas y que los usuarios obtengan animaciones diferentes tras la ejecución de los algoritmos. También se encarga de proporcionar información acerca de la implementación de los algoritmos y las estructuras de datos.

5.3. Subsistema JAWAA

El subsistema JAWAA se encarga de proveer la visualización multimedia de la animación. Para lograr dicho objetivo, interpretará las secuencias de animación especificadas por el componente Animador de objetos complejos.

5.4. Interfaz gráfica

El componente interfaz gráfica es el responsable de gestionar, por un lado, la interacción entre el usuario y el componente generador de animaciones, y por otro, la interacción entre el usuario y el subsistema JAWAA. Así pues permite la construcción de la animación correspondiente a la ejecución del algoritmo seleccionado por el usuario, mostrando las condiciones de error que se producen, si procede, y activando la animación si la ejecución es satisfactoria. Además, se encarga de gestionar los eventos que producen llamadas desde/hacia el subsistema JAWAA para reproducir, parar, pausar, reanudar, aumentar o disminuir la velocidad y visualizar mediante pequeños pasos la reproducción de la animación.

6. Implementación

El sistema tutor interactivo está programado en Java porque: a) está especialmente diseñado para programar en Internet y permite transmitir, mediante el protocolo HTTP, programas autoejecutables e independientes de la plataforma;

b) verifica su código al mismo tiempo que se escribe, y una vez más antes de ejecutarse; c) hace posible la incorporación de aplicaciones interactivas y especializadas, y aporta la posibilidad de distribuir contenidos ejecutables al ordenador del usuario (Applet); d) no permite realizar llamadas a funciones globales ni acceder a recursos arbitrarios del sistema, lo que aumenta la seguridad.

Como sistema básico de animación se utilizó la herramienta de animación JAWAA, que posee la misma arquitectura que la mostrada en la Figura 2. Recordemos que los comandos que modelan las animaciones se almacenan en un fichero de script, de modo que cuando se desea visualizar una animación JAWAA interpretaría el contenido de dicho fichero. Para hacer uso de JAWAA, necesitaríamos generar un fichero de script cada vez que el usuario quiere probar el funcionamiento de un algoritmo sobre una estructura de datos. Sin embargo, una de las restricciones de seguridad que impone el uso de Applets atañe a las funciones de entrada/salida a disco, puesto que no es posible utilizar ficheros en la máquina del usuario a menos que éste lo permita. De esta forma era necesario implementar un mecanismo más flexible para generar el script que modela cada animación. Para ello se añadió un método, que se encarga de leer los comandos de script desde un vector. Este nuevo planteamiento de colaboración entre componentes nos permitirá construir animaciones sin tener que escribir ficheros a disco para su posterior lectura en el intérprete de JAWAA (Figura 4).

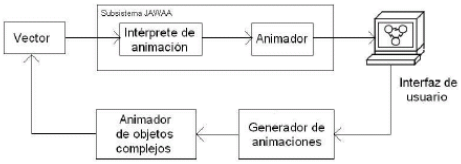


Figura 4. Modificación de la herramienta JAWAA

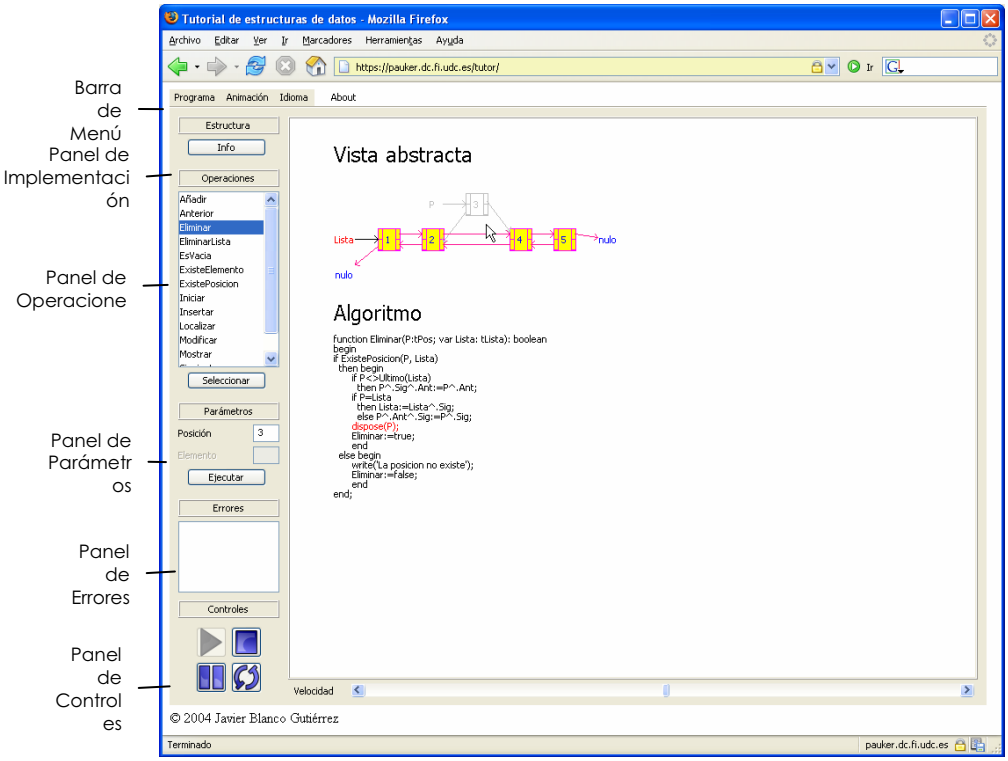


Figura 5. Interfaz de usuario del sistema tutor

La interfaz gráfica de la aplicación (Figura 5) se divide en cuanto a cometido en:

- Barra de menú: Menú principal
- Panel de implementación: Informa acerca de la implementación de las estructuras de datos
- Panel de operaciones: Muestra las distintas operaciones disponibles para cada estructura de datos y permite obtener información de cada una de ellas
- Panel de parámetros: Permite ejecutar el algoritmo seleccionado. Ofrece si se necesitan campos de texto para introducir parámetros
- Panel de errores: Muestra errores de ejecución relativos a la violación de precondiciones del algoritmo o a la introducción errónea de los parámetros
- Panel de controles: Reproducir, parar, pausar/reanudar y visualizar paso a paso
- Panel de velocidad: Ajuste de la velocidad
- Panel de animación: Expone información acerca de la implementación de las estructuras y algoritmos y anima su ejecución.

7. Conclusiones

La animación de algoritmos es un recurso didáctico muy importante como apoyo docente en los primeros cursos de las titulaciones de informática. El desarrollo de herramientas que apoyen al estudiante en el estudio de algoritmos y estructuras de datos combinando adecuadamente gráficos y texto ha sido el motivo de desarrollo de la herramienta que se ha presentado. El tutor interactivo de estructuras de datos permite que el estudiante refuerce fuera del aula los conocimientos adquiridos durante las clases presenciales en base a un mecanismo de animación de algoritmos. El alumno experimenta la ejecución paso a paso de cualquier operación sobre la estructura, modificando a su antojo las condiciones iniciales. Si la ejecución es incorrecta se le informa y tiene la posibilidad de corregirla. La herramienta ha sido implementada en Java y se sirve como un sencillo Applet.

Referencias

[1] A.J. Sánchez-Ruiz y A. Pereira. Algunos criterios para la construcción de animaciones de algoritmos con propósitos pedagógicos. En CoRIBIE 98 IV Congresso da Rede Iberoamericana de Informática Educativa Brasilia Brasil, Octubre 1998. http://www.unf.edu/~asanchez/papers/

[2] M.H. Brown and R. Sedgewick. A system for algorithm animation. Proceedings of the 11th annual conference on Computer graphics and interactive techniques, págs. 177–186, Julio 1984. http://portal.acm.org/

[3] J.T. Stasko. Tango: A framework and system for algorithm animation, Septiembre 1990. http://portal.acm.org/

[4] POLKA, Samba, xTANGO, JSamba. http://www.cc.gatech.edu/gvu/softviz/algoanim/samba.html

[5] M.H. Brown. Zeus: A system for algorithm animation and multi-view editing. IEEE Workshop on Visual Languages, págs. 4–9, 1991. http://research.compaq.com/SRC/zeus/biblio.html

[6] P. Crescenzi, C. Demetrescu, I. Finocchi, and R. Petreschi. Leonardo: a software visualization system. Proceedings WAE'97, pages 146–155, 1997. http://citeseer.nj.nec.com/crescenzi97leonardo.html

[7] G.F. Italiano, G.Cattaneo, U. Ferraro, and V. Scarano. CATAI: Concurrent Algorithms and data Types Animation over the Internet. Proceedings of 15th IFIP World Computer Congress, Vienna, 1998. http://catai.dia.unisa.it/

[8] J.E. Baker, I.F. Cruz, G. Liotta, and R. Tamassia. The Mocha Algorithm Animation System. Proceedings of International Workshop on Advanced Visual Interfaces (AVI '96), págs 248–250, 1996. http://citeseer.nj.nec.com/baker96mocha.html

[9] F. Almeida, V. Blanco, L. M. Moreno, EDApplets: Estructuras de datos y Técnicas Algorítmicas en la Web, Libro de actas X Jornadas de Enseñanza Universaria de la Informática (JENUI), 2004.

[10] A. Lawrence, A. Badre, and J.T. Stasko. Empirically evaluating the use of animations to teach algorithms. Proceedings of the 1994 IEEE Symposium on Visual Languages, St. Louis, MO, págs 48–54, 1994. http://citeseer.nj.nec.com/lawrence94empirically.html

[11] C.M. Kehoe, J.T. Stasko, and A. Talor. Rethinking the evaluation of algorithm animations as learning aids: an observational study. International Journal of Human Computer Studies, 54(2):265–284, 2001. http://citeseer.nj.nec.com/kehoe99rethinking.html.

[12] W. Pierson and S.H. Rodger. Web-based animation of data structures using JAWAA. In Twenty-ninth SIGCSE Technical Symposium on Computer Science Education, 1998. http://citeseer.nj.nec.com/78398.html

Calidad y evaluación de la docencia

Doce Propuestas y una Reflexión

Rosana Satorre[1], Faraón Llorens[1], Pere Palmer[2], Joe Miró[2]

[1]Dto Ciencia de la Computación e Inteligencia Artificial
Universidad de Alicante
03080 Alicante
{rosana, faraon}@dccia.ua.es

[2]Dpt. Mat. i Informàtica
Universitat de les Illes Balears
Campus UIB
07122, Palma de Mallorca
{pere.palmer, joe.miro}@uib.es

Resumen

Últimamente vivimos inmersos en una atmósfera de cambio en la universidad española. El ministerio, las consejerías autonómicas y el Jenui han generado muchos documentos reflejando esta situación. En esta ponencia recogemos doce propuestas realistas –aunque no necesariamente fácilmente realizables– que se pueden poner en práctica por profesores y direcciones de centro sin necesidad de esperar a directivas externas. La intención de estas propuestas no es dar consignas de qué hacer, sino estimular un debate que genere métodos e iniciativas específicas que mejoren y hagan más competitivas nuestras universidades. Acabamos con una reflexión, que es más general que las propuestas, y que tiene la intención de que pensemos en cómo obtener un reconocimiento por el trabajo que dedicamos a la investigación de la labor docente.

1. Introducción

La universidad española necesita un cambio y muchos están esperando a que los responsables nos indiquen el camino. Ahora, el Espacio Europeo de Educación Superior parece que venga a resolvernos las cosas: sólo tenemos que esperar a que los expertos del Ministerio nos digan qué tenemos que hacer y cómo debemos hacerlo. Pero nosotros consideramos que no debemos esperar, que debemos tomar la iniciativa en este cambio. Es cierto que hay directivas ministeriales y documentos como los libros blancos que nos marcan el *contexto curricular* (el *qué enseñar*), y que los estatutos, las directivas de nuestras universidades y los medios de los que disponemos nos marcan el *contexto institucional* (el *dónde enseñar*) en el que forzosamente nos debemos desenvolver, pero la manera de desarrollar nuestra labor en este nuevo marco, el *contexto de aprendizaje* (el *cómo enseñar*), está completamente en nuestras manos.

Es importante que los profesores de a pie, los que estamos cada día delante de los alumnos, tomemos la iniciativa en marcar cómo vamos a enseñar, cómo debe ser el aprendizaje a partir de ahora. El método de enseñanza es, según uno de los últimos documentos de trabajo del Ministerio [7], parte de la organización de los estudios, de los planes de estudios. Si tomamos la iniciativa, en el mejor de los casos no nos veremos obligados a seguir recetas impuestas por personas más preocupadas por la administración que la docencia, y en el mejor de los casos influiremos en ellos y conseguiremos unos mejores contextos donde llevar a cabo nuestra labor docente.

Hemos querido plantear doce propuestas para mejorar el contexto de aprendizaje. Estas propuestas son concretas, realistas y viables – aunque su realización en algunos casos puede ser dificultosa– y se pueden poner en práctica por profesores y direcciones de centro sin necesidad de esperar a directivas de rectorados y ministerios. Ideamos estas propuestas no para dar consignas de qué hacer, sino para estimular un debate en los centros, grupos docentes y en las conciencias de los profesores que genere métodos e iniciativas específicas que mejoren y hagan más competitivas nuestras universidades. Nacieron de una exposición hecha por los dos primeros autores en el seminario de docencia del Dep. de Matemáticas e Informática de la UIB. A partir del documento inicial y el debate posterior que se prolongó durante varias sesiones del seminario, se han recogido y resumido las ideas principales expuestas. Aunque al final hemos dado forma a la ponencia los cuatro que aparecemos como autores, esta es una obra coral de profesores de informática de dos universidades.

Acabamos estas propuestas con una reflexión que va más allá de una ponencia, y que pretende hacernos conjugar nuestro amor y dedicación a la docencia con las exigencias y reglas que rigen la universidad española en estos momentos. Una

reflexión sobre qué debemos hacer para no tener que contraponer la docencia con la investigación.

2. Doce propuestas

2.1. Estudiemos con cuidado el proceso de evaluación.

La evaluación es una pieza clave en el proceso educativo: sin un mecanismo de evaluación adecuado de nada servirán nuestros esfuerzos de mejora en la calidad. Hay que tener muy presente que la evaluación es el referente para el estudiante, ya que sus esfuerzos se orientan principalmente hacia la superación de estas pruebas. Escogiendo adecuadamente las pruebas de evaluación podemos conseguir un mejor aprovechamiento por parte del alumno.

El esfuerzo del profesor consistirá en establecer mecanismos de evaluación tanto del proceso de enseñanza–aprendizaje, como del correcto aprendizaje de los alumnos y, por supuesto, de la adecuación de nuestro propio trabajo.

La evaluación debe responder a cinco cuestiones de capital importancia:

- ¿Quién evalúa?
- ¿Qué se evalúa?
- ¿Cómo se evalúa?
- ¿Cuándo se evalúa?
- ¿Para qué se evalúa?

De la respuesta a estas cinco cuestiones obtendremos diferentes tipos de evaluación. Existen muchos tipos y cada uno es adecuado para determinado momento del proceso u objetivo a alcanzar. Un mecanismo de evaluación inadecuado convertirá todo esfuerzo educativo, tanto el realizado por parte del profesor como por parte del alumno, en estéril.

2.2. Reconozcamos que el estudiante es un recurso limitado y compartido.

Nadie mejor que nosotros los informáticos para entender este concepto. Los estudiantes disponen de un tiempo limitado y deben atender a distintas asignaturas (además de su vida personal). Debemos ser conscientes del tiempo que deben dedicar a nuestra asignatura y pedirles el esfuerzo correspondiente, ni más, ni menos. Si les pedimos más horas, estas serán a costa de quitárselas a su vida personal, a que no las dediquen a las otras asignaturas o a que la duración de las carreras sea superior a la oficial. Y ninguna de las tres alternativas nos parece sensata.

Entendemos que es un difícil equilibrio conseguir que el alumno haga el trabajo que le corresponde, ni más ni menos. Ya hemos dicho que no es adecuado sobrecargar al alumno con excesivo trabajo, pero tampoco lo es darle demasiado poco: la asignatura se convierte en una *maría* y el alumno le pierde el respeto. Este equilibrio debe conseguirse con una buena coordinación entre profesores –que como problema humano que es, es de difícil solución y sin recetas– y entre profesor y alumno: el profesor debe ser consciente de la cantidad de trabajo que le corresponde hacer al alumno, y el alumno debe ser igualmente consciente. Si una tarea, típicamente un ejercicio de programación, está requiriendo al alumno una cantidad desorbitada de tiempo, debe parar y consultar al profesor y no dedicar hora tras hora a encontrar ese pequeño error.

Otra cuestión es la de los picos de trabajo. Las asignaturas con prácticas suelen requerir más horas al final de curso que al principio. No pueden iniciarse las prácticas al principio ya que el alumno aún no tiene los conocimientos necesarios, pero si hay varias asignaturas de este estilo, la demanda de trabajo al final de curso se vuelve intolerable. La coordinación entre profesores puede ayudar, pero hoy en día no todos los alumnos realizan las mismas asignaturas y es difícil saber con quién te tienes que coordinar. De aquí que creemos que la demanda de trabajo al alumno debe ser lo más plana posible. El profesor debe usar su ingenio para ir creando la práctica a lo largo del curso y no sólo al final, o deben crearse asignaturas de prácticas, independientes de las de teoría que permitan al alumno empezar a trabajar en la práctica desde el primer día.

2.3. Debemos coordinarnos con la entrada.

No podemos ignorar cuáles son los conocimientos que deben tener los alumnos de nuevo ingreso en la Universidad. En general, recordamos lo que se nos exigía a nosotros y después ponemos el grito en el cielo ante la 'ignorancia' de la juventud de hoy en día, sin saber –ni importarnos– que la materia causa de nuestra indignación jamás se les haya impartido.

Es importante distinguir entre dos problemas diferentes. El primero es que consideremos que los conocimientos y habilidades que se exige a los estudiantes de bachillerato y de formación profesional en los institutos sean pobres y no vayan en consonancia con lo que se les debe exigir a futuros alumnos universitarios. Este es un problema que como ciudadanos nos puede preocupar y en el que podemos involucrarnos, pero como profesores de universidad no es un asunto que nos concierne. A diferencia de otros países, no es la universidad la que decide el conocimiento de los alumnos que ingresan, sino la sociedad. Como servidores públicos (y muchos de nosotros, empleados públicos) es nuestra obligación conocer el nivel que la sociedad exige al alumno universitario de entrada y educarle partiendo de estos conocimientos.

Otro problema diferente es el caso del alumno que ni siquiera ha adquirido los conocimientos exigidos. Podríamos desentendernos de ellos, pero eso es una pérdida de recursos humanos, e incluso cruel: si por lo que sea un joven no ha podido tener una buena educación hasta el momento, le debemos la oportunidad de obtenerla. Naturalmente, no es cuestión de convertir primero de carrera en segundo de bachillerato–bis. Hay otros medios: se pueden preparar apuntes con los conocimientos previos necesarios –y que se supone saben– de cada asignatura y ponerlos a su disposición; se puede, como ya hacen varias universidades, entre ellas la UIB y la UA, crear unas *asignaturas cero* que permitan al alumno remediar las carencias que tiene antes de empezar el curso; también se puede dedicar una pequeña porción del curso –dos semanas por ejemplo– a hacer un repaso rápido para todos de los conocimientos previos necesarios, que siempre será beneficioso y nos ayudará a aumentar el ritmo más adelante.

Hay muchas soluciones, y cualquiera de ellas es mejor que el abandonar la esperanza de poder educar a un joven.

2.4. Debemos coordinarnos con la salida

Estamos formando profesionales altamente especializados que deben incorporarse a un mercado laboral y no podemos vivir de espaldas a esta realidad: es necesario conocer qué es lo que la sociedad demanda y en qué medida nuestros alumnos son capaces de integrarse en esa sociedad y aportar algo positivo. Debemos escuchar a esta sociedad y entender cuáles son sus necesidades. De esta forma podremos intentar preparar a nuestros alumnos adecuadamente.

Varios son los agentes involucrados en esta tarea. Por una parte están los propios egresados. De ellos podemos conocer su grado de satisfacción con la formación recibida, en qué medida sienten que su formación les adecúa para poder desempeñar su profesión, qué aspectos parecen poco adecuados y, en definitiva, cuál ha sido el valor de formación desde la perspectiva del que la ha recibido. Otro de los agentes a tener en cuenta es la empresa en sentido amplio, es decir tanto la pública como la privada, tanto las pequeñas, medianas como las grandes empresas que son las que absorberán a nuestros estudiantes y para las que los preparamos. Ya hay estudios [1] que podemos consultar y que indican algunas de las deficiencias que las empresas opinan tienen nuestros egresados. Una tercera fuente de información a la que debemos consultar son los organismos colegiados que representan a las diferentes partes, desde organizaciones empresariales a colegios profesionales.

Si bien es cierto que pueden aparecer reticencias en algunos sectores, no es menos cierto que resulta tremendamente atractivo para cualquier tipo de empresa el saber que los futuros egresados tendrán un perfil adecuado a las necesidades que ellos planteen, al menos en parte. Esto será así siempre y cuando estos agentes puedan comprobar que sus propuestas no caen en saco roto. Es cierto que no debemos simplemente hacer lo que los agentes externos quieren. Esto es prácticamente imposible ya que cada uno de estos agentes aportará una visión diferente. Es nuestra labor saber qué es lo que necesitan y combinarlo con nuestro conocimiento y experiencia para poder llevar al aula unos planes formativos que mejoren la formación de nuestros alumnos.

2.5. Debemos coordinarnos entre nosotros

Otro gran problema, y vergüenza, de la universidad española es la fragmentación: titulaciones, departamentos, cursos, asignaturas, temas... todas ellas independientes, en general sin coordinación alguna. Los temarios de las asignaturas repiten algunos conocimientos y omiten otros; los métodos pedagógicos más que diferentes, son distantes. Incluso en asignaturas compartidas por dos o más profesores, cada uno hace lo que quiere, para confusión y desengaño de los alumnos. Los profe-

sores que intervenimos en la formación de un estudiante debemos coordinarnos. Con la atomización se pierde la perspectiva global y se pierde eficiencia. Los conocimientos generales, por ejemplo la capacidad de comunicación o el trabajo en equipo, deben usarse en todas las asignaturas y de forma similar. No puede ser que al alumno se le exija trabajar de una manera ahora y de otra completamente distinta dentro de una hora. Es una pérdida de esfuerzo por parte de profesores y alumnos. Es por ello bueno que de cuando en cuando levantemos la cabeza de nuestros papeles y miremos alrededor. Las Comisiones Docentes deben ser más útiles y efectivas.

La coordinación se puede plantear de varias formas diferentes, no excluyentes entre sí. Por un lado debería existir una coordinación entre los profesores que imparten una misma materia. También entre profesores que imparten materias diferentes a un mismo grupo de alumnos. Otras posibilidades son establecer mecanismos de coordinación entre profesores que impartan materias relacionadas, coordinaciones transversales, etc. De todas formas ningún esfuerzo de coordinación será positivo si los que debemos coordinarnos no nos adaptamos a las premisas establecidas en las Comisiones Docentes. Este compromiso es, pues, imprescindible.

Otro aspecto importante es que dichas Comisiones han de ser operativas, es decir, no deben consistir en meras reuniones autocomplacientes en donde los reunidos se dedican a alabarse a sí mismos por lo bien que lo hacen. Se trata más bien de lo contrario: hay que buscar debilidades en el modelo educativo propuesto y estudiado en cada caso. Hay que ser crítico y, por supuesto, hay que querer mejorar.

2.6. Creemos especialistas de didáctica universitaria en los centros

En cuanto a nuestra labor, como ya hemos dicho, se deben tener en cuenta tres dimensiones: la referente al nivel cognitivo (no es lo mismo conocer, que comprender, que trasmitir, que enseñar), al ámbito del conocimiento (informática, matemáticas, lengua) y al nivel educativo (infantil, primaria, secundaria, bachillerato, universidad, adultos). Cada vez es más habitual que las universidades, a través de los ICE, creen cursos de formación del profesorado. Pero estos cursos suelen ser generales, dedicados a todo el profesorado de la universidad. Los centros deben tener profesores interesados en la docencia que hagan de puente entre los ICE y los profesores, ya que estos serán los que mejor sepan cómo adaptar las propuestas metodológicas a las características particulares de cada campo y a la realidad concreta de cada centro/titulación.

Este puente puede tomar varias formas. Por ejemplo en el Departamento de Matemáticas e Informática de la UIB ya hace varios años que funciona un seminario de docencia en la que se tratan problemas docentes concretos que afectan al profesorado del departamento. Este seminario ha fomentado la reflexión sobre docencia en el departamento. Otra posibilidad es tener responsables docentes con categoría de subdirector en las escuelas y departamentos con la responsabilidad de supervisar, coordinar y fomentar las innovaciones docentes en nuestros estudios.

Sea de la forma que sea cada centro debe fomentar la discusión pedagógica entre sus miembros sacando la docencia ‘del armario’.

2.7. El equipo docente está formado por PAS, PDI y alumnos

Uno de los problemas de la universidad es la marcada separación entre los tres colectivos universitarios. Muchos profesores ni siquiera conocemos los nombres de los PAS de las secretarías y bibliotecas, y mucho menos sus funciones; los PAS no conocen las características docentes de la escuela o facultad ni de los diferentes estudios y asignaturas; los alumnos procuran vivir alejados de profesores y PAS. Lo malo de esta situación es que es una fuente de problemas, trabajo innecesario y pérdida de capacidad docente. Un PAS que conozca bien el proceder y necesidad de los profesores podrá tomar mejores decisiones a la hora de responder preguntas de los alumnos o asignar recursos; si conocemos bien el funcionamiento de las secretarías y bibliotecas nos evitaremos horas de marañas burocráticas. Los alumnos deberían ser nuestros colaboradores y nos podrían ayudar a poner al día a los alumnos más atrasados o en la corrección de trabajos [11]. También nos ayudarán con sus ideas y críticas a mejorar nuestra labor [13]. Debemos reconocer que formamos un equipo y que nos necesitamos y complementamos. Además, nos ahorrará trabajo y frustraciones. Tenemos un objetivo común y tenemos que trabajar conjuntamente para alcanzarlo.

2.8. Mejoremos la distribución de las actividades docentes

En la actualidad en la universidad española, cuando un profesor se encarga de una asignatura se encarga de *toda* la asignatura, de todos los detalles, desde los más importantes (programa, temario) a los más nimios (redactar los problemas, tener al día la página web). En las grandes universidades americanas los profesores principales se ocupan de las cuestiones importantes, como la preparación del temario y coordinación con otras asignaturas, mientras que los TA (*Teaching Assistants*), en general alumnos de doctorado, se ocupan de las clases de problemas y buena parte de las tutorías, y otros ayudantes (a menudo estudiantes de últimos cursos) se encargan de resolución de problemas, corrección de las tareas y asuntos menores. Esto da lugar a una mejor atención al alumno, ya que cada miembro del equipo puede ocuparse de la parte para la que está más motivado o mejor dotado y además va enseñando el oficio al aspirante a profesor, que más adelante tendrá más experiencia cuando le llegue el momento de encargarse de una asignatura.

En vez de usar un perfil único (el profesor) para todas las partes de todas las asignaturas, debemos definir distintos perfiles profesionales para realizar los diferentes tipos de actividades que conforman la docencia de una asignatura. No hemos de pensar sólo en los profesores con mayor o menor grado y experiencia, sino que nos podemos aprovechar de las competencias de los técnicos de laboratorio, e incluso de los estudiantes de cursos superiores que pueden ayudar como tutores de otros estudiantes. Debemos involucrarnos de forma conjunta en todas las actividades docentes, aportando cada uno de nosotros, cada uno de los colectivos, nuestra propia formación, tanto para nuestro bien como para una mejor educación del alumno.

2.9. Enseñemos a los alumnos a trabajar en equipo

En los últimos años la sociedad ha pasado de trabajar individualmente o en pequeños grupos en un entorno agrícola y de pequeña empresa familiar, a trabajar en grandes empresas en donde gente de formación diversa ha de cooperar formando equipos que deben resolver problemas complejos. Si se tiene un buen equipo los frutos del trabajo son mayores que los conseguidos a partir del que se elabora individualmente. Pero si no se sabe trabajar con otros, si simplemente eres un grupo y no un equipo, el rendimiento conseguido es incluso inferior al rendimiento de una persona brillante trabajando sola. Esto no es extraño: los enfrentamientos o la falta de sincronización da lugar a trabajo inútil, a reinventar la rueda varias veces, a discusiones eternas sin salida. Esto se comprueba enseguida poniendo a trabajar a los alumnos en grupos de distinto tamaño: no hay correlación entre el resultado obtenido y el tamaño del grupo.

Si queremos que nuestros alumnos lleguen al mundo laboral sabiendo trabajar en equipo debemos enseñarles a hacerlo. No basta agrupar a tres o cuatro alumnos y decirles que trabajen juntos, hemos de enseñarles a dividir el trabajo dando a cada uno la parte que hará mejor; a saber reunirse, obteniendo soluciones y no discusiones inútiles; a responsabilizarse cada uno de su tarea y no fallar al equipo; a comunicarse entre ellos cuando es necesario; a no enfadarse ni dejar que el ego de alguno sea el que domine la situación. El equipo ha de poder hacer mucho más que una persona. A esto se le llama *interdependencia positiva.*

Si usamos el equipo para mejorar el aprendizaje usamos lo que se llama *aprendizaje cooperativo* [5]. En este aprendizaje todos trabajan en equipo para aprender más. Otra vez, no es una mera división del trabajo (*yo hago los problemas pares, tú los impares*), sino un trabajo en equipo para que todos aprendan, para que el *equipo* aprenda: es necesario que todos los integrantes aprendan juntos todo lo que el equipo ha hecho. Esto recibe el nombre de *exigibilidad individual.*

Como vemos el trabajo en equipo y el aprendizaje cooperativo son conocimientos complejos y que no pueden enseñarse en unas horas sino que deben enseñarse a lo largo de la carrera. Es difícil, y quizá por eso se use poco [9]. Por suerte, existe abundante material [2,6] que nos puede ayudar a formar a nuestros alumnos en esta competencia.

2.10. Usemos el Aprendizaje Basado en Proyectos

El ABP es un método didáctico en el que el estudiante aprende los conceptos de la carrera mediante la realización de un proyecto. Supone un cambio importante en la dinámica de las clases, pues se pierde la filosofía de clase tradicional, e incluso

de asignatura, ya que a menudo el proyecto base del aprendizaje corresponde al conocimiento de varias asignaturas, lo que es más cercano a la realidad profesional. A grandes rasgos las características de las clases son: la lección magistral desaparece y en su lugar se imparten seminarios sobre temas de interés; el profesor se convierte en un director o guía del proceso de aprendizaje; el trabajo se realiza en grupos reducidos, que deben organizarse y gestionarse de forma adecuada para realizar el proyecto; se tiende a que se deje para casa el trabajo que se ha de desarrollar de forma individual y la mayor parte de las reuniones del equipo tienen lugar en las sesiones de clase, con la ayuda del profesor; se proporciona gran cantidad de información y los recursos necesarios para obtener información complementaria; se dispone de un conjunto completo de herramientas en línea de trabajo colaborativo, etc.

Aunque en principio esta idea de aprender problemas y no asignaturas choque con la organización administrativa de la universidad, se ha demostrado que es posible: la EPS de la UPC en Castelldefels funciona completamente de esta forma y en la UA se ha hecho la experiencia con cuatro asignaturas optativas en las titulaciones de informática [10].

2.11. Combinemos el aprendizaje con el uso adecuado de las tecnologías en la enseñanza

Es obvio que a mayor cantidad de medios y recursos, más fácil es conseguir los objetivos. La utilización de la pizarra y la exposición oral de los contenidos de las asignaturas es un recurso muy importante con capacidades únicas. Pero podemos mejorar el uso de pizarra y las exposiciones utilizando nuevas tecnologías, como las transparencias y presentaciones. Los nuevos recursos no sustituyen a los antiguos sino que los complementan: dan la posibilidad al profesor de utilizar lo mejor de cada uno. Y podemos combinar lo mejor de cada herramienta tanto en clase como fuera de ella, mejorando la docencia que ya no tiene lugar sólo en el aula. Además, la docencia semipresencial está siendo demandada por alumnos y administradores. Esta combinación del aprendizaje a distancia con los métodos presenciales, usando lo mejor de cada parte, recibe el nombre de *B–Learning* (*Blended–Learning*).

Esta modalidad de enseñanza–aprendizaje semipresencial incluye tanto formación presencial como *E–learning*. La utilización de la *educación virtual,* a través de Internet aumenta la capacidad de enseñanza ya que permite que el alumno adapte el ritmo de aprendizaje a sus circunstancias. La disponibilidad de herramientas de aprendizaje independientes de límites horarios o geográficos ayuda al acercamiento entre el profesor y sus alumnos y también de los diferentes alumnos entre sí. La utilización de herramientas de correo electrónico y de *chat* permite que exista un proceso de tutorización con soporte electrónico, que aparezcan foros de debate en los cuales se fomenta la participación y la discusión de las ideas y conceptos que tanto los profesores como los propios alumnos plantean.

El uso de todas las tecnologías, usadas adecuadamente, permite una mejor educación que el uso exclusivo de las herramientas presenciales.

2.12. Utilicemos videojuegos como apoyo al aprendizaje

Cuando aparece una nueva tecnología empieza usándose de la misma forma que las que sustituye: la información de la pizarra pasó a las transparencias y después a presentaciones que se podían colgar de la red, pero era la misma información. La innovación empieza siendo una pequeña mejora pero se convierte en una gran mejora cuando cambia la forma de enseñar, cuando se pasa de *hacer lo viejo de manera nueva* a *hacer cosas nuevas de formas nuevas.*

Los alumnos que entran ahora en la universidad han crecido en un mundo muy distinto del nuestro. Presentarles la materia con las formas viejas, aunque sea usando tecnología nueva dificulta captar su atención. Una buena clase, usando ejemplos y métodos que les son más afines ayuda, pero podemos también captar la atención del alumno cambiando la forma en que la materia se ofrece al alumno haciendo que sea interesante, incluso lúdica.

En este sentido la utilización de videojuegos es un recurso atractivo [4,8]. Para ellos un videojuego es algo natural, un juego más, algo que conocen y que *entienden*. Podemos hacer uso de esta comodidad que tienen los alumnos, integrando los videojuegos en el proceso de aprendizaje. No se trata, obviamente, de la utilización de las herramientas pedagógicas del estilo de los juegos

educativos para niños pequeños sino proponer como práctica la implementación de un juego o utilizar un juego, o partes de uno, para explicar, por ejemplo, los conceptos de la orientación a objetos. En particular, las prácticas de la asignatura de *Lógica computacional* de la UA han consistido en programar en Prolog un videojuego de tipo aventura conversacional llamado: «Aquí no hay quien estudie… ¿lógica?»

3. La reflexión

Somos Personal Docente e Investigador por lo que nos debemos a ambas tareas. Esto crea en el profesor universitario un conflicto interno: lo que dedicas a una actividad, se lo quitas a la otra y hace necesario equilibrar ambas actividades. Este conflicto es doblemente doloroso en la universidad española para los que queremos inclinar la balanza hacia la docencia: la docencia se nos valora simplemente por la asistencia al aula y la investigación por los trabajos publicados; los quinquenios se 'regalan' los sexenios hay que ganárselos; a la hora de promocionar se mirará tu curriculum investigador, y se echará una breve ojeada al docente. Esto hace muy duro, sobre todo a los jóvenes, el poder dedicar tiempo a buscar nuevas maneras de mejorar su capacidad docente.

No debemos contraponer docencia e investigación, ya que deben convivir juntas. No creemos que sea bueno separar las dos carreras profesionales, pero tampoco debemos estar en inferioridad de condiciones los que dedicamos grandes esfuerzos a la docencia. La solución no está en optar en exclusiva por docencia o investigación sino en complementar la docencia con la investigación. Para ello podemos investigar en docencia aplicando los métodos de la investigación a la tarea docente [3,12] y publicar estudios serios, tan serios como cualquier investigación en un área 'tradicional'.

No se trata de desprestigiar y abandonar la investigación sino de elevar el prestigio de la docencia. Esto ya se ha conseguido en buena parte con el Jenui, pero aún queda mucho camino por recorrer. La investigación en docencia requiere unos métodos de planteamiento, ejecución, medida de resultados, tan rigurosos como la investigación en cualquier otro campo. Debemos trabajar para mejorar la investigación en docencia y hacer que deba valorarse tanto como la investigación en cualquier otra área.

4. Conclusión

En el clima de cambio que vivimos es importante que los profesores 'de a pie' tomemos la iniciativa y promovamos la mejora docente, dando ideas y caminos a las instituciones, en vez de esperar que estas nos dicten el camino. Hemos mostrado doce propuestas que pueden ser más o menos difíciles de llevar a cabo, pero que están a nuestro alcance para iniciarlas ya.

Proponemos también como reflexión más a largo plazo, que la docencia en informática se convierta en una disciplina de investigación tan seria como cualquier otra. Esto obliga a mejorar aún más y a ser más rigurosos en nuestro trabajo docente, pero es una salida que puede reducir los conflictos de la dicotomía docencia/investigación en la que forzosamente nos vemos envueltos.

Como punto final, creemos que es importante para todos nosotros recordar que aunque a menudo no se ven los resultados de la docencia, no significa que no sean trascendentes, como explica esta cita de la película *The Emperor's club:*
«*A great teacher has little external history to record: his life goes over into other lives. These men are pillars in the intimate structure of our schools. They are more essential than its stones or beams, and they will continue to be a kindling force and a revealing power in our lives.*»
(Un gran maestro tiene poca historia externa que mostrar: su vida se vierte en otras vidas. Estas personas son las columnas de la estructura central de nuestras escuelas. Son más esenciales que sus piedras o vigas, y continuarán siendo una fuerza inspiradora y un poder revelador de nuestras vidas)

Referencias

[1] Cifre, E., Montaño, J., Munar, A., Socias, M. *Les competències genèriques i els titulats universitaris a les Illes Balears.* UIB, 2006. Información adicional en URL: http://www.uib.es/servei/ocihe/competencies/index.html. Fecha última visita, Feb. 2007.

[2] Felder, R.M.. *Active and cooperative learning.* URL: http://www.ncsu.edu/felder-

public/Cooperative_Learning.html. Fecha última visita, Feb. 2007.

[3] Fincher, S. y Petre, M. *Computer science education research.* RoutledgeFalmer, 2004.

[4] Gallego, F., Satorre, R. Y Llorens, F. *Computer games tell, show, involve, ...and teach.* Actas del VIII Simposio Internacional de Informática Educativa SIIE'06, León 2006.

[5] Johnson, R. y Johnson, D. *Cooperative learning center* URL: http://www.cooperation.org/. Fecha última visita, Feb. 2007.

[6] McConnell, J. *Active and cooperative learning. (4 parts)*. SIGCSE Bulletin **37**(2), pp 27 – 30; **37** (4), pp. 34 – 38; **38**(2), pp. 24 – 28; y **38** (4), pp 25 – 28.

[7] Ministerio de Educación y Ciencia. *Directrices para la elaboración de títulos universitarios de grado y máster. (Documento de trabajo).* 21 de Diciembre de 2006. URL:http://wwwn.mec.es/mecd/gabipren/documentos/directrices.pdf

[8] Prensky, M. *Digital Game-Based Learning*, NewYork, McGraw-Hill, 2001

[9] Traver, V.J. y Traver, J.. *¿Por qué no enseñamos a aprender cooperativamente?* Actas de las X Jornadas de Enseñanza Universitaria de Informática, Jenui 2004, pp. 297 – 304. Alicante, 2004.

[10] Universidad de Alicante, Escuela Politécnica Superior *Aprendizaje Basado en Proyectos* URL: http://www.dccia.ua.es/ABP/

[11] Valero-García, M y Díaz de Cerio, L.M. *Evaluación continuada a un coste razonable.* Actas de las IX Jornadas de Enseñanza Universitaria de la Informática (Jenui 2003), pp. 183 – 190. Cádiz, 2003.

[12] Van der Vleute, C.P.M. *Más allá de la intuición* Boletín de la Red-U, Vol. 1, num. 2, 2001. URL: http://www.uc3m.es/uc3m/revista/n-2-bol-red-estatal-docencia-univ.html#intuicion

[13] Walker, Henry M. *Thoughts on student feedback to help teaching.* SIGCSE Bulletin **38** (4), pp 13 – 14.

¿Compensa el Esfuerzo Realizado al Aplicar Técnicas de Aprendizaje Activo?

Carlos E. Vivaracho Pascual, Mª Aránzazu Simón Hurtado, Alejandra Martínez Monés, Noelia de las Heras Osorno
Departamento de Informática
Universidad de Valladolid
E.T.S. Ing. Informática
{cevp,arancha,amartine}@infor.uva.es, nheroso@ulises.tel.uva.es

Resumen

En el presente trabajo, desarrollado dentro de las actividades llevadas a cabo por el grupo GREIDI de la Universidad de Valladolid, se muestra el resultado de la evaluación realizada a la aplicación de técnicas de aprendizaje activo, a alumnos de primero de Ingeniería Técnica en Informática de Gestión. Para la evaluación se ha utilizado el método del "focus group", ya que permite recoger la opinión del alumno de una manera más amplia y con más matices que la mostrada en una encuesta. Tras varios años de aplicación de técnicas de aprendizaje activo en nuestras asignaturas, la realización de esta evaluación viene motivada por un cierto pesimismo por parte de los profesores. Dicho pesimismo es debido a la nula influencia observada en el rendimiento académico y en la motivación del estudiante, y a la constatación de que numerosos alumnos habían desarrollado estrategias de "adaptación" al sistema que hacen que en vez de trabajar, simplemente se aprovechen del esfuerzo de otros. Ante esta situación, un poco desalentadora para el profesor debido al esfuerzo realizado, hemos querido profundizar en cómo perciben los estudiantes los cambios que se están produciendo en la forma de impartir docencia y, sobre todo, si estos nuevos métodos están suponiendo realmente una mejora en su proceso de aprendizaje.

1. Introducción

Es de sobra conocido que uno de los aspectos importantes de la adaptación de los estudios universitarios al Espacio Europeo de Educación Superior (EEES) es el cambio en las metodologías docentes [1]. Este cambio afecta tanto al profesor como al estudiante, por lo que creemos que es importante profundizar en cómo se está viviendo por ambas partes.

