



Actas de las XXVII Jornadas sobre la Enseñanza Universitaria de la Informática

Emitiendo desde València
7/8 de Julio de 2021

ISSN: 2531-0607

Editores:

Carlos Catalán Cantero
Francisco Grimaldo Moreno

Revisores de formato:

Agustín Cernuda del Río
Antoni Pérez Poch



Actas de las XXVII Jornadas sobre la Enseñanza Universitaria de la Informática

Editores:

Carlos Catalán Cantero
Francisco Grimaldo Moreno

Revisores de formato:

Agustín Cernuda del Río
Antoni Pérez Poch

2531 – 0607



Este texto está sujeto a una licencia **Reconocimiento – NoComercial – CompartirIgual (by-nc-sa)**: No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

Presentación

Desde hace más de dos décadas, las *Jornadas de Enseñanza Universitaria de la Informática* (JENUI) son el foro donde los profesores universitarios de informática de España, y en menor medida de países latinoamericanos, reflexionan y discuten cómo mejorar los procesos de enseñanza/aprendizaje que atañen a la disciplina. Tras nacer en 1994 como una sesión de Informática dentro de las *II Jornadas sobre Innovación Docente en las Enseñanzas Técnicas Universitarias* celebradas en València, las primeras JENUI tuvieron lugar al siguiente año en Sitges (Barcelona), y de forma ininterrumpida se han celebrado desde 1996 hasta este año. Las JENUI propiciaron la creación de la *Asociación de Enseñantes Universitarios de la Informática* (AENUI) en el año 2000, y desde entonces esta asociación es la responsable de las Jornadas, entre otras actividades que realiza cada año.

Esta edición número XXVII será de nuevo organizada por la Escola Tècnica Superior d'Enginyeria de la Universitat de València (ETSE-UV). Debido a la pandemia asociada al COVID-19 en 2020 las Jornadas debieron cambiar a un formato virtual. Aunque este cambio no afectó a la parte académica, el programa de actividades síncronas se redujo al mínimo. Este año las Jornadas han vuelto a repetir comité organizador con la esperanza de que su formato volviese a ser el habitual. Desgraciadamente la prolongación de la pandemia nos ha vuelto a obligar a celebrarlas en 2021 en formato virtual. No obstante, a diferencia del año pasado, en esta edición el formato se desarrollará de manera totalmente síncrona (*streaming*), y con un número de actividades mayor, durante dos días completos. Esta edición es una nueva oportunidad, en un camino iniciado en años anteriores, para abrir las Jornadas a otros ámbitos interesados en la docencia de calidad de la informática. Por este motivo, se ha extendido la invitación para a institutos de enseñanza media, colegios y asociaciones profesionales.

La apertura de JENUI se materializa en la habitual publicación en formato abierto de las actas¹, así como en el mantenimiento de la modalidad individual de inscripción gratuita añadida en la edición anterior. Esta permitirá acceder a todas las actividades desarrolladas. Además, en esta edición se ha añadido una modalidad de inscripción institucional a precio reducido. Mediante ella miembros de centros, departamentos universitarios u otras instituciones podrán participar en las Jornadas. Al igual que el año pasado se usará la plataforma virtual de la Universitat de València para el acceso a las diferentes sesiones y actividades plenarias. Los vídeos de todas las presentaciones serán publicados en el canal de AENUI en YouTube².

El programa contará con diez sesiones paralelas de presentación de trabajos, así como sesiones plenarias de candidatos a mejores trabajos y de pósteres. Como en ediciones pasadas, en estas sesiones los autores realizarán sus presentaciones y podrán resolver las dudas y atender a las sugerencias del resto de participantes. También se mantienen en esta edición los trabajos relevantes. Esta categoría permite a autores de contribuciones previamente publicadas en revistas o congresos de prestigio su presentación en las JENUI, además de incluir un resumen de las mismas en las actas.

Como en todas las ediciones en noviembre se lanzó la llamada a presentar contribuciones sobre la docencia universitaria de la Informática en las categorías experiencia docente, recurso docente, investigación en educación y reflexión. En particular, en las siguientes áreas temáticas:

- Aplicación de las TIC al proceso de enseñanza-aprendizaje
- Calidad y evaluación de la docencia
- Compromiso social y medioambiental
- Perspectiva de género en los estudios de Informática.
- Didáctica en los estudios de Ingeniería Informática
- Didáctica de la Informática en otros estudios de grado y máster
- Didáctica de la Informática en secundaria: programación, pensamiento computacional y máster de profesorado

¹JENUI 2021: <http://actasjenui.aenui.net>

²JENUI 2021: <http://bit.ly/AENUI>

- Desarrollo de competencias transversales y profesionales
- Estrategias institucionales en los estudios de Ingeniería Informática
- Evaluación del aprendizaje
- Mejoras pedagógicas en las asignaturas
- Organización curricular y planes de estudio
- Optimización del tiempo y del trabajo del profesor
- Promoción de los estudios de Ingeniería Informática
- Trabajos fin de grado, prácticum, proyectos y participación de alumnos en la investigación
- Docencia y evaluación semipresencial y en línea

La última área ha sido añadida este año con el fin de acoger, especialmente en esta edición, a trabajos relacionados con la docencia durante la pandemia del COVID-19, aunque la idea es mantenerla en próximas ediciones con carácter general.

Se han recibido 72 contribuciones, las cuales han sido evaluadas por un Comité de Programa formado por 124 revisores. Se han efectuado 5 revisiones por contribución, 4 de ellas de contenido y 1 de formato. Y se han aceptado 45 trabajos, de los cuales 5 son recursos docentes, 11 son pósteres y 3 son trabajos relevantes. Esto supone una tasa de aceptación global del 62,5 %. Esta tasa refleja la gran calidad de los trabajos presentados dado que se ha mantenido el alto nivel de exigencia. Todos los trabajos aceptados se incluyen en estas actas organizados por las sesiones en las que han sido presentados: diez sesiones temáticas paralelas, una sesión de pósteres y una sesión de candidatos a mejores trabajos. Estos últimos han sido seleccionados por el Comité directivo entre los mejor valorados por los revisores del Comité de Programa.

Uno de los actos más esperados dentro de las JENUI es la entrega del Premio AENUI a la Calidad e Innovación Docente, que reconoce la labor realizada en aras de la calidad e innovación docente en el área de la docencia universitaria de la Informática. Este año ha recaído en Mercedes Marqués, profesora de la Universitat Jaume I de Castelló. Entre las actividades plenarias habrá dos conferencias impartidas por los ganadores del premio AENUI en las últimas dos ediciones. Cabe recordar que el ganador en 2020 fue David López, profesor de la Universitat Politècnica de Catalunya.

El programa incluirá también una interesante mesa redonda sobre “Perspectiva de género en los estudios de informática”. Entre sus participantes estarán: Paloma Moreda, profesora del Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Alicante; María Villarroja, profesora del Departamento de Informática e Ingeniería de Sistemas de la Universidad de Zaragoza; Andrea Benito, alumna de la Universidad de Granada; Juan Pablo Peñarrubia, Vicepresidente del Consejo General de Colegios Profesionales de Ingeniería en Informática (CCII) y Silvia Rueda, profesora del Departamento de Informática de la Universitat de València. La perspectiva de género tiene todavía más relevancia en esta edición, ya que uno de los trabajos presentados en esta área temática será premiado por el Consejo General de Colegios Profesionales de Ingeniería Informática (CCII), dentro del acuerdo de colaboración que tiene firmado con AENUI.

También se entregará el premio SISTEDES al mejor trabajo presentado en el ámbito de la Ingeniería del Software y de las tecnologías del desarrollo de software, el cual es otorgado por la sociedad SISTEDES³.

Como todos los años, en el marco de las Jornadas se realizará la asamblea anual de AENUI. En ella se efectuará la votación de los miembros de la asociación para elegir los dos mejores trabajos. El resultado se mostrará en la página web de JENUI 2021⁴, entregándose el premio antes de la clausura de las Jornadas. Los autores de algunos de los candidatos a mejor trabajo serán invitados a enviar versiones extendidas para ser publicados en la revista IEEE-RITA (Revista iberoamericana de tecnologías del aprendizaje)⁵, que está indexada en SJR, SCImago y Emerging Sources en la Web of Science.

Para finalizar, queremos expresar nuestro agradecimiento a los verdaderos protagonistas de JENUI cada año, los cuales han hecho posible la celebración de esta edición: los autores y revisores. A los primeros por enviar contribuciones y participar en las actividades organizadas, y a los segundos por su esfuerzo en conseguir mantener el nivel de calidad de JENUI en esta edición. Gracias a todos por seguir apostando por la calidad de la docencia, y participando y mejorando las Jornadas a pesar de las circunstancias adversas que estamos viviendo.

Comité directivo JENUI 2021
Comité organizador JENUI 2021

³SISTEDES: <http://www.sistedes.es>

⁴JENUI 2021: <http://jenui2021.uv.es>

⁵Revista IEEE RITA: <http://rita.det.uvigo.es/IEEERITA2018>

Comités

Comité directivo

Coordinadores de programa

- Carlos Catalán Cantero, *Universidad de Zaragoza*
- José Manuel Badía Contelles, *Universitat Jaume I de Castelló*

Coordinadoras de artículos relevantes

- Marcela Genero Bocco, *Universidad de Castilla-La Mancha*
- Yania Crespo González-Carvajal, *Universidad de Valladolid*

Coordinador de relaciones institucionales

- Faraón Llorens Largo, *Universitat d'Alacant*

Coordinador de publicaciones

- Germán Moltó Martínez, *Universitat Politècnica de València*

Coordinadores de difusión

- José Antonio Cruz Lemus, *Universidad de Castilla-La Mancha*

Coordinadores de actas

- Agustín Cernuda del Río, *Universidad de Oviedo*
- Antoni Pérez Poch, *Universitat Politècnica de Catalunya*

Internacional

- Marta E. Barría Martínez, *Universidad de Valparaíso, Chile*
- César Collazos Ordóñez, *Universidad del Cauca, Colombia*
- Ernesto Cuadros-Vargas, *Universidad de Ingeniería y Tecnología, Perú*
- Martha Dunia Delgado Sapena, *Ciudad Universitaria José Antonio Echeverría, Cuba*
- Oswaldo Juan Figueroa Domejean, *Universidad Católica de San Pablo, Bolivia*
- Emilio Gutiérrez, *Universidad Católica, Paraguay*
- Raúl Antonio Aguilar Vera, *Universidad Autónoma de Yucatán, México*
- Efraín Rodrigo Fonseca, *Universidad de las Fuerzas Armadas ESPE, Ecuador*
- Cecilia Sanz, *Universidad Nacional de La Plata, Argentina*
- Martín Solari Buela, *Universidad ORT, Uruguay*
- Claudia Werner, *Universidade Federal do Rio de Janeiro, Brasil*

Comité organizador

- **Francisco Grimaldo Moreno, Coordinador**, *Escola Tècnica Superior d'Enginyeria, Universitat de València*
- **Inmaculada Coma Tatay, Coordinadora**, *Escola Tècnica Superior d'Enginyeria, Universitat de València*
- Xaro Benavent García, *Escola Tècnica Superior d'Enginyeria, Universitat de València*
- Jose A. Boluda Grau, *Escola Tècnica Superior d'Enginyeria, Universitat de València*
- Carmen Botella Mascarell, *Escola Tècnica Superior d'Enginyeria, Universitat de València*
- Jose M. Claver Iborra, *Escola Tècnica Superior d'Enginyeria, Universitat de València*
- Adriana Dapena Janeiro, *Facultade de Informática, Universidade da Coruña*
- Esther De Ves Cuenca, *Escola Tècnica Superior d'Enginyeria, Universitat de València*
- Esther Durá Martínez, *Escola Tècnica Superior d'Enginyeria, Universitat de València*
- Ricardo Ferris Castell, *Escola Tècnica Superior d'Enginyeria, Universitat de València*
- Ariadna Fuertes Seder, *Escola Tècnica Superior d'Enginyeria, Universitat de València*
- Daniel García Costa, *Escola Tècnica Superior d'Enginyeria, Universitat de València*
- Míriam Gil Pascual, *Escola Tècnica Superior d'Enginyeria, Universitat de València*
- Emilia López-Iñesta, *Facultat de Magisteri, Universitat de València*
- Francisco Martínez Gil, *Escola Tècnica Superior d'Enginyeria, Universitat de València*
- Ignacio Panach Navarrete, *Escola Tècnica Superior d'Enginyeria, Universitat de València*
- Fernando Pardo Carpio, *Escola Tècnica Superior d'Enginyeria, Universitat de València*
- Carlos Pérez Conde, *Escola Tècnica Superior d'Enginyeria, Universitat de València*
- Sandra Roger Varea, *Escola Tècnica Superior d'Enginyeria, Universitat de València*
- Silvia Rueda Pascual, *Escola Tècnica Superior d'Enginyeria, Universitat de València*

Comité de programa cuerpo de revisores

Silvia Abrahao, *Universidad de Politecnica de Valencia*
Silvia Acuña Castillo, *Universidad Autonoma de Madrid*
Pedro Pablo Alarcón Cavero, *Universidad Politécnica de Madrid*
Marc Alier, *Universidad Politécnica de Cataluña*
Carlos Álvarez Martínez, *Universidad Politécnica de Cataluña*
Pedro Javier Álvarez Pérez-Aradrós, *Universidad de Zaragoza*
Fernando Álvarez García, *Universidad de Oviedo*
Darío Álvarez Gutiérrez, *Universidad de Oviedo*
Mancia Anguita López, *Universidad de Granada*
Fidel Aznar Gregori, *Universidad de Alicante*
Pilar Bachiller Burgos, *Universidad de Extremadura*
José Manuel Badía Contelles, *Universitat Jaume I de Castellón*
José Ángel Bañares, *Universidad de Zaragoza*
David Bañeres Besora, *Universitat Oberta de Catalunya*
Sergio Barrachina Mir, *Universitat Jaume I de Castellón*
Marta Barría Martínez, *Universidad de Valparaíso*
Bruno Baruke Zanon, *Universidad de Burgos*
Antonio Becerra, *Universidad de Almería*
Luis Bengochea, *Universidad de Alcalá*
Jaime Benjumea Mondéjar, *Universidad de Sevilla*
José Vicente Berná Martínez, *Universidad de Alicante*
Carlos Blanco Bueno, *Universidad de Santander*
José Manuel Burgos Ortiz, *Universidad Politécnica de Madrid*
Cristina Cachero Castro, *Universidad de Alicante*
Xavier Canaleta Llampallas, *Universitat Ramon Llull*
Óscar Cánovas Reverte, *Universidad de Murcia*
Maria José Casañ Guerrero, *Universidad Politécnica de Cataluña*
M^a Asunción Castaño Álvarez, *Universitat Jaume I de Castellón*
Pedro Ángel Castillo Valdivieso, *Universidad de Granada*
Carlos Catalán Cantero, *Universidad de Zaragoza*
Agustín Cernuda del Río, *Universidad de Oviedo*
José M. Claver Iborra, *Universidad de Valencia*
César A. Collazos, *Universidad del Cauca Colombia*
Yania Crespo, *Universidad de Valladolid*
Jose Antonio Cruz Lemus, *Universidad de Castilla-La Mancha*
Ernesto Cuadros Vargas, *Universidad de Ingeniería y Tecnología Perú*
María Valeria De Castro, *Universidad Rey Juan Carlos*
Antonio Jesús De Vicente Rodríguez, *Universidad de Alcalá*
Adelaida Delgado Domínguez, *Universitat de les Illes Balears*
Jessica Díaz Fernández, *Universidad Politécnica de Madrid*
Josuka Díaz Labrador, *Universidad de Deusto*
Juan José Escribano, *Universidad Europea de Madrid*
Joaquín Ezpeleta Mateo, *Universidad de Zaragoza*
José Luis Fernández Alemán, *Universidad de Murcia*
Francisco Javier Fernández Baldomero, *Universidad de Granada*
Jesualdo Tomás Fernández Breis, *Universidad de Murcia*
Xavier Ferré, *Universidad Politécnica de Madrid*

Efraín Rodrigo Fonseca, *Universidad de las Fuerzas Armadas ESPE Ecuador*
Ariadna Fuertes Seder, *Universidad de Valencia*
Joaquim Gabarró, *Universidad Politécnica de Cataluña*
Jesús Gallardo Casero, *Universidad de Zaragoza*
Francisco José Gallego Durán, *Universidad de Alicante*
María José García García, *Universidad Europea de Madrid*
Jesús García Molina, *Universidad de Murcia*
Félix García Rubio, *Universidad de Castilla-La Mancha*
Marcela Genero Bocco, *Universidad de Castilla-La Mancha*
Consolación Gil Montoya, *Universidad de Almería*
Alberto Gómez Mancha, *Universidad de Extremadura*
Marco Antonio Gómez Martín, *Universidad Complutense de Madrid*
Julia González Rodríguez, *Universidad de Extremadura*
Francisco Grimaldo Moreno, *Universidad de Valencia*
José María Gutiérrez Martínez, *Universidad de Alcalá*
Juan Francisco Rodríguez Herrera, *Universidad de Almería*
Emilio Insfran, *Universidad de Politecnica de Valencia*
Inés Jacob Taquet, *Universidad de Deusto*
María Carmen Juan Lizandra, *Universidad Politécnica de Valencia*
Elena Jurado Málaga, *Universidad de Extremadura*
María del Carmen Lacave Rodero, *Universidad de Castilla-La Mancha*
Martín Llamas Nistal, *Universidad de Vigo*
Faraón Llorens Lago, *Universidad de Alicante*
Silvia Llorente Viejo, *Universidad Politécnica de Cataluña*
David López Álvarez, *Universidad Politécnica de Cataluña*
Emilia López Iñesta, *Universidad de Valencia*
Gabriel López Millán, *Universidad de Murcia*
Carlos López Nozal, *Universidad de Burgos*
Julio Alberto López-Gómez, *Universidad de Castilla la Mancha*
Adolfo Lozano Tello, *Universidad de Extremadura*
Sergio Luján Mora, *Universidad de Alicante*
Hermenegilda Macià, *Universidad de Castilla la Mancha*
María Jesús Marco Galindo, *Universitat Oberta de Catalunya*
Mercedes Marqués Andrés, *Universitat Jaume I de Castellón*
Antonio Martí Campoy, *Universidad Politécnica de Valencia*
Raúl Marticorena Sánchez, *Universidad de Burgos*
Rafael Martínez Gasca, *Universidad de Sevilla*
Jorge Más Estellés, *Universidad Politécnica de Valencia*
Manuel Mejías Risoto, *Universidad de Sevilla*
Marcos Menárguez Tortosa, *Universidad de Murcia*
Rafael Molina Carmona, *Universidad de Alicante*
Germán Moltó, *Universidad Politécnica de Valencia*
Ángel Mora Bonilla, *Universidad de Málaga*
Francisco Mora Lizán, *Universidad de Alicante*
Ana Moreno, *Universidad Politécnica de Madrid*
Antonio Moreno Ribas, *Universitat Rovira i Virgili*
Jose Carlos Moreno Úbeda, *Universidad de Almería*
Joaquín Nicolás Ros, *Universidad de Murcia*
Gloria Ortega López, *Universidad de Almería*
Beatriz Otero Calviño, *Universidad Politécnica de Cataluña*
José Ignacio Panach, *Universitat de València*
María Luisa Parody Núñez, *Universidad de Sevilla*

Vicente Pelechano Ferragud, *Universidad de Politecnica de Valencia*
 Jenifer Pérez Benedi, *Universidad Politécnica de Madrid*
 Antoni Perez Poch, *Universidad Politécnica de Cataluña*
 Ángel Francisco Perles Ivars, *Universidad Politécnica de Valencia*
 Álvaro Prieto Ramos, *Universidad de Extremadura*
 María del Mar Pujol López, *Universidad de Alicante*
 Miguel Angel Redondo Duque, *Universidad de Castilla-La Mancha*
 Lluís Ribas Xirgo, *Universidad Autònoma de Barcelona*
 María Elena Rodríguez González, *Universitat Oberta de Catalunya*
 Roberto Rodríguez Echeverría, *Universidad de Extremadura*
 Francisco Pascual Romero, *Universidad de Castilla-La Mancha*
 Silvia Rueda, *Universitat de València*
 Mercedes Ruiz, *Universidad de Cádiz*

Francisco Ruiz González, *Universidad de Castilla-La Mancha*
 Fernando Saénz Pérez, *Universidad Complutense de Madrid*
 Fermín Sánchez Carracedo, *Universidad Politécnica de Cataluña*
 Cecilia Sanz, *Universidad Nacional de La Plata Argentina*
 Jesús Serrano Guerrero, *Universidad de Castilla-La Mancha*
 Martín Solari, *Universidad ORT Uruguay*
 Vicente Ramón Tomás López, *Universitat Jaume I de Castelló*
 Belén Vaquerizo García, *Universidad de Burgos*
 Juan Manuel Vara Mesa, *Universidad Rey Juan Carlos*
 Ángel Jesús Varela Vaca, *Universidad de Sevilla*
 J. Ángel Velázquez Iturbide, *Universidad Rey Juan Carlos*
 Aurora Vizcaino Barceló, *Universidad de Castilla-La Mancha*
 Francisco Javier Zarazaga Soria, *Universidad de Zaragoza*

Artículo pendiente de revisión

Artículo pendiente de revisión

Artículo pendiente de revisión

Artículo pendiente de revisión

Artículo pendiente de revisión

Artículo pendiente de revisión

Artículo pendiente de revisión

Artículo pendiente de revisión

Artículo pendiente de revisión

Artículo pendiente de revisión

Artículo pendiente de revisión

Artículo pendiente de revisión

Artículo pendiente de revisión

Artículo pendiente de revisión

Artículo pendiente de revisión

Artículo pendiente de revisión

Artículo pendiente de revisión

Artículo pendiente de revisión

Artículo pendiente de revisión

Artículo pendiente de revisión

Artículo pendiente de revisión

Artículo pendiente de revisión

Artículo pendiente de revisión

Artículo pendiente de revisión

Artículo pendiente de revisión

Artículo pendiente de revisión

Artículo pendiente de revisión

Aprendizaje basado en proyectos de analítica de software en estudios de ciencia e ingeniería de datos

Silverio Martínez-Fernández, Cristina Gómez, Xavier Franch
Departamento de Ingeniería de Servicios y Sistemas de Información (ESSI)
Universitat Politècnica de Catalunya (UPC) – BarcelonaTech
Barcelona, España
{silverio.martinez, cristina.gomez, xavier.franch}@upc.edu

Resumen

El Grado en Ciencia e Ingeniería de Datos (GCED) de la Universidad Politècnica de Catalunya (UPC) forma a sus estudiantes en análisis e ingeniería de datos con una base matemática y habilidades propias de la ingeniería que les permiten modelizar y resolver problemas complejos. En el último curso, los estudiantes aún no han analizado conjuntos de datos que permitan ayudar a mejorar la calidad del software, ni explotado buenas prácticas de ingeniería del software en proyectos de ciencias de datos. En el curso 2020/2021 se cursa por primera vez la asignatura Temas Avanzados en Ingeniería de Datos 2 (TAED2) en el GCED. Esta experiencia docente relata la introducción del aprendizaje basado en proyectos en el laboratorio de la asignatura, donde los estudiantes aplicaron la ciencia de datos y buenas prácticas de ingeniería del software para analizar y detectar mejoras en la calidad del software. Realizamos un caso de estudio en siete sesiones de laboratorio con 33 estudiantes del GCED. Los estudiantes encontraron la realización del proyecto positiva para su aprendizaje. En el proyecto, destacaron la utilidad de usar un proceso de minería de datos CRISP-DM y una buena práctica de ingeniería del software: una convención de estructura de un proyecto software aplicada a un proyecto de ciencia de datos.

Abstract

The Bachelor's Degree in Data Science and Engineering (GCED) at UPC-BarcelonaTech trains its students in data analysis and engineering with a mathematical basis and engineering skills that allow them to model and solve complex problems. In the last year, students have not yet analyzed data sets that help improving software quality, nor have they exploited good software engineering practices in data science projects. In the 2020/2021 academic year, the subject Advanced Topics in Data Engineering 2 (TAED2) is taken at the GCED for the first time. This teaching experience shows the introduction of project-based learning in the subject laboratory, where students applied data

science and good software engineering practices to analyze and detect improvements in software quality. We carried out a case study in seven laboratory sessions with 33 GCED students. The students found the execution of the project positive for their learning. In the project, they highlighted the utility of using a CRISP-DM data mining process and good software engineering practice: a software project structure convention applied to a data science project.

Palabras clave

Ingeniería del software, ciencia de datos, procesos de minería de datos, analítica de software, buenas prácticas de software, docencia.

1. Introducción

Debido a la creciente adopción de las tecnologías Big Data, la ciencia de datos ha ganado atención en múltiples áreas de conocimiento, incluyendo la ingeniería del software. Las empresas de software tienen la necesidad de comprender las habilidades de aprendizaje necesarias para organizar equipos multidisciplinares con profesionales expertos en software y en datos. Por ejemplo, Microsoft e IBM han informado de la necesidad de organizar equipos multidisciplinares compuestos por científicos de datos e ingenieros del software para llevar a cabo el desarrollo y mejora de calidad de software basados en datos [12, 15].

Consecuentemente, se están realizando acciones de innovación docente para enseñar ingeniería del software para la creación de sistemas basados en inteligencia artificial [10, 11] (v. Sección 5).

Esta ponencia muestra la experiencia docente de diseñar, ejecutar y evaluar una nueva actividad usando la metodología de *aprendizaje basado en proyectos (ABP)* [6, 19], en el laboratorio de la asignatura TAED2 del GCED en la UPC. El objetivo de dicha actividad es llevar a cabo un proyecto (en adelante, el proyecto) de analítica de software para detectar mejoras en la calidad de software de código abierto. El proyecto tiene dos aspectos de ingeniería del software innovadores para los alumnos: (i) la aplica-

ción de ciencia de datos en un nuevo dominio: detección de problemas en la calidad de software; y (ii) la generación de una estructura de proyecto de datos basada en una buena práctica de ingeniería del software.

El diseño, ejecución, y evaluación de este estudio tiene tres contribuciones principales:

- Estudiar la viabilidad de aplicar ABP para detectar problemas en la calidad de software por alumnos de último curso del GCED, así como identificar sugerencias de mejora para tratar los problemas encontrados.
- Analizar la utilidad de un proceso de minería de datos (CRISP-DM, v. Sección 2.1) y una convención de estructura de proyecto de ciencia de datos (Cookiecutter Data Science, v. Sección 2.2) por parte de los estudiantes del GCED en el contexto de su proyecto.
- Reportar los objetivos de analítica de software (v. Sección 2.3) escogidos por los estudiantes a partir de un conjunto de datos relativos a proyectos de software de código abierto (TechDebt, v. Sección 2.4).

El resto de este documento se estructura de la siguiente manera. La Sección 2 introduce el contexto y artefactos usados en la asignatura. La Sección 3 resume la guía docente, junto a objetivos de aprendizaje y las actividades del proyecto de laboratorio. La Sección 4 muestra los resultados de la actividad en la primera edición de la asignatura, y reflexiona sobre aspectos fuertes y a considerar en próximas ediciones. La Sección 5 muestra el trabajo relacionado. Finalmente, la Sección 6 concluye este documento.

2. Contexto y artefactos de la asignatura

Esta asignatura es obligatoria y del séptimo cuatrimestre del GCED en la UPC. El punto de partida de esta experiencia docente fueron las habilidades aprendidas por los estudiantes durante los tres primeros años del GCED. A continuación, mostramos los artefactos más importantes utilizados en el laboratorio (CRISP-DM y estructura de proyecto), así como la problemática de analítica de software y el conjunto de datos usado.

2.1. CRISP-DM

CRISP-DM es el acrónimo de “*Cross-industry standard process for data mining*”. Según dos encuestas realizadas en 2014 y 2007, es el proceso de minería de datos más usado en la industria¹. De hecho, en

la actualidad, adaptaciones suyas se aplican no sólo en minería de datos, sino en el contexto de aprendizaje automático [18]. Otros procesos de minería de datos son KDD [5] y SEMMA² (v. Kurgan et al. para comparativa [13]).

CRISP-DM fue el resultado de un proyecto europeo de investigación. El objetivo era crear un proceso estándar para la comunidad de minería de datos. Está basado en la experiencia práctica de cómo profesionales conducen proyectos de minería de datos. Entre las 300 empresas que colaboraron en una serie de talleres, cabe destacar tres principales del consorcio europeo: Daimler Chrysler (después Daimler-Benz), SPSS (después ISL), y NCR. Destacamos que el estándar no es propietario, está disponible gratuitamente, y es agnóstico respecto a la industria, las herramientas, o los dominios de aplicación.

Una de las contribuciones principales de CRISP-DM es su modelo de referencia. Este modelo de referencia contiene seis fases (v. Figura 1): comprensión del negocio, comprensión de los datos, preparación de los datos, modelado, evaluación, y despliegue. Asimismo, el estándar incluye la metodología, guía de usuario, y plantilla para informes de cada una de las fases del proyecto. Toda la información se puede consultar en la guía de usuario³.

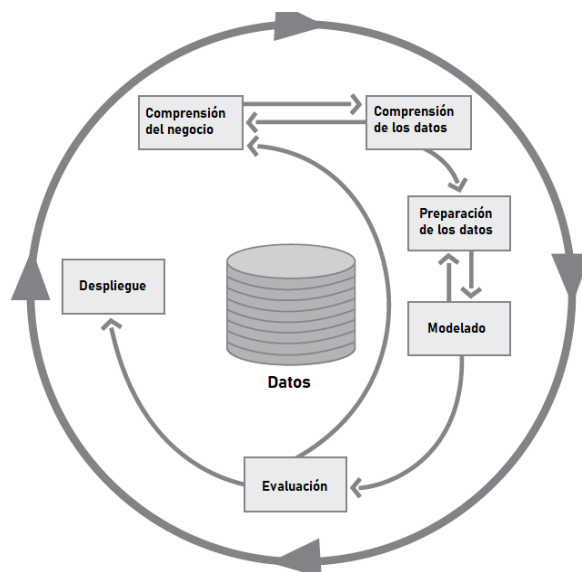


Figura 1: Fases de CRISP-DM.

Recientemente, el uso de este proceso en equipos de desarrollo software ha mostrado ser efectivo para la mejora de calidad de software dirigida por datos en empresas [7]. Por este motivo, estudiamos su viabilidad y utilidad en un contexto académico en esta asignatura y experiencia docente.

¹ <https://www.kdnuggets.com/2014/10/crisp-dm-top-methodology-analytics-data-mining-data-science-projects.html>

² <https://documentation.sas.com/?docsetId=emref&docsetTarget=n061bzurmej4j3n1jn8bbjm1a2.htm&docsetVersion=15.1>

³ https://spss.ch/upload/1107356429_CrispDM1.0.pdf

2.2. Estructura de proyecto de ciencia de datos

Los proyectos de ciencia de datos contienen scripts, cuadernos interactivos, archivos de configuración, archivos de datos, informes, documentación, pruebas y figuras. Un problema común de estos proyectos de ciencia de datos es su reproducibilidad. Si algún científico de datos o desarrollador necesita ejecutar un script, cuaderno interactivo, o modelo de inteligencia artificial, necesita saber dónde se encuentra la versión correcta de los modelos, los datos de entrenamiento, etc. Este problema es análogo al que se encuentran los equipos de desarrollo de software cuando necesitan encontrar algún artefacto dentro de la estructura de su proyecto software.

Por este motivo, recientemente se ha impulsado el uso de múltiples convenciones de estructuras de proyecto de ciencia de datos, como la ofrecida para Tensorflow⁴ y Azure⁵. En el momento del inicio de la asignatura, la estructura de proyecto de ciencia de datos con mejor valoración en GitHub era Cookiecutter Data Science⁶, lo que motivó su elección.

Cookiecutter Data Science: (i) proporciona una estructura de repositorio genérica para modelos, datos, configuración, etc.; (ii) permite iniciar un proyecto personalizado por el autor, el nombre del proyecto, la licencia, la versión de Python, etc.; (iii) especifica una convención de nombres para los cuadernos, proporciona un directorio para los modelos entrenados y una estructura de carpetas para separar los datos en bruto de los datos externos/procesados.

2.3. Analítica de software

La analítica de software consiste en utilizar enfoques basados en los datos para obtener información reveladora y procesable que ayude a los profesionales del software en sus tareas relacionadas con los datos [8]. Tiene como objetivos mejorar la productividad del proceso de desarrollo, la calidad de los sistemas de software, y/o la experiencia de los usuarios finales [19]. En este laboratorio, el proyecto se centra en la detección de mejoras de la calidad de los sistemas software a partir del análisis de sus datos. Para ello, se comienza ejecutando tareas de minería en datos brutos (p.ej., código fuente, repositorios de fallos) para descubrir patrones (p.ej., construir un modelo de predicción de fallos), y finalmente producir información procesable (p.ej., programar tareas específicas de garantía de calidad, asignar recursos).

2.4. El conjunto de datos de deuda técnica

Existen múltiples conjuntos de datos para realizar analítica de software para diversas problemáticas de ingeniería del software⁷ (p.ej., estimación de costes, elicitación de requisitos, etc.). Se puede encontrar en repositorios propios como PROMISE⁸ o incluso en Kaggle⁹.

En esta experiencia docente nos centramos en un problema específico de calidad de software: la deuda técnica. La deuda técnica refleja el costo implícito del trabajo adicional causado por elegir una solución fácil en un punto concreto de desarrollo en lugar de utilizar una alternativa de más calidad, pero más costosa en su implementación. Para estudiar este problema, usamos el conjunto de datos “TechDebt” [14].

En el momento del inicio de la asignatura, el conjunto de datos TechDebt consistía en 33 proyectos Java de código abierto con diferentes datos calculados y/o extraídos por diferentes herramientas. Para cada proyecto podemos encontrar: métricas de análisis estático de código realizado por la herramienta de SonarQube, las tareas de los backlogs de Jira, commits, datos sobre refactorización extraídos por la herramienta RefactorMiner, y fallos identificados por el algoritmo SZZ (para más información, v. [14]).

El conjunto de datos TechDebt, con su documentación y trabajo de limpieza de los datos previo, facilita que realizar el proyecto en siete semanas sea factible.

3. Guía docente de la asignatura

En esta sección explicamos los objetivos de aprendizaje específicos de la asignatura, y nos centramos en el objeto de estudio de esta experiencia docente: el proyecto que realizarán los estudiantes.

3.1. Objetivos de aprendizaje

La asignatura TAED2 tiene cuatro objetivos. En los cuatro objetivos el sujeto es el estudiante, se utiliza un verbo relacionado con un nivel de la taxonomía de Bloom [2, 4], y se especifican el contenido y la circunstancia. Los objetivos son:

- Interpretar los conceptos básicos de la Ingeniería del Software, especialmente en relación al uso y explotación de datos.
- Aplicar y analizar conceptos y métodos referentes al uso de datos provenientes del proceso de desarrollo en la gestión de la calidad del sistema software.

⁴ <https://github.com/Mrgemy95/Tensorflow-Project-Template>

⁵ <https://github.com/Azure/Azure-TDSP-ProjectTemplate>

⁶ <https://github.com/drivendata/cookiecutter-data-science>

⁷ <https://danroddgar.github.io/DASE/repositories.html>

⁸ <http://promise.site.uottawa.ca/SERepository/>

⁹ <https://www.kaggle.com/>

- Aplicar conceptos y métodos referentes al uso de datos obtenidos durante la utilización del sistema por parte de sus usuarios en la planificación de nuevas versiones evolutivas.
- Describir conceptos y métodos referentes al uso de datos de contexto obtenidos durante la utilización del sistema, para la adaptación autónoma de los sistemas en tiempo de ejecución en respuesta a cambios.

El objetivo número dos se trata en profundidad en el proyecto de laboratorio, alcanzando los niveles 3 (aplicación) y 4 (análisis) de la taxonomía de Bloom.

3.2. El proyecto

El proyecto es la actividad principal del laboratorio de TAED2. Se ha escogido la metodología de *aprendizaje basado en proyectos (ABP)* debido a la necesidad de equipos multidisciplinarios en analítica de software, la complejidad del objetivo a alcanzar, y para favorecer la participación activa y crítica de los estudiantes identificando oportunidades de mejora de calidad de software. El ABP es una metodología de aprendizaje activo en la que los alumnos trabajan sobre un problema del mundo real durante un periodo de tiempo.

Equipos. Los equipos se componen de 3-4 estudiantes. Además, el profesor semanalmente da retroalimentación sobre el proyecto y resuelve dudas.

Plan de trabajo. El proyecto se organiza de forma iterativa e incremental, siguiendo las fases de CRISP-DM, con flexibilidad para saltar de manera ágil a fases anteriores si es necesario, tal y como recomienda el propio proceso. El proyecto se desarrolla en siete semanas, como muestra el Cuadro 1.

Sem.	Tareas de cada sesión de laboratorio
1	Inicio del proyecto: Equipos creados y carpetas de Google drive configuradas. Introducción al conjunto de datos. Los equipos comienzan a definir el objetivo del proyecto.
2	Discusiones a partir de las lecturas asignadas. Presentación de la selección inicial del objetivo del proyecto.
3	Presentación de la comprensión del negocio y los pasos iniciales de la comprensión de los datos.
4	Presentación de la comprensión de los datos (y plazo final para refinar el objetivo si es necesario). Inicio de la preparación de los datos. Sesión “pregúntame lo que quieras” con expertos.
5	Presentación de pasos de preparación de datos y modelado.
6	Presentación de pasos de modelado (si hay alguna actualización) y evaluación
7	Presentación final, incluyendo reflexión del paso de despliegue. Entrega de: (a) informe final (siguiendo CRISP-DM) y (b) software (siguiendo la plantilla del proyecto). Cierre de proyecto.