El interés por estos temas llevó a un grupo de profesores de Ingeniería de la Universidad de Valladolid a formar un grupo de investigación docente, interesado en la aplicación y estudio de técnicas de aprendizaje activo en el aula, GREIDI (Grupo de Estudio en Innovación Docente en Ingeniería) [3], al que pertenecemos los autores de este artículo. Sin embargo, tras varios años aplicando en nuestras asignaturas técnicas de aprendizaje activo basadas en el Aprendizaje Cooperativo (AC) [2], hemos observado una influencia nula en el rendimiento académico, entendiendo éste como porcentaje de aprobados, tanto sobre matriculados como sobre presentados, y una adaptación de numerosos alumnos al nuevo esquema de aprendizaje, que hace que en vez de trabajar, que es uno de los pilares básicos en las nuevas metodologías, se aprovechen del esfuerzo de los demás, por lo que las técnicas docentes planteadas dejan de ser efectivas. Tampoco hemos observado un incremento en la motivación del estudiante.

Por otro lado, es difícil valorar a corto plazo, y más para alumnos de primero, si se está consiguiendo otro de los objetivos perseguidos con la implantación de las nuevas metodologías docentes: la adquisición de competencias adicionales como, por ejemplo, el hábito de trabajar de manera continuada, la capacidad de realizar trabajo cooperativo, la capacidad de aprender de manera autónoma, el desarrollo de habilidades de expresión oral y escrita, etc.

Quizás el panorama planteado es pesimista, pero creemos, por nuestra experiencia, que la decepción ante los cambios metodológicos implantados es una realidad a la que muchos docentes nos enfrentamos, y más cuando la

aplicación de esas técnicas suponen un gran esfuerzo por nuestra parte; esfuerzo, por otro lado, no recompensado de manera oficial (proyectos piloto a "coste cero", al final lo que realmente nos cuenta es nuestra actividad investigadora, ...), por lo que uno espera, al menos, la recompensa en términos de formación del estudiante.

Ante esta situación uno se pregunta si merece la pena, y ese interrogante fue lo que nos motivó a realizar lo que presentamos en este artículo.

Hasta ahora la evaluación del impacto del cambio metodológico sobre los alumnos se ha venido haciendo, sobre todo, a partir de las observaciones del profesor, encuestas y valoración de los resultados académicos. Estos métodos son efectivos, pero el primero es demasiado subjetivo para la información que queríamos obtener, el segundo demasiado limitado, ya que no muestra la opinión del alumno más allá de la elección entre las respuestas que se le dan, y el rendimiento académico sólo es uno de los aspectos que queremos valorar. Necesitábamos un análisis más profundo con el fin no solo de comprobar si merece la pena el cambio metodológico realizado con respecto a los objetivos iniciales planteados, sino también de detectar posibles fallos y apuntar nuevas soluciones.

Por estas razones nos planteamos utilizar la técnica del "focus group", ya que permite conocer de manera amplia y abierta la opinión del alumno acerca de la metodología empleada

La técnica del "focus group" [4] consiste en plantear a un grupo de personas, alumnos en este caso, una serie de preguntas para que de manera abierta y libre respondan a ellas. El hecho de trabajar con un grupo permite aprovechar la interacción que surge entre los participantes para ahondar en posibles opiniones contradictorias, o conflictos que pudieran haber pasado desapercibidos. Aunque existe un guión previo en cuanto a las preguntas a plantear, éste se puede cambiar sobre la marcha de acuerdo a como vaya la discusión. Es importante que la persona que coordina sea totalmente ajena al profesor. Todo lo que se hable es grabado, para que no se pierda nada de las opiniones aportadas.

Anticipando un poco los resultados, podemos comentar que, en contra de nuestra percepción pesimista inicial, y a pesar de los aspectos negativos que aparecen, la opinión de los alumnos es, en general, favorable a las técnicas empleadas, ya que el trabajo planteado les ha ayudado a mejorar su proceso de aprendizaje, incluso a alumnos con muy buenas calificaciones.

El resto del artículo se organiza como sigue. En el apartado 2 se resumirá la metodología docente empleada y que se evalúa. A continuación (apartado 3) se mostrará cómo se llevó a cabo la realización del "focus group". Los resultados de éste, en forma de resumen de las respuestas dadas a las preguntas planteadas, se verán en el apartado 4. Finalizaremos con un análisis y discusión de esas respuestas (apartado 5) y las correspondientes conclusiones (apartado 6).

2. Metodología Docente

La asignatura de Fundamentos de Informática I tiene dos partes diferencias:

- Teoría, centrada en la codificación de la información.
- Laboratorio, donde se explica el uso básico de los comandos del Sistema Operativo UNIX.

Son dos partes completamente independientes y en cada una se siguen metodologías diferentes, aunque ambas basadas en AC. Se comparte la composición de los grupos cooperativos formados, cuyo tamaño es de tres miembros.

Con respecto a la parte de teoría, la explicación de cada tema se realiza mediante lección magistral, tras la cual el profesor realiza una serie de problemas. Acabado esto se les plantea a los alumnos la realización en grupos de un entregable que consiste en una serie de problemas sobre ese tema contenidos en exámenes de años anteriores. Este trabajo se realiza fuera del aula y es evaluado en una hora de clase mediante exposición pública, escogiendo al azar un grupo, y dentro de él a uno de sus componentes, que tiene que explicar la solución dada por su grupo al problema que le haya tocado; la calificación que obtenga (se valora la habilidad de expresión oral) se extiende al resto de miembros del grupo. Para no perder tiempo escribiendo la solución en la pizarra, se usa un proyector de opacos, lo que permite al alumno centrarse en la explicación y a nosotros realizar más preguntas acerca de la solución planteada, para ver si la entiende o no. La calificación obtenida en los entregables, junto con la lograda en una prueba parcial individual realizada a mediados de cuatrimestre, suponen el

20% de la calificación final de la teoría. El examen es el 80% restante.

En cuanto al laboratorio, la materia a estudiar se divide en dos partes. Para cada una se plantea la misma dinámica de trabajo:

1. **Sesión de laboratorio 1**. Explicación en el aula por parte del profesor de parte de la materia involucrada en esa parte.
2. Trabajo individual fuera del horario lectivo. Este trabajo será necesario para la siguiente sesión de laboratorio. Cada miembro del grupo prepara una parte diferente de la materia no explicada por el profesor.
3. **Sesión de laboratorio 2**. Se aplica la técnica del puzzle para el estudio de la parte de la materia que se tienen que preparar ellos. En la primera parte se reúnen por "grupos de expertos", es decir, aquellos alumnos de varios grupos que les haya tocado estudiar la misma materia, comparten dudas y realizan unos ejercicios. Acabado esto vuelven al grupo y cada miembro explica al resto la parte que le haya tocado estudiar. Para finalizar se resuelven en público los ejercicios realizados en grupos de expertos.
4. Fuera del horario lectivo y por grupos realizan una serie de ejercicios.
5. **Sesión de laboratorio 3**. Se evalúa el trabajo realizado en el paso 4 mediante exposición pública, de la misma manera que la indicada en teoría. En la última media hora un alumno elegido al azar de cada grupo resuelve una pequeña prueba con ejercicios similares a los evaluados durante la sesión. La calificación que obtenga se extiende a todo el grupo.

Este trabajo supone un 20% de la calificación de la asignatura y el examen el 80% restante.

Más información acerca de las metodologías expuestas se puede encontrar en [6] para la parte de teoría y en [5] para la parte de laboratorio.

3. Planificación del "focus group"

Con el fin de evaluar la sensación general que sobre la asignatura tenían los estudiantes, el "focus group" se realizó finalizada la asignatura y conocidas las calificaciones en ella obtenidas.

Para obtener una evaluación más completa se planteó la clasificación de los estudiantes en tres grupos, de los que nos interesaba obtener información específica:

- **Grupo 1.** Alumnos con muy buenas calificaciones en el examen (notable alto, sobresaliente y matrícula de honor). La información específica que queremos obtener con este grupo es si la dinámica seguida les ha ayudado a obtener ese buen rendimiento, les ha perjudicado o creen que hubieran comprendido igual la asignatura si se hubiera seguido la metodología clásica basada en lección magistral y examen final.
- **Grupo 2.** Alumnos con aprobado justo. Nos interesa saber si el trabajo realizado durante el curso les ha ayudado a aprobar, es decir, a tener un conocimiento suficiente de la materia, o creen que a lo mejor les ha perjudicado quitándoles tiempo para el estudio personal de esta u otra asignatura.
- **Grupo 3.** Alumnos suspensos. Nos interesa conocer por qué no han superado la materia, si han seguido o no el trabajo propuesto durante el curso, y de ser así por qué no ha sido eficaz. Si no lo han seguido estamos interesados en saber la razón y si creen que de haberlo hecho habrían aprobado.

Para favorecer la interacción y la participación es importante que el tamaño de los grupos no sea muy grande, por lo que se planteó que éste fuera de entre 8 y 10 alumnos. Los profesores seleccionamos los candidatos a formar parte de cada grupo y preparamos los cuestionarios iniciales. Ese material se le paso a la persona que se iba a encargar de realizar el trabajo, una becaria del grupo GREIDI totalmente ajena a la asignatura. Ésta se encargó de contactar con los alumnos seleccionados, contarles el objetivo de la dinámica a realizar y quedar con aquellos que voluntariamente se quisieran apuntar. Tras esto dirigió cada sesión, grabándola y trascribiendo estas grabaciones.

De los grupos de alumnos inicialmente propuestos se ofrecieron como voluntarios a participar en la dinámica casi todos los que habían obtenido muy buen rendimiento académico, la mayoría de los aprobados pero, curiosamente, ninguno de los suspensos en la asignatura.

Hay que comentar que el "focus group" se realizó a mediados del segundo cuatrimestre, cuando la carga lectiva no era grande.

4. Resultados del "focus group"

Dada la composición de los alumnos que formaban los grupos, todos ellos habían superado con éxito la asignatura, nuestro interés en el cuestionario se centró en saber si la metodología seguida les había ayudado, perjudicado o hubiera sido indiferente haber seguido una metodología clásica. Este aspecto es importante para el profesorado, ya que solo en el primer caso se podría justificar el esfuerzo extra que supone para nosotros llevar a cabo este tipo de metodologías.

Durante el curso 2005/2006 se llevó a cabo en todas las asignaturas de primero de Informática Técnica, tanto Gestión como Sistemas, de la Universidad de Valladolid, una experiencia piloto de aplicación de nuevas metodologías docentes dentro del plan de adaptación al EEES. Aunque el "focus group" se centró en la experiencia realizada en Fundamentos de Informática I, es conveniente comentar que algunas de las respuestas se refieren no sólo a la asignatura indicada sino a la experiencia global realizada.

A continuación se muestra de manera resumida las respuestas dadas por los alumnos a las preguntas que dio tiempo a tratar, el cuestionario completo está disponible en la página Web del grupo: www.greidi.uva.es. El resumen ha sido realizado por la persona encargada de dirigir el "focus group", es decir, una persona ajena a los docentes de la asignatura.

Respuestas del Grupo 1

Pregunta. *El método que habéis utilizado ha mejorado vuestro proceso de aprendizaje en la asignatura o creéis que no. ¿Cómo ha sido para vosotros el método que han seguido?*

Respuesta. El método utilizado les parece bueno, pero opinan que los entregables se juntan con los de otras asignaturas y no les da tiempo a hacerlos algunas semanas. Creen que debería haber más coordinación entre los profesores para que no se les junten los entregables en una semana. Prefieren hacer los entregables por temas y no hacer menos y más largos (*Aclaración a este último comentario: cuando se hizo el "focus group" se estaba impartiendo Fundamentos de Informática II en la que, debido a la carga de los profesores implicados, se tuvo que disminuir el número de entregables, haciendo uno por cada dos temas, en vez de uno por tema como en Fundamentos de Informática I).*

Preg. *¿Creéis que la calificación que habéis obtenido se debe al método seguido, o si se hubiera utilizado otro diferente hubierais obtenido la misma calificación?*

Res. Los entregables les gustan, y les ayudan a mejorar el aprendizaje, pero sin embargo, lo que no les gusta nada son las pruebas individuales en las que la nota del alumno se extiende a todo el grupo. Consideran que es injusto que si una persona del grupo sabe hacer los problemas todos saquen buena nota, y si le toca a una que no lo sabe bien la calificación mala se extiende a todos, y hay gente que pasa de la asignatura y no la estudia.

Preg. *Habéis trabajo en grupo o de manera individual.*

Res. La mayoría de las veces han trabajado en grupo, sin embargo, cuando se les juntaban muchos entregables se repartían los problemas entre las personas del grupo y luego se juntaban para ponerlos en común. Para las pruebas individuales cada uno estudiaba por su cuenta y no se reunían para nada, ya que consideran que bastante tienen con estudiarse su parte como para perder tiempo en explicar al resto de los compañeros la materia.

Valoran positivamente el trabajo en grupo, porque lo ven como un método de trabajo en su futuro profesional.

Preg. *¿El trabajo con otros compañeros os ayuda a aclarar conceptos cuando trabajáis en grupo?*

Sí, porque les resulta más fácil preguntar a un compañero que al profesor y además les pone ejemplos que a ellos les han servido para entenderlo.

Preg. *¿Si no hubieran existido los trabajos en grupo y los entregables, este tiempo lo hubierais dedicado al trabajo personal de la asignatura o de otras asignaturas, o por vuestra cuenta vosotros no hubierais hecho nada?*

Res. Les parece una buena ayuda, porque si no, lo hubieran dejado todo para el último momento. Además les guía para ver por dónde tienen que ir y se hacen una idea de las cosas a las que se tiene que prestar más atención.

Les ayuda a estudiar más la asignatura ya que cuando llega el examen ya lo tienen casi todo

preparado. Por lo tanto, trabajar los entregables no les parece una pérdida de tiempo.

Preg. *Sobre la presentación oral de las soluciones al entregable.*

Res. No les gusta nada porque se ponen nerviosos y aunque lo sepan no saben explicarlo. Se debiera dar la oportunidad al resto de compañeros de ayudar a la persona del grupo que ha salido ya que puede ser que una persona tenga un mal día, y por los nervios no sepa salir airoso y perjudicar a todos los del grupo.

Preg. *¿Las presentaciones orales os ayudan a aprender a expresaros mejor?*

Sol. No, porque están pensando en decir lo que tienen en la hoja y sentarse, y no piensan en cómo se expresan. Sólo quieres decirlo rápido para que el profesor vea que lo tienes hecho y sentarte cuanto antes. Además, ha habido muchas disputas entre los grupos por el hecho de no saber decirlo bien y tener mala nota todos los del grupo.

Otros piensan que sí, ya que los profesores les van ayudando a expresar las cosas y les van insistiendo hasta que lo dicen bien.

Preg. *¿Os han ayudado los entregables a llevar la asignatura al día?*

Res. Sí, porque les ayuda a estudiar para preparar el examen, ya que "hasta que no te pones a estudiar no te das cuenta de las cosas que no entiendes".

Aquellos que no han seguido el método, en junio o septiembre debieran tener la oportunidad de recuperar este trabajo realizado durante el curso.

Preg. *¿Ha mejorado vuestro rendimiento académico (resultado del examen) gracias al trabajo realizado durante el curso?*

Res. Sí, ya que ya saben a lo que se tienen que dedicar y no pierden tiempo en otras cosas.

En época de exámenes no le dedican tanto tiempo a esta asignatura y lo pueden emplear en otras.

Preg. *¿Por qué razón se copian los entregables?*

Res. Hay veces que no tienen tiempo, y para no tener un cero en el entregable lo copian, aunque también depende mucho de cómo trabaje el grupo y de otras situaciones, como por ejemplo, copiar algún ejercicio para que el entregable no quede incompleto. Además, se trasmite mucho miedo porque si un ejercicio no lo tienen hecho y les toca salir se les pone un 0, entonces lo copian, ya que es injusto que se tenga 4 ejercicios hechos y por uno que no, se les ponga un 0.

Preg. *¿Prefieres trabajar a tu ritmo, o es bueno que te lo marquen porque ayuda a no relajarse y dejarlo todo para el final?*

Res. Prefieren que se les marquen pautas de estudio.

Preg. *Teóricamente la metodología empleada (entregables, parciales, etc.) debe potenciar el trabajo durante el curso. ¿Es esto verdad en mi caso y, en general, en el resto de mis compañeros? ¿Por qué?*

Res. En las asignaturas que les obligaban a entregar y realizar trabajos sí, pero en las que no les mandaban nada, no hacen nada.

Preg. *Teóricamente la realización de entregables y pruebas parciales debe mejorar la comprensión de la asignatura, por lo que debiera mejorar el rendimiento académico. ¿Por qué crees, entonces, que en el examen el porcentaje de suspensos ha sido tan alto?*

Res. Una razón que comentan es que como la llevan más al día, al estudiarla durante el curso y creerse que la entienden, la dejan de estudiar para centrarse en otras que ven que llevan peor.

También puede influir las asignaturas a las que se presentan a examen. Además, hay alumnos que se piensan que es muy fácil y se confían, no estudian lo necesario y luego se dan el "porrazo".

Preg. *¿Preferís la metodología tradicional o ésta?*

Res. En esta materia sí, pero en otras que no se entiendan por nadie del grupo, el hacer entregables les perjudica. Alguno comenta que la metodología empleada le ha perjudicado ya que los entregables le han quitado puntos en la calificación final.

La metodología del segundo cuatrimestre no les gusta nada porque no lo entienden, lo ven poco operativo ya que tienen que estar preguntando todo el tiempo al profesor, así que creen que es mejor que los temas los explique éste. *(Nota aclaratoria: con este comentario se refieren a una parte de Fundamentos de Informática II donde en vez de explicar el tema el profesor se ha utilizado autoaprendizaje y aprendizaje cooperativo, es decir, los temas se los tienen que preparar por su cuenta, apoyándose en los compañeros, en la realización de ejercicios y en tutorías con el profesor).*

Preg. *¿Preferencia del trabajo en grupo o individual?*

Res. Depende del grupo, porque si éste no funciona les perjudica, y esto ha pasado mucho en los grupos de clase. Hay personas del grupo que no hacen nada o que han dejado la asignatura, perjudicando al resto.

Una ventaja de trabajar individualmente es que no pierden tiempo en cosas que ya se saben. Sin embargo, ven bueno el trabajo en grupo porque en su vida profesional van a tener que trabajar de esa manera.

La dinámica es buena.

Se debería dar la oportunidad a las personas que trabajan y no pueden trabajar en grupo el poder presentarse al examen solamente, sin perder puntos de los entregables *(Nota: esto ya se hace, aunque por la respuesta está claro que no todos los alumnos lo conocen).*

Respuestas del Grupo 2

Preg. *El método que se ha seguido ha mejorado vuestro proceso de aprendizaje en la asignatura o creéis que no. ¿Cómo ha sido para vosotros el método que han seguido?*

Res. Les ha parecido mejor la metodología del primer cuatrimestre que la del segundo (menos entregables, uno por cada dos temas), porque están más encima de ti, con entregables, controles, etc. Además, es la única forma de que nos tómenos la asignatura en serio y no lo dejemos todo para el final.

Preg. *¿La calificación en el examen se ha visto influida por la metodología seguida?*

Res. Creen que sí, que ha ayudado porque cada semana se tenían que mirar lo de la semana pasada y les ayudaba, aunque les quita tiempo de estudio personal de la asignatura.

Preg. *¿Habéis trabajado realmente en grupo? ¿Cómo?*

Res. Sí lo han hecho, en general (ha ayudado ser todos del mismo sitio), pero ha habido personas del grupo que no han aportado mucho y que además molestaban. Otros grupos han realizado los ejercicios de manera individual y luego se han juntado para ver las soluciones. Otros simplemente lo han copiado de los demás.

Preg. *¿Trabajar con vuestros compañeros os ayuda a entender mejor la asignatura?*

Res. Sí que les ha aportado mucho, ya que si alguien del grupo lo entiende, un compañero se lo explica mejor y lo entienden mejor. Por eso, en esa parte, el trabajo en grupo les ayuda.

Preg. *¿Si no hubiera existido el trabajo en grupo durante el curso, hubieras empleado ese tiempo en estudio personal tanto de esta asignatura como de otras y habrías asimilado mejor los conceptos*?

Res. No. La realización de entregables les ayuda a llevar al día la asignatura, ya que si no, lo dejarían todo para el último día, y dedicarían el tiempo a hacer cosas que no tienen nada que ver con los estudios.

Preg. *Sobre la presentación oral de los ejercicios por una persona del grupo elegida al azar: ¿Anima a estudiar a todos los del grupo sabiendo que le puede tocar salir a cualquiera y que tiene la responsabilidad de la nota de todos?*

Res. Aquí han comentado lo de la extensión de la nota a todos los miembros del grupo, y su opinión depende de la calificación obtenida por el compañero, ya que si le toca a uno que no ha hecho nada, el que se lo haya trabajado va a tener una mala nota por su culpa y no es justo.

Preg. *¿La presentación oral os ayuda a aprender a expresaros mejor en público?*

Res. No, porque dicen que hablan "para el cuello de su camisa", la voz se les acelera y quieren sentarse cuanto antes. No aporta nada. Además, no se expresan como lo harían en otras situaciones menos forzadas. Lo pasan mal.

Preg. *¿Prefieres que la nota de la asignatura sea responsabilidad exclusiva de cada alumno o te parece bien que en tu nota influya lo que han hecho los de tu grupo?*

Res. Depende, más que por las notas buenas por las malas, porque si tienen un mal día y obtienen una mala calificación, si fuera para uno la asumen, pero la responsabilidad de darles una mala nota a sus compañeros es muy grande.

Preg. *¿El trabajo continuado basado en la realización de entregables y pruebas parciales te ha permitido ir aclarando dudas más fácilmente?*

Res. Sí, porque las preguntas de los entregables van sobre la materia que han dado pocos días antes, y, además, los compañeros les ayudan a resolverlas.

Preg. *Los entregables se copian,¿por qué?*

Res. Por falta de tiempo, por el agobio, por la falta de coordinación entre los profesores que se

excedieron en poner entregables en las mismas fechas, y no les dio tiempo a hacerlo todo. También comentan que hay mucho vago que se acomoda y si los demás se lo dejan, pues eso que han ganado…

Preg. *Los entregables preferís que sean evaluados o no.*

Res. Si no son evaluables no pierden tanto tiempo. Ya que tienen que trabajar, que se valore.

Preg. *El tiempo que habéis dedicado a la asignatura ¿creéis que ha sido excesivo?*

Res. No, creen que ha sido normal ya que si no se hubiera seguido ese ritmo lo hubieran dejado todo para el final y les llevaría mas tiempo el entender las cosas.

Preg. *Teóricamente esta metodología debe potenciar vuestro trabajo durante todo el curso ¿es verdad?*

Res. Sí, ya que lo llevan todo al día, porque te obliga, y aprenden.

Preg. *Teóricamente la realización de entregables y pruebas parciales debe mejorar la comprensión de la asignatura, por lo que debiera mejorar el rendimiento académico. ¿Por qué crees, entonces, que en el examen el porcentaje de suspensos ha sido tan alto?*

Res. Algunos lo han ido dejando porque ha habido muchísimo trabajo.

Además, comentan que hay alumnos que este año se lo han tomado sabático, ya que les han dado una beca. Indican que se debiera mirar sólo por la gente que quiere estar aquí y que le importe.

Hay gente que quiere ir a sistemas y están por estar, y otros probando si les gusta la carrera, por lo tanto por la metodología nueva no es, sino porque no saben qué hacer y están pasando el rato.

Preg. *¿Preferís trabajar en grupo o de manera individual?*

Res. Prefieren trabajar en grupo, pero con personas competentes, porque algunos grupos no han funcionado bien.

5. Análisis y Discusión

A la vista de las respuestas de los alumnos en el "focus group", que fueron similares tanto en el Grupo1 como en el Grupo 2, vamos a resumir las opiniones más generalizadas, comentarlas y abrir alguna línea de discusión futura.

Ante todo hemos de señalar que sus respuestas nos sorprendieron enormemente. Pensábamos que, al menos en el Grupo 1, con alumnos de muy buenas notas (varias matrículas de honor), responderían que esta metodología les había hecho perder mucho tiempo, les había impedido organizar su estudio con libertad y podían haber obtenido los mismos resultados sin necesidad de tener que entregar trabajos constantes a lo largo del curso. Sin embargo, todos han señalado que les gustaba la metodología, en general, porque les ha ayudado a llevar la asignatura al día. Ven necesario que les marquen pautas para obligarse a estudiar de forma continuada a lo largo del curso porque les falta fuerza de voluntad para hacerlo por sí solos. Si los entregables no se evaluaran, no se molestarían tanto en hacerlos.

Este tema plantea un punto de discusión que dejamos abierto: ¿esta metodología ayuda al alumno a madurar y crecer en responsabilidad, o simplemente retrasa el momento de enfrentarse a una planificación propia y más a largo plazo? Es verdad que son alumnos de primer curso y que llegan a la universidad sin hábito de trabajo continuado y con una base muy mala, pero a veces tenemos la duda de si así les beneficiamos o realmente le estamos perjudicando.

No les gusta el método de calificación porque les parece injusto que si la nota de uno es mala sea la misma para todos, aunque no lo ven mal si es al revés, que la nota buena sea para todos. La realidad es que en casos muy aislados ha ocurrido que el trabajo en grupo haga que un alumno tenga al final una calificación inferior a la obtenida en el examen, y, sin embargo, sí ocurre que algunos alumnos ven incrementada su nota, aunque normalmente no lo suficiente para pasar del suspenso al aprobado.

Les gusta el trabajo en grupo porque les ayuda a adquirir una habilidad que seguramente van a necesitar en su profesión, y porque entienden mejor la explicación de un compañero que se pone más a su nivel, pero también le ven muchos inconvenientes ya que hay miembros de algunos grupos que no han trabajado, aprovechándose del esfuerzo de los demás. Esto ha ocasionado conflictos.

No les gusta tener que exponer de forma oral los entregables. Aunque se entiende que esto puede llevar consigo un esfuerzo de superación extra, pensamos que adquirir esa habilidad forma parte de los objetivos formativos y que se deben entrenar desde primero de carrera para conseguir

cierta soltura. Al final los beneficios obtenidos compensan el esfuerzo.

La opinión de los alumnos que aprueban, ante la cantidad de suspensos en la asignatura, a pesar de usar esta metodología, es que algunos se confían: creen que por haber hecho los entregables ya dominan la asignatura y no la estudian para el examen. Nosotros pensamos que la mayor parte de los suspensos se deben realmente a una falta de trabajo: alumnos que no han trabajado a lo largo del curso y tampoco al final del cuatrimestre.

No queremos dejar de resaltar los comentarios de los alumnos participantes en el "focus group" acerca de la actitud que manifiestan algunos de sus compañeros: no se dedican de lleno a los estudios porque no están seguros de que les guste la carrera, o porque no la eligieron como primera opción y están pasando un curso a la espera de entrar en la que les gusta, otros se toman un año de descanso, etc. Esto está ocurriendo cada vez más y hace que el profesor obtenga impresiones más negativas al aplicar una metodología, sea ésta la que sea.

6. Conclusiones

¿Compensa el esfuerzo que debe invertir un profesor en horas de trabajo, a costa de disminuir la investigación, al aplicar metodologías de aprendizaje activo como la señalada en este trabajo? Con compensar nos referimos a si los beneficios que obtiene el alumno son proporcionales a ese esfuerzo.

Podemos concluir, que según la opinión del alumno, compensa. Él prefiere este tipo de metodología y señala que le ayuda más. Opinión que viniendo de alumnos que han superado con éxito la asignatura creemos que la hace todavía más interesante.

Permanece abierto, sin embargo, el tema de la "proporcionalidad". A falta de herramientas más objetivas que nos permitan medir esa proporcionalidad entre esfuerzo en horas del profesor y beneficio del alumno, la respuesta queda, por el momento, a expensas de la motivación intrínseca del docente, lo cual es claramente insuficiente si se desea aplicar con éxito las ideas del ECTS en las aulas españolas. Otra línea de actuación sería la de intentar disminuir o reconocer el coste que para el docente supone la aplicación de estas metodologías.

Agradecimientos

Este trabajo ha sido financiado parcialmente por la Junta de Castilla y León, proyectos UV31/04 de la Agencia para la Calidad del Sistema Universitario en Castilla y León, y UV08/06 de la Consejería de Educación.

Referencias

[1] European Ministers of Education, *The European Higher Education Area Bologna Declaration*, Bologna, 1999.

[2] Johnson, D.W., Johnson, R.T., and K.A. Smith, *Cooperative Learning: Increasing College Faculty Instructional Productivity,* ASHE-ERIC Higher Education Report No. 4, George Washington University, 1991.

[3] Martínez, A., Hernández, C., Vivaracho, C. E.., Simón, A., Arranz, G., Martínez, M. *Introducción de metodologías activas en la enseñanza de la informática: Experiencia del Grupo Greidi.* Jornadas sobre la Enseñanza Universitaria de la Informática. JENUI 2006, pp. 347-354, Bilbao, julio 2006,

[4] Morgan, David L. *The Focus Group Guidebook*, Sage Publications, Thousand Oaks, CA, 1998.

[5] Simón Hurtado, Arancha, Vivaracho, Carlos E., *Learning UNIX in First Year of Computer Engineering*, Proceedings ACM 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education, SIGCSE Bulletin, Inroads, Conference Proceedings, ISBN 1-59593-024-8, pp. 392, Monte de Caparica, Portugal, 27-29 junio 2005.

[6] Vivaracho, Carlos E., Simón Hurtado, Arancha y Martínez Monés, Alejandra, *Aplicación de Técnicas de Aprendizaje Cooperativo en la Parte de Teoría de una Asignatura de Primero con Aulas Masificadas*, Actas XI Jornadas de Enseñanza Universitaria de la Informática (JENUI), ISBN 84-9732-421-8, pp. 37-44, Madrid, 13-15 julio 2005.

Un estudio sobre el absentismo y el abandono en asignaturas de programación

Agustín Cernuda del Río[1], Sonia Hevia Vázquez[2],
María del Carmen Suárez Torrente[1], Daniel Gayo Avello[1]

[1]Dpto. de Informática
Universidad de Oviedo
Facultad de Ciencias – C/ Calvo Sotelo, S/N
33007 (Oviedo)
{guti, macamen, dani}@uniovi.es

[2]Centro de Innovación Tecnológica
Universidad de Oviedo
C/ Gonzalo Gutiérrez Quirós, S/N
33600 (Mieres)
uo155721@uniovi.es

Resumen

Aun en caso de que el profesorado evalúe sus propios métodos a fin de mejorar individualmente la docencia o reducir las tasas de fracaso escolar, es frecuente que los alumnos que abandonan una asignatura o no acuden a clase queden (salvo a efectos puramente contables) excluidos del estudio, pese a su gran influencia en el resultado final. Forman así una suerte de *materia oscura* cuyas motivaciones no se tienen en cuenta.

Esta falta de información no sólo se traduce en una mayor dificultad para mejorar la docencia, sino que choca frontalmente con el enfoque basado en el alumno que plantea el Espacio Europeo de Educación Superior. El desconocimiento de sus pautas de comportamiento fuera de clase hace muy difícil desarrollar estimaciones y programaciones docentes realistas y efectivas.

Muchos son los factores con potencial influencia en las tasas de abandono y absentismo, y muchas las formas de abordar el problema. En este artículo se presentan los resultados de un estudio que se dirige directamente a la base del mismo, examinando el perfil de los alumnos en relación con estos fenómenos en tres asignaturas relacionadas con la programación.

1. Introducción

En los últimos años, el Espacio Europeo de Enseñanza Superior ha propiciado la investigación sobre los hábitos de trabajo de los estudiantes fuera del aula [1][2][4]. El abandono escolar (entendido como la renuncia *de facto* a examinarse de una asignatura, simplemente dejando de acudir a clases y exámenes) y el absentismo (es decir, la baja o nula asistencia a las clases, aunque el alumno sí tenga intención de examinarse) han sido siempre objeto de preocupación de los buenos docentes, aun cuando en la universidad se ha dado tradicionalmente a ambos parámetros una importancia muy relativa. Pero la situación actual de la universidad española, bien sea en relación con el proceso de implantación del Espacio Europeo de Enseñanza Superior, bien sea en relación con los apremiantes criterios de rentabilidad empresarial que se pretende imponer, exige atacar lo que se ha dado en llamar las *bolsas de suspensos*.

En ocasiones se comprueba que la tasa de éxito de los alumnos presentados no es excesivamente baja, sino que es la alta tasa de abandono la que dispara los indicadores de fracaso. En el caso de las asignaturas aludidas aquí, de hecho, una parte considerable del abandono se produce antes del primer día de clase.

Evidentemente, poco puede hacer el profesorado para mejorar la situación con los alumnos que no acuden. Por eso, para poder plantearse cualquier acción de mejora, parece condición necesaria obtener previamente información sobre esos alumnos que engrosan las cifras de fracaso y sobre los que no se puede ejercer influencia alguna, ni buena ni mala.

Por supuesto, de lo que se trata no es de considerar de forma totalmente aislada a los alumnos que abandonan respecto a los que no lo hacen, ya que es de suponer que la pertenencia de un alumno dado a uno u otro grupo en una asignatura dada no responde a factores genéticos e inamovibles, sino más bien a circunstancias. Por tanto, el estudio se orienta a abarcar a todo tipo de alumnos para establecer un perfil de sus motivaciones o percepciones en relación con el abandono o el absentismo en una asignatura.

Hay que decir que existen estudios anteriores sobre el absentismo en nuestra Universidad [5], pero abordan el problema globalmente en muchas titulaciones científico-técnicas; los alumnos del centro estudiado aquí aportan sólo un 5,3% de las

encuestas, con lo que los resultados globales no son de aplicación particular. Además, las encuestas se realizaron a alumnos que se encontraban en clase y, una vez más, no hay dato alguno sobre los alumnos que no acuden ([5], p. 138).

2. Entorno

La investigación que aquí se describe se llevó a cabo en la Escuela Universitaria de Ingeniería Técnica en Informática de Oviedo. Concretamente, en tres asignaturas de primer curso, relacionadas con la programación, materia que habitualmente plantea notable dificultad para los estudiantes noveles: Introducción a la Programación (en adelante IP), Metodología de la Programación (MP) y Comunicación Persona-Máquina (CPM). En el plan de estudios del centro las tres figuran como asignaturas cuatrimestrales y tienen una clara orientación práctica; IP y MP tienen 6 créditos y CPM 4,5. La relación temporal (y, a efectos prácticos, de dependencia) entre ellas es la de la Figura 1. De hecho, el plan de estudios vigente (aprobado en 2002) incluye estas dos asignaturas cuatrimestrales de programación de 6 créditos sustituyendo a una anual de 12 créditos del plan anterior (de 1991). CPM se diseñó originalmente como asignatura de segundo curso, pero por imposición de los gestores universitarios el centro tuvo que ubicarla en primer curso, lo que plantea a los alumnos una dificultad añadida.

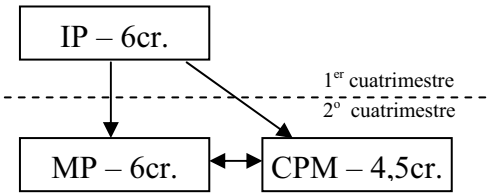


Figura 1. Esquema temporal de las asignaturas.

IP es una asignatura clave, ya que constituye para muchos alumnos el primer contacto con la programación de ordenadores. Los profesores suelen considerar que tener éxito en esta asignatura es un factor decisivo para cursar con eficacia MP y CPM.

3. Objetivos

La finalidad principal de este estudio es obtener información descriptiva sobre el perfil del alumnado de estas asignaturas en lo que tiene relación con el abandono y el absentismo, y obtener una impresión general sobre los motivos y su solución.

Hay que hacer explícito que en modo alguno se pretende obtener conclusiones definitivas con validez predictiva o acotar parámetros con precisión. El estudio pretende realizar una primera aproximación al problema con datos reales, para identificar las desviaciones más significativas y evidentes si es que existen, y por tanto las áreas de mejora más prometedoras.

Partiendo de estos datos, quizás sí quepa plantearse estudios de carácter más minucioso (en este sentido, el planteamiento es parecido al de [5]), pero no creemos que sea viable hacerlo sin la fase preliminar de aproximación que aquí se presenta. En primer lugar, resulta especialmente difícil aislar las variables que inciden en algo tan complejo como el rendimiento escolar de un alumno [3], y en segundo lugar resulta también difícil la recolección de datos. Se dispone de información sobre los alumnos que asisten a clase y se examinan, pero el problema aquí es que el objeto ideal de estudio son precisamente los alumnos que están fuera de todo proceso evaluador y son una completa incógnita; obtener datos de cada uno de ellos requiere una inversión importante.

Por tanto, este estudio pretende conseguir una primera caracterización del problema que permita o bien encontrar áreas de actuación directa (si emergen elementos cualitativos claramente relevantes) o bien ofrecer una base para futuros estudios cuantitativos más precisos (aunque es discutible hasta qué punto cabe llevar un enfoque cuantitativo preciso).

4. Metodología

Se preparó una encuesta encaminada a revelar la incidencia de los aspectos que se consideró eran más relevantes para los objetivos planteados. El esquema adoptado fue de una encuesta (relativamente extensa) con preguntas cerradas, en las que el alumno debía o bien dar datos concretos o bien calificar su actitud respecto a la identificación con determinada propuesta, en un esquema de tipo Likert. Todas y cada una de las preguntas llevaban, en cualquier caso, un área de texto libre para que el alumno pudiera consignar

cualquier opinión que no estuviera contemplada en las respuestas cerradas.

Respecto a la población, puesto que las asignaturas en cuestión las cursan todos los alumnos del centro en el primer curso, se dirigió una campaña de información a todos ellos, solicitando su colaboración, tanto mediante cartelería como mediante mensajes personales y a través del foro web del centro. 78 alumnos respondieron a este llamamiento, con el perfil que se describe en la Figura 2 (el número de matriculados en estas asignaturas en 2002-2003, por dar una aproximación, fue algo superior a 400). Se pidió a los alumnos que detallaran a qué promoción concreta se referían sus respuestas, lo que también se refleja en la figura.

Titulación:
I. T. I. Sistemas – 69,23%
I. T. I. Gestión – 30,77%

Sexo:
Hombres 75,64% - Mujeres 24,36%

Edad media:
21 años

Preferencia por este centro para el ingreso:
94,87% en primera opción.

Calificación media de ingreso:
6,78 en la PAU / 7,32 en el Bachiller.

Año académico:

37,18%	2002-2003
26,92%	2003-2004
16,67%	2004-2005
19,23%	2005-2006

Convocatoria:

	1ª	2ª	3ª
IP	78,20%	16,67%	2,57%
MP	69,23%	19,23%	10,26%
CPM	85,90%	10,26%	2,56%

Figura 2. Ficha de características y distribución general de los alumnos encuestados.

5. Extracto de los resultados

5.1. Motivación del alumnado

Respecto a la motivación, a casi todos los encuestados les gusta mucho o bastante la Informática (sólo el 2,57% manifestó que le gustase "poco o nada"). Centrándose en la programación en particular, las cifras son similares (sólo un 3,85% de indiferencia o rechazo). Yendo a las asignaturas concretas, las tres reflejan aproximadamente esta misma distribución.

Los alumnos encuestados, además, consideran que conocen las aplicaciones de estas asignaturas en el mundo laboral. El porcentaje de alumnos que creen conocer "muy bien" o "bastante bien" esta posible aplicación es del 73,08% para IP, 85,9% para MP y 83,33% para CPM. Perciben también las asignaturas como útiles; el porcentaje de alumnos que las consideran "poco o nada útiles" varía desde el 3,85% hasta el 7,7% dependiendo de la asignatura.

El principal motivo aducido para elegir la carrera es la vocación, seguido por la facilidad para encontrar trabajo. La curiosidad o las perspectivas económicas aparecen a continuación en la lista.

Resumiendo parcialmente los resultados relacionados con la motivación de los alumnos y la utilidad que perciben en estas asignaturas, parece evidente que en términos generales los alumnos encuestados responden de manera bastante positiva, cosa que se analizará en las conclusiones.

5.2. Las clases y las academias

Uno de los factores estrechamente relacionados con el absentismo es el valor añadido que los alumnos perciben en las clases [5]. Por tanto, parecía lógico indagar sobre su percepción de las mismas, antes de entrar en los motivos específicos de absentismo.

A la pregunta de si consideran las clases amenas, en términos generales no hay un gran rechazo. El porcentaje de alumnos que califica las clases como poco o nada amenas es el 15,38% para IP, 8,97% para MP y 21,79% para CPM. Un porcentaje notable, que se mueve alrededor del 30%-40%, mantiene una opinión neutral a este respecto, y el resto considera las clases muy amenas o bastante amenas.

Respecto a si las clases son participativas, hay un grado similar de opiniones neutrales, aunque el porcentaje de alumnos que las consideran poco o nada participativas ronda en general el 25%.

Y respecto a la posible existencia de mal ambiente en clase, no parecen encontrarse cómodos un 26% en IP, un 10% en MP y un 6% en CPM. En lo que se refiere al riesgo de que el alumno sea censurado en clase, dicen percibirlo

del orden del 40% para IP, 6,41% para MP y 3,84% para CPM.

Por otra parte, el profesorado suele sospechar que la asistencia a academias externas tiene un papel en el absentismo, usando algunos alumnos dichas academias como sustitutas de las clases (alrededor de la mitad de los encuestados consideran que las academias sustituyen las clases, y la otra mitad que las complementan). De los encuestados, el porcentaje de alumnos que utilizan estos servicios es en realidad el 19,23% para IP, 16,67% para MP y 11,54% para CPM. Consideran imprescindible esta ayuda para aprobar el 14,1% para IP, el 10,26% para MP y el 8,97% para CPM.

	IP	MP	CPM
Antes de empezar	0%	8,97%	3,85%
Principios de curso	6,41%	16,67%	2,56%
Mediados de curso	10,25%	17,95%	11,54%
No abandona	83,33%	56,41%	82,5%

Figura 3. Abandono de las asignaturas por los encuestados

5.3. El abandono

Evidentemente, es necesario poner los datos recogidos en relación con las variables de abandono y absentismo. Se comprueba, una vez más, lo difícil que resulta llegar a los alumnos que abandonan, ya que el porcentaje de abandonos reflejado en las encuestas es relativamente bajo. Se resume en la Figura 3. Esto nos lleva a apreciar que, si nos limitamos a alumnos que abandonen totalmente la asignatura, hemos podido encuestar a unos 13 alumnos para IP y CPM, y 34 para MP.