Cuadro 1: Calendario del proyecto.

En la primera sesión se forman los equipos de trabajo. También se introduce el conjunto de datos y se explican sus características.

A partir de la segunda semana, los equipos presentan sus progresos en el proyecto y reciben retroalimentación semanalmente. Comienzan desde la fase 1 de CRISP-DM, especificando un objetivo de negocio como parte de la comprensión del negocio, hasta el despliegue de la presentación final.

Además del ABP, cabe destacar la integración de otras metodologías. Un ejemplo es la *clase invertida* [3]. En la primera semana, se asigna a cada estudiante de un equipo la lectura de un artículo entre los siguientes [1, 12, 14, 17]. Estos artículos ayudan a escoger un objetivo de analítica de software y a comprender el conjunto de datos. Entre las dos primeras sesiones, los alumnos individualmente leen el artículo asignado. En la segunda semana, deben primero presentar y consensuar un resumen con componentes de otros equipos que tienen el mismo artículo asignado. Posteriormente, presentan el resumen consensuado de cada artículo a los miembros de su equipo. Por otra parte, aunque se presentan las fases de CRISP-DM en clase, cada semana los alumnos tienen que leer detalladamente las actividades a realizar de una fase de CRISP-DM en la guía de uso fuera del aula (presentado en la Sección 2.1). Esto es importante de cara a la presentación sobre los progresos del proyecto en la siguiente semana, pues como muestra el Cuadro 1 y hemos indicado anteriormente, cada semana los equipos van progresando en las fases de CRISP-DM en el proyecto (ver Figura 1).

En el ecuador del laboratorio, se realiza una sesión “*pregúntame lo que quieras*” por videoconferencia. En dicha sesión, en el curso 2020/2021, participaron tres panelistas de ciencia de datos que forman parte de equipos de desarrollo de software. Los panelistas eran de diferente sexo, lugar de trabajo (empresa, instituto de investigación, y universidad), y país. Después de una breve introducción, durante una hora los alumnos podían realizar cualquier pregunta, como por ejemplo consejos sobre el proyecto o sobre su futura carrera laboral.

Hacia el final del proyecto, en la semana 6 se crea un *libro digital* con los informes de todos los equipos al que todos los estudiantes pueden acceder. Este libro está inspirado en otras asignaturas (p.ej., de arquitectura de software) que fomentan el *aprendizaje colaborativo* [21]. El libro, que es un documento en Google Docs, permite que cada equipo pueda ver el resumen de los proyectos de otros equipos en la última semana. Así, cada equipo puede ver qué hace el resto de equipos antes de la presentación final.

Evaluación. La evaluación del laboratorio consistió en un 70% del informe final entregado y un 30% del software generado. El informe debe seguir las guías de CRISP-DM, que también define cómo

realizar un informe. Desde la semana 1, los alumnos van documentando el proyecto con la ayuda de CRISP-DM. En el software se evalúa el código y su adherencia a la convención de estructura de proyecto de ciencia de datos. Los equipos disponen de una *rúbrica* [9] para conocer cómo se evalúan tanto el informe final como el software.

4. Evaluación del curso piloto 2020/2021

La primera edición de esta asignatura ha constado de 33 estudiantes divididos en dos grupos. Entre los dos grupos, se formaron nueve equipos de 3-4 estudiantes. Esta sección muestra respectivamente los datos coleccionados antes de comenzar el proyecto, los objetivos escogidos por los estudiantes para el proyecto, y la retroalimentación y opinión del alumnado y profesorado tras realizar el proyecto.

4.1. Cuestionario inicial

Para preparar las sesiones de laboratorio, se realizó un cuestionario inicial dos semanas antes de su inicio. Se preguntó sobre experiencia en lenguajes de programación y entornos de desarrollo, experiencia en procesos de minería de datos, proyectos de ciencia de datos, y conocimientos en ingeniería del software.

La participación en el cuestionario, voluntario, fue del 90% (30 alumnos sobre 33). Para medir la experiencia, usamos una escala ordinal con los siguientes valores sobre experiencia: ninguna (nunca usado), básica (usado en un proyecto o asignatura), intermedio (usado de 2 a 5 asignaturas o proyectos), avanzado (usado en más de 5 asignaturas o proyectos), y n/a (no aplica). Los datos se muestran en la Figura 2.

Respecto a lenguajes de programación, 28 alumnos (93%) tenían un conocimiento avanzado de Python. C++ y R también eran conocidos, con 28 alumnos con un conocimiento entre intermedio y avanzado. Respecto a Java, sólo 8 alumnos tenían algún conocimiento. Aunque no se preguntó explícitamente por ellos, en la sección abierta al menos 5 alumnos mencionaron conocimientos en Matlab y SQL.

El entorno de desarrollo más usado por los estudiantes es Jupyter Notebook, con conocimiento avanzado por 27 de los alumnos, seguido de R Studio, con 13 alumnos con conocimiento avanzado.

Puesto que el objetivo del laboratorio no es la enseñanza de nuevos lenguajes o entornos de programación sino la aplicación de los aprendidos en los tres primeros años del GCED, el laboratorio se diseñó para usar Python y Jupyter Notebook.

A su vez, los equipos tenían libertad para elegir otros lenguajes y herramientas si estaba justificado.

Antes del séptimo cuatrimestre, un 94% de los alumnos no tenían conocimientos de procesos de

minería de datos, como CRISP-DM, KDD, o SEMMA. Igualmente, el 83% de los alumnos no conocía qué era la Ingeniería del Software. Esto se alinea con los objetivos de aprendizaje 1 y 2 de la asignatura.

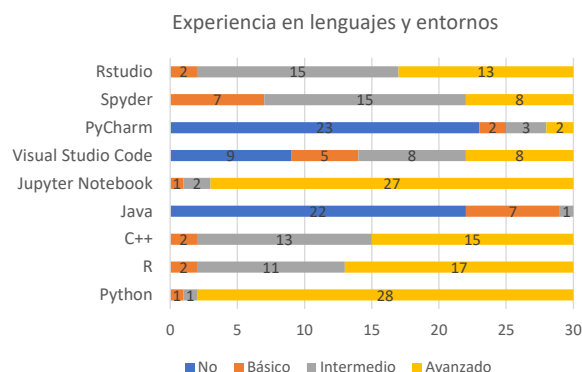


Figura 2: Experiencia de los alumnos en lenguajes de programación y entornos de desarrollo.

Finalmente, destacar que 22 alumnos tenían experiencia previa en conducir proyectos de ciencia de datos (73%), aunque ninguno en el contexto de datos relacionados con el software. Ejemplos de proyectos de ciencia de datos realizados anteriormente a TAED2 por los alumnos son: predecir si un usuario clicará en publicidad web, crear subtítulos de películas, desarrollar algoritmos de reconocimiento de imágenes y de clasificación de textos.

4.2. Objetivos resueltos en los proyectos

Cada equipo tenía que definir un objetivo aplicando conceptos y métodos sobre el uso de datos en el proceso de desarrollo para la gestión de calidad de sistemas software. Por lo tanto, dentro del objetivo genérico de mejorar la calidad de software, cada equipo podía centrarse en un aspecto a su elección. Para ello, disponían de un listado de ideas, y era posible trabajar sobre una nueva idea propuesta. Cinco equipos se centraron en la predicción y clasificación de commits respecto a su introducción de fallos o hediondez del código (en inglés “code smell”). Por ejemplo, un equipo integró su modelo de predicción en un sistema en el cual a partir de característica de un commit (como descripción, número de líneas añadidas/modificadas/eliminadas, porcentaje de comentarios, complejidad de código, etc.) predecía la probabilidad de que contuviera fallos. Un equipo clasificó tipos de commits, sin elaborar algoritmos de predicción. Dos equipos se centraron en la clasificación de los desarrolladores creando perfiles en base a las características de su código (p.ej., uso de comentarios, complejidad de las funciones, realización de tareas de refactorización, trabajo en resolver fallos, etc.). Este tipo de perfil de desarrolladores son usados en empresas software para la contratación de personal. Otro equipo clasificó proyectos por su capacidad de mantenimiento y sostenibilidad en el tiempo.

4.3. Cuestionario final

El objetivo del cuestionario final era conocer el grado de satisfacción del alumnado. La primera pregunta trataba de ver la utilidad de aplicar un proceso de minería de datos. Como muestra el Cuadro 2, la aplicación de CRISP-DM les parece útil para alcanzar los objetivos de la Sección 4.2 (Media = 4.21, sobre 5). Esta pregunta fue seguida de otra pregunta con espacio abierto para dejar comentarios. En esta pregunta abierta, 17 alumnos dijeron explícitamente que volverían a usar CRISP-DM en otros proyectos: *“definitivamente lo tendré en cuenta porque es una manera muy sensata y bien estructurada de trabajar”, “es bueno para la organización, como una convención y como una buena manera de saltar a los proyectos de ciencia de datos. También creo que es muy eficiente. Por supuesto que sí, lo utilizaré, y también lo he estado utilizando”, “sí, fue útil para saber lo que había que hacer, no perder de vista el proyecto y lograr nuestros objetivos”, “creo que es útil ya que nos da la pauta de las partes de un proyecto, así como lo que hay que incluir en cada una de ellas”*. Sin embargo, tres alumnos expresaron explícitamente algunas dudas respecto a su uso futuro. Algunos ejemplos: *“no creo que tenga una opinión firme sobre CRISP-DM todavía, ya que no creo que lo haya explorado lo suficiente. Pero creo que la guía era un poco confusa a veces y un poco larga”, “en mi opinión, lo encuentro un poco rígido y repetitivo, pero supongo que es la mejor manera de diseñar una estructura como esta, ya que hay que otorgarle coherencia”*.

La segunda pregunta trataba de ver la utilidad de aplicar una estructura de proyecto. Como muestra el Cuadro 2, los alumnos consideraron útil utilizar la estructura del proyecto "Cookiecutter Data Science" para fomentar la corrección y la reproducibilidad

(Media = 4.04). Dicha estructura de proyecto no recibió comentarios negativos. Entre los comentarios positivos podemos encontrar, por ejemplo: *“es bastante interesante ver el equilibrio entre la libertad de organización, la exploración y el trabajo con ‘otros ojos’, y el orden que se consigue con estructuras de proyectos como ‘Cookiecutter Data Science’”, “encuentro muy útil la estructura del proyecto ‘Cookiecutter Data Science’. Me ha ayudado a organizar mejor el software y, sobre todo, a ser limpio. Parece ‘profesional’ cuando tu software está bien estructurado”*. Sin embargo, desde el profesorado, sí tenemos que indicar que durante el proyecto hubo varias dudas respecto a algunos apartados opcionales de la plantilla que fueron planteadas por e-mail.

La sesión *“pregúntame lo que quieras”* fue bien acogida (Media = 4.11). De hecho, los alumnos opinaron que podría haber sido incluso más larga de una hora. Al ser videoconferencia, los panelistas sí mostraron su interés en ver las caras de los alumnos durante la sesión, y no sólo de la alumna que realizaba la pregunta.

La actividad peor acogida fue el libro digital (Media = 3.45). Este libro digital está inspirado en otros cursos para fomentar el *aprendizaje colaborativo* [21]. Los estudiantes no dieron retroalimentación explícita de los problemas. Una hipótesis a investigar en las próximas ediciones es si mejora la acogida si se crea este libro digital desde el comienzo del proyecto.

Las tres últimas preguntas del cuestionario (Cuadro 2) fueron sobre aspectos de ABP. Los alumnos están de acuerdo que les fue útil para aprender (Media = 3.86), con que el proyecto fue factible en un contexto de software (Media = 4.52), y que el trabajo en equipo les ayudó a aprender (Media = 4.31). Estas preguntas fueron seguidas de otra pregunta con espacio abierto para dejar comentarios sobre problemas o retos resueltos durante el proyecto.

Pregunta	N	Mínimo	Máximo	Media	Mediana	Moda	σ
La aplicación de CRISP-DM me parece útil para alcanzar los objetivos de nuestro proyecto.	29	2	5	4.21	4.00	4 ^a	.819
Considero que utilizar la estructura del proyecto “Cookiecutter Data Science” es útil para fomentar la corrección y la reproducibilidad.	28	2	5	4.04	4.00	5	.999
La sesión “pregúntame lo que quieras” me ayudó a aprender	27	2	5	4.11	4.00	4	.801
El libro digital común en el laboratorio me ayudó a aprender	29	1	5	3.45	3.00	3	.985
El proyecto en el laboratorio me ayudó a aprender	29	1	5	3.86	4.00	4	1.093
El trabajo en equipo en el laboratorio me ayudó a aprender	29	3	5	4.31	4.00	5	.712
Para mi equipo en el laboratorio era factible realizar un proyecto de ciencia de datos en un contexto de software.	29	4	5	4.52	5.00	5	.509

Cuadro 2: Cuestionario final de satisfacción del laboratorio. Las respuestas se calificaron en la escala de 1 a 5 con el siguiente significado: 1 - muy en desacuerdo; 2 - en desacuerdo, 3 - neutral, 4 - de acuerdo, 5 - muy de acuerdo.

^a Existen múltiples modas. Se muestra el valor más pequeño.

Seis estudiantes mencionaron explícitamente que les costó la parte de comprensión de los datos. Algunos ejemplos: *“los principales problemas fueron la comprensión de los datos y el significado exacto de las variables, a pesar de lo cual fue bastante bien”, “el principal problema fue lidiar con el nuevo conjunto de datos, es decir, tratar de entender qué significa cada variable, sus valores...”* Otros dos estudiantes encontraron retos al definir un objetivo y criterios de satisfacción como implica la comprensión de negocio en CRISP-DM: p.ej., *“los principales problemas tal vez hayan sido definir el conjunto de datos/objetivos, etc., pero con los comentarios y la comunicación cada semana todo ha funcionado bastante bien”*. Finalmente, nueve estudiantes indicaron que el principal reto fue hacer el proyecto en siete semanas: p.ej., *“no fue muy difícil, pero tal vez un poco más de tiempo habría sido apreciado”*.

Finalmente, al igual que en el cuestionario inicial, se preguntó sobre conocimientos en ingeniería del software (un 90% afirmó que conocía qué era, respecto al 17% inicial) y CRISP-DM (un 100% afirmó que tenía experiencia, respecto al 6% inicial) en el cuestionario final. Además, en la última pregunta de satisfacción general con el laboratorio, tan sólo un 10% se mostró insatisfecho con los ejercicios de laboratorio.

4.4. Discusiones y posibles mejoras

En general, desde el profesorado hacemos una evaluación positiva y estamos satisfechos con el rendimiento académico (nota media total de TAED2 de 7.5, y nota media de los proyectos 8.9). Con la experiencia del curso piloto, hay datos suficientes para reflexionar sobre la retroalimentación de los estudiantes y realizar mejoras en la próxima edición.

Respecto a la rúbrica de evaluación, pensamos realizar dos mejoras. La primera, simplificar la rúbrica mostrando explícitamente qué partes de la estructura de proyecto son opcionales. Por otra parte, incluir un punto de evaluación en el ecuador del proyecto con la mitad del informe (hasta la comprensión de los datos).

Respecto al contenido, hemos visto que los alumnos han mostrado mucho interés en la convención de estructura de proyecto de ciencia de datos. Con esto, analizaremos reemplazar el último objetivo de aprendizaje de la asignatura y añadir un nuevo objetivo para enseñar otras buenas prácticas de ingeniería del software para este tipo de proyectos [16]. Como un estudiante pidió en los comentarios abiertos: *“más educación sobre buenas prácticas”*. Esto puede facilitar el despliegue del modelo realizado en el proyecto en un sistema basado en inteligencia artificial.

Finalmente, para resolver los principales retos mencionados (dificultad de comprensión de los datos y tiempo limitado) la primera semana se introducirán más detalladamente conceptos de las variables del conjunto de datos (en el curso se hizo brevemente con documentación del conjunto de datos), y se pondrá un límite de páginas para el informe realizado para reducir el tiempo empleado en documentación.

5. Trabajo relacionado

En la última década, han proliferado numerosos cursos de analítica de software asociados a minería en repositorios de software y la conferencia MSR (del inglés Mining Software Repositories). Un excelente ejemplo en España es el curso “Data Analysis in Software Engineering using R” de Daniel Rodríguez y Javier Dolado¹⁰.

Con la última ola de inteligencia artificial y aprendizaje profundo, podemos ver también la necesidad de enseñar técnicas de ingeniería del software para crear sistemas basados en inteligencia artificial [16]. Dos experiencias docentes relevantes se han llevado a cabo en el último año en la Universidad Carnegie Mellon [11] y la Universidad Fontys de Eindhoven [10].

La experiencia docente de este documento es innovadora en aplicar analítica de software con estudiantes de ciencias de datos, así como enseñar buenas prácticas de ingeniería del software (p.ej., convención de estructura de proyectos) a este perfil de estudiantes en la UPC. Como trabajo futuro, se estudiará la inclusión de más buenas prácticas de ingeniería del software en TAED2. Además, en la última fase de despliegue, se estudiará evaluar la creación de un pequeño sistema software basado en el modelo creado y evaluado. Un ejemplo creado por un equipo de TAED2 ha sido una aplicación web donde se incluían las características de un commit y predecía la probabilidad de contener fallos.

6. Conclusiones

En esta experiencia docente comenzamos preguntándonos: ¿cómo educar a estudiantes de ciencias de datos para realizar proyectos de analítica de software? Esta ponencia ha mostrado la experiencia docente de diseñar, ejecutar y evaluar una nueva actividad de ABP en el laboratorio de la asignatura TAED2 del GCED en la UPC. El objetivo de dicha actividad de proyecto ha sido detectar oportunidades de mejora de la calidad de software en repositorios de código abierto en base a datos.

¹⁰ <https://danrodgar.github.io/DASE/>

La experiencia de aplicar ABP en este contexto ha sido positiva. Además, los estudiantes destacaron la utilidad del proceso de minería de datos CRISP-DM y la estructura del proyecto de datos.

Agradecimientos

Nos gustaría agradecer a Davide Taibi y Santiago del Rey, por la ayuda con el conjunto de datos, y también a Adam Trendowicz, Davide Taibi, y Christine Doig-Cardet por su participación como panelistas en la sesión de “pregúntamente lo que quieras”. También agradecemos a los profesores del Institut de Ciències de l'Educació (ICE) de la UPC, por las actividades para mejorar la docencia. Finalmente, esta experiencia docente está parcialmente financiada por el programa “Beatriz Galindo”.

Referencias

- [1] Andrew Begel y Tom Zimmermann. Analyze this! 145 questions for data scientists in software engineering. En *36th International Conference on Software Engineering*, pp. 12-23, 2014.
- [2] Mary Besterfield-Sacre, Larry Shuman, Harvey Wolfe, et al. Defining the outcomes: A framework for EC-2000. *IEEE Transactions on Education*, 43(2), 100-110, 2000.
- [3] Jacob Bishop y Matthew Verleger. The flipped classroom: A survey of the research. En *ASEE national conference proceedings*, Vol. 30, No. 9, pp. 1-18, 2013.
- [4] Benjamin Bloom. Taxonomy of educational objectives. Vol. 1: Cognitive domain. *New York: McKay*, 20, 24, 1956.
- [5] Ronald Brachman y Tej Anand. The Process of Knowledge Discovery in Databases: A First Sketch. En *KDD workshop* vol. 3, pp. 1-12, 1994.
- [6] Clive Dym et al. Engineering design thinking, teaching, and learning. *Journal of engineering education*, 94(1), 103-120, 2005.
- [7] Christof Ebert, Jens Heidrich, Silverio Martínez-Fernández, et al. Data science: technologies for better software. *IEEE Software*, 36(6), 66-72, 2019.
- [8] Harald Gall, Tim Menzies, Laurie Williams, et al. Software development analytics (dagstuhl seminar 14261). En *Dagstuhl Reports* (Vol. 4, No. 6). *Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik*, 2014.
- [9] Fiorina Gatica-Lara y Teresita del Niño Jesús Uribarren-Berrueta. ¿Cómo elaborar una rúbrica? *Investigación en educación médica*, 2(5), pp. 61-65, 2013.
- [10] Petra Heck y Gerald Schouten. Turning Software Engineers into AI Engineers. *arXiv preprint arXiv:2011.01590*, 2020.
- [11] Christian Kästner, Eunsuk Kang. Teaching Software Engineering for AI-Enabled Systems. En *IEEE/ACM 42nd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*, pp. 45-48, 2020.
- [12] Miryung Kim, Thomas Zimmermann, Robert DeLine, et al. Data scientists in software teams: State of the art and challenges. *IEEE Transactions on Software Engineering*, 44(11), 1024-1038, 2017.
- [13] Lukasz Kurgan y Petr Musilek. A survey of knowledge discovery and data mining process models. *Knowledge Engineering Review*, 21(1), pp. 1-24, 2006.
- [14] Valentina Lenarduzzi, Nyyti Saarimäki, y Davide Taibi. The technical debt dataset. En *15th International Conference on Predictive Models and Data Analytics in Software Engineering*, pp. 2-11, 2019.
- [15] P. Santhanam, Eitan Farchi, Victor Pankratius. Engineering reliable deep learning systems. *arXiv preprint arXiv:1910.12582*, 2019.
- [16] Silverio Martínez-Fernández, Justus Bogner, Xavier Franch, et al. Software Engineering for AI-Based Systems: A Survey, *arXiv:2105.01984*, 2021
- [17] Ezequiel Scott, Fredrik Milani, y Dietmar Pfahl. Data Science and Empirical Software Engineering. En *Contemporary Empirical Methods in Software Engineering*, pp. 217-233, 2020.
- [18] Stefan Studer, Thanh Bui, Christian Drescher, et al. Towards CRISP-ML (Q): a machine learning process model with quality assurance methodology. *arXiv preprint arXiv:2003.05155*, 2020.
- [19] Dongmei Zhang, Shi Han, Yingnong Dang, et al. Software analytics in practice. *IEEE software*, 30(5), pp. 30-37, 2013.
- [20] Miguel Valero, Javier García. Cómo empezar fácil con PBL. *Jornadas de Enseñanza Universitaria de la Informática* (17es: 2011: Sevilla), 2011.
- [21] Arie Van Deursen, Maurício Aniche, Joop Aué, et al. A collaborative approach to teaching software architecture. En *ACM SIGCSE Technical Symposium on Computer Science Education*, pp. 591-596, 2017.

Introducción de una herramienta OCL en la asignatura de Ingeniería de Requisitos

Loli Burgueño¹, Javier Luis Cánovas Izquierdo¹, Elena Planas¹, Jordi Cabot^{1,2}

¹Estudis d'Informàtica, Multimèdia i Telecomunicació, Universitat Oberta de Catalunya; ²ICREA
{lburguenoc, jcanovasi, eplanash}@uoc.edu, jordi.cabot@icrea.cat

Resumen

En muchas asignaturas de Ingeniería Informática se pide a los alumnos que programen en papel. Entre sus ventajas, esto permite a los alumnos centrarse en la funcionalidad evitando las distracciones causadas por la sintaxis y sin limitar su pensamiento a un lenguaje o paradigma de programación concreto. Aún así, los estudiantes se quejan de no poder probar su código, lo que les impide descubrir errores.

En contraposición al estado del arte que ha estudiado este problema para lenguajes de propósito general como Java con estudiantes de cursos iniciales, nosotros nos centramos en lenguajes estándares de la definición de reglas como OCL que se imparten en cursos avanzados.

En este artículo presentamos nuestra experiencia tras introducir una herramienta de modelado para la definición de restricciones OCL en una asignatura de Ingeniería de Requisitos. Esta asignatura es optativa y se ofrece en el tercer y cuarto curso del grado de Ingeniería Informática. El estudio que hemos realizado abarca dos cursos académicos, 2019 y 2020, en los cuales hubo 136 y 161 estudiantes matriculados respectivamente. Presentamos los detalles de nuestro experimento, los resultados obtenidos del estudio empírico llevado a cabo y nuestras conclusiones, las cuales respaldan la idoneidad del uso de herramientas.

Abstract

In many Computer Engineering courses, students are asked to program on paper. Among its advantages, this allows students to focus on functionality avoiding the distractions caused by syntax and without limiting their thinking to a specific programming language or paradigm. Still, students complain about not being able to test their code, preventing them from discovering bugs.

While the state of the art has studied this problem for general-purpose languages like Java and with students of initial courses, we focus on standard languages for constraint definition like OCL, which are taught in advanced courses.

In this article, we present our experience after introducing a modeling tool for the definition of OCL constraints in the Requirements Engineering course. This course is optional and is offered in the third and fourth years of the Computer Engineering degree. The study we have carried out covers two academic years, 2019 and 2020, in which there were 136 and 161 students enrolled, respectively. We present the details of our study, the results obtained from the empirical study carried out and our conclusions, which support the suitability of the use of tools.

Palabras clave

Ingeniería de requisitos, herramientas, OCL

1. Introducción

En los últimos años, los profesores responsables de la asignatura de Ingeniería de Requisitos en la Universitat Oberta de Catalunya hemos detectado que los estudiantes mostraban cierto descontento durante la realización de una práctica evaluable en la que les pedíamos que definieran en papel una serie de restricciones usando el lenguaje declarativo OCL [8]. Las peticiones de algunos estudiantes y la sospecha de que el desarrollo de este ejercicio en papel y sin ningún soporte digital pudiera estar ocasionando un problema de aprendizaje nos llevó a plantearnos la introducción de una herramienta de modelado con soporte para que permita la ejecución de dichas restricciones OCL.

El uso del papel en el ámbito de la programación tiene sus defensores y sus oponentes. Por una parte, muchos educadores respaldan sus beneficios para estudiantes de ingeniería informática o cualquier otra enseñanza técnica superior. Estos afirman que la programación en papel permite a los estudiantes centrarse en la lógica del programa que están escribiendo, evita las distracciones causadas por errores de sintaxis y no limita a los estudiantes a pensar en un lenguaje de programación o paradigma específico¹. Estas afirmaciones también

¹<https://classcube.com/write-code-paper/>

están respaldadas por muchas empresas del ámbito nacional e internacional, las cuales lo hacen explícito, por ejemplo, durante su proceso de contratación. Una de las varias pruebas que las empresas hacen a sus potenciales trabajadores suele ser un ejercicio en una pizarra, es decir, piden a sus entrevistados que resuelvan un problema en una pizarra usando el (pseudo-)lenguaje que ellos prefieran, y de este modo es cómo evalúan sus conocimientos y habilidades.

En las universidades, la programación con lápiz y papel se utiliza con frecuencia no solo como medio de aprendizaje sino también a la hora de evaluar los conocimientos de los estudiantes en ejercicios y prácticas evaluables y exámenes finales [1, 12, 13].

A pesar de sus beneficios y adopción, la programación en papel también tiene sus enemigos y algunos inconvenientes ampliamente reconocidos. Por ejemplo, los estudiantes suelen quejarse que no pueden ejecutar, probar y depurar el código que están escribiendo, lo que les impide verificar su corrección y detectar errores que podrían ser fácilmente descubiertos con una simple ejecución del programa.

Dado que no nos consta que haya ningún estudio que aborde el uso de papel en contraposición a una herramienta a la hora de aprender lenguajes formales y/o lenguajes estándares de la definición de reglas (como es el caso de OCL) y que sabemos que los lenguajes de modelado como UML/OCL son extensivamente usados, no solo en nuestra universidad sino en muchas otras², decidimos documentar y publicar nuestra experiencia.

Nuestro estudio muestra que el uso de herramientas para el aprendizaje de lenguajes de definición de reglas es percibido de forma positiva tanto por estudiantes como profesores. En comparación con los resultados del curso anterior, donde no se les ofreció a los estudiantes la posibilidad de usarla, las notas han mejorado.

El resto de este artículo está estructurado de la siguiente forma: el apartado 2 explica el contexto de la asignatura donde se ha realizado el experimento, el apartado 3 describe el experimento y la metodología que hemos seguido, el apartado 4 presenta los resultados obtenidos, en el apartado 5 presentamos nuestra discusión sobre los resultados, el apartado 6 el trabajo relacionado y, por último, el apartado 7 presenta brevemente nuestras conclusiones y líneas de trabajo futuro.

2. Contexto

El experimento que describe este artículo se contextualiza en la asignatura Ingeniería de requisitos del Grado en Ingeniería Informática de la Universitat Oberta de Catalunya (UOC). Se trata de una asignatura optativa que se imparte una vez al año y que los estudiantes pueden cursar a partir del tercer año. La asignatura parte de

un conocimiento básico sobre la ingeniería del software y profundiza en la primera etapa del ciclo de vida del desarrollo del software. Los contenidos de la asignatura se organizan en módulos docentes: (1) introducción a la ingeniería de requisitos; (2) obtención de requisitos; (3) gestión de requisitos; (4) documentación de requisitos; y (5) validación y verificación de requisitos. El lenguaje OCL se introduce en el módulo 4, como herramienta para documentar formalmente los requisitos.

El modelo de evaluación de la asignatura se basa en la evaluación continua, y está compuesta por cuatro pruebas (PEC) de carácter obligatorio. Todas las PEC trabajan un mismo caso de estudio a lo largo del semestre, lo que permite a los estudiantes tener una visión completa de todas las fases de la ingeniería de requisitos. En concreto: la PEC1 se centra en la obtención de los requisitos; la PEC2 trabaja la gestión de los requisitos de forma colaborativa en grupos de 4 o 5 personas; durante la PEC3 los estudiantes documentan los requisitos de forma ágil mediante casos de uso e historias de usuario; y, finalmente, en la PEC4, que es la actividad objeto de nuestro experimento, los estudiantes documentan los requisitos de modo formal (mediante UML/OCL) y aplican técnicas de validación y verificación. A modo de ejemplo, la PEC4 del semestre de 2020 consistía en documentar formalmente y validar los requisitos de una red social (ver Pregunta 14 en [6]). En concreto, dado un diagrama de clases, se pedía a documentar cuatro restricciones de integridad mediante OCL.

La evaluación de la asignatura es formativa, dado que se proporciona un retorno a los estudiantes de las diferentes PEC mediante una calificación acompañada de comentarios individuales, así como la publicación de la solución de cada una de ellas.

La asignatura cuenta con dos profesores responsables (los cuales son autores de este artículo) y tiene un alrededor de unos 150 estudiantes matriculados cada semestre de impartición, que se distribuyen en aulas virtuales de unos de 70 estudiantes cada una. Cada aula es dinamizada por un profesor colaborador, que guía y acompaña a los estudiantes durante su aprendizaje y finalmente los evalúa y les retorna el *feedback* de cada una de las actividades.

A lo largo del semestre, toda la comunicación se realiza de forma asíncrona y *online*, a través de los foros del aula virtual (donde los mensajes son públicos para todos los estudiantes del aula) y a través del correo electrónico (generalmente utilizado para resolver dudas directamente y de forma privada con los profesores).

3. Descripción del experimento

En este trabajo hemos construido un experimento con el objetivo de responder la siguiente pregunta de investigación: ¿Qué impacto tiene en el aprendizaje de los estudiantes el uso de una herramienta de modelado

²<https://mde-network.github.io/#miembros>

con soporte para la definición y ejecución de restricciones OCL? Proponemos medir el aprendizaje mediante tres medidas indirectas que nos permiten llevar a cabo una validación cruzada de los resultados: la autopercepción del estudiante, la percepción del profesorado y el rendimiento académico. Nuestra hipótesis es que el uso de la herramienta tiene un impacto positivo en la comprensión y aprendizaje del lenguaje OCL ya que la experimentación durante el proceso de aprendizaje les permite conocer la sintaxis y semántica del lenguaje, detectar errores tempranos (que pueden corregir y de los cuales aprender) y, por tanto, definir restricciones correctas con un mayor grado de éxito.

Dado nuestro contexto académico y el hecho de que estamos ante un caso de investigación-acción (o *Action Research* en inglés), para responder a la pregunta de investigación hemos hecho uso de los métodos empíricos más apropiados para abordar nuestra problemática [5].

Hemos construido un experimento controlado para estudiar la relación causa-efecto entre el uso de la herramienta de modelado (variable binaria independiente) y sus consecuencias. Hemos identificado un número de variables dependientes, entre ellas: tiempo dedicado, percepción de utilidad y dificultad de la herramienta, y rendimiento académico. A continuación presentaremos dichas variables más detalladamente.

El experimento se desarrolla en la cuarta PEC de la asignatura, que contiene tres ejercicios, donde uno de ellos evalúa los conocimientos sobre OCL y tiene un peso del 40 %. En particular, el ejercicio de OCL presenta un diagrama de clases y contiene cuatro sub-apartados donde el estudiante debe definir restricciones OCL. En el semestre de desarrollo del experimento hemos continuado con la misma estructura para la PEC que en años anteriores y, para facilitar la comparación del impacto de utilizar una herramienta de modelado, las restricciones que deben definirse en los sub-apartados son similares a las requeridas en el mismo ejercicio de la PEC del semestre anterior (i.e., evalúan las mismas funcionalidades de OCL).

Antes de comenzar la actividad, los estudiantes son informados del experimento y se les ofrece la opción de utilizar una herramienta de modelado para resolver los ejercicios. La decisión final de utilizar dicha herramienta recae en el estudiante. La herramienta de modelado seleccionada fue MAGICDRAW, ya que es una de las opciones más populares para modelado en UML, soporta la definición y ejecución de OCL, y se utiliza en la asignatura de Ingeniería del Software, la cual es pre-requisito de nuestra asignatura.

Los estudiantes disponen de cuatro semanas para trabajar en la PEC y entregar su solución. Durante este tiempo, los estudiantes pueden publicar sus dudas en el foro del aula virtual o contactar directamente con el profesor colaborador de la asignatura vía email. Al

finalizar este periodo de tiempo, los autores de este artículo recopilamos los datos del experimento a partir de tres fuentes diferentes. Estas son: (1) los estudiantes, mediante el uso de un cuestionario online; (2) los profesores colaboradores, mediante entrevistas online; y (3) los medios de comunicación y calificaciones obtenidas. A continuación describimos en detalle cada parte.

3.1. Cuestionarios

En nuestro experimento recopilamos información del alumnado mediante cuestionarios anónimos online. La estructura y el contenido de los cuestionarios se definieron de forma iterativa por los autores de este trabajo. Una vez alcanzada la versión final, se validó con los profesores colaboradores.

Cada cuestionario tiene una primera sección con preguntas generales que nos permiten caracterizar al estudiante que participa en el experimento y conocer su conocimiento del lenguaje OCL. Para caracterizar al estudiante se incluyen preguntas demográficas típicas así como la nota del estudiante en la asignatura de Ingeniería del Software. Las preguntas sobre OCL están enfocadas a saber si el estudiante lo conoce, la percepción que tiene (i.e., si es un lenguaje fácil o difícil) y si ve necesario el uso de herramientas para practicarlo.

A continuación, la segunda sección del cuestionario contiene una serie de preguntas específicas de OCL que dependen de si el estudiante ha decidido utilizar MAGICDRAW o no. Para estudiantes que sí han utilizado la herramienta, el cuestionario incluye preguntas para conocer el motivo de dicha decisión, la experiencia del estudiante, el tiempo dedicado a resolver los ejercicios y la percepción de dificultad de cada sub-apartado de los ejercicios. Adicionalmente, se incluyen preguntas para conocer si MAGICDRAW ha permitido a los estudiantes experimentar más allá de los ejercicios, permitiendo explorar el lenguaje y resolver dudas de forma autónoma.

Si el estudiante no ha utilizado la herramienta de modelado, el cuestionario solamente incluye preguntas para conocer el motivo de dicha decisión, el tiempo dedicado a resolver los ejercicios y la percepción de dificultad para cada sub-apartado de los ejercicios.

El formulario completo puede encontrarse en [6].

3.2. Entrevistas estructuradas al profesorado colaborador

El proceso de recogida de comentarios del profesorado colaborador se planteó a través de entrevistas estructuradas. Estas entrevistas se organizaron para cubrir cuatro bloques: (1) impresión general, para conocer la percepción y opinión de los profesores sobre los efectos del uso de MAGICDRAW a la hora de resolver la PEC; (2) su opinión sobre si darle o no continuidad al uso de la herramienta y a la recomendación al alumnado de su

uso en cursos posteriores; (3) el grado de interacción con los estudiantes, para saber si el hecho de introducir la herramienta de modelado afecta esta dimensión (p.ej., conocer si tuvieron que responder muchas preguntas técnicas sobre el uso de MAGICDRAW no relacionadas con el lenguaje OCL); y (4) conclusión general y comentarios adicionales, para identificar, mejorar y proponer cambios en futuros semestres.

3.3. Medios de comunicación y rendimiento académico

Para complementar la evaluación de nuestro experimento, recopilamos el número de mensajes publicados en el foro y los emails referidos a los ejercicios de OCL así como las calificaciones de los estudiantes. De forma adicional, y con la intención de hacer un estudio comparativo, también recopilamos estos datos para el semestre anterior.

4. Evaluación empírica

4.1. Recolección de datos

Tal y como se comentó en la sección anterior, la recogida de datos se produjo al finalizar la PEC4 y los datos provienen de tres fuentes diferentes, para las cuales indicamos el momento de ejecución: (1) el cuestionario para los estudiantes se realizó durante la primera semana de enero de 2021; (2) las entrevistas al profesorado colaborador se realizaron a finales de enero de 2021 después de haber revisado y evaluado la PEC; y (3) los datos del registro de mensajes intercambiado en los foros del aula y las notas obtenidas en el curso 2020/2021 se recogieron después de las acciones anteriores.

4.2. Resultados

Encuesta a los estudiantes

En la encuesta hubo una participación del 32 % (52 de 161 estudiantes la respondieron). De ellos, 46 se identificaron con el género masculino, 6 con el femenino y no hubo estudiantes que se identificaran con otro. Un estudiante se encuentra en la franja de edad entre los 18-23 años, 41 estudiantes entre los 24-45 años y 10 estudiantes son mayores de 45 años.