Por supuesto, permanece sin resolver el problema de obtener mucha información de los alumnos que *desaparecen* del proceso. Para conseguirlo, seguramente se necesita un esfuerzo mayor, empezando por una financiación adecuada del estudio. No obstante, en este caso se ha conseguido, al menos, una cierta cantidad de información de la que nunca se había dispuesto.

El mayor abandono se produce en MP. Poniendo estos datos en relación con los motivos de abandono directamente relacionados con el plan de estudios, se confirma la idea reflejada en la Figura 1: los alumnos que fracasan en materias de programación en el primer cuatrimestre con frecuencia tiran la toalla en el segundo. De hecho, ante esta impresión (antes de que se apoyase en datos) se adoptaron en su día algunas estrategias de captación (como el envío de cartas al inicio del cuatrimestre animando a los alumnos a acudir a estas asignaturas, bien ante la posibilidad de aprobarlas, o bien porque cursarlas facilitaría mucho la recuperación de IP). Este estudio ha confirmado nuestras sospechas, ya que más del 50% de los abandonos en MP se deben a haber abandonado (42,3%) o bien suspendido (11,53%) IP. De los abandonos en CPM, aunque son mucho menos frecuentes, el 41,02% también se debe a haber abandonado IP.

Dejando aparte la relación entre asignaturas, también se han recogido datos sobre motivos genéricos, cuyos resultados se resumen en la Figura 4. Se aprecian varios detalles significativos. El primero, que hay un claro patrón de respuestas; ante cada posible motivo, las respuestas se polarizan en las categorías “muy de acuerdo / muy en desacuerdo”. Esto puede apreciarse en la Figura 5, donde se muestran gráficamente las distribuciones para IP, que son similares para las demás asignaturas. Por otra parte, los resultados numéricos también son similares a grandes rasgos; ninguno de los motivos propuestos ha sido descartado ni aceptado de forma abrumadora por el conjunto de los alumnos, para ninguna de las asignaturas.

Esto nos lleva a interpretar que en realidad cada alumno tiene su propia combinación de motivos relevantes, bastante clara, pero esta combinación varía bastante de un alumno a otro dependiendo de su perfil. Como era de esperar, el problema del fracaso escolar no admite una interpretación sencilla basada en un solo problema que se pueda atacar con facilidad.

5.4. El absentismo

Los datos de asistencia (referidos a encuestas recibidas) aparecen en la Figura 6. Respecto a los motivos genéricos para el absentismo, los datos están en la Figura 7. Resulta significativo que, a diferencia de lo que ocurría con el abandono, donde casi todos los factores tenían simultáneamente aceptación y rechazo, en este caso se observa una clara distribución *asimétrica hacia la derecha*; es decir, muchos motivos son más bien rechazados que aceptados. Por tanto, o bien no se ha acertado con los posibles motivos propuestos y el absentismo obedece a otros, o bien las respuestas están mucho más concentradas.

	IP					MP					CPM				
	Muy de ac.	De ac.	Neutro	En desac.	Muy en des	Muy de ac.	De ac.	Neutro	En desac.	Muy en des	Muy de ac.	De ac.	Neutro	En desac.	Muy en des
Llevar otros aprobados a casa	37,18	6,41	5,13	14,1	37,18	39,74	10,26	10,26	8,97	30,77	38,46	5,13	7,69	7,69	41,03
Demasiado difícil de entender	41,03	11,54	11,54	10,26	25,64	48,72	16,67	10,26	2,57	21,79	38,46	3,85	14,1	7,69	35,9
Requerir demasiado trabajo	44,87	7,69	10,26	10,26	26,92	52,56	10,26	7,69	8,97	20,51	46,15	7,69	14,1	6,41	25,64
No gustar la asignatura	42,3	1,28	12,82	8,97	34,61	38,46	3,85	8,97	10,26	38,46	41,03	1,28	8,97	10,26	38,46
Mal impartida	44,87	6,41	10,26	10,26	23,08	37,18	2,57	11,54	8,97	39,74	41,03	2,57	12,82	11,54	32,05
Los profesores	47,44	8,97	11,54	8,97	23,08	41,03	8,97	5,13	8,97	35,9	43,59	2,57	8,97	10,26	34,61
Información de terceros ("rumores")	39,74	2,57	12,82	5,13	39,74	38,46	5,13	11,54	5,13	39,74	38,46	1,28	8,97	7,69	43,59
Método de evaluación	44,87	7,69	11,54	10,26	25,64	46,15	10,26	6,41	10,26	24,36	39,74	7,69	6,41	14,1	32,05
Tener ya apuntes	48,72	2,57	8,97	11,54	28,2	44,87	3,85	11,54	8,97	30,77	46,15	5,13	14,1	7,69	29,49
Las transparencias son suficiente	42,3	3,85	10,26	15,38	28,2	43,59	1,28	12,82	15,38	26,92	52,56	10,26	7,69	10,26	19,23

Figura 4. Causas de abandono. Porcentajes de alumnos que, habiendo abandonado la asignatura, se muestran muy de acuerdo / de acuerdo / posición neutra / en desacuerdo / muy en desacuerdo con la influencia de cada motivo en su decisión. (Símbolos % omitidos por claridad).

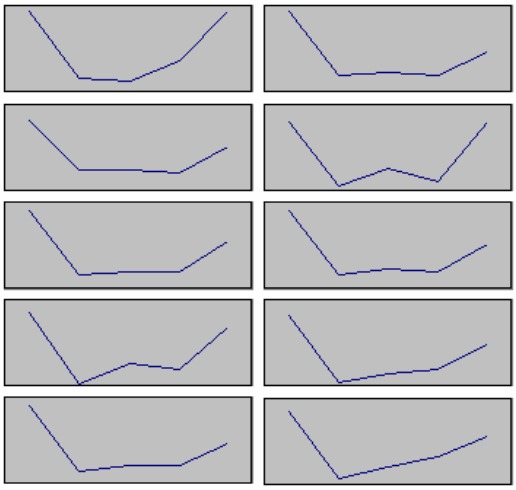


Figura 5. Distribuciones de aceptación de los 10 motivos de la Figura 4 para el abandono de IP (por columnas). En cada gráfica, el extremo izquierdo es *muy de acuerdo*, y el derecho *muy en desacuerdo*.

Destaca con claridad el trabajo sobre los demás motivos, y parece evidente que si los alumnos faltan a clase no es principalmente porque las clases no sean útiles, o porque resulten aburridas. Las respuestas abiertas dadas por los alumnos a esta cuestión, previstas para clarificar estas situaciones oscuras, no han aportado ideas demasiado relevantes u originales.

	IP	MP	CPM
Prácticamente nunca	11,54%	15,38%	17,95%
Días aislados	2,57%	10,26%	3,85%
La mitad de los días	7,69%	6,41%	19,23%
Más de la mitad	12,82%	11,54%	16,67%
Casi siempre	65,38%	56,41%	42,3%

Figura 6. Asistencia a clase

5.5. Cómo mejorar

Además de intentar diagnosticar la situación por nosotros mismos, se ha planteado también directamente a los alumnos la cuestión de cómo reducir los efectos del absentismo y el abandono.

Por ejemplo, para el caso de evitar el abandono, hay división de opiniones respecto a la conveniencia de hacer los contenidos más fáciles. Del orden de un tercio de los alumnos está muy de acuerdo, y una cantidad similar muy en desa-

cuerdo. Hay que reseñar que en las categorías "tibias" de la escala el porcentaje de alumnos "en desacuerdo" es mucho mayor que el de alumnos "de acuerdo". En resumen, no parece ser la mejor solución.

Respecto al cambio de profesores, tampoco parece la solución idónea. También hay porcentajes apreciables de "muy de acuerdo" con esa medida, dependiendo de la asignatura, pero el patrón general es que hay más alumnos en desacuerdo o muy en desacuerdo. Algo similar cabe decir sobre eliminar contenidos.

El control de asistencia también genera división de opiniones. Actualmente, la asistencia a clases prácticas se controla (y se valora positivamente) en estas asignaturas; la implantación de un sistema similar para las clases de teoría podría mitigar el abandono en opinión de porcentajes que rondan el 40%, pero este porcentaje también lo alcanzan los alumnos en desacuerdo o muy en desacuerdo.

Sin embargo, la idea de dar formación previa es la que más aceptación tiene, con una distribución claramente asimétrica. En todos los casos más del 50% de los alumnos (llegando al 60% en IP) están de acuerdo o muy de acuerdo con ella.

Otra propuesta con cierta aceptación (no tanta como la formación previa) es la de poner menos tareas. Es cierto que las tareas impuestas pueden ser un motivo por el que los alumnos abandonen, pero parece claro que eliminar esas tareas (y el trabajo continuo que suponen) implicaría muy probablemente cambiar el abandono temprano por más abandono tardío y fracaso, lo que no parece una buena solución.

Nuevamente, las respuestas de texto libre no añaden información muy relevante a las respuestas cerradas.

	IP					MP					CPM				
	Muy de ac.	De ac.	Neutro	En desac.	Muy en des	Muy de ac.	De ac.	Neutro	En desac.	Muy en des	Muy de ac.	De ac.	Neutro	En desac.	Muy en des
Clases inútiles	11,54	2,57	6,41	25,64	53,85	10,26	3,85	7,69	21,79	56,41	12,82	20,51	8,97	24,36	33,33
Clases aburridas / cansadas	14,1	14,1	20,51	21,79	29,49	11,54	15,38	20,51	25,64	26,92	14,1	28,2	19,23	19,23	19,23
Trato del profesor	24,36	8,97	14,1	17,95	34,61	11,54	2,57	10,26	16,67	58,97	8,97	2,57	1,28	25,64	61,54
Horario	16,67	10,26	17,95	14,1	41,03	15,38	8,97	20,51	14,1	41,03	14,1	8,97	21,79	15,38	39,74
Trabajo	44,87	7,69	10,26	10,26	26,92	52,56	10,26	7,69	8,97	20,51	46,15	7,69	14,1	6,41	25,64
Otros estudios	11,54	3,85	19,23	3,85	52,56	11,54	6,41	17,95	12,82	51,28	12,82	3,85	17,95	14,1	51,28
Academia	16,67	2,57	7,69	3,85	69,23	15,38	1,28	8,97	5,13	69,23	11,54	1,28	5,13	2,57	79,49

Figura 7. Algunas de las causas de absentismo. (Símbolos % omitidos por claridad).

6. Conclusiones

6.1. Consideraciones previas

En primer lugar, a efectos de delimitar la población para la que se han obtenido datos, cabe insistir en que estos no se limitan a alumnos que abandonan, sino que la distribución responde a la de la Figura 3 y la Figura 6, y además han decidido voluntariamente contestar al requerimiento de la encuesta.

Otra salvedad es que estamos hablando de alumnos que, en una abrumadora mayoría, eligieron no ya la titulación, sino el centro concreto (en nuestra Universidad hay dos) como primera opción, que eligieron la carrera por vocación, a los que los estudios de Informática les interesan (lo mismo cabe decir de la disciplina de la programación en particular) y que creen conocer y apreciar la utilidad de estas materias. Parece probable que haya una correlación entre este perfil tan positivo y la voluntad de responder a la encuesta, y que tal perfil no sea extrapolable a todos los matriculados.

6.2. Diagnóstico de la situación

Partiendo de este grupo, la calidad docente en general no parece ser un problema relevante. La

amenidad de las clases no es motivo de rechazo, aunque hay un margen de mejora. El grado de participación también puede aumentar bastante, pero está por ver que esto incida positivamente en la asistencia, ya que los datos y la experiencia revelan un cierto rechazo del alumno a esta participación (que interpreta, a veces, como un riesgo de "ser censurado" en clase y no como una oportunidad de aprendizaje).

Hay un número apreciable de alumnos que utilizan la ayuda de academias externas (entre el 10% y el 20%) y alrededor del 10% consideran tal ayuda imprescindible para aprobar (tratándose, insistimos, de estudiantes supuestamente motivados).

Respecto al abandono, está claro que los resultados en asignaturas previas pueden ser el factor decisivo en un 50% de los casos. Dejando este aparte, sin embargo, la respuesta a diversos factores genéricos es muy variada, y al parecer cada alumno tiene su propia combinación de motivos muy marcada, aduciendo unos con claridad y a la vez rechazando categóricamente otros. No ha sido posible aislar factores percibidos como especialmente relevantes en conjunto.

Respecto al absentismo, lo más destacable en comparación con el abandono es que hay mucho más rechazo que aceptación hacia los motivos sometidos a consideración. La única salvedad es que el estudiante trabaje además de estudiar, causa con la que sí se identifican claramente muchos de los alumnos encuestados. Evidentemente, esta tiene difícil solución por parte del profesorado.

6.3. Propuestas de mejora

Respecto a la opinión de los alumnos sobre las mejores opciones para mitigar el abandono y el absentismo, una pauta importante es que hay controversia en muchos factores, con lo que es de suponer que no existe ningún remedio global fácilmente identificable, sino que cada alumno tiene sus prioridades o intereses. Hacer contenidos más fáciles, cambiar a los profesores, eliminar contenidos o premiar la asistencia entran en esta categoría de factores discutidos.

La única medida que parece haber despertado cierto consenso es la formación previa. Los alumnos encuestados apoyan decididamente que con este recurso se podría combatir el abandono.

Esto, por supuesto, plantea un desafío de cierta envergadura. Aunque en muchas universidades existen *cursos cero* para facilitar la transición hacia la Universidad, lo cierto es que:

- Puede no resultar fácil implementar una preparación adecuada para materias como la programación en tan poco tiempo.
- Surge el debate, nada trivial, de si es más efectivo que la Universidad solucione posibles carencias formativas, o es a través de una colaboración y coordinación adecuada con las instituciones de enseñanza media como se puede paliar el problema de manera más eficiente.
- Tanto en un caso como en otro, estaríamos una vez más ante una mejora universitaria que requeriría recursos e inversiones, y no puede conseguirse, como por desgracia muchos gestores pretenden, a *coste cero*.

La otra medida que parece atractiva a los encuestados es imponerles menos tareas, pero aunque la supuesta abundancia de tareas (que no es tal [1]) sea una razón para el abandono parece evidente que las consecuencias negativas (menos trabajo continuo, peor preparación, y por tanto mayor abandono tardío y fracaso) superarían a las positivas.

Agradecimientos

Este trabajo ha recibido financiación del Vicerrectorado de Calidad e Innovación de la Universidad de Oviedo a través de su programa de proyectos de innovación docente en la convocatoria de 2005.

Referencias

[1] Cernuda del Río, Agustín; Gayo Avello, Daniel et al. *Análisis de los hábitos de trabajo autónomo de los alumnos de cara al sistema de créditos ECTS*. XI Jornadas de Enseñanza Universitaria de la Informática (JENUI 2005), Universidad Europea de Madrid. Actas de las jornadas, Ed. Thomson, ISBN 84-9732-421-8.

[2] García Guzmán, Javier; Sánchez Segura, María Isabel, et al. *Análisis experimental de la carga de trabajo requerida para completar una asignatura universitaria de cara a la transición hacia el Espacio Europeo de Educación Superior*. XII Jornadas de Enseñanza Universitaria de la Informática

(JENUI 2006), Universidad de Deusto (Bilbao). Actas de las jornadas, Ed. Thomson, ISBN 84-9732-545-1.

[3] Martín Cabrera, Eduardo; García García, Luis A.; Hernández Hernández, Pedro. *Determinantes de éxito y fracaso en la trayectoria del estudiante universitario*. Servicio de Publicaciones, Universidad de La Laguna, 1999. ISBN: 84-7756-485-X

[4] Posadas Yagüe, Juan Luis; Gómez Requena, Mª Engracia, et al. *Estudio de la carga de trabajo del alumnado en las titulaciones de ITIG e ITIS para la adaptación al EEES*. XII Jornadas de Enseñanza Universitaria de la Informática (JENUI 2006), Universidad de Deusto (Bilbao). Actas de las jornadas, Ed. Thomson, ISBN 84-9732-545-1.

[5] Rodríguez, Raquel; Hernández, Jesús et al. *El absentismo en la Universidad: resultados de una encuesta sobre motivos que señalan los estudiantes para no asistir a clase*. Aula Abierta, 82 (2003), 117-145. ISSN: 0210-2773

¿Cómo cambia a los alumnos la asignatura de Gestión de Sistemas Informáticos?

Daniel González Morales, José Luis Roda García, Luz Marina Moreno de Antonio
Escuela Técnica Superior de Ingeniería Informática
Universidad de La Laguna
Camino San Francisco de Paula, s/n Campus Anchieta, 38271 La Laguna
{dgonmor, jlroda, lmmoreno}@ull.es

Resumen

El aprendizaje implica un cambio. Desde el punto de vista de los profesores, nos parece que este cambio es importante, pero ¿qué opinan los alumnos?. Lo que se pretende es responder a esta pregunta tratando de evaluar el cambio que produce la asignatura en los alumnos. El objetivo es conocer la evolución de los alumnos entre el primer día y el último día de clase. No nos parece tan importante evaluar el conocimiento técnico adquirido sobre la gestión de proyectos, descriptor fundamental de la asignatura, que ya es evaluado de forma continua durante el desarrollo de la misma. El propósito es evaluar la influencia que ha tenido la asignatura en sus creencias, autovaloraciones, habilidades, expectativas, autoestima, etc.

Para ello, se ha modificado la encuesta del perfil del estudiante que se ha realizado otros años el primer día de clase, y que permite tener una primera valoración de los alumnos que realizan la asignatura. Se le ha añadido una serie de preguntas muy diversas que pretenden obtener información sobre sus creencias y autovaloraciones. Los alumnos completaron esta encuesta el primer y el último día del curso. Las conclusiones presentadas se han obtenido a partir de los resultados de estas dos encuestas.

En este trabajo se presenta el cuestionario y el proceso seguido, los datos más importantes sobre el perfil del estudiante, una breve descripción de las actividades realizadas durante el curso, los resultados obtenidos en ambas encuestas, las conclusiones más importantes, una forma de repetir la experiencia en otras asignaturas y las conclusiones finales.

1. Motivación

Los objetivos que nos planteamos en la asignatura son tres:

- Lograr que los alumnos obtengan un conocimiento técnico sobre la gestión de proyectos.
- Introducir a los alumnos en algunas de las habilidades sociales necesarias para la efectiva gestión de proyectos. Fundamentalmente, trabajo en equipo, relaciones interpersonales, liderazgo, gestión de expectativas del cliente, gestión del cliente, dirección de equipos, presentación de resultados, redacción de informes y valoración del trabajo y el rendimiento.
- Producir un cambio de paradigma en los alumnos. Estos se ven a si mismos como programadores, en gran parte debido al propio plan de estudios [1]. Nuestra intención es que se vean capaces de ejercer otro tipo de perfiles profesionales.

Creemos que logramos cumplir los dos primeros objetivos y disponemos de mecanismos de evaluación de los resultados. Sin embargo, hasta el momento desconocíamos lo que pasaba con el tercer objetivo, debido a que no teníamos un instrumento de medida. Disponer de éste es básico para poner en práctica actividades de mejora, cuyos resultados puedan ser evaluables.

La experiencia se realizó en la asignatura de Gestión de Sistemas Informáticos (GSI), que tiene 6 créditos (3 teóricos y 3 prácticos), y se imparte en el primer cuatrimestre del quinto curso de la Ingeniería en Informática.

En la siguiente sección se muestra el cuestionario utilizado y el proceso seguido. En la sección tres se describen los resultados del perfil

de estudiante extraído de dicha encuesta que creemos que afectan de forma más significativa a la asignatura. Las actividades desarrolladas durante la asignatura se muestran, muy brevemente, en la sección cuatro. En la cinco, presentamos los resultados más importantes que hemos obtenido a partir de los datos de las encuestas. En la sección seis, describimos el proceso a seguir para repetir la experiencia en otras asignaturas. Por último, en la sección siete se comentan las conclusiones más importantes y los trabajos futuros.

2. El Cuestionario y el Proceso

En el curso 2005/2006 se pretendió que los alumnos autoevaluaran el cambio producido. Para ello se utilizó una encuesta con preguntas directas sobre los cambios. Sin embargo, los resultados no fueron satisfactorios. En gran parte, debido a que los alumnos no son concientes aún del cambio producido y la encuesta realizaba preguntas directas sobre el mismo.

El proceso que hemos seguido este curso para el desarrollo del estudio se muestra a continuación.

En primer lugar se modificó la encuesta de perfil de estudiante, que se viene realizando desde hace varios años, para añadirle cuestiones mediante la que pudiéramos evaluar el cambio en los estudiantes.

El primer día de clase se les entregó a los alumnos la encuesta en formato papel. La encuesta está disponible en la página web del grupo Taro http://www.taro.ull.es/docencia.

El cuestionario se compone de una serie de preguntas orientadas a evaluar el perfil de estudiante [2], las expectativas una vez finalizados los estudios, las apreciaciones sobre sus propias competencias y sobre las competencias que deberían tener determinados perfiles profesionales y la respuesta ante determinados supuestos. Al finalizar la encuesta se indicó a los alumnos que escribieran una marca que solamente ellos fueran capaces de identificar, garantizando el anonimato.

Durante el curso se realizaron las actividades que se describen en la sección cuatro. En ningún momento se comentó nada sobre la encuesta o sus resultados.

El último día del curso, se realizó la segunda encuesta. Algunas preguntas cuya respuesta lógicamente iba a ser la misma se eliminaron. Posteriormente, se les presentaron las encuestas realizadas a principio de curso y cada alumno, mediante su marca, identificó la suya. Los alumnos realizaron una comparación entre las dos encuestas tratando de identificar los cambios. Estos resultados fueron comentados en clase y permiten, por un lado, que los alumnos tomen conciencia sobre los cambios producidos y como elemento captador de información complementaria sobre las apreciaciones de los alumnos.

3. Perfil del Estudiante

Uno de los resultados más útiles de la primera encuesta es que permite conocer el perfil de los estudiantes y orientar las actividades durante el curso. Los resultados que creemos que tienen más influencia en el correcto desarrollo de la asignatura son los siguientes.

En el curso 2006/2007 han realizado la asignatura 37 alumnos, de los que el 82% son hombres y el 18% mujeres. Su edad media es de 24 años.

El 76% realizó la Ingeniería Técnica en Informática de Sistemas (ITIS) y el 24% restante la Ingeniería Técnica en Informática de Gestión (ITIG).

Respecto a los conocimientos previos, el 70% realizó la asignatura Ingeniería del Software de Gestión I, el 30% Ingeniería del Software de Gestión II, ambas se cursan en el tercer curso. Un resultado importante es que el 30% de los alumnos no realizó ninguna de las dos asignaturas de ingeniería del software en el ciclo medio. En el cuarto curso existe otra asignatura llamada Ingeniería del Software que ha realizado el 73% de los alumnos.

Otro resultado que creemos que tiene una influencia importante en el desarrollo de la asignatura es que el 30% de los alumnos trabaja y otro 15% de los mismos es becario de la propia universidad.

En la figura 1 se muestra el número de asignaturas pendientes de cursos anteriores. Únicamente el 33% de los alumnos se dedica en exclusividad a las asignaturas del curso presente. Nos parece importante que un 25% de los alumnos tengan más de tres asignaturas pendientes de otros cursos.

Los datos que se describen en los dos párrafos anteriores sobre el alto índice de alumnos que trabajan y el de alumnos con asignaturas pendientes influyen de forma muy negativa en la dinámica de los grupos de trabajo.

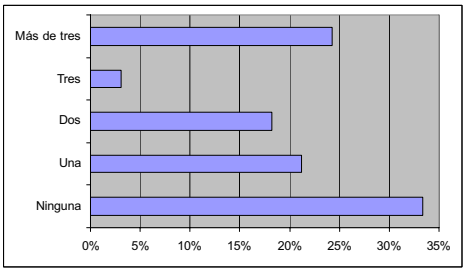


Figura 1. Número de asignaturas pendientes de cursos anteriores.

4. Las Actividades Desarrolladas Durante el Curso

Para el desarrollo de las competencias técnicas sobre la gestión de proyectos se utiliza PMBOK (http://www.pmi.org). Los alumnos deben leer cada semana un tema y responder a un cuestionario online, todos los viernes, sobre los conocimientos de dicho tema. Durante el curso se realizan 10 test de este tipo que tienen un peso del 40% de la calificación.

Para el desarrollo de las actividades se utiliza la herramienta moodle de forma intensiva. Existen dos tipos de actividades [3]: las competitivas y las no competitivas. Las competitivas se realizan en grupos de 12 a 15 alumnos dirigidos por un alumno-jefe. El objetivo de estas actividades es la realización de un informe sobre un tema determinado, su presentación y su defensa. Estas se desarrollan fundamentalmente online y en las prácticas de laboratorio. La actividad comienza con la entrega del guión de la actividad por parte del profesor. La gestión de estas actividades corresponden únicamente a los alumnos. El profesor interviene únicamente a solicitud del alumno-jefe. Las clases de laboratorio se dedican fundamentalmente a la planificación de los trabajos, la organización interna y la puesta en común de las ideas fundamentales para el desarrollo de la actividad. La última parte de la actividad consiste en la presentación pública del informe en la clase teórica. Cada grupo selecciona y prepara a un presentador. Para esto se les suministra una guía sobre presentaciones. Una vez todos los grupos han realizado sus presentaciones se realiza un turno de preguntas. En algunos casos, cada grupo plantea a los otros, cuestiones que previamente han preparado. Otra posibilidad consiste en invitar a esta clase a especialistas en el tema concreto que planteen las preguntas a los grupos.

La evaluación de estas actividades corresponde fundamentalmente al alumno-jefe, y en menor medida al profesor. El profesor evalúa los trabajos presentados por los grupos y únicamente el mejor evaluado se lleva la puntuación. El resto de alumnos evalúan al alumno-jefe. La actividad es competitiva porque sólo uno de los tres grupos recibe puntuación.

Estas actividades están más orientadas al desarrollo de habilidades sociales para la gestión de proyectos. Al final de cada una de las actividades de este tipo, se realiza una encuesta entre los alumnos con cuestiones de muy diversa índole, como por ejemplo: ¿Cuánto tiempo han dedicado a la actividad? ¿Qué puntuación le ponen al jefe? ¿Cómo valoran el control de la actividad por parte del profesor?

La última actividad es no competitiva y consiste en desarrollar un plan de gestión de un proyecto real en equipos de cuatro alumnos. Durante la asignatura de GSI planifican los trabajos a realizar y durante la asignatura de Laboratorio de Ingeniería del Software, que se imparte en el segundo cuatrimestre, lo ejecutan. En esta actividad se desarrollan fundamentalmente las competencias de gestión de los usuarios.

La evaluación de los alumnos es un sistema bastante complejo que se compone de 29 valoraciones con sobrenota, distribuidas en tres categorías:

- Actividades voluntarias. Los alumnos las desarrollan de forma voluntaria. En general, son evaluadas por el profesor.
- Actividades obligatorias. Los alumnos deben realizar las actividades y, como mínimo, aprobarlas. Los alumnos deben aprobar todas las actividades obligatorias para aprobar la asignatura.
- Actividades competitivas. Los alumnos deben realizar las actividades, pero sólo los miembros del grupo ganador reciben

puntuación. Los otros grupos no reciben puntos.

Inicialmente, podría parecer que realizar la evaluación de la asignatura es una carga de trabajo enorme. Sin embargo, se usan algunas técnicas para reducirlo.

Los test se realizan en la plataforma moodle, tienen un coste importante en la realización las preguntas, pero el sistema realiza la evaluación automática e instantáneamente, lo que produce una realimentación inmediata.

Se usan las revisiones técnico-formales de la misma forma que en el entorno empresarial, unos equipos revisan el trabajo de los otros. El profesor valora únicamente el producto final, que debido al proceso tiene, en general, una alta calidad.

La evaluación se realiza de forma continua a través de la herramienta moodle. Esto permite, por un lado, una disponibilidad total, y por otro lado, encajar más fácilmente la evaluación en el calendario del profesor.

Y por último, algunas evaluaciones la realizan los propios alumnos utilizando fundamentalmente técnicas de coevaluación. Gran parte de esta coevaluación es realizada por los alumnos-jefes.

La distribución de las calificaciones obtenidas por los alumnos en el curso 2006/2007 es la siguiente:

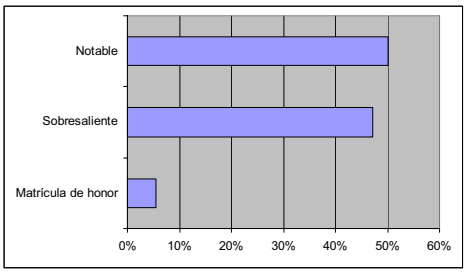


Figura 2. Calificaciones obtenidas en el curso 2006/2007

5. Los Resultados Obtenidos

En esta sección se muestran los resultados más importantes obtenidos de la comparación entre las dos encuestas realizadas.

Respecto a las expectativas de futuro nada más finalizar los estudios, los resultados se muestran en la figura 3.

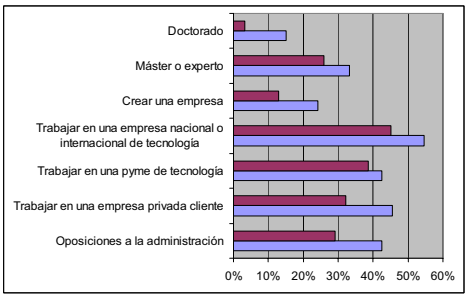


Figura 3. Expectativas de futuro tan pronto finalicen los estudios

En esta pregunta los alumnos pueden marcar varias respuestas. Se muestra el porcentaje de respuestas respecto al total de alumnos. En color azul o claro, los datos al comienzo del curso, y en color rojo u oscuro, los datos al final del curso. Esto es válido para todas las figuras en adelante, a no ser que se indique lo contrario.

Como se puede observar se ha producido una reducción significativa del porcentaje de respuestas, lo que indica que los alumnos tienen más claras sus expectativas de futuro al final del cuatrimestre. Resultado que era de esperar.

Sus mayores expectativas, entorno al 40%, están centradas en trabajar en una empresa de tecnología: nacional, internacional o PYME. En segundo lugar, trabajar dentro de la empresa cliente, sea en una empresa privada o en la administración, estas opciones son seleccionadas aproximadamente por el 30% de los alumnos. Continuar formándose en un master o experto es la opción elegida por el 26% de los alumnos. Sin embargo, los programas de doctorado despiertan poco interés, únicamente el 3% de los alumnos. Además hay que destacar la importante caída de esta expectativa de futuro entre el inicio (15%) y el final de curso (3%). Crear una empresa es seleccionada por el 13% de los alumnos.

Otra de las cuestiones que se les planteó respecto a su futuro profesional fue cuáles eran sus preferencias respecto a los diferentes perfiles profesionales. En la figura 4 se muestran los resultados al inicio y al final del cuatrimestre.

Como se puede observar, las preferencias de casi todos los perfiles profesionales se reducen al final del cuatrimestre respecto al inicio, excepto

en dos categorías: jefe de proyectos y consultor informático.

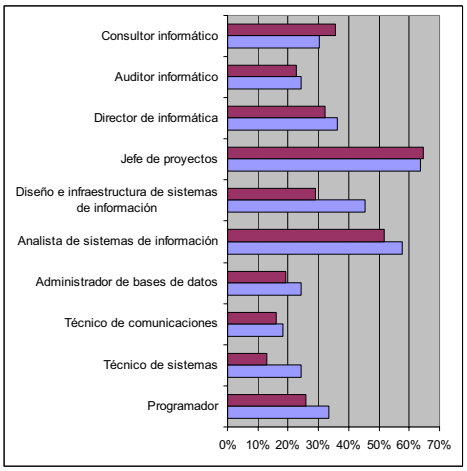


Figura 4. Preferencias respecto a los perfiles profesionales

A más del 60% de los alumnos les gustaría ejercer de jefe se proyectos. La segunda categoría con más expectativas es analista de sistemas de información, 52%. El siguiente grupo lo componen: consultor informático, director de informática, y diseño e infraestructura de sistemas de información, que se encuentran en torno al 30%.

La categoría de programador es preferida por un 26% de los alumnos. Este dato fue objeto de una importante reflexión en la última clase. No se entiende para qué estudian una ingeniería superior cuando les es suficiente una titulación de enseñanzas medias.

Las categorías de administrador de bases de datos y auditor informático están en torno al 20%.

Por último, las categorías de técnico de sistemas y técnico de comunicaciones se sitúan entorno al 15%. Este dato nos parece sumamente contradictorio respecto a la titulación obtenida por los alumnos en el ciclo medio: el 76% realizó la ITIS, frente al 24% que la realizó en ITIG.

En la figura 5 se muestran los perfiles profesionales para los que los alumnos se sienten preparados.

El perfil para el que los alumnos se sienten más preparados es el de programador. Sin embargo, a pesar de la importante carga lectiva de los planes de estudio para desarrollar las competencias de este perfil es curioso que no se alcance el 100%. Más sorprendente es el descenso de este perfil en la encuesta realizada a final de curso, que desciende del 94% al 84%. Creemos que esta reducción se debe a una muestra de desinterés por este perfil profesional debido a la influencia de la asignatura, en la que se trabajan competencias de otras categorías profesionales.

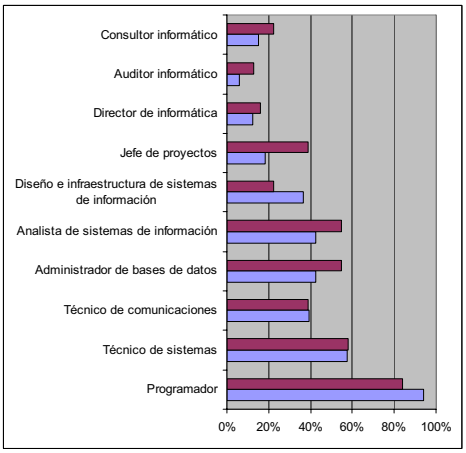


Figura 5. Perfiles profesionales para los que se sienten preparados

Las categorías de técnico de sistemas y técnico de comunicaciones mantienen sus respectivos porcentajes entre las encuestas inicial y final. Es importante resaltar que durante el cuatrimestre no hay ninguna asignatura que desarrolle las competencias de estos perfiles.

El perfil de técnico se sistema se mantiene como segunda preferencia, sin embargo, al final de la asignatura es alcanzado por los perfiles de administrador de bases de datos y análisis de sistemas de información. Estos dos han experimentado una subida importante. Respecto al segundo perfil entendemos que se debe a la influencia de la asignatura. Recordemos que un porcentaje importante de alumnos apenas había desarrollados estas competencias. Sin embargo, no tenemos una explicación para la subida de administrador de bases de datos. Además en este cuatrimestre no existen otras asignaturas que desarrollen estas competencias.

En torno a un 40% de preferencias se encuentran los perfiles de técnico de

comunicaciones y jefe de proyectos. La subida más importante corresponde a esta última categoría que pasa de un 18% a un 39%. Entendemos que debido al desarrollo de estas competencias en la asignatura de GSI.

En la figura 6 se muestra una comparativa entre los perfiles profesionales que les gustaría ejercer, en color rojo u oscuro, y que se creen capaces de ejercer, en color amarillo o claro. Todos los datos corresponden a la encuesta realizada a final del primer cuatrimestre.

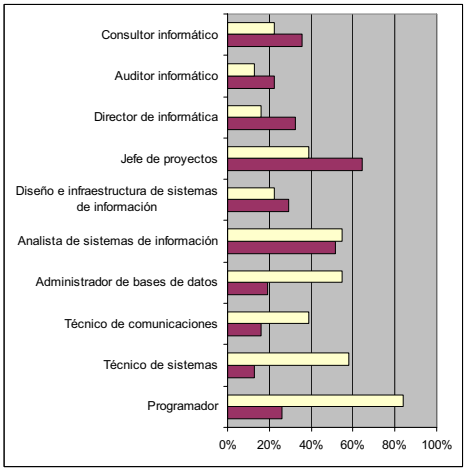


Figura 6. Comparativa entre los perfiles profesionales que les gustaría ejercer y para los que se sienten preparados

Podemos agrupar los perfiles profesionales en tres grupos. El primero de ellos está compuesto por los perfiles para los que los alumnos se sienten preparados, pero sin embargo, no les gustaría ejercer. Lo forman las categorías de programador, técnico de sistemas, técnico de comunicaciones y administrador de bases de datos. El segundo grupo se compone de los perfiles en los que coinciden aproximadamente el número de alumnos que creen estar preparados para ejercerlos y el número de alumnos que les gustaría ejercerlos. Se trata de diseño e infraestructura de sistemas de información y análisis de sistemas de información. Por último, tenemos el grupo de las categorías profesionales donde los alumnos se sienten que no están suficientemente preparados para ejercer. Se trata de consultor informático, auditor informático, director de informática y jefe de proyecto.

En la encuesta se incluyeron una serie de supuestos que tratan de evaluar la percepción que tienen los alumnos de si mismos en cuanto a sus capacidades. Mostramos algunas de ellas.

En la figura 7 se muestran las respuestas a la pregunta: "Imagínate que tienes que convencer a un usuario cabreado de que el producto previsto entregar este mes se va a retrasar seis meses. ¿Crees que lograrías convencerlo?"

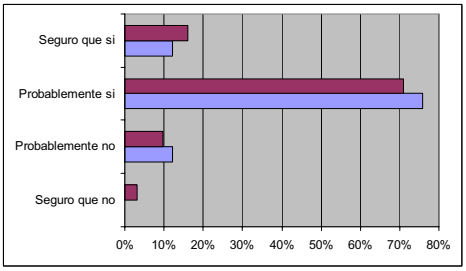


Figura 7. Respuestas a cuestión sobre gestión de conflictos con el usuario

Es un sorprendente el nivel tan alto de respuestas afirmativas. Además, las respuestas prácticamente no varían entre el inicio y el final del curso. Creemos que esto es debido a que en la asignatura de GSI se explica lo importante que es la gestión de usuarios, pero no se imparten técnicas de gestión de conflictos, negociación, gestión de expectativas, etc.

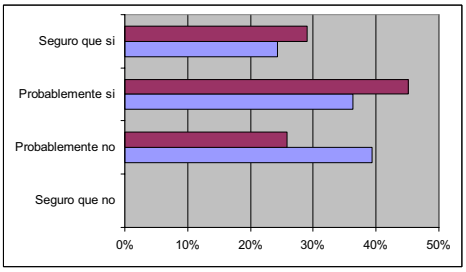


Figura 8. Respuestas a la cuestión sobre presentación de un proyecto

En la figura 8 se muestran las respuestas a la pregunta: "Imagínate que trabajas para una empresa que ha desarrollado un plan de informatización de las bibliotecas y tienes que

presentarlo al Consejo de Gobierno de la Universidad de La Laguna. ¿Crees que estás preparado para realizar la presentación?".

La mayoría de los alumnos se creen capaces de realizar la presentación. Además, los resultados han mejorado entre el inicio y el final del curso, la suma de las dos respuestas afirmativas han pasado del 60% al 75%. Creemos que este cambio se debe al desarrollo de las competencias sobre presentaciones que se han realizado tanto en la asignatura GSI como en otras del mismo cuatrimestre.

Otra de las situaciones que se les planteó fue la siguiente: "Imagínate que un ayuntamiento contacta con tu empresa para que le presupueste un proyecto consistente en la implantación de la firma electrónica en la expedición de los certificados, y te lo encargan a ti. ¿Crees que podrías hacer el presupuesto?". Las respuestas de los alumnos se muestran en la siguiente figura.

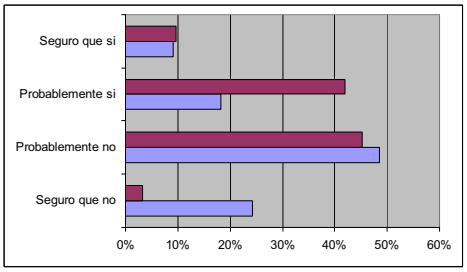


Figura 9. Respuesta a la pregunta sobre la realización de un presupuesto

Como se puede observan hay un cambio importante entre las respuestas al inicio y final de la asignatura. La respuesta "seguro que no" se reduce aproximadamente en un 20%, mientras que la respuesta "probablemente sí" se incrementa en el mismo porcentaje. Al final de la asignatura, aproximadamente el 50% de los alumnos se creen capaces de realizar el presupuesto. Realmente nos parece importante el otro 50% que no se cree capaz de realizar el presupuesto. Esta actividad, fundamentalmente técnica, se desarrolló ampliamente durante la asignatura con casos prácticos reales en los que los alumnos fueron evaluados muy positivamente. Sin embargo, desde su punto de vista no se creen capaces de realizar ese trabajo.

En la siguiente figura se muestran las respuestas a la pregunta "¿Cuál de las siguientes características crees que son más importantes en el perfil profesional de jefe de proyectos?"

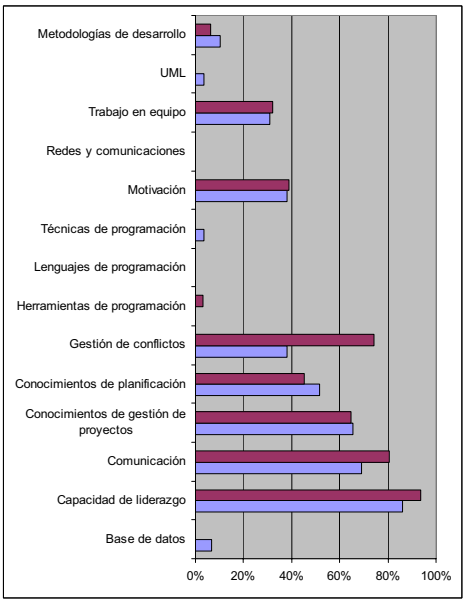


Figura 10. Características del perfil jefe de proyectos.

Como se puede observar, los resultados al inicio y final de curso son aproximadamente los mismos, a excepción de la habilidad de gestión de conflictos, que pasa de un 38% a un 74%. Creemos que este cambio se debe a su propia experiencia en las actividades competitivas de la asignatura GSI, en las que precisamente esta característica de competitividad creó ciertas situaciones de conflicto dentro de los grupos en cuanto a qué solución era la más adecuada al problema planteado, como plantear la solución y como realizar la presentación de los resultados. Estos conflictos fueron resueltos por los alumnos-jefes de forma adecuada, por lo que no fue necesaria la intervención del profesor.

Otras características que suben entre el inicio y el final del curso son las siguientes: capacidad de liderazgo, comunicación, motivación y trabajo en equipo. Por otro lado, bajan muy poco: conocimientos de planificación, conocimiento de gestión de proyectos y metodologías de desarrollo.

Curiosamente, en general, suben las habilidades sociales y bajan los conocimientos técnicos. Creemos que esto se debe a su propia experiencia en el desarrollo de las actividades de la asignatura.

6. Aplicabilidad a Otras Asignaturas

La experiencia puede ser aplicable a otras asignaturas. Inicialmente, basta con copiar las cuestiones de la encuesta del perfil del estudiante no dependientes de la asignatura GSI. El proceso más complicado consiste en definir cuales son los objetivos que queremos alcanzar y plantear una serie de preguntas, que de forma indirecta nos permita obtener información sobre dichos objetivos. En este sentido, recomendamos tener en cuenta los siguientes elementos:

- No realizar preguntas directas sobre los cambios sufridos por el alumno, ya que algunas veces son incapaces de identificarlos.
- No realizar preguntas sobre los conocimientos que ya se evalúan en la asignatura. Se trata de una encuesta no de un examen.
- Orientar las cuestiones hacia situaciones prácticas lo más reales posible en las que tengan claro que conocimiento o habilidad aprendido tienen que aplicar, y preguntar sobre si serían capaces de resolver el problema.
- A la hora de describir las situaciones prácticas debemos intentar definirlas desde su punto de vista. Por ejemplo, las valoraciones que hacemos los profesores y los alumnos sobre lo difícil que es un problema pueden ser muy diferentes.
- Utilizar escalas de valoración de tipo Likert [4]

7. Conclusiones y Trabajos Futuros

Como se comentó en la introducción, el objetivo de este trabajo es evaluar el cambio que produce en los alumnos la asignatura de Gestión de Sistemas Informáticos.

Creemos que hemos influido positivamente en la autoestima de los alumnos acercándose las apreciaciones que tienen los alumnos sobre si mismos a las apreciaciones que tienen los profesores sobre lo que son capaces de realizar.

Los alumnos comienzan a darle importancia a otros elementos competenciales, como las relaciones interpersonales.

Creemos que en algunos aspectos, la encuesta se puede haber visto influenciada por realizarse en una asignatura concreta. Los alumnos puede que hayan dado respuestas políticamente correctas en el marco de la misma. Nos parece interesante que la encuesta se realice a nivel de centro. En este caso, se debería complementar con cuestiones sobre las competencias que desarrollan el resto de asignaturas.

En estos momentos, no podemos emparejar las dos encuestas de un mismo alumno. Esto nos permitiría disponer de más información sobre los cambios experimentados. Una forma fácil sería que en la última clase, los alumnos escriban un mismo símbolo en las dos encuestas.

Nos parece importante desarrollar actividades destinadas a mejorar la expectativa de crear una empresa una vez finalicen los estudios. Creemos que los alumnos tienen un importante desconocimiento sobre este tema.

Referencias

[1] Luis Fernández, M. J. et al. *Análisis de empleabilidad y perfiles de egresado en las titulaciones de informática en España*. JENUI 2006.

[2] Villar Angulo, L. M. *Programa para la mejora de la docencia universitaria*. Pearson Prentice Hall, 2004.

[3] González Morales, D., Roda García, J. L. *Aprendiendo a ser jefes*. JENUI 2006.

[4] *Guía de orientación para la realización de estudios de análisis de la demanda y de encuestas de satisfacción.* Subdirección General de Calidad de Servicios. Ministerio de Administraciones Públicas.

Trabajos fin de carrera, practicum y participación de alumnos en la investigación

La automatización de un laboratorio de ingeniería química para uso docente: una plataforma multidisciplinar para la realización de trabajos de fin de carrera e investigación

Gerard Escudero[1], Moisès Graells[2], Josep M. Guerrero[3] y Sebastián Tornil[3]
Escola Universitaria d'Enginyeria Tècnica Industrial de Barcelona
[1] Dpto. Llenguatges i Sistemas Informàtics
[2] Dpto. Enginyeria Química
[3] Dpto. Enginyeria de Sistemes, Automàtica i Informàtica Industrial
Universitat Politècnica de Catalunya
Urgell 187, 08036 Barcelona
{gerard.escudero,moises.graells,josep.m.guerrero,sebastian.tornil}@upc.edu

Resumen

Este artículo describe cómo un proyecto a largo plazo, la informatización de los laboratorios docentes de ingeniería química, puede ser utilizado para que alumnos de distintas especialidades de Ingeniería Técnica Industrial hagan trabajos de fin de carrera de carácter multidisciplinar, con objetivos claros y que a la vez tengan una aplicación práctica evidente. Algunos de estos proyectos incluso pueden ser llevados al ámbito de la investigación.

1. Introducción y motivación

La propuesta de este artículo se enmarca dentro de las titulaciones de ingeniería técnica industrial en electrónica y en química industrial de la *Escola Universitària d'Enginyeria Tècnica Industrial de Barcelona* (EUETIB), adscrita a la *Universitat Politècnica de Catalunya* (UPC). Los autores, pertenecientes a diferentes departamentos de esta universidad, forman un grupo multidisciplinar[1] que tiene como uno de sus objetivos la automatización del laboratorio de química del centro.

Durante los años de impartición de las respectivas asignaturas y la dirección de diferentes trabajos de fin de carrera (TFC), a los autores les han surgido una serie de inquietudes. Por un lado, dentro del ámbito de la ingeniería química se da el caso de que los alumnos realizan prácticas y proyectos de laboratorio en un entorno totalmente tradicional. Por ejemplo, la medida de una temperatura se realiza mediante un termómetro de mercurio y se anota en un papel. Esta es una opción barata y también didáctica, pero no debería ser la única. En un ámbito industrial real se utilizaría una sonda de temperatura conectada a una tarjeta de adquisición, la cual permitiría el acceso a estos datos y facilitaría su posterior procesamiento (almacenamiento, monitorización, control...).

Por otro lado, muchos de los TFCs desarrollados en la especialidad de electrónica [1,2,3,4,5] se centran en la realización de placas electrónicas conectables a un ordenador personal vía algún puerto o bus (serie, paralelo, USB, PCI...). Muchas de ellas son tarjetas de adquisición de datos de propósito general, para las cuales no siempre es fácil encontrar procesos realistas a los que aplicarlas.

Dadas estas inquietudes, en el año 2003 se propusieron y realizaron una serie de proyectos de fin de carrera multidisciplinares [6,7,8], donde alumnos de las especialidades de química y electrónica trabajaron de forma conjunta para abordar proyectos de monitorización y control de procesos químicos. En este marco, los alumnos de electrónica tenían un proceso realista que instrumentalizar y los de química las herramientas para implementar el control del proceso.

La buena marcha de estos trabajos y los resultados obtenidos hicieron que los autores de este artículo se plantearan el proyecto a largo plazo para mejorar la calidad docente descrito en la siguiente sección. Este objetivo se recogió entre los compromisos asumidos por el profesorado en el proyecto piloto de Planificación Estratégica de

[1] Centro para la Integración, Automatización y Optimización de procesos químicos (CIAO): http://ciao.euetib.upc.es

las asignaturas de Experimentación en Ingeniería Química (PEEEQ 2003-2006) [9].

La sección 2 está dedicada a la descripción del proyecto que nos sirve de plataforma para la realización de estos TFC, la sección 3 nos describe el trabajo realizado y los que vienen a corto plazo y la sección 4 concluye el trabajo.

2. Descripción del proyecto

El laboratorio de ingeniería química de la EUETIB tiene una serie de equipos para la realización de experimentos químicos (baños termostáticos, columnas de destilación, fermentadores...). La figura 1 muestra una de las columnas de destilación del laboratorio.

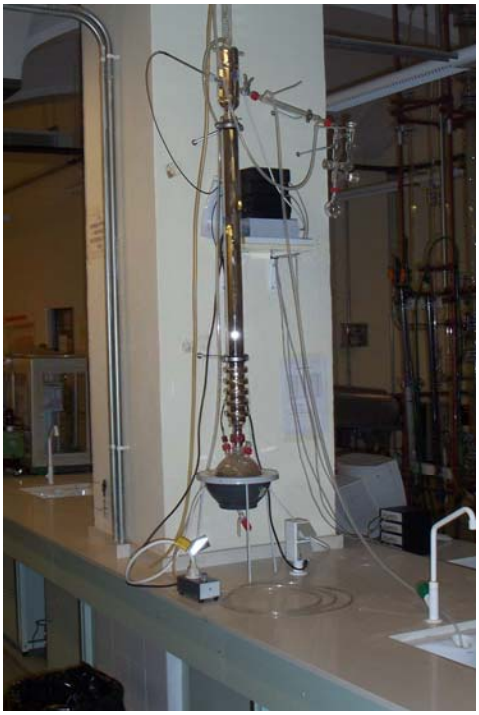


Figura 1. Columna de destilación

El objetivo final del proyecto es doble. Por un lado queremos desarrollar un laboratorio virtual y uno remoto de procesos químicos del estilo del que se describe en [10] para simular y controlar el movimiento de un brazo mecánico en una asignatura de robótica. Por otro lado queremos crear los instrumentos para que los alumnos de ingeniería química puedan realizar prácticas de informática industrial a todos los niveles, es decir, desde instrumentar uno de los equipos citados a programar la monitorización y/o control de los equipos. Los dos subapartados de esta sección están dedicados a explicar estos dos objetivos con mayor profundidad.

2.1. Automática e instrumentación industrial para ingenieros químicos

Se quiere que los alumnos de ingeniería química, sin conocimientos de electrónica y escasos de programación (la necesidad de esta formación debería revisarse), vean la problemática y el funcionamiento del entorno de la informática industrial. Para conseguir este objetivo se ha pensado en montar algún dispositivo altamente usable que nos permita la instrumentación de los equipos químicos.

El principal problema para lograr esto, dada la experiencia de los TFC anteriores, es la escasez de conocimientos de electrónica que tienen los alumnos de química. Los primeros TFC se resolvieron creando una placa exclusiva para cada equipo químico. Esto nos genera tres grandes inconvenientes: 1) la rigidez, el equipo queda montado y fijo y los alumnos no tienen capacidad para manejarlo; 2) el mantenimiento, la necesidad de tener a un electrónico a mano cuando se estropea algún componente de la placa; y 3) la falta de escalabilidad, el tener que realizar una placa para cada equipo químico o cuando queremos añadir alguna nueva variable al proceso.

Para solucionar estos problemas se diseño e implementó una placa simple a la que pudiéramos conectar un único sensor o actuador y que funciona vía el puerto USB [11]. La figura 2 muestra el esquema de conexión de un proceso con dos sensores y dos actuadores, y la figura 3 una foto del resultado de la implementación de dicha placa. Para este caso necesitaríamos cinco placas. La etapa de acondicionamiento de la señal de los sensores no está incluida en el circuito de conexión al USB. Esta está incluida en el recuadro sensor. Este recuadro representa un circuito con la etapa de acondicionamiento a la que se conecta la sonda. Tendremos un diseño diferente para cada tipo de sensor (temperatura, presión…). Para los actuadores tendremos un modelo parecido. Todos

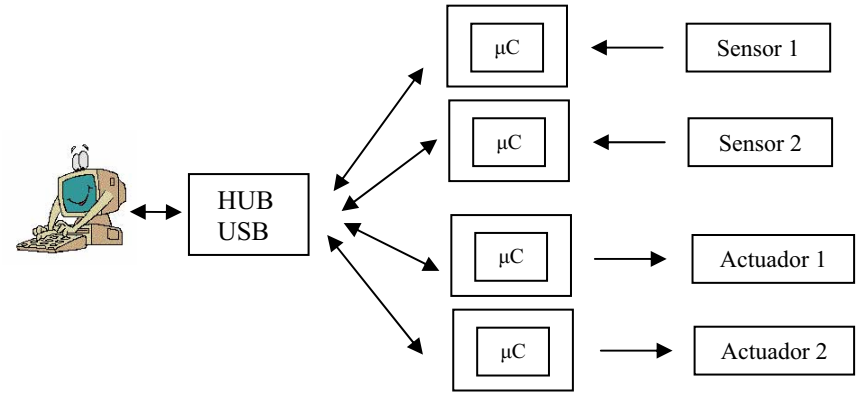


Figura 2. Esquema de conexión

estos dispositivos tienen la conexión a la placa USB.

Las ventajas de este modo de actuar son: 1) la usabilidad; un ingeniero químico al que se le proporcionan estos dispositivos –tipo *plug & play*– es capaz de conectar sin complicaciones un equipo químico a un ordenador personal. 2) El mantenimiento; en caso de defecto en un componente se puede cambiar la placa entera y listos. Y 3) la escalabilidad, si queremos añadir una variable más a un equipo o instrumentar otro equipo, sólo tenemos que coger más placas y sensores y/o actuadores y listo. No hay necesidad de disponer de un electrónico.

El inconveniente principal de este diseño es el precio de la replicación de componentes, aunque no es excesivo. Otro punto discutible del diseño es la seguridad industrial al usar la interfaz USB y no un bus industrial. Este hecho no creemos que se importante, dado que el diseño está pensado para ser utilizado en entornos docentes o en entornos de fabricación flexible para especialidades de demanda reducida y gran valor añadido (química fina, farmacéutica, etc.).

Esta placa se controla desde el ordenador personal vía los controladores USB de *Human Input Device* (HID). Existen librerías y/o controladores para todos los lenguajes de programación actuales para manipular este tipo de dispositivos. En nuestro caso, estamos utilizando el entorno Delphi de Borland[2], que tiene una versión para uso no comercial que llaman *Personal Edition*, y una librería de componentes de libre distribución conocida como el proyecto *Jedi*[3]. Esta librería de componentes Delphi contiene cientos de componentes de uso general, uno de ellos nos permite programar la comunicación con dispositivos HID del USB de manera muy sencilla.

Otro de los problemas surgidos al realizar los primeros TFC ha sido la visualización de los datos que se van recogiendo. Para paliar este problema se realizó una paleta de componentes Delphi para la monitorización de señales [12] que contiene, como más importantes, la representación visual de un termómetro, un tacómetro y un gráfico de líneas que admite más de una señal. La figura 4 muestra una captura de los componentes tacómetro, termómetro, gráfico de líneas y la paleta completa.

2.2. Laboratorio virtual y remoto de procesos químicos

La segunda parte del proyecto que se tiene en mente es la simulación de procesos químicos. Es decir, se quiere disponer de unas herramientas que permitan que los alumnos realicen prácticas con simuladores y que después apliquen lo simulado a un caso real en el laboratorio. Algo del estilo de lo que aplican en [10] a la robótica, pero aplicado a la ingeniería química. Se quieren construir simuladores de componentes sencillos como objetos que después puedan ser utilizados para construir un simulador de procesos químicos.

[2] http://www.borland.com

[3] http://www.delphi-jedi.org

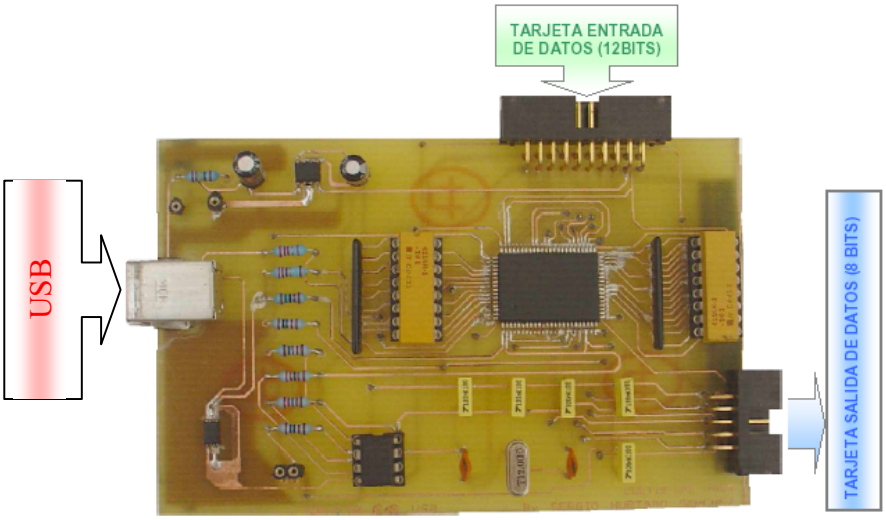


Figura 3. Placa de conexión USB para sensor o actuador

3. Trabajos de fin de carrera

Los proyectos descritos en la sección anterior están siendo utilizados como plataforma para el desarrollo de TFC. Los profesores tienen que dedicar un esfuerzo a la descomposición en pequeños componentes que puedan ser desarrollados en el tiempo que dura un TFC. Además de dedicar especial atención a que el trabajo quede bien documentado de cara a ser utilizado por otros alumnos posteriormente. Está sección está dedicada a detallar algunos de los TFC que se han dirigido.

3.1. Automática para ingeniería química

En este apartado describiremos tres de los TFC dedicados a este ámbito. El primero ha sido el diseño e implementación de la paleta de componentes Delphi para monitorización [12]. Este trabajo contiene los componentes de monitorización previamente detallados, así como una ayuda para los mismos en formato HTML y un tutorial de cómo usarlos y otro de cómo crear nuevos componentes[4]. En la figura 4 se pueden ver varios de estos componentes en uso: termómetro, tacómetro, *displays*…

El segundo TFC importante [11] ha sido el diseño del dispositivo HID para la conexión de sensores y actuadores a un ordenador personal vía USB mostrado en la figura 2 y descrito en la sección anterior. La figura 3 muestra la placa resultado de la parte intermedia del esquema anterior.

El tercer TFC consiste en la automatización de la columna de destilación [13] de la figura 1. Una columna de destilación es un equipo químico que sirve para separar una mezcla líquida aprovechando la distinta volatilidad de sus componentes. Partiendo de los TFC comentados anteriormente se ha llegado a monitorizar la columna de destilación, que involucra dos sensores y dos actuadores. Actualmente, lo están utilizando alumnos de química industrial para realizar experimentos. Este proyecto ha llevado la idea de escalabilidad, modularidad y *plug&play* hasta sus máximas consecuencias; tanto a nivel hardware como software. En la figura 4 podemos ver las ventanas que aparecen tras conectar al sistema un sensor de temperatura, un actuador de control de reflujo y un actuador de manta calefactora. Estan ventanas internas a la aplicación se activan a través del evento *onPlug* del USB (este evento se activa cuando hay una conexión o desconexión en el bus). La aplicación funciona adicionalmente con tecnología web–flash.

[4] Toda la información y software está disponible bajo licencia de libre distribución.

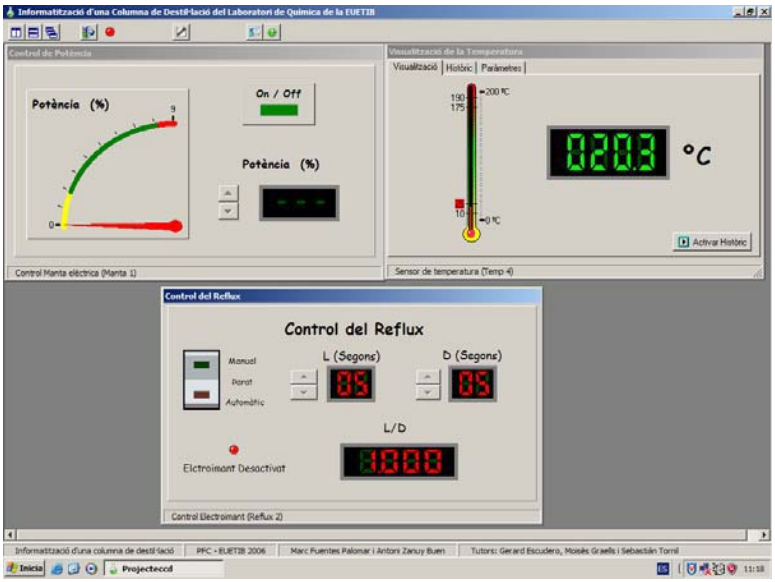


Figura 4. Software para la monitorización de una destiladora

Actualmente, se está llevando a cabo un TFC que sigue en la línea de crear componentes electrónicos que encapsulen diferentes tipos de sensores y actuadores. En concreto, se pretende encapsular un sensor de caudal y mejorar la precisión de un actuador que controla una fuente de calor. A partir de este momento se están proponiendo TFC a realizar por parte de alumnos de la ingeniería electrónica y software para la monitorización y control de equipos químicos por parte de alumnos de la ingeniería química.

3.2. Laboratorios virtuales y remotos

En el terreno de los laboratorios virtuales se han llevado a cabo diferentes proyectos. Aquí destacaremos dos de ellos. El primero de ellos ha consistido en la conexión de un simulador de procesos químicos comercial (HYSYS™) al Delphi vía la tecnología COM [8]. En este caso se diseñó un proceso químico en un simulador comercial y se creó una interfaz de manejo al mismo que funciona remotamente bajo el protocolo TCP/IP.

El segundo TFC [14] ha sido el desarrollo de una herramienta para la simulación y optimización de redes de tratamiento de aguas residuales. Este trabajo se sitúa en otro nivel y en lugar de pretender el detalle de las medidas de cada operación, aborda problemas de toma de decisiones a nivel de proceso, en el que se integran un conjunto de operaciones modeladas como cajas negras.

4. Investigación

Este último TFC se publicó en sendos congresos internacionales de ingeniería química [15,16,17]. A partir de este primer trabajo se ha desarrollado un prototipo más extenso para hallar configuraciones más óptimas de redes de depuradoras para el tratamiento de aguas residuales y se está planteando el uso en redes eléctricas. Hay que añadir que el tema no está cerrado en absoluto y que la alumna en cuestión sigue trabajando con los autores después de haber terminado el TFC[5].

5. Conclusiones

El plantearse un proyecto ambicioso y a largo plazo puede ser una plataforma ideal para

[5] Gracias, por una parte, a que esta alumna prosigue sus estudios de segundo ciclo y por otra, al Programa de Formación de Estudiantes en Departamentos e Institutos de la UPC, promovido por el ICE de la Universidad (http://www.ice.upc.edu/formacio_departaments.htm)

Trabajos de Fin de Carrera. Los alumnos acogen con muy buenos ojos el hecho de participar en un proyecto útil, en algo que quede y que lo utilicen las generaciones posteriores de alumnos de la escuela. Otro punto que ven muy favorable es el hecho de estar tomando y afectando a decisiones de un proyecto de gran envergadura, que incluso puede llegar a publicarse en congresos de investigación.

Agradecimientos

A todos los alumnos que han querido ayudar al aprendizaje de las siguientes generaciones de alumnos de la escuela: David Agustí, Núria Arranz, Xavi Cano, Gerard Español, Roger Fernández, Marc Fuentes, Rafael Gallego, Ferrán Gascón, Manuel González, Sergio Hurtado, Cristina Martín-Sistac, Chakir M'Rabet, Francisco J. Pablos, Enrique Pérez, Enric Queralt, Cristina Rubio, Pau Serrano y Toni Zanuy. A Jordi Jorba por su ayuda y a la dirección de la EUETIB por su respaldo y financiación.

Referencias

[1] Queralt, E. "Disseny d'una tarjeta d'adquisició de dades per a bus PCI". TFC de la EUETIB (UPC). 2004.

[2] Gallego, R. "Tarjeta d'adquisició de dades per a bus PCI". TFC de la EUETIB (UPC). 2006.

[3] M'Rabet, C. "Adquisición remota de datos mediante tarjeta PCI". TFC de la EUETIB (UPC). 2006.

[4] Fernandez, R. "Módulo de adquisición de datos con interfaz USB". TFC de la EUETIB (UPC). 2005.

[5] Español, G. "Velocimetre digital amb conexió USB". TFC de la EUETIB (UPC). 2005.

[6] Agustí, D. "Informatización de un proceso de destilación". TFC de la EUETIB (UPC). 2003.

[7] Arranz, N.; Pablos, F. J. y Rubio, C. "Informatización de un fermentador". TFC de la EUETIB (UPC). 2003.

[8] González, M. y Pérez, E. "Instrumentación y monitorización de columnas de rectificación". TFC de la EUETIB (UPC). 2003.

[9] Graells, M. y Jorba, J.. Un proyecto piloto para la planificación estratégica y acreditación de las asignaturas. A: Actas del XII Congreso Universitario de Innovación Educativa en las Enseñanzas Técnicas. CEIB - EUETIB, 2004, p. 1-6.

[10] Candelas, F. A.; Torres, F.; Gil, P.; Ortiz, F.; Puente, S. y Pomares, J. "Laboratorio virtual remoto para robótica y evaluación de su impacto en la docencia". Revista Iberoamericanca de Automática e Informática Industrial. pp. 49-57. 2004.

[11] Hurtado, S. "Desenvolupament d'un sensor de temperatura USB". TFC de la EUETIB (UPC). 2005.

[12] Cano, X. y Serrano, P. "Paleta de components visuals per a control industrial". TFC de la EUETIB (UPC). 2004.

[13] Fuentes, M. y Zanuy, T. "Informatització d'una columna de destil•lació". TFC de la EUETIB (UPC). 2006.

[14] Martín-Sistac, C. "Disseny de xarxes de tractament d'aigües residuals". TFC de la EUETIB (UPC). 2004.

[15] Martín-Sistac, C. y Graells, M. "A Robust Hybrid Search Technique for Solving Distributed Wastewater Treatment Systems" Computer Aided Chemical Engineering Series: European Symposium on Computer Aided Process Engineering – 15, Elsevier (ISBN 0-444-51987-4). Computer Aided Chemical Engineering Series Vol. 20A, pp. 949-954. Amsterdam, The Nedeerlands, 2005.

[16] Escudero, G.; Martín-Sistac, C. y Graells, M. "A modeling framework for addressing the design and operation of distributed wastewater treatment system"s (Poster T10-027). 10th Mediterranean Congress of Chemical Engineering. Barcelona, Spain, November 2005.

[17] Martín-Sistac, C., Escudero, G. y Graells, M. "General Framework for Solving the Design and Operation of Wastewater Treatment Networks". 17th European Symposium on Computer Aided Process Engineering. 2007.

Hacia una metodología para el desarrollo de trabajos y Proyectos Fin de Carrera en Ingeniería Informática

Antonio Polo Márquez, Jorge Martínez Gil, Luis J. Arévalo Rosado[1]
Dpto. de Ingeniería de Sistemas Informáticos y Telemáticos
Universidad de Extremadura
Avda. de la Universidad s/n, 10071 Cáceres
{polo, jmargil, ljarevalo}@unex.es

Resumen

En este trabajo se proporcionan las claves para el diseño de una metodología docente de trabajos y Proyectos Fin de Carrera en Ingeniería Informática. La metodología resultante se adapta a la complejidad del problema y propugna técnicas de trabajo en equipo. Se enfoca tanto desde la perspectiva del alumno, resaltando las normas que debe conocer y aplicar para obtener un resultado efectivo, como desde el punto de vista del profesor, que debe gestionar un número considerable de proyectos. Se destacan aquellos aspectos más relevantes para la evaluación del trabajo, distinguiéndose entre la evaluación del desarrollo del proyecto y la del producto que se genera. Finalmente se exponen las conclusiones de su aplicación en el desarrollado en algunos PFC y en trabajos de asignaturas de 4º y 5º de Ingeniería Informática en la Escuela Politécnica de Cáceres.

1. Introducción

Una de las tareas principales de aprendizaje en Ingeniería Informática es el desarrollo de trabajos y del Proyecto Fin de Carrera (PFC). En la mayoría de los Centros se dispone de *Guías de Proyecto Fin de Carrera*, pero suelen restringirse a normas de presentación de tipo administrativo. Por ello nos parece imprescindible disponer de una metodología que sirva de guía al alumno y que, a su vez, facilite al profesor el proceso de evaluación de estas actividades.

Dicha metodología debería ser acordada y aceptada por todos los profesores de la titulación que desarrollen soluciones informáticas, con las adaptaciones necesarias para las particularidades de cada asignatura. De esta forma, deberá contener los principios básicos de Ingeniería de Software, pero dispuestos con la mayor flexibilidad para aplicarlos al desarrollo de problemas independientemente de su grado de complejidad. Con ello queremos abarcar un amplio dominio de problemas, desde un PFC hasta un ejercicio a desarrollar en una clase de problemas y en los que se fomenta el hábito de trabajo en equipo. Para lograrlo hemos distinguido tres niveles de complejidad de trabajos:

1. PFC, con una duración aproximada de un curso académico y desarrollado por un grupo de 3 a 4 personas. Sería deseable que se siguiera el ciclo de vida de desarrollo de sistemas de manera completa.
2. Trabajo o práctica para una asignatura de tipo cuatrimestral, a realizar por un equipo de 2 a 3 personas, y donde algunas de las etapas del ciclo de desarrollo pueden ser simplificadas con las indicaciones del profesor.
3. Problema o ejercicio propuesto en clase, y cuyo período estimado para su realización puede ser desde un día hasta varias semanas, siendo resueltos por 1 ó 2 personas.

En la metodología que se plantea distinguiremos dos tipos de usuarios, con necesidades y objetivos diferentes, como son el profesor y el alumno.

Desde el punto de vista de un alumno, el punto clave es saber ubicar el problema y hacer una estimación acertada de los recursos, en términos de tiempo y esfuerzo que necesitará para resolverlo. Por tanto, si el alumno conoce su tiempo de dedicación, deberá identificar, con la ayuda de su profesor/tutor, cada una de las etapas que debe completar y decidir la distribución en

[1] Este trabajo ha sido financiado por el MEC a través del proyecto "Perspectivas de ICARO": TIN2005-09098-C05-05 y TIN2005-25882-E.

términos de esfuerzo para dar solución al problema que se le está planteando. En concreto, el estudio de viabilidad debe adecuarse a estas restricciones de tiempo y esfuerzo más que a recursos económicos.

Desde el punto de vista del profesor, el problema inicial consiste en seleccionar los temas de trabajo y determinar la forma en que va a realizar el seguimiento de todos ellos. Quizás tenga que dirigir a la vez muchos proyectos, para los que se desea una temática y evaluación de carácter individual. Se necesitan ideas para especializar cada asignación y mecanismos para facilitar la gestión de todos los proyectos. Además, los proyectos deben ser prácticos, es decir, además de ilustrar una parte teórica debe obtenerse un producto final. El alumno debe disponer de ejemplos de productos que proporcionará el profesor y que servirán de modelos en el proceso de evaluación.

En este artículo se exponen los aspectos más destacados para definir esta metodología. En la siguiente Sección se reflexiona sobre una serie de consideraciones preliminares, basadas en la idea de aceptar un contrato profesor/alumno que fije las normas de desarrollo y evaluación del trabajo. En la Sección 3 se destacan las pautas metodológicas para definir dicho contrato, destacando qué debe contener. La Sección 4 propone un modelo de seguimiento y evaluación del trabajo y, finalmente, la Sección 5 expone las conclusiones y trabajos futuros.

2. Preliminares

Existen numerosas propuestas de *Aprendizaje Basado en Problemas* y profesores que propugnan dicha técnica de enseñanza [1],[2],[3], en cuyas ideas nos hemos basado para este artículo. En [4] nos planteamos proporcionar una guía para facilitar a los alumnos de la carrera de Ingeniería Informática la realización de su PFC. Las reflexiones que ahora presentamos no son sino una revisión y extensión de aquella propuesta, junto a la experiencia acumulada durante todos estos años en algunos PFC y asignaturas de 4º y 5º de II en la Escuela Politécnica de Cáceres.

Antes de empezar a tratar nuestra propuesta creemos conveniente hacer unas consideraciones preliminares:

- *Globalización*. El trabajo que se proponga siempre debe estar dentro de un dominio más amplio que el del problema a resolver por cada grupo. Los productos que se desarrollen deben contribuir a la realización de un producto global.
- *Modularización*. Debe existir cierta relación de unos problemas con otros. Es preferible que se solapen, es decir, que haya cierto grado de afinidad entre los grupos. Esto permite describir el problema global y establecer un plano de subproblemas de menor grado de complejidad que agrupa de forma equilibrada problemas relacionados en bloques. Aunque se pueda complicar el proceso de evaluación, se introducirán situaciones más adecuadas al mundo real.
- *Especialización*. En cada trabajo se deben definir determinadas áreas de especialización que se asignarán a los componentes del grupo. Esto permitirá una evaluación individualizada sobre las tareas y temas asignados a cada alumno en el trabajo y que deberá reflejarse en su calificación final.
- *Concreción*. En cada tema de trabajo se debería tener un producto final, que deje sentadas las bases para que un siguiente grupo pueda desarrollar una nueva versión operativa de dicho producto. Tareas como aplicar mejoras, ampliaciones, mantenimiento, etc. a este producto, estarán más cercanas al mundo laboral; puesto que la gran mayoría de los sistemas informáticos no empiezan desde cero, sino que se asientan sobre otros ya consolidados.
- *Noción de contrato*. En las primeras reuniones, en especial durante el proceso de especificación del producto, debe quedar claro qué espera el profesor del alumno y que el alumno conozca claramente qué es lo que el profesor le va a evaluar y cómo se va a realizar ese proceso de evaluación. En esta noción de contrato debería simularse la forma de uso del producto generado. Nosotros proponemos en nuestras experiencias que se desarrolle bajo licencia Creative Common 2.5 con libertad de uso y modificación [6], y restricciones de uso no comercial, obligación de referencia de autores y mantenimiento del tipo de licencia de cualquier producto derivado.

- *Ejercicio preliminar fuertemente guiado*. En la línea de proporcionar modelos o ejemplos, sería interesante realizar previamente al trabajo un ejercicio muy guiado, con la especificación concreta de lo que deberá realizar en el trabajo que se le asigne finalmente.
- *Dirección de grupos*. Deben articularse los mecanismos de trabajo en grupo. El profesor decidirá cómo se forman los grupos y cómo se asignan los trabajos buscando un equilibrio entre las afinidades de los componentes del grupo y el nivel de base de cada uno de ellos. Esto puede originar productos finales de diferente calidad que origina problemas de valoración final ante el alumno, en especial si el profesor valora fuertemente la adquisición de habilidades de aplicación de metodologías de desarrollo antes que el producto final que se obtenga.

3. Pautas metodológicas

En esta sección vamos a desarrollar nuestra propuesta de contenidos que deben acordarse en la elaboración del contrato profesor/alumno y que indican las pautas metodológicas para abordar la resolución de problemas en Ingeniería Informática. El objetivo es elaborar el esquema de la guía que se proporciona al alumno indicando las actividades que debe realizar y el conjunto de recursos que deben acompañar a dicha guía. En la próxima sección se indica la forma de seguimiento y evaluación de estas actividades.

3.1. Planteamiento del problema

El planteamiento del problema es la etapa inicial de cualquier proyecto de desarrollo software. En nuestro caso, creemos que se debería considerar como una negociación entre alumnos y profesor, y en el que se debería obtener un documento de trabajo que debería servir a modo de contrato para todas las etapas posteriores del proyecto. El documento de definición del problema ha de ser inamovible, recordemos que el problema no suele estar sujeto a cambios, aunque sí las aproximaciones para resolverlo. No obstante, para compensar que, tal vez la resolución completa del problema inicial pudiera ser no abordable, será obligatorio fijar unos requisitos básicos, medios y avanzados.

3.2. Objetivos. ¿Qué se quiere?

La primera respuesta que se nos puede ocurrir al evaluar esta pregunta puede ser: *resolver un problema*, pues es esa la misión fundamental de toda tarea de Ingeniería. Sin embargo, si consideramos que el ámbito donde se nos plantea tal cuestión es el educativo, entendemos que la respuesta más apropiada sería la de *aprender a resolver un problema.* La diferencia entre ambas respuestas es sutil, mientras que la primera *exige* que el problema sea resuelto, la segunda *recomienda* adquirir *buenos hábitos* que en un futuro permitan resolver problemas de la manera más adecuada. No creemos razonable exigir, aunque sí valorar, como posteriormente veremos, el hecho de que un alumno salga airoso de su primer acercamiento a un problema de complejidad media-alta.

Una vez que tenemos claro que el objetivo es el aprendizaje, y que éste se basa en la adquisición de un conjunto de buenos hábitos o destrezas, cabe preguntarse por estos últimos. Nuestra propuesta los divide en cinco grandes categorías representadas en la Tabla 1.

Destreza	Modo de conseguirlo
Metodología de desarrollo	Trabajo en equipo
Recolección de información	Portafolio con Entrevistas y Búsquedas cualitativas
Desarrollo del producto	Proceso de ingeniería
Elaboración de informes	Documentación
Exposición en público	Presentación

Tabla 1. Destrezas y modo de conseguirlas

3.3. Resultados. ¿Qué se pide?

Creemos que los resultados tangibles de un PFC deben ser fundamentalmente tres:

1. **Documentación**. En la que podemos distinguir tres componentes: Un **resumen**, un **texto** y una **presentación**. El resumen sería deseable que fuera presentado en castellano e inglés, para facilitar su inserción en catálogos públicos. En todos los casos deberá fijarse el formato. Y para la presentación se pueden

probar diferentes estrategias, pudiendo considerarse que la realice sólo uno de los miembros del grupo o que debe estar preparada en tantos bloques como componentes del grupo y se sortea la asignación de los bloques que deba presentar cada uno. Esto responsabiliza a los miembros del grupo y asegura un conocimiento global del trabajo para todos sus miembros.

2. **Producto**. El sistema final que trata de resolver el problema que se planteó inicialmente, y que debe cumplir unos requisitos mínimos en cuanto a calidad, funcionalidad y usabilidad. Los métodos para medir estos factores los detallaremos en una sección posterior.

3. **Evaluación de la actividad**. Creemos que el proceso formativo debe estar sujeto a revisión crítica por las partes que lo componen. En este sentido, creemos conveniente la elaboración de un informe en el que se detalle de manera exhaustiva las conclusiones que se puedan derivar del desarrollo del proyecto, así como la distinción entre las decisiones que supusieron un acierto, en contraste con aquellas que no deberían volver a repetirse en proyectos futuros. Debe realizarse una evaluación tanto del trabajo particular asignado a cada grupo, como la del resto de trabajos presentados y de la labor desarrollada por el profesor en la puesta en marcha de todo el proyecto.

3.4. Medios. ¿Qué se da?

Entre los materiales que se debería entregar al alumno creemos conveniente destacar:

- **Fuentes de información**. Destacando trabajos o PFCs de cursos anteriores que servirán como modelo o como punto de partida para su ampliación. Se incluirán las presentaciones de los trabajos, las calificaciones obtenidas y, si es posible, el producto final en producción (instalado o listo para descargarse desde un servidor).

- **Metodología de trabajo**. El presente documento puede servir como base para que un grupo de alumnos y su tutor lleguen a un acuerdo para la realización de un trabajo. En esta metodología vamos a distinguir:

  Especificación del sistema:
  - *Detección de problemas*
  - *Propuesta de soluciones*
  - *Selección de soluciones*
  - *Selección de nuevas tecnologías*

  Desarrollo del sistema

  Puesta en marcha y evaluación

- **Modelos de trabajo**. Para poder llevar a cabo el proyecto y facilitar el seguimiento y valoración final del mismo se proporcionan "modelos" para realizar las siguientes tareas:
  - Normativas de realización del trabajo, que incluye claramente, qué se quiere, qué se pide, qué se valora y cómo se valora; así como las normativas de preparación de las reuniones de seguimiento.
  - Modelos de documentación
  - Modelos de programación
  - Modelo de evaluación de calidad del producto, de evaluación del proyecto, de los demás proyectos y de la asignatura.

Debe quedar claro que el profesor valorará el proceso de evaluación que realiza el alumno y no el resultado de la evaluación en sí misma, siendo ambos conceptos independientes entre sí. De esa forma, es posible que el producto final sea maravilloso, pero la evaluación del mismo no se haya realizado correctamente y viceversa.

3.5. Evaluación. ¿Qué se valora?

Se valoran estos elementos:

- Reuniones de seguimiento.
- Presentación.
- Producto.
- Informes de evaluación.

Las reuniones de seguimiento se detallan en la siguiente sección, mientras que el producto y los informes se comentan a continuación.

Producto. En cuanto a la evaluación del producto cabe destacar cuatro aspectos fundamentales:

a) *Documentación:* La documentación debe ser clara, estructurada y completa. Debe contener la definición del problema, el análisis y diseño de la solución, una

sección de bibliografía que incluya las referencias que aparecen en el texto y, finalmente, la recopilación del material utilizado en el Portafolio.

b) *Producto de Desarrollo*: Debe incluir un manual de desarrollo, todos los ficheros fuente y todas las herramientas empleadas. Deben estar claras las versiones de los compiladores empleados, y sería deseable ofrecer un fichero de tipo makefile que automatice la compilación. Por último, deberían incluirse baterías de test para el sistema.

c) *Producto de Distribución*: La distribución debería incluir el fichero ejecutable, un manual de instalación y uso, además de ejemplos.

Informes de evaluación. Contendrá cuatro niveles de evaluación: evaluación del proyecto indicando cómo se ha desarrollado, evaluación del producto obtenido, evaluación del resto de trabajos y evaluación de la asignatura.

Evaluación del proyecto

La evaluación del proyecto analiza la planificación inicial con los autoregistros de seguimiento para obtener las desviaciones, y además deberá exponer conclusiones sobre estos datos. Para ello es esencial que en cada grupo, tanto de forma individual como colectiva, se mantenga un diario del proyecto. El diario del proyecto debería contener un diagrama con cada tarea y el coste (tiempo y esfuerzo) empleado para completarla. La evaluación estará basada en una comparación detallada entre el coste que se estimó y el que finalmente fue necesario. Hemos encontrado de gran utilidad el uso de sencillas hojas de cálculo para realizar las asignaciones de recursos previstas y reales de cada proyecto, y posteriormente "unificar" todas estas hojas de cálculo para obtener estudios de comportamiento sobre todos los trabajos realizados. Esta labor suele ser compleja de realizar con las herramientas de control de proyectos especializadas.

Evaluación de la calidad del producto

Se trata de evaluar no sólo qué requisitos se han logrado sino también el grado de calidad alcanzado. Si denominamos *servicio* a la facilidad funcional que nos proporciona la herramienta o programa desarrollado, deberemos evaluar los siguientes aspectos:

- **Comparación de beneficios del sistema nuevo con respeto al antiguo.** Para ello es necesario determinar si se ha mejorado la amigabilidad o usabilidad del sistema con tablas de comparación con sistemas similares, si se ofrece una API (Application Programmer Interface), que no es más que un protocolo para el intercambio de datos, basada en estándares simples y difundidos. Y por último, la confidencialidad y la seguridad de la funcionalidad ofrecida.