En primer lugar, reportamos los resultados de la sección de preguntas generales. A la pregunta de si conocían OCL con anterioridad a este curso, el 75 % contestaron que no, el 19 % dijeron que habían oído hablar de él y solo el 6 % (3 estudiantes) dijeron que sí. Cabe destacar que para estos 3 estudiantes, esta era su primera matrícula en la asignatura, por lo que su uso ha debido producirse en otro contexto ajeno a nuestra asignatura. Cuando les preguntamos por su percepción general sobre OCL, en particular, sobre el nivel de dificultad de

	Núm. estud.	Moda	Mediana	Media	Std.
MH	2	-	0	0,50	0,71
Sobres.	3	0	0	0,33	0,57
Notable	27	0	0	0	0,88
Aprob.	7	0	0	-0,57	0,79
Conval.	13	0 y 1	0	-0,38	0,87
Total	52	0	0	-0,13	0,86

Cuadro 1: Percepción de la dificultad de OCL usando la escala de acuerdo de Likert [-2=muy complejo..+2=muy fácil] en función de la calificación obtenida en ingeniería del software.

OCL usando la escala de acuerdo de Likert de 5 valores que va desde muy complejo (-2) a muy fácil (+2), tanto la moda como la mediana de sus respuestas fue de 0.

A pesar del uso de la escala cuantitativa de Likert, la cual normalmente se traduce a valores enteros entre 1 y 5 a partir de los cuales no se suelen calcular medias, nosotros hemos optado por migrarla esta escala a una escala decimal en el rango [-2..2], y hemos calculado la media y desviación típica³.

En la encuesta preguntamos a los alumnos por la nota que obtuvieron en la asignatura de Ingeniería del Software (la cual es pre-requisito). A continuación, usando la misma escala anterior, evaluamos la opinión sobre OCL condicionada a la nota que dichos estudiantes obtuvieron en Ingeniería del Software y obtuvimos los resultados que mostramos en el Cuadro 1. Podemos observar una relación directa entre las notas de Ingeniería del Software y la percepción de los estudiantes: los estudiantes con notas más altas tienden a pensar que OCL es más fácil, mientras que los estudiantes con notas más bajas lo encontraron más complejo.

Cuando les preguntamos a los estudiantes si consideraban que el aprendizaje de OCL necesita de herramientas para poder hacer ensayo-error usando la escala de acuerdo de Likert, obtuvimos una moda de 2, una mediana de 1 y una media de 1,04 con una desviación típica de 1,03, lo cual indica que tienden a pensar que el uso de herramientas es beneficioso o incluso necesario.

El Cuadro 2 muestra los resultados estadísticos a esta misma pregunta considerando de forma independiente al grupo de 35 estudiantes que no usó MAGICDRAW (grupo de control) y al grupo de 52 estudiantes que sí lo usó (grupo experimental). La media de las opiniones muestra como incluso aquellos estudiantes que no usaron MAGICDRAW, tienden a pensar que su uso es más necesario que innecesario.

Respecto al tiempo dedicado a realizar el ejercicio de OCL, los estudiantes que usaron MAGICDRAW dijeron haber tardado una media de $6,11 \pm 4,61$ horas

³La UOC usa esta estrategia para obtener datos más precisos e informativos al analizar las encuestas de satisfacción del alumnado.

	Num. estud.	Moda	Media	Media	Std.
Sí MD	35	2	2	1,34	0,84
No MD	17	0	0	0,41	1,12
Total	52	2	1	1,04	1,03

Cuadro 2: Respuestas a la pregunta: “Considero que el aprendizaje de OCL necesita herramientas para poder hacer ensayo-error” usando la escala de acuerdo de Likert en función de si usaron MAGICDRAW (MD).

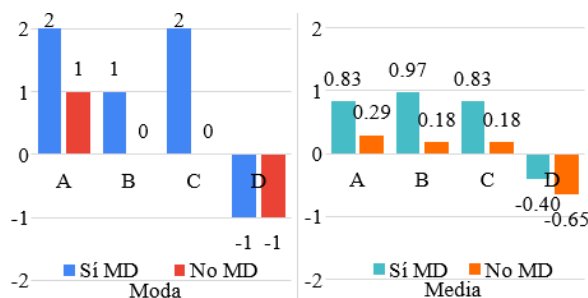


Figura 1: Percepción de los estudiantes sobre la dificultad del ejercicio de OCL.

mientras que los estudiantes que no lo usaron tardaron $5,56 \pm 4,84$ horas. Para comparar estadísticamente si hay diferencia en el tiempo que tardaron ambos grupos, hemos hecho la prueba t de Welch de dos colas y varianzas desiguales. Nuestra hipótesis nula es que los tiempos de los alumnos de ambos grupos no difieren. La prueba arroja un valor de 0,71, que al ser mayor que $\alpha = 0,05$, nos hace aceptar la hipótesis nula y concluir que no hay diferencias temporales significativas.

A continuación mostramos los resultados a las preguntas específicas a estudiantes que no usaron MAGICDRAW. Les preguntamos cuál había sido la razón para optar por no usar MAGICDRAW y 5 de ellos dijeron que “consideraban saber escribir restricciones y no necesitaban usar una herramienta”, 6 de ellos que “no consiguieron hacer MAGICDRAW funcionar”, 4 dijeron no haberlo usado “por no querer perder el tiempo usando una herramienta” y dos personas por “falta de tiempo”. Cabe destacar que varios estudiantes se quejaron de la incompatibilidad de MAGICDRAW con la nueva actualización de macOS. Curiosamente, de los 5 estudiantes que consideraban saber escribir OCL sin probarlo, 3 habían usado MAGICDRAW anteriormente, 2 pensaron usarlo pero luego descartaron la idea y uno de ellos dijo que la herramienta es difícil de usar y que no consideró que tuviera un beneficio sustancial.

Respecto a la percepción de dificultad para resolver cada uno de los subapartados del ejercicio de OCL (A-D), preguntamos a los estudiantes por el grado de dificultad mediante la pregunta: “Me ha resultado fácil resolver el apartado...”, y de nuevo usamos la escala de acuerdo de Likert y presentamos los resultados traslada-

dos al rango [-2..2]. La Figura 1 muestra a la izquierda en azul la moda de las respuestas que dieron los estudiantes que usaron MAGICDRAW y en rojo los que no lo usaron; y a la derecha, en azul turquesa la media de las respuestas que dieron los estudiantes que usaron MAGICDRAW y en naranja los que no lo usaron. Se puede observar cómo en general todos los estudiantes valoraron que los ejercicios A, B y C eran más fáciles que el ejercicio D. No obstante, se aprecia que quienes usaron MAGICDRAW consideraron que los ejercicios eran más fáciles.

Por último, presentamos los resultados a las preguntas específicas que hicimos a estudiantes que usaron MAGICDRAW. Usando nuevamente la escala de Likert, preguntamos a los estudiantes si:

- MAGICDRAW les había ayudado a resolver el ejercicio más rápidamente, para lo cual obtuvimos una respuesta de moda y media 0, lo cual indica que los estudiantes no consideraban que MAGICDRAW hubiera tenido ningún impacto temporal;
- MAGICDRAW les había permitido resolver el ejercicio más fácilmente, para lo cual obtuvimos una moda de 1 y una media de 0,54, que indica que los estudiantes tienden a estar de acuerdo;
- si MAGICDRAW les permite aprender OCL de manera práctica y plantear ejercicios más realistas y complejos, para la cual obtuvimos una moda de 0 y media de 0,63, lo cual indica que los estudiantes tienden a estar de acuerdo con esta afirmación;
- si recomendarían usar MAGICDRAW en cursos posteriores, a lo cual los estudiantes respondieron con una moda de 1 y una media de 0,86.

También preguntamos a los estudiantes si:

- consideraban que uso de MAGICDRAW les había permitido experimentar con restricciones OCL más allá de las que se pedían en los ejercicios; y 21 estudiantes (60 %) nos dijo que sí, mientras que 14 (40 %) dijeron que no;
- el uso de MAGICDRAW les había permitido resolver dudas de OCL de manera autónoma, sin tener que recurrir al profesor o a los foros del aula, y 23 estudiantes (66 %) nos dijeron que sí, 9 (26 %) dijeron que no, y un solo estudiante nos dijo que no le habían surgido dudas.

Con el objetivo de proporcionar claridad y permitir la reproducibilidad de nuestro análisis, tanto las preguntas del cuestionario como las respuestas de los estudiantes pueden ser consultadas en [6].

Entrevistas a los profesores colaboradores

Los tres profesores colaboradores, y especialmente dos de ellos que impartieron docencia en ambos cursos, coincidieron en que el cambio ha sido positivo y que

Curso	Núm. estud.	Msj. en foro	Msj. por email	Msj/ Estud.
2019	136	35	20	0,40
2020	161	61	70	0,81

Cuadro 3: Interacción profesorado-estudiantes

deberíamos seguir dando a los estudiantes la opción de usar una herramienta. También dijeron que no había habido muchas preguntas relacionadas con el uso de MAGICDRAW, más allá de algún problema de instalación en macOS BigSur.

Cuando les preguntamos si les había supuesto una mayor carga de trabajo, su respuesta fue negativa.

También nos dijeron que tenían la sensación de que las preguntas de los estudiantes este curso eran más concretas y que parecía que tenían más idea de cómo abordar los ejercicios, en contraposición a otros años en los que los estudiantes estaban más desorientados.

Como comentario adicional, uno de los profesores nos dijo que había notado que la calidad de las soluciones había aumentado y, sobre todo, le había llamado la atención que los estudiantes habían hecho un uso más correcto de la sintaxis del lenguaje. Apuntó a que en años anteriores algunos estudiantes tenían tendencia a escribir consultas con una sintaxis más propia de SQL que de OCL, mientras que este año ha encontrado que esto no ocurría con la misma frecuencia.

Comparativa curso 2019 y 2020

El Cuadro 3 recoge los resultados sobre la interacción entre los profesores y los estudiantes en foros y por email. En la última columna podemos ver como, para esta PEC, el ratio de mensajes por estudiante en 2019 era de 0,40 mientras que en 2020 incrementó a 0,81. Esto refleja que en 2020 hubo una mayor actividad en los foros del aula y más preguntas a los profesores.

El Cuadro 4 presenta los datos que nos permiten hacer una comparativa entre las calificaciones de los cursos 2019 y 2020. En ella hemos registrado el número de estudiantes, el número de esos estudiantes que no entregaron la PEC (columna NP) y la nota media y la desviación típica de la nota de la PEC4. Para probar si, estadísticamente, la nota en ambos cursos ha sido significativamente diferente, hemos realizado la prueba *t* de Welch con dos colas e iguales varianzas, siendo la hipótesis nula que no hay diferencia en las notas. La prueba arroja un valor de 0,0065, el cual al ser menor que $\alpha = 0,05$, nos hace rechazar la hipótesis nula y afirmar que las notas en ambos cursos han sido significativamente diferentes. Además, como puede darse el caso de que, por diversas razones y a pesar de haber usado PECs similares ambos años, haya diferencias en las notas, de cara a hacer una comparativa lo más justa posible, hemos recopilado la nota media final de la asignatura en ambos cursos y hemos normalizado la nota de la PEC con respecto a ella. Así podemos concluir que mientras que la nota normalizada en el curso 2019 fue un 6,39; en el curso 2020 ha sido de 6,70, es decir, una media de 0,31 puntos más alta.

Curso	Núm. estud.	NP	Nota PEC	Nota final	Nota PEC Norm.
2019	136	6	$8,08 \pm 1,14$	7,91	6,39
2020	161	9	$8,46 \pm 1,15$	7,92	6,70

Cuadro 4: Notas de la PEC en los cursos 2019 y 2020

4.3. Amenazas a la validez

Por lo que respecta a la validez interna, los métodos de evaluación empíricos no son infalibles y diversos factores pueden influir en su validez, como por ejemplo, ignorar alguna variable que pueda tener un impacto en los resultados obtenidos y por tanto en las conclusiones. Es por ello que hemos intentado usar más de un método (cuestionario a los estudiantes, entrevistas estructuradas a los profesores y una comparativa cuantitativa de resultados objetivos entre ambos cursos) para intentar responder a la misma pregunta.

El cuestionario que proporcionamos a los estudiantes fue anónimo. Esto nos impidió hacer un análisis más profundo como por ejemplo relacionar el uso (o no uso) de MAGICDRAW con la nota final de los estudiantes. No obstante, creemos que nuestra decisión fue acertada ya que el anonimato mejora la honestidad de las respuestas.

Preguntar a los alumnos sobre si aconsejan el uso de MAGICDRAW tras haberlo usado o no puede propiciar un resultado sesgado. La validación cruzada mediante el uso de diferentes métodos empíricos contribuye a mitigar estos sesgos.

De los tres profesores colaboradores, dos de ellos formaron parte de la asignatura en ambos cursos y solo uno se incorporó en el año 2020. Esto garantiza que al menos estos dos profesores han evaluado e interactuado con los estudiantes de la misma forma. Además, están en posición de expresar su opinión personal sobre el desarrollo de ambos cursos.

Para poder hacer la comparativa entre los cursos 2019 y 2020 lo más justa posible, la PEC4 en 2020, aunque describiendo un caso de uso diferente, intentaba imitar fielmente la estructura y contenido de la PEC4 de 2019. Cabe decir que las PECs de años anteriores no se publican y, en teoría, los estudiantes no tienen acceso a ellas, con lo cual la similitud entre ambas no debería haber alterado las notas. Además, a la hora de comparar las notas de los cursos 2019 y 2020, hemos calculado la nota de la PEC normalizada para mitigar el impacto de la comparación PECs y estudiantes diferentes.

Por otro lado, la validez externa se puede ver amenazada ya que el experimento se desarrolla en un entorno

real donde los estudiantes deben tener la opción de elegir si usar MAGICDRAW o no, no pudimos hacer una asignación equitativa y aleatoria. Esto dio lugar a grupos no balanceados lo que puede afectar a los resultados estadísticos. No obstante, dado que el número de estudiantes en cada grupo es significativo, creemos que los resultados pueden ser generalizables a otros cursos y asignaturas donde se aborde la definición de restricciones OCL.

5. Discusión

En primer lugar, hemos observado que los estudiantes apoyan el uso de herramientas de modelado. Los resultados de nuestro experimento muestran que la percepción de dificultad de los estudiantes es significativamente menor cuando estos usan una herramienta de modelado. Los estudiantes que utilizaron MAGICDRAW reconocieron que la herramienta les permitió ser más autónomos y experimentar no solo con los ejercicios planteados en la PEC sino con más. Además, estos recomiendan el uso de MAGICDRAW en cursos posteriores.

El estudio del rendimiento académico de los estudiantes ha sido ligeramente más elevado. Aunque la encuesta a los estudiantes fue anónima y no podemos relacionar las respuestas con las notas de los estudiantes, la calificación final media ha sido ligeramente superior. Aunque esto apunta a un resultado satisfactorio, este aspecto requiere un estudio más profundo en los próximos cursos ya que, al no ser una diferencia excesivamente destacable, ésta podría estar ligada también a otros factores que no hayamos considerado en nuestro estudio.

El experimento ha mostrado que no existe una barrera de entrada técnica relevante. No hemos apreciado problemas significativos que dificulten la instalación y uso de MAGICDRAW más allá de los problemas de instalación en la última actualización del sistema operativo macOS. Cabe decir que MAGICDRAW no había tenido problemas en macOS con anterioridad, y que éste no ha sido el único software que falló tras la actualización a BigSur. Por lo tanto, lo consideramos más bien una desafortunada coincidencia que finalmente pudo ser resuelta mediante el uso de una máquina virtual.

Las entrevistas con los profesores colaboradores indican que estos apoyan la incorporación de herramientas de modelado. Ninguno apuntó a ningún inconveniente que las desaconsejara y todos estuvieron de acuerdo en que dar la opción a usar MAGICDRAW fue un cambio positivo, y recomiendan su uso de cara al futuro.

La introducción de herramientas de modelado no ha significado un aumento de dedicación del profesorado. Curiosamente, a pesar de que el número de mensajes intercambiados entre estudiantes y profesores durante el curso 2020 aumentó con respecto a 2019, en la entrevista a los profesores colaboradores, estos nos transmitieron que no habían notado una mayor carga de trabajo.

Esto probablemente es debido a que las preguntas de los estudiantes fueron más concretas y rápidas de resolver.

El uso de MAGICDRAW ha sido un complemento a las actividades y no ha implicado un rediseño de la evaluación de la asignatura. No obstante, si seguimos confirmando las ventajas del uso de herramientas, podremos plantearnos incorporar en el plan docente de la asignatura objetivos de aprendizaje más ambiciosos.

La introducción de herramientas supone a los profesores responsables el esfuerzo de hacer un estudio de todas aquellas que podrían ser usadas y, una vez se ha optado por una (o varias), la creación de manuales y actualización de los mismos año tras año. Esto es especialmente importante en el caso de MAGICDRAW ya que dependiendo de la versión de la herramienta y del tipo de licencia la funcionalidad es diferente.

6. Trabajo relacionado

Hay numerosos estudios en la literatura sobre el uso de herramientas de modelado. Muchos de estos estudios se basan en experimentos que comparan diversas características de las herramientas de desarrollo con el fin de determinar su usabilidad [11, ?]. Si nos centramos en OCL, Burgueño *et al.* [3] describen los principales problemas que se han encontrado al enseñar modelado en un curso dedicado a la Ingeniería del software, y presentan un caso de estudio con UML/OCL que ha sido utilizado con éxito durante el curso. No obstante, en este curso todos los estudiantes usaron la herramienta USE tanto para el modelado como para la definición de restricciones OCL y no se valoró las ventajas/desventajas de usar una herramienta frente a no usarla. Por otro lado, Maraee *et al.* [7] comparan la eficacia de desarrollar restricciones usando dos enfoques diferentes: el lenguaje declarativo OCL mediante la herramienta USE, y el lenguaje imperativo Java mediante un IDE de Java.

En menor medida, también existen estudios que comparan el uso de herramientas versus la escritura con lápiz y papel y su impacto en el aprendizaje de los estudiantes. Buena parte de estos estudios, que se centran principalmente en el aprendizaje de lenguajes de programación, determinan las ventajas de la evaluación electrónica por delante la evaluación tradicional (en papel u oral). Bessedsen *et al.* [2] reportan un experimento con exámenes de laboratorio en una asignatura introductoria de programación con más de 500 estudiantes. El resultado fue satisfactorio en todos los aspectos: simplicidad, eficacia, rendimiento y satisfacción de los participantes (docentes y estudiantes). Por otro lado, Rytkönen *et al.* [10] describen su experiencia enseñando programación en C. Sus conclusiones determinan que los estudiantes perciben los exámenes de programación en formato electrónico como más realistas y naturales en comparación con exámenes en papel. En esta misma línea, Bottcher *et al.* [4] informan sobre su

transición del papel a los exámenes electrónicos para un curso de introducción a la programación con Java.

Finalmente, un estudio similar a nuestro trabajo es el publicado por Öqvist et al. [9], donde los autores describen su experimento para evaluar los efectos en el rendimiento de los estudiantes comparando la codificación a mano frente a herramientas de programación. En este caso, los resultados del experimento, en el que los sujetos eran estudiantes principiantes y se usó Java, no fueron concluyentes. Lamentablemente, no tenemos conocimiento de estudios similares centrados en el uso de herramientas de modelado frente a dibujos manuales para lenguajes gráficos como UML o lenguajes de definición de reglas como OCL.

7. Conclusiones

Este artículo presenta un estudio donde se compara el uso de una herramienta de modelado para definir restricciones OCL en contraposición a la definición de las mismas en papel. El estudio usa diferentes métodos empíricos (p.ej., entrevistas y cuestionarios) e involucra a diferentes sujetos (estudiantes y profesores colaboradores) para hacer una validación cruzada y así confirmar la validez de los resultados. El estudio concluye que los estudiantes tienen una percepción positiva hacia el uso de herramientas para definir restricciones OCL, y que los profesores colaboradores también aprueban y aconsejan su uso en cursos posteriores. Además, los resultados parecen indicar que el uso de MAGICDRAW tiene cierto impacto en la nota final de los estudiantes, siendo ligeramente más alta cuando estos lo usan.

Este estudio es un primer intento hacia mejorar el aprendizaje de lenguajes estándares de definición de reglas por parte de los estudiantes universitarios así como su satisfacción con el proceso de aprendizaje. Después de este experimento, hemos optado por mantener el uso opcional de MAGICDRAW en nuestra asignatura y seguiremos analizando la evolución del rendimiento y la satisfacción de los estudiantes. También estudiaremos qué otros factores (como por ejemplo los materiales proporcionados u otras herramientas de modelado) pueden ayudar a mejorar el rendimiento y la satisfacción de los estudiantes, así como soluciones para poder relacionar los cuestionarios con la nota obtenida por el estudiante en la actividad asegurando su anonimato. A largo plazo, el éxito en el uso de estas herramientas nos permitirá a los profesores plantearnos objetivos de aprendizaje más ambiciosos tanto en esta asignatura como su propagación a otras asignaturas de ingeniería del software.

Referencias

- [1] Jony Adkins y Diana Linville: *Testing Frequency in an Introductory Computer Programming Course*. Information Systems Education Journal, 15:22–28, 2017.
- [2] Jens Bennedsen y Michael E. Caspersen: *Assessing Process and Product – A Practical Lab Exam for an Introductory Programming Course*. En *Proc. of Frontiers in Education (FIE)*, 2007.
- [3] Loli Burgueño, Antonio Vallecillo y Martin Gollola: *Teaching UML and OCL models and their validation to software engineering students: an experience report*. Computer Science Education, 28(1):23–41, 2018.
- [4] Alex Böttcher, Veronika Thurner y Daniela Zehetmeier: *Alignment of Teaching and Electronic Exams and Empirical Classification of Errors for an Introductory Programming Class*. En *Proc. of CSEET'20*, páginas 1–10, 2020.
- [5] Steve Easterbrook, Janice Singer, Margaret-Anne D. Storey y Daniela E. Damian: *Selecting Empirical Methods for Software Engineering Research*. En *Guide to Advanced Empirical Software Engineering*, páginas 285–311. 2008.
- [6] Loli Burgueño, Javier Luiz Cánovas Izquierdo, Elena Planas, and Jordi Cabot: *Materiales de MagicDraw/OCL (curso 2020)*. <http://hdl.handle.net/20.500.12004/1/C/JENUI/2021/001>.
- [7] Azzam Maraee, Eliran Nachmani y Arnon Sturm: *Constraints Specification Via Tool Support: A Controlled Experiment*. J. Object Technol., 19(3):3:1–18, 2020.
- [8] Object Management Group: *Object Constraint Language (OCL) Specification. Version 2.4*, 2014.
- [9] Martina Öqvist y J. Nouri: *Coding by hand or on the computer? Evaluating the effect of assessment mode on performance of students learning programming*. Journal of Computers in Education, 5:199–219, 2018.
- [10] Anni Rytönen y Venla Virtakoivu: *Comparative Student Experiences on Electronic Examining in a Programming Course - Case C*. En *Proc. of Koli Calling*, 2019, ISBN 9781450377157.
- [11] Safdar Aqeel Safdar, Muhammad Zohaib Iqbal y Muhammad Uzair Khan: *Empirical Evaluation of UML Modeling Tools-A Controlled Experiment*. En *Proc. of ECMFA'15*, páginas 33–44, 2015.
- [12] Michael de Raadt Anne Philpott Judy Sheard Mikko Jussi Laakso Daryl D'Souza James Skene Angela Carbone Tony Clear Raymond Lister Geoff Warburton Simon, Donald Chinn: *Introductory Programming: Examining the Exams*. En *Proc. of ACE'12*, volumen 123, 2012.
- [13] Arto Vihavainen, Matti Paksula y Matti Luukkainen: *Extreme Apprenticeship Method in Teaching Programming for Beginners*. En *Proc. of SIGCSE '11*, página 93–98, 2011, ISBN 9781450305006.

Modelos de inteligencia artificial para asesorar el proceso evaluador de trabajos informáticos complejos

Jose Divasón, F. Javier Martínez de Pisón, Ana Romero, Eduardo Sáenz de Cabezón

¹Departamento de Matemáticas y Computación; ²Escuela Técnica Superior de Ingeniería Industrial

Universidad de La Rioja

{jose.divason, fjmartin, ana.romero, eduardo.saenz-de-cabezon}@unirioja.es

Resumen

Planteamos el uso de modelos de inteligencia artificial (IA) para asesorar el proceso evaluador de trabajos informáticos complejos, con componentes técnica y creativa. El objetivo es generar un procedimiento para analizar las variables principales que describen la evaluación realizada, descubrir posibles sesgos y discrepancias y generar rúbricas adecuadas que los eviten. La metodología propuesta se ha aplicado a una asignatura de introducción a la informática (grados de informática y matemáticas, primer curso) en que una tarea consiste en la elaboración de una página web por parte de los estudiantes. Dicho trabajo debe cumplir unos requisitos técnicos (compatibilidad con estándares, número de documentos HTML y CSS, etc.) y tiene una componente creativa (maquetación, aspecto, etc.). Se han desarrollado modelos de IA optimizados mediante algoritmos evolutivos para identificar las variables que intervinieron en la calificación de dichos trabajos durante cinco cursos. Los resultados obtenidos permiten extraer conclusiones sobre la práctica evaluadora, posibles mejoras para la objetividad de la evaluación y la posibilidad de generar rúbricas adecuadas para la evaluación de este tipo de trabajos. La metodología es aplicable a otras materias al ser esta tipología técnica-creativa frecuente en trabajos universitarios.

Abstract

We propose the use of artificial intelligence (AI) models to help evaluate complex projects in computer science courses that involve technical and creative components. The goal is to provide a methodology to analyze the main variables that describe the evaluation, to discover possible biases and discrepancies, and to generate appropriate rubrics that avoid them. The proposed methodology has been applied to a first-year introductory course on computer science, which is taught in both computer science and mathematics degrees. In that course, the students must develop a web page ac-

cording to some technical requirements (compatibility with standards, number of HTML and CSS files, etc.), but it also possesses a creative component (layout, appearance, etc.). Optimized AI models have been developed using evolutionary algorithms to identify the most important variables that took part in the evaluation of these projects during five courses. The results obtained allow us to draw conclusions about the evaluation process, possible improvements in the impartiality of the evaluation, and the possibility of generating suitable rubrics for grading assignments. The methodology is applicable to projects in other subjects since this technical-creative typology is frequent in university tasks.

Palabras clave

Evaluación, inteligencia artificial, algoritmos evolutivos, sesgos de evaluación.

1. Introducción

En muchas asignaturas informáticas y técnicas de los diferentes estudios de grado y máster una parte de la evaluación se basa en la elaboración por parte de los estudiantes de proyectos y trabajos. El auge de esta metodología de enseñanza y evaluación, en particular del aprendizaje basado en proyectos (PBL), es debido sobre todo a su eficiencia en el desarrollo en los estudiantes de habilidades profesionales y competencias transferibles [6]. Sin embargo, el empleo de este tipo de trabajos y proyectos plantea diversas dificultades que tienen ciertas particularidades en los cursos fundamentales de informática [12]. Entre las dificultades y retos encontrados en relación a los métodos basados en trabajos y proyectos están las relacionadas con su evaluación [5, 13], que resulta en ocasiones difícil por tener los trabajos un doble carácter técnico y creativo y por resultar compleja la evaluación de competencias transferibles y habilidades adquiridas en el desarrollo del

proyecto [14]. Otro factor complicado de acometer es la posible presencia de sesgos, discrepancias e inconsistencias en la evaluación, que resultan muy difíciles de detectar para los propios docentes implicados [24]. La detección de discrepancias, inconsistencias y sesgos en la evaluación se lleva a cabo en la literatura de dos formas: exógena (basada en medidas externas del desempeño de los estudiantes) y endógena (basada en las propias calificaciones) habiéndose mostrado más adecuada la segunda [1]. Por otro lado, se han propuesto diferentes metodologías para detectar discrepancias entre profesores en la evaluación de proyectos en enseñanzas de ingeniería e informática [17, 20].

Nuestra contribución consiste en una metodología basada en aprendizaje automático para identificar discrepancias, inconsistencias y sesgos en la calificación de un proyecto informático. Estudiamos su aplicación en la evaluación de un trabajo en el primer semestre del primer curso de los grados en ingeniería informática y matemáticas en la Universidad de La Rioja. El proyecto consiste en la realización de un sitio web formado por varias páginas, se realiza por grupos pequeños (de dos o tres miembros) y se califica por el equipo de profesores de la asignatura de forma que cada trabajo es calificado por solo uno de los profesores implicados. Los estudiantes reciben una lista de requisitos que debe cumplir el sitio web relacionados con la calidad del código HTML y CSS generado, adecuación con los estándares y cantidad de páginas e información requerida. Aquellos trabajos que cumplen los requisitos son evaluados atendiendo a la calidad de la información en el sitio web, legibilidad y calidad del código, diseño y navegabilidad. Utilizamos algoritmos genéticos para identificar las variables que influyen en la calificación final de los trabajos. En particular podemos detectar discrepancias entre profesores observando la variable *profesor*, de forma que identificamos trabajos que son equivalentes respecto a los criterios de corrección pero difieren en la nota final dependiendo del profesor que los califica. Observando variables como el género, la titulación que cursa cada estudiante u otras podemos detectar sesgos en la corrección. La virtud de los algoritmos genéticos es que nos permiten identificar aquellas variables que *sobreviven* gracias a su capacidad interpretativa en los modelos evaluados, lo que indica una prevalencia significativa.

Tras una primera sección de introducción, en la segunda sección de este trabajo exponemos el contexto académico y describimos el proyecto y los criterios de evaluación. En la tercera sección mostramos la metodología seguida. En la cuarta sección presentamos los algoritmos genéticos y los resultados obtenidos en el ejemplo de aplicación. Finalmente, exponemos las conclusiones extraídas y el trabajo futuro sugerido por ellas así como otros campos de aplicación.

2. Contexto académico y descripción de la tarea

Sistemas Informáticos es una asignatura común al grado en ingeniería informática y al grado en matemáticas en la Universidad de La Rioja. Se imparte en el primer semestre del primer curso como asignatura común a ambos grados y consta de seis créditos ECTS. Las horas lectivas se dividen en una hora semanal de teoría y dos sesiones semanales de laboratorio informático (de 90 minutos cada una). El trabajo autónomo de los estudiantes se estima en 90 horas a lo largo del semestre. Los estudiantes de ambos grados están mezclados tanto en las sesiones de teoría como en las sesiones prácticas. Cada año cursan la asignatura aproximadamente 75 estudiantes (50 del grado en ingeniería informática y 25 del grado en matemáticas).

La primera parte de la asignatura es una introducción básica a la arquitectura de ordenadores, protocolos de internet, codificación de caracteres, sistemas de ficheros y administración básica de sistemas operativos mediante línea de comandos e interfaz gráfica. La segunda parte se centra en los lenguajes HTML5 y CSS3 y consiste en una introducción al diseño de páginas web. El curso no sigue un libro de texto en particular, pero los contenidos están cubiertos en [2, 21].

La evaluación de la asignatura se divide en tres partes. Un 60 % de la nota final depende de una prueba escrita, un 20 % de los informes de prácticas del laboratorio informático y un 20 % de un proyecto realizado en grupos consistente en la realización de un sitio web relacionado con los contenidos de la asignatura.

Para realizar el proyecto los estudiantes son divididos en grupos de dos o tres personas sin tener en cuenta el grado que están estudiando. Se asigna un tema a cada grupo y los estudiantes deben colaborar para crear un sitio web que tenga exactamente el mismo contenido para todos los miembros del grupo (los ficheros HTML han de ser iguales) pero cada estudiante debe aplicar a su sitio web un diseño diferente (diferentes ficheros CSS). Así, cada grupo presenta un proyecto por cada estudiante, que recibirá una calificación individual. Los estudiantes disponen de treinta días para finalizar sus proyectos. El proyecto entregado debe satisfacer una serie de requisitos, de los que los más importantes son los siguientes:

1. Se deberá respetar la diferenciación entre contenido y aspecto, es decir, los ficheros HTML únicamente deberán centrarse en el contenido y su estructura y no deberán incluir ningún elemento sobre el aspecto o presentación de la información.
2. Debe haber al menos ocho ficheros HTML.
3. Debe haber al menos dos ficheros CSS.
4. Todos los ficheros HTML deberán verse afectados por alguna hoja de estilo y, al menos dos, deberán

verse afectados por al menos dos hojas de estilo.

5. Todos los ficheros HTML deberán pasar la validación de HTML5 según las especificaciones del World Wide Web Consortium (W3C) para HTML versión 5.
6. Los ficheros CSS deben ser validados por el validador CSS versión 3 del W3C.
7. Todas las páginas HTML deben incluir el atributo `charset`.
8. Las páginas web deben ser subidas al servidor que la universidad proporciona para los estudiantes de ingeniería informática y matemáticas.
9. Toda la web debe funcionar correctamente si es migrada a otro servidor. Para ello todos los enlaces internos deberán ser enlaces relativos (incluidas las imágenes).
10. Todas las páginas HTML deben incluir los iconos de validación HTML y CSS, que deben ser enlazados por rutas relativas y sin usar atributos de estilo en las etiquetas HTML correspondientes.
11. El `charset` declarado en cada fichero HTML debe coincidir con el usado al codificar y almacenar el fichero.
12. El sitio web debe tener un sistema de navegación que permita moverse de forma razonable por las diferentes páginas del sitio. Cada fichero HTML debe tener un menú que permita volver al primer nivel de páginas de la web.
13. No pueden usarse elementos `table` o `frame` para maquetar las páginas del sitio.
14. El trabajo debe ser implementado sin usar herramientas automáticas de desarrollo web.
15. Las páginas deben verse correctamente en los navegadores más habituales (Firefox, Chrome, Microsoft Edge).

Estos requisitos están diseñados para que los estudiantes no solo desarrollen código HTML y CSS correcto, sino también de acuerdo a los estándares [8, 10] y siguiendo los principios de diseño de HTML5: compatibilidad, utilidad, interoperabilidad y acceso universal [22]. Se añaden también algunas tareas opcionales que pueden suponer un aumento de la nota. Por ejemplo, se valora positivamente que se faciliten versiones del sitio web en distintos idiomas, que se haga un diseño adaptativo (*responsive design*) o que se cumplan las directrices de accesibilidad WCAG en alguno de sus niveles [3].

A lo largo de la asignatura se pone especial énfasis en la importancia de la validación de los documentos HTML y CSS y de la adhesión a los estándares. Por un lado, esto facilita desarrollos futuros y actualizaciones del trabajo, evita posibles problemas y contribuye a la adquisición de buenos hábitos, lo que resulta crucial en los primeros cursos de informática [7, 11]; siendo estas algunas de las habilidades transferibles y competencias

de desarrollo profesional propias de la asignatura que se pretenden mejorar con el proyecto. Por otro lado, la validación es un método eficaz para comprobar errores en HTML y CSS, facilitando el aprendizaje de estos lenguajes [18, 19]. Pese a estas ventajas muchos estudiantes no validan su código en los cursos de introducción al desarrollo web [19], por lo que se decidió imponerlo como un requisito.

3. Metodología

La evaluación del trabajo tiene aspectos técnicos que son fácilmente cuantificables, como comprobar si se cumplen los requisitos mínimos exigidos, el número de etiquetas HTML y propiedades CSS diferentes que se utilizan, el número de idiomas o si la página web es o no *responsive*. Por el contrario, la parte creativa no es tan fácil de evaluar por ser más subjetiva, como por ejemplo el aspecto general, las funcionalidades, el contenido, etc. Debido a que la asignatura de Sistemas Informáticos pertenece al primer curso de dos titulaciones y que además tiene muchos grupos de prácticas, a lo largo de los años ha habido un número variable de profesores que han impartido docencia y han tenido que calificar trabajos. De hecho, es habitual que cada curso haya nuevos docentes que no han impartido antes la asignatura. Esto hace que sospechemos de que no haya una uniformidad en la manera de corregir, sobre todo al evaluar la parte creativa. Por tanto, el objetivo de este estudio es intentar detectar posibles discrepancias e inconsistencias entre la corrección de los distintos profesores. Un segundo objetivo es detectar qué variables tienen más influencia en la calificación, y lo haremos mediante algoritmos genéticos: las variables que más persisten en la carrera evolutiva de los modelos son aquellas que más contribuyen a explicar la nota. Esto nos permitiría generar una rúbrica adecuada que se ajuste mejor a los objetivos de la evaluación de los trabajos. Observando variables como el género, la titulación que cursa cada estudiante u otras también podemos detectar sesgos en la corrección.

Lo primero que hicimos fue un análisis para intentar determinar qué variables podían influir en la calificación final del trabajo. Consideramos un total de 33, que fueron separadas en varios grupos:

- 23 variables acerca de la parte técnica y los requisitos mínimos, como el número de ficheros HTML, de ficheros CSS, su tamaño, el número de etiquetas distintas HTML y CSS utilizadas, etc.
- 6 variables categóricas acerca de la parte creativa: aspecto general, funcionalidad, contenido, posicionamiento, contrastes y legibilidad del código.
- 4 variables acerca del contexto del estudiante: titulación, género del estudiante, profesor que co-

rigió y número de miembros de su grupo.

Esta separación en grupos nos permite descubrir sesgos y discrepancias, siendo una metodología más sencilla que otros métodos que se han propuesto [17].