- **Prueba de los servicios.** No sólo interesa que los servicios de la herramienta funcionen con un ejemplo, sino que necesitan ser validados. Para ello es necesario un *entorno de pruebas* con un *generador de pruebas parametrizado* que permita generar y evaluar *baterías de pruebas.*

- **Plan de puesta en marcha.** Que comprende un estudio de las fases en las que el sistema ha de estar en desarrollo y en las que ha de estar en producción, es decir, operativo. Deberá basarse en un principio de *oferta incremental de servicios* que presenta una planificación temporal de oferta de servicios.

4. ¿Cómo se realiza el proceso de evaluación?

En el contrato que se establece en el inicio de la asignación del trabajo debe quedar claro qué tareas se van a valorar y la importancia de cada una con respecto al resto. Debemos destacar la importancia de exponer de forma explícita *todo* lo que esperamos del alumno, por trivial que nos parezca. Esto proporciona al alumno seguridad y al profesor garantías de que el alumno va a trabajar en los aspectos que él espera.

La Tabla 2 muestra un ejemplo de valoración del trabajo utilizada en una de nuestras asignaturas:

Valoración de tareas en el trabajo	
Actividad	**Peso en %**
Reuniones de control	30
Presentación	20
Producto	40
Documentación del producto	(25)
Evaluación de calidad del producto	(15)
Evaluación del proyecto	10
+/- Ajuste del trabajo individual	+/- 20
Peso del trabajo en la asignatura	*50*

Tabla 2. Ejemplo de asignación de pesos para la evaluación

4.1. Seguimiento

Para el seguimiento de los trabajos proponemos realizar al menos tres reuniones de control con el grupo. Estas reuniones deben estar planificadas y previamente a cada una de ellas se proporciona el material e instrucciones con especificación de los objetivos y actividades que se van a llevar a cabo. Para ello se proporciona una guía de entregas que detalla explícitamente el contenido de cada reunión. De esta forma, antes de celebrarse una de estas reuniones de control será necesario que cada grupo entregue a su tutor: el informe de estado del proyecto, el portafolio y el test de autoregistro (de acuerdo con la planificación realizada al principio del proyecto). Además, después de cada reunión deben quedar claras cuáles son las siguientes pautas a seguir. Veamos con más detalle cuáles son las reuniones mínimas que se proponen:

Reunión preliminar. En esta reunión, que puede hacerse por bloques de trabajo en los que intervienen varios grupos relacionados, se indica el problema global, una introducción al problema de cada grupo y las relaciones y dependencias que pueden existir entre los grupos. Además, se dan las indicaciones de cómo se realizan búsquedas y qué fuentes se deben consultar, como repositorios de trabajos de cursos anteriores o bibliotecas.

Además se proporcionan recomendaciones para el desarrollo del trabajo en equipo, abordándose cuestiones de asignación de funciones o de aspectos psicológicos en el plano de las relaciones personales y para el buen funcionamiento colaborativo del equipo, aportando indicaciones para la detección de problemas y sugerencias de cómo afrontarlas [5].

1ª Reunión de control. En la primera reunión de control se exige entregar el siguiente material:

a) *Definición del problema*, que incluya la descripción del sistema actual, además de la detección y análisis de problemas. Para ello se solicitan especificaciones abstractas mediante mapas conceptuales [7].
b) *Solución propuesta,* que incluya los requisitos básicos, medios y máximos, así como un borrador de la arquitectura del prototipo y de su API.
c) *Planificación,* con las horas individuales y en grupo de dedicación prevista al proyecto.
d) *Portafolio,* que debe incluir toda la información recopilada, es decir: un índice, trabajos de cursos anteriores, entrevistas con los usuarios, formularios de entrada/salida, sistemas similares, tablas comparativas y material de referencia o bibliografía.

Para elaborar este material se recomienda una serie de actividades secuenciales que permiten su elaboración, que se detallan a continuación.

Planificación preliminar del tiempo disponible para el trabajo a nivel individual y a nivel de grupo. Recalcándose que esto es independiente del tema que deba realizarse, y que podemos planificar las actividades que se conocen en este momento: reuniones de seguimiento, entregas, presentación y entrega del producto final.

La primera tarea que debe planificarse consiste en la *recopilación de información*. Debe especificarse cómo va a realizarse la búsqueda de información (individual o en grupo), dónde va a buscarse, qué temas va a buscar cada miembro del grupo,… Es necesario definir una estrategia de búsqueda de información, con aspectos como qué tipo de material necesito, la terminología del problema, palabras clave y expresiones de búsqueda, datos de la empresa o sistema de información que se está estudiando, entrevistas con usuarios y solicitud de fichas o formularios.

Se debe destacar la necesidad de determinar la lista de fuentes en las que se van a realizar las búsquedas. Sitios web especializados para nuestro

dominio de búsqueda, para ello se dan sugerencias: librerías digitales de organizaciones, editoriales especializadas, Google Scholar, servicios que ofrece la biblioteca de la Universidad, repositorios de trabajos de cursos anteriores,… poniendo de relieve que, en muchas ocasiones, las fuentes de mayor calidad no son abiertas o tienen acceso restringido. Estas sugerencias se completan con técnicas de búsqueda y recuperación de información (almacenamiento de consultas, notificaciones y suscripciones a buscadores,…).

Además, debe acordarse el formato de *Portafolio* en el que van a guardarse los resultados de estas búsquedas, que debe permitir la integración final de las búsquedas realizadas por los integrantes del equipo. Para ello se facilita una definición de la estructura que debe tener el Portafolio que consta de:

- Un fichero de referencias bibliográficas, que consta de un fichero BibTex gestionado mediante JabRef [8].
- Un directorio en el que se almacenan los recursos encontrados (ficheros, programas, enlaces, copias de páginas descargadas de la red,...)
- Un documento, denominado *interpretación del Portafolio*, que explica el contenido que se va guardando en el Portafolio, y los criterios que se han seguido para organizarlos y clasificarlos.

El siguiente paso es una *reunión de tormenta de ideas*, que permita exponer las conclusiones que cada alumno ha obtenido al revisar el material que ha recuperado en sus búsquedas. En este momento se intercambian ideas sobre problemas, crítica del sistema actual y posibles soluciones que permitan detectar las fortalezas y debilidades del sistema, plantear casos de uso y propuestas de posibles soluciones. Este proceso conducirá a una selección de problemas y posibles soluciones que conducen a la Especificación de Requisitos y a que vayamos definiendo determinadas áreas de desarrollo y especificaciones individuales.

Después de esta tormenta de ideas, tras la que ya sabemos los posibles problemas y soluciones que vamos a abordar, es el momento de revisar el borrador de planificación con los requisitos, tecnologías que se van a estudiar o utilizar, actividades de especialización, y de grupo. De esta forma obtenemos un nuevo nivel de planificación ajustado a los requisitos, y ya estamos hablando de actividades y tareas concretas que se van a realizar para las próximas reuniones de seguimiento. Puede que el grupo ya esté en condiciones de buscar un acrónimo y logotipo para el trabajo… y dispuestos para presentar el material de la primera reunión.

2ª Reunión de control. A la segunda reunión de control, además de las correcciones que se pudieran derivar de la anterior reunión, se debería entregar el siguiente material de trabajo:

a) *Informe de estado del proyecto,* que incluya un prototipo para el sistema, un resumen de las tecnologías que se han probado, un test de autoevaluación y el informe de autoregistro. Fundamentalmente se entrega el análisis funcional y diseño de datos.
b) *Versión preliminar de solución,* que de solución al menos a los requisitos básicos propuestos en la anterior sesión de control. Se hace hincapié en que se prohíbe la entrega de cualquier codificación y son sólo prototipos basados en interfaces abstractas de entrada/salida.
c) *Mapa conceptual,* un mapa conceptual que permita a una persona no especialista comprender el sistema que se está desarrollando.
d) *Evaluación individual* en la que se explican los conocimientos de especialización de cada miembro del equipo según su asignación dentro del trabajo.

3ª Reunión de control. A la tercera y última reunión de seguimiento sería muy conveniente llevar (además de las correcciones que pudieran derivarse de las anteriores sesiones de control):

a) *Producto,* con la documentación y el mapa conceptual definitivo, la distribución resultante con los resultados obtenidos en los tests y un formulario que evalúe la calidad de dicho producto.
b) *Borrador de la presentación,* necesario para pulir algunos detalles antes de la defensa ante el tribunal calificador. Aunque la tercera reunión se celebre con poco tiempo antes de la presentación

final, es importante porque suele convertirse en una prueba o ensayo general de la presentación y revisión de la documentación que debe entregarse.

Uno de los problemas que hemos detectado es que se prepara el material para cada una de las reuniones de forma independiente, de modo que al final, cuando debe entregarse toda la documentación, es posible que se omita material que se entregó en alguna de las reuniones anteriores. Por ello, debe recalcarse que cada entrega consiste en la versión *en ese momento de desarrollo* de todo el trabajo asignado, de forma que las entregas constituyen **versiones** preliminares del trabajo final entregado.

Entrega final

Se recomienda que la documentación se presente en forma lineal según el orden de numeración que aparece en la Tabla 3:

1. Documentación general del producto
2. Documentación o Manual del programador
3. Documentación o Manual del usuario
4. Evaluación del Producto
5. Documentación del Proyecto, que suele resumirse en una tabla que representa la planificación prevista y la real
6. Evaluación del Proyecto

<table>
<tr><td>Implementación</td><td colspan="2">Documentación</td><td>Evaluación</td></tr>
<tr><td>(P) Programas de desarrollo</td><td>(2) Del programador</td><td rowspan="2">(1) General del producto</td><td rowspan="2">(4) Del Producto</td></tr>
<tr><td>(U) Programas de distribución</td><td>(3) Del usuario</td></tr>
<tr><td></td><td colspan="2">(5) Del Proyecto</td><td>(6) Del Proyecto</td></tr>
</table>

Tabla 3. Relación de material a entregar

Nótese que el alumno debe presentar tanto una versión del producto orientada al programador para su desarrollo posterior (2)+(P), como la versión de distribución orientada al usuario (3)+(U).

5. Conclusiones y trabajo futuro

Pensamos que de cara a la implantación de los nuevos Planes de Estudio basados en los créditos ECTS del EEES, tiene gran importancia disponer de metodologías y sugerencias para actividades docentes como las que planteamos.

Una debilidad de este tipo de enseñanza es que puede resultar imposible alcanzar los programas de contenidos actuales de modo uniforme para todos los alumnos del curso que, en cambio, pueden presentar de forma individualizada un alto nivel en sus temas de especialización [2]. Por el contrario, se potencia el desarrollo de habilidades como el trabajo en grupo y metodologías para gestión, análisis, desarrollo y evaluación de proyectos.

Esta metodología se puede proponer como punto de partida para la realización de trabajos y prácticas en la titulación. También es aplicable, con ligeras modificaciones a estudios de doctorado en los que se plantea, en lugar de un trabajo práctico, la realización de un artículo de investigación [3].

Referencias

[1] Joaquim Anguas, Luis Díaz, Isabel Gallego, Carmen Lavado, Angélica Reyes, Eva Rodríguez, Kanapathipillai Sanjeevan, Eduard Santamaría, Miguel Valero. *La técnica del Puzzle al servicio del aprendizaje de la programación de ordenadores.* JENUI 2006.

[2] Página del Profesor Miguel Valero García. http://epsc.upc.edu/~miguel%20valero/

[3] Página del profesor Alfredo Prieto Martín http://www.alfredoprietomartin.tk

[4] A. Polo, M. Salas, J.C. Manzano, L.J. Arévalo (2001). *Guía para la realización de un proyecto fin de carrera en informática*. Actas de la JENUI01, pp. 281-286. Palma de Mallorca (España).

[5] B. Oakley, R.M. Felder, R. Brent y I. Elhajj (2004). *Turning Student Groups into Effective Teams*. Journal of Student Centered Learning. Vol. 2, No. 1.

[6] Creative Commons. http://es.creativecommons.org/

[7] CmapTools. http://cmap.ihmc.us/

[8] JabRef reference Manager http://jabref.sourceforge.net/

PFC: Los dos lados del espejo. Proyectista – Director ¿Una visión compartida?

Josep Fernández[1], Antonio B. Martínez[1], Raul Marín[2], Juan Pablo Arboleda[1]

[1]Dpto. de Ingeniería de Sistemas, Automática e Informática Industruial
Universitat Politècnica de Catalunya
c/ Jordi Girona 1-3, 08034 Barcelona
josep.fernandez@upc.edu, antonio.b.martinez@upc.edu

[2]Dpto. de Informática e Ingeniería de Sistemas
Universitat Jaume I
Pedro Cerbuna 12, 50009 Castellón de la Plana
raul.marin@uji.es

Resumen

La realización de un Proyecto Final de Carrera (PFC) se ha ido consolidando como una de las actividades formativas de mayor importancia en una titulación universitaria. En el PFC el estudiante tiene que poner en práctica los conocimientos y habilidades adquiridos durante su formación. Con la realización del PFC se espera que el estudiante aborde una tarea de cierta complejidad en la que sea necesario combinar los conceptos adquiridos en las diferentes asignaturas. La complejidad, extensión y carácter integrador de diferentes conocimientos, es lo que diferencia el PFC de las tradicionales prácticas de asignaturas.

En la realización de un PFC destacan dos actores principales, el/la estudiante en su papel de proyectista, y el profesor/a en su papel de director/a o facilitador/a.

Esta ponencia estudia la visión que estudiantes y profesores tienen de lo que es un PFC en el ámbito de las Ingenierías Informáticas, sus objetivos, su realización y resultados. A partir de las experiencias recogidas y una reflexión conjunta, profesor-estudiante, vamos a destacar aquellos aspectos en los que las dos visiones son más parecidas y más alejadas.

1. Introducción

En las últimas revisiones de los planes de estudios universitarios (y esta tendencia parece que se va a mantener), el "practicum" de las titulaciones ha ido aumentando, y como un componente destacado de la formación aparecen los Proyectos Finales de Carrera (PFC). Por lo general, el PFC es un trabajo que debe realizar individualmente cada alumno para finalizar la carrera. La finalidad del PFC es principalmente que el alumno se enfrente a una tarea similar a las que posiblemente desarrollará en el mundo laboral.

Con la realización del PFC, el estudiante tiene que poner de manifiesto que sabe aplicar los conocimientos adquiridos durante los estudios en los diferentes apartados característicos de los proyectos o estudios de Ingeniería. El desarrollo de esta actividad le va a enfrentar ante problemas mucho más complejos que los que ha tenido que resolver a lo largo de la carrera. Entre otras cosas, deberá hacer un estudio bibliográfico bastante amplio del tema del proyecto, concretar sus objetivos, estudiar las técnicas y herramientas necesarias para su desarrollo y aplicarlas. Y sobre todo, ha de tomar decisiones razonadas para obtener los resultados esperados. Además, tiene que elaborar una memoria en la que se describa con precisión todo lo anterior y hacer una presentación ante un tribunal resumiendo los aspectos más importantes de su proyecto.

La ponencia se estructura en 4 apartados. En la sección 2 se presenta la visión que los autores tienen respecto los PFC en las titulaciones universitarias de informática, y de sus fases. La sección 3 se centra en presentar los datos recogidos en las encuestas realizadas a directores de PFC y proyectistas, para posteriormente, sección 4, realizar un análisis comparativo entre las opiniones expresadas por profesores y alumnos. La sección 5 está dedicada a las conclusiones.

2. Etapas y Contenidos de un PFC

Como paso previo a la realización del estudio del funcionamiento de los PFCs, hemos creído necesario describir y unificar los conceptos básicos relacionados con un PFC. En esta sección, se presenta la opinión de los autores respecto las principales etapas en la realización de un PFC en el ámbito de las Ingenierías Informáticas, así como los puntos claves en cada fase. Esta reflexión resulta necesaria a la hora de identificar los objetivos de este estudio, así como para el diseño de la encuesta que vamos a realizar con el propósito de recoger la opinión de proyectistas y directores. La descripción de las fases de un PFC

se ha realizado desde los puntos de vista de los dos actores principales: proyectista y director.

Fases del PFC para el Proyectista

1. *Definición / Elección del tema*. En esta fase inicial, el estudiante tendría que reflexionar sobre sus motivaciones e informarse con el propósito de hacer una elección acertada teniendo en cuenta sus objetivos. El estudiante debería abordar un tema asequible, que le sirva de experiencia en su futura vida profesional, que sea interesante y motivador. Asimismo el alumno tiene que tener claro desde el inicio que el PFC es algo más amplio que un trabajo de asignatura y abarca diferentes aspectos tecnológicos relativos a varias asignaturas.

El caso más frecuente es que el estudiante limite la búsqueda a la oferta de PFC realizadas por los profesores, ya sea por comodidad, por falta de iniciativa o por facilitar la búsqueda de un profesor para que realice las funciones de director. En estos casos, podemos considerar que el alumno renuncia a ejercer parte de la autonomía que se le brinda a la hora de realizar el PFC.

2. *Documentación y fuentes de información respecto al tema*. Ya sea para la definición de una propuesta propia, o bien para la elección de una propuesta realizada por un profesor, el estudiante antes de iniciar el proyecto (tanto a nivel administrativo como de trabajo) tendría que buscar información respecto al tema que va a abordar. Esta búsqueda bibliografía le ofrecerá una primera visión del estado de arte, de la complejidad del tema y de la viabilidad de proyecto atendiendo a su formación y de los recursos que tiene a su disposición.

3. *Elección / Asignación de Director del proyecto*. En los casos en que el alumno selecciona un PFC entre una lista propuesta, el director ya viene asociado al PFC, por tanto la elección del tema de trabajo y director no puede ser disociada.

En el caso que sea el propio alumno quien defina un proyecto a partir de sus propias motivaciones, tendrá que iniciar un proceso de concertación con un profesor que se sienta, por su parte, motivado por la dirección del proyecto que presenta el estudiante. En este momento los alumnos han de presentar una propuesta clara, de interés y flexible.

Los alumnos tendrían que orientarse hacia un profesor que conozca el tema y disponga de recursos, ya que será la persona que le tiene que ayudar durante todo el proceso de la realización del PFC. Tal vez más que encontrar "un buen director" o "una buena directora" tendría que encontrar el “director/a adecuado/a". El/la profesor/a que dirija el proyecto es quien ayudará al alumno a perfilar el tema y el contenido del proyecto, le dará “el empuje” cuando se encalle y las buenas ideas a la hora de realizar el proyecto, la memoria y su presentación. Entre los criterios que los alumnos deberían considerar en el momento de elegir un director para el PFC, podemos destacar tres: 1) *la experiencia* del profesor en el tema o temas del proyecto, 2) *la accesibilidad*, a la hora de consultar dudas o intercambiar opiniones, y 3) *la confianza mutua*, como base para establecer una relación sincera y un ambiente de trabajo fructífero.

4. *Especificación del Proyecto*. Una vez definido el tema del proyecto es necesario que proyectista y director definan los objetivos y el alcance del proyecto. Para ello tienen que tener en cuenta la dificultad de los conceptos y herramientas a manejar, el periodo de realización, los recursos disponibles y la carga de trabajo (normalmente definida en número de créditos en el plan de estudios) que ha de suponer para el estudiante. Como resultado de esta fase se puede generar una lista de requerimientos del sistema o/y de la aplicación informática a desarrollar. La definición de los objetivos y alcances del proyecto ha de ir acompañada de una planificación temporal de las tareas a realizar. Estos elementos son muy útiles para valorar el progreso del proyecto, y para que el propio estudiante pueda responsabilizarse del trabajo que realiza. Esta fase es a menudo obviada o menos tenida en los PFC, ya que proyectistas y directores no “tienen claro” el alcance del proyecto en su inicio. Este hecho puede generar una indefinición que lleve al proyecto fuera de sus objetivos, inseguridad en el proyectista a la hora de realizar su trabajo, y priva al alumno de uno de los elementos a los que posiblemente tendrá que habituarse en su futura vida profesional; trabajar con unos objetivos concretos, y con una planificación temporal prefijada.

5. *Realización del proyecto*. Esta fase, dado que es la más extensa en dedicación, a veces se confunde

con el proyecto en su totalidad. Nuestra opinión es que un proyecto de ingeniería sin un estudio del estado del arte inicial, o sin una especificación previa, o bien sin una documentación completa, no es un buen proyecto. Dada la gran extensión de esta fase, a menudo se distinguen varias etapas, como son el análisis, la elección de tecnologías y herramientas, la implementación y la evaluación de los componentes desarrollados. Lo más normal es que todas estas etapas se realicen de forma cíclica, e incluso, a menudo conforme el proyecto se concretiza se produzcan modificaciones de su especificación.

6. *Redacción de la memoria del proyecto*. En la memoria del PFC, el alumno tiene que ser capaz de recoger y organizar la información del trabajo realizado. La memoria tiene que satisfacer al menos dos funciones: 1) una descripción del trabajo realizado clara, directa y atractiva, para poder realizar su evaluación, y 2) una documentación técnica completa, que permita conocer los detalles de la implementación para facilitar posteriores modificaciones, ampliaciones o simplemente una evolución del sistema gracias a la aparición de nuevas tecnologías.

Según las indicaciones anteriores, una memoria típica de un PFC en el ámbito de las Ingenierías Informáticas tiene que recoger los siguientes apartados:

- Introducción Objetivos y alcance del Proyecto. Contexto y estado del arte.
- Análisis de Requisitos (Funcionales y No-Funcionales)
- Análisis y Diseño del sistema
- Implementación del sistema
- Evaluación del sistema
- Presupuesto, y análisis de viabilidad
- Conclusiones
- Bibliografía
- Anexos. Manuales de usuario y de administración. Documentación técnica y/o comercial de los componentes y sistemas utilizados en el PFC.

7. *Presentación y/o defensa del PFC*. Esta fase final, es a menudo la más temida por los proyectistas, sin duda por la falta de hábito a hablar en público. Hoy en día un buen profesional tiene que ser un buen comunicador. No es suficiente con desarrollar buenas ideas y productos, también se tiene que ser capaz de transmitir ideas y conocimientos, y saber "vender" su producto. Las habilidades comunicativas (tanto orales como escritas) no suelen tenerse en cuenta en los planes de estudios de las Ingenierías Informáticas, y ello queda de manifiesto en estas fases finales del PFC.

Fases del PFC para el Director

1. *Definición de una propuesta de PFC y su difusión*. Ya sea a petición de la dirección del centro o bien por una motivación personal, los profesores se ven con la obligación y al mismo tiempo con el interés de realizar propuestas de PFC. Normalmente los profesores realizan propuestas pensando en motivaciones propias como la realización de trabajos para mejorar su actividad docente (programas o herramientas para la realización de prácticas o demostraciones) o trabajos relacionados con los proyectos de investigación que realiza. Si bien los alumnos a menudo aprecian poder realizar un trabajo que tenga un uso concreto, no tenemos que olvidar que el PFC es un trabajo académico. Por ello, tiene que estar bien definido en objetivos, en el tiempo, y en absoluto tiene que ser un pequeño proyecto de investigación.

2. *Asignación / Elección del Proyectista*. De la misma forma que el estudiante tiene que documentarse y elegir un tema de PFC en función de sus motivaciones y capacidades, parece lógico que el profesor, antes de asignar un proyecto a un estudiante, valore las aptitudes del estudiante respecto a un proyecto concreto. En este momento pueden chocar dos intereses, el derecho de todo alumno a disponer de un director de PFC para completar su formación, y la obligación del profesor a desestimar la candidatura de un estudiante que quiere realizar un PFC que parece, de forma sensata, fuera de su alcance. Es ante este dilema que los centros docentes y los propios profesores han de tener mecanismos para no perjudicar a ningún alumno (disponer de una oferta de PFC de diferentes características o la designación "de oficio" de directores de PFC a aquellos alumnos que tengan dificultades).

3. *Especificación del Proyecto*. En esta fase se establece lo que tendrá que ser el proyecto, y tendría que realizarse de forma conjunta, con el

propósito de integrar plenamente al estudiante en el proyecto. Asimismo, el director tendría que utilizar esta actividad académica para formar al proyectista en técnicas de dirección de proyectos.

4. *Seguimiento del Proyecto*. A lo largo de esta fase que suele ser la más larga, el director tendría que establecer mecanismos para realizar reuniones de trabajo de forma periódica, con el propósito de validar el trabajo realizado por el proyectista (y evitar que dedique tiempo y esfuerzos a aspectos irrelevantes), y marcar el ritmo de trabajo para alcanzar los objetivos.

5. *Definición de la estructura y seguimiento del contenido de la memoria*. El director tiene que supervisar la concreción de la memoria del proyecto con el propósito que sea estructurada, didáctica y completa.

6. *Recopilación de la información*. Antes de que el proyectista concluya su PFC, y que por regla general abandone la Universidad, el director tendría que recoger la máxima información posible respecto el diseño, tecnologías utilizadas e implementación del proyecto, con el propósito de poder transferir esa información a terceras personas que quieran hacer uso, modificar o ampliar el sistema desarrollado durante el PFC.

7. *Evaluación del Proyecto*. Por regla general el director forma parte o informa a la comisión encargada de evaluar el proyecto. En este momento el director tiene que sacar a relucir de forma sincera las virtudes, defectos y/o deficiencias del PFC.

3. La visión de los actores: Director y Proyectista

A parte de la propia experiencia de los autores, el estudio realizado se basa en las opiniones de proyectistas y directores recogidas mediante un cuestionario. Hemos diseñados dos cuestionarios diferentes (ver tablas 1 y 2) que se estructuran en seis partes.

Hemos optado por realizar una encuesta con preguntas muy concretas pero de respuesta abierta, donde no se da a priori a los encuestados un conjunto de respuestas entre las que escoger. En una primera fase hemos recogido la opinión de 30 estudiantes y 9 profesores vinculados con la Facultad de Informática de Barcelona o la Escuela Superior de Tecnología de Castellón de la Plana.

En los siguientes apartados resumimos los resultados más significativos que hemos obtenido.

Número de preguntas por aspectos analizados	Proyectista	Director
Contexto del PFC	4	5
Definición del PFC	4	4
Realización del PFC	2	3
Relación Proyectista-Director	2	2
Resultados del PFC	4	4
Conclusiones	3	3
TOTAL	19	21

Tabla 1. Aspectos analizados y Estructura del cuestionario.

Proyectistas

Respecto el Contexto del PFC

Las primeras preguntas que nos aparecen de forma natural son ¿Qué es para los estudiantes un PFC?, y ¿Qué creen que tendría que ser?.

La mayoría de estudiantes tienen una doble visión del PFC; por una parte como un trámite académico para completar sus estudios, y por otra parte como una actividad similar a la realización de un proyecto en una empresa, o lo que es lo mismo, como una preparación para su futura vida profesional.

A la pregunta qué tendría que ser un PFC, resulta destacable que los alumnos consideran que tendría que ser lo que es. No proponen cambiar el concepto o objetivos de lo que hoy en día se considera un PFC.

En la mayoría de los casos los estudiantes prefieren realizar un PFC propuesto por un profesor. Este hecho puede tener una doble lectura; o bien la comodidad y una cierta falta de iniciativa, o por lo contrario podemos destacar que los profesores realizan ofertas atractivas para los estudiantes.

Entre los criterios de los estudiantes consideran a la hora de seleccionar un PFC, destacan tres: el interés por el tema del proyecto (33%), el director del proyecto (25%) y la utilidad o uso final que tendrá el proyecto (16%).

A la vista de estos resultados nos inclinamos a pensar que los estudiantes ceden el protagonismo

a los profesores a la hora de definir el contenido y objetivos de los PFC que han de realizar. Seria deseable que en una de las últimas actividades formativas que realizan, los estudiantes tuvieran un papel más activo.

Respecto la Definición del PFC

Una vez seleccionado el tema y el director del PFC, y establecidos unos primeros objetivos globales, se ha de pasar a una definición concreta del PFC. En los casos que el PFC nace a partir de la propuesta de un alumno, el PFC está orientado a alguna motivación personal o de su entorno, y el proyecto está muy orientado y delimitado a una aplicación concreta. En los PFCs a propuesta de los profesores, los alumnos, en lo general, no ven la necesidad de aportar su opinión para completar la definición del proyecto. En lo referente a la planificación de las tareas y a la planificación temporal del proyecto, en el 50% de los casos se carece en su inicio de estas planificaciones. Así mismo sorprende que sólo el 10% de los estudiantes manifiestan disponer de los conocimientos necesarios para abordar el proyecto al iniciarlo. Del resto, la mitad (45%) carecen de los conocimientos y la otra mitad (45%) disponen de una parte de estos conocimientos.

Estos datos muestran que los estudiantes se sienten atraídos por aquellos proyectos que le resultan un reto, y no eligen la opción fácil de realizar un PFC que les aportará poco.

Cuestionario Proyectista	Cuestionario Director
¿Qué es para ti el PFC?	¿Qué es para ti un PFC?
¿Qué tendría que ser para ti un PFC?	¿Qué tendría que ser para ti un PFC?
¿Qué criterios has tenido en cuenta para seleccionar el PFC?	¿Qué criterios tienes en cuenta al definir una propuesta PFC?
¿El tema del PFC es una propuesta personal o una propuesta del profesor?	¿Qué criterios tienes en cuenta a la hora de asignar a un estudiante un PFC de tu oferta?
En el caso de que el tema del PFC sea a propuesta del profesor, ¿Has realizado propuestas y aportación a los objetivos y contenido de la propuesta inicial del proyecto?	¿Aceptas propuestas de PFC realizadas por los alumnos?
¿Al inicio del PFC, disponías de los conocimientos necesarios para su realización?	Si el PFC es una propuesta tuya, ¿Aceptas modificaciones de los estudiantes sobre la definición del PFC?
¿Existía una planificación inicial del trabajo a realizar en el PFC?. El trabajo realizado se ajusta a la planificación inicial.	Al inicio del PFC, ¿los alumnos disponen de los conocimientos necesarios para su realización?
¿Existía una planificación temporal inicial de las tareas a realizar en el PFC?. La duración del PFC se ajusta a la planificación inicial.	¿Existía una planificación inicial del trabajo a realizar en el PFC?. El trabajo realizado se ajusta a la planificación inicial.
¿Qué dificultades has encontrado en la realización del PFC?	¿Existía una planificación temporal inicial de las tareas a realizar?. La duración del PFC se ajusta a la planificación.
¿Has dispuesto de los recursos necesarios para la realización del PFC?	¿Cómo es (cualitativamente y cuantitativamente) tu implicación en la realización del PFC?
¿Qué relación has tenido con tu Directo / Ponente?	¿Qué dificultades aparecen durante la realización del PFC?
¿Qué relación te hubiese gustado tener con tu Director / Ponente?	¿Los estudiantes disponen de los recursos necesarios para la realización del PFC?
¿Los resultados finales del PFC han sido los esperados?	¿Qué relación tienes con los Proyectistas?
¿Estas satisfecho de la formación que has obtenido gracias ala realización del PFC?	¿Qué relación te gustaría tener con los Proyectistas?
¿Te pareció adecuada la calificación obtenida por el PFC?	¿Los resultados finales de los PFC suelen ser los esperados?
¿El aprendizaje obtenido con la realización del PFC ha sido útil en tu posterior actividad profesional?	¿Qué aspectos destacarías de la formación del alumno adquirida durante el PFC y qué le será de utilidad en su vida profesional?
Indica los aspectos más positivos en la realización del PFC.	¿Qué nivel de satisfacción crees que alcanzan los alumnos con el trabajo realizado en el PFC?
Indica los aspectos más negativos en la realización del PFC.	¿Te parece adecuada la metodología utilizada para la evaluación de los PFC?
Otras observaciones:	Indica los aspectos más positivos en la realización del PFC.
	Indica los aspectos más negativos en la realización del PFC.
	Otras observaciones:
	¿Cuántos PFCs (aprox.) has dirigido en los últimos 5 años?

Tabla 2. Preguntas de las dos encuestas realizadas a proyectistas y directores.

Respecto la Realización del PFC

Las dificultades más frecuente que los proyectistas manifiestan tener durante la realización del proyecto son: la falta de una definición clara de los objetivos y del alcance del PFC, el desconocimiento del tema del proyecto y de las herramientas utilizadas para su realización, y, finalmente, la problemática de tener que trabajar con equipo reales (a menudo profesionales) que suelen ser una novedad respecto los equipos didácticos utilizados en las asignaturas. Los estudiantes manifiestan tener a su disposición los recursos necesarios para la realización de su PFC.

Respecto la Relación Proyectista-Director

La mayoría de los proyectistas (86%) manifiestan tener una relación muy buena con su director. El resto (14%) indica que ha tenido una buena relación. Asimismo cabe destacar que al preguntar que relación les hubiese gustado tener con su director manifiestan casi unánimemente que "la que tuvieron". De estos datos se puede deducir que los estudiantes aprecian mucho más la labor docente del profesor durante el PFC que durante la impartición de una asignatura, momento en que por lo general, las opiniones de los estudiantes respecto los profesores no son tan favorables.

Respecto los Resultados del PFC

Al ser preguntados por los resultados obtenidos en su PFC, el 30% manifiesta que son superiores a los esperados y el 70% restantes que fueron los esperados. Al preguntar por el nivel de satisfacción de la formación obtenida con la realización del PFC, los proyectistas responden que están muy satisfechos en un 43% de los casos y satisfechos en un 57%.

Asimismo de forma mayoritaria, indican que la formación obtenida gracias al PFC ha sido útil para su posterior vida profesional aunque sus temáticas no estén relacionadas. Otro aspecto analizado ha sido la percepción de los proyectistas de la calificación obtenida por su trabajo, en el que se manifiestan mayoritaria satisfechos.

Conclusiones. Aspectos positivos y negativos.

En la parte final del cuestionario se pregunta por aquellos aspectos que el estudiante considera más positivos y más negativos de su PFC. Entre los positivos destacamos (citamos textualmente):

- "Crear un sistema útil de la nada"
- "Realización de todo el conjunto del proyecto."
- "Toma de responsabilidad. Independencia."
- "Poder realizar un proyecto muy interesante en el que he aprendido mucho."

Y entre los aspectos más negativos:

- "El problema de trabajar con versiones Beta del software. También el no estar seguro de hasta donde abarcaba el PFC."
- "La redacción de la memoria"
- "Las 2 últimas semanas antes de la entrega".

Directores

Respecto el Contexto del PFC

La definición de consenso que se puede construir a partir de las repuestas de los profesores es que un PFC es una actividad formativa en la que el estudiante pone en práctica e integra los conocimientos adquiridos. Para este colectivo es lo que se tendría que hacer y lo que hacen. Por lo general, las motivaciones de los profesores a la hora de definir un PFC son la obtención de programas y sistemas que permitan mejorar la docencia (65%) e incluso la implementación o mejora de sistemas relacionados con su actividad investigadora (42%). En la definición de los objetivos tienen en cuenta la complejidad del tema, y de las herramientas de desarrollo a utilizar así como la carga en tiempo que representará al alumno. En todo momento se ha de tener claro la factibilidad del proyecto en las condiciones de tiempo y recursos disponible.

Los profesores, al asignar un PFC a un alumno, se centran en su currículum académico, su actitud personal, la motivación que el estudiante muestra por el tema y su disponibilidad horaria. Si bien aproximadamente la mitad de los encuestados aceptan dirigir un PFC propuesto por un alumno, el tema de la propuesta ha de estar necesariamente dentro de su ámbito de trabajo.

Respecto la Definición del PFC

Del apartado anterior se desprende que los profesores realizan de forma unilateral la definición de los objetivos y el alcance de los PFC que proponen. Los profesores, en su papel de director de PFC, indican que los estudiantes pocas veces realizan aportaciones a la definición del proyecto, y estas sólo son tenidas en cuenta cuando existen evidencias que así lo aconsejan.

Para los directores seria deseable que el proyectista contase con los conocimientos necesarios para realizar el proyecto, pero la realidad muestra que siempre es necesario un periodo formativo (a tener en cuenta a la hora de establecer la carga de trabajo) ya que tiene que abordar temáticas complejas o hacer uso de herramientas sofisticadas.

La mayoría de los profesores encuestados (72%) manifiestan que tiene que haber (y así lo hacen) una planificación inicial del proyecto, indicando las fases y tareas a realizar.

Respecto la Realización del PFC

Los directores de PFC manifiestan que siguen el trabajo desarrollado por los proyectistas, siempre dejándoles la iniciativa al mismo tiempo que se muestran disponibles. Sus intervenciones suelen se cualitativas, dando consejos para superar los problemas y orientado el trabajo pero nunca entran en el terreno de la implementación.

Las dificultades más frecuentes que han detectado son los retrasos que se generan por la necesidad de aprender el uso de nuevas tecnologías y herramientas (37%), la falta de conocimientos para abordar fases de experimentación (45%), y a veces, la imposibilidad de materializar un proyecto en el intervalo de tiempo que se impone a nivel administrativo -un cuatrimestre- (24%).

Los directores de PFC manifiestan que una de sus funciones es velar que los proyectistas dispongan de los recursos necesarios, pero también avisan que en aquellos caso en que se tiene que realizar alguna compra de material durante el proyecto, la rigidez de los servicios universitarios, y el hecho que se realice el proyecto en un intervalo tan corto de tiempo son una dificultad añadida.

Respecto la Relación Director-Proyectista

Los profesores resumen su actitud respecto los proyectistas como sincera, cordial y de disponibilidad. Manifiestan que ceden la iniciativa al proyectista y su función es que el estudiante rentabilice su esfuerzo con la consecución de los objetivos del PFC. Asimismo indican que mantienen un nivel de exigencia respecto el trabajo, siguiendo la planificación, pero sin llegar a la presión. Tienen en cuenta que al final es una actividad en la que la responsabilidad de su realización recae sobre el estudiante.

En general los profesores consiguen mantener la relación en los términos deseado (y descritos anteriormente) con los proyectistas. Aun así, resulta a veces difícil romper la barrera estudiante-profesor, lo cual dificulta crear la complicidad deseable en el sentido que el director está para guiar y ayudar al proyectista.

Respecto los Resultados del PFC

En lo concerniente a los resultados generados por los PFC, los directores manifiestas que con cierta frecuencia los proyectistas superan los objetivos marcados (26%). Pocos son los casos que no se alcanzan los objetivos, y en todo caso, lo que no se alcanza es el nivel de fijado inicialmente (18%). Los profesores indican que gracias a la realización del PFC los estudiantes desarrollan habilidades como la iniciativa en el trabajo, tener capacidad y organización para abordar problemas complejos, tener criterio para seleccionar tecnologías y herramientas, saber sintetizar lo importante, e integrar conocimientos multidisciplinares, así como documentar y presentar un proyecto. Como consecuencia de todos estos aprendizajes y el hecho de que los estudiantes ven como su esfuerzo se concretiza en un sistema, los proyectistas acaban manifestando un alto grado de satisfacción en relación con el proyecto que han desarrollado.

Por último, al preguntar sobre el sistema de evaluación, los directores de PFC destacan que para realizar una evaluación consecuente de un proyecto (y no quedarse en los aspectos más superficiales) es necesario un análisis completo, lo que requiere un gran esfuerzo. A menudo la única persona que tiene un conocimiento detallado del proyecto es su director, el resto de personas que intervienen en la evaluación no dejan de apoyarse en apreciaciones subjetivas a la hora de evaluar el proyecto.

Conclusiones respecto el PFC

Los profesores consideran que el PFC es una herramienta docente de gran potencia, y que permite lograr objetivos formativos difíciles de alcanzar dentro de las asignaturas que forman un plan de estudios. Otro aspecto que se destaca de los PFC es la posibilidad de desarrollar

herramientas que enriquecen la Universidad, y permiten mejorar la docencia que se imparte.

Otro comentario que aparece a menudo es la conveniencia de revisar el periodo temporal en el que se considera que se tiene que realizar un PFC (habitualmente un cuatrimestre, aunque en la práctica la realización de un PFC sobrepasa los 6 meses de trabajo por parte del estudiante).

4. Proyectista – Director : ¿Una visión compartida?

En las secciones anteriores hemos expuesto las opiniones que acerca del PFC tienen los autores de esta comunicación, los proyectistas y los directores. En este punto nos disponemos a comparar las opiniones de directores y proyectistas, ya no con un interés didáctico sino con un enfoque crítico que nos permita obtener conclusiones que sean la base de propuestas de mejora.

Tanto profesores como estudiantes coinciden en ver el PFC como una actividad importante en la formación. El concepto y diseño actual del PFC se considera correcto, y ni unos ni otros proponen cambios estructurales. A la hora de concretar una propuesta de PFC, la mayoría de los estudiantes se muestran generalmente pasivos. Los estudiantes no suelen aportar ideas ni en la elección del tema ni en la definición concreta de los objetivos del PFC. Seria deseable que esta situación cambiase, no tenemos que olvidar que los estudiantes en breve tendrán que dirigir y liderar proyectos profesionales.

En lo referente a la planificación de las tareas del proyecto y su temporalización se aprecia una divergencia entre directores y proyectistas. Si bien más del 70% de los directores dicen realizar esta planificación al inicio del proyecto, sólo la mitad de los estudiantes (50%) disponen de ella. Otro aspecto sorprendente es el que los profesores manifiestan asignar los proyectos en función de los conocimientos que en el tema posea el estudiante. En cambio, sólo el 10% de los estudiantes manifiestan tener los conocimientos necesarios al inicio del PFC.

Respecto la realización del PFC, los proyectistas aprecian tres dificultades mayores: la falta de una definición clara de los objetivos y el alcance del proyecto, la falta de conocimientos y el trabajo con equipos profesionales. Los directores sólo destacan los dos últimos aspectos como problemáticos. Se aprecia que durante la realización del proyecto, son los directores los que permanecen prácticamente inactivos. No se ha de confundir el ceder la iniciativa al proyectista con "abandonarlo" a su libre albedrío. La falta de una definición concreta del proyecto combinado con la falta de comunicación entre proyectista y director se ve como una combinación que puede llevar fácilmente al fracaso del proyecto.

A pesar que los aspectos y diferencias destacados con anterioridad puedan verse como una amenaza, los proyectistas y directores manifiestan de una forma rotunda su satisfacción con la realización de los PFC, tanto a nivel de relación personal, como en los resultados obtenidos, como en la percepción de la formación obtenida y su utilidad en una futura vida profesional.