Una vez determinadas las variables, procedimos a extraer datos de las entregas de trabajos realizadas en los cursos 15-16, 16-17, 17-18, 18-19 y 19-20. En total, hicimos el estudio con 325 estudiantes. Para extraer la información de las 23 variables de la parte técnica, obtuvimos los datos automáticamente mediante una herramienta de desarrollo propio implementada para analizar todos los trabajos [9]. De manera similar, las variables del contexto pudieron obtenerse también automáticamente a partir de nuestros listados de estudiantes y notas. Sin embargo, la parte creativa no permite una automatización y sus variables son más difíciles de medir. A la hora de recabar dichos datos, intentamos evitar introducir discrepancias de diferentes profesores, de forma que un único profesor de la asignatura analizó manualmente todos los trabajos. De esta manera, en esta fase de recogida de datos para alimentar los modelos, todos los trabajos se analizaron en igualdad de condiciones. Para poder abordar de forma homogénea la utilización de diversos métodos estadísticos y de aprendizaje automático, se realizaron diversas transformaciones básicas como la normalización de las variables numéricas con *Z-Score* y la binarización de las variables categóricas. Previamente, fue necesario eliminar casos anómalos y reemplazar valores ausentes. La base de datos quedó finalmente formada por 322 instancias (se eliminaron 3 entregas que estaban incompletas) y 38 características de entrada más la variable de salida a estimar, *Calificación*.

La siguiente fase del análisis se centró en la creación de modelos de regresión que pudieran condensar el conocimiento intrínseco existente de las variables cuantitativas, que caracterizaban los aspectos técnicos del trabajo, así como de aquellas variables cualitativas que pudieran ayudar a detectar ausencia o presencia de discrepancias debidas a criterios de evaluación de cada profesor o, incluso, posibles sesgos que pudieran existir relacionados con las características propias de cada estudiante o grupo de estudiantes evaluados (género, titulación que cursan, etc.). El problema se abordó mediante dos métodos: uno clásico basado en modelos lineales Ridge con selección de variables hacia atrás y otro basado en optimización evolutiva que, mediante algoritmos genéticos (AG), realiza conjuntamente el ajuste de los modelos de aprendizaje automático (ML por sus siglas en inglés) y la selección de las variables más importantes. Todos los experimentos se desarrollaron en los lenguajes R y Python.

4. Modelos, análisis y resultados

4.1. Selección inicial de modelos

Inicialmente se seleccionaron los algoritmos de ML que pudieran ser más adecuados para un problema de regresión de este tipo. En este caso, las técnicas usadas fueron: regresión Ridge (RIDGE), máquinas vectores soporte para regresión (SVR), redes neuronales artificiales (ANN), métodos basados en vecinos más próximos (KNN), árboles de decisión (DT) y *Extreme Gradient Boosting Machines* (XGB). Los modelos fueron entrenados y ajustados mediante búsqueda en rejilla usando la librería `scikit-learn` de Python. Para homogeneizar el proceso de ajuste y validación de los diversos modelos de regresión, se seleccionó como métrica la raíz cuadrada del error cuadrático medio (RMSE por sus siglas en inglés). El motivo de elegir esta métrica es intentar evitar casos extremos donde la predicción falle en demasía, es decir, usando RMSE se penalizan, en mayor medida, las estimaciones más alejadas del valor real que si usásemos otras métricas como el error medio absoluto (MAE). La relación entre RMSE y MAE también es importante, pues si se observa que la proporción del RMSE es mucho mayor que el MAE, nos estará indicando que el modelo de regresión tiene un elevado número de errores mucho mayores que la media de los mismos. Debido al reducido tamaño de la base de datos, se diseñó una validación cruzada de 16 pliegues repetida 25 veces que aseguraba la robustez en las estimaciones. En el Cuadro 1 se muestran los parámetros de los mejores modelos obtenidos así como el error medio absoluto (MAE), el RMSE, la desviación estándar de los errores (SD), el coeficiente de determinación (R^2) y el intervalo de confianza del error de las estimaciones al 95 %. De entre estas técnicas, se seleccionaron los modelos RIDGE, por su simplicidad e interpretabilidad, y las SVR. Aunque la ANN demostró ser el método que obtuvo los menores MAE y RMSE, se eligió SVR por su facilidad a la hora de entrenar y su capacidad para alcanzar un mínimo global en el proceso de ajuste.

4.2. Análisis con *GA-PARSIMONY*

La búsqueda de modelos parsimoniosos (modelos de baja complejidad), es uno de los retos actuales en el campo del aprendizaje automático. Siguiendo el principio de parsimonia, también denominado “la regla de la navaja de Ockham”; en igualdad de condiciones, la explicación más sencilla suele ser la más probable. Aplicado al mundo del aprendizaje automático, entre modelos con similar precisión se recomienda elegir aquellos que tengan menor complejidad pues suelen ser mejores generalizadores del problema; además de que son más fáciles de comprender y más robustos

Modelo	MAE	RMSE	SD	R2	CI95 %
RIDGE (alpha=1.0e+02)	0,106	0,138	0,138	0,542	[-0,244, 0,328]
KNN (n_neighbors=2.0)	0,103	0,149	0,149	0,464	[-0,370, 0,280]
ANN (num_neuronas=6, alpha=0.000010)	0,082	0,115	0,115	0,683	[-0,248, 0,204]
DTreeRegressor (mae=random,max_depth=22)	0,094	0,132	0,132	0,582	[-0,306, 0,270]
SVR (C=1.0, epsilon=0.003, gamma=0.025119)	0,088	0,123	0,122	0,638	[-0,292, 0,252]
XGB (mdep=7,nroun=150,subsam=0.95,colsam=0.70)	0,091	0,124	0,124	0,629	[-0,261, 0,266]

Cuadro 1: Resultados obtenidos por los distintos modelos (junto con sus hiperparámetros optimizados mediante algoritmos genéticos) en Python.

frente a perturbaciones en sus entradas. Los mecanismos utilizados dentro de los algoritmos de ML, como la regularización o la selección de características, están enfocados en este sentido.

En este estudio, el entrenamiento y selección de los mejores modelos de ML se realizó con la metodología *GA-PARSIMONY* [25]. Esta metodología realiza una búsqueda de modelos de baja complejidad mediante algoritmos genéticos. El objetivo final es obtener modelos de alta precisión y baja complejidad a través del uso de selección de características, ajuste de los parámetros de entrenamiento del algoritmo y selección basada en parsimonia. Esto último es lo que diferencia a *GA-PARSIMONY* de otros métodos parecidos, pues la selección de los mejores individuos o soluciones de cada generación se realiza siguiendo un principio de búsqueda de parsimonia que consiste de dos pasos consecutivos: una preselección de los modelos más precisos y, de entre aquellos con similar coste, una promoción a puestos superiores de los que tengan menor complejidad. Todos los experimentos fueron implementados con el paquete *GAparsimony* [15] desarrollado en lenguaje R. Para realizar la optimización de AG con *GAparsimony* es necesario definir los cromosomas de cada individuo que va a ser entrenado con el algoritmo de ML correspondiente. El cromosoma viene definido por una combinación de los valores que se asignarán a los parámetros de entrenamiento del algoritmo y un vector que indicará cuáles son los atributos de entrada seleccionados para ese individuo. En particular, cada individuo i de cada generación g se define con un cromosoma λ_g^i formado por la concatenación de dos vectores P y Q , donde P corresponde con los parámetros de entrenamiento del algoritmo, y Q representa un vector de probabilidades usado en la selección de las características de modo que la variable j será incluida en el modelo si $q_j \geq 0,5$. Como función de ajuste J , se utilizó el RMSE de validación cruzada, $RMSE_{val}$. Por último, la complejidad del modelo quedó definida como el número de atributos seleccionados N_{FS} . Esta medida de complejidad ha mostrado ser muy eficaz en experiencias pasadas [16]. El proceso de optimización con AG se definió con una población de 40 individuos evaluados en 40 iteraciones pero con un criterio de pa-

rada si el error $RMSE_{val}$ no mejoraba en 20 generaciones seguidas. El proceso de selección utilizó el 20 % de los mejores individuos (elitistas). Con respecto a la elección de dichos parámetros, tanto el criterio de parada como el número de iteraciones se obtuvieron tras un proceso de ajuste mediante prueba y error. El resto de parámetros se eligieron de otras experiencias previas con bases de datos de similar tamaño y número de variables [16]. Posteriormente a la selección de los mejores individuos de cada generación, *GAparsimony* realizó los procesos clásicos de cruce de los cromosomas de los mejores individuos para crear la siguiente generación de individuos; así como la mutación de los cromosomas para poder crear más diversidad de soluciones en generaciones posteriores.

4.3. Resultados

Mediante el proceso de optimización con *GAparsimony*, se obtuvieron los mejores modelos RIDGE y SVR, con 15 y 18 variables respectivamente. Concretamente, el mejor modelo RIDGE obtuvo un MAE de 0,11 y RMSE 0,14; mientras que el mejor modelo SVR consiguió un MAE de 0,103 y RSME 0,137. A la vista de los datos mostrados en el Cuadro 2, podemos observar los siguientes resultados:

- Las variables *Genero* y *Titulacion* no son seleccionadas para ninguno de los dos modelos, lo que nos lleva a concluir que no se producen sesgos en las calificaciones de los trabajos en base a estas dos características.
- Las variables correspondientes a tres de los profesores (*Profesor4*, *Profesor5*, *Profesor6*) no han sido seleccionadas, luego entre estos profesores no se producen discrepancias. Sin embargo, dos de los profesores (*Profesor1* y *Profesor3*) sí que aparecen como características seleccionadas para ambos modelos, un hecho que nos ha permitido detectar posibles discrepancias en las correcciones. Dichas discrepancias en la corrección han sido confirmadas por medio de test estadísticos que muestran diferencias significativas ($p < 0,05$).
- Las variables que más influyen en la calificación

RIDGE	SVR
U1 (100)	AspectoGeneral (99,69)
EtiquetasHTML (100)	Profesor1 (99,06)
Profesor1 (100)	FicherosImagenes (98,75)
AspectoGeneral (100)	Responsive (98,13)
Contrastes (99,69)	Profesor3 (97,81)
Profesor3 (99,06)	noCumplenAccesibilidad (96,88)
Posicionamiento (99,06)	Contrastes (96,88)
EtiquetaIMG (98,75)	TotalBytesCSS (95,31)
LegibilidadCodigo (98,75)	NumJavaScript (94,69)
HTMLAnalizados (98,44)	Contenido (94,69)
PosiblesProblemasAcc (96,25)	Idiomas (94,06)
NumJavaScript (96,25)	U1 (90,94)
EtiquetasCSS (92,5)	Funcionalidad (87,19)
Contenido (90,63)	ModificadoresCSS (85,63)
Tablas (80,94)	ErroresPotencialesAcc (76,56)
MiembrosGrupo (46,56)	EtiquetasHTML (71,56)
Form (38,75)	Profesor6 (56,25)
Genero (27,82)	Profesor2 (55,31)
Ol (25,63)	ErroresAccesibilidad (47,19)
Profesor2 (23,8)	CSSAnalizados (40,31)
Videos (23,44)	Tablas (40,31)
Profesor6 (22,5)	LegibilidadCodigo (34,38)
ErroresPotencialesAcc (20,31)	UsosValidador (28,75)
HTMLcon2CSS (15)	TotalBytesHTML (22,19)
TotalBytesCSS (14,38)	Profesor5 (19,06)
Funcionalidad (13,12)	HTMLcon2CSS (17,5)
Titulacion (12,81)	EtiquetasCSS (16,56)
ErroresAccesibilidad (10)	PosiblesProblemasAcc (14,69)
TotalBytesHTML (7,81)	HTMLAnalizados (14,38)
Responsive (7,5)	Videos (14,38)
UsosValidador (7,5)	MiembrosGrupo (13,44)
Idiomas (7,19)	Genero (13,12)
noCumplenAccesibilidad (6,88)	Form (12,81)
Profesor4 (6,88)	Ol (9,69)
Profesor5 (6,88)	Profesor4 (9,69)
CSSAnalizados (6,25)	Titulacion (8,75)
ModificadoresCSS (3,75)	EtiquetaIMG (8,44)
FicherosImagenes (2,81)	Posicionamiento (6,56)

Cuadro 2: Variables seleccionadas (en negrita) para los mejores modelos RIDGE y SVR. Entre paréntesis se muestra el porcentaje de aparición de cada variable dentro de los elitistas de las últimas generaciones.

(son elegidas para ambos modelos), además de los profesores comentados anteriormente, son las siguientes: **AspectoGeneral**, **Contenido**, **LegibilidadCodigo**, **Contrastes**, **NumJavaScript**, **EtiquetasHTML** y **U1**. Hacemos notar que cuatro de ellas son variables cualitativas acerca de la parte creativa del trabajo y, por lo tanto, subjetivas; lo que puede explicar el hecho de que las calificaciones sufran diferencias significativas entre algunos de los profesores y posibles inconsistencias en cada profesor. El

hecho de que la variable **EtiquetasHTML** aparezca entre las seleccionadas parece razonable ya que uno de los aspectos que se tiene en cuenta para la evaluación por parte de los profesores es la utilización del mayor número posible de elementos HTML explicados en clase.

- Las características que menos influyen en la calificación (no son seleccionadas para ninguno de los dos modelos), además del género, la titulación y los profesores anteriormente comentados, son: **CSSAnalizados**, **HTMLcon2CSS**, **Form**,

Ol, Videos, ErroresAccesibilidad, ErroresPotencialesAcc, MiembrosGrupo y UsosValidador. Algunas de estas variables tienen un valor similar en casi todos los trabajos o bien responden a procesos auxiliares en la realización del trabajo cuyo efecto en la nota final no es significativo.

- Algunas variables son elegidas para un modelo y no para el otro, aunque las consideramos relevantes porque en ambos modelos existen variables relacionadas que sí que son elegidas. Por ejemplo, podemos observar que para el modelo RIDGE no se elige la variable correspondiente al número de propiedades CSS (`ModificadoresCSS`); sin embargo, sí que se selecciona la variable `EtiquetasCSS`, por lo que podemos deducir que en la corrección sí que se tiene en cuenta el número de propiedades CSS utilizadas. Además, en SVR hay variables que apenas aparecen en los elitistas de las últimas generaciones (por ejemplo, `TotalBytesHTML`), sin embargo sí que han sido seleccionadas en el mejor modelo. Esto nos indica que su importancia podría ser en realidad menor, a pesar de haber sido elegidas.

5. Conclusiones y trabajo futuro

Los resultados obtenidos aplicando algoritmos genéticos a los resultados de evaluación de un trabajo informático con doble componente técnica y creativa nos permiten sacar varias conclusiones. En primer lugar, estos modelos, al basar su funcionamiento en la persistencia de las variables al ir evolucionando los modelos explicativos del fenómeno, nos permiten hacer una valoración ajustada de la influencia de las variables en el resultado final. En el caso de la evaluación de un trabajo como el que estamos analizando, nos permite valorar el papel que cada una de las variables tiene en la explicación de variaciones en las calificaciones, lo que supone una aproximación al propio proceso de evaluación de los trabajos. La evaluación de los métodos de enseñanza y corrección se basa usualmente en la propia subjetividad del profesor [4], algo que la metodología que presentamos puede contribuir a reducir. Por otro lado, uno de los retos fundamentales de la aplicación de metodologías basadas en proyectos o trabajos es el diseño de sistemas de evaluación adecuados [6, 13]. Una rúbrica correctamente establecida proporciona una estandarización de la evaluación al especificar una guía para la calificación de acuerdo a criterios claramente establecidos de antemano. Esto tiene el efecto de reducir discrepancias, inconsistencias y sesgos. El diseño de rúbricas útiles es una componente importante para una mejora de la evaluación en particular en lo que se refiere a aspectos que mezclan

subjetividad y objetividad, como la calidad del código en cursos de programación [23]. La metodología que presentamos proporciona una forma de iniciar el trabajo de desarrollo de rúbricas basada en la experiencia docente previa y con la intención explícita de reducir sesgos y discrepancias. La discriminación de las variables que tienen más influencia en la variabilidad de la nota son señaladas por esta metodología y serán por tanto tenidas en cuenta para establecer de forma precisa los criterios de evaluación en torno a ellas en el diseño de rúbricas. Finalmente, la evaluación de trabajos con componente técnica y componente creativa es muy común en muchos estudios de grado y máster, en particular en estudios técnicos. La detección de discrepancias y sesgos en la evaluación puede ser complicada y costosa. La metodología propuesta automatiza la mayor parte del proceso de selección de las variables que nos permiten descubrir sesgos y discrepancias, de modo que ambos puedan ser evitados en el diseño de la evaluación. Partiendo de un conjunto de variables que definan las características que intervienen en el trabajo, los algoritmos genéticos nos permiten una evaluación adecuada de estas variables y al ser el proceso automático, podemos refinar o ampliar el conjunto de variables en sucesivas interacciones. Como trabajo futuro nos planteamos facilitar una interfaz y guía que permita el uso de la metodología propuesta en otras situaciones similares, además de una herramienta que genere automáticamente una posible rúbrica.

Referencias

- [1] Daniel W. Allen. Toward a Common Currency: Using Item Response Theory to Adjust for Grading Inconsistency. *Mathematics, Statistics, and Computer Science Honors Projects*, Macalester College, 2007. Disponible en https://digitalcommons.macalester.edu/mathcs_honors/7. Fecha de acceso: 16 de abril de 2021.
- [2] Glenn Brookshear y Dennis Brylow. *Computer Science: An Overview*. Pearson Education, 2018.
- [3] Alastair Campbell, Andrew Kirkpatrick, Michael Cooper y Joshue O Connor. Web Content Accessibility Guidelines (WCAG) 2.1 *W3C Recommendation*, 2018. Disponible en <https://www.w3.org/TR/2018/REC-WCAG21-20180605/>. Fecha de acceso: 16 de abril de 2021.
- [4] Angela Carbone y Jens J. Kaasbøll. A survey of methods used to evaluate computer science teaching. *SIGCSE Bulletin ACM*, 30(3):41–45, 1998.
- [5] Julián Chaparro-Peláez, Santiago Iglesias-Pradas, Félix J. Pascual-Miguel y Ángel

- Hernández-García. Factors Affecting Perceived Learning of Engineering Students in Problem-Based Learning Supported by Business Simulation. *Interactive Learning Environments*, 21(3):244–262, 2013.
- [6] Juebei Chen, Anette Kolmos y Xiangyun Du. Forms of implementation and challenges of PBL in engineering education: a review of literature. *European Journal of Engineering Education*, 46(1):90–115, 2021.
- [7] Al Cuoco, E. Paul Goldenberg y June Mark. Habits of mind: An organizing principle for mathematics curricula. *The Journal of Mathematical Behavior*, 15(4):375–402, 1996.
- [8] Alex Danilo, Sangwhan Moon, Steve Faulkner, Arron Eicholz y Travis Leithhead. HTML 5.2 W3C Recommendation, 2017. Disponible en <https://www.w3.org/TR/2017/REC-html52-20171214/>. Fecha de acceso: 16 de abril de 2021.
- [9] Jose Divasón, Ana Romero y Eduardo Sáenz-de-Cabezón. Experiencia con una herramienta para automatizar la comprobación de requisitos de los trabajos HTML–CSS de Sistemas Informáticos. En *Actas de las XXIV Jornadas de Enseñanza Universitaria de Informática, Jenui 2018*, vol. 3, páginas 173 – 180, Universitat Oberta de Catalunya, 2018.
- [10] Erika J. Etemad, Tab Atkins Jr. y Florian Rivoal. CSS Snapshot 2020, Disponible en <https://www.w3.org/TR/2020/NOTE-css-2020-20201222/>. Fecha de acceso: 16 de abril de 2021.
- [11] Saj-Nicole A. Joni y Elliot Soloway. But my program runs! Discourse rules for novice programmers. *Journal of Educational Computing Research*, 2(1):95–125, 1986.
- [12] Judy Kay, Michael Barg, Alan Fekete, Tony Greening, Owen Hollands y Jeffrey H. Kingston. Problem-Based Learning for Foundation Computer Science Courses. *Computer Science Education*, 10(2):109–128, 2000.
- [13] Tanya Kumberger. Revising a Design Course from a Lecture Approach to a Project-Based Learning Approach. *European Journal of Engineering Education*, 38(3):254–267, 2013.
- [14] José Luis Martín Núñez, Edmundo Tovar Caro y José Ramón Hilera González. From Higher Education to Open Education: Challenges in the Transformation of an Online Traditional Course. *IEEE Transactions on Education*, 60(2):134–142, 2016.
- [15] Francisco Javier Martínez-De-Pisón. GAparsimony: GA-based optimization R package for searching accurate parsimonious models. R package version 0.9-4, 2020. Disponible en <https://github.com/jpison/GAparsimony>.
- [16] F.J. Martínez-de-Pison, J. Ferreiro, E. Fraile y A. Pernia-Espinoza. A comparative study of six model complexity metrics to search for parsimonious models with GAparsimony R Package. *Neurocomputing*, In Press., 2020. DOI: 10.1016/j.neucom.2020.02.135
- [17] Juan M. Montero, Rubén San-Segundo, Javier Macías-Guarasa, Ricardo de Córdoba y Javier Ferreiros. Methodology for the Analysis of Instructors' Grading Discrepancies in a Laboratory Course. *International Journal of Engineering Education*, 22(5):1053–1062, 2006.
- [18] Thomas H. Park y Susan Wiedenbeck. Learning Web Development: Challenges at an Earlier Stage of Computing Education. In *Proceedings of the Seventh International Workshop on Computing Education Research*, 2011. DOI: 10.1145/2016911.2016937
- [19] Thomas H. Park, Brian Dorn y Andrea Forte. An Analysis of HTML and CSS Syntax Errors in a Web Development Course. *ACM Transactions on Computing Education*, 15(1):1–21, 2015.
- [20] Stefan Pero y Tomas Horváth. Detection of Inconsistencies in Student Evaluations. In *Proceedings of the 5th International Conference on Computer Supported Education (CSEDU-2013)*, Mayo 2013. DOI:10.5220/0004385602460249
- [21] G. Michael Schneider y Judith Gersting. *An Invitation to Computer Science*. Wadsworth Publishing Co. Inc., sexta edición, 2012.
- [22] Maciej Stachowiak y Anne van Kesteren. HTML Design Principles W3C Working Draft, 2007. Disponible en <https://www.w3.org/TR/2007/WD-html-design-principles-20071126/>. Fecha de acceso: 16 de abril de 2021.
- [23] Martijn Stegeman, Erik Barendsen y Sjaak Smetsers. Designing a Rubric for Feedback on Code Quality in Programming Courses. En *Proceedings of the 16th Koli Calling International Conference on Computing Education Research*, 2016. DOI:10.1145/2999541.2999555
- [24] Pamela Steinke y Peggy Fitch. Minimizing Bias When Assessing Student Work. *Research & Practice in Assessment*, 12:87–95, 2017.
- [25] R. Urraca, E. Sodupe-Ortega, J. Antonanzas, F. Antonanzas-Torres y F.J. Martínez-de-Pison. Evaluation of a novel GA-based methodology for model structure selection: The GAPARSIMONY. *Neurocomputing*, 271:9–17, 2018.

¿Qué nos dicen los estudiantes sobre lo que hace que funcione la clase invertida?

Mercedes Marqués Andrés, José Manuel Badía Contelles

Departamento de Ingeniería y Ciencia de los Computadores, Universitat Jaume I de Castelló

mmarques@uji.es, badia@uji.es

Resumen

La clase invertida es uno de los modelos docentes que más se ha popularizado en los últimos años. La mayoría de los estudios publicados indican que mejora el aprendizaje y la satisfacción del estudiantado. Sin embargo, hay profesorado que dice haberla aplicado sin éxito y no hay un acuerdo sobre qué factores hacen que este modelo funcione. En este trabajo estudiamos las opiniones del estudiantado de los últimos cinco cursos en los que hemos aplicado la clase invertida. En concreto, estudiamos las razones que aducen para considerarla una metodología mejor que la tradicional clase magistral, y estudiamos si la dedicación temporal que requiere va en detrimento de otras asignaturas. Las opiniones recabadas confirman que la clase invertida es efectiva si se diseña e implementa de forma adecuada, esto es, proporcionando buenos materiales (libros, vídeos, etc.), haciendo un buen diseño de las actividades previas y enlazándolas con las que se lleven a cabo en clase, que deben servir para resolver dudas y ampliar y profundizar en lo aprendido. Además, el estudiantado indica que puede llevar a cabo las actividades requeridas sin perjudicar a otras asignaturas. Por último, las opiniones obtenidas durante el pasado curso 19/20 indican que esta metodología es robusta y fácilmente adaptable a la docencia en línea.

Abstract

Flipped classroom is one of the most popular teaching models in recent years. Most studies indicate that it improves student's learning and satisfaction. However, there are teachers who have tried to use this model unsuccessfully, and there is no agreement on what factors contribute to its success. In this paper we study our students' opinion in the last five years in which we have flipped the classroom. Specifically, we study the reasons they give for considering it is better than the traditional lecture class, and we study whether the time committed to it is detrimental to other subjects. The opinions we collected from our students confirm that for the flipped classroom to be effective it has to be designed

and implemented appropriately, i.e. by providing good course materials (books, videos, etc.), providing well-designed out-of-class activities, and planning the class so that students have feedback, can ask doubts, and also work on deepening what they learnt before the class. In addition, students indicate that they can carry out the required activities without affecting other subjects. Finally, the opinions obtained during the last academic year 19/20 show that this methodology is robust and easily adaptable to online teaching.

Palabras clave

Clase invertida, aprendizaje, percepción del alumnado, investigación educativa.

1. Introducción

En lo que consideramos enseñanza tradicional, la exposición de contenidos se realiza mediante la clase magistral. Se espera que los estudiantes comprendan todo en el momento, aunque en realidad no tienen tiempo para reflexionar sobre lo que se dice y es fácil perderse alguna cuestión importante mientras se intenta escribir todo lo que el profesor explica. Cuando se invierte la clase (*flipped classroom*), lo que se hace es trasladar fuera del aula la exposición de los contenidos, que se proporcionan a los alumnos en forma de documentos, vídeos, etc. que deberán revisar antes de la clase [4]. De este modo, cada estudiante puede dedicar el tiempo que necesite para comprender los conceptos: puede volver a leer o pasar el vídeo adelante y atrás, revisar los ejemplos, etc. Después, la clase se dedica a trabajar los contenidos con más profundidad y ampliarlos, aclarar dudas, etc., con la ayuda del profesor y los compañeros.

Si bien encontramos numerosos estudios que presentan las bondades de esta manera de plantear las clases, algunos de los cuales se repasan en el siguiente apartado, es también cierto que en la comunidad docente universitaria existe cierta controversia. A través de conversaciones informales encontramos que no son

pocos los profesores que, sin haberlo intentado, opinan que sería imposible impartir todas las asignaturas de este modo por la sobrecarga de trabajo que supone para los estudiantes; otros opinan que con su asignatura o con sus alumnos nunca podría funcionar; y también los hay que aún habiéndolo intentado, no han logrado aplicarla con éxito.

Ya que la nuestra es una experiencia de éxito, nos planteamos investigar sobre qué es aquello que hace que funcione según la perspectiva de nuestros estudiantes. Para ello, hemos llevado a cabo una investigación cualitativa a partir de la encuesta de final de curso, en la que les preguntamos sobre la metodología docente empleada en la asignatura. Concretamente, las preguntas de investigación planteadas en este trabajo son las siguientes:

1. ¿Cómo repercute la clase invertida en la organización del tiempo de dedicación de nuestros estudiantes?
2. ¿Qué aspectos de la clase invertida son los que favorecen el aprendizaje de nuestros estudiantes?
3. ¿Cómo ha afectado al desarrollo de la docencia de la asignatura el paso a la docencia no presencial sufrida a mitad del segundo semestre del curso 19/20?

Las justificaciones aportadas por los estudiantes en las encuestas de los últimos cinco cursos han sido objeto de un análisis de contenido, del cual extraemos los resultados referentes a las preguntas de investigación a las que pretendemos dar respuesta.

Este artículo se ha organizado en siete apartados. Después de esta introducción, se hace una revisión de diversos estudios previos sobre la clase invertida. A continuación se explica la metodología empleada en este trabajo y el contexto en que se ha realizado. Después se presentan los resultados del análisis de contenido realizado a partir de las encuestas y se realiza la discusión de los mismos para dar respuesta a las preguntas de investigación, finalizando con las conclusiones.

2. Antecedentes

Existe una creciente cantidad de artículos sobre la clase invertida. La mayoría de ellos estudian los resultados de aplicar alguna variante de este modelo docente en un determinado contexto e intentan extraer conclusiones sobre su funcionamiento y eficacia para mejorar el aprendizaje del alumnado [1]. Sin embargo, son pocos los artículos que se centran en la satisfacción del alumnado con la clase invertida o su percepción sobre la utilidad para su aprendizaje. Los artículos que analizan estos aspectos afirman, en su mayoría, que los estudiantes están satisfechos con el uso de la

clase invertida en comparación con métodos tradicionales basados en la clase magistral [3, 10, 11, 13]. Sin embargo, diversos estudios indican que es difícil obtener respuestas concluyentes y que la opinión de los estudiantes puede depender de otros factores como la experiencia del profesorado, el uso que se haga de la tecnología o el curso en que se aplique [6, 8, 12].

Un objetivo perseguido por diversos estudios es intentar determinar los aspectos del diseño o implementación de la clase invertida que influyen en la satisfacción del alumnado. Por ejemplo, la disponibilidad de vídeos en línea o la posibilidad de trabajar de modo colaborativo con otros alumnos durante las clases es un aspecto valorado positivamente [3]. Otros estudios destacan que la simple incorporación de vídeos o la planificación de actividades previas a la sesión no son suficientes para implementar la clase invertida. Ni siquiera la incorporación de cuestionarios asociados a los vídeos garantiza su funcionamiento. En algún caso se ha observado que añadir ciertos aspectos de gamificación a la clase invertida mejora la motivación de los alumnos [9, 15]. En cuanto al tipo de actividad a realizar durante las sesiones y cómo se llevan a cabo, se ha observado que también tiene un impacto directo en la satisfacción del alumnado [12, 14].

Otro factor destacado en algunos de los estudios consultados es la adecuada presentación de la metodología para evitar la frustración inicial que se puede provocar por tener que preparar la materia previamente a las clases [10] o por cuestionar el papel que juega el profesorado [12]. Los estudios citados detectan una rápida adaptación de los alumnos a la metodología, sobre todo si se introduce desde los primeros cursos. En otro estudio se llega a conclusiones similares, los alumnos opinan que la clase invertida requiere un esfuerzo inicial y disciplina, pero en pocas semanas les parece un mejor uso del tiempo de clase y de preparación para el aprendizaje autónomo [6].

Uno de los obstáculos fundamentales revelados por ciertos estudios es la necesidad de formar al profesorado en el diseño e implementación adecuado de la clase invertida, así como en el uso de las tecnologías implicadas [10]. En este sentido es importante la participación de un profesorado reflexivo e implicado de modo activo colaborando con otros profesores.

3. Metodología

Esta investigación se ha desarrollado en el marco de una asignatura de Bases de datos del Grado de Diseño y Desarrollo de Videojuegos de la Universitat Jaume I de Castelló. La asignatura se viene impartiendo desde el curso 13/14 siguiendo el modelo de la clase invertida. Todos los cursos se pasa una encuesta propia de la asignatura para recoger la opinión del alumnado sobre

su desarrollo y así identificar posibles áreas de mejora. Básicamente, en la encuesta nos interesamos en conocer el tiempo semanal dedicado y la percepción del alumnado en cuanto a si la metodología docente contribuye a su aprendizaje.

La encuesta se proporciona a los alumnos en el momento en que entregan el examen de la primera convocatoria, se facilita en papel y no es obligatorio contestarla. En el curso 19/20 el examen se realizó de manera no presencial, por lo que la encuesta se proporcionó a través de un formulario electrónico enlazado al final del enunciado del examen. Tanto la encuesta en papel como la electrónica son anónimas.

En la encuesta hay una combinación de preguntas cerradas y abiertas (en relación al tipo de respuesta). Para cada aspecto sobre el que se quiere conocer la percepción de los estudiantes se hace una pregunta cerrada para hacer una valoración dentro de un rango, y una pregunta abierta para aportar una justificación a la valoración realizada. Por ejemplo, en la pregunta cerrada «Preparando tú la clase con antelación y resolviendo ejercicios en el aula ¿crees que has aprendido peor, mejor o igual que ...», se ha de escoger entre peor, mejor o igual, y a continuación aparece la pregunta abierta: «Por favor, justifica tu respuesta».

Si bien la encuesta se ha pasado en los siete cursos en que se lleva impartiendo la asignatura en el grado, el formato de preguntas combinadas, cerrada y abierta, se ha utilizado solamente en los cinco últimos cursos, por lo que son estos los cursos que hemos analizado.

Dado que en el curso 19/20 se impartió la segunda mitad de la asignatura de forma no presencial debido a las medidas sanitarias impuestas por la pandemia de la COVID-19, se incluyó en la encuesta una pregunta abierta adicional para conocer la percepción de nuestros alumnos en relación a cómo había sido la adaptación realizada en la asignatura.

Las justificaciones aportadas por los estudiantes en las preguntas abiertas de la encuesta de los últimos cinco cursos han sido objeto de un análisis de contenido cualitativo sin cuantificación [7]. Concretamente, se han analizado las preguntas de la encuesta directamente relacionadas con cada una de las preguntas de investigación planteadas en el primer apartado de este trabajo. El procedimiento inductivo que se ha seguido es un proceso de codificación abierta. Las respuestas dadas por los alumnos se han leído detenidamente y se han dividido en fragmentos según el tema al que hacían referencia, obteniendo así una lista de temas clave. Estos temas clave se han comparado entre ellos y se han agrupado en categorías. En relación al tiempo de dedicación, las categorías que han emergido son tres: adecuación de las actividades no presenciales, adecuación de la distribución semanal de la carga de trabajo y aportación del trabajo no presencial a la nota final. En

relación a la repercusión de la clase invertida sobre el aprendizaje emergen cuatro categorías: aportación de las actividades previas a las clases de teoría, uso del tiempo de clase, oportunidad de aprender del error y materiales de trabajo de la asignatura. Y, por último, en relación a la adaptación a la docencia no presencial en la segunda mitad de la asignatura durante el curso 19/20, las categorías han sido: continuidad de las clases sincrónicas, adaptación de la evaluación, preocupación por los alumnos y capacidad de adaptación de la metodología a la docencia no presencial. Ya que nuestro interés radica en tener un visión condensada de todo aquello que los estudiantes aportan en las justificaciones que realizan, la frecuencia con la que aparecen las categorías no es relevante y por eso el estudio cualitativo se ha realizado sin cuantificación.

4. Contexto

En la asignatura en la que se ha llevado a cabo este estudio, el alumnado debe asistir a 58 horas de clase presencial y realizar 92 horas de trabajo no presencial: 62 horas de trabajo personal y 30 horas de preparación del examen. Distribuyendo las horas de trabajo personal entre las 15 semanas del semestre se obtiene una dedicación semanal media de unas 4 horas. En total, cada semana hay: 2 horas de clase de teoría, 2 horas de clase práctica (de laboratorio o de problemas) y 4 horas de trabajo personal no presencial. En cuanto a la evaluación, la ficha de la asignatura establece que la evaluación continua supone un 50 % y el examen el otro 50 %. Las horas de trabajo personal, y su contribución a la evaluación continua, están planificadas como se describe a continuación:

- El alumno debe preparar la clase de teoría leyendo determinados apartados del libro y realizando ejercicios que le sirven para aplicar los conceptos estudiados y comprobar si los ha entendido; se dispone también de vídeos donde se resuelven ejercicios. La dedicación estimada es más alta las primeras semanas del curso y disminuye al final, siendo la media de 2 horas semanales. Entregar todas las actividades previas tiene un peso de 1,5 puntos en la evaluación continua.
- Antes de cada práctica de laboratorio el alumno debe completar un formulario con preguntas que le sirven para prepararla. El tiempo estimado es de media hora para cada una de las diez prácticas programadas.
- Después de cada práctica de laboratorio el alumno debe hacer una autoevaluación comparando sus soluciones con las soluciones publicadas para detectar las diferencias y hacer una reflexión sobre sus causas (errores, soluciones alternativas, etc.).

El tiempo estimado para cada autoevaluación es de una hora y media. El alumno obtiene 0,2 puntos de la evaluación continua por cada práctica realizada y autoevaluada adecuadamente.

- Se realizan cuatro sesiones de problemas en las semanas en que no hay laboratorio, no requieren preparación previa. Después de la clase de problemas, el alumno debe reflexionar en un diario sobre lo que ha aprendido y las dudas que han quedado, para lo que se estima una hora. Asistir a todas las sesiones de problemas y realizar las reflexiones tiene un peso de 0,5 puntos en la evaluación continua.

5. Resultados

En este apartado analizamos, en primer lugar, la percepción de nuestros alumnos sobre la implementación de la clase invertida en relación al tiempo que dedican y, a continuación, analizamos cómo contribuye a mejorar su aprendizaje. Ambos análisis se han realizado a partir de las encuestas a los alumnos de los últimos cinco cursos. Por último, en este apartado analizamos también las opiniones de los alumnos sobre cómo se realizó la adaptación a la docencia no presencial con motivo del decreto del estado de alarma en marzo de 2020 (se recogieron en la encuesta realizada en el curso 19/20).

En el Cuadro 1 se muestra el número de encuestas respondidas en cada curso sobre el número de alumnos presentados en la primera convocatoria (recordemos que se pasa la encuesta, anónima y voluntaria, al terminar dicho examen) y en el Cuadro 2 se muestra el número de alumnos matriculados en cada curso y las tasas de éxito del curso (aprobados sobre presentados teniendo en cuenta las dos convocatorias). Queremos puntualizar aquí que, aunque no se refleja en los cuadros, observamos en los cinco años de estudio que alrededor del 10 % del alumnado matriculado no llega a cursar la asignatura, ya que no constan entregas de actividades de evaluación continua o las han hecho de manera muy esporádica. Desconocemos el motivo por el que estos alumnos tienen este comportamiento, si bien no encontramos motivos para achacarlo a la metodología seguida ya que no hay evidencias de que lleguen a experimentarla.