5. Conclusiones

A la vista del éxito de los PFC como actividad formativa, seria interesante valorar la utilización de sus métodos en otras actividades de los estudios de las Ingenierías en Informática. Los métodos, los aprendizajes, así como la relación profesor-alumno resultan más satisfactorios que en las asignaturas tradicionales. La realización de este trabajo, que nace como una inquietud de los autores por su propia labor de directores, nos ha sorprendido y enriquecido, y muestra que no todo lo que parece evidente para el director lo es para el proyectista. Este trabajo va a servir como base de un estudio mucho más amplio y sistemático que abarque a un gran grupo de alumnos y profesores.

Referencias

[1] Guia metodològica per a la redacció del Projecte Final de Carrera. Servei de Biblioteques de la UPC. 2005. http://bibliotecnica.upc.es/e-portals/pfc/inici.htm

[2] Guia Pràctica pels PFC de la Carrera d'Informàtica UPF. http://www.iua.upf.es/~xamat/PFC/index.html.

[3] Guía para la realización de PFC. http://www.gsi.dit.upm.es/guia-pfc/guia.html.

[4] How to Write a Master's Thesis in Computer Science. http://www.cs.fit.edu/~wds/guides/howto

Atención a la diversidad (género, discapacidad, proyección social)

El estilo de aprendizaje como característica distintiva entre alumnos: un enfoque estadístico

Carlos Tavares, Juan-Carlos Cano, Pietro Manzoni
Dpto. de Informática de Sistemas y Computadores
Escuela Técnica Superior de Informática Aplicada, Facultad de Informática
Universidad Politécnica de Valencia (UPV)
{calafate, jucano, pmanzoni}@disca.upv.es

Resumen

En este trabajo se realiza un estudio relativo a los estilos de aprendizaje de dos poblaciones de alumnos procedentes de titulaciones universitarias con características muy distintas: Ingeniería Informática y Licenciatura en Documentación.
A partir de una clasificación del estilo de aprendizaje en función de cuatro dimensiones relativas al estudiante: activo/reflexivo, sensitivo/intuitivo, visual/verbal y secuencial/global, y utilizando una aplicación telemática, se ha realizado un estudio basado en técnicas estadísticas para determinar las características de aprendizaje de estos grupos de alumnos. El estudio permite analizar aquellos casos dónde parte de la población estudiantil tendría dificultades de aprendizaje ante métodos docentes con determinadas características.

Se realiza un estudio comparativo entre ambas poblaciones de alumnos, analizando las diferencias de aprendizaje bajo distintas hipótesis de partida.

1. Motivación

El camino hacia un espacio común de educación superior requiere de un consenso básico que permita llevar a cabo los cambios necesarios para alcanzar los objetivos de normalización previstos. Las distintas universidades españolas se encuentran actualmente en este proceso de cambio que requiere, entre otras cosas, transitar del paradigma tradicional de enseñanza centrado en el profesor y la enseñanza, al polo opuesto dónde se encuentra el alumno y su aprendizaje. Desafortunadamente, este proceso de cambio como tal no es sencillo. No es suficiente que los docentes sean expertos en las materias impartidas, sino que además se requiere que éstos profundicen en sus competencias sociales y humanas, de cara a lograr un mayor acercamiento y una mejor comprensión del alumno.

Es sabido que cada alumno tiene características distintivas que definen los ejes de su proceso de aprendizaje. A pesar de ello, en lo que se refiere a alumnos procedentes de diferentes titulaciones, sigue sin haber un marco de referencia que permita caracterizar de forma estadística estas poblaciones de estudiantes.

En este trabajo se presentan los resultados de un estudio realizado en la Universidad Politécnica de Valencia, cuyo propósito es caracterizar los modelos de aprendizaje de dos grupos de alumnos procedentes de las titulaciones de Ingeniería Informática y Licenciatura en Documentación. Mientras que la primera de ellas tiene un enfoque eminentemente técnico, la segunda está más orientada hacia el área de la gestión documental. El principal objetivo de este trabajo es validar estadísticamente la hipótesis de que, por lo menos, en determinados ámbitos del aprendizaje, las dos poblaciones de estudiantes presentan características distintas.

El estudio se basa en una taxonomía propuesta por Felder y Silverman [3-4], que asocia cuatro dimensiones al estilo de aprendizaje. Estas cuatro dimensiones reflejan las dicotomías: activo/reflexivo, sensitivo/intuitivo, visual/auditivo y, por último, secuencial/global.

Para realizar este estudio se ha partido de una encuesta con 44 cuestiones, inicialmente propuesta por Felder [13], y cuyas dimensiones subyacentes son justamente las que caracterizan el estilo de aprendizaje por él definido. El estudio se ha realizado mediante una plataforma web, la cual permite automatizar todo el proceso de generación y tratamiento de encuestas. Además, la herramienta utilizada ofrece la posibilidad de dar a conocer a los participantes los resultados obtenidos al instante, así como una explicación de los mismos.

El resto del trabajo se organiza como sigue. La sección 2 presenta otros trabajos relacionados. En la sección 3 se describen las cuatro dimensiones del aprendizaje definidos por Felder.

En la sección 4 se presenta la implementación de la plataforma web que da soporte al estudio realizado. Los resultados de este estudio se presentan en la sección 5 y finalmente, la sección 6 hace referencia a las conclusiones de este trabajo.

2. Trabajos relacionados

El campo de investigación docente asociado con las características de aprendizaje ha atraído el interés de numerosos investigadores en todo el mundo, que han propuesto diferentes modelos para el proceso de aprendizaje, así como distintas dimensiones para los estilo de aprendizaje subyacentes a esos modelos.

Autores como Witkin [13] relacionan el estilo de aprendizaje con las características físicas de los individuos, especialmente con el cuadrante dominante del cerebro, definiendo cuatro dimensiones para el aprendizaje.

Otros trabajos, como el de Swassing et al.[12], relacionan el estilo de aprendizaje con el sistema de representación de la información, definiendo tres dimensiones para el aprendizaje.

Gardner y James [9] relacionan el estilo de aprendizaje con el tipo de inteligencia y define nueve dimensiones subyacentes.

Finalmente, autores como Kolb [10] y Felder [4-5] relacionan el aprendizaje con los mecanismos utilizados para procesar la información. Ambos estudios caracterizan el proceso de aprendizaje mediante cuatro dimensiones, con ligeras diferencias.

La búsqueda de estilos de aprendizaje, así como sus aplicaciones prácticas, siguen siendo hoy día tema de numerosos estudios, que abordan el problema con distintos enfoques teóricos y metodológicos [1,3,6-8,11].

3. Las cuatro dimensiones del aprendizaje según Felder

Los trabajos realizados por Felder y Silverman, y en particular las cuatro dimensiones por ellos definidas [4-5], son un marco de referencia en el modelado del aprendizaje. Procedemos ahora a definir cada una de estas cuatro dimensiones.

Con respecto a la primera dimensión (activo/reflexivo), los aprendices activos prefieren realizar actividades prácticas, interactuando con el mundo exterior a partir de información que han adquirido en trabajos grupales. Los aprendices reflexivos prefieren realizar un examen y un procesado mental de la información que ellos mismos han reunido.

La segunda dimensión (sensitivo/intuitivo) permite distinguir entre aquellos aprendices sensitivos que prefieren memorizar datos y afrontar problemas típicos mediante procedimientos estándar (sensitivos), de aquellos intuitivos que prefieren buscar soluciones a problemas novedosos y complejos aplicando principios y teorías. Además, estos últimos demuestran más facilidad en la adquisición de nuevos conceptos.

La tercera dimensión permite diferenciar los aprendices en visuales/auditivos. Los aprendices visuales prefieren información visual ya que su capacidad de retención y comprensión es mejor. Los aprendices auditivos están en el polo opuesto, por lo que necesitan escuchar la información, además de verbalizarla ellos mismos (por ejemplo, explicando los conceptos a otros) para retenerla mejor.

Finalmente, la cuarta dimensión distingue entre aprendices secuenciales y globales. Los aprendices secuenciales prefieren que la información sea presentada gradualmente, y por orden creciente de dificultad, siguiendo normalmente una línea de razonamiento lineal para solucionar problemas. Los aprendices globales prefieren que ciertos conceptos complejos sean presentados por adelantado, de cara a obtener una visión más global de las interrelaciones presentes. Una vez que los conceptos son asimilados como un todo, los aprendices globales logran sintetizarlos con facilidad, siendo así capaces de solucionar problemas más complejos.

4. Caracterización de la plataforma web de soporte a la encuesta

La principal ventaja de disponer de una plataforma web es que ésta es accesible desde cualquier terminal con conexión a Internet. Además, la generación, almacenamiento y el posterior tratamiento de los datos se pueden automatizar, permitiendo ahorrar tiempo y papel.

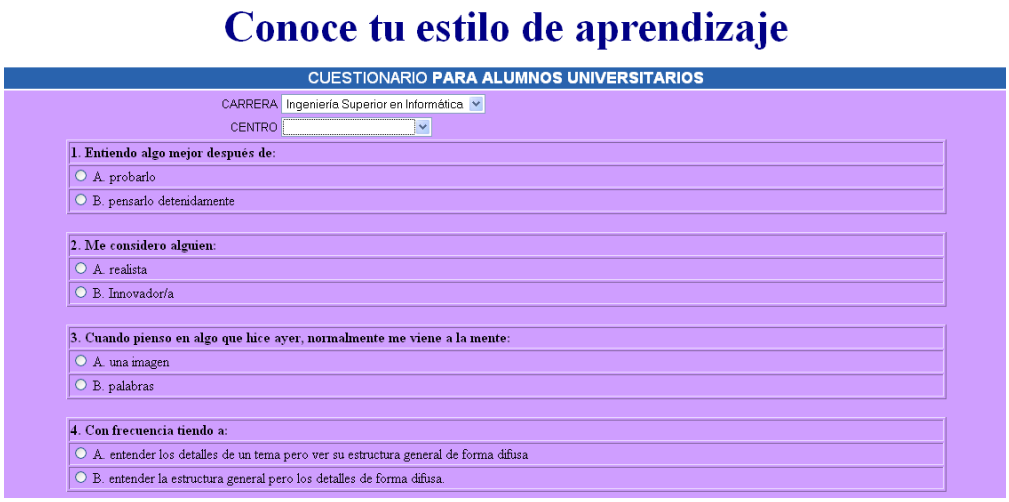


Figura 1. Captura de pantalla del formulario web generado (limitado a las 4 primeras preguntas).

La principal desventaja es que la fiabilidad y el control sobre la misma no pueden ser tan estrictos como en una encuesta tradicional. A pesar de este inconveniente, consideramos que los resultados de este estudio son fiables por dos razones principales. En primer lugar, solamente se ha notificado la disponibilidad de la herramienta a los alumnos objeto de este estudio, estando además el acceso bloqueado desde el exterior de la UPV. En segundo lugar, debido a la naturaleza de la encuesta, y a la participación de los estudiantes consideramos que la muestra obtenida es bastante representativa.

La encuesta originalmente propuesta por Felder, en la cual nos basamos, utiliza un total de de 44 cuestiones con dos posibilidades de respuesta cada una. A cada dimensión analizada se le asocia 11 cuestiones intercaladas.

En cuanto a la puntuación de cada una de las cuestiones, a cada respuesta se le asigna el valor ±1 en función de la opción seleccionada. Al final del cuestionario, y para cada dimensión del aprendizaje, se suman estos valores, por lo que la puntuación obtenida para cada una de las dimensiones se sitúa en el rango de -11 a 11 (sólo valores impares son posibles).

En la figura 1 se presenta una captura de pantalla dónde se puede ver la cabecera y las cuatro primeras preguntas de la encuesta realizada. Cómo se puede observar, en la zona superior el alumno indica la carrera y el centro al que está adscrito para poder diferenciar entre distintos grupos de alumnos.

Una vez el alumno termina de contestar a la encuesta, dispone de un botón para proceder al envío de la información al servidor. El cálculo de los valores finales para cada dimensión de aprendizaje es validada y pre-procesada en el lado del cliente. Esta técnica tiene dos ventajas, permite reducir el ancho de banda consumido en la red, y al mismo tiempo alivia la carga del servidor, el cual solamente tiene que almacenarla. Finalmente, el servidor genera una página de respuesta dónde ofrece al alumno toda la información relativa a su estilo de aprendizaje, así como una explicación breve de los conceptos mencionados y la interpretación de los resultados obtenidos.

La generación y tratamiento de la encuesta se realiza mediante una plataforma web. La plataforma web tiene como elemento central un servidor *MS Internet Information Services* (IIS) que nos permite, mediante código ASP, acceder a una base de datos para generar el cuestionario relativo a la encuesta, así como almacenar los resultados obtenidos.

5. Resultados del estudio realizado

El estudio incide sobre una población de alumnos limitada a dos grandes grupos: alumnos de

Ingeniería Informática y alumnos de la Licenciatura en Documentación.

Los alumnos de Ingeniería Informática provienen de dos centros de enseñanza superior distintos, la Facultad de Informática y la Escuela Técnica Superior de Informática Aplicada, los cuales permiten obtener los títulos de Ingeniero Superior en Informática e Ingeniero Técnico en Informática, respectivamente. En cuanto a la Licenciatura en Documentación, ésta es una carrera de segundo ciclo con carácter no-técnico que también se imparte en la Facultad de Informática. Para realizar el presente estudio, durante el curso académico 2005-06 se solicitó la participación de los alumnos de estas titulaciones. La tabla 1 muestra el numero de alumnos matriculados en cada una de las titulaciones, así como el porcentaje de participación en la encuesta. Además, con respecto a los estudiantes de Informática, no se ha discriminado entre alumnos de las dos titulaciones debido a que se verificó que los resultados obtenidos eran similares.

En cuanto a la interpretación general de los resultados, por cada dimensión el rango de valores posibles oscila entre -11 y 11. Aquellos alumnos cuya puntuación para una determinada dimensión es de -3, -1, 1 o 3 presentan un buen equilibrio para esa dimensión del aprendizaje, lo que significa que tienen la capacidad de aprender sin problema ante cualquiera de los estilos de la dimensión. Los alumnos que obtengan una puntuación de -7, -5, 5 o 7 muestran una clara preferencia por uno de los estilos asociados con esa dimensión, teniendo algunas dificultades para aprender cuando el estilo contrario predomina en el proceso de enseñanza. Finalmente, aquellos alumnos cuya puntuación esté en los extremos de la escala (-11, -9, 9 y 11) son propensos a sufrir de serias dificultades en su proceso de aprendizaje en presencia del estilo contrario, por lo que hay que optimizar al máximo la interacción profesor-alumno para evitar situaciones de desinterés en clase, malos resultados en los trabajos propuestos y en los exámenes.

5.1. Análisis para la dimensión activo/reflexivo

Con respecto a la primera dimensión, la figura 2 presenta el diagrama *"box and whisker"* o de caja y bigotes, con los resultados obtenidos para ambas poblaciones de alumnos. Las siglas INF y LD hacen referencia a los alumnos de Ingeniería Informática y de Licenciatura en Documentación, respectivamente.

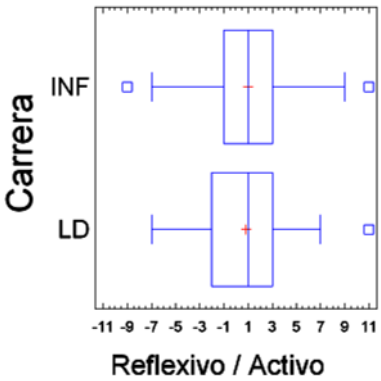


Figura 2. Diagrama de caja y bigotes para la dimensión activo/reflexivo.

El diagrama de caja indica los valores que se sitúan entre el primer y el tercer cuartil, estando la mediana representada por una linea divisoria que parte la caja en dos. La cruz representa el valor medio, y los cuadrados en los extremos indican valores atípicos, fuera del rango de valores normales definidos por el diagrama de caja.

Carrera Universitária	Ingeniería Técnica Informática	Ingeniería Informática	Licenciatura en Documentación
Numerus clausus	400	150+50	75
Número de estudiantes	2156	1320	227
Numero de participantes	119	245	36
Tasa de participación	5,5%	18,6%	15,9%

Tabla 2. Características de la población estudiantil analizada.

Como se puede observar, en ambos casos hay una ligera preferencia por el aprendizaje activo. Los datos presentados en la tabla 2 muestran que la media y la varianza son similares para ambas poblaciones de alumnos, aunque para la Licenciatura en Documentación hay un mayor equilibrio en el estilo de aprendizaje para sus alumnos (no se rechaza la hipótesis nula para α=0.05). En ambos casos se verifica que la distribución no se asemeja a una distribución normal. Además, puesto que el P-valor del test F es superior a 5%, concluimos que no hay diferencia estadísticamente significativa entre ambas distribuciones. El contraste de Chi-cuadrado confirma que no hay una dependencia entre la carrera y el estilo de aprendizaje para esta dimensión.

5.2. Análisis para la dimensión sensitivo/intuitivo

En cuanto a la segunda dimensión del aprendizaje, la figura 3 nos muestra que también para esta dimensión hay importantes similitudes entre ambas titulaciones, aunque la presencia de alumnos con preferencia por el estilo intuitivo sea más común entre los alumnos estudiantes de Ingeniería. Este resultado se puede considerar lógico ya que se espera que un ingeniero sea lo suficientemente versátil como para utilizar los conceptos aprendidos en contextos completamente nuevos.

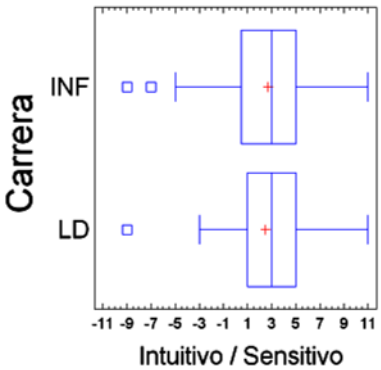


Figura 3. Diagrama de caja y bigotes para la dimensión sensitivo/intuitivo .

Por otra parte, cabe destacar que en ambos casos se verifica que alrededor del 50% de los alumnos muestran un desequilibrio hacia el aprendizaje sensitivo (recolección de información), y que para un 25% esta tendencia es especialmente preocupante, ya que superan los 5 puntos.

Los datos presentados en la tabla 3 muestran que la media es algo superior para los alumnos de Ingeniería Informática, siendo la varianza ligeramente inferior. En ambos casos hay una clara tendencia hacia el lado sensitivo (se rechaza la hipótesis nula para α=0.05), verificándose además que las dos distribuciones no se asemejan a una distribución normal. El P-valor del test F nos indica que también para esta dimensión no hay diferencia estadísticamente significativa entre ambas distribuciones, datos corroborados por el contraste de Chi-cuadrado.

5.3. Análisis para la dimensión visual/auditivo

Con respecto a la dimensión visual/auditivo, una simple apreciación de la figura 4 nos permite constatar que existen diferencias evidentes entre ambas poblaciones.

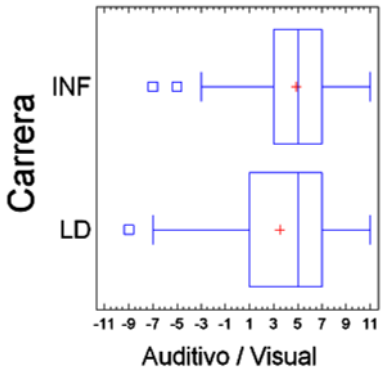


Figura 4. Diagrama de caja y bigotes para la dimensión visual/auditivo.

En ambos casos, queda patente la fuerte tendencia hacia un aprendizaje visual en detrimento del auditivo. Para los alumnos de Ingeniería Informática se verifica, además, que es muy poco común encontrar alumnos con preferencia hacia un aprendizaje auditivo.

Tomando de nuevo los indicadores estadísticos presentados en la tabla 4 como referencia, las diferencias en términos de media son claros. El P-valor del test F nos indica que esta diferencia sí es significativa. También el contraste Chi-cuadrado nos indica que, para esta dimensión, si hay una correlación importante entre la titulación y los resultados obtenidos. En ambos casos se rechaza la hipótesis nula, así como la normalidad de las distribuciones.

Indicador estadístico	Informática	Licenciatura en Documentación
Valor médio (μ)	0,986264	0,833333
Desviación estándar (σ)	4,00032	3,92428
P-valor para hipótesis nula (α=0.05)	3,79638e-6 (rechazada)	0,211017 (no rechazada)
Test de Kolmogorov-Smirnov	0,10577	0,15583
Normalidad de la distribución	Hipótesis rechazada (99%)	Hipótesis rechazada (95%)
Análisis inter-grupo		
Cociente-F / P-valor	0,05 / 0,8266	
Constraste de Chi-cuadrado / P-valor	9,84 / 0,5446	

Tabla 3. Resumen de indicadores estadísticos para la dimensión activo/reflexivo.

Indicador estadístico	Informática	Licenciatura en Documentación
Valor médio (μ)	2,68956	2,5
Desviación estándar (σ)	3,9841	4,4175
P-valor para hipótesis nula (α=0.05)	1,97014E-7 (rechazada)	0,00171873 (rechazada)
Test de Kolmogorov-Smirnov	0,149189	0,239506
Normalidad de la distribución	Hipótesis rechazada (99%)	Hipótesis rechazada (99%)
Análisis inter-grupo		
Cociente-F / P-valor	0,07 / 0,7876	
Constraste de Chi-cuadrado / P-valor	17,17 / 0,1031	

Tabla 4. Resumen de indicadores estadísticos para la dimensión sensitivo/intuitivo.

Indicador estadístico	Informática	Licenciatura en Documentación
Valor médio (μ)	4,88187	3,5
Desviación estándar (σ)	3,89105	5,02281
P-valor para hipótesis nula (α=0.05)	0,0 (rechazada)	0,000184624 (rechazada)
Test de Kolmogorov-Smirnov	0,154707	0,173711
Normalidad de la distribución	Hipótesis rechazada (99%)	Hipótesis rechazada (99%)
Análisis inter-grupo		
Cociente-F / P-valor	3,90 / 0,0489	
Constraste de Chi-cuadrado / P-valor	29,52 / 0,0019	

Tabla 5. Resumen de indicadores estadísticos para la dimensión visual/auditivo.

Indicador estadístico	Informática	Licenciatura en Documentación
Valor médio (μ)	1,17308	-0,222222
Desviación estándar (σ)	4,33516	4,33663
P-valor para hipótesis nula (α=0.05)	5,66946E-7 (rechazada)	0,760317 (no rechazada)
Test de Kolmogorov-Smirnov	0,101091	0,154505
Normalidad de la distribución	Hipótesis rechazada (99%)	Hipótesis rechazada (95%)
Análisis inter-grupo		
Cociente-F / P-valor	3,39 / 0,0662	
Constraste de Chi-cuadrado / P-valor	15,65 / 0,2077	

Tabla 6. Resumen de indicadores estadísticos para la dimensión secuencial/global.

5.4. Análisis para la dimensión secuencial/global

Finalmente, con respecto a la última de las dimensiones del aprendizaje (secuencial/global), la figura 5 nos muestra que hay diferencias claras. En ambos casos se pueden encontrar alumnos con dificultades de aprendizaje para ambos estilos, lo que puede ser un problema a la hora de establecer una estrategia óptima común.

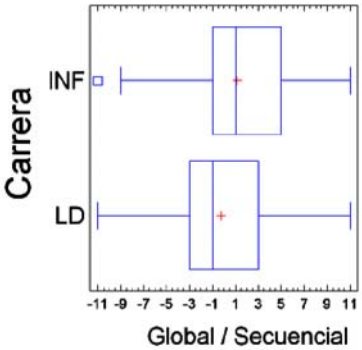


Diagrama de caja y bigotes para la dimensión secuencial/global.

Como nos muestra la tabla 5, los valores medios tienen signos distintos, lo que indica que los alumnos de Documentación aprenden preferentemente de forma global, mientras que los alumnos de Informática aprenden mejor de forma secuencial. El P-valor del test F nos indica que esta diferencia es significativa, aunque el contraste de Chi-cuadrado nos indica que ambas distribuciones tienen características comunes.

Por último, cabe destacar que para la población de alumnos de Documentación no se rechaza la hipótesis nula (distribución equilibrada), y se verifica que el grado de normalidad es algo superior respecto a la población de alumnos de Informática.

Terminado el análisis detallado para las distintas dimensiones del proceso de aprendizaje, en la siguiente sección, se presenta un análisis conjunto de estas dimensiones.

5.5. Análisis de interrelaciones entre las cuatro dimensiones del aprendizaje

Para completar nuestro estudio procedemos ahora a realizar un análisis multivariante cuyo objetivo es determinar la correlación existente entre los distintos estilos de aprendizaje. Para realizar un análisis global, tomamos la población de alumnos como un todo sin distinguir entre titulaciones, ya que es preferible disponer de una población con el mayor tamaño posible para el contraste de la hipótesis.

La tabla 6 muestra la correlación existente entre distintos pares de dimensiones. El rango de estos coeficientes de correlación va de -1 a +1 y miden la fuerza de relación lineal entre las variables. Se presenta además el P-valor correspondiente, que nos indica la importancia estadística de las correlaciones estimadas.

Como se puede comprobar, existe una correlación evidente entre las dimensiones activo/reflexivo y visual/auditivo. Esto quiere decir que los alumnos con preferencia por un aprendizaje activo suelen preferir que la información se presente de forma visual. Por otra parte, para aquellos alumnos dónde predomina el aprendizaje reflexivo, se verifica que asimilan mejor la información que escuchan.

Otra de las dimensiones correlacionadas son secuencial/global y sensitivo/intuitivo. Se verifica que los alumnos sensitivos prefieren adquirir la información de forma secuencial. Finalmente, aquellos alumnos que prefieren solucionar

Dimensiones	*Coeficiente de correlación*	*P-valor para la correlación*
Activo/reflexivo vs. Secuencial/global	-0,0518	0,3014
Activo/reflexivo vs. Sensitivo/intuitivo	-0,0320	0,5236
Activo/reflexivo vs. Visual/auditivo	0,1536	0,0021
Secuencial/global vs. Sensitivo/intuitivo	0,3386	0,0000
Secuencial/global vs.Visual/auditivo	-0,0824	0,0998
Sensitivo/intuitivo vs. Visual/auditivo	0,0433	0,3874

Tabla 6. Correlaciones momento-producto de Pearson

problemas novedosos utilizando principios generales (intuitivos) normalmente prefieren tener una visión global del conocimiento.

6. Conclusiones

En este trabajo se han presentados los resultados de un estudio estadístico relativo a técnicas de aprendizaje según la metodología de Felder. El propósito del estudio ha sido analizar los estilos de aprendizaje de los alumnos de dos titulaciones distintas, Ingeniería Informática y Licenciatura en Documentación. Los resultados obtenidos muestran que existen similitudes entre ambas poblaciones de alumnos para las dimensiones activo/reflexivo y sensitivo/intuitivo, aunque para las dimensiones visual/auditivo y secuencial/global hay diferencias significativas que pueden afectar al proceso enseñanza/aprendizaje. Para las diferentes dimensiones, un determinado número de alumnos presenta una tendencia notable hacia uno de los estilos, lo que puede provocar dificultades en su aprendizaje si no se selecciona la metodología adecuada. En particular, para la dimensión visual/auditivo se verifica que hay una fuerte tendencia hacia un aprendizaje visual, lo que debería estimular a los docentes a basar el proceso de enseñanza en gráficas, esquemas, o diagramas que permitan facilitar la comprensión de los contenidos. Se ha realizado un análisis de las interdependencias observadas entre las distintas dimensiones del aprendizaje, observándose una correlación significativa entre las dimensiones activo/reflexivo y visual/auditivo, así como entre las dimensiones secuencial/global y sensitivo/intuitivo. Con respecto a la aplicabilidad de los resultados aquí presentados y a otros aspectos no tratados, remitimos los lectores a un trabajo complementario a este [2], de los mismos autores.

7. Agradecimientos

Los autores agradecen a la alumna Fatima Zohra Abdou por su participación en el estudio como parte de su proyecto final de carrera en la Facultad de Informática de la UPV.

8. Referencias

[1] Alonso, C; Gallego, D. y Honey, P. *Los estilos de aprendizaje*. Ediciones Mensajero. Bilbao (1997)

[2] Calafate, C., Cano, J. C., Manzoni, P.. *Assesing the learning styles of students from technical and non-technical carrers: a comparative study*, International Technology, Education and Development Conference (INTED 2007), Valencia, Spain. March 7-9, 2007.

[3] Costa, G.; Salgueiro, F. A., Cataldi, Z., García Martínez, R. y Lage, F. J. *Sistemas inteligentes para el modelado del estudiante.* GCETE'2005, Global Congress

[4] Felder, R. M. y Silverman, L. K. *Learning and teaching styles in Engineering Education*, Engineering Education 78(7), pgs. 674–681, 1988.

[5] Felder, R. M., *Reaching the Second Tier: Learning and Teaching Styles in College Science Education*, J. College Science Teaching, 23(5), pgs. 286-290 (1993).

[6] García, A, Catalina M. *Análisis y diagnóstico de los estilos de aprendizaje en estudiantes universitarios*. Facultad de Filosofía y Ciencias de la Educación. Universidad Complutense de Madrid, Madrid, 1992.

[7] Honey, P. & Munford, A.. *Using our learning styles*. 2a ed., Maidenhead, Berkshire,1986.

[8] Honey, P. *Improve your people skills.* Institute of Personel Management, Buckingham, U.K., 1988.

[9] James, W. B., y Gardner, D. L. *Learning Styles: Implications for Distance Learning.* New Directions for Adult and Continuing Education no. 67 (Fall 1995): 19-32.

[10] Kolb, D.A *Experiential learning: experience as the source of learning and development.* Prentice Hall, Englewood Cliffs, N.J., 1984.

[11] Sánchez, M. *La investigación sobre el desarrollo y la enseñanza de las habilidades de pensamiento.* Revista Electrónica de Investigación Educativa 4, (1). http://redie.ens.uabc.mx/ vol4no1/contenido-amestoy.html

[12] Swassing, R., Barbe, W., Milone, M. *Teaching Through Modality Strengths: Concepts and Practices.* Zaner-Bloser, Inc. N.Y. (1979)

[13] Witkin, H. and Goodenough, D., *Cognitive Styles: Essence and Origins: Field Dependence and Field Independence.* New York: International Universities Press, 1981.

[14] http://www.engr.ncsu.edu/learningstyles/ilsweb.html

[15] http://www.engr.ncsu.edu/learningstyles/ilsweb.html

Adaptación de contenidos docentes informáticos para alumnos sordos: una propuesta para abordar esta problemática

Fco. Javier Bueno Guillén[1], Soledad García[2],
José Raúl Fernández del Castillo[1], Jaime Oyarzo Espinosa[1]

[1]Dpto. de Ciencias de la Computación
Universidad de Alcalá
E.T.S.I. Informática
Campus Universitario s/n
28871 Alcalá de Henares (Madrid)
{fjavier.bueno, joseraul.castillo}@uah.es

[2]Centro de Recuperación de Minusválidos Físicos
IMSERSO
Avda. Villamayor 79
37007 Salamanca

Resumen

Existe la idea muy extendida de que las personas sordas pueden leer y comprender correctamente documentos escritos de carácter general. Esto no es cierto en la inmensa mayoría de los casos, representando una de las barreras de comunicación más graves con las que se encuentra el alumno sordo y una clara desventaja al acceder a la Universidad.

En este trabajo se han estudiado los problemas que se encuentra este tipo de alumnos al acceder a un curso básico de Informática impartido mediante *e-learning*. En una base de conocimiento se han recogido las palabras, expresiones y estructuras gramaticales que les plantean dificultades de comprensión. Con el fin de asistir al profesorado universitario en la tarea de adaptación de textos para potenciales alumnos sordos se propone una serie de recomendaciones prácticas.

1. Introducción

En la actualidad el acceso a la formación se ha generalizado, presentando nuestro país elevados porcentajes de universitarios. Sin embargo el porcentaje de alumnos con discapacidad auditiva que cursan estudios superiores no ha crecido al mismo ritmo a pesar de los enormes avances tecnológicos que nuestra sociedad ha experimentado en el campo de la gestión del conocimiento.

Según la Confederación Nacional de Sordos de España [2] mientras que un 20,9% de la población española posee estudios universitarios, tan sólo un 2,8 % de población con discapacidad auditiva acaba dichos estudios. Dentro de ese 2,8 % se encuentran personas con distintos grados de sordera, siendo los sordos prelocutivos los que mayores problemas se encuentran a la hora de acceder a la Universidad y de finalizar los estudios.

La Universidad española no está adecuadamente preparada para formar a este tipo de alumnos. Tradicionalmente se han planteado dos alternativas: contratar intérpretes de Lengua de Signos Española (LSE) o formar al personal docente en dicha materia. Como ejemplo de la primera alternativa, recientemente, la Universidad Complutense de Madrid (88.000 alumnos [8]) se ha comprometido a garantizar la existencia de intérpretes de lengua de signos para el próximo curso en sus aulas. Según FeSorCam [4] el número de intérpretes requeridos para el curso 2006 - 2007 ha sido de nueve para el mismo número de alumnos sordos. En el caso de la Universidad Politécnica el número de alumnos ha sido incluso menor y no cuentan con intérprete. En el caso de la formación de docentes en LSE ésta se suele realizar dentro de talleres o jornadas que se enmarcan dentro de Universidades de Verano, como el taller que se celebró en la Universidad de Verano de Jaca [3] (dependiente de la Universidad de Zaragoza).

Sin embargo, la presencia de intérpretes no siempre es suficiente para que los alumnos sordos sigan las clases en igualdad de condiciones con el resto de sus compañeros. Gran parte de los contenidos que se imparten en la universidad están en formato escrito bien porque constituyan el texto básico seguido en

una materia o porque sea material complementario. Este hecho es independiente de si los textos están impresos o se pueden leer a través del ordenador. De este modo, es el alumno sordo el que se debe enfrentar al texto.

Existe la creencia de que el alumno con discapacidad auditiva tiene un acceso completo a la gran mayoría de los materiales docentes (contenidos) generados para dar soporte al proceso formativo del común de los alumnos. Se asume que el sordo tan solo tiene problemas para oír, lo que dificulta, por ejemplo, el grado de asimilación de las clases impartidas de forma magistral. Por otro lado, dado que el alumno discapacitado auditivo puede ver, no hay nada que le impida leer y por tanto tiene acceso a toda la literatura disponible.

Sin embargo, la realidad es bien distinta. El nivel de comprensión lectora es muy bajo para personas cuya forma de comunicación habitual es la lengua de signos. Según cita Alvira [1] *"... el Equipo Específico de Deficiencia Auditiva del MEC de Madrid realizó en el curso 1997 una investigación con 34 alumnos sordos con edades comprendidas entre 12 y 18 años, de los cuales 18 tenían un nivel de vocabulario correspondiente a un niño de 8 años ..."* lo que nos da idea de las dificultades que se encuentra este tipo de alumnos. Textos sobre materias conocidas pueden resultar ininteligibles para alumnos sordos si no se adaptan de algún modo.

El problema tiene, por tanto, dos caras. Por un lado, se debería trabajar convenientemente la lecto-escritura y la comprensión lectora del alumno sordo en las primeras etapas de su educación. Por otro, se deberían adaptar de algún modo los textos y materiales que se facilitan a este tipo de alumnado para que siga en igualdad de condiciones las respectivas materias en su paso por la universidad. Dado que el objetivo de esta trabajo es el de facilitar la integración de este tipo de discapacitados en la universidad en igualdad de condiciones, nos centraremos en el segundo aspecto

2. Hipótesis de trabajo

Antes de abordar el problema descrito más arriba es necesario plantear algunas cuestiones:

1. Características del alumno "sordo": La denominación de "sordo" agrupa un complejo y no homogéneo conjunto de personas con el común de tener las capacidades auditivas disminuidas. Existen distintos tipos de sorderas y distintas características. Dependiendo del tipo de sordera, la edad a la que devino la sordera y el nivel lingüístico alcanzado en ese momento, una persona sorda ante un texto puede entenderlo perfectamente, entender la idea o tener problemas para entender lo tratado [1]. De esto podemos concluir que una persona sorda tiene su propio ritmo de aprendizaje y que en cualquier caso este ritmo es distinto al de una persona oyente, como serían los compañeros de clase de un alumno sordo.
2. El método formativo para un alumno sordo: Existe un amplio conjunto de estudios sobre cual es el proceso cognitivo de las personas sordas y como éste ha de modificar el método educativo a aplicar [5], [6], [7]. En estos estudios aparece que para aquellas personas sordas cuya forma de comunicación es la Lengua de Signos (LS) procesar imágenes es mucho más fácil y eficiente que procesar palabras. De ello se sigue la necesidad del uso de imágenes para adaptar o facilitar contenidos docentes.
3. El gestor del proceso y gestor de contenidos: En la actualidad se ha generalizado el uso de sistemas de formación o aprendizaje basados en gestores de contenidos *online* o sistemas *eLearning*. En el común de los casos se habla de mejoras en el proceso formativo basadas en la independencia de ubicación y de horarios, bajos costes y sobre todo, la posibilidad de que el proceso formativo se realice al ritmo que el propio alumno demanda.

De lo anterior se deduce que un Gestor de Contenidos de Aprendizaje *(Learning Management System, LMS*), o sistema *eLearning*, es una excelente vía para gestionar el proceso formativo de alumnos sordos. Así mismo aparece como necesaria la adaptación de los contenidos electrónicos tradicionales (texto, multimedia).

Esta necesidad de la adaptación de contenidos se viene resolviendo mediante una traducción completa a LS de los contenidos docentes. Sin embargo, no todos los sordos usan o, incluso, entienden la LS. Por otro lado existen los costes de un traductor y la necesidad de readaptar los contenidos tras cada modificación.

Nuestra opinión es que esta sea, si bien una, la última de las posibilidades a considerar para afrontar el problema de la adaptación de contenidos.

Pasos previos en el proceso de adaptación de contenidos docentes es el de la simplificación de textos, siempre sin pérdida de información. Familiares, educadores e intérpretes tienen conocimiento y experiencia sobre mecanismos de comunicación con las personas sordas distintos de la LS. La comunicación mediante *emails*, carta y SMS pone de manifiesto los tipos de frase, vocabulario y frases coloquiales que dificultan la comprensión y que consecuentemente forman parte de los elementos que en los textos han de ser adaptados.

Desgraciadamente la mayor parte de los educadores y de la comunidad universitaria en general, desconocen los problemas a los que s han de enfrentar los alumnos con discapacidad auditiva. La práctica totalidad de los recursos educativos disponibles para ser distribuidos mediante un LMS solo están pensados para alumnos oyentes, careciendo de los necesarios niveles de adaptación.

Es posible facilitar el proceso de adaptación de contenidos docentes para que este pueda ser realizado por los propios formadores. Como paso inicial parece necesario reunir y agregar el conocimiento proveniente de los participantes en el proceso formativo: educadores, intérpretes, familiares y los propios alumnos, para llegar a una identificación de los elementos problemáticos en los textos en cuanto a vocabulario y a la sintaxis.

3. Método

3.1. Materiales

El estudio sobre las necesidades de adaptación fue realizado sobre los contenidos de un curso básico de Informática (Figura 1), cuya calidad y facilidad de entendimiento fue puesta a prueba con alumnos oyentes como paso previo al estudio realizado en este trabajo. La adecuada selección del tema del curso se consideró necesaria con el fin de disponer de un conjunto de usuarios motivados y participativos con las pruebas. En nuestra experiencia la comunidad sorda demanda este tipo de formación y/o muestra interés por las tecnologías informáticas debido a las enormes posibilidades de comunicación que les brinda.

Figura 1. Contenidos del curso antes de la adaptación. Pantalla inicial.

3.2. Participantes

Uno de los objetivos fundamentales de este trabajo es el de acercar la formación universitaria al colectivo de alumnos sordos. Sin embargo el presente estudio no es realizable sobre alumnos sordos universitarios por la enorme dispersión de la muestra que cabría obtener, al unir a la enorme disparidad de perfiles de los alumnos sordos el escaso grado de participación en los estudios universitarios.

El presente trabajo, centrado en contenidos técnicos, de gran interés para el colectivo, permite definir un cuerpo de estudio homogéneo en el que observar una evolución promedio. El análisis de los datos indica la existencia de problemas de comprensión con expresiones cotidianas en lugar de con expresiones técnicas, al contrario de lo inicialmente esperado. Por ello es posible extender las conclusiones de este trabajo a la comunidad universitaria, objetivo futuro de la investigación.

El conjunto de usuarios de prueba estuvo formado por 12 estudiantes sordos de enseñanzas medias, pertenecientes al Centro de Recuperación de Minusválidos Físicos (CRMF) perteneciente al Instituto de Mayores y Servicios Sociales (IMSERSO).

Diez de ellos sufrían sordera prelocutiva, que en la práctica presenta los peores índices de alfabetización y/o capacidades lectoras. Uno de ellos sufría una sordera prelocutiva severa, y el último presentaba una disminución en las capacidades auditivas.

Con el grupo, atendiendo las incidencias y gestionando la prueba, trabajaron dos intérpretes en LS y dos educadores especialistas.

3.3. Fases del trabajo

El trabajo aquí presentado se realizó en tres fases de estudio claramente diferenciadas:

1. Adquisición de información experimental. Los estudiantes sordos accedieron a dos módulos previamente seleccionados del curso bajo estudio y respondieron cuestionarios sobre su nivel de comprensión y los problemas encontrados. La compilación de una Base de Conocimiento (BC) con expresiones y vocabulario que, de forma general, presenta problemas o dificultades para los lectores con capacidad de lectura de nivel medio, ha sido el resultado de esta fase.
2. Adaptación de dos módulos seleccionados del curso siguiendo las directrices deducidas en el punto anterior obtenidas de las indicaciones de alumnos y educadores.
3. Comprobación de la validez de la adaptación realizada. Los estudiantes volvieron a acceder a los contenidos del curso (dos módulos seleccionados) y completaron un *test* con preguntas cuyas respuestas reflejan el grado de asimilación de los conceptos en ellos consignados. Las cuestiones fundamentales fueron, ¿se ha producido algún incremento en la valoración de los estudiantes sobre los contenidos? ¿aparecen los conceptos presentes en los contenidos con claridad?

4. Resultados

4.1. Base de conocimiento

El curso *eLearning* presentados en la primera parte contienen unas 6000 palabras, de las cuales aproximadamente una tercera parte de ellas eran palabras distintas y con significado propio.