5.1. Sobre el tiempo de dedicación

En la encuesta se pregunta a los alumnos por la dedicación semanal para realizar el trabajo no presencial. Es una pregunta cerrada formulada de la siguiente manera: «En promedio, ¿cuántas horas no presenciales has dedicado cada semana a la asignatura para preparar

	Curso académico				
	15/16	16/17	17/18	18/19	19/20
Presentados 1a	51	47	45	49	48
Encuestas	51	41	44	42	39
%enc/pres	100 %	87 %	98 %	86 %	81 %

Cuadro 1: Encuestas respondidas sobre los alumnos presentados en la primera convocatoria.

	Curso académico				
	15/16	16/17	17/18	18/19	19/20
Matriculados	58	55	52	60	54
Presentados 1a+2a	52	47	46	51	50
Aprobados	46	45	35	42	45
%apro/pres	88,5 %	95,7 %	76,1 %	82,4 %	90 %

Cuadro 2: Alumnos matriculados y tasa de éxito (aprobados sobre presentados entre las dos conv.).

la clase de teoría, la de prácticas y hacer la autoevaluación?». Las opciones que se dan son: menos de 3 horas, alrededor de 3 horas, alrededor de 4 horas, alrededor de 5 horas y más de 5 horas. En la Figura 1 se observa que, a excepción del curso pasado, que se vio afectado por el cambio a la docencia no presencial, en el resto de los cursos más del 85 % de los alumnos declara haber dedicado a la asignatura alrededor de 4 horas o menos, por lo que está en línea con el trabajo personal que se requiere en la ficha de la asignatura y la estimación realizada por el profesorado. Hay que tener en cuenta que estas horas incluyen tanto la preparación previa de la clase de teoría, como la preparación de las clases prácticas, junto a las autoevaluaciones y reflexiones posteriores.

En el curso 19/20 encontramos una diferencia importante: baja al 69,2 % el alumnado que dice haber dedicado alrededor de 4 horas o menos (la dedicación esperada) y aumenta el número de quienes dicen haber dedicado más tiempo: el 20,5 % ha dedicado alrededor de 5 horas y el 10,2 % ha dedicado más de 5 horas. Es interesante observar que aunque en total el 30,7 % dice haber dedicado más de las 4 horas no presenciales previstas, solo un 18 % contesta «sí» a la pregunta «¿Crees que esta asignatura te ha robado tiempo de dedicación a otras asignaturas?». Este efecto podría deberse al confinamiento, que ha permitido dedicar más tiempo a los estudios en general, aunque no tenemos evidencias de ello. Hay que tener en cuenta también que los cambios que se introdujeron ante el paso a la docencia no presencial consistieron en aumentar el peso de la evaluación continua añadiendo algunas actividades y rebajando el peso del examen al 40 %, siguiendo así las recomendaciones de la universidad.

En relación a la pregunta cerrada «¿Crees que esta asignatura te ha robado tiempo de dedicación a otras

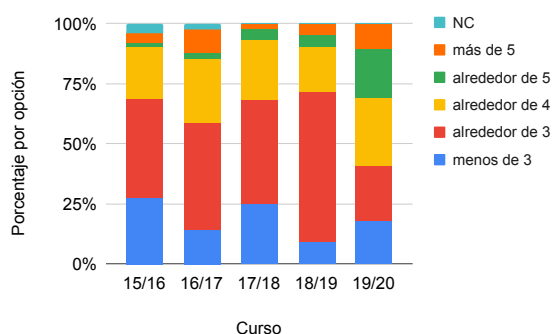


Figura 1: Dedicación semanal al trabajo no presencial en la asignatura.

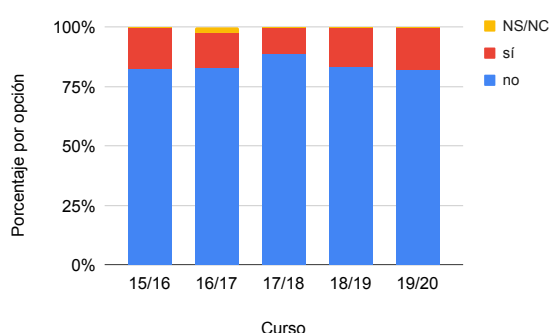


Figura 2: Perjuicio sobre la dedicación a otras asignaturas causado por el trabajo no presencial.

asignaturas?», en la Figura 2 se observa que en todos los cursos el porcentaje de alumnos que contesta «no» está por encima del 82 %. Nos parece que este dato es muy revelador ya que un temor bastante habitual entre el profesorado que no ha usado la metodología es, precisamente, que si todas las asignaturas trabajáramos de la misma forma, los estudiantes no tendrían suficiente tiempo para todas ellas. Sin embargo, nuestra experiencia de los últimos cinco años es que con una planificación bien realizada, los estudiantes encuentran más ventajas que inconvenientes, como veremos a continuación, y que su dedicación semanal a otras asignaturas no se ve perjudicada.

Después de la pregunta cerrada «¿Crees que esta asignatura te ha robado tiempo de dedicación a otras asignaturas?» hay una pregunta abierta: «Por favor, justifica tu respuesta». Ya que han tenido que trabajar semanalmente en la asignatura, les haya robado tiempo o no, queremos que piensen sobre la respuesta dada en la pregunta anterior para valorar cómo ha sido el tiempo que han dedicado de manera no presencial. Analizando las respuestas dadas a esta pregunta en las encuestas de los últimos cinco cursos identificamos tres categorías en el análisis de contenido que ponen de relieve lo que se pueden considerar como ventajas de la metodología empleada.

Una primera categoría es relativa las características de las actividades no presenciales: se consideran esenciales para el aprendizaje. Es por ello que se valora positivamente el tiempo dedicado, porque se considera que es útil:

«Pienso que las actividades que hacemos durante las horas no presenciales son muy importantes para esta asignatura y que son necesarias. Además de no quitarme tiempo de otras, ya que tampoco son tantas horas a la semana.»

Ser esencial se traduce en que si no realizaran las actividades, el aprendizaje sería más costoso:

«Está muy bien organizada para que estudies lo justo cada semana antes de la práctica o la teoría sin que te robe tiempo de otras. Las previas son adecuadas para lo que damos y te ayudan a aprender bastante más que si directamente no hubiese. Creo que de no tener previas, pasaríamos el doble de tiempo estudiando el temario.»

Una segunda categoría es relativa al tiempo que se ha de dedicar semanalmente, ya que se encuentra razonable:

«No creo que sea una asignatura con excesivo trabajo, además el trabajo está bien repartido y por el planteamiento de la asignatura se hace llevadero.»

La tercera categoría identificada tiene que ver con el hecho de que las actividades puntúen. Aunque no son requisito para aprobar (no hay nota mínima exigida), sí forman parte de la nota final y eso hace que se perciban como obligatorias.

«En teoría a todas las asignaturas hay que dedicarles un tiempo mínimo para el estudio. Si el estudio es obligatorio, por medio de unas prácticas o actividades previas a entregar, fuerzas al alumno a realizar este trabajo, sin tener que confiar en que estudie la teoría de forma tediosa una vez que llegue a su casa después de la clase. Me parece que el tiempo que se le tiene que dedicar a la asignatura es el necesario para seguir las clases, mientras adquieres los conocimientos para ir ampliando temario. »

Como la carga de trabajo está distribuida semanalmente a lo largo de todo el semestre y las actividades que se realizan se consideran productivas para el aprendizaje, a final del curso se sienten mejor preparados para el examen:

«Creo que en esta asignatura hay que trabajar mucho para poder llevarla al día, pero luego lo compensa de forma que si has hecho todas las prácticas, correcciones, etc., vas mucho más preparado para el examen, así que al final de curso que es donde hay más carga de trabajo no hace falta dedicarle tantas horas.»

5.2. Sobre la percepción en el aprendizaje

Las preguntas de la encuesta que nos interesan ahora son: «Preparando tú la clase con antelación y resolviendo ejercicios en el aula ¿crees que has aprendido

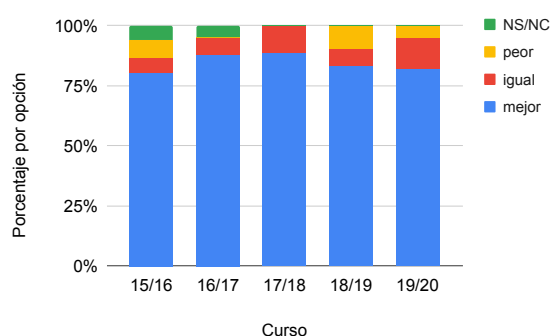


Figura 3: Percepción sobre el aprendizaje comparando la clase magistral con la clase invertida aplicada en la asignatura.

peor, mejor o igual que si hubiéramos seguido la metodología de la clase magistral, donde en clase de teoría explica los conceptos el profesor y los ejercicios se resuelven después?» donde se presentan las opciones: mejor, igual, peor; y la pregunta abierta: «Por favor, justifica tu respuesta».

En la Figura 3 se observa que el porcentaje de alumnos que dicen haber aprendido mejor se mantiene por encima del 80 % a lo largo de todos los cursos. El porcentaje de los que dicen haber aprendido peor está siempre por debajo del 10 % habiendo un par de cursos en que ha sido el 0 %.

En el análisis de contenido de las justificaciones aportadas en las encuestas de los últimos cinco cursos se identifican cuatro categorías relativas a las características de la clase invertida que favorecen el aprendizaje. Una primera categoría es la que se refiere a lo que aporta la entrega previa: se llega a clase con ciertas nociones, por lo que en clase se acaba de entender, se aclaran las dudas y es más fácil no perderse:

«Yo creo que mejor, porque de primeras implica que debo hacer un esfuerzo en llegar al menos a entender mínimamente los ejercicios para poder entregarlos antes de cada clase.»

«Es mejor, esto es debido a que te obligas a entender previamente algo, las dudas generadas ya las tienes y puedes preguntarlas en la clase y que te las expliquen y así incluso alguna persona con la misma duda la puede aclarar.»

La segunda categoría se refiere al uso que se hace del tiempo de clase para resolver dudas, hacer más ejercicios y reforzar los conceptos trabajados:

«En la clase había mucho más tiempo para poner en práctica la teoría, hacer ejercicios y preguntar dudas, lo cual es mil veces mejor que escuchar a un profesor explicando la teoría.»

La tercera categoría hace referencia al feedback y cómo permite aprender de los errores, incluso cuando puede resultar un tanto frustrante:

«Ya que de esta forma es más fácil aprender de nuestros propios errores.»

«Aunque es un poco más frustrante, aprendo más si antes me he dado de bruces con el problema y he fallado ahí.»

La cuarta categoría es relativa a los materiales que se proporcionan para trabajar la asignatura:

«En este caso ha ido muy bien gracias a todo el material complementario que nos dabais. Tanto los vídeos, como los ejemplos, como los apuntes ayudan mucho.»

Incluso prefiriendo la clase magistral, el hecho de disponer de buenos materiales facilita el trabajo con la clase invertida y se encuentra que se aprende mejor así:

«Suelo preferir la metodología de clase magistral ya que me cuesta un poco ser autodidacta, pero si cuentas con unos buenos apuntes como los que nos proporcionáis vosotros, la información se encuentra muy fácil y más eficiente que dando la clase y llevándote las dudas a la siguiente clase.»

5.3. Sobre la adaptación a la docencia no presencial

La última pregunta de la encuesta del curso 19/20 estaba relacionada con el cambio a la docencia no presencial debido al estado de alarma, que sucedió a mitad del semestre. La pregunta es la siguiente: «¿Qué valoración haces sobre la manera en que se ha hecho la adaptación de la docencia y la evaluación con motivo de la suspensión de la docencia presencial (estado de alarma)?».

Todos los alumnos que completaron la encuesta contestaron haciendo una valoración positiva sobre la adaptación realizada. Las justificaciones aportadas se han organizado en cuatro categorías:

- Continuar con las clases sincrónicas: «Ha sido la mejor clase sin duda, muchas de las asignaturas se resignaban a subir vídeos o pdfs que dejaban todo a tu entendimiento. El hacer las clases en directo las hace mucho más productivas por su interactividad. Os felicito por haber adaptado tan bien la asignatura.».
- Adaptar la evaluación: «Pienso que es una de las asignaturas que mejor se ha adaptado. Con otras asignaturas nos sentimos bastante asfixiados respecto a su forma de evaluar, por tanto se agradece que en esta se haya tenido en cuenta la situación que estamos viviendo.»
- Preocuparse por los alumnos y tener en cuenta su opinión y necesidades: «Me parece perfecta y que se ha entendido muchísimo la situación, además de escuchar a los alumnos, aplicar sus consejos y proponer cambios a los alumnos en lugar de aplicarlos sin preguntarnos previamente. Además el

cambio ha sido ínfimo, pues la docencia me ha parecido muy similar a la presencial, salvo que se ha tenido que reducir un poco la carga en la clase de teoría pues a todos nos costaba un poco dada la situación.»

- Capacidad de adaptación de la metodología a la docencia no presencial: «La estructura de la asignatura ha permitido que el cambio a docencia online no se haya casi notado, lo cuál ha sido muy de agradecer.»

6. Discusión

En este apartado retomamos las preguntas de investigación planteadas para discutir los resultados anteriormente presentados.

1. ¿Cómo repercute la clase invertida en la organización del tiempo de dedicación de nuestros estudiantes? La dedicación a sus estudios por parte de los alumnos no se ve afectada negativamente por emplear la clase invertida. Ello es debido a que la carga de trabajo de la asignatura está distribuida semanalmente durante todo el semestre y el tiempo que se ha de dedicar cada semana es el que corresponde al trabajo no presencial establecido por el plan de estudios. La percepción del alumnado en cuanto a que las actividades no presenciales son esenciales e imprescindibles para el aprendizaje es un factor relevante para valorar positivamente la dedicación semanal. Consideran que el tiempo que se ha de dedicar cada semana es productivo y resulta rentable de cara a final de curso ya que hace falta menos tiempo para preparar el examen. El hecho de que las actividades no presenciales puntúen motiva a dedicar el tiempo requerido cada semana, lo que contribuye a llevar la asignatura al día.
2. ¿Qué aspectos de la clase invertida son los que favorecen el aprendizaje de nuestros estudiantes? Es fundamental hacer un buen diseño de las actividades previas, de manera que sean las necesarias y adecuadas para que los alumnos empiecen a aprender, puedan surgir dudas, incluso cometer errores habituales, pero sin exceder el tiempo semanal requerido. Es imprescindible también disponer de buenos materiales (libros, vídeos, etc.). En cuanto a las sesiones presenciales, es importante hacer un buen diseño de las actividades a llevar a cabo, de manera que los alumnos puedan resolver sus dudas, darse cuenta de sus errores y reforzar y profundizar los conceptos trabajados.
3. ¿Cómo ha afectado al desarrollo de la docencia de la asignatura el paso a la docencia no presencial sufrida a mitad del segundo semestre del cur-

so 19/20? La clase invertida que se lleva a cabo en la asignatura se adapta perfectamente a la docencia no presencial, realizando las sesiones de clase de manera síncrona, ya que son fundamentales para que el alumno tenga *feedback*. Al estar planificado el trabajo no presencial semanal desde el principio, el alumno sabe cuánto tiempo ha de ocupar y en qué tareas.

Si bien la clase invertida aporta ventajas importantes para el aprendizaje de los alumnos, hemos encontrado comentarios de alumnos que relatan otras experiencias en las que este modelo no ha funcionado bien:

«Este cuatrimestre ha sido el primero que ha introducido esta mecánica en un par de asignaturas. Es cierto que en esta asignatura ha funcionado muy bien y, sinceramente, creo que he aprendido mejor y más en esta asignatura que en cualquier otra que he cursado en todos mis estudios del grado. ¿Por qué digo esto? Porque sí, puede que esa mecánica mejore el aprendizaje pero otras asignaturas que siguen (o pretenden seguir) este tipo de mecánica no han funcionado tan bien, así que no tengo del todo claro si atribuirlo a esta nueva metodología.»

«Mira que a mí esto de estudiar nosotros antes y luego tú nos enseñas me parecía un método que EN MÍ no funcionaba porque ya lo entendía todo antes, pero ahora me he dado cuenta de que sí funciona ya que está bien aplicado y en otras asignaturas no lo está.»

Como comentábamos al inicio de este trabajo, en conversaciones informales con profesorado que no ha tenido éxito invirtiendo sus clases hemos observado que siempre faltaba alguno de los aspectos clave de los que indica Marqués [4]:

- Las actividades previas han de aportar contenido importante para la clase.
- Los estudiantes deben percibir que las actividades son realizables, aunque deben plantear cierto reto.
- Hay que dar algún incentivo a los estudiantes por el hecho de preparar la clase.
- Hay que planificar con detalle cómo se llevará a cabo la clase, utilizando metodologías activas y conectando con la actividad de preparación previa.

Esto coincide con las opiniones recogidas en la encuesta, donde vemos que los mismos alumnos tienen la percepción de que no es la asignatura o el profesorado, sino la implementación del modelo lo que hace que funcione.

7. Conclusiones

La literatura sobre la clase invertida, nuestra larga experiencia aplicándola y las percepciones de nuestros estudiantes coinciden en que lo que hace que funcione

la clase invertida: diseñarla e implementarla adecuadamente. Sin intención de ser exhaustivos, esto implica fundamentalmente:

- Disponer de buenos materiales para proporcionar a los estudiantes. En nuestro caso, los alumnos valoran que el libro está bien estructurado, tiene buenas explicaciones y es fácil de seguir; los vídeos son útiles para aclarar dudas; las actividades y los boletines de prácticas y de problemas tienen una buena progresión y están alineados con lo que se ha de aprender.
- La preparación de la clase se debe hacer en base a actividades que resulten esenciales para el aprendizaje. Para ello, deben diseñarse de manera que el alumno dedique más tiempo a pensar activamente sobre el objeto de aprendizaje, de una manera análoga a como posteriormente será evaluado, y no tanto a perderse en detalles superfluos que solo pretendan dar contexto o estructura a la actividad [5].
- La planificación semanal de las actividades no presenciales debe realizarse de manera ajustada al tiempo asignado la asignatura y debe tener un peso en la calificación final.
- La clase presencial debe conectar con las actividades previas y servir para resolver las dudas que traen los estudiantes, aprender del error, profundizar, etc. Un aspecto clave del diseño de cualquier metodología docente es que las actividades de aprendizaje y de evaluación estén alineadas con los resultados de aprendizaje [2].

Por último cabe decir que la clase invertida ha demostrado ser una metodología muy robusta que permite su adaptación rápida a situaciones de confinamiento como la provocada por la pandemia, tal y como queda reflejado claramente en las opiniones de los estudiantes.

Referencias

- [1] Gökçe Akçayır and Murat Akçayır. The flipped classroom: A review of its advantages and challenges. *Computers & Education*, 126:334–345, 2018.
- [2] John Biggs and Catherine Tang. *Teaching for quality learning at university*. Open University Press. McGraw-Hill Education, 2011.
- [3] Casey E Davenport. Evolution in student perceptions of a flipped classroom in a computer programming course. *Journal of College Science Teaching*, 47(4), 2018.
- [4] Mercedes Marqués. Qué hay detrás de la clase al revés (flipped classroom). *Actas de las XXII JENUi*, pages 77–84, 2016.
- [5] Héctor Ruiz Martín. *¿Cómo aprendemos? Una aproximación científica al aprendizaje y la enseñanza*, volume 1. Graó, 2020.
- [6] Gregory S. Mason, Teodora Rutar Shuman, and Kathleen E. Cook. Comparing the effectiveness of an inverted classroom to a traditional classroom in an upper-division engineering course. *IEEE transactions on education*, 56(4):430–435, 2013.
- [7] Philipp Mayring. Qualitative content analysis. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research*, 1(2):Art. 20, 2000.
- [8] Kathleen K. Molnar. What effect does flipping the classroom have on undergraduate student perceptions and grades? *Education and Information Technologies*, 22(6):2741–2765, 2017.
- [9] Luis R. Murillo-Zamorano, José Ángel López Sánchez, and Ana Luisa Godoy-Caballero. How the flipped classroom affects knowledge, skills, and engagement in higher education: Effects on students' satisfaction. *Computers & Education*, 141:103608, 2019.
- [10] Jacqueline O'Flaherty and Craig Phillips. The use of flipped classrooms in higher education: A scoping review. *The internet and higher education*, 25:85–95, 2015.
- [11] Jarutas Pattanaphanchai. An investigation of students' learning achievement and perception using flipped classroom in an introductory programming course: A case study of thailand higher education. *Journal of University Teaching and Learning Practice*, 16(5):4, 2019.
- [12] Silvia Rueda, J. Ignacio Panach, Miriam Gil, Sergio Casas, and Mariano Pérez. Cuando la clase invertida no es suficiente: estrategias para motivar al alumnado en ingeniería del software. *Actas de las Jenui*, 5:125–132, 2020.
- [13] Javier Troya, Sergio Segura, José A. Parejo, Adela del Río-Ortega, Antonio Gámez-Díaz, and Alfonso E. Márquez-Chamorro. Invirtiendo las clases de laboratorio en ingeniería informática: Un enfoque ágil. *Actas JENUi*, 4:15–22, 2019.
- [14] Yanqing Wang, Xinzhuo Huang, Christian Dieter Schunn, Yan Zou, and Wenguo Ai. Redesigning flipped classrooms: a learning model and its effects on student perceptions. *Higher Education*, 78(4):711–728, 2019.
- [15] Zamzami Zainuddin. Students' learning performance and perceived motivation in gamified flipped-class instruction. *Computers & education*, 126:75–88, 2018.

¿Quién tropieza dos veces con la misma piedra en una asignatura inicial de programación?

Maria-Jesús Marco-Galindo
Universitat Oberta de Catalunya
Barcelona, Spain
mmarcog@uoc.edu

David García-Solórzano
Universitat Oberta de Catalunya
Barcelona, Spain
dgarciaso@uoc.edu

Julià Minguillón
Universitat Oberta de Catalunya
Barcelona, Spain
jminguillona@uoc.edu

Teresa Sancho-Vinuesa
Universitat Oberta de Catalunya
Barcelona, Spain
tsancho@uoc.edu

Resumen

Aprender a programar no es sencillo y es sabido que un porcentaje elevado de estudiantes fracasa en su primer intento. De hecho, en asignaturas introductorias de programación hay muchos estudiantes repetidores. La experiencia de “Fundamentos de Programación” muestra que hay dos tipos de estudiantes repetidores: los que no hacen nada o casi nada y abandonan, y los que lo intentan, pero suspenden. En este trabajo presentamos un análisis de los resultados de los estudiantes repetidores en esta asignatura y su participación en las actividades propuestas, con la finalidad de diseñar una intervención que les ofrezca un itinerario formativo adaptado a su situación y necesidades, y que tenga en cuenta sus logros previos. Basándonos en la literatura y el análisis realizado, identificamos los factores que explican el fracaso de los estudiantes en su primer intento.

Abstract

Learning to code is not easy and a high percentage of students fail on their first attempt. In fact, in introductory programming courses there are many students who are repeating the course. The experience of “Programming Foundations” shows that there are two types of students repeating the course: those who do nothing or almost nothing and drop out, and those who try but fail. In this paper we present an analysis of the results of the repeating students repeating and their participation in the proposed activities, with the ultimate objective of designing an intervention that provides them with a learning itinerary adapted to their situation and needs, taking into account their previous achievements. Based on the literature and the analysis carried out, we enumerate the factors that explain students’ failure in their first attempt.

Palabras clave

Curso inicial de programación, estudiantes repetidores, análisis del rendimiento, diseño de un cuestionario, abandono temprano.

1. Introducción

Diversos artículos [11] ponen de manifiesto que muchos estudiantes tienen dificultades para dominar los contenidos y competencias de las asignaturas introductorias de programación. Esto conlleva que las tasas de abandono de estas asignaturas sean altas –alrededor del 28-33 % [1]– y los índices de superación, bajos [26].

Una de las dificultades que presentan estas asignaturas es, como apunta Robins [18], el hecho de que los conceptos están fuertemente relacionados entre sí, de manera que los introducidos más tarde dependen de aquellos adquiridos anteriormente. En consecuencia, la acumulación de «lagunas» impide a muchos estudiantes progresar en este tipo de aprendizajes. En este sentido, varias investigaciones [23] evidencian que los estudiantes pueden necesitar más tiempo del esperado para adquirir los conocimientos de programación. De hecho, algunos se quejan de que, en el tramo final de la primera asignatura de programación, el ritmo es demasiado rápido [27]. En este sentido, los repetidores pueden ser aquellos estudiantes que implícitamente han manifestado necesitar más tiempo para asimilar los conceptos y competencias de una asignatura introductoria de programación. El objetivo final es entender las dificultades que experimentaron los repetidores, para poder diseñar una intervención que les ayude a superar la asignatura cuando la repiten. Con este propósito nos proponemos, mediante un análisis exploratorio preliminar, delimitar cuáles son los factores que llevaron a los estudiantes repetidores a no superar la asignatura

en su primer semestre.

Este artículo se organiza de la siguiente manera: el apartado 2 presenta un breve estado de la cuestión sobre los factores que influyen en el abandono en asignaturas de programación y el perfil de los estudiantes repetidores. El apartado 3 describe la metodología usada en este trabajo. El apartado 4 presenta el análisis realizado y los resultados obtenidos. En el apartado 5 se discuten los resultados y se propone una categorización de los factores identificados como relevantes para diseñar una intervención con los estudiantes repetidores. Finalmente, el apartado 6 presenta las conclusiones y las líneas de investigación futuras de este estudio.

2. Asignaturas de programación

Las asignaturas introductorias de programación tienen, en general, altas tasas de abandono y un índice de superación menor que el de otras asignaturas de primer curso [26]. Además, la literatura muestra que los resultados académicos de estas asignaturas se comportan de manera casi bimodal, con un grupo de estudiantes (llamado *effective novices*) que progresa adecuadamente obteniendo buenos resultados y otro grupo de estudiantes (llamado *ineffective novices*) que suspenden tras hacer, muchos de ellos, un esfuerzo extraordinario durante el semestre [19]. Todo esto supone que el porcentaje de estudiantes repetidores en asignaturas de programación no sea negligible. Para hacer frente a esta problemática y poder crear estrategias específicas para los estudiantes repetidores, es importante, por un lado, identificar qué características definen a estos estudiantes y, por otro, qué factores favorecen el abandono en estas asignaturas.

2.1. Motivos de abandono

Estudios como el de Kinnunen y Malmi [8] y el de Petersen et al. [16] concluyen que los dos motivos principales que llevan a los estudiantes a abandonar una asignatura inicial de programación son la falta de tiempo y motivación. Respecto a la escasez de tiempo, existen factores tales como:

- Priorización de otras asignaturas.
- Eventos ineludibles durante el semestre.
- Actividades que exigen más tiempo del esperado.
- Falta de habilidades para gestionar el tiempo.
- Sensación de no poder seguir el ritmo si uno se queda rezagado en un tema.

En cuanto a la falta de motivación, se identifican los factores siguientes:

- Desequilibrio entre la carga de trabajo y el aprendizaje obtenido.

- Frustración al perder demasiado tiempo buscando y corrigiendo errores en el código.
- Ayuda insuficiente o a destiempo por parte del profesorado.

En [16] también se revisan aspectos que influyen en el bajo rendimiento e incluso abandono de los estudiantes. Por ejemplo, se destaca que muchos estudiantes perciben que las estrategias de estudio que utilizan en otras asignaturas –más orientadas a revisar los materiales docentes– no sirven para las de programación. En esta línea, una comparativa entre estudiantes con alto y bajo rendimiento [10] identificó que estos últimos tienden a memorizar soluciones concretas de código, en lugar de entender los conceptos subyacentes.

Asimismo, el diseño de algunas actividades formativas –como puede ser un test con múltiples intentos– puede dar una sensación equivocada a los estudiantes sobre su grado de adquisición de los conceptos y competencias [16]. Del mismo modo, algunos trabajos en parejas pueden suponer que el estudiante, sin ser consciente, tenga la sensación de que es capaz de terminarla individualmente, cuando en verdad sin la ayuda de su compañero no la hubiera finalizado creándole una falsa percepción de sí mismo, de la que no es consciente hasta realizar a una actividad individual y evaluable.

2.2. Perfil de los estudiantes repetidores

Trabajos como [21] señalan que muchos repetidores no tienen una gran motivación por la programación, ni siquiera por la informática, pero que escogieron la titulación, entre otras razones, por tener una mejor salida profesional o no tener nota de acceso suficiente para realizar la carrera deseada. El mismo trabajo [21] también muestra que la mayoría de los repetidores no quieren dedicarse a trabajos directamente relacionados con la programación. Este conocimiento puede ser utilizado para, por ejemplo, contextualizar los ejercicios en diferentes ámbitos (p.e. tales como financieros, artísticos, etc.), lo cual hace que los estudiantes perciban los conceptos como más relevantes y la tasa de éxito sea mayor [6]. Asimismo, gracias a dicha contextualización, se puede llegar a motivar al estudiante haciéndole ver que los conocimientos de programación pueden ser una ventaja profesional si se dedican a estos ámbitos. Un factor indudablemente a tener en cuenta y analizado en diferentes estudios como [5] es la autoestima, menor entre los repetidores, probablemente a causa de haber suspendido en el semestre anterior.

La actividad en las primeras semanas de curso puede ser percibida por los repetidores como una pérdida de tiempo. Así lo demuestran los resultados obtenidos por [4] a través de una herramienta de monitorización, donde la participación en los ejercicios iniciales fue menor por parte de los repetidores en comparación con los es-

tudiantes que cursaban la asignatura por primera vez. En cambio, según [28], el uso de la programación por parejas (*pair programming*) durante la realización de las actividades en las semanas iniciales favorece la motivación de los repetidores. Una de las posibles razones es el hecho de poder debatir con otro estudiante de nivel similar, por ejemplo, diferentes soluciones para el mismo ejercicio. En la misma línea, la investigación de Sheard y Hagan [21] también destaca que la utilidad del trabajo en grupo era mayor para los estudiantes repetidores que para los nuevos. Por otro lado, muchos trabajos como [22] analizan qué contenidos resultan más difíciles para los estudiantes. La identificación de dichos contenidos puede resultar de mucho interés en el diseño de actividades específicas para los estudiantes que repiten la asignatura.

3. Metodología

Como parte de un proyecto de investigación más amplio sobre el diseño y análisis de intervenciones en asignaturas introductorias a la programación [14], en este trabajo nos planteamos responder las preguntas de investigación siguientes:

- RQ1: ¿Existen diferencias en los resultados obtenidos por los estudiantes repetidores en función de lo que hicieron en la convocatoria anterior?
- RQ2: ¿Existen características del perfil de los estudiantes repetidores que permitan explicar posibles diferencias en sus resultados?

A partir del análisis de los resultados de los repetidores y una revisión de la literatura, seguidamente se identificarán los factores sobre los cuales se puede incidir para mejorar la experiencia de los repetidores.

3.1. Contexto

La asignatura “Fundamentos de programación” es obligatoria en los grados de Ingeniería Informática e Ingeniería de Tecnologías de Telecomunicación de la UOC, pero también un complemento de formación en algunos másteres especializados, y asignatura libre dentro de otros programas de la universidad. Cuenta con un perfil de estudiantes muy heterogéneo.

Como asignatura introductoria, introduce los principios básicos de la programación a partir de la algorítmica lo que combina con la práctica de ejercicios sencillos en lenguaje C que progresivamente van avanzando en dificultad. La evaluación es continua y se basa en una secuencia de actividades donde se combinan ejercicios de diseño algorítmico y de programación en lenguaje C. Las actividades son opcionales, así que el estudiante decide cuántas y cuáles realiza, teniendo en cuenta que cada una de ellas representa una parte de

la calificación final de la evaluación continua. Las actividades se acompañan de los recursos necesarios para resolverla: los contenidos teóricos de algorítmica, indicaciones y ejemplos de codificación en C y una máquina virtual con el entorno de desarrollo *Codelite* para programar. Cada semana se entrega un ejercicio y a continuación se publica su solución. Una vez completados los cuatro ejercicios que forman una actividad, se recibe la calificación y también una retroalimentación personalizada del profesor. Es, pues, una evaluación formativa.

El profesor utiliza el tablón del aula virtual para comunicar cualquier cuestión relativa al desarrollo de las actividades. Las dudas de los estudiantes, en cambio, se comparten y resuelven desde el foro del aula o a través del buzón personal del profesor y estudiante. Se trata de un espacio donde se espera que los estudiantes participen compartiendo sus dudas y creando conocimiento de forma colaborativa.

3.2. Datos

Los datos para el análisis se obtuvieron del *Learning Record Store* (LRS) institucional [15], el cual almacena todas las evidencias relacionadas con las actividades que realiza el estudiante, desde su matrícula hasta las notas finales obtenidas en las asignaturas matriculadas, pasando por las actividades parciales entregadas. En el estudio se incluyen todos aquellos estudiantes que se han matriculado de la asignatura de Fundamentos de Programación al menos una vez desde el semestre 2017/2 hasta el semestre 2019/1 (cuatro semestres), así como los que habían cursado la asignatura antes del semestre 2017/2 y la repiten al menos una vez en el semestre 2017/2 o posteriormente, un total de 1,206 estudiantes. Se trata, por lo tanto, de un estudio realizado con toda la población de interés. Cada semestre los estudiantes son diferentes y las diferentes cohortes (semestres) no son completamente equivalentes, pero teniendo en cuenta que el diseño de la asignatura es el mismo desde el semestre 2017/2, el análisis se realizó para todos los estudiantes en un solo conjunto de datos, en función de sus resultados del semestre anterior.

Para cada estudiante y cada semestre se dispone de los datos siguientes: sexo (Hombres: 976, 80.9 %; Mujeres: 230, 19.1 %); grupo de edad (E1 - hasta 20 años: 41, 3.4 %; E2 - 21-30 años: 538, 44.6 %; E3 - 31-40 años: 353, 29.3 %; E4 - 41 años o más: 274, 22.7 %); y si está cursando la asignatura dentro del grado de Ingeniería Informática o lo hace como complemento de otro programa (SÍ: 1016, 84.2 %; NO: 190, 15.8 %). También se dispone del número de asignaturas matriculadas y superadas simultáneamente (sin incluir la analizada en este trabajo) y de los resultados obtenidos en las actividades de evaluación continua.

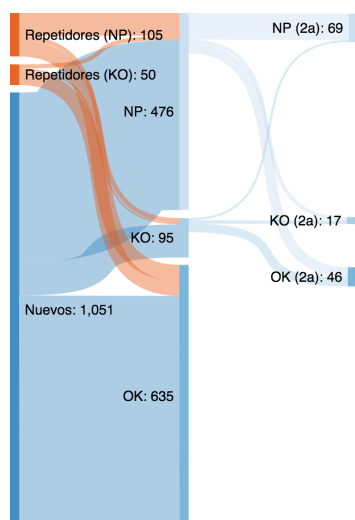


Figura 1: Resultados obtenidos por los estudiantes.

4. Resultados

En base a las preguntas de investigación se detalla a continuación el resultado de los análisis realizados.

4.1. Comportamiento de los estudiantes

Un primer análisis visual muestra que el comportamiento de los estudiantes depende de los resultados obtenidos anteriormente (OK = aprobado, KO = suspendido, NP = no presentado). La Figura 1 muestra resultado obtenido por los estudiantes en función del resultado anterior. Se puede observar que pocos estudiantes lo volvieron a intentar una segunda vez (69+17+46, 23.1 %) y que la mayor parte de ellos volvieron a suspender (69+17, 65.2 %). Estos primeros datos sugieren plantearse si el diseño actual de la asignatura es el más adecuado para los estudiantes repetidores.

De forma más detallada, el Cuadro 1 muestra el desglose de los estudiantes en función de los resultados que obtienen en cada uno de los intentos consecutivos. Desde el semestre 2017/2 (incluido), matricularon la asignatura por primera vez 1,051 estudiantes (Grupo G1), mientras que 155 la cursaron después de no haberla superado anteriormente, de los cuales 50 la suspendieron (G2) y 105 no se presentaron (G3). Distinguimos tres subgrupos para cada uno de ellos (los que aprueban (OK), los que suspenden (KO) y los que no se presentan (NP)), con el objetivo de analizar sus trayectorias. Así, tomando como referencia solamente los resultados de los estudiantes nuevos (G1), una primera inspección muestra que los estudiantes que no se habían presentado a la asignatura (G1-3) obtuvieron peores resultados que el resto. Pocos estudiantes lo intentaron una vez más, y además incluso obtuvieron peores resultados. Por otra parte, los estudiantes que habían

suspendido la asignatura (G1-2) se matricularon más y además obtuvieron mejores resultados, pero parece que ya no lo intentaron de nuevo si no superaron la asignatura en su segundo intento. Por lo que respecta a la rematrícula, de los 411 estudiantes que no se presentaron en su primer intento (G1-3), sólo 79 (19.2 %) lo volvieron a intentar una segunda vez en el periodo analizado. En cambio, de los 80 estudiantes que suspendieron (G1-2), un total de 29 (36.4 %) lo volvieron a intentar, un porcentaje aún bajo pero considerablemente superior al anterior.

Otro dato que muestra el Cuadro 1 es el número de actividades entregadas (A_4 , para las cuatro primeras, y A_8 para las ocho primeras). Los estudiantes que lo intentaron y suspendieron realizaron casi las mismas actividades que los que aprobaron, mientras que los que no se presentaron realizaron muy pocas de las cuatro primeras, y casi ninguna de las cuatro siguientes, después de la retroalimentación del profesor. Este hecho era también cierto para los estudiantes que ya venían de un intento anterior (G2 y G3), aunque en el caso de estudiantes no presentados el número de actividades realizadas era aún menor.

Los resultados obtenidos nos permiten responder la primera pregunta de investigación (RQ1). Efectivamente, los datos muestran que los repetidores que suspenden la asignatura tienen una segunda oportunidad que pueden aprovechar, y de hecho un porcentaje elevado así lo hizo. Puede ser que necesiten más tiempo (que un semestre) para superarla y que quizás tuvieron problemas al final, no al principio del semestre. Por otra parte, los estudiantes que no se presentaron en la convocatoria anterior se encontraron de nuevo con los mismos obstáculos y volvieron a dejar la asignatura, haciendo buena la frase (desde la perspectiva del equipo docente) «la estupidez es hacer lo mismo y esperar resultados diferentes». En este contexto, es necesario proporcionar alguna alternativa (formativa) para que estos estudiantes superen el obstáculo que les impide progresar y no repitan los mismos errores de nuevo.

4.2. Características de los repetidores

La segunda pregunta de investigación es si hay alguna característica de los estudiantes que nos permita explicar las diferencias encontradas en los resultados de los repetidores (RQ2). El Cuadro 2 muestra los indicadores seleccionados para los grupos más relevantes del Cuadro 1. Para cada grupo o subgrupo se muestra el porcentaje de hombres y mujeres, el desglose por grupo de edad, el porcentaje de estudiantes que cursan el grado de Ingeniería Informática, el número de asignaturas matriculadas simultáneamente y el número de asignaturas superadas, en ambos casos sin incluir la asignatura en cuestión. También se muestra el porcentaje de estudiantes que realizan todas las actividades

Grupo	Origen	1a vez	2a vez	N (%)	A ₄	A ₈	Rematrícula
G1	Nuevos			1,051 (87.2 %)	3.10	5.64	108 (10.3 %)
G1-1		OK		560 (53.3 %)	3.91	7.70	—
G1-2		KO		80 (7.6 %)	3.56	6.54	29 (36.3 %)
G1-2-1			OK	17 (58.6 %)	3.53	6.94	—
G1-2-2			KO	7 (24.1 %)	3.43	6.29	—
G1-2-3			NP	5 (17.3 %)	1.60	3.00	—
G1-3		NP		411 (39.1 %)	1.91	2.66	79 (19.2 %)
G1-3-1			OK	20 (25.3 %)	3.70	7.15	—
G1-3-2			KO	8 (10.1 %)	3.63	6.75	—
G1-3-3			NP	51 (64.6 %)	1.69	2.29	—
G2	KO			50 (4.1 %)	3.58	6.76	5 (10.0 %)
G2-1		OK		36 (72.0 %)	3.89	7.64	—
G2-2		KO		4 (8.0 %)	3.75	7.5	3 (75.0 %)
G2-2-1			OK	2 (66.7 %)	4.00	8.00	—
G2-2-2			KO	1 (33.3 %)	4.00	8.00	—
G2-2-3			NP	0 (0.0 %)	—	—	—
G2-3		NP		10 (20.0 %)	2.40	3.30	2 (20.0 %)
G2-3-1			OK	2 (100.0 %)	3.50	7.50	—
G2-3-2			KO	0 (0.0 %)	—	—	—
G2-3-3			NP	0 (0.0 %)	—	—	—
G3	NP			105 (8.7 %)	2.58	4.36	19 (18.1 %)
G3-1		OK		39 (37.1 %)	3.72	7.28	—
G3-2		KO		11 (10.5 %)	3.27	5.45	2 (18.2 %)
G3-2-1			OK	2 (100.0 %)	4.00	7.50	—
G3-2-2			KO	0 (0.0 %)	—	—	—
G3-2-3			NP	0 (0.0 %)	—	—	—
G3-3		NP		55 (52.4 %)	1.64	2.07	17 (30.9 %)
G3-3-1			OK	3 (17.6 %)	3.00	7.00	—
G3-3-2			KO	1 (5.9 %)	4.00	7.00	—
G3-3-3			NP	13 (76.5 %)	1.62	1.85	—

Cuadro 1: Estudiantes según su origen y resultados académicos en semestres sucesivos.

propuestas y el índice (mediana) de la primera actividad que no entregan, un posible indicador de desconexión del ritmo de actividades propuesto.

Se puede observar que respecto al grupo de estudiantes que se matriculan de la asignatura por primera vez (G1), el grupo de los estudiantes que suspendieron la asignatura (G1-2) tenía una proporción ligeramente mayor de mujeres, así como de estudiantes más jóvenes, y de estudiantes que no estaban haciendo el grado de Ingeniería Informática. También estaban ligeramente matriculados de más asignaturas. Por otra parte, los estudiantes que aprobaron la asignatura (G1-1) obtuvieron mucho mejor rendimiento en las otras asignaturas de las que se habían matriculado en el mismo semestre, a diferencia de los que lo intentaron y suspendieron y los que no se presentaron, especialmente en este último caso. Esto puede indicar que los estudiantes que dejaron la asignatura podían estar dejando todas las asignaturas de las que se habían matriculado, es decir, que abandonaron los estudios, lo cual explica que no sea posible recuperarlos y que el indicador de rematrícula sea tan bajo. Además, se observa que los estudiantes que obtuvieron un no presentado (G1-3) son los que dejaron de entregar alguna actividad propuesta más pronto. Concretamente, fue en la segunda actividad, mientras que los que lo intentaron y suspen-

dieron o aprobaron no lo hicieron hasta la quinta o la sexta actividad respectivamente, bien entrado el curso, después de la retroalimentación recibida por parte de su profesor. El porcentaje de estudiantes que no se presentaron pero realizaron los ejercicios propuestos fue también considerablemente menor con respecto a los que suspendieron y los que aprobaron (6.7 % vs 51.3 % y 81.8 % respectivamente). De nuevo parece factible pensar que parte de los estudiantes que abandonaron la asignatura lo hicieron porque estaban abandonando sus estudios, en realidad, al inicio del semestre, casi sin esperar a la retroalimentación del profesor de los cuatro primeros ejercicios.

5. Discusión

A partir de los resultados observados, queda claro que los estudiantes que no superan la asignatura pueden clasificarse en dos grupos, los que lo intentan hasta bien entrado el curso y suspenden, y los que no lo intentan y dejan el curso en las primeras actividades propuestas. Los dos grupos están compuestos por estudiantes similares en sexo, edad, etc., pero son claramente diferentes respecto a los resultados obtenidos, respondiendo a la segunda pregunta (RQ2). Esto es consistente con los resultados descritos por [18] so-

Grupo	Sexo	Edad	Informática	Asignaturas	Todas	Índice
G1	H: 838 (79.7 %) M: 213 (20.3 %)	E1: 38 (3.6 %) E2: 474 (45.1 %) E3: 303 (28.8 %) E4: 236 (22.5 %)	NO: 179 (17.0 %) Sí: 872 (83.0 %)	MAT: 1.98 SUP: 1.61	50.1 %	3
G1-1	H: 454 (81.1 %) M: 106 (19.9 %)	E1: 19 (3.4 %) E2: 228 (40.7 %) E3: 172 (30.7 %) E4: 141 (25.2 %)	NO: 95 (17.0 %) Sí: 465 (83.0 %)	MAT: 2.00 SUP: 1.66	81.8 %	6
G1-2	H: 57 (71.3 %) M: 23 (28.7 %)	E1: 8 (10.0 %) E2: 43 (53.8 %) E3: 17 (23.2 %) E4: 12 (15.0 %)	NO: 18 (22.5 %) Sí: 62 (77.5 %)	MAT: 2.30 SUP: 0.99	51.3 %	5
G1-3	H: 327 (79.6 %) M: 84 (20.4 %)	E1: 11 (2.7 %) E2: 203 (49.4 %) E3: 114 (27.7 %) E4: 83 (20.2 %)	NO: 66 (16.1 %) Sí: 345 (83.9 %)	MAT: 1.90 SUP: 0.31	6.7 %	2

Cuadro 2: Características de los estudiantes en función de sus resultados académicos.

bre el perfil socio-demográfico de los estudiantes de programación. No obstante, hay un subgrupo de estudiantes que provienen de otros grados, más jóvenes y con mayor proporción de mujeres, que deben cursar la asignatura como complemento de formación y que suspenden más, aunque esto también implica que lo intentan más. El hecho de ofrecer la misma asignatura a todos los estudiantes independientemente de su perfil puede causar que algunos no se adapten al ritmo y tipo de actividades propuesto [3]. Esta circunstancia debería tenerse presente en el diseño de una intervención con los estudiantes repetidores. Por lo tanto, dicho diseño debe orientarse a discriminar entre estudiantes que abandonan y los que lo intentan y suspenden, con el objetivo de corregir factores seguramente diferentes.

5.1. Factores determinantes

En términos generales, Lee y Choi [9] identificaron tres grupos de factores relacionados con el abandono: (1) intrínsecos al *estudiante*, (2) intrínsecos a la *asignatura* e (3) intrínsecos al *entorno o contexto*. El Cuadro 3 incluye los elementos de la asignatura sobre los que se puede actuar así como otros factores relativos al estudiante, destacando aquellos trabajos de la literatura donde se describen y su tipología de acuerdo a la clasificación anterior. Se indican en cursiva los factores que no pueden variar entre un semestre y otro. No se han incluido los factores socio-demográficos del estudiante y otros relacionados con su situación académica que se pueden extraer del LRS institucional. Estos factores determinarán el contenido de un cuestionario que se enviará a los estudiantes repetidores.

Cada uno de los factores descritos en el Cuadro 3 deberá desarrollarse en forma de preguntas de un cuestionario orientado a conocer la percepción de los estudiantes repetidores sobre los motivos por los que suspendieron la asignatura. Por ejemplo, con respecto al factor relativo al apoyo y guía del profesor, sería in-

terésante conocer: ¿El profesor responde mis dudas a tiempo? ¿La retroalimentación que recibo me ayuda a entender los errores que he cometido en las actividades? ¿Las indicaciones que da el profesor me ayudan a seguir el calendario de actividades? De este modo podría saberse si el mecanismo de retroalimentación actual es diferente para estudiantes que suspenden o no se presentan, y si es necesario que el profesor actúe diferente según del perfil del estudiante, por ejemplo.

6. Conclusiones

En este trabajo hemos presentado un análisis de los resultados de los estudiantes repetidores de una asignatura de Fundamentos de Programación, en función de los resultados obtenidos. Los repetidores muestran dos comportamientos claramente diferentes: los que en su primer semestre lo intentan casi hasta el final y suspenden, y los que abandonan de forma temprana. El rendimiento de estos dos grupos en su segundo intento es claramente diferente, aprobando mucho más los que lo intentaron y suspendieron la primera vez. Así pues, ofrecer la misma asignatura a los estudiantes que la repiten es casi sinónimo de volver a fracasar, por lo que es necesario plantearse algún tipo de intervención basada en su experiencia previa, perfil y necesidades.

De acuerdo con la literatura científica sobre el comportamiento de los estudiantes repetidores y los problemas inherentes a la enseñanza de la programación en cursos iniciales, se han identificado algunos de los factores más relevantes que podrían explicar dichas diferencias, y que deberían servir para recoger más datos sobre los estudiantes y su situación en el momento de cursar la asignatura, con el objetivo de diseñar una intervención. Partiendo de los resultados de esta investigación exploratoria, el trabajo futuro pasa por el diseño, implementación y evaluación de una intervención que mejore el resultado de los repetidores en su segun-

Factor	Descripción	Tipo
Motivación	Baja motivación del estudiante, éste no busca información adicional y/o no intenta superar los retos que se le presentan [8, 10, 7].	1
Creencias y actitudes	Los estudiantes asumen que abandonar una asignatura de programación es normal, o que aprender programación es muy difícil [16, 17].	1
Gestión del tiempo	Los estudiantes carecen de habilidades de autorregulación y planificación [8, 10].	1
Estrategia de aprendizaje	Los estudiantes utilizan técnicas de estudio inapropiadas para la programación (p.e. memorización) [10, 16].	1
Dificultad	Los contenidos de la asignatura son complejos y/o abarcan demasiados conceptos [12, 22, 24].	2
Materiales teóricos	Los recursos son claros, suficientes y/o están bien organizados y actualizados [19].	2
Lenguaje y entorno de programación	El lenguaje de programación no es apropiado para aprender y/o el entorno de programación no es sencillo de instalar y utilizar [20].	2
Tipología de actividades	Hay suficiente variedad de actividades y éstas están contextualizadas y ayudan a aprender a programar de manera progresiva [8, 16].	2
Planteamiento de las actividades	Los enunciados de las actividades son claros y comprensibles, los recursos y materiales disponibles son suficientes para resolverlas y/o la cara de trabajo es adecuada [13].	2
Calendario de actividades	El número y ritmo de las actividades es adecuado. [16]	2
Apoyo y guía del profesor	La ayuda y la retroalimentación del docente no es adecuada y/o llega tarde [8, 16].	2
Presencia social	Los canales de comunicación con profesores y entre compañeros son adecuados para favorecer un sentido de pertinencia al grupo [25].	2
Conciliación	Incompatibilidad con responsabilidades laborales y/o familiares [2].	3
Imprevistos	Circunstancias sobrevenidas como enfermedades o cambios laborales [16].	3
Entorno de aprendizaje	Problemas de disponibilidad y accesibilidad al campus virtual en el caso de estudios en línea y/o dificultades para manejarse con el entorno [2].	3

Cuadro 3: Factores determinantes para el diseño de una intervención.

do intento. Para ello, se propone empezar por diseñar un cuestionario que permita conocer la perspectiva de los estudiantes repetidores en relación a su proceso de aprendizaje en la asignatura iniciado en su primera matrícula. El análisis de los datos recogidos nos permitirá definir una intervención específica, adecuada a sus necesidades, que mejore su experiencia en su segundo intento de superar la asignatura. Por descontado, y de acuerdo con el paradigma de investigación basada en el diseño, deberá analizarse dicha experiencia para introducir las mejoras necesarias en una segunda iteración.

Referencias

- [1] Jens Bennedsen y Michael E. Caspersen. Failure rates in introductory programming: 12 years later. *ACM Inroads*, 10(2):30-36, abril de 2019.
- [2] Janice Catterall, Janelle Davis y col. Supporting new students from vocational education and training: finding a reusable solution to address recurring learning difficulties in e-learning. *Australasian Journal of Educational Technology*, 29(5), 2013.
- [3] Jessica Q. Dawson, Meghan Allen, Alice Campbell y Anasazi Valair. Designing an introductory programming course to improve non-majors' experiences. En *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, páginas 26-31, 2018.
- [4] Nuno Gil Fonseca, Luis Macedo y António José Mendes. Codeinsights: monitoring programming students' progress. En *Proceedings of the 17th International Conference on Computer Systems and Technologies*, CompSysTech '16, páginas 375-382, Palermo, Italy. ACM, 2016.
- [5] Anabela Jesus Gomes, Alvaro Nuno Santos y António José Mendes. A study on students' behaviours and attitudes towards learning to program. En *Proceedings of the 17th ACM Annual Conference on Innovation and Technology in Computer Science Education*, ITiCSE '12, páginas 132-137, Haifa, Israel. ACM, 2012.
- [6] Mark Guzdial y Allison Elliott Tew. Imagining inauthentic legitimate peripheral participation: an instructional design approach for motivating computing education. En *Proceedings of the Second International Workshop on Computing Education Research*, ICER'06, páginas 51-58, Canterbury, United Kingdom. Association for Computing Machinery, 2006.
- [7] Geetha Kanaparan, Rowena Cullen, David Mason y col. Effect of self-efficacy and emotional engagement on introductory programming students. *Australasian Journal of Information Systems*, 23, 2019.
- [8] Päivi Kinnunen y Lauri Malmi. Why students drop out cs1 course? En *Proceedings of the second international workshop on Computing education research*, páginas 97-108, 2006.
- [9] Youngju Lee y Jaeho Choi. A review of online course dropout research: implications for practice and future research. *Educational Technology Research and Development*, 59(5):593-618, 2011.

- [10] Soohyun Nam Liao, Sander Valstar, Kevin Thai, Christine Alvarado, Daniel Zingaro, William G. Griswold y Leo Porter. *Behaviors of higher and lower performing students in CS1*. En *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*. Association for Computing Machinery, New York, NY, USA, 2019, páginas 196-202.
- [11] Raymond Lister, Elizabeth S. Adams, Sue Fitzgerald, William Fone, John Hamer, Morten Lindholm, Robert McCartney, Jan Erik Moström, Kate Sanders, Otto Seppälä, Beth Simon y Lynda Thomas. A multi-national study of reading and tracing skills in novice programmers. *SIGCSE Bull.*, 36(4):119-150, 2004.
- [12] Andrew Luxton-Reilly. *Learning to program is easy*. En *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*. Association for Computing Machinery, New York, NY, USA, 2016, páginas 284-289.
- [13] Andrew Luxton-Reilly y Andrew Petersen. The compound nature of novice programming assessments. En *Proceedings of the Nineteenth Australasian Computing Education Conference*, páginas 26-35, 2017.
- [14] María-Jesús Marco-Galindo, Julià Minguillón y Teresa Sancho-Vinuesa. Análisis de la progresión de los estudiantes en una asignatura introductoria a la programación mediante redes bayesianas. *Actas de las Jornadas sobre Enseñanza Universitaria de la Informática (JENUi)*, 5:69-76, 2020.
- [15] Julià Minguillón, Jordi Conesa, María Elena Rodríguez y Francesc Santanach. Learning analytics in practice: providing e-learning researchers and practitioners with activity data. En *Frontiers of Cyberlearning*, páginas 145-167. Springer, 2018.
- [16] Andrew Petersen, Michelle Craig, Jennifer Campbell y Anya Taffioviich. Revisiting why students drop cs1. En *Proceedings of the 16th Koli Calling International Conference on Computing Education Research*, Koli Calling '16, páginas 71-80, Koli, Finland. Association for Computing Machinery, 2016.
- [17] Yizhou Qian y James Lehman. Students' misconceptions and other difficulties in introductory programming: a literature review. *ACM Trans. Comput. Educ.*, 18(1), octubre de 2017.
- [18] Anthony Robins. Learning edge momentum: a new account of outcomes in CS1. *Computer Science Education*, 20(1):37-71, 2010.
- [19] Anthony V. Robins. *Novice programmers and introductory programming*. En *The Cambridge Handbook of Computing Education Research*. Sally A. Fincher y Anthony V. Robins, edición. Cambridge Handbooks in Psychology. Cambridge University Press, 2019, páginas 327-376.
- [20] Sonia Pamplona Roche y Nelson Medinilla Martínez. Evaluación de entornos de programación para el aprendizaje. *JENUi 2011*:83, 2011.
- [21] Judy Sheard y Dianne Hagan. Our failing students: a study of a repeat group. En *Proceedings of the 6th Annual Conference on the Teaching of Computing: Changing the Delivery of Computer Science Education*, ITiCSE'98, páginas 223-227, Dublin, Ireland. ACM, 1998.
- [22] Juha Sorva. *Visual program simulation in introductory programming education*. Aalto University, 2012.
- [23] Donna Teague y Raymond Lister. Longitudinal think aloud study of a novice programmer. En *Proceedings of the Sixteenth Australasian Computing Education Conference - Volume 148*, ACE '14, páginas 41-50, Auckland, New Zealand. Australian Computer Society, Inc., 2014.
- [24] Arto Vihavainen, Jonne Airaksinen y Christopher Watson. A systematic review of approaches for teaching introductory programming and their influence on success. En *Proceedings of the 10th Annual Conference on International Computing Education Research*, ICER '14, páginas 19-26, Glasgow, Scotland, United Kingdom. ACM, 2014.
- [25] Joe Warren, Scott Rixner, John Greiner y Stephen Wong. Facilitating human interaction in an online programming course. En *Proc. of 45th ACM technical symposium on Computer science education*, páginas 665-670, 2014.
- [26] Christopher Watson y Frederick W.B. Li. Failure rates in introductory programming revisited. En *Proceedings of the 2014 Conference on Innovation in Computer Science Education*, ITiCSE '14, páginas 39-44, Uppsala, Sweden. Association for Computing Machinery, 2014.
- [27] Keith J. Whittington, Dianne P. Bills y Lawrence W. Hill. Implementation of alternative pacing in an introductory programming sequence. En *Proceedings of the 4th Conference on Information Technology Curriculum*, CITC4 '03, páginas 47-53, Lafayette, Indiana, USA. Association for Computing Machinery, 2003.
- [28] Krissi Wood, Dale Parsons, Joy Gasson y Patricia Haden. It's never too early: pair programming in CS1. En *Proceedings of the 15th Australasian Computing Education Conference*, volumen 136 de ACE'13, páginas 13-21, Adelaide, Australia. Australian Computer Society, Inc., 2013.

Un recorrido actualizado por la docencia de Sistemas Digitales en el Grado en Ingeniería Informática

Carlos Guerrero, Isaac Lera y Carlos Juiz
Departamento de Matemáticas e Informática
Universitat de les Illes Balears
07122 Palma de Mallorca
{carlos.guerrero,isaac.lera,cjuiz}@uib.es

Resumen

Este artículo presenta un recorrido por la docencia de los contenidos de sistemas digitales en el Grado de Ingeniería en Informática en las universidades españolas pertenecientes a la Conferencia de Directores y Decanos de Ingeniería Informática. Para ello se han analizado las competencias, los contenidos, la planificación y los recursos utilizados para las asignaturas directamente relacionadas con la docencia de este campo de conocimiento. Finalmente se presenta un análisis de los resultados obtenidos, dando una visión global de la situación nacional, y llevando a cabo una agrupación de universidades en función de los contenidos de las asignaturas utilizadas. Estos resultados son de interés para los docentes involucrados en estas asignaturas y para conocer la estandarización de los conocimientos sobre la materia, es decir, si el diseño curricular es o no homogéneo.

Abstract

This article presents a review of the contents related to Digital Systems in the Degree in Computer Engineering in the Spanish universities that belong to the *Conferencia de Directores y Decanos de Ingeniería Informática*. For this, the competences, contents, planning and resources used for the subjects, directly related to the teaching of this field, have been analyzed. Finally, an analysis of the results obtained is presented, giving a global vision, and carrying out a clustering of universities based on the contents of the subjects. These results are of interest to teachers involved in these subjects and to know the standardization of the knowledge area, i.e., whether or not the curricular design is homogeneous.

Palabras clave

Sistemas digitales, revisión sistemática, Grado en Ingeniería Informática.

1. Introducción

Los contenidos de sistemas digitales suelen ser el primer contacto de los estudiantes del Grado de Ingeniería en Informática con el campo de tecnología y organización de los computadores. En general, ofrecen las bases para que conozcan cómo se representa la información y cómo esta se transforma y opera a través del uso de circuitos digitales. Los conocimientos obtenidos establecerán las bases para que, posteriormente, los estudiantes sean capaces de entender la arquitectura básica de un computador.

Si vemos un computador desde un punto de vista de sus niveles de abstracción, los sistemas digitales y la lógica digital estarían situados de forma intermedia entre el nivel que corresponde a los componentes electrónicos analógicos y el nivel de la arquitectura básica de un computador.

Una asignatura que esté relacionada con el bloque temático de los sistemas digitales debería trabajar las competencias necesarias para el análisis y diseño de componentes lógicos digitales, que corresponden a los módulos básicos que forman un computador. En este proceso de análisis y diseño, los alumnos han de ser capaces de evaluar las alternativas disponibles y escoger la mejor para cada caso concreto.

Por tanto, será habitual encontrar este tipo de asignaturas situadas en el primer curso de unos estudios de Grado de Ingeniería en Informática. En los cursos iniciales, además del objetivo docente de adquisición de conocimientos y competencias, las asignaturas han de tener una importante componente motivadora. El abandono de estudiantes en los primeros cursos de las ingenierías es muy elevado, y por tanto, los docentes de las asignaturas de estos primeros cursos han de ser capaces de ir más allá de la presentación de conocimientos básicos, y ayudar al estudiante a ver el tipo de problemas a los que será capaz de enfrentarse una vez acabados los estudios.

En el campo de la investigación científica, es muy habitual encontrar revisiones sistemáticas de áreas par-

ticulares de investigación. Sin embargo, y desde el conocimiento de los autores, la realización de revisiones sobre la docencia universitaria de un bloque de conocimiento o temática no es tan habitual, a pesar de que este tipo de análisis puede ser de gran interés para profesores de las asignaturas de ese campo o para los propios gestores de los títulos.

En este trabajo se presenta un recorrido sobre la docencia de sistemas digitales en el ámbito del Grado de Ingeniería en Informática en las universidades españolas. Para ello, se han analizado los planes de estudios de dichas titulaciones en busca de las asignaturas relacionadas con los sistemas digitales. Una vez detectadas estas asignaturas, se han analizado sus guías docentes en relación a las competencias trabajadas, los contenidos incluidos y los recursos utilizados. Este artículo presenta un resumen resultado del análisis y comparación de dichas guías docentes.

El artículo, en primer lugar presenta los referentes curriculares de los contenidos de sistemas digitales (Sección 2). Posteriormente se presenta la metodología realizada para la selección de las guías docentes a analizar (Sección 3). La segunda parte de esta sección muestra el resultado de la revisión de las guías docentes analizadas. La Sección 4 muestra un análisis de los resultados obtenidos. Finalmente, se muestran las conclusiones del trabajo y se fijan unas reflexiones más detalladas que sería interesante realizar en este análisis y que se marcan como trabajos futuros.

2. Diseño curricular

Los contenidos relacionados con los sistemas digitales vienen definidos a nivel internacional por el conocido como *Computing Curricula* (CC). El CC es un compendio de recomendaciones curriculares cuyo objetivo es proporcionar una visión de los distintos programas de grado en el área de la informática. Este conjunto de documentos están elaborados por el grupo de trabajo formado por la *Association of Computing Machinery* (ACM), la *Association of Information Systems* (AIS) y la *IEEE Computer Society* (IEEE-CS). El CC identifica las especializaciones más relevantes de la informática y define las distintas áreas de conocimiento y su nivel de profundidad para cada una de dichas especializaciones. Desde la primera versión aprobada en el 1991 [7], se han ido actualizando de forma periódica, hasta la última versión aprobada a finales del 2020 (CC2020) [3].

Los informes particulares de las especializaciones en Ingeniería de computadores (CE2016) [4] y en Ciencias de la computación (CS2013) [5] del *Computing Curricula* incluyen contenidos relacionados directamente con los sistemas digitales.

En el caso del CE2016, una de las 12 áreas de conocimiento (KA) que define en su cuerpo de conoci-

CE-DIG-1	History and overview
CE-DIG-2	Relevant tools, standards and/or engineering constraints
CE-DIG-3	Number systems and data encoding
CE-DIG-4	Boolean algebra applications
CE-DIG-5	Basic logic circuits
CE-DIG-6	Modular design of combinational circuits
CE-DIG-7	Modular design of sequential circuits
CE-DIG-8	Control and datapath design
CE-DIG-9	Design with programmable logic
CE-DIG-10	System design constraints
CE-DIG-11	Fault models, testing, and design for testability

Cuadro 1: Unidades de conocimiento del área de conocimiento CE-DIG *Digital Design* que define el CE2016. (Fuente: [4])

miento (BoK) corresponde al diseño digital (CE-DIG, *Digital Design*). Para esta área de conocimiento define 11 unidades de conocimiento (KU), como podemos observar en el Cuadro 1. Además, el CE2016 desarrolla un conjunto de temas y de resultados de aprendizaje para cada una de estas unidades de conocimiento (KU). Estos temas y resultados de aprendizaje están separados en fundamentales y complementarios. Al área de conocimiento CE-DIG se le asignan un mínimo de 50 horas.

La guía para la especialización en Ciencias de la computación, CS2013, incluye los contenidos relacionados con sistemas digitales como unidades de conocimiento del área de conocimiento de Arquitectura y organización (AR, *Architecture and Organization*). Más concretamente incluye dos unidades de conocimiento, una relacionada con los sistemas lógicos y digitales (*AR/Digital Logic and Digital Systems*) y otra sobre representación de datos (*AR/Machine Level Representation of Data*). Ambas unidades de conocimiento están clasificadas como fundamentales de nivel 2 (*Core Tier2*)¹ y se le asignan un mínimo de 6 horas.

La referencia para la elaboración de planes de estudios de Ingeniería en Informática utilizada en las universidades españolas es el Libro Blanco sobre Ingeniería en Informática. El Libro Blanco surgió del seno de la CODDII (Conferencia de Directores y Decanos de Ingeniería Informática) como una propuesta no vinculante para la definición de los títulos del Grado de Ingeniería en Informática [1]. El Libro Blanco defi-

¹La guía de recomendaciones CS2013 clasifica las unidades de conocimiento fundamentales en dos niveles, *Tier-1* y *Tier-2*, de forma que los planes de estudios deberían incluir el 100 % de las unidades de tipo *Tier-1*, y un mínimo del 80 % de las *Tier-2*.

ne los contenidos relacionados con sistemas digitales dentro de los Contenidos Formativos Comunes (CFC). Más concretamente, se sitúan en la subcategoría 2.4, *Ingeniería de Computadores*, de la categoría 2 de *Contenidos Específicos de la Ingeniería Informática*. Tal y como indica la descripción de dicha subcategoría, los contenidos de la misma hacen referencia a *Fundamentos, Estructura y Arquitectura de computadores. Tecnología de Computadores*.

3. Revisión de las guías docentes

3.1. Criterios de revisión

Para la revisión de las asignaturas se han establecido una serie de pasos sistemáticos, que consisten en la determinación de los criterios de selección de las asignaturas a analizar y la información a recopilar de las asignaturas seleccionadas.

El primer criterio de selección ha sido limitar el análisis a las universidades y centros que pertenecen a la CODDII. La CODDII engloba a la mayoría de centros y universidades que imparten el Grado de Ingeniería en Informática en España y existe un compromiso entre sus miembros de respetar las directrices del Libro Blanco de Informática [1] y las competencias indicadas en Acuerdo del Consejo de Universidades, por el que se establecieron las recomendaciones para la propuesta por las universidades de memorias de solicitud de títulos oficiales en el ámbito de la Ingeniería Técnica en Informática (Resolución de 8 de junio de 2009 de la Secretaría General de Universidades, BOE núm. 187 de 4 de agosto de 2009).

En aquellas universidades en las que coexisten más de un plan de estudios que conduce a la obtención de Títulos de Ingeniero Técnico en Informática, el análisis se ha limitado a uno de los planes de estudios. En primer lugar se ha optado por analizar el plan de estudios del Grado de Ingeniería en Informática genérico, sin una mención específica. Si solo existían planes de estudios con menciones asociadas, se ha escogido el que correspondía a la mención en Ingeniería de Computadores.

Una vez delimitado el plan de estudios a analizar, era necesario establecer el criterio de selección de la asignatura. Para ello, se han escogido asignaturas entre cuyos contenidos se incluyeran los temas sobre *sistemas combinacionales y/o sistemas secuenciales*. Esta decisión se ha tomado pues consideramos que el caso de sistemas combinacionales y secuenciales son el núcleo básico de cualquier asignatura que esté relacionada con el diseño digital y los sistemas digitales.

También se ha recogido y sintetizado la información referente a la planificación de la asignatura (número de créditos, semestre y módulo al que pertenece), las

competencias que trabaja, los contenidos cubiertos y los recursos utilizados. De esta forma, se proporciona una visión global y contextualizada de la asignatura.

En relación a las competencias que trabaja cada asignatura, el estudio se ha centrado en las competencias recogidas en el Acuerdo del Consejo de Universidades de títulos oficiales en el ámbito de la Ingeniería Técnica en Informática, anteriormente citado. Los contenidos se han englobado en temas globales extraídos del análisis de las propias asignaturas, y que por tanto se han establecido durante el propio análisis de las guías docentes. Finalmente, y en relación a los recursos, el análisis se ha centrado en la bibliografía recomendada y las herramientas utilizadas para la realización de las prácticas o como soporte para el aprendizaje. Para el caso de la bibliografía, se han seleccionado únicamente aquellos libros relacionados con sistemas digitales y que además se encuentren en al menos 3 guías docentes de las analizadas.

3.2. Proceso de revisión

La revisión se ha llevado a cabo durante el mes de enero de 2021 utilizando como base las guías docentes publicadas en las webs de cada centro o universidad, tomando como referencia la guía docente correspondiente al curso 2020/21. Inicialmente se partía de un listado de 72 planes de estudios de universidades pertenecientes a la CODDII. Tras aplicar los criterios de selección explicados en la sección anterior, el estudio se ha realizado sobre un total de 58 planes de estudios de otras tantas universidades. En cualquier caso, como las asignaturas analizadas son mayoritariamente de formación básica, las asignaturas seleccionadas se encontraban habitualmente en todas las menciones del título de una misma universidad, con lo que la asignatura escogida es representativa para todos los casos, ya que en la mayoría de casos compartían la misma guía docente.

En la mayoría de los casos, los conceptos relacionados con sistemas digitales se trabajan en una única asignatura, y por tanto esta ha sido la seleccionada. En un número muy pequeño de casos, ha sido necesario seleccionar dos asignaturas pues los contenidos se encontraban divididos entre ambas.

De las asignaturas analizadas, se ha extraído el conjunto de competencias que se trabajan (Cuadro 2), que incluyen competencias del módulo de formación básica (FB), común a la rama de informática (MC), de tecnología específica de Ingeniería de Computadores (IC), de tecnología específica de Tecnologías de la Información (TI) y de tecnología específica de Ingeniería del Software (IS). El Cuadro 2 muestra las competencias del Acuerdo de Consejo de Universidades anteriormente citado y que se han encontrado en alguna de las guías docentes analizadas.

Id.	Competencia	Módulo
CFB1	Capacidad para la resolución de los problemas matemáticos que puedan plantearse en la ingeniería. Aptitud para aplicar los conocimientos sobre: álgebra lineal; cálculo diferencial e integral; métodos numéricos; algorítmica numérica; estadística y optimización.	FB
CFB2	Capacidad para comprender y dominar los conceptos básicos de matemática discreta, lógica, algorítmica y complejidad computacional, y su aplicación para la resolución de problemas propios de la ingeniería.	FB
CFB3	Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.	FB
CFB4	Comprensión de la estructura, organización, funcionamiento e interconexión de los sistemas informáticos, los fundamentos de su programación, y su aplicación para la resolución de problemas propios de la ingeniería.	FB
CFB5	Comprensión y dominio de los conceptos básicos de campos y ondas y electromagnetismo, teoría de circuitos eléctricos, circuitos electrónicos, principio físico de los semiconductores y familias lógicas, dispositivos electrónicos y fotónicos, y su aplicación para la resolución de problemas propios de la ingeniería.	FB
CI103	Capacidad de dar solución a problemas de integración en función de las estrategias, estándares y tecnologías disponibles.	IS
CI201	Capacidad de diseñar y construir sistemas digitales, incluyendo computadores, sistemas basados en microprocesador y sistemas de comunicaciones.	IC
CI202	Capacidad de desarrollar procesadores específicos y sistemas empujados, así como desarrollar y optimizar el software de dichos sistemas.	IC
CI502	Capacidad para seleccionar, diseñar, desplegar, integrar, evaluar, explotar y mantener las tecnologías de hardware, software y redes, dentro de los parámetros de coste y calidad adecuados.	TI

Cuadro 2: Competencias trabajadas en las asignaturas relacionadas con sistemas digitales.

Del análisis de los contenidos se han encontrado una serie de bloques o temas que corresponden a: abstracción digital, representación de la información, álgebra de Boole, puertas lógicas, sistemas combinacionales, sistemas secuenciales, lógica programable, organización básica de la CPU y memorias. De forma menos representativa, también se han encontrado algún otro contenido relacionado con la física, la electrónica o temas adicionales de la arquitectura de computadores. Todos estos últimos casos los hemos englobado como *Otros*. Para indicar el nivel de profundidad con el que se trabaja cada uno de los conceptos o temas, se ha incluido un círculo, cuyo relleno indica el nivel de profundidad de cada uno de los temas.

Los Cuadros 3 y 4 muestran la asignaturas analizadas, la información de su planificación y los contenidos trabajados.

Finalmente, en el caso de los recursos, la bibliografía que se ha encontrado en las guías docentes seleccionadas se muestra en el Cuadro 5. Este listado solo incluye aquellos libros incluidos en al menos tres guías docentes y que estén relacionados con sistemas digitales.

La recopilación de las herramientas y software utilizado como soporte de la asignatura o para la realización de las prácticas de las mismas ha sido más complicada. Esta información no se ha encontrado siempre directamente explicitada en las guías docentes, y a veces ha sido necesario deducirla a partir de la bibliografía u otros contenidos. Incluso, en la mayoría de las ocasiones no ha sido posible establecerse. De forma resumida, algunas de las herramientas que se han encontrado

explicitadas son Proteus², LogicWorks³, Digital⁴, OrCAD⁵ y PSpice⁶.

4. Análisis

En relación a las planificación y modalidad de la asignatura, la mayoría de asignaturas análogas son de formación básica, a excepción de algunos casos que está definida como materia de módulo común. Incluso existen dos universidades en las que los contenidos relacionados con sistemas digitales se imparten en asignaturas del módulo de especialización del itinerario de Ingeniería de Computadores.

De forma mayoritaria, las asignaturas relacionadas con sistemas digitales se encuentran situadas en el primer curso de los estudios, aunque no destaca ninguno de los dos semestres por encima del otro.

En relación a las competencias trabajadas, la más habitual es la identificada como CFB4 (Cuadro 2), seguidas de la CFB5, y en tercer lugar por la competencia CCM09. Esto se debe al propio hecho de que la mayoría de las asignaturas analizadas son de formación básica, y por tanto están asociadas a competencias de dicho módulo.