La BC creada contiene unas 900 palabras y/o expresiones, un 50% del total de las palabras con contenido. Se pueden clasificar del siguiente modo (Figura2):

1. Palabras derivadas o pertenecientes a lengua extranjera: Palabras inglesas o derivadas del inglés que en España son frecuentes: PC, BIOS, CD-ROM, etc. y otras palabras hispanizadas, con la dureza propia de la disciplina, como resetear derivada de vocablo inglés "*to reset*". Representan un 7.94 % del total.
2. Palabras técnicas empleadas en el mundo informático. En este grupo aparecen palabras como periféricos, placa base, disco duro, etc. En este caso era de esperar que aparecieran dificultades al tratarse de palabras propias de la disciplina. Constituyen en 29.76 %.
3. Términos de uso común pero fuera de su contexto natural, es decir, que no presentan su significado habitual. Ejemplos de esto lo tenemos en monitor que puede significar instructor. Son el 5.95 %.
4. Términos de uso común. Palabras de uso común no pertenecientes al contexto informático (independientes de la materia). A priori no cabría esperar problemas de comprensión. Sin embargo existe un conjunto de este tipo de palabras que causan problemas y que pasan a formar parte de la BC. Representa el grupo más numeroso (56.35 %).

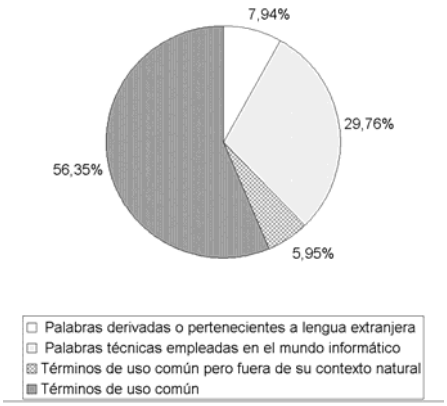


Figura 2. Distribución del vocabulario de la base de conocimiento según los grupos enunciados en el texto.

4.2. Metodología de adaptación

El proceso de compilación de la BC, atendiendo a las observaciones realizadas sobre los usuarios, pone de manifiesto cuatro aspectos fundamentales a ser tenidos en cuenta durante el proceso de adaptación de contenidos:

1. No dé por asumidos conocimientos previos al tema en cuestión. Los contenidos docentes adaptados han de ser autocontenidos en la medida de lo posible.
2. El texto ha de ser escrito con frases simples. Se ha detectado como especialmente problemático el uso de oraciones subordinadas y pasivas. El uso de lenguaje figurativo haciendo uso de metáforas ha presentado dificultades insalvables.
3. En la medida de lo posible utilice palabras sencillas. En el caso de que se perdiese contenido por el hecho de simplificar el vocabulario, facilite sinónimos, imágenes del concepto o explicaciones en lengua de signos (LS) (Figuras 3 y 4).

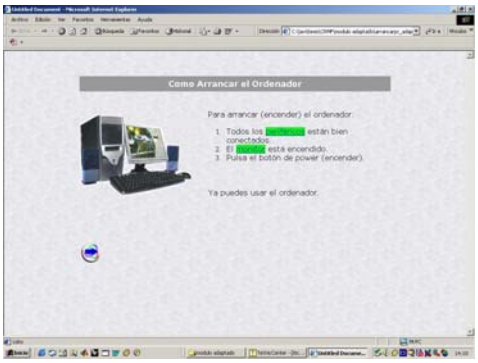


Figura 3. Módulo adaptado: simplificación de frases y vocabulario sin perder contenido

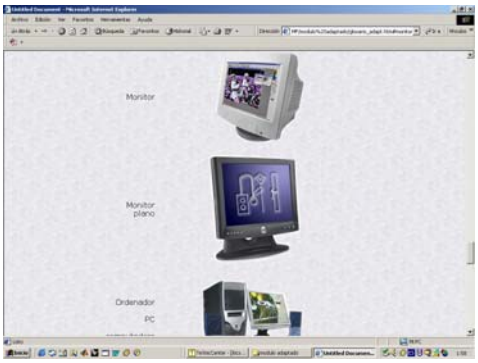


Figura 4. Módulo adaptado: inserción de imágenes para explicar conceptos.

4. Evite el uso de jergas (jergas profesionales), abreviaturas e iniciales salvo necesidad, en cuyo caso han de ser explicadas en LS o mediante ayuda gráfica como en el punto anterior (Figura 5).

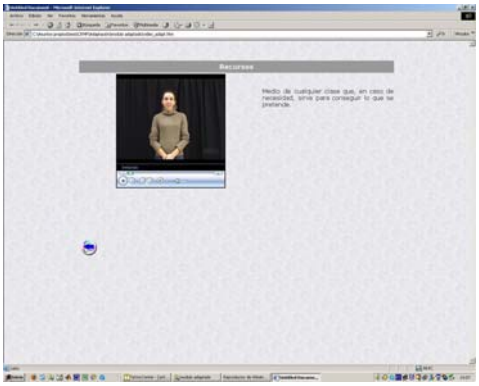


Figura 5. Módulo adaptado: inserción de vídeos en Lengua de Signos Española.

5. Conclusiones

En este trabajo se propone una serie de recomendaciones para facilitar la adaptación de contenidos docentes informáticos basados en texto (escritos, *Web* o mediante sistemas de *eLearning*).

Estas recomendaciones han sido obtenidas mediante el estudio de las necesidades de usuarios sordos accediendo a contenidos docentes informáticos distribuidos mediante un sistema de *eLearning*.

Se ha recopilado una Base de Conocimiento (BC) que recoge las palabras y expresiones que han mostrado problemas de comprensión y que simula el conocimiento basado en la experiencia diaria de educadores, intérpretes y familiares.

El conjunto de recomendaciones y el contenido de la BC han sido probados con éxito en la adaptación de los contenidos bajo estudio.

Como hecho a ser destacado en este trabajo es que tanto la BC como las recomendaciones para la adaptación de contenidos han sido realizadas atendiendo a los principales participantes en el proceso educativo. Estos elementos permitirán en el futuro el desarrollo de sistemas semiautomáticos que asistan al educador en la creación de contenidos docentes.

Agradecimientos

Queremos agradecer al Ministerio de Educación y Ciencia por su financiación mediante el proyecto EA02007-0026, y por la ayuda prestada a todas las personas que han participado en el desarrollo de este trabajo. Muy especialmente queremos agradecer la ayuda al conjunto de estudiantes pertenecientes al Centro de Recuperación de Minusválidos Físicos de Salamanca perteneciente al (IMSERSO).

Referencias

[1] F. Alvira, A. Cruz y F. Blanco. *Los Problemas, Necesidades y Demandas de la Población con Discapacidad Auditiva en España: una Aproximación Cualitativa.* Colección Estudios e Informes. Serie Estudios nº 1. Observatorio de la Discapacidad. IMSERSO, 1999. (pág. 123)

[2] Confederación Estatal de Personas Sordas de España (CNSE). *Libro Blanco de la Lengua de Signos Española en el Sistema Educativo*, CNSE, Madrid, 2004. (pág. 22).

[3] Confederación Estatal de Personas Sordas de España (CNSE) URL: *http://www.cnse.es/noticia_der.php?ID=334*, 2005. (Septiembre, 2006).

[4] Federación de Personas Sordas de la Comunidad de Madrid (FeSoRCaM). URL: *http://www.fesorcam.org/educacion.html*, 2006. (Septiembre, 2006).

[5] Federación de Personas Sordas de la Comunidad Valenciana (FESORD*), Rasgos sociológicos y culturales de las Personas Sordas*. FESORD. Valencia. 2000.

[6] A. Marchesi. *El desarrollo cognitivo y lingüístico de los niños sordos*. Alianza Editorial. Madrid. 1987.

[7] J.M. Rodríguez Santos. *La deficiencia auditiva: un enfoque cognitivo*. Publicaciones Universidad Pontificia de Salamanca. Salamanca. 1990.

[8] Universidad Complutense de Madrid. URL: *http://www.ucm.es/info/ucmp/pags.php?tp=Datos\\\%20estadísticos\&a=presenta\&d=0000691.php*, 2006 (Diciembre, 2006).

Pósteres
Sesión Abierta

7 minutos para un objetivo transversal: ¡L E A A T E N T A M E N T E!

Rosalía Peña y Miguel Garre
Dpto. de Ciencias de la Computación
Universidad de Alcalá de Henares
Edif.. Politécnico Alcalá de Henares (Madrid)
rpr@uah.es y miguel.garre@uah.es

Resumen

Se presenta una acción docente que llevamos realizando 3 años en las Ingenierías Técnicas de Informática de la Universidad de Alcalá, con el objetivo de recalcar la necesidad de leer atentamente, hasta comprender las especificaciones de un problema, antes de iniciar su resolución. La acción fomenta las habilidades analíticas del alumno. La actividad docente consume solamente unos pocos minutos de la primera sesión docente, en la que se presentan al alumno: contenido, objetivos y metodología de la materia.

1. Motivación

Uno de los mayores problemas con los que tropieza la docencia de materias que implican resolución de problemas, en la actualidad, es la impulsividad de los alumnos, que les aboca "a hacer", aun antes de "pensar" e incluso antes de terminar de "escuchar" las especificaciones del problema a resolver. Esta dificultad se aprecia especialmente cuando las especificaciones se transmiten de forma escrita en vez de oral.

Aun siendo un problema que cualquier profesor habrá experimentado, y a pesar de la brevedad requerida para este documento, no queremos dejar de hacer una sucinta referencia bibliográfica que sustente la anterior afirmación. Martín [3] en su análisis de las causas del fracaso académico, aunque aborda el tema de una forma general, encuentra que los profesores estiman en un 71% que "los métodos de trabajo seguidos por el alumno son inadecuados" y en un 70,5% que "el alumno planifica mal su trabajo". Cernuda [2] en un documento, dirigido a los alumnos, en clave de humor, les insta una y otra vez a leer y analizar profundamente antes de lanzarse a "hacer". McFarland [4] insiste en la necesidad de enseñar "acerca del aprendizaje" además de enseñar contenidos. Brookshire [1] recomienda proporcionar un entrenamiento suplementario (extra-coaching) en el proceso de traducción desde el problema enunciado en palabras al modelo matemático (en nuestro caso algorítmico) del problema, en todas aquellas materias que implican resolución de problemas.

2. Descripción de la acción docente

- **Objetivo concreto**: poner de relieve la importancia de leer, comprender y analizar profundamente el problema planteado antes abordar su resolución.
- **Encaje en el objetivo general:** El objetivo general del curso es la formación en la resolución de problemas. La experiencia propuesta es la acción inicial, que se reforzará a lo largo del curso, en la resolución de cada ejercicio.
- **Soporte:** Cuestionario a cumplimentar por el alumno, de extensión no mayor de un folio, resaltando claramente y en primer lugar la solicitud de no cumplimentar hasta leerlo íntegramente. La última pregunta, indica no cumplimentar las anteriores. En [5] se ofrece un ejemplo del cuestionario utilizado.
- **Consumo tiempo lectivo:** 7 minutos
- **Coste de preparación profesor:** escasamente 1 hora.
- **Ubicación temporal y mecánica de la acción**: En la presentación de la asignatura de Fundamentos de la Programación, tras indicar como su objetivo fundamental la resolución de problemas, y por tanto la necesidad de entender profundamente el problema antes de abordar su resolución, se pasa el cuestionario, indicándoles que hay

poco tiempo para su cumplimentación. Se da el tiempo justo de que los alumnos lean el documento íntegro o contesten las 8 primeras preguntas. Se recoge el cuestionario. Se leen las instrucciones del mismo y la pregunta última. En la siguiente sesión docente se indica el porcentaje de cuestionarios bien cumplimentados y se resalta la inutilidad del trabajo realizado por un altísimo porcentaje de alumnos.

3. Evaluación de resultados

El cuestionario se ha pasado en los tres últimos cursos a un grupo de alumnos de la asignatura de Fundamentos de la Programación de las titulaciones de Ingeniería Técnica en Informática de Sistemas y de Gestión. La asistencia a esa primera clase es de unos 60-80 alumnos por grupo, cifra sorprendentemente baja cuando la matrícula es del orden de 125 alumnos. Este dato es, en si, revelador de la importancia que otorgan a la comprensión de los objetivos y metas de la tarea que se proponen realizar (en este caso, cursar la asignatura). La tabla 1 recoge los porcentajes de los alumnos que han contestado más de las preguntas solicitadas, exactamente las solicitadas y menos de las solicitadas. Este último grupo, junto con el de los que conocen la encuesta, no resulta evaluable en nuestra consideración.

	%			
Curso	Conocen	Mas	Exacta	Menos
04-05	0,8	95	3	1,2
05-06	3	90	5	2
06-07	3	91	4	2

Tabla 1. Porcentajes de cuestionarios presentados

La tabla 1, muestra el elevado grado de establecimiento del problema presentado (la impulsividad en “hacer” y la dificultad de atenerse a normas escritas). También muestra la viabilidad de reutilizar la experiencia en años sucesivos, a pesar de que estén presentes alumnos que la conocen.

No se ha pretendido evaluar si existe una correlación entre la habilidad inicial del alumno y sus resultados académicos, de una parte, debido a

a) Uno grupos en los que clasifica el cuestionario es tan pequeño que no resultaría estadísticamente significativo.
b) es un tema trabajado continuamente a lo largo del curso, por lo que nunca podría considerarse una variable independiente.

Nos hubiera gustado comentar alguna rentabilidad adicional sacada al paso del cuestionario, pero la extensión del documento no lo permite.

4. Conclusión

Se ha presentado una acción docente encaminada a motivar las habilidades analíticas del alumno, fomentar su capacidad de reflexión y de atención a las necesidades específicas de su futuro cliente, en su ejercicio profesional.

Estimamos el coste de la actividad es de 7 minutos de tiempo lectivo y menos de una hora entre tiempo de preparación y evaluación por parte del profesor. Los recursos económicos son 3 céntimos por alumno. Los resultados de la acción son independientes del hecho de que se haya realizado con anterioridad.

Dado el pequeño coste y la reproducibilidad de su aplicación, creemos que es una acción muy rentable.

Referencias

[1] R. Brookshire y S. Palocsay. Factors contributing the success of undergraduate Businnes Students in Management Science Courses. Decision Sciences Journal of Innovative Education 3(1) 99-108 2005.

[2] A. Cernuda del Río. Cómo **NO** hacer una práctica de programación, Actas de las VIII Jornadas de Enseñanza Universitaria de Informática 2002, pag. 351 – 358 2002.

[3] *M.A. Martín y alumnos. Análisis de las causas de fracaso escolar en las Ingenierías Técnicas*. Ref EA2004-0155. Universidad de Valladolid 2004.

[4] R. McFarland *Teaching students to learn in the computer science and information systems curriculum: creating a distinction between content and methods.* J. of computing in small colleges. V 19(1) 235-245 2003.

[5] http://www.cc.uah.es/mgarre/Docencia/cuestionarioSaberLeer.pdf

Una herramienta para la evaluación automatizada de la disciplina de Bases de Datos

Piedad Garrido Picazo[1], Gabriel Fuertes Muñoz[1], Jesús Tramullas Saz[2]
[1]Dpto. de Informática e Ingeniería de Sistemas y [2]Dpto. Ciencias de la Documentación
Universidad de Zaragoza
{piedad, gfuertes, tramullas}@unizar.es

Resumen

La presente contribución muestra la experiencia llevada a cabo por un grupo de profesores de distintas áreas de conocimiento de la Universidad de Zaragoza, en el desarrollo y uso de una herramienta didáctica para llevar a cabo, la evaluación automatizada de la disciplina de Bases de Datos.

1. Introducción

Bases de Datos es una disciplina, a la que se dedican entre 4,5 y 16 créditos, en las distintas Escuelas de Ingeniería Técnica de Informática de Gestión en el panorama nacional [1]. Lo que supone un porcentaje bastante elevado, en algunos casos, del número de créditos cursados en esta titulación con respecto al número total de créditos de la misma. Es más, uno de los aspectos más importantes de esta disciplina si se consultan las guías académicas de las 102 entidades universitarias a nivel nacional que imparten esta titulación [5], es el aprendizaje del lenguaje SQL (Structured Query Language) [2].

Realizado un estado del arte del software utilizado en la enseñanza de este aspecto práctico de las Bases de Datos, por un lado nos encontramos con que hay software enfocado [3, 10] al diseño y compilación de los distintos lenguajes de manipulación de datos como pueden ser el Álgebra Relacional, el Cálculo Relacional de Tuplas, de Dominios y el propio SQL. Mientras que por otro lado, existe un amplio uso por parte de las entidades universitarias de los Analizadores de Consultas que incluyen los Sistemas Gestores de Bases de Datos Relacionales (SGBDR) existentes en el mercado, ya sean de software libre o de software propietario (MySQl, OpenOffice.org Base, SQL-Server, Oracle, etc).

Ambas opciones presentan las siguientes desventajas: alta complejidad, aprender un lenguaje formal adicional, trabajar con versiones simplificadas de SQL, de muchas de ellas no se dispone de una versión en castellano y la más importante y común a todas ellas, es que el alumno invierte una cantidad de tiempo en aprender su entorno (administrador corporativo, arquitectura software, etc.) que podría estar usando en la resolución de consultas y restricciones.

Por lo que la importancia del análisis, diseño e implementación de una herramienta software que permita llevar a cabo la evaluación automatizada del diseño, desarrollo y ejecución de un conjunto de consultas resueltas por el alumno en SQL es muy interesante e importante, tanto desde un punto de vista teórico como práctico, puesto que se han conseguido cubrir los objetivos de obtención de una herramienta robusta que permita evaluar al alumno en el laboratorio y no sobre consultas realizadas en papel, conseguir que la herramienta sea indepedendiente del SGBDR utilizado y demostrar la siguiente hipótesis de trabajo:

"Que el nº de aprobados sea mayor que en años anteriores al conseguir que el alumno no cometa errores sintácticos".

2. Motivación, metodología y resultados

La idea de desarrollar una herramienta de este tipo surgió por dos razones: la primera de ellas debida a las frecuentes peticiones por parte del alumnado de la realización del examen de Bases de Datos I, asignatura de primer. cuatrimestre de tercero de la titulación de ITIG en el laboratorio y la segunda, por la detección por parte del profesorado de un elevado número de errores sintácticos cometidos por el alumnado en el desarrollo de sus consultas en papel. Por ejemplo, la ausencia de paréntesis a la hora de colocar en la SELECT una función de valor de agregado como puede ser el COUNT.

Mal	Bien
SELECT COUNT *	SELECT COUNT (*)

Tabla 1. Ejemplo error sintáctico

Para ello, se realizó un análisis estadístico de las distintas pruebas de evaluación llevadas a cabo en los últimos tres cursos académicos. Datos de los que interesaba obtener: el porcentaje del número de suspensos debidos a errores sintácticos, y el porcentaje, en general, del número de errores sintácticos cometidos, independientemente de que el alumno hubiera aprobado o suspendido la materia.

Los resultados obtenidos del análisis estadístico muestran por un lado que un 30% de los alumnos que cursaban la asignatura de Bases de Datos I suspendían debido a errores sintácticos y por otro lado, que un 95 por ciento de los alumnos cometían errores sintácticos durante su proceso de evaluación.

Con la puesta en marcha de SQLOnly durante este curso académico y la consecuente realización de las pruebas de evaluación en el laboratorio,se ha observado que se han conseguido eliminar por completo la entrega de consultas que contengan errores sintácticos, se ha incrementado el número de alumnos presentados en primera convocatoria, puesto que el alumno se siente más motivado y seguro a la hora de presentarse al examen y por último y no por ello menos importante, que el número de aprobados se haya visto incrementado en un 20%.

3. Conclusiones

Al ser los resultados obtenidos tan satisfactorios, concluiremos diciendo que SQLOnly está siendo utilizada y se utilizará, como herramienta software para proceder a la evaluación, en el laboratorio, de los alumnos de la asignatura de Bases de Datos I, y por otro lado, comentar que dichos resultados nos han animado a ponerla en funcionamiento durante este segundo cuatrimestre, en otras titulaciones de la Universidad de Zaragoza, como herramienta de apoyo para facilitar el proceso de aprendizaje de los alumnos de asignaturas básicas relacionadas con las Bases de Datos, como es el caso de los alumnos que cursan asignaturas sobre Documentación Digital en la Diplomatura de Biblioteconomía y Documentación.

Para terminar destacar que esta herramienta va a ser una pieza fundamental en la implantación del Espacio Europeo de Educación Superior (EEES), ya que su objetivo principal es que el estudiante sea el principal actor, dando más importancia al manejo del lenguaje SQL que a la mera acumulación de conocimientos del manejo de distintas herramientas de aprendizaje que lo utilizan, así como resaltar su gran utilidad dentro de dos cursos académicos cuando la asignatura de Bases de Datos I se empiece a impartir de forma paralela en el grupo de alumnos de carácter semipresencial, debido a que se trata de una herramienta sencilla, multiplataforma, portable, segura, distribuida bajo licencia GNU/GPL y lo más importante, independiente de los SGBDR existentes en el mercado, por lo que el alumno aprende el SQL "ANSI", o estándar SQL, y no SQL característico de un sistema de manejo de bases de datos específico [4].

Agradecimientos

Nos gustaría agradecer al alumnado su colaboración en forma de sugerencias y críticas a la herramienta. Así como al Servicio de Informática de la Escuela Universitaria Politécnica de Teruel por su amplia disponibilidad y disposición ante estos desarrollos de software "ad-hoc".

Referencias

[1] Garrido, P. *Proyecto Docente de la asignatura de Bases de Datos 1.* Teruel, 2007.

[2] Gulutzan, P. *SQL-99 Complete: an example-based reference manual of the new standard. R & D*, 1999.

[3] Gutiérrez, J.J, Escalona, M.J., Villadiego, D., Mejías, M. *Comparativa de Herramientas para la Enseñanza de Lenguajes Relacionales*. Madrid, España. Asociación de Enseñantes Universitarios de Informática (Aeuni). 2005. Pag. 297-304. ISBN: 84-9732-421-8.

[4] Koopmann, J. Just SQL: Part I. Database Journal, 2005. Disponible en: http://www.databasejournal.com/features/oracle/article.php/3517901

[5] *Libro Blanco de Título de Grado en Ingeniería Informática.* ANECA, 2005.

[6] WinSQL: http://www.synametrics.com/SynametricsWebApp/WinSQL.jsp

Enseñanza de la Informática en el marco de la Educación Musical

M. Isabel Dorta González[1], Pablo Dorta González[2]
[1]Dpto. de Estadística, I.O. y Computación
Universidad de La Laguna
C/ Astrofísico Francisco Sánchez s/n, 35271 La Laguna
isadorta@ull.es
[2]Dpto. de Métodos Cuantitativos en Economía y Gestión
Universidad de Las Palmas de Gran Canaria
Campus Universitario de Tafira s/n, 35017 Las Palmas de Gran Canaria
pdorta@dmc.ulpgc.es

Resumen

En este artículo se presenta un nuevo enfoque para la enseñanza de las nuevas tecnologías aplicadas a la música. Nuestro objetivo es promover el aprendizaje activo de conceptos y el manejo de herramientas que favorezcan, en lo posible, la adquisición de experiencia. Se propone para ello unos sencillos pero efectivos ejercicios que están siendo utilizados en la asignatura Informática Básica de la Titulación de Maestro Especialista en Educación Musical impartida en la Universidad de La Laguna.

1. Introducción

La introducción de las nuevas tecnologías en el campo de la educación durante los últimos años está permitiendo ensanchar el propio ámbito de actuación educativa. El proceso de la educación musical no puede quedar al margen de toda esta evolución. Desde la aparición de periféricos como la tarjeta de sonido con las funciones de reproducción o grabación, la aparición del estándar de conexión de cualquier instrumento con el ordenador o interfaz midi, la aparición de las enormes posibilidades que ofrece la conexión a Internet, parece claro que estos avances obligan a plantearnos la utilización de todas estas posibilidades en el proceso de la educación musical.

En este trabajo se plantea la integración de la tecnología musical en los contenidos específicos de la asignatura optativa Informática Básica en la titulación de Maestro especialista en Educación Musical de la Universidad de La Laguna. Esta propuesta educativa presenta un enfoque novedoso para promover el aprendizaje activo de conceptos, y el manejo de herramientas que favorezcan, en lo posible, la adquisición de experiencia en el uso de nuevas tecnologías aplicadas a la música.

2. Metodología docente y grado de implicación de los alumnos

El método de enseñanza utilizado tiene como reto hacer que las horas que el alumno pasa en el centro sean lo suficientemente provechosas para que el aprendizaje se produzca en el propio centro. Para lograr este objetivo, la metodología didáctica consiste en dar una mayor concreción en las exposiciones y aprovechar el trabajo del alumno como sujeto activo de su propio proceso de enseñanza-aprendizaje [1].

El desarrollo de la asignatura es fundamentalmente práctico. Parte del horario lectivo es dedicado a prácticas en el laboratorio de Informática. Otra parte del tiempo es dedicado a clases colectivas en las que se abordan los aspectos teórico-prácticos de los fundamentos de la asignatura.

Cada alumno recibe 3 horas lectivas por semana, de las cuales una hora es dedicada a clases teóricas en las que se desarrollan contenidos sobre procesamiento de la información a nivel de usuario: sistemas operativos, tratadores de texto, hojas de cálculo, creación de presentaciones, bases de datos, navegadores y exploradores de Internet, software y hardware musical [5]. Las dos horas restantes corresponden a las clases de laboratorio destinadas básicamente al manejo de las aplicaciones y herramientas software explicadas en las clases teóricas.

Cada unidad temática se estructura de la siguiente manera:

- Introducción
- Objetivos
- Exposición de contenidos teóricos
- Actividades guiadas para la comprensión de los contenidos
- Actividades propuestas de autoaprendizaje
- Autoevaluación de los conceptos tratados

La unidad temática destinada a los contenidos sobre software y hardware musical tiene una duración aproximada de 9 horas lectivas. Se proponen las siguientes actividades utilizando distintas estrategias metodológicas [3]:

- *Actividad 1*: Encuesta del nivel de conocimientos de tecnología informática musical de los alumnos y expectativas que tienen de la unidad didáctica.
- *Actividad II*: Manejo y análisis de aplicaciones musicales en CD-ROM. Esta actividad consiste en que el alumno analice de forma individual el material propuesto por el profesor. Posteriormente se realiza una puesta en común sobre las ventajas e inconvenientes del uso de dichas aplicaciones.
- *Actividad III*: Búsqueda y evaluación de portales educativos musicales y de webs docentes. Tras la recopilación de información por el alumnado, se enumera en la pizarra la información localizada y se van eliminando las que el grupo no considera adecuadas.
- *Actividad IV*: Manejo básico de un editor musical [4]. La estrategia consiste en mostrar los componentes básicos del entorno para comenzar a trabajar (crear y guardar documentos, entrada de eventos musicales a tiempo diferido, reproducción, etc). A continuación, el alumno desarrolla una investigación individualizada del manejo de la aplicación informática.
- *Actividad V:* El sistema MIDI: hardware y software [2]. Tras una sesión expositiva por parte del profesor de los conceptos básicos de un sistema MIDI, se le pide a los alumnos que coloquen los conceptos de manera que construyan una red de conceptos. Es una estrategia indicada para el análisis y diferenciación de conceptos.

Dado el carácter fundamentalmente práctico de esta unidad temática, así como del resto, se hace necesario la implicación del alumno en su propio proceso de aprendizaje. Dicho grado de implicación lo medimos considerando parámetros como: asistencia a las clases, realización de las actividades propuestas, resultados de los tests de nivel y realización de trabajos opcionales. La evaluación del alumnado se lleva a cabo de forma continua. El alumno debe asistir y participar en al menos un 75% de las actividades propuestas a lo largo del cuatrimestre. Además, al finalizar cada unidad didáctica se realiza un test de nivel que tiene como objetivo valorar los conocimientos adquiridos durante dicha unidad. Finalmente, se le propone al alumnado un trabajo en grupo opcional sobre alguno de los contenidos reflejados en el temario. El desarrollo del trabajo es tutorizado por el profesor para indicar las correcciones y mejoras a llevar a cabo. Una vez finalizado el trabajo, cada grupo de alumnos debe hacer una exposición en clase de éste y entregar la documentación correspondiente. Sin la realización de dicho trabajo no se puede obtener la máxima calificación en la asignatura.

3. Conclusiones

Este artículo presenta una experiencia docente que se está llevando a cabo para adaptar las nuevas tecnologías en la enseñanza de la Informática en la Facultad de Educación para el Título de Maestro especialista en Educación Musical. Se ha propuesto un conjunto de actividades en las que se aplican distintas estrategias metodológicas destinadas a favorecer el aprendizaje por parte del alumnado.

Referencias

[1] Bain, K. *Lo que hacen los mejores profesores universitarios*. Valencia: Universitat de Valencia. 2006.

[2] Eiche, J. F. ¿Qué es MIDI?. Barcelona Ed. Music Distribution. 2000.

[3] Hannan, A., Silver, H. *Innovar en la Universidad*. Madrid: Narcea. 2003.

[4] Vergara, V. y Ruíz, J. *Crear, editar y compartir música digital.* Anaya. 2007

[5] Thomson, K.J. *El PC y la música*. Madrid. Anaya. 1996.

Experiencia en Proyectos Fin de Carrera de cooperación con países en vías de desarrollo

Javier Alonso[1], David López[1] y Javier Larrosa[2]
1 Dpto. de Arquitectura de Computadores
2 Dpto. de Lenguajes y Sistemas Informáticos
Universidad de Politécnica de Cataluña.
UPC-Campus Nord, C/ Jordi Girona 1-3, 08034 Barcelona
{alonso, david}@ac.upc.edu, larrosa@lsi.upc.edu

Resumen

Los Proyectos Fin de Carrera son una buena ocasión para implicar a los alumnos en proyectos de cooperación con países en vías de desarrollo. En este póster se presentan dos experiencias de proyectos ya finalizados y en explotación en los campamentos de refugiados de la República Árabe Saharaui Democrática. Se explica cómo se contactó con los usuarios finales, cómo se financió y los resultados obtenidos.

1. Motivación

Cuando se define un Proyecto Fin de Carrera (PFC) lo primero que debe tenerse en cuenta es que reúna las condiciones técnicas adecuadas. El objetivo es garantizar que los estudiantes estén formados en todas las competencias técnicas que un Ingeniero en Informática debe poseer. Pero mientras que los formadores tenemos claro que hay que incidir en la formación técnica del futuro titulado, su formación humana muchas veces es dejada de lado durante la formación universitaria.

Para que un ingeniero en informática pueda ser útil a la sociedad debe ser capaz de desarrollar una serie de competencias transversales [4], referentes a valores éticos, compromiso con la sociedad, sensibilidad hacia el medio ambiente, etcétera. Si solo se forman sus cualidades técnicas estamos creando ingenieros incompletos ya que no han desarrollado su vertiente humana.

Además, la mayoría de los PFC son "artificiales" y una vez finalizados se olvidan para siempre. Con la misma motivación que en otros trabajos en este campo [5] pensamos que los PFC deberían ser utilizables, ya que esto hará que tanto el estudiante como el director obtengan una mayor satisfacción por el trabajo realizado. En esta línea nosotros proponemos el desarrollo de PFC en el ámbito de la cooperación con países en vías de desarrollo ya que cumplen ambas misiones. Por un lado son PFC que refuerzan la vertiente humanista del ingeniero y por otro son socialmente útiles.

En este póster presentaremos los diferentes elementos que conforman un PFC en el área de la cooperación, así como nuestra experiencia en dos proyectos de este tipo.

2. ¿Qué es un proyecto de cooperación?

Un Proyecto Fin de Carrera en el ámbito de la cooperación en países en vías de desarrollo debe cumplir dos requisitos: primero ha de tener una entidad técnica suficiente que lo justifique como PFC en el contexto universitario; y además ser útil en el lugar donde vaya a ser utilizado.

Partiendo de esta idea básica, podemos definir que un PFC de este tipo está formado por cuatro elementos:

- Ha de suponer un reto técnico adecuado para un Proyecto Fin de Carrera.
- Ha de aportar un valor añadido en cuanto al enriquecimiento personal que obtienen el estudiante y el profesor.
- Ha de ser útil, es decir, el proyecto ha de ser instalado *in situ* y el estudiante ha de formar adecuadamente al personal que lo utilizará.
- Ha de solucionar un problema real.

Decidir si un proyecto tiene o no la dificultad técnica suficiente no suele ser difícil, puesto que los profesores tenemos suficiente experiencia. Lo que sí suele ser difícil es encontrar el proyecto. Esto se puede hacer de dos formas diferentes. De forma independiente, es decir, viajando al lugar donde queremos colaborar y contactando directamente con entidades locales y ONG que trabajen en la zona. Este método está solo recomendado a personas con cierta experiencia en proyectos de cooperación. La otra opción es contactar con ONG de nuestro entorno y

ofrecerles "nuestros servicios" como complemento a sus proyectos de cooperación.

Si queremos que el PFC se utilice finalmente, es necesario que el estudiante y/o el director viaje a la zona de cooperación, instale el "producto" y, forme al usuario final para su correcto uso y mantenimiento. Esto requiere financiación que se puede obtener a través de fundaciones como por ejemplo *la Obra social de la Fundación la Caixa*. Algunas universidades cuentan con oficinas y recursos dedicados este tipo de proyectos. En nuestra universidad existe el *Centro de Cooperación para el Desarrollo* [2], además de ONG formadas por miembros de la comunidad universitaria como "*Tecnología para todos*" [7] o "*Ingenieros sin Fronteras*" [3]. Por último, para ser capaces de determinar si el proyecto soluciona un problema real, lo mejor es que sean las ONG con años de experiencia sobre el terreno los que nos proporcionen proyectos, ya que ellos saben las necesidades reales y las prioridades de los usuarios finales.

3. Dos experiencias

3.1. Informatización de la farmacia en Tindouf

Se desarrolló una aplicación para dar soporte a la gestión del almacén de medicamentos de la farmacia central de la República Árabe Saharaui Democrática (RASD)[6]. Este proyecto nació tras un viaje a los campamentos de refugiados Saharauis en Tindouf (Argelia) durante el cual se realizó un análisis de situación para la detección de deficiencias tecnológicas. Durante el viaje detectamos un posible problema en la gestión de la farmacia y propusimos una solución que fue aceptada por los gestores de la farmacia de la RASD. Desgraciadamente, y por falta de conocimiento de la realidad, resolvimos un problema que no era una prioridad, por lo que fue aceptado y agradecido, pero nos quedamos con la sensación de no haber resultado útiles.

3.2. Proyecto de vacunación

En este caso se contactó con un pediatra mallorquín, coordinador del programa de vacunación infantil del Sahara, y fue él quien nos expuso sus necesidades para gestionar las cartillas de vacunación de una manera más eficiente. Al ser una persona con gran experiencia y con una visión muy clara del problema, las necesidades fueron identificadas de manera rápida y precisa, desarrollándose un software a medida, adecuado a las necesidades del personal sanitario que lo iba a utilizar. En este caso, la sensación de utilidad por nuestra parte (especialmente para la proyectista) fue muy gratificante, pues notamos que habíamos ayudado realmente a solventar un problema real.

4. Conclusiones

Los PFC de cooperación son altamente motivadores y formadores para los estudiantes y los profesores que los dirigen, pero deben resolver un problema real y ser puestos en explotación *in situ*. Para encontrar PFC de este tipo, lo mejor es dirigirse a ONG con experiencia en el terreno.

Agradecimientos

Primero de todo a nuestras proyectistas Sara Timón y Caterina Canyelles por su esfuerzo y dedicación. A Pep Bibiloni, pediatra de la asociación "Amics del poble sahrauí de les Illes Balears"[1]. A todos los cooperantes que nos aportaron ideas. Al admirable pueblo saharaui por su hospitalidad y su forma de llevar el exilio. Este trabajo ha sido apoyado por Ministerio de Ciencia y Tecnología de España (Proyecto TIN 2004-07739-C02-01)

Referencias

[1] Amics del poble sahrauí de les Illes balears. http://saharabalears.org/
[2] Centre de Cooperació per al Desenvolupament. http://www.upc.edu/ccd/
[3] Enginyers sense fronteras. http://catalunya.isf.es/NOVA/inicio.htm
[4] J. Casanovas, et al *El libro blanco de la Ingeniería en informática*: el proyecto EICE, JENUI 2004, p. 13.
[5] P. L. Sánchez, et al. *¿Son las universidades socialmente responsables? Una iniciativa a través de los Proyectos Fin de Carrera*, JENUI 2006, p.209.
[6] República Árabe Saharaui Democrática. http://web.jet.es/rasd
[7] Tecnologia per a tothom. http://txt.upc.es/

Iniciativas para la adaptación al EEES de la asignatura de Fundamentos Físicos de la Informática

Esmeralda Úbeda, Josep-Abel González
Dpto. de Física
Universidad de Girona
Campus de Montilivi, 17071 Girona
@udg.es, @udg.es

Resumen

Se presentan las iniciativas elaboradas para la adaptación al Espacio Europeo de Educación Superior (EEES) de la asignatura Fundamentos Físicos de la Informática, cuatrimestral de primer curso de las titulaciones de Ingeniería Técnica en Informática de Sistemas (ETIS), y de Gestión (ETIG). Los objetivos e iniciativas se han centrado en mejorar la programación de las actividades a partir del tiempo de dedicación a la asignatura declarado por los estudiantes, aumentar la motivación de los estudiantes hacia la asignatura, aumentar la asistencia y participación en clase y evaluar al estudiante de manera continuada.

1. Introducción

Desde el curso 2004-05 y en el marco de un Plan Piloto [1] de adaptación de las titulaciones de Ingeniería Técnica en Informática de Gestión y de Sistemas al EEES, se han ido introduciendo una serie de cambios en la metodología docente con el fin de mejorar diversos aspectos en la disciplina de la Física.

A continuación, se presentan los objetivos e iniciativas así como las actividades llevadas a cabo para continuar la adaptación de la asignatura al formato de los créditos ECTS. Se extienden los resultados y conclusiones de las iniciativas del curso 2005-2006 al curso 2006-2007.

2. Metodología docente y resultados

La Universidad de Gerona ha desarrollado e implantando una Intranet académica denominada "La meva UdG" [2]. Es a la vez un vehículo de comunicación dentro de la comunidad universitaria y una herramienta con la cual se ha diseñado y planificado la asignatura desde el punto de vista de competencias, actividades y evaluación. Los objetivos que se han perseguido y las iniciativas tomadas han sido:

- Mejorar la programación de las actividades a partir del tiempo de dedicación a la asignatura declarado por los estudiantes. Hemos comprobado a través de encuestas semanales realizadas durante dos cursos consecutivos que el tiempo de dedicación personal manifestado por los estudiantes ha sido 2 horas menor que el previsto por los profesores. Concretamente, la media semanal de trabajo no presencial previsto por los profesores en los cursos 2005/06 y 2006/07 ha sido de 5.2 h y 5.3 h respectivamente, mientras que los estudiantes han manifestado 3.2 h y 3.3 h respectivamente.
- Aumentar la motivación de los estudiantes hacia la asignatura. Se ha reformado parcialmente el contenido de algunos temas para hacerlos más interesantes al alumno de informática, incluyendo el uso de herramientas informáticas para la resolución de problemas y prácticas de laboratorio, y se ha mejorado el procedimiento de entrega y control de trabajos prácticos. Como se pretende que el alumno utilice la bibliografía recomendada, no hay apuntes del curso disponibles, sino únicamente las presentaciones en PowerPoint utilizadas en las clases teóricas, que son suficientemente esquemáticas para que el alumno no las interprete como apuntes.
- Evaluar al estudiante de manera continuada aumentando la exigencia para superar la asignatura pero sin subir el nivel. Para ello, se ha exigido la superación de todos y cada uno de los bloques temáticos de la asignatura, impidiendo que se pueda aprobar la asignatura con un conocimiento parcial de ella. Esta mayor exigencia se ha compensado con un mayor número de oportunidades de demostrar cada competencia. La participación de los alumnos en la pruebas de evaluación ha ido evolucionando,

estabilizándose en torno al 50% de los alumnos matriculados. El otro 50 % abandonan la asignatura en las primeras semanas del curso. Las anulaciones de matrícula han estado en torno al 55% en ambas titulaciones. La tasa de éxito final ha estado en torno al 85%, como se observa en la Figura 1. Se concluye que aprueban aquellos alumnos que desde el inicio de curso han decidido seguir sistemáticamente la asignatura.

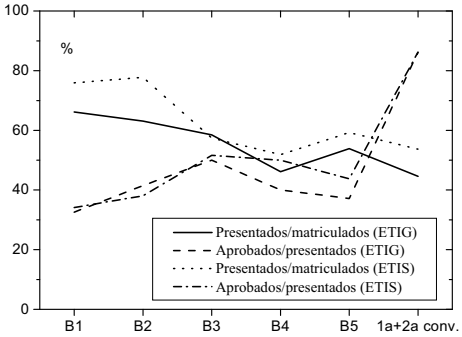


Figura 1. Evolución de la tasa de participación y de éxito en ETIG y ETIS

La Figura 2, por otra parte, muestra la evolución del rendimiento en los últimos cursos. Éste se mantiene, con altibajos, en torno al 45%. Se observa un aumento importante de la tasa de éxito en los dos últimos cursos, durante los cuales se ha hecho evaluación continuada. Desde el punto de vista de la evaluación, pensamos que la competencia del alumno en la materia se puede mejorar con una mayor exigencia, pero a cambio de ofrecer más oportunidades de demostrarlo.

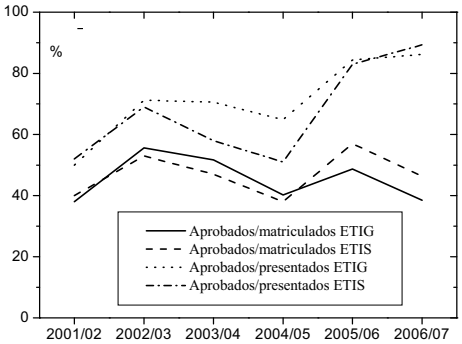


Figura 2. Evolución del rendimiento y del éxito

3. Conclusiones

La intención principal de esta experiencia de mejora del proceso de enseñanza-aprendizaje ha sido incidir sobre la motivación, la regularidad en el trabajo de la asignatura y en el necesario cambio de actitud, habitualmente negativo, del alumno hacia la asignatura de Fundamentos Físicos. Se ha exigido más para superar la asignatura, sin subir el nivel. Aunque no se puede afirmar que esta iniciativa haya mejorado el rendimiento (aprobados/matriculados), sí se puede asegurar que, manteniéndolo, los alumnos que aprueban dominan mejor y de una manera global los contenidos de la asignatura, o, en otras palabras, "saben más". La participación en la evaluación continua se ha ido manteniendo, pero una parte importante de los alumnos no se ha presentado sistemáticamente. Pensamos que aún se puede hacer mucho para mejorar la motivación de los alumnos a aprender los fundamentos físicos y hacer que el aprendizaje de esta materia se desligue de las puras exigencias del expediente: adaptar los contenidos a los intereses del ingeniero técnico en informática, mejorar y actualizar los ejemplos, los problemas y las prácticas de laboratorio, fomentar la discusión y el aprendizaje profundo de los conceptos, etc.

Para el curso 2007-08 uno de los objetivos seguirá en la línea de aumentar la asistencia y participación en clase. Nos planteamos evaluar la competencia de la responsabilidad en el trabajo haciendo un seguimiento personalizado en el que los alumnos tendrán de evidenciar los progresos realizados a lo largo del curso.

Agradecimientos

Programa de adaptación de los estudios a la convergencia europea de la Universidad de Girona promovido por el Vicerrectorado de Docencia.

Referencias

[1] Convenio Generalitat–Universidades http://www10.gencat.net/dursi/ca/un/eees_pla_pilot.htm

[2] Guía para la adaptación de las asignaturas al EEES http://www.udg.edu/Portals/79/VRDPA/04_Guia_EEES.pdf

Una invitación a la participación de la mujer en Ingeniería Informática

Itziar Cortés, Victoria Fernández, Edurne Larraza, Montse Maritxalar, Kepa Sarasola, Aitziber Zumarraga
(victoria.fernandez, edurne.larraza, montse.maritxalar, kepa.sarasola)@ehu.es
Universidad del País Vasco UPV/EHU
Facultad de Informática
Apartado 649-20018 Donostia-San Sebastián

Resumen

Actualmente el número de alumnas que se matricula en la UPV/EHU es superior al de alumnos. El mundo de la ingeniería ha sido y sigue siendo predominantemente masculino, aunque la mujer se vaya incorporando lenta pero paulatinamente. El caso de la informática es especialmente llamativo por el descenso del porcentaje de mujeres a pesar de ser una profesión reciente. En este trabajo presentamos una iniciativa destinada a impulsar la participación de la mujer en el área de la informática.

2. Motivación

El fenómeno de la baja participación de la mujer en informática se produce a nivel mundial y existen numerosos estudios tanto cualitativos como cuantitativos sobre el tema. También han surgido organizaciones que han desarrollado programas destinados a invertir la tendencia con resultados positivos en algunos casos. Entre los escasos estudios e iniciativas en el Estado Español podemos citar los siguientes [1, 2, 3].

Con la idea de que es importante motivar a las mujeres para que accedan a entornos que han sido tradicionalmente masculinos hemos creado un grupo de trabajo compuesto por tres profesoras, un profesor y dos alumnas de la facultad. El objetivo del grupo es doble. Por una parte queremos conocer la opinión del profesorado de la universidad sobre el tema y, por otra, impulsar iniciativas dirigidas a los centros de secundaria y bachiller.

3. Propuesta

La primera de las iniciativas desarrollada durante el curso 2006/07 ha sido la realización de unas Jornadas Culturales bajo el título *Informática y Género* con distintas actividades. Durante una semana se realizaron proyecciones de un DVD dirigidas al profesorado del centro. Asimismo se realizó una exposición gráfica de mujeres relevantes en el mundo de la informática, datos de matriculación donde se puede apreciar la mencionada tendencia y datos de profesorado. Para finalizar las jornadas se hizo una presentación dirigida a responsables de orientación de los centros de bachiller dividida en tres partes: presentación de la información estadística, proyección del DVD y realización de una mesa redonda. Dicho DVD consta de 3 capítulos: el primero con información general sobre los estudios de informática, el segundo con una breve historia de mujeres que han tenido influencia en el desarrollo de dicha disciplina (Figura 1) y el tercero con entrevistas a mujeres informáticas trabajadoras en distintos ámbitos laborales (Figura 2).

Figura 1. Algunas de las mujeres relevantes en el desarrollo de la informática

En primer lugar queremos destacar la buena acogida que las jornadas tuvieron por parte de las/os responsables de orientación que asistieron y la valoración positiva del DVD como material muy adecuado para el alumnado de sus centros.

Actualmente el DVD está en fase de corrección para su versión definitiva con el objetivo de proyectarlo en los centros de secundaria y bachiller.

Figura 2. Mujeres entrevistadas en el DVD

La segunda iniciativa ha sido elaborar una página Web que contiene información estadística, enlaces de interés y encuestas dirigidas a profesorado y alumnado de distintos ámbitos [http://www.ehu.es/e-makumeak]. A través de la página hemos realizado una encuesta al profesorado y becarias/os cuyos resultados se muestran a continuación.

Datos de la muestra y de participación

	Becarias/os		*Profesorado*	
	Total	*Respuestas*	*Total*	*Respuestas*
Hombres	43	8	73	30
Mujeres	15	3	49	25
Total	58	11	122	55

Pregunta 1. La familia, la escuela y la sociedad motivan más a los hombres que a las mujeres para los estudios tecnológicos.

	SI	*NO*	*SI (%)*	*NO (%)*
Hombres	24	14	63.2	36.8
Mujeres	18	10	64.3	35.7
Total	42	24	63.6	36.4

Pregunta 2. Es importante que aumente el número de mujeres en informática.

	SI	*NO*	*SI (%)*	*NO(%)*
Hombres	27	11	71.1	28.9
Mujeres	21	7	75.0	25.0
Total	48	18	72.7	27.3

Pregunta 3. Se matricularían más mujeres si tuvieran más información sobre los estudios

	SI	*NO*	*SI (%)*	*NO(%)*
Hombres	18	20	47.4	52.6
Mujeres	20	8	71.4	28.6
Total	38	28	57.6	42.4

Pregunta 4. La facultad debe impulsar programas dirigidos a los centros de secundaria para dar un punto de vista más adecuado de los estudios.

	SI	*NO*	*SI (%)*	*NO (%)*
Hombres	30	8	78.9	21.1
Mujeres	24	4	85.7	14.3
Total	54	12	81.8	18.2

De las diferencias que hay entre hombres y mujeres en las respuestas, la más destacable tiene que ver con la importancia que tiene la información a la hora de elegir una orientación profesional; en este caso un porcentaje mucho más elevado de mujeres que de hombres piensa que mayor información favorecería la participación de las mujeres.

Agradecemos al Vicerrectorado de Extensión Universitaria la financiación recibida. A las secretarias de la facultad por su ayuda en la recopilación de datos, a todas/os las/os asistentes a las jornadas su participación y sus sugerencias. Y al profesorado y becarias/os que han respondido la encuesta les agradecemos su colaboración desinteresada. Y por supuesto a todas las mujeres que han participado en la confección del DVD.

Referencias

[1] Dona (2004). *Programa Dona de la UPC*. http://www1.universia.net/CatalogaXXI/C10056PPESII1/E158567/index.html

[2] Fernández, V., Ruiz, T., Larraza, E., Maritxalar, M., Lazkano, E., Sarasola, K. *Evolución del número de mujeres en la matrícula de los estudios de informática en la Universidad del País Vasco, UPV/EHU.* XII Jornadas de Enseñanza Universitaria de la Informática, JENUI 2006.

[3] Sanz, V. (2005). *Women´s careers in Computer Engineering. Case study: Technical University of Madrid.* Third European Symposium on Gender and ICT. http://ict.open.ac.uk/gender/papers/sanz.doc

Pósteres
Sesión Especial

Las fases del aprendizaje.
Un esquema para el análisis y diseño de actividades
de enseñanza/aprendizaje.

Las fases del aprendizaje: Un esquema para el análisis y diseño de actividades de enseñanza/aprendizaje

Pau Bofill[1] , Joe Miró[2]

[1]Dept. Arq. de Computadors
Universitat Politècnica de Catalunya
Campus Nord, Jordi Girona, 1-3,
08034, Barcelona
pau@ac.upc.edu

[2]Dept. Mat. i Informàtica
Universitat de les Illes Balears
Campus UIB
07122, Palma de Mallorca
joe.miro@uib.es

Resumen

El objetivo de esta propuesta es establecer un marco común para el análisis y diseño de actividades de enseñanza/aprendizaje, basado en una caracterización esquematizada de las fases que debe superar el aprendiz en todo proceso de aprendizaje. Las fases que se proponen son: la motivación, el conocimiento, la comprensión, la aplicación, y la validación de todas ellas.

1. Introducción

El objetivo de la sesión de posters con un tema específico es promover la reflexión sobre el tema propuesto antes de las jornadas para poder llevar a cabo un debate fructífero durante las mismas. De esta manera el poster no es simplemente un instrumento para comunicar unas ideas sino el inicio de un proceso de reflexión común cuyos resultados se comunicarán a los asistentes de las jornadas.

El esquema para centrar el tema de la sesión que se propone a continuación no pretende ser exhaustivo ni original. Lo que buscamos es una descripción simple y operativa que permita contrastar las actividades de aprendizaje, tanto desde la perspectiva del estudiante como desde la perspectiva del profesor.

2. Las fases del aprendizaje

Para que se produzca aprendizaje el aprendiz debe pasar por las fases siguientes:

a. La *motivación*, entendida en sus dos acepciones:
   - La *motivación subjetiva*, que refleja la actitud del estudiante ante la materia y ante la actividad de estudio. Un estudiante está motivado si desea aprender los contenidos y, en consecuencia, está dispuesto a invertir esfuerzo para conseguirlo. La motivación puede ser interna (desea aprender porque le gusta) o externa (estudia porque se ve obligado). Sin un mínimo de motivación interna el aprendizaje está condenado al fracaso.
   - La *motivación de los contenidos*. Los contenidos están bien motivados si el estudiante comprende la finalidad del aprendizaje que va a emprender, y su relación con sus conocimientos actuales. Los contenidos deben estar contextualizados y relacionados con la experiencia del aprendiz. La motivación de contenidos debe ir acompañada de una formulación clara de los *objetivos formativos* (qué es lo que se va a aprender).

b. El *conocimiento* es la primera de las categorías de la taxonomía de Bloom. Un objetivo es de nivel conocimiento si requiere únicamente actividades de tipo memorización (p.ej., enumerar los nombres de los ríos de Galicia). En nuestro contexto, el conocimiento de los contenidos es una fase por la que hay que pasar para aprenderlos. Así por ejemplo, no se puede entender la segunda ley de Newton sin antes conocer su enunciado. Para que el estudiante pueda acceder a los conocimientos debe tener o ser capaz de conseguir el *material* y *recursos* necesarios.

c. La *comprensión*, la segunda de las categorías de Bloom, requiere ser capaz de explicar y relacionar conceptos. A menudo suele confundirse comprender con aprender (lo entiendo, ergo lo se) y existe la tendencia a pensar que la comprensión es automática por el mero hecho de asistir a una clase expositiva. La comprensión requiere que los

contenidos sean *significativos* para el aprendiz.

d. La *aplicación* (englobamos aquí las 3 categorías superiores de Bloom) es la capacidad de *transferir* lo aprendido a otro contexto y corresponde al aprendizaje profundo. La transferencia incluye procesos como la evaluación de la nueva situación, la identificación de los conocimientos pertinentes, y la adaptación de dichos conocimientos al nuevo contexto. La capacidad de transferencia implica la existencia de una red compleja de conocimientos que relacione el contexto y el lenguaje de aprendizaje con el contexto y el lenguaje de aplicación. La aplicación no debe confundirse con la mera realización de problemas tipo (conocimiento procedural) a partir de una teoría (conocimiento conceptual).

e. La *validación* es la realimentación necesaria para saber que vamos por el buen camino, y debe aplicarse a todas las fases anteriores (validar los objetivos, validar la información, validar la comprensión, validar la transferencia). La validación requiere retroalimentación (ejemplos, modelos, consultas o tutorías) y, en un contexto académico, requiere *evaluación formativa,* por contraposición a la mera evaluación selectiva de los exámenes finales.

3. Cuestiones abiertas

Mostramos aquí unas cuestiones abiertas sobre las que esperamos debatir durante las jornadas y que den lugar a unas conclusiones útiles y operativas.

- ¿Es coherente y operativo el esquema de fases propuesto?
- ¿Cómo podemos profundizar en la descripción de las distintas fases?
- ¿Cómo podemos analizar la actividad del estudiante en cada fase?¿Cómo verificamos su proceso de aprendizaje?¿Qué actividades son las más adecuadas en cada fase?¿Cómo podemos mejorar las actividades de estudio?
- ¿Cuál es la actividad del profesor?¿Cómo planificamos una unidad de aprendizaje?¿Qué actividades llevamos a cabo en tiempo de clase y cuáles como trabajo personal de cada estudiante?
- ¿Debe el profesor negociar con el estudiante su labor?¿Cómo?
- ¿Cómo se concreta esto en las disciplinas propias de la informática?
- ¿Cómo podemos aumentar la actividad del aprendizaje en cada fase?
- ¿Cómo integramos diferentes métodos de aprendizaje, como el aprendizaje cooperativo o el basado en proyectos, en las fases?
- ¿Cómo puedo usar las fases para estimular la creatividad?¿Qué puedo hacer en cada fase?

Referencias

[1] Bloom B.S., *Taxonomy of educational objectives,* Pearson Education, 1984.

Las fases del aprendizaje como pauta para el diseño de un libro interactivo de Introducción a los Ordenadores

Beatriz Otero y Pau Bofill
Dpto. Arquitectura de Computadores, Universidad Politécnica de Catalunya
C/ Jordi Girona, 1-3, 08034 Barcelona
{botero, pau}@ac.upc.edu

Resumen

En esta comunicación se utilizan las fases del aprendizaje (motivación, conocimiento, comprensión, aplicación y validación) como un criterio de diseño en la *propuesta* de un libro interactivo para la asignatura Introducción a los Ordenadores (programación en C), del primer semestre de la ETSETB-UPC.

En su concepción, el libro utiliza un enfoque basado en proyectos, y una organización de contenidos basada, implícitamente, en tipos abstractos de datos. El libro se desarrolla en 3 etapas, centradas en estructuras de datos de complejidad creciente.

El libro, que se realizará en soporte Moodle, ofrece, por una parte, recursos estáticos tales como objetivos, descripción de contenidos, ejemplos y ejercicios. Por otra parte, el libro ofrece recursos interactivos, como faqs, forums, cuestiones, entregas, pautas de autocorrección y estrategias de recuperación. Una guía de estudio organiza el trabajo del estudiante en itinerarios alternativos, y se proponen juegos y otras actividades de aprendizaje en grupo. El libro puede ser usado como herramienta para el autoaprendizaje.

La comunicación describe los elementos que van a componer el libro, y como mediante dichos elementos se pretenden cubrir cada una de las fases del aprendizaje.

1. Motivación

La asignatura Introducción a los Ordenadores (IO) de la ETSETB-UPC es un curso de introducción a la programación en C. En los últimos cursos el enfoque y la metodología de la asignatura han cambiado [3,4], y los profesores nos hemos encontrado con la necesidad de actualizar la documentación. En la actualidad estamos trabajando en un libro interactivo, sobre Moodle [5], cuya propuesta se describe aquí.

En dicha propuesta, los elementos diferenciales por los que creemos necesario escribir un nuevo libro son los siguientes: el enfoque orientado a proyectos, la perspectiva modular basada, implícitamente, en tipos abstractos de datos (TADs), la concepción como herramienta de autoaprendizaje, la interactividad proporcionada por Moodle, y la oferta de juegos y actividades de aprendizaje en grupo.

Para verificar que los recursos ofrecidos por el libro son completos, lo que hacemos en esta contribución es revisar que los recursos ofrecidos cubren cada una de las fases del aprendizaje: motivación, conocimiento, comprensión, aplicación y validación [1].

2. Criterios y componentes en el proyecto del libro interactivo para IO

En este apartado enumeramos los elementos a tener en cuenta en la elaboración del libro.

1. Objetivos formativos, organizados en 6 dimensiones: entorno, estructura, lenguaje, algorítmica, modularidad y desarrollo de programas.
2. Objetivos transversales: trabajo autónomo, conciencia de progreso y trabajo en grupo.
3. Organización del aprendizaje por proyectos.
4. Enfoque de contenidos basado implícitamente en TADs.
5. Articulación en 3 bloques o etapas. En cada etapa se construyen y utilizan TADs de complejidad creciente, que se utilizan en las etapas posteriores.
6. Recursos estáticos y dinámicos: faqs, glosarios, forums; objetivos específicos, proyectos parciales, modelos de evaluación; exposición de contenidos, ejemplos y modelos; cuestiones de comprensión, ejercicios; problemas resueltos, herramientas y criterios de evaluación y diagnóstico; estrategias de mejora; entrega, revisión y

mejora de proyectos; ¿qué he aprendido?, para profundizar, ejemplos de evaluación.
7. Planificación detallada del tiempo de dedicación del estudiante (total 120h).
8. Guía de estudio: ¿Cómo utilizar los recursos? Itinerarios. Soporte al autoaprendizaje.
9. Juegos y actividades para aprender en grupo: técnicas de aprendizaje cooperativo, concursos, juegos de mesa, etc.
10. Formato y herramientas de edición.

3. Análisis desde la perspectiva de las fases del aprendizaje

En este apartado revisamos la propuesta desde el punto de vista de las fases del aprendizaje.

1. Motivación: La organización basada en proyectos estimula la motivación en el sentido de que antepone la necesidad de unos conocimientos (los conocimientos necesarios para realizar el proyecto) a la adquisición de los mismos. El éxito alcanzado por el estudiante en cada una de las etapas genera satisfacción por aprender y motivación intrínseca. Los juegos y las actividades en grupo estimulan la motivación lúdica y social.
2. Conocimiento: Desde la perspectiva de un libro, facilitar el conocimiento significa explicitar los objetivos y modelos de evaluación, y poner al alcance del estudiante, de una forma u otra, todos los contenidos necesarios para alcanzar dichos objetivos. Una parte importante de los recursos ofrecidos en el epígrafe 6 del apartado anterior se destinan, naturalmente, a esta fase.
3. Comprensión: El enfoque por TADs y la organización por etapas facilita la comprensión, puesto que el objetivo último de cada etapa es el mismo, aprender a programar, pero con recursos de complejidad creciente. La claridad de las exposiciones, los ejemplos y modelos, las cuestiones, los ejercicios y la reflexión sobre el propio aprendizaje son herramientas para ayudar a la comprensión. A fin de tener en cuenta los distintos ritmos de aprendizaje, en la guía de estudio se proponen itinerarios alternativos.
4. Aplicación: Para esta fase se proponen dos tipos de actividades. Los proyectos de etapa, ya mencionados, y otros problemas complementarios para acabar de cubrir los objetivos de la etapa.
5. Validación: Esta fase va íntimamente ligada a la comprensión y la aplicación. El estudiante necesita saber si lo está haciendo bien. El uso de la herramienta Moodle, ampliada [2], facilita la interactividad con la propia herramienta, con el profesor/autor, y con los demás estudiantes/lectores. En nuestro caso, el libro ofrece faqs, forums, cuestiones resueltas, ejercicios resueltos, problemas resueltos con herramientas y criterios de valoración, y estrategias de mejora.

4. Conclusión

En esta comunicación hemos ilustrado como el análisis en términos de las fases del aprendizaje es una buena herramienta para el diseño de un recurso docente. En particular, para diseñar un libro interactivo para la asignatura IO de la ETSETB-UPC.

Agradecimientos

Este trabajo ha recibido la ayuda ICE de mejora a la docencia y la ayuda UPC de proyectos específicos. Además, ha sido apoyado por el Ministerio de Educación y Ciencia, bajo proyecto TIN 2004-07739-C02-01.
Por último, agradecemos a los estudiantes de los últimos cursos y a los compañeros de asignatura.

Referencias

[1] Bofill P., Miró J. *Las fases del aprendizaje: un esquema para el análisis y diseño de actividades de enseñanza/aprendizaje.* JENUI'07. Propuesta de tema específico.
[2] López L. *Módulo interactivo para la entrega y autoevaluación de ejercicios sobre Moodle.* PFC dirigido por Bofill, P., ETSETB, 2006.
[3] Otero B., Bofill P. *La asignatura Introducción a los ordenadores en el contexto de la experiencia Áncora, en la ETSETB-UPC.* JENUI'05, 229-236, 2005.
[4] Otero B., Bofill P. *Introducción a los Ordenadores: una forma diferente de aprender y de enseñar.* JENUI'06, 331-337, 2006.
[5] Web de la plataforma Moodle, http://moodle.org

Fomento de la motivación y uso de mapas conceptuales en el primer día de clase

Luis J. Saiz Adalid[1], Joaquín Gracia Morán[2]

[1]Dpto. de Informática de Sistemas y Computadores
ETS Ingenieros de Telecomunicación
Universidad Politécnica de Valencia
Camino de Vera s/n, 46022 Valencia
ljsaiz@disca.upv.es

[2]Dpto. de Informática de Sistemas y Computadores
Escuela Técnica Superior de Informática Aplicada
Universidad Politécnica de Valencia
Camino de Vera s/n, 46022 Valencia
jgracia@disca.upv.es

Resumen

El estudio de Informática por estudiantes de Ingenierías 'no Informáticas' es visto a menudo como un apéndice poco práctico de su titulación. A pesar de que cualquier estudiante de carreras técnicas tiene conocimientos de Informática a nivel de usuario, al profundizar en conceptos de arquitectura y estructura de computadores, la Informática se ve como una asignatura que deben aprobar, pero que no aportará grandes beneficios de cara al desempeño de su profesión.

En este sentido, cobra especial importancia el primer día de clase para motivar al alumnado ante la materia, situando los conocimientos que se van a adquirir en un contexto adecuado, de tal forma que pase a convertirse en algo complementario y no en un simple apéndice de su currículo.

El presente trabajo muestra la utilización de mapas conceptuales durante el primer día de clase para facilitar la comprensión de la asignatura, y de esta forma fomentar la motivación del alumnado.

1. Motivación

Hoy en día la mayoría de estudiantes universitarios de carreras técnicas tienen conocimientos informáticos a nivel de usuario. No obstante, pocos tienen conocimientos a un nivel superior. Muchas Ingenierías incluyen en su plan de estudios asignaturas enfocadas a ampliar conocimientos de arquitectura, estructura y tecnología de computadores en el currículo de sus titulados.

En Ingenierías no Informáticas, si estas asignaturas son de carácter troncal u obligatorio, por lo general son recibidas con poco interés por parte del alumnado, que considera la materia poco importante para su currículo, en vez de ser considerada un conocimiento complementario a la formación del Ingeniero.

En esta situación cobra especial importancia la primera clase [1], donde se producen las primeras impresiones hacia la asignatura. Fomentar la motivación desde el principio es fundamental para intentar convertir su aprendizaje en algo útil para el currículo del alumnado. Si conseguimos que comprendan la importancia de los conocimientos que se pretende transmitir a lo largo del curso para su futura carrera profesional, la motivación se incrementará. Si no lo conseguimos, la asignatura se convertirá en un obstáculo a salvar, pero sin ninguna motivación hacia su aprendizaje.

2. Primer día de clase y fomento de la motivación

El primer día de clase es un momento muy importante del curso, tanto para el alumnado como para los profesores. En lo que respecta al alumnado, hay que tener en cuenta sus posibles sentimientos: ansiedad, entusiasmo, etc. ante los nuevos conocimientos a adquirir y ante su nuevo profesor (qué profesor será, cómo evalúa, etc.) Otros posibles sentimientos son indiferencia o rechazo. En todos los casos, hay que conseguir motivar a los alumnos hacia la materia, aunque ésta no sea la base fundamental de su carrera. La primera impresión es muy importante, y será decisiva para el transcurso de la asignatura.

Los profesores debemos utilizar estrategias dirigidas a crear un clima favorable para motivar el interés, disminuir la ansiedad de los alumnos, mostrar una imagen aceptable (como "buenos profesores") y crear expectativas positivas.

La transmisión de conocimientos, clave en la enseñanza universitaria, implica también que el alumno necesita orientación para aprender a relacionar e integrar en su mente los nuevos conocimientos. Se pretende motivar al alumnado desde sus conocimientos previos. Situar la

asignatura en un punto intermedio entre su conocimiento básico de la Informática y los conocimientos adquiridos durante su carrera para relacionarlos y ampliarlos fomenta la motivación del alumnado. La percepción y comprensión de la relevancia del nuevo conocimiento hará que se incremente la motivación inicial.

3. Contexto y desarrollo de la experiencia

El presente trabajo se está aplicando en la asignatura de Informatización Industrial, una asignatura obligatoria del segundo semestre del 4º curso de la titulación de Ingeniero Industrial [2]. Los objetivos generales de esta asignatura son:

- Conocer los conceptos básicos relacionados con la arquitectura de los computadores.
- Comprender la aplicación industrial que tienen este tipo de sistemas.
- Ser capaz de analizar, seleccionar, diseñar, montar y poner a punto sistemas basados en microprocesadores, sobre todo aquellos que tienen una fuerte componente industrial.
- Tener los conocimientos básicos que permitan conocer y poder emitir juicios de valor sobre las distintas aplicaciones que día a día van utilizando los computadores como eje principal de las mismas.

En este trabajo los autores planteamos un cambio en la metodología docente en la asignatura, con objeto de fomentar la motivación. Pensamos que el uso de mapas conceptuales como técnica didáctica ayudará a situar la asignatura en un contexto en el que el futuro Ingeniero vea la utilidad del conocimiento que se le proporciona, y al mismo tiempo, conseguirá que ese conocimiento sea asimilado más fácilmente.

Básicamente, un mapa conceptual es un diagrama que relaciona conceptos [3]. La utilidad de esta excelente herramienta viene dada por la forma gráfica de representación, de tal manera que resulta más sencillo e intuitivo para el alumnado relacionar los diferentes conceptos que se ven en la asignatura y situarlos en su entorno adecuado.

Aprovechando este hecho, y utilizando la herramienta CmapTools [4], el uso de los mapas conceptuales desde la primera clase puede ser de gran ayuda durante el desarrollo de la misma. Un ejemplo de mapa conceptual es el que se muestra en la Figura 1, donde se puede ver el esquema del tema de introducción de la asignatura.

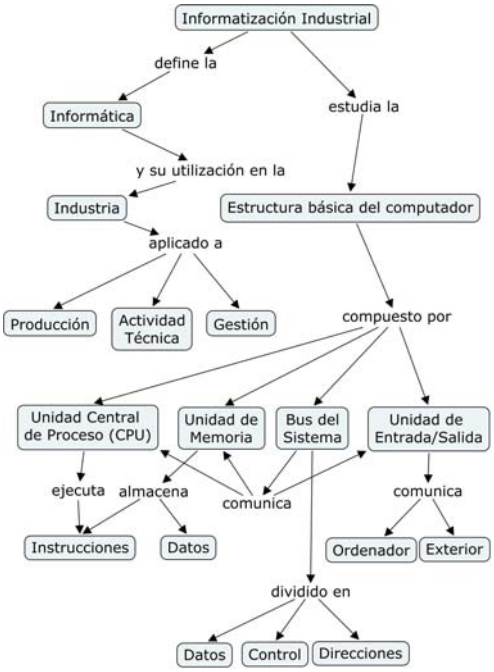


Figura 1. Mapa conceptual

Desde el primer día de clase, con la introducción general, planificación y organización del curso, es importante anticipar de lo que se va a hablar situándolo en su contexto.

Los mapas conceptuales, usados como técnica didáctica, pueden ser una herramienta válida para esta tarea, ya que facilitan en gran medida la comprensión. Esta comprensión de los conceptos básicos hará al alumno ver la relevancia de la materia, fomentando así su motivación desde el primer día de clase.

Referencias

[1] García, E. Apuntes del curso *"Habilidades básicas para el trabajo en el aula: primer día de clase"*, ICE-UPV, Valencia, 2006.
[2] http://www.etsii.upv.es/
[3] Cañas, A.J. et al. *Concep Maps: Theory, Methodology, Technology*, Proc. 1st Int. Conference on Concept Mapping, Pamplona, Spain, 2004.
[4] http://cmap.ihmc.us/

Diseño y evaluación de actividades de aprendizaje para un curso de introducción a la programación

Pedro J. Clemente, Alberto Gómez, Julia González,
Roberto Rodríguez, Encarna Sosa
Departamento de Ingeniería de Sistemas Informáticos y Telemáticos
Universidad de Extremadura
Escuela Politécnica, Avda. de la Universidad s/n, 10071 Cáceres
{pjclemente, agomez, juliagon, rre, esosa}@unex.es

Resumen

Cuando se plantea una asignatura de introducción a la programación, además de seleccionar los contenidos y las competencias transversales que se desean alcanzar, es fundamental diseñar de manera adecuada las actividades de enseñanza para cada una de las fases del aprendizaje, así como los mecanismos de evaluación que nos permitan determinar si se han alcanzado los objetivos previstos y calificar al estudiante.

Se presenta una propuesta de recursos, actividades y evaluación que pueden utilizarse en cada fase del aprendizaje de un alumno en un primer curso de introducción a la programación.

1. Introducción

Hasta ahora, en la asignatura de introducción a la programación para ingenieros informáticos que impartimos, no se plantean explícitamente actividades de enseñanza/aprendizaje para cada una de las fases del aprendizaje de un estudiante, sino que aparecen mezcladas a lo largo del desarrollo de la asignatura y en la evaluación.

Al analizar los resultados, se observa que no todos los estudiantes alcanzan el mismo nivel de dominio de la materia. En una prueba final, es frecuente que los alumnos superen un test de conceptos básicos y trazas, pero no sepan resolver correctamente problemas. Eso indica que han adquirido un cierto nivel de comprensión de la materia , pero no el suficiente para alcanzar un objetivo fundamental.

Estamos rediseñando la asignatura de forma que la materia se presente atendiendo a distintas fases del aprendizaje, usando los recursos más útiles en cada fase, proponiendo las actividades más adecuadas para ayudar a desarrollar las competencias y empleando los métodos de evaluación más apropiados para verificar que se han alcanzado los objetivos. En muchos casos, se trata simplemente de reorganizar y hacer explícitas muchas de las actividades que hemos venido desarrollando.

2. Las fases del aprendizaje

Hay muchas teorías sobre los factores que influyen en el aprendizaje de la programación [1]. Parece claro que el estudiante va pasando por diversas fases hasta alcanzar un conocimiento profundo de la materia.

Tomando la dimensión cognitiva de la taxonomía de Bloom y la propuesta [2] como referencias, para el aprendizaje de los fundamentos de programación (independientemente del paradigma o del lenguaje que se presente), se deben diseñar actividades e instrumentos de evaluación adecuados que incidan en cada fase: motivación, conocimiento, comprensión, aplicación, análisis, síntesis y evaluación.

Aquí damos por supuesta una motivación inicial por la materia en los estudiantes de ingeniería informática que, desgraciadamente, es menos común de lo que sería deseable.

En los siguientes apartados se detalla, para cada una de las fases, los objetivos principales, los recursos, actividades y herramientas de evaluación que consideramos adecuados.

<u>Fase 0: Motivación por los contenidos</u>

Objetivos: Interesar al estudiante en los nuevos temas y mostrar su necesidad.

Actividades: Presentar un problema atractivo que no se puede resolver todavía o que es crítico (sobre todo para aspectos arduos como la complejidad o la especificación).

<u>Fase 1: Conocimiento</u>

Objetivos: terminología, conceptos básicos (estructuras de control y de datos, módulos, objetos, parámetros, etc.), clasificaciones, uso y aplicación, abstracciones.

Actividades: clases magistrales, estudio individual, búsqueda de información.

Recursos: libros, transparencias, aulas virtuales, referencias a páginas web.

Evaluación: test, definiciones, listas de conceptos. (Es fácil la autoevaluación.)

Fase 2: Comprensión

Objetivos: comprender el modelo de ejecución para los conceptos tratados (estructuras de control, paso de mensajes, algoritmos de ordenación, manipulación de estructuras de datos, etc.); problemas básicos.

Actividades: laboratorio, clases magistrales combinadas con ejemplos y resolución de problemas. Por ejemplo, se obtienen buenos resultados al explicar, de forma explícita, los papeles que pueden desempeñar las variables en los programas [3], un conocimiento básico de los programadores expertos y que no suele detallarse.

Recursos: ejemplos de problemas básicos resueltos, libros, transparencias, animaciones.

Evaluación: aplicar algoritmos sobre distintos valores, realizar trazas.

Fase 3: Aplicación

Objetivos: resolver nuevos problemas, variaciones de los problemas básicos, tanto en los algoritmos como en los dominios de aplicación. Es fundamental partir de una buena colección de problemas y aplicaciones básicos ya resueltos y de aplicaciones de ejemplo correctas.

Actividades: modificación de programas, resolución de problemas, laboratorios cerrados.

Recursos: colecciones de problemas básicos resueltos y propuestos.

Evaluación: resolución de problemas, añadir nuevas restricciones a problemas conocidos, modificación de algoritmos, completar programas.

Fase 4: Análisis

Objetivos: análisis de la complejidad, comparación de algoritmos, buscar la solución más adecuada y más eficiente para distintos problemas relacionados.

Actividades y recursos: como en la fase anterior. Además, normas, ejemplos y actividades relacionadas con las competencias de comunicación oral y escrita.

Evaluación: test, presentaciones e informes con estudio de soluciones posibles, ventajas e inconvenientes (interesante para desarrollar otras competencias transversales como la capacidad de expresión oral y escrita).

Fase 5: Síntesis

Objetivos: realización de un proyecto de programación completo, a partir de una especificación, con un estudio completo de las posibilidades, documentación completa, etc.

Actividades: podrán desarrollarse de manera individual o en grupo. El proyecto se divide en fases (definición de requisitos, diseño de la interfaz, justificación de la estructura de datos); tras cada una de ellas se produce un informe.

Recursos: ejemplos de proyectos resueltos.

Evaluación: Documentación y código del proyecto. Debe ser una parte importante de la calificación final, valorando la asimilación de los conceptos, su aplicación y combinación, etc. Se puede realizar una presentación escrita o, si es posible, también oral. Aquí pueden fomentarse competencias de trabajo en grupo.

Fase 6: Evaluación

Objetivos: la capacidad de juzgar un trabajo ya realizado (por uno mismo o por otros) debe fomentarse en cada una de las fases, proponiendo ejemplos y herramientas para comparar la bondad o eficiencia de cada aspecto concreto.

Evaluación: comparación del trabajo propio con otros.

3. Conclusiones

Creemos que fijar objetivos para cada fase del aprendizaje en cada tema, así como hacerlos explícitos para el estudiante, junto con el uso de las actividades, recursos e instrumentos de evaluación adecuados, son necesarios para mejorar el aprendizaje de los fundamentos de la programación de ordenadores.

Además, así se sistematiza la enseñanza, alejando del estudiante la idea de que se requieren ciertas habilidades innatas para programar.

Referencias

[1] Shaffer, S. *A brief overview of theories of learning to program.* http://www.ppig.org /newsletters/2005-11.html.

[2] Bofill, P., Miró, J. *Las fases del aprendizaje: un esquema para el análisis y diseño de actividades de enseñanza/aprendizaje*. Web Jenui 2007. http://jenui2007.unizar.es.

[3] Sajaniemi, J. *The Roles of Variables Home Page*. http://cs.joensuu.fi/~saja/var_roles/.

ÍNDICE DE AUTORES

Albert, Rosa 121
Alcover, Rosa Mª 121,163
Aliane, Nourdine 139
Alonso, Amparo 65,113
Alonso, Amparo 65
Alonso, Javier 551
Alonso, Arantxa 293
Antonio, José 73
Aramendia, Pablo 387
Arboleda, Juan Pablo 519
Arévalo, Luis J. 81,511
Arnáiz, Andrés 425
Arnáiz, Álvar 425
Azuara, Guillermo 257
Baltasar, J. 371
Bellas, Francisco 65
Bemposta, Sergio 139
Benlloch, J. 163
Berenguer, V.J. 327
Bermejo, Miren 41,401
Blanco, Javier 461
Blesa, P. 163
Bofill, Pau 559,561
Brotons, V. 129
Bueno, Fco. Javier 537
Cabrero, Mariano J. 461
Calafate, Carlos T. 529
Calduch, Ángeles 121
Calduch, M. A. 121,163
Cano, Juan-Carlos 529
Canto, Pablo del 33
Casanova, Assumpció 263
Castillo, Ester, del 379
Castro-Schez, Jose Jesus 379
Cataldi, Zulma 187
Celma, M. 163
Cernuda, Agustín 487
Clares, Buenaventura 73
Claver, José M. 197
Clemente, Pedro J. 565
Cortés, Itziar 555
Díaz, Jesús Luis 145
Díaz, Margarita 179
Díaz, Marián 277
Domínguez, R. 409
Dorta, M. Isabel 549
Dorta, Pablo 549
Duran, Alejandro 171
Egido, Verónica 139
Escribano, Juan José 301
Escribano, César Alberto 301
Escudero, Gerard 505
Fabregat, Germán 197
Fdez-Riverola, F. 363,371,409
Fernández, Ray 41
Fernández, Agustín 19
Fernández, Josep 519
Fernández, José Raúl 537
Fernández, Victoria 387,555
Fernández, Javier 139
Fernández, Juan Luis 153
Fernández, Luis 319
Fernández, Victoria 387
Fernández, Rafael 441
Ferrando, E. 129
Ferri, C. 163
Fiol, Gabriel 353
Fiol, Catalina 353
Forcada, Jordi 455
Freixenet, Jordi 57
Fuertes, Gabriel 547
Gallego, Isabel 33
Gallego, Francisco J. 231
Gallego, Antonio Javier 241,285
Garayoa, Juan 263
García, Félix 11
García, Soledad 537
García, Javier 145
García, César 425
Garre, Miguel 545
Garrido, Piedad 213,223,547
Garriga, Ernest 293
Gayo, Daniel 487
Gil, D. 327
Glez-Peña, D. 363,371,409
Gómez, Alberto 565
Gómez, José Antonio 73
Gonzalez, E.J. 393
González, Daniel 495
González, Josep-Abel 553
González, Julia 205,565
González, José Daniel 179,495

González, Carolina 205
González, C. 269,393
González, E.J. 269
Gracia, Joaquín 89,249,563
Graells, Moisès 505
Guerrero, Josep M. 505
Guijarro, Bertha 113
Guitart, Isabel 311
Gutiérrez, Julián 401
Hamilton, A. 269
Hassan-Montero, Carmel 153
Heras, Noelia, de las 479
Hernández, Zenón 179
Hernández-Orallo, J. 163
Herrero, José R. 171
Hevia, Sonia 487
Hidalgo, Rubén 33
Hortolano, Julián 379
Iniesta, L. 163
Jacob, Inés 145
Lage, Fernando Javier 187
Larraza, Edurne 555
Larrosa, Javier 551
Laza, R. 363,371
Laza, R. 363,371
León, Coromoto 417
Llorens, Faraón 231,471
Llosa, Josep 19
López, Johann 33
López, José Manuel 33
López, David 171,551
López, Leobardo 337
López, Carlos 345
Lozano, Jose Antonio 401
Manzoni, Pietro 529
Marco-Galindo, M. Jesús 311
Marco-Simó, Josep M. 311
Marín, Raul 519
Maritxalar, Montse 555
Marqués, Francisco 263
Marticorena, Raúl 345
Martí-Farré, Jaume 293
Martín, J. 327
Martínez, Jorge 81,551
Martínez, Alejandra 479
Martínez, Jorge 81,511
Martínez, Antonio B. 519
Martínez, Francisco 213
Marzo, Josep Ll. 57
Más, J. 163
Méndez, J.R. 363,371,409
Merelo, Juan Julián 153
Merino, J. 269
Miranda, Gara 417
Miró, Joe 471,559
Miró, Margarita 353
Molina, Rafael 285
Montoyo, Javier 241
Mora, Javier 33
Moreno, M. 129
Moreno, L. 269,393,495
Moreno, Pablo 455
Naranjo, Fernando 213
Navarro, Juan J. 97
Oliver, Javier 145
Orduña, Pablo 145
Otero, Beatriz 293,561
Oyarzo, Jaime 537
Pajuelo, Alex 171
Palmer, Pere 471
Pardo, Carlos 345
Pavón, R. 363
Peña, Rosalía 545
Pinar, María A. 89,249
Piñeiro, J.D. 269
Polo, Antonio 81,511
Popescu, B. 269,393
Prat, Lluis 293
Prieto, Manuel 441
Puertas, Enrique 301
Pujol, Mar 241
Ramírez-Quintana, M. J. 163
Reboiro, M. 371
Requena, José 241
Reverte, Juan R. 285
Ribas, Lluís 105
Riesco, Miguel 277
Rius, Àngels 311
Robles, A. 163
Roda, José Luis 495
Rodríguez, Eva 33
Rodríguez, Roberto 565
Rodríguez, Juan Carlos 179
Rodríguez, M.Elena 311
Ruiz, D. 129,327
Ruíz, Francisco 11

Ruiz-Bertol, Fran J. 447
Sáenz, José María 145
Sáenz, Fernando 433
Saiz, Luis J. 563
Salgueiro, Fernando 187
Sánchez, Fermín 19
Sánchez-Carrasco, José Luis 205
Santamaria, Eduard 33
Santos, Germán 27
Sarasola, Kepa 555
Satorre, Rosana 285,471
Segura, Joan 49
Sigut, J. 269
Simón, M. Aránzazu 479
Solé, Josep 27
Soler, Josep 57
Soriano, A. 327
Sosa, Encarna 565
Suárez, María del Carmen 487
Timor, Pascual 455
Toledo, J. 269,393
Tornil, Sebastián 505
Torres, J. 269,393
Tovar, Edmundo 49
Tramullas, Jesús 223,547
Tricas, Fernando 153
Úbeda, Esmeralda 553
Valero-García, Miguel 33,97
Valiente, J. M. 163
Velasco, A.Josep 105
Vicent, M. J. 163
Vicente, Antonio J., de 441
Vidal, Vicente 263
Virgós, Ferran 49
Vivaracho, Carlos E. 479
Zarazaga-Soria, F. Javier 447
Zelaia, Ana 387
Zumarraga, Aitziber 555
Zúnica, L. R. 163