Para una mejor análisis de los contenidos de las asignaturas, hemos clasificado las asignaturas en grupos. Para ello, se ha aplicado un algoritmo de agrupación sobre las columnas correspondientes a los contenidos. Más concretamente se ha aplicado un algoritmo jerárquico aglomerativo [6]. Este tipo de algoritmos empie-

²<https://www.labcenter.com/>

³<https://www.designworksolutions.com/logicworks/>

⁴<https://github.com/hneemann/Digital>

⁵<https://www.orcad.com/>

⁶<https://www.pspice.com/>

		Planif.			Competencias												Contenidos										
	Universidad - Asignatura	Créditos	Módulo ^a	Semestre/Curso	CFB1	CFB2	CFB3	CFB4	CFB5	CCM04	CCM08	CCM09	CCM14	CI103	CI201	CI202	CI502	Abstracción digital	Representación información	Alg. Boole	Puertas lógicas	Combinacionales	Secuenciales	Lógica programable	Estructura CPU	Memoria	Otros
Grupo 1	UHU Fundamentos de Computadores	6	FB	1/1	✓							✓						●	●	●	●	●	●	○	○	○	○
	UVA Sistemas Digitales	6	FB	1/1				✓										●	○	●	●	●	●	○	○	○	○
	UAB Fundamentos de los Computadores	6	FB	1/2				✓										●	○	●	●	●	●	○	○	○	○
	USJ Sistemas lógicos	6	MC	1/2														●	○	○	●	●	●	○	○	○	○
	UMH Tecnología de computadores	6	MC	2/2				✓	✓			✓	✓					●	○	●	●	●	●	○	○	○	○
	ULL Sistemas electrónicos digitales	6	FB	1/2					✓									○	○	○	●	●	●	○	○	○	○
	UFV Electrónica y tecnología de computadores	6	FB	1/2					✓									●	○	●	●	●	○	○	○	○	○
Grupo 2	UJA Electrónica digital	6	FB	1/2					✓									○	●	○	●	●	●	○	○	○	○
	UPSA Fundamentos y tecnología de computadores	6	FB	1/1					✓		✓				✓			○	●	○	●	●	○	○	●	●	○
	UNILEON Electrónica	6	FB	1/1								✓						○	●	○	●	●	○	○	○	○	○
	URV Fundamentos de computadores	6	FB	1/2				✓	✓									○	●	○	●	●	○	○	○	○	○
Grupo 3	UBU Diseño e implementación de sistemas digitales	6	O	4/2										✓	✓			○	○	○	○	●	●	○	○	○	○
	UJI Diseño de sistemas digitales	6	O	4/1											✓			○	○	○	○	●	●	○	○	○	○
	UdC Fundamentos de los computadores	6	FB	1/2				✓			✓				✓			○	○	○	○	●	●	○	○	○	○
Grupo 4	UCLM Tecnología de computadores	6	FB	1/1	✓				✓		✓							●	●	●	●	●	○	○	○	○	○
	ULPGC Fundamentos de computadores	6	FB	1/1				✓	✓		✓							○	●	●	●	●	○	○	○	○	○
	UNED Fundamentos de sistemas digitales	6	FB	1/1					✓									●	●	●	●	●	○	○	○	○	○
	UPM Sistemas digitales	6	FB	1/2														○	○	○	○	○	○	○	○	○	○
	UAM Fundamentos de computadores	6	FB	1/1				✓										○	●	●	●	●	○	○	○	○	○
	UAH Fundamentos de tecnologías de computadores	6	FB	1/1					✓		✓							●	●	●	●	●	○	○	○	○	○
	UC3M Tecnología de computadores	6	FB	1/2				✓	✓									○	●	●	●	●	○	○	○	○	○

^a Módulo: FB Formación Básica; MC Módulo común; O Optativa.

Cuadro 3: Asignaturas relacionadas con sistemas digitales de las universidades pertenecientes a la CODDII y agrupación obtenida en base a los contenidos.

zan clasificando cada asignatura de forma individual en un grupo independiente, y en cada paso del algoritmo se juntan los dos grupos más similares. De esta forma se genera un estructura en forma de dendograma. Posteriormente, y mediante la visualización del dendograma, se pueden establecer el número de grupos adecuados para cada problema. De este proceso hemos obtenido 10 grupos de universidades clasificadas en función de los contenidos incluidos. Los grupos resultantes se muestran en los Cuadros 3 y 4 separados por líneas horizontales.

Del análisis de los contenidos, se observa que la mayoría de asignaturas trabajan los bloques temáticos correspondientes a representación de la información, álgebra de Boole, puertas lógicas, sistemas combinacionales y sistemas secuenciales. La inclusión de contenidos relacionados con la lógica programable y los lenguajes de descripción de hardware es mucho menos representativo, dado que estos contenidos están incluidos en solo aproximadamente el 30 % de los casos. Ade-

más, se observa también que hay un número de casos, que aunque no es representativo sí que es destacable, en los que se incluyen bloques temáticos relacionados con la estructura de computadores (CPU, memoria, Entrada/Salida) y algunos más minoritarios que incluyen contenidos relacionados con la electrónica y los circuitos eléctricos.

A primera vista, se ve que los bloques temáticos de representación de información, álgebra de Boole, puertas lógicas, sistemas combinacionales y sistemas secuenciales son el núcleo básico de la mayoría de los grupos de asignaturas. Aunque aquellas asignaturas que no incorporan, o bien representación de la información, o bien álgebra de Boole, han sido clasificadas en dos grupos diferentes (grupos 1 y 2 del Cuadro 3). Otro de los grupos aparece por la inclusión de contenidos relacionados con memoria, pero dejando aparte otros temas de la estructura de computadores (grupo 4 del Cuadro 3). También se observa como un número reducido de asignaturas solo cubren contenidos de sis-

					Planif.		Competencias										Contenidos													
					Créditos	Módulo ^a	Semestre/Curso	CFB1	CFB2	CFB3	CFB4	CFB5	CCM04	CCM08	CCM09	CCM14	CI103	CI201	CI202	CI502	Abstracción digital	Representación información	Alg. Boole	Puertas lógicas	Combinacionales	Secuenciales	Lógica programable	Estructura CPU	Memoria	Otros
Universidad - Asignatura																														
Grupo 5	UdG Estructura i tecnologia de computadores I	9	FB	1/1				✓	✓												○	●	●	●	●	●	●	●	○	○
	UCAM Fundamentos de computadores	4,5	FB	1/1						✓	✓										○	●	●	●	●	○	●	○	○	○
	Deusto Electrónica digital	6	FB	1/1																	○	●	●	●	●	●	○	○	○	○
	UV Tecnología de computadores	6	FB	1/1			✓	✓			✓										○	○	○	○	○	○	○	○	○	○
	y Fundamentos de computadores	6	FB	1/2			✓	✓		✓											○	○	○	○	○	○	○	○	○	○
	URJC Fundamentos de computadores	6	MC	1/2						✓	✓				✓					✓	○	●	●	○	○	○	○	○	○	○
Grupo 6	UCO Fundamentos y Estructura de Computadores	6	FB	1/2					✓												●	●	●	●	●	○	○	○	○	○
	UCA Sistemas Digitales	6	FB	1/2																	●	●	●	●	●	○	○	○	○	○
	UNIOVI Fundamentos de computadores y redes	6	FB	1/2				✓	✓												●	●	●	●	○	○	○	○	○	○
	y Tecnología electrónica de computadores	6	MC	2/1				✓	✓	✓	✓										○	○	○	○	○	○	○	○	○	○
Grupo 7	UPC Introducción a los computadores	7,5	FB	1/1					✓						✓						●	●	●	●	●	○	○	○	○	○
	UIB Sistemas Digitales	6	MC	1/1											✓						●	●	●	●	○	○	○	○	○	○
	UMA Fundamentos de electrónica	6	FB	1/1							✓										●	●	●	●	○	○	○	○	○	○
	UGR Tecnología y Organización de Computadores	6	MC	1/2						✓											●	●	●	○	○	○	○	○	○	○
	UNEX Tecnología de computadores	6	FB	1/1						✓	✓										●	●	●	●	○	○	○	○	○	○
	UNICAN Sistemas digitales	6	FB	1/1						✓	✓										●	●	●	●	○	○	○	○	○	○
	UEM Bases de la informática	6	FB	1/1					✓												●	●	●	●	○	○	○	○	○	○
Grupo 8	y Estructura de computadores	6	FB	1/2					✓						✓						○	○	○	○	○	○	○	○	○	○
	UAL Estructura y Tecnología de Computadores	6	FB	1/2					✓												○	●	●	●	●	○	○	○	○	○
	USAL Computadores I	6	FB	1/1					✓												○	●	●	●	○	○	○	○	○	○
	UNIR Tecnología de computadores	6	FB	1/1											✓						○	●	●	●	○	○	○	○	○	○
	UM Fundamentos de computadores	6	FB	1/1				✓	✓												○	○	○	○	○	○	○	○	○	○
	y Estructura y tecnologías de computadores	6	FB	1/2				✓	✓												○	○	○	○	○	○	○	○	○	○
	EHU/UPV Principios de diseño de sistemas digitales	6	FB	1/1							✓										○	●	●	●	○	○	○	○	○	○
Grupo 9	UDIMA Tecnología y estructura de computadores	6	MC	1/-						✓				✓							○	●	●	●	○	○	○	○	○	○
	UPO Sistemas Digitales	6	FB	1/1					✓	✓											○	●	●	●	○	○	○	○	○	○
	US Circuitos Electrónicos Digitales	6	FB	1/1					✓												○	●	●	●	○	○	○	○	○	○
	UB Diseño digital básico	6	FB	1/1				✓	✓												○	●	●	○	○	○	○	○	○	○
	URL Introducción a los ordenadores	9	FB	1/A																	○	●	●	●	○	○	○	○	○	○
	UdL Estructura de computadores I	6	FB	1/1					✓						✓						○	●	●	●	○	○	○	○	○	○
	UNIRIOJA Estructura de computadores	6	FB	1/2					✓						✓						○	●	●	●	○	○	○	○	○	○
	UNIZAR Introducción a los computadores	6	FB	1/1				✓	✓												○	●	●	●	○	○	○	○	○	○
	UPV Fundamentos de computadores	6	FB	1/1				✓	✓												○	●	●	●	○	○	○	○	○	○
	USC Sistemas digitales	6	FB	1/1				✓		✓											○	●	●	●	○	○	○	○	○	○
	UVIGO Sistemas digitales	6	FB	1/1				✓			✓	✓	✓				✓		✓		○	●	●	●	○	○	○	○	○	○
	UOC Fundamentos de computadores	6	FB	1/-																	○	●	●	●	○	○	○	○	○	○
	UCM Fundamentos de computadores I	6	MC	1/1						✓											○	●	●	●	○	○	○	○	○	○
UA Fundamentos de los computadores	6	FB	1/1						✓	✓										○	●	●	●	○	○	○	○	○	○	
Grupo 10	UPF –	–	–	–																										
	UEMC –	–	–	–																										
	UPNA –	–	–	–																										

^a Módulo: FB Formación Básica; MC Módulo común; O Optativa.

Cuadro 4: Asignaturas relacionadas con sistemas digitales de las universidades pertenecientes a la CODDII y agrupación obtenida en base a los contenidos.

Thomas L. Floyd. <i>Fundamentos de Sistemas Digitales</i> . Ed. Prentice Hall. 2016. ISBN: 849035300X.
John P. Hayes. <i>Introducción al Diseño Lógico Digital</i> . Addison-Wesley Iberoamericana. 1996. ISBN 0-201-62590-3
Mano M. Morris y Charles R. Kime. <i>Fundamentos de diseño lógico y de computadores</i> . 2005. Pearson Prentice-Hall. ISBN: 8420543993.
Javier García Zubía. <i>Problemas resueltos de electrónica digital</i> . Editorial Paraninfo. Madrid. 2012
Ronald J. Tocci. <i>Sistemas digitales. Principios y aplicaciones</i> . Pearson International. 2007 ISBN: 978-970-26-0970-4
Daniel D. Gajski. <i>Principios de diseño digital</i> . Prentice Hall. Madrid. 1997. ISBN 8483220040.
Antonio Lloris Ruiz, Alberto Prieto Espinosa y Luis Parrilla Roure. <i>Sistemas digitales</i> . McGraw-Hill. Madrid. 2003. ISBN: 9788448191887.
Charles H. Roth. <i>Fundamentos de diseño lógico</i> . Thomson. 2009.
John F. Wakerly. <i>Diseño Digital. Principios y Prácticas</i> . Pearson Educación. 3ª edición. 2001. ISBN 9702607205
David Harris y Sarah Harris. <i>Digital Design and Computer Architecture</i> . Morgan Kaufmann. 2012. ISBN: 9789382291527
José M ^a Angulo y Javier García Zubía. <i>Sistemas Digitales y Tecnología de Computadores</i> . Paraninfo.
Enrique Mandado Pérez. <i>Sistemas electrónicos digitales</i> . Editorial Barcelona Marcombo. 2014
Roger Tokheim. <i>Electrónica digital. Principios y aplicaciones</i> . McGraw Hill. 2008. ISBN 970-10-6667-7
Milos Ercegovic, Tomás Lang y Jaime H. Moreno. <i>Introduction to Digital Systems</i> . John Wiley and Sons. 1999
Carmen Baena Oliva. <i>Problemas de circuitos y sistemas digitales</i> . Madrid Mac Graw-Hill 1997.
Herbert Taub. 'Circuitos Digitales y microprocesadores'. Ed. McGraw-Hill. 1983

Cuadro 5: Listado de la bibliografía referenciada en las guías docentes de las asignaturas seleccionadas.

temas combinatoriales y secuenciales, sin incluir los temas anteriores, e incluyendo lógica programable o estructura de la CPU (grupo 3 del Cuadro 3).

Existe también una clara agrupación de aquellas asignaturas que no cubren abstracción digital, pero sí que añaden al bloque básico comentado anteriormente los contenidos de lógica programable y estructura de CPU (grupo 5 del Cuadro 4). En el grupo 6 del Cuadro 4 vemos que se concentran las asignaturas que cubren prácticamente la totalidad de los bloques temáticos considerados. En el grupo 7 de este segundo cuadro se engloban las asignaturas que incluyen el bloque básico junto con el tema inicial de abstracción digital. Los grupos 8 y 9 del Cuadro 4 incluyen asignaturas que cubren los contenidos del núcleo básico, sin incluir los de abstracción digital. La diferencia entre ambos grupos es la inclusión de los contenidos referentes a estructura de computadores en uno de ellos, o, para el otro grupo, la inclusión adicional de contenidos mínimos o la no inclusión de ningún contenido adicional.

También es destacable el hecho de que hay tres universidades que no contemplan la inclusión de contenidos de sistemas digitales en sus planes de estudios. Situación que se observa en el grupo 10 del Cuadro 4.

Del Cuadro 5, que incluye las referencias bibliográficas encontradas en las guías docentes, hay que clarificar que se encuentran ordenadas por número de ocurrencias. Es decir, el primero de la lista es el libro que se encuentra incluido en un mayor número de guías docentes, y el último se encuentra únicamente en 3 guías docentes (ya que 3 era el valor mínimo de veces que se había establecido para que un libro en particular apareciera en el listado analizado). Es importante indicar que el primero de los libros, cuyo autor es Thomas L. Floyd [2], es el más utilizado en todas las asignaturas con una enorme diferencia sobre el segundo. Por lo que podríamos decir que este es el libro más representativo para los contenidos de sistemas digitales, si se usa el

criterio de popularidad en la bibliografía de las guías docentes.

5. Conclusiones

Este artículo presenta el resultado de llevar a cabo una análisis exhaustivo de las guías docentes de asignaturas que incluyen contenidos relacionados con sistemas digitales y que se imparten en titulaciones del Grado de Ingeniería en Informática. Gracias a este análisis se han elaborado unos cuadros resumen que muestran la temporalización de las asignaturas, las competencias trabajadas y los contenidos incluidos. Además, se ha hecho un análisis de los recursos (bibliografía y herramientas informáticas) utilizados en estas asignaturas.

De forma general, se ha visto que los contenidos de sistemas digitales están incluidos en los módulos de formación básica y situado en el primer curso de los estudios. Hay un gran rango de competencias trabajadas, aunque destacan principalmente aquellas de formación básica relacionadas con la estructura y organización de sistemas informáticos y la teoría de circuitos y principios físicos.

Sería interesante conocer las consecuencias de esta distribución de contenidos, competencias y planificación. Para poder llegar a hacer un análisis más profundo sería necesario plantear preguntas tales como ¿Qué efecto tiene abordar únicamente contenidos relacionados con sistemas digitales? ¿Qué beneficios ofrece el anticipar contenidos de la estructura de computadores en este tipo de asignaturas? ¿Es adecuado incorporar contenidos de Electrónica o de Física? ¿Qué efecto tiene ubicar esta asignatura en el primer o segundo cuatrimestre? ¿Cuáles son los motivos para no incorporar contenidos de sistemas digitales, o incluirlos en una única de las menciones, en los planes de estudios del Grado en Ingeniería Informática?, etc. Todas estas pre-

guntas no pueden ser respondidas exclusivamente mediante el análisis de las guías docentes. La realización de encuestas o entrevistas a los profesores responsables de dichas asignaturas, o incluso a los gestores de las titulaciones, aportarían información más detallada para el análisis de las preguntas planteadas. Como trabajo futuro se establece este objetivo para poder ofrecer una visión más exacta de la situación.

Adicionalmente, podría ser muy interesante extender este análisis a toda la rama de asignaturas de arquitectura de computadores. De esta forma se podría tener una visión global de todo el área de conocimiento. Por tanto, otro trabajo futuro a considerar es extender este análisis a todas las asignaturas de la rama.

En cualquier caso, hay que recordar que toda la información presentada en este artículo ha sido extraída de las guías docentes del curso 2020/21. En algunas ocasiones, la interpretación de la guía docente puede haber dado lugar a errores en los contenidos impartidos debido a la falta de detalle de los mismos. La revisión de la docencia de sistemas digitales aquí presentada tiene como objetivo ser un trabajo vivo, que se vaya actualizando. Por lo que si cualquier persona relacionada con las asignaturas listadas quiere concretar o clarificar alguna información aquí presentada, puede contactar con el primero de los autores.

Referencias

- [1] CODDII. Libro Blanco del Título de Grado en Ingeniería Informática. ANECA. http://www.aneca.es/var/media/150388/libroblanco_jun05_informatica.pdf, 2003. Accedido: 11-01-2021.
- [2] Thomas L Floyd. *Fundamentos de sistemas digitales*. Prentice Hall, 2016.
- [3] CC2020 Task Force. *Computing Curricula 2020. Paradigms for Global Computing Education*. ACM, AIS and IEEE-CS, 2020.
- [4] Joint Task Group on Computer Engineering Curricula. *Guidelines for Undergraduate Degree Programs in Computer Engineering*. ACM, AIS and IEEE-CS, 2016.
- [5] Joint Task Group on Computing Curricula. *Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. ACM, AIS and IEEE-CS, 2013.
- [6] Lior Rokach and Oded Maimon. *Clustering Methods*, pages 321–352. Springer US, Boston, MA, 2005.
- [7] Allen B. Tucker. Computing curricula 1991. *Commun. ACM*, 34(6):68–84, June 1991.

Agilizando el aprendizaje de Bases de Datos

Miguel A. Martínez-Prieto¹, Jorge Silvestre¹, María F. Escudero², Anibal Bregon¹

¹ Depto. de Informática, Universidad de Valladolid

² Servicio Público de Empleo de Castilla y León (ECyL)

{migumar2, jsilvestre, anibal}@infor.uva.es, maria.f.escudero@gmail.com

Resumen

La educación ágil busca implantar las prácticas ágiles en el sistema educativo y establecer nuevas metodologías más orientadas al aprendizaje permanente. En este artículo se describe cómo hemos usado UVAGILE para implantar el aula ágil de “Sistemas de Bases de Datos”, así como las actividades formativas y los mecanismos de evaluación y retroalimentación que hemos utilizado. La experiencia ha sido muy satisfactoria: los alumnos han valorado de forma positiva la metodología, su rendimiento académico ha mejorado notablemente y han destacado su avance en algunas de las *soft skills* más demandadas en el mercado de trabajo.

Abstract

Agile education brings agile practices into educative environments to consolidate methodologies more oriented towards lifelong learning. This paper describes our implementation of UVAGILE to consolidate the agile classroom for “Database Systems”, including the training activities and the evaluation and feedback procedures we have used. The experience has been successful: the students have rated UVAGILE positively, their academic performance has improved significantly, and they have highlighted the acquisition of soft skills that are in high demand in the labour market.

Palabras clave

UVAGILE, educación ágil, proyecto de aprendizaje, *soft skills*, bases de datos, *Agile*, *Scrum*.

1. Introducción

La tecnología se ha convertido en el principal agente del cambio de nuestra sociedad. Las personas, los procesos y los entornos evolucionan con rapidez en un mundo interconectado globalmente [7], en el que la capacidad para adaptarse a los cambios es la base del éxito de cualquier organización. Los *marcos de*

trabajo ágiles (Agile) se han posicionado como una solución eficiente para ello, posibilitando que empresas y, en menor medida, instituciones públicas aborden con éxito los desafíos organizativos y de gestión de proyectos que se les presentan. *Agile* se apoya en los cuatro valores propuestos en el Manifiesto Ágil (<https://www.agilemanifesto.org/>), que promueven la interacción entre las personas, la entrega continua de valor (en forma de productos funcionales), la colaboración con el cliente y la flexibilidad para responder ante los cambios. Implantar *Agile* demanda profesionales con una mentalidad ágil [6], en la que se requiere una actitud positiva ante los desafíos y una alta motivación por seguir aprendiendo y mejorando, además de la capacidad de trabajo en equipo, la comunicación o la adaptabilidad, entre otras *soft skills*.

El éxito de *Agile* también ha repercutido en el mercado de trabajo [3]. Las empresas ya no solo requieren conocimientos técnicos, sino que valoran, cada vez más, la capacidad de los candidatos para integrarse en equipos de trabajo ágiles, bajo la premisa de que los conocimientos técnicos se pueden aprender, y aprender es una *soft skill* en sí misma [11]. Por lo tanto, el aprendizaje debe abordarse como un proceso permanente en la vida de una persona, y es responsabilidad del sistema educativo proporcionarle las destrezas necesarias para que sea capaz de dirigirlo satisfactoriamente. El Sistema Educativo Español reconoce que “aprender a aprender” es fundamental para el aprendizaje permanente de las personas y lo establece como una competencia clave para su desarrollo personal, indicando que la motivación y la confianza son cruciales para adquirirla. La *Orden ECD/65/2015, de 21 de enero*, también plantea que los procesos de enseñanza-aprendizaje deben potenciar las estrategias de planificación, supervisión y evaluación que permitan a los alumnos determinar sus objetivos de aprendizaje (y el plan de acción para conseguirlos), examinar su evolución y analizar tanto los resultados de aprendizaje como el proceso seguido para alcanzarlos.

El aprendizaje permanente requiere de nuevas metodologías docentes que ayuden a transmitir las competencias necesarias para desenvolverse en el mercado laboral y ahí, nuevamente, los marcos de trabajo ági-

les se están posicionando como una solución efectiva. Los procesos de enseñanza-aprendizaje presentan paralelismos con la organización de proyectos en entornos de negocio [4], por lo que abordar sus objetivos de aprendizaje con una aproximación ágil permitirá obtener productos (de aprendizaje) de mayor calidad y más satisfactorios para los estudiantes. La *educación ágil* (Sección 2) surge, sobre esta premisa, con el objetivo de implantar los valores ágiles en el sistema educativo, posibilitando que los estudiantes adquieran nuevos hábitos de aprendizaje inspirados en las dinámicas ágiles.

Este artículo presenta cómo hemos utilizado UVAGILE [10] para implantar el aula ágil de la asignatura Sistemas de Bases de Datos. La Sección 3 detalla los diferentes tipos de actividades realizadas, los mecanismos de evaluación y retroalimentación utilizados y las características del entorno de aprendizaje virtual desplegado. Finalmente, la Sección 4 analiza los resultados de esta experiencia, en los que se constata una mejora en el rendimiento académico y una alta valoración de UVAGILE como metodología de aprendizaje.

2. Antecedentes

Los marcos de trabajo ágiles comenzaron a transformar la industria del software, hace dos décadas, con la implantación de prácticas ágiles destinadas a construir productos de alta calidad en entornos de cambio.

Agile aborda los proyectos de forma iterativa e incremental, implicando a los clientes en su desarrollo y generando valor de forma regular. Este último aspecto es básico para maximizar la satisfacción del cliente, al proporcionarle versiones funcionales del producto, que le permiten experimentar con él y retroalimentar su desarrollo.

2.1. Educación ágil

Compañías de diferentes sectores han adoptado *Agile* para gestionar sus proyectos y consolidar equipos de trabajo centrados en la mejora continua. Recientemente, el jefe de transformación ágil de un banco afirmó que “*Agile* es seguir aprendiendo constantemente” [8], lo que motiva su adopción para fines docentes.

López-Alcarria, *et al.* [7] hipotetizan que introducir *Agile* en las aulas ayudará a mejorar los procesos de enseñanza-aprendizaje de forma comparable a cómo lo ha hecho con el desarrollo de proyectos y la entrega de productos en la industria. Los beneficios de adoptar *Agile* en educación están relacionados con el *aprendizaje experiencial* [5]. Este tipo de aprendizaje requiere la participación activa del alumno y una reflexión continua, para así comprender sus avances (y bloqueos) y consolidar el conocimiento obtenido. Por lo tanto, la educación ágil también se relaciona con el

constructivismo, fomentando la adquisición de competencias como la autonomía para construir conocimiento, la capacidad para evaluar soluciones alternativas, o el pensamiento crítico [7].

La educación ágil (i) respeta los diferentes estilos de aprendizaje y promueve la interacción entre estudiantes y profesores; (ii) mejora la empleabilidad de los estudiantes y los capacita para su formación permanente; (iii) apoya a los estudiantes de forma continua y más allá de los objetivos de aprendizaje; (iv) responde con rapidez a las necesidades que se detecten en el progreso del aprendizaje y ajusta los procesos de enseñanza-aprendizaje en consecuencia; y (v) proporciona mecanismos regulares y frecuentes de evaluación formativa [3]. Todo ello consolida entornos de aprendizaje que favorecen la motivación y la satisfacción de estudiantes y profesores, y la adquisición de *soft skills*.

2.2. UVAGILE

UVAGILE [10] surge con el objetivo de asentar entornos de aprendizaje de calidad basados en los fundamentos de la educación ágil. UVAGILE organiza los procesos de enseñanza-aprendizaje como *proyectos de aprendizaje ágiles*, cuya dinámica está basada en *Scrum* [12]. El proyecto parte de una visión general, que establece las características del *producto de aprendizaje* (de acuerdo con los objetivos de la asignatura) y determina los contenidos a impartir para que los estudiantes puedan “construirlo”. El proyecto se divide en *sprints de aprendizaje* de la misma duración (entre 2 y 5 semanas) y cada *sprint* consta de cuatro etapas:

- En la etapa de *Inicio* se fija el objetivo del *sprint* y se determinan las actividades que se realizarán para conseguirlo. Toda esta información se publica en el *tablero de aprendizaje* del proyecto.
- En la etapa de *Aprendizaje* se llevan a cabo las actividades planificadas en la etapa anterior.
- La etapa de *Evaluación* se realiza el último día del *sprint* para valorar el *incremento* (del producto de aprendizaje) consolidado por cada estudiante.
- La *Retrospectiva* finaliza el *sprint* con una actividad de reflexión colectiva, centrada en la mejora continua del proceso de aprendizaje.

El esquema anterior requiere que el profesor, como facilitador del aprendizaje, elija las mejores actividades formativas para abordar el objetivo de cada *sprint*. Asimismo, es responsable de plantear mecanismos de evaluación que ayuden a los estudiantes a retroalimentar su aprendizaje al finalizar cada *sprint* y a consolidar su producto de aprendizaje. Finalmente, el profesor también usa el *feedback* obtenido en las retrospectivas para mejorar el proceso de enseñanza-aprendizaje.

La organización de los procesos de enseñanza-aprendizaje como *proyectos ágiles* es el primero de los

cinco principios de UVAGILE. Además, promueve el *aprendizaje activo*, pero sin restringir los métodos docentes que pueden utilizarse en las actividades formativas de cada *sprint*. Asimismo, fomenta la *interacción* entre alumnos y profesores, como base para el aprendizaje social [1] y como mecanismo para afianzar competencias como la comunicación, la iniciativa o las relaciones interpersonales. La *adquisición de soft skills* es otro de los principios de UVAGILE, aprovechando el potencial de la educación ágil para abordar este tipo de competencias. Finalmente, UVAGILE implanta un *entorno virtual* en cada proyecto, fomentando que el aprendizaje trascienda a las actividades presenciales.

UVAGILE se despliega como un *aula ágil* en la que cada alumno construye su producto de aprendizaje, de acuerdo con los principios de la metodología. También puede utilizarse para el desarrollo de actividades de aprendizaje basado en proyectos (*ABP ágil*). En este caso, el producto de aprendizaje se construye entre varios alumnos, que trabajan de forma colaborativa como un equipo *Scrum* [12], y el profesor adopta los roles de *product owner* y cliente, facilitando la planificación del proyecto y retroalimentando su aprendizaje.

3. Aula ágil SBD

Sistemas de Bases de Datos (SBD) es una asignatura obligatoria de 2º curso del Grado en Ingeniería Informática de Servicios y Aplicaciones (Universidad de Valladolid), cuyo objetivo es que los alumnos aprendan a diseñar, gestionar y utilizar bases de datos.

En esta sección, se describe cómo hemos utilizado UVAGILE para implementar el proceso de enseñanza-aprendizaje de SBD en forma de aula ágil, durante el primer cuatrimestre del curso 2020-2021. Este curso ha sido el primero en el que la asignatura ha tenido dos profesores y el número de alumnos matriculados (43) ha sido más alto que en cursos anteriores.

3.1. Diseño del proyecto de aprendizaje

Implantar un aula ágil requiere “traducir” la guía docente de la asignatura en los términos de un proyecto de aprendizaje ágil. Esto significa establecer la *visión* del proyecto, determinar sus *contenidos* y plantear el calendario de *sprints de aprendizaje*. En nuestro caso, esta información se difundió a los alumnos de SBD durante la última semana de septiembre de 2020, con el objetivo de proporcionarles una descripción general de la asignatura, antes de la primera clase.

La *visión del proyecto* establece el objetivo global del aula ágil y, por tanto, caracteriza el “producto de aprendizaje” que se construirá en ella. En nuestro caso, el producto deseado es que “el estudiante sea capaz de obtener el diseño conceptual de una base de datos

Como Graduat@ en Ingeniería Informática quiero conocer cómo se modela la información del mundo real para ser capaz de obtener el diseño conceptual de una base de datos.

☒ CA-2.1. Conozco y comprendo los conceptos de entidad, relación y atributo y lo que representan desde el punto de vista del diseño conceptual de una base de datos.

☒ CA-2.2. Soy capaz de identificar entidades de datos en el pliego de características que describe un sistema de información.

☒ CA-2.3. Sé diferenciar cuándo una entidad debe modelarse como *débil*, de acuerdo a su dependencia de una entidad fuerte.

...

Cuadro 1: Ejemplo de historia de aprendizaje.

(a partir de un conjunto de especificaciones de negocio), transformarlo en el diseño lógico correspondiente y desplegarlo en un sistema gestor de bases de datos (SGBD) relacional, sobre el que implementará diversas funcionalidades, utilizando SQL”.

El refinamiento de la visión permite concretar los objetivos de aprendizaje del aula ágil, que se expresan en forma de *historias de aprendizaje* y se publican en el *tablero de aprendizaje* del proyecto, cuyo uso es comparable al del *backlog de producto* en *Scrum* [12]. El Cuadro 1 muestra una parte de la historia de aprendizaje “Diseño Conceptual”, en la que se plantea el “qué” y el “porqué” de este objetivo de aprendizaje y se presentan tres de los criterios de aceptación que se utilizaron en su evaluación. La especificación completa de esta historia, junto con las de los otros cuatro objetivos de aprendizaje de SBD, está disponible en el siguiente tablero: <https://trello.com/b/hx2G5r0E>.

El siguiente paso es elegir los *contenidos*, que se desarrollarán para abordar los objetivos de aprendizaje del proyecto. Este es un aspecto delicado, ya que los contenidos deben ajustarse a la guía docente, pero dejando suficiente margen de maniobra para poder adaptarlos de acuerdo con la evolución del proyecto de aprendizaje. En SBD se propusieron cuatro bloques temáticos: “Introducción a los SGBDs”, “Diseño Conceptual”, “Diseño Lógico” y “SQL” y se plantearon los epígrafes generales de cada uno de ellos.

La última decisión en el diseño del proyecto de aprendizaje es establecer el *calendario de sprints*. Esta decisión establece el ritmo de aprendizaje, considerando que cada *sprint* es un “mini-proyecto” con un objetivo concreto que contribuye a la construcción del producto de aprendizaje. El calendario de sprints de SBD se ajustó a las 12,5 semanas de cuatrimestre (“recortado” debido a la situación sanitaria). La asignatura comenzó con un *Sprint #0* (de 2 semanas de duración), destinado a que los alumnos se familiarizaran con UVAGILE y adquirieran los conocimientos básicos sobre bases de datos. Los tres *sprints* regulares se planificaron con una duración de 3,5 semanas. Por lo tanto, la evaluación de los incrementos de aprendizaje se llevó a cabo durante las semanas 6, 9 y 12.

Criterio	Condiciones de aceptación	Puntos
CA-2.3.	El modelo ER plantea, al menos, el 50 % de las entidades débiles esperadas de acuerdo con el pliego.	1
CA-2.5.	Al menos el 90 % de los atributos propuestos declaran un tipo correcto.	3,5
...		

Cuadro 2: Ejemplo de rúbrica de corrección.

3.2. Sprints de aprendizaje

Los *sprints* establecen los ciclos de aprendizaje que se llevan a cabo en el proyecto y comprenden la ejecución de las etapas de Inicio, Aprendizaje, Evaluación y Retrospectiva.

La etapa de *Inicio* se ha realizado a través de un vídeo, en el que el profesor planteaba el *alcance* del *sprint*, en términos de las historias de aprendizaje que constituían su objetivo. Además, el profesor presentaba el calendario de actividades formativas del *sprint*, indicando cómo contribuía cada una de ellas a alcanzar el objetivo del *sprint*. Toda esta información también se publicaba en el tablero de aprendizaje. Así, cada actividad exponía sus contenidos formativos y los criterios de aceptación que ayudaba a superar.

La ejecución de la etapa de *Aprendizaje* comprendió la realización de actividades presenciales y no presenciales, planteadas específicamente para abordar el objetivo de cada *sprint*. Una vez finalizada cada actividad, el profesor actualizaba el tablero de aprendizaje, para que reflejase el estado del *sprint*.

La etapa de *Evaluación* se llevó a cabo el último día del *sprint*, mediante dos actividades complementarias:

1. La realización de un examen, cuya corrección se basó en una rúbrica, publicada días antes por el profesor. El Cuadro 2 muestra un fragmento de la rúbrica utilizada en el examen del *Sprint #1*, en la que se enumeran los criterios de aceptación y las condiciones sobre las que se evalúan, así como la puntuación de cada uno de ellos.
2. La entrega del incremento correspondiente del proyecto *SeriesDB*. En este caso, la evaluación se realizó respecto a los criterios de aceptación establecidos en el proyecto, que fueron un subconjunto de los asociados con las historias de aprendizaje seleccionadas en la etapa de Inicio.

La *Retrospectiva* se realizó al finalizar el examen. A diferencia de otros cursos, usamos una herramienta digital (para guardar la distancia de seguridad) y proyectamos las respuestas, de forma que todos los alumnos viesen las reflexiones de otros compañeros en tiempo real. Este aspecto resultó positivo, ya que la calidad de la reflexión colectiva mejoró respecto a otros cursos y se plantearon acciones que ayudaron a mejorar inmediatamente el desarrollo de la asignatura.

3.3. Elección de actividades

La educación ágil explota sus fortalezas cuando se combina con métodos docentes de carácter más práctico, destacando cuando se aplica en actividades de aprendizaje basado en proyectos [3].

Este curso hemos mantenido la *clase magistral* como actividad presencial de referencia, programando 18 sesiones magistrales, 5 de ellas dedicadas a la resolución de supuestos prácticos. Además, hemos incorporado una batería de *píldoras de conocimiento*, destinadas a reforzar la explicación de conceptos que, tradicionalmente, han resultado más complejos. El resto de actividades del curso se han orientado a que los alumnos ejercitasen sus conocimientos en un contexto más práctico, buscando el aprendizaje experiencial en el que se sustenta la educación ágil. Para esto último, hemos planteado varias actividades.

Resolución de problemas. Se ha planteado como una actividad en tres fases, inspirada en el concepto de aula invertida y en la experiencia planteada en [9]:

- La primera fase arranca justo después de la sesión magistral de resolución de supuestos prácticos. El profesor publica el enunciado de varios problemas, junto con una rúbrica para su evaluación, y se abre un plazo de 3-4 días para que los alumnos entregasen sus soluciones.
- En la segunda fase, el profesor publica la solución de los problemas y se abre un plazo de 1-2 días para que cada alumno se auto-evalúe y entregue el resultado obtenido.
- Tras las fases anteriores, el profesor revisa las soluciones (y evaluaciones) entregadas, prestando especial atención a los errores cometidos. Esta información se usa para preparar una clase de revisión, en la que se presentan las soluciones correctas y se plantean cuestiones, en relación a los errores detectados, a través de un cuestionario interactivo. La clase se desarrolla como una actividad gamificada, en la que los alumnos obtienen puntos de acuerdo con sus respuestas.

FreeWheeling Travelers. La actividad anterior se desarrolló con éxito en los *sprints* 1 y 2, pero en los cursos anteriores observamos que sólo una minoría de los alumnos resolvía los problemas que se planteaban sobre SQL, por lo que en este curso decidimos *gamificarlos*. Para ello, hemos definido una narrativa que convierte a los alumnos en viajeros y les ofrece la posibilidad de realizar actividades turísticas usando SQL. Este juego, al que hemos denominado *FreeWheeling Travelers*, ha planteado un “viaje semanal”, durante el último *sprint* de la asignatura, en el que los alumnos han puesto en práctica los conocimientos explicados en las clases previas. La actividad finalizó con una clase gamificada, similar a las de las sesiones de problemas.

SeriesDB. El aprendizaje basado en proyectos (ABP) en una actividad fundamental en SBD, ya que le proporciona a los alumnos un contexto realista en el que practicar los conocimientos adquiridos durante la asignatura y, además, entrenar numerosas *soft skills*, considerando que la actividad se realiza en equipo. El curso pasado ya organizamos esta actividad como un proyecto de *ABP ágil* [10], pero detectamos algunos problemas en la dinámica interna de los equipos, relacionados principalmente con su inexperiencia para trabajar de forma colaborativa y con la dificultad de desarrollar el proyecto de forma iterativa e incremental.

Este curso se ha planteado el diseño y construcción de una base de datos sobre *series*. Al igual que el curso pasado, los 10 equipos de trabajo (de 4-5 alumnos) se han formado aleatoriamente y el proyecto se ha dividido en 3 *sprints*, cuya temporización y alcance se han alineado con los *sprints de aprendizaje* del aula ágil.

Como novedad, este curso hemos publicado varias píldoras de conocimiento para que los alumnos comprendiesen mejor los roles y las ceremonias [12] que debían incorporar a su dinámica de trabajo. También hemos creado un nuevo evento: la “ceremonia de bienvenida”, para que los alumnos se conociesen, pusiesen un nombre representativo a su equipo (que les ayude a sentirse parte de él) y estableciesen su dinámica de trabajo antes de comenzar el proyecto. Para ello, les proporcionamos una plantilla de *working agreement*, que podían modificar libremente y actualizar durante el proyecto, para integrar sus propuestas de mejora.

Hall of Fame. Esta actividad se ha introducido con el objetivo de mejorar la motivación y el compromiso de los alumnos con su proceso de aprendizaje. El *Hall of Fame* es un ranking en el que se registra la participación de cada alumno en las actividades opcionales, planteadas para reforzar competencias (comunicación, gestión del tiempo...), resolver problemas adicionales (de mayor complejidad) o participar en la organización del proyecto de aprendizaje de la asignatura.

La participación en cada actividad se premiaba con una “nominación”, que permitía al alumno conseguir las “estrellas condecorativas” (oro, plata y bronce), que a su vez, se traducían en nota extra (0,75; 0,5 y 0,25 puntos, respectivamente), a sumar en su calificación final, siempre que hubiese alcanzado todos los objetivos de aprendizaje. El estado del ranking se actualizaba y publicaba, tras cada actividad, para intensificar el “síndrome de la casilla vacía” [2] y, con ello, motivar a los alumnos a participar en las siguientes actividades.

El Cuadro 3 resume la organización de esta actividad, indicando las nominaciones (n) necesarias para alcanzar cada reconocimiento. La respuesta de los alumnos fue tan positiva que, durante el curso, decidimos añadir la “estrella de platino” (con una recompensa de 1 punto) para reconocer a quienes obtuvieron la totali-

Estrella	Nominaciones	Alumnos	Nota (extra)
Platino	$n = 25$	11	1 punto
Oro	$20 \leq n < 25$	11	0,75 puntos
Plata	$10 \leq n < 20$	11	0,5 puntos
Bronce	$4 \leq n < 10$	3	0,25 puntos
×	$n < 4$	7	-

Cuadro 3: Resultados finales del *Hall of Fame*.

dad de las nominaciones (25) repartidas en la actividad. Como puede observarse, más del 50 % de los alumnos (22 de 43) alcanzaron la estrella de oro, y 11 de ellos completaron las 25 nominaciones. Por el contrario, 7 alumnos no consiguieron ninguna estrella, aunque solo uno de ellos no obtuvo ninguna nominación.

3.4. Evaluación y retroalimentación

La evaluación influye de forma determinante en la implicación de los alumnos para mejorar su aprendizaje [9]. A este respecto, los alumnos valoran mejor la evaluación formativa y frecuente, pero esta supone una mayor carga de trabajo para el profesor. La evaluación en la educación ágil es eminentemente formativa, y la retroalimentación es un elemento primordial para garantizar la satisfacción del cliente en *Agile*. Nuestra aula ágil implementa ambos aspectos, añadiendo una componente de evaluación sumativa.

Examen. Cada examen se evaluó de acuerdo con la rúbrica publicada por el profesor y los resultados se presentaron en términos de los criterios de aceptación evaluados. En el caso de los alumnos que alcanzaron los objetivos de aprendizaje, se indicó también la calificación numérica obtenida. La nota correspondiente a los exámenes supuso el 50 % de la calificación final.

SeriesDB. Los incrementos del proyecto se evaluaron respecto a los criterios de aceptación de los objetivos planificados en cada *sprint* (ver <https://trello.com/b/NvUrS5rJ/>). Como resultado, el profesor entregó a cada equipo un informe sobre el estado de su proyecto respecto a cada uno de los criterios, para así poder mejorarlos en el siguiente *sprint*. Cada incremento también se valoró cuantitativamente, obteniendo un calificación que no se publicó hasta el final del proyecto, con el objetivo de evitar el “conformismo” y fomentar la mejora continua. Para obtener la calificación final se estableció una ponderación incremental. Así, la valoración del producto final fue la que más contribuyó en la nota de la actividad, que supuso un 40 % de la calificación de la asignatura.

Problemas. Los alumnos asumieron la responsabilidad de autoevaluar los problemas planteados durante los *sprints*, utilizando para ello las rúbricas entregadas por el profesor. Esto les proporcionó una retroalimentación inmediata, que luego se complementó en las clases de revisión de problemas. Por otra parte, los resul-

tados de *FreeWheeling Travelers* se retroalimentaron a través del ranking del juego, en el que se acumulaban los puntos obtenidos por los alumnos de acuerdo con sus respuestas. La nota de estas actividades supuso, en total, el 10 % de la calificación de la asignatura.

Además de lo anterior, los profesores de la asignatura fomentaron la realización de tutorías individuales y/o colectivas para mejorar la calidad del *feedback* entregado. Estas tutorías se realizaron de forma remota debido a las medidas de distanciamiento social.

3.5. Entorno de aprendizaje virtual

El cuarto de los principios de UVAGILE establece que el éxito de un proyecto de aprendizaje ágil va más allá de las actividades que se realizan en el aula y requiere que el alumno se sienta acompañado y apoyado en el trabajo que hace fuera de ella [10].

Este curso hemos apostado por *Microsoft Teams* como herramienta central para la comunicación y la interacción dentro del aula ágil. Para ello, hemos creado varios canales temáticos (de uso general) y, además, un canal específico para cada uno de los 10 equipos, conformados en el ámbito de *SeriesDB*. Los equipos han utilizado sus canales para comunicarse internamente y para interactuar con los profesores de la asignatura, además de como espacio de trabajo compartido. También se ha usado Teams para la mensajería privada alumno-profesor, para la realización de tutorías y para la difusión de clases a los alumnos confinados.

Por otra parte, hemos utilizado *Trello* para dar soporte al tablero de aprendizaje de la asignatura y a los tableros de los equipos en *SeriesDB*. Se ha usado *VoxVote* para gamificar los cuestionarios planteados en las sesiones de revisión de problemas y *EasyRetro* para la realización de retrospectivas. Finalmente, se ha usado el Campus Virtual institucional (*Moodle*) para publicar los contenidos de la asignatura y dar soporte a las entregas de todas las actividades no presenciales.

4. Análisis de los resultados

Esta sección analiza el efecto que ha tenido la organización del proceso de enseñanza-aprendizaje de SBD en forma de aula ágil. La asignatura se ha impartido presencialmente, aunque 7 de los 43 matriculados estuvieron confinados en algún periodo durante el curso.

4.1. Resultados académicos

Los resultados académicos han mejorado respecto a los cursos anteriores, como puede observarse en el diagrama de cajas de la Figura 1a. Las notas correspondientes a la parte teórica ("Examen") de la asignatura se elevan hasta alcanzar una mediana de 7,47 puntos,

frente a los 5,97 y 5,75 puntos de los dos cursos anteriores. El rendimiento también mejora en la actividad de ABP ágil (*SeriesDB*), alcanzando una mediana de 7,92 puntos, frente a los 7,04 y los 7,70 de los cursos anteriores. Cabe destacar que los productos entregados por todos los equipos superaron los criterios de aceptación y se calificaron entre 5,1 y 9,5 puntos.

Las notas finales confirman la efectividad de UVAGILE. En este curso, la mediana (sin incluir la nota del *Hall of Fame*), ha sido 7,34, lo que supone mejoras de más de 0,6 puntos respecto al curso 2019-2020 (primera vez que se utilizó UVAGILE en la asignatura) y de casi 1 punto respecto a 2018-2019. Además, el menor tamaño del rango intercuartílico demuestra una mayor concentración de las notas de los alumnos en torno al notable. Este hecho se debe, principalmente, a que la evaluación basada en criterios de aceptación de este curso ha sido más exigente que la utilizada en los cursos anteriores. Finalmente, la tasa de rendimiento (aprobados/matriculados) alcanza el 83,72 % (frente al 76,92 % y el 59,38 % de los cursos anteriores), ya que solo 4 alumnos no superaron la asignatura.

4.2. Valoración de los alumnos

La última actividad de *Hall of Fame* consistió en una encuesta de valoración sobre UVAGILE y sobre la experiencia de haber llevado a cabo el proceso de enseñanza-aprendizaje de forma ágil. Los siguientes resultados se obtuvieron a partir de las respuestas de los 25 alumnos que completaron la encuesta.

Valoración general. Los alumnos han calificado UVAGILE con una nota de 7,6 puntos, valorando su utilidad para el aprendizaje, el esfuerzo requerido, el interés suscitado y la motivación que ha generado. En términos cualitativos, los alumnos valoran "el cambio de sistema respecto a otras asignaturas", "la guía para el estudio" que proporcionan los criterios de aceptación de cada objetivo de aprendizaje o "la necesidad de llevar al día la asignatura", aunque "en algunos momentos ha costado". El 72 % de los alumnos se muestran a favor de usar UVAGILE en más asignaturas.

El 40 % de los alumnos manifestaron dificultades iniciales para adaptarse a la dinámica de aprendizaje sostenido, pero la mayoría (74,9 %) reconocen que la organización como aula ágil no les ha supuesto un sobreesfuerzo. Finalmente, el 64 % de los alumnos consideran que haber pasado por un aula ágil les será útil para organizarse en otras asignaturas y el 68 % cree que les ayudará en su futuro profesional. Estos resultados refuerzan la idea de la educación ágil como medio para el aprendizaje permanente.

Proyecto de aprendizaje ágil. La encuesta también planteó un conjunto de preguntas (tipo *Likert*) centradas en la organización de los proyectos de aprendizaje

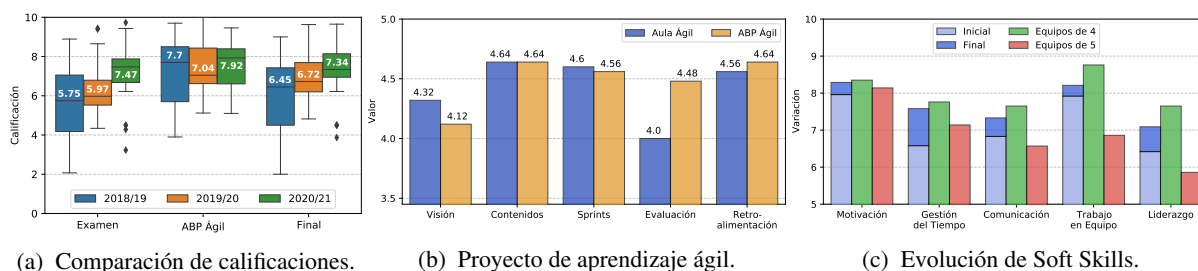
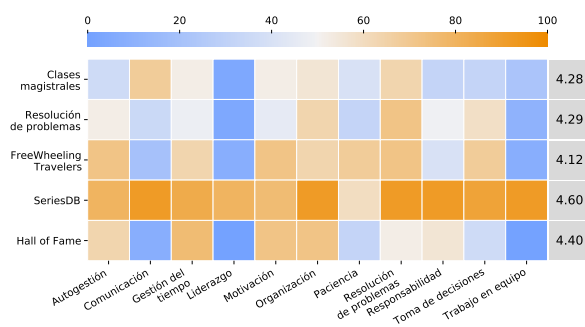


Figura 1: Rendimiento académico y resultados de la encuesta de valoración de los alumnos.

Figura 2: *Soft skills* y actividades formativas.

ágiles. La Figura 1b presenta los resultados obtenidos para el proyecto que implementa el aula ágil y para el de ABP ágil. Como puede observarse, todas las características superan los 4 puntos (sobre 5). En un análisis más detallado, se aprecia que los alumnos valoran mejor la visión del proyecto y la organización del aprendizaje en *sprints* dentro del aula ágil. La valoración de la calidad de los contenidos es comparable en ambos casos (4,64 puntos), lo que demuestra la satisfacción de los alumnos con la forma en la que se han desarrollado los objetivos de aprendizaje de la asignatura. Por otra parte, los mecanismos de evaluación y retroalimentación se valoran mejor en el ABP ágil, debido, principalmente, al mayor detalle de la retroalimentación entregada en los informes de corrección.

Soft skills y actividades formativas. Finalmente, se analiza cómo UVAGILE facilita la adquisición de *soft skills*. Para ello, planteamos varias actividades en el *Hall of Fame* destinadas a que los alumnos entrenasen algunas competencias y reflexionasen sobre su nivel de adquisición. Para esto último, nos apoyamos en la plataforma SkillsMatch [11] y en dos encuestas internas, que los alumnos completaron al inicio y al final del curso, en las que valoraban cómo UVAGILE les había ayudado a mejorar sus capacidades de gestión del tiempo, comunicación, trabajo en equipo y liderazgo. También preguntamos sobre cómo había evolucionado su motivación a lo largo de la asignatura y qué otras competencias habían mejorado durante el curso.

La Figura 1c resume los principales resultados de

este estudio. En cada competencia, la primera columna (“Inicial” / “Final”) muestra el nivel con el que los alumnos la valoraron al principio y al final de curso. Como se observa, todas las competencias mejoran durante el curso: la motivación y el trabajo en equipo presentan una calificación más alta (superior a 8 puntos), pero las otras tres: gestión del tiempo, liderazgo y comunicación, son las que más crecen (hasta superar los 7 puntos, en todos los casos). Durante el estudio detectamos que la evolución de las competencias no era uniforme: mejoraban por encima de la media en el caso de los alumnos que habían realizado el proyecto *SeriesDB* en “Equipos de 4” miembros, pero no así en los alumnos que estaban en “Equipos de 5”. En este caso, solo mejoraron la motivación y la gestión del tiempo, reduciéndose especialmente el trabajo en equipo.

Aunque el resultado anterior es una clara demostración del papel que juega el ABP en la adquisición de *soft skills*, fuimos un paso más allá, tratando de averiguar la relación entre las actividades formativas y la adquisición de competencias. Para ello, la encuesta planteaba una serie de cuestiones en las que preguntamos a los alumnos qué competencias habían desarrollado en cada actividad. En este análisis incluimos las 11 competencias que fueron elegidas por, al menos, 2/3 de los alumnos en una actividad previa del *Hall of Fame*. El mapa de calor de la Figura 2 muestra que los alumnos consideran que *SeriesDB* ayuda a desarrollar todas las competencias estudiadas, en especial la comunicación, la organización, la resolución de problemas, la responsabilidad, la toma de decisiones y el trabajo en equipo. Por otra parte, los alumnos también consideran que *FreeWheeling Travelers* y el *Hall of Fame* facilitan la adquisición de algunas competencias (autogestión, gestión del tiempo o motivación), mientras que las actividades más “tradicionales”: clases magistrales y de resolución de problemas tienen un impacto menor.

Finalmente, destacar la valoración de cada tipo de actividad formativa, que se refleja en la parte derecha de la Figura 2. A la pregunta (*Likert*) “Valora en qué medida te ha ayudado cada tipo de actividad a alcanzar los objetivos de aprendizaje”, los alumnos han asignado 4,6 y 4,4 puntos (sobre 5) a *SeriesDB* y *Hall of Fame*, dejando en 4,29 y 4,28 puntos, respectivamente,

las actividades de resolución de problemas y las sesiones magistrales, y 4,12 puntos al juego *FreeWheeling Traversers*. En este último caso, nos consta que la valoración está influida por algunas cuestiones técnicas de la implementación del juego y por la forma en la que se entregaba el *feedback* después de cada viaje.

4.3. Conclusiones

Implantar el aula ágil de SBD ha supuesto un importante desafío, dado el cambio que supone abordar el proceso de enseñanza-aprendizaje en el marco de una metodología como UVAGILE y considerando que más del 85 % de los alumnos no tenían experiencia previa con *Agile* ni habían cursado ninguna asignatura de Ingeniería de Software. Aún así, hemos encontrado un grupo motivado y abierto al cambio, que ha facilitado enormemente el proceso y ha participado de forma muy activa, tanto en las actividades regladas como en las actividades opcionales del *Hall of Fame*. Los resultados académicos obtenidos avalan la efectividad de UVAGILE, tanto por la mejora en las calificaciones finales, como por la reducción de la tasa de abandono, que ha sido la mitad que en el curso anterior. Esto nos permite concluir que UVAGILE fomenta el compromiso de los alumnos con su aprendizaje.

Por otra parte, la combinación de actividades formativas ha sido muy bien valorada por los estudiantes, lo que avala el principio de UVAGILE de no limitar los métodos docentes que pueden utilizarse en el aula ágil. Finalmente, uno de los resultados más interesantes ha sido la confirmación de que UVAGILE promueve la adquisición de *soft skills*, lo que hace de esta metodología una buena propuesta para el aprendizaje permanente, ya que es capaz de adaptar las prácticas ágiles, de uso habitual en entornos laborales, a la formación de los alumnos. Cabe destacar que el ABP se postula como método de referencia para la adquisición de *soft skills*, sobresaliendo cuando se lleva a cabo en equipos, cuyo tamaño tiene un efecto claro en el resultado.

Agradecimientos

Este trabajo ha sido financiado por el *Proyecto de Innovación Docente* No. 90, del plan PID-2020/2021 de la Universidad de Valladolid (UVa). El segundo autor disfruta de un contrato predoctoral de la Universidad de Valladolid, cofinanciado por el Banco de Santander.

Referencias

- [1] John Biggs y Catherine Tang. *Teaching for quality learning at University*. 2011.

- [2] Pablo del Canto, Isabel Gallego, José M. López, Javier Mora, Angélica Reyes, Eva Rodríguez, Kanapathipillai Sanjeevan, Eduard Santamaría y Miguel Valero. «La evaluación en el contexto del Espacio Europeo de Educación Superior». En: *Revista de Educación a Distancia* (2010).
- [3] Marija Cubric. «An agile method for teaching agile in business schools». En: *The International Journal of Management Education* 11.3 (2013), págs. 119-131.
- [4] Richard Glassey, Mattias Wiggberg y Philipp Haller. «Agile and Adaptive Learning via the ECK-model in the Software Development Academy». En: *13th European Conference On Technology Enhanced Learning*. 2018.
- [5] David A. Kolb. *Experiential learning: Experience as the source of learning and development*. FT press, 2014.
- [6] Howard Leanne. *What Does It Mean to Have an Agile Mindset?* www.agileconnection.com/article/what-does-it-mean-have-agile-mindset. 2015.
- [7] Abigail López-Alcarria, Alberto Olivares-Vicente y Fátima Poza-Vilches. «A Systematic Review of the Use of Agile Methodologies in Education to Foster Sustainability Competencies». En: *Sustainability* 11.10 (2019).
- [8] Oleg Lukin. *Por qué las empresas adoptan cada vez más el método agile*. cincodias.elpais.com/cincodias/2021/01/25/fortunas/1611604386_003512.html. 2021.
- [9] Mercedes Marqués, José M. Badía y Gregorio Quintana. «Contribución sobre el aprendizaje de la evaluación formativa: percepción del alumnado». En: *Actas de las XXVI Jornadas sobre Enseñanza Universitaria de la Informática* 5 (2020), págs. 13-20.
- [10] Miguel A. Martínez-Prieto, Jorge Silvestre, Aníbal Bregón y José I. Farrán. «Hacia la consolidación de las aulas ágiles». En: *Actas de las XXVI Jornadas sobre Enseñanza Universitaria de la Informática* 5 (2020), págs. 29-36.
- [11] Vera Pospelova, Luis Fernández, Manuel de Buenaga y Ana Castillo. «Skills Match: cómo los datos abiertos permiten analizar la demanda de las habilidades no cognitivas en el mercado laboral». En: *Actas de las XXVI Jornadas sobre Enseñanza Universitaria de la Informática* 5 (2020), págs. 293-300.
- [12] Ken Schwaber y Jeff Sutherland. *The Scrum Guide*. 2020.

Análisis de la mejora de los resultados de una asignatura inicial de programación tras cambiar la planificación de actividades

Alberto Gómez¹, Maria-Jesús Marco-Galindo², Julià Minguillón², Jordi Escayola Mansilla²

¹Universidad de Extremadura, ²Universitat Oberta de Catalunya
Cáceres Barcelona

agomez@unex.es, mmarcog@uoc.edu, jminguillona@uoc.edu, jescayola@uoc.edu

Resumen

El objetivo de este trabajo es comprobar que los cambios introducidos en la planificación de actividades de evaluación continua de una asignatura de introducción a la programación han sido el factor determinante en la mejora de los resultados de los estudiantes. La evaluación continua pasó de tres entregas en el semestre a distribuir semanalmente los mismos contenidos y tareas en diez entregas. Aunque un análisis descriptivo de los datos muestra una mejora importante en la tasa de rendimiento tras el cambio, los estudiantes de cada semestre pueden tener perfiles distintos, circunstancia que puede influir también en los resultados. Así, se ha aplicado la técnica de análisis cuasi experimental *Propensity Score Matching* para poder comparar los resultados en estas poblaciones potencialmente distintas y comprobar si la mejora se debe efectivamente a la nueva planificación de actividades. Además, se han analizado varios modelos de regresión logística para intentar explicar la influencia del cambio de la planificación en la mejora de los resultados de la asignatura. A la vista del estudio realizado, se concluye que la temporalización semanal de las actividades ha supuesto un cambio positivo en los resultados de la asignatura.

Abstract

The aim of this project is to prove if the planification changes done in the continuous evaluation for the programming course is the key factor that explains the increase in the student's results. The continuous evaluation of this introductory programming course went from having three deliveries in the semester in the past to distribute the same contents and tasks in ten weekly deliveries. Although the descriptive analysis of the data shows a significant improvement in the performance rate after the change, the students in each semester could have different profiles that also influenced the results. The quasi-experimental analy-

sis technique *Propensity Score Matching* has been applied to be able to compare the results in these potentially different populations and check if the improvement is due to the new planning of activities. In addition, some logistic regression models have been analysed to explain the influence in the planification changes and its relationship with the results improvement. According to the obtained results, it is concluded that the weekly timing of the activities has led to a positive result in the students' performance.

Palabras clave

Propensity Score Matching, learning analytics, planificación de actividades, online education.

1. Introducción

El principal objetivo de este trabajo es comprobar si los cambios introducidos en la planificación de las actividades de evaluación de una asignatura de introducción a la programación en una universidad en línea han influido en la mejora observada en el rendimiento de los estudiantes.

El análisis de las tasas de rendimiento de los dos semestres anteriores y los dos posteriores a la intervención realizada muestra un aumento del porcentaje de estudiantes que supera la asignatura y un acusado descenso del porcentaje de no presentados en la evaluación continua.

Sin embargo, en el análisis de los datos hay que valorar el hecho de que los grupos de estudiantes de cada semestre tengan perfiles distintos puede influir también en la mejora observada. La población de estudiantes de cada semestre es diferente y, por tanto, no se deberían comparar directamente los resultados obtenidos en distintos semestres. Habrá que aplicar técnicas de análisis de datos que permitan comparar poblaciones potencialmente distintas. En nuestro caso se ha usado la técnica de análisis cuasi experimental denominada *Propensity Score Matching* (PSM).

Los diseños cuasi experimentales son una herramienta muy útil para poder comparar resultados cuando se hacen cambios en el diseño y planificación de asignaturas y titulaciones. De esta manera se puede evaluar el alcance de los cambios introducidos cuando los grupos son distintos.

Este trabajo se organiza de la siguiente forma: el apartado 2 revisa trabajos previos relacionados con la enseñanza de la programación y las técnicas de diseño cuasi experimentales. El apartado 3 describe el contexto de aplicación de esta intervención. En el apartado 4 se presenta un breve resumen del análisis de datos realizado. A continuación, se explica el emparejamiento de datos realizado (apartado 5) y los modelos explicativos construidos para determinar la influencia de la intervención realizada (apartado 6). Por último, en el apartado 7, se resumen las conclusiones principales del trabajo y se presentan líneas de trabajo futuras.

2. Estado de la cuestión

Las asignaturas de introducción a la programación suelen tener un alto nivel de abandonos y suspensos. Hay muchos trabajos que tratan de explicar los motivos del abandono, desarrollar estrategias para minimizarlo y, en general, mejorar el aprendizaje de esta materia básica para los cursos posteriores [15]. La motivación (ya sea intrínseca o extrínseca) se presenta como un aspecto fundamental para conseguir buenos resultados, especialmente si hablamos de estudios no presenciales. Como explica Bueno, “para motivarnos o para conseguir que los demás se motiven, hay que tener un objetivo concreto que sea tangible” [2]. Objetivos concretos, a corto plazo, mantienen y favorecen la motivación. Biggs y Tang indican que las actividades de evaluación a lo largo del curso mantienen la asistencia a clase y el trabajo con los materiales de la asignatura, además de servir para la evaluación formativa y la calificación [1].

Este hecho se puede extrapolar a la enseñanza virtual, donde una adecuada planificación de actividades es imprescindible para que el estudiante se mantenga motivado y participativo. Así, la evaluación formativa continua y el *feedback* que se proporciona a los estudiantes de las actividades que realizan es muy importante e influye en la mejora del rendimiento. Una mala planificación de las actividades de evaluación puede tener consecuencias negativas en estudiantes y profesorado [17]. Dar *feedback* se considera uno de los mejores métodos para incrementar el aprendizaje de los estudiantes [13,18].

La minería de datos educativa (*educational data mining*) es una disciplina que investiga la aplicación de algoritmos de minería de datos sobre distintos tipos de datos educativos para comprender mejor cómo es el proceso de aprendizaje del estudiante [16].

El uso de plataformas de aprendizaje y los estudios a través de Internet han incrementado mucho la cantidad de datos disponibles. Empleando distintas técnicas de análisis de datos, el comportamiento de los estudiantes a lo largo del curso dentro de la plataforma virtual puede dar información importante sobre qué actividades son más importantes, qué factores influyen más en los resultados o cómo dar mejor *feedback* [16]. La predicción del abandono o de los resultados es también un área de interés en la analítica del aprendizaje (*learning analytics*) [4,11]. El resultado del análisis de los datos se utiliza para proporcionar información relevante a los profesores, visualizar datos de estudiantes, construir sistemas de recomendación, predecir su desempeño, categorizarlos, detectar comportamientos no deseables (fraude), planificar actividades, etc. [16].

2.1. Diseño experimental y PSM

Para determinar la influencia de una intervención en un sistema se debe diseñar un experimento donde se establezcan grupos similares que se diferencien solo en ese punto que se está estudiando. Así se podrá analizar posteriormente la influencia de dicha intervención en los resultados obtenidos. En campos como la medicina, el diseño experimental debe ser siempre aleatorio, donde el grupo de control y el grupo sujeto del estudio se formen siguiendo una metodología estricta que asegure que no influyen otros factores y que el estudio no pierde validez [3]. En investigación en educación es difícil hacer diseños aleatorios de experimentos con grupos de control y selección aleatoria. Las normativas universitarias no permiten tratar a los estudiantes del mismo grupo de forma distinta. Además, no sería ético establecer grupos de control o realizar acciones que puedan perjudicar el aprendizaje de algunos estudiantes [6].

En muchos estudios con experiencias de innovación docente se comparan los resultados del grupo en el que se ha aplicado la intervención con los resultados de un grupo del curso anterior, para intentar demostrar que ha habido una mejora. En general, este análisis no es completamente válido porque los grupos no se han formado aleatoriamente y las poblaciones pueden ser distintas. En estos casos estamos ante cuasi experimentos donde no se puede seleccionar a los individuos que están en cada categoría analizada [6]. Por tanto, hay que utilizar las técnicas adecuadas para estudiar estas situaciones.

En los diseños cuasi experimentales, los métodos de emparejamiento por índice de propensión (*Propensity Score Matching*, PSM) son muy utilizados [7,14]. Con estos métodos se pretende reducir el posible sesgo provocado por las variables que intervienen en la formación de los grupos y así poder analizar los datos posteriormente como si provinieran de un ensayo aleatorio. El índice de propensión (*propensity*

score, *PS*) de una observación del conjunto de datos es la probabilidad o propensión de formar parte o no del grupo que ha sido sometido a la intervención, calculada a partir de las variables explicativas con las que se cuenta. Generalmente, los *PS* se calculan mediante un modelo de regresión logística con la participación o no en la intervención como resultado y algunas variables del modelo como predictoras.

Posteriormente, se emplea un algoritmo de emparejamiento de los datos con *PS* similares (*Propensity Score Matching*) a partir de los conjuntos de datos iniciales, rechazando los elementos que no se puedan emparejar. Con los elementos emparejados ya se puede analizar el impacto de la intervención como si los datos se hubieran conseguido en un experimento aleatorio. El emparejamiento entre los dos conjuntos puede hacerse atendiendo a distintos criterios de similitud que dará lugar a distintos conjuntos de datos emparejados. Además, se debe comprobar que no se han perdido demasiadas observaciones en el emparejamiento y que las características de los conjuntos que reciben o no la intervención están equilibradas [8]. Harris presenta una amplia revisión de la literatura donde se usa *PSM* en educación superior [7]. En [10] se muestra un caso donde se usa *PSM* para eliminar el sesgo producido al no haber podido formar el grupo de control de manera aleatoria.

3. Contexto

Los datos de este trabajo provienen de la asignatura "Fundamentos de programación" que se oferta como primera asignatura de programación en varios grados y programas formativos de la Universitat Oberta de Catalunya (UOC) [11].

El perfil de las personas que cursan estudios virtuales es muy variado, y puede cambiar mucho de un semestre a otro. La mayoría compatibiliza sus estudios con su trabajo, así que cursan pocas asignaturas cada semestre. En la UOC, cada estudiante consulta con su tutor la idoneidad de las asignaturas elegidas para evitar problemas de coordinación y ajustarlas a su tiempo de dedicación. Hay un proyecto institucional para coordinar asignaturas y mejorar la experiencia de los estudiantes en sus primeros cursos [5].

La mayoría de los estudiantes provienen de grados en Ingeniería en Informática. El modelo de calificación principal está basado en la evaluación continua. A lo largo del semestre se proponen una serie de actividades de evaluación continua (PAC) que ponen en práctica los principales conceptos que se van presentando en la asignatura. Además, los estudiantes deben entregar dos prácticas de programación más complejas que integran todas las competencias (PR1 y PR2). La realización de la primera práctica es requisito imprescindible para poder aprobar la asignatura. Cada semestre se desarrolla a lo largo de unas 15

semanas. La primera práctica se entrega aproximadamente en la novena semana del curso. La entrega de PR1 es un buen indicador de si el estudiante va a poder superar la asignatura o no. Desafortunadamente, hay un alto porcentaje de estudiantes que no llegan a entregarla.

Los cambios en la planificación implantados por el profesorado de la asignatura han estado encaminados a potenciar el trabajo continuo en la asignatura, con entregas más frecuentes y la obtención de *feedback* más a menudo. De esta forma se ha pretendido reducir el abandono y mejorar los resultados. Antes de la modificación de la planificación, los estudiantes entregaban tres pruebas de evaluación continua (PAC1 a PAC3) a lo largo del semestre, una cada 4 o 5 semanas. La entrega de PAC2 se realizaba poco antes de la entrega de PR1. Tras cada prueba, el estudiante recibía *feedback* general con una solución comentada y *feedback* personalizado con la corrección y calificación de su trabajo.

El cambio en la planificación supuso, principalmente, dividir cada PAC anterior en PAC más cortas, de forma que las entregas son más frecuentes. El tipo de problemas incluidos era el mismo. Simplemente se reorganizaron los mismos ejercicios en tres bloques con múltiples entregas. La PAC1 se dividió en 4 (nuevas PAC1 a PAC4), la PAC2 en 4 (nuevas PAC5 a PAC8) y la PAC3 se partió en 2 (nuevas PAC9 y PAC10). Con esta nueva planificación, las entregas pasan a ser semanales, recibiendo el estudiante *feedback* cada semana con la publicación de la solución. Además, hay que considerar también la propia percepción de progreso del estudiante a medida que avanza en sus tareas. El *feedback* individualizado del profesor y la calificación se han seguido entregando en tres momentos del semestre, después de cada bloque.

4. Análisis exploratorio de datos

Para este trabajo se ha contado con los datos de cuatro semestres: dos previos a la modificación de la planificación de actividades (segundo semestre del curso 2016-17 y primero del curso 2017-18) y dos posteriores a los cambios (segundo semestre del curso 2017-18 y primero del 2018-19). Los datos provienen del *Learning Record Store* institucional [12].

Se ha contado con un total de 1487 observaciones: 721 anteriores al cambio y 766, posteriores.

4.1. Variables

Los dos conjuntos de datos tienen las mismas variables que caracterizan al estudiante y a sus estudios. Solo se diferencian en las variables que hacen referencia a las calificaciones de las 3 PAC (datos previos al cambio, *pre*) o de las 10 PAC (datos posteriores al cambio, *post*).

Cada observación describe el perfil y calificaciones de un estudiante en un semestre. Las principales variables analizadas se pueden agrupar en distintas categorías:

- Perfil socio-demográfico: sexo, grupo de edad.
- Perfil de estudios: grado en el que está matriculado, vía de acceso a la titulación, si proviene de estudios de formación profesional o de estudios universitarios, número de asignaturas matriculadas en el semestre, número de semestres que lleva matriculado en el grado, veces que ha repetido la asignatura.
- Calificaciones obtenidas: calificaciones de cada PAC y de PR1 y PR2, calificación final de prácticas y de evaluación continua.

Las calificaciones vienen dadas con una escala que, ordenada de mayor a menor, es: *A*, *B*, *Cm*, *Cn*, *D*, *N*. La calificación *N* supone que no se entregó la PAC correspondiente; *Cn* y *D* es suspenso; el resto son valores aprobados.

En el análisis previo de los datos se recodificaron algunas variables para agrupar categorías que contaban con pocas observaciones

Además, se ha añadido la variable *TRATAM* para separar las observaciones de los semestres previos al cambio (con valor 0) y los posteriores (con valor 1).

4.2. Comparación de los resultados

El objetivo principal del estudio es comprobar si los cambios en la planificación de las actividades de evaluación continua han provocado la mejora de los resultados y una menor tasa de abandono.

La entrega de la primera práctica (variable *faPR1*) es uno de los indicadores más claros del rendimiento final del estudiante en la asignatura. De las personas que entregan la práctica 1, aprueban las prácticas el 83.4% y superan la evaluación continua el 93.3%. Solo el 21% de los que no entregan la práctica superan la evaluación continua de la asignatura (aunque no pueden aprobar la asignatura al ser la práctica un requisito). Así, el porcentaje de *PR1* presentadas ha aumentado en los semestres posteriores al cambio en la planificación pasando del 48.13% al 60.70%.

Muchos de los que entregan *PR1* aprueban las prácticas *CPR* y la evaluación continua *CPAC*, sobre todo en los datos posteriores al cambio.

Datos	faPR1 = 1	Y CPR aprobada	Y CPAC	Am bas
pre	347	244	315	234
post	465	441	443	426

Cuadro 1: Relación entre *faPR1* y aprobados

El cambio podría haber mejorado también las calificaciones finales de evaluación continua *CPAC*. El

cuadro 2 muestra las frecuencias relativas para los dos conjuntos de datos.

Datos	A	B	Cm	Cn	D	N
pre	0.16	0.12	0.13	0.13	0.07	0.39
post	0.16	0.24	0.18	0.09	0.19	0.14

Cuadro 2: Distribución de notas *CPAC*

Se puede observar un incremento de los aprobados, especialmente de las calificaciones *B* y *Cm*. Y, sobre todo, hay un importante descenso de no presentados. El cambio en la planificación parece haber mejorado también las calificaciones finales de las prácticas *CPR*, tal como muestra el cuadro 3. Se aprecia el incremento en aprobados y el descenso en suspensos y no presentados.

Datos	A	B	Cm	Cn	D	N
pre	0.09	0.15	0.08	0.02	0.12	0.54
post	0.27	0.23	0.06	0.02	0.03	0.39

Cuadro 3. Distribución de notas *CPR*

Todos los estudiantes que superan la práctica pueden optar a aprobar la asignatura por examen final, aunque no hayan superado *CPAC*, así que el incremento de personas que superan *CPR* es importante.

4.3. Evolución de las calificaciones

Resulta interesante comparar la evolución de las calificaciones de los estudiantes en las pruebas de evaluación continua antes y después de los cambios en la planificación. En las figuras 1 y 2 se puede ver la evolución de las calificaciones hasta el momento de la presentación de la práctica 1 en ambos conjuntos de datos. En verde están representados los estudiantes que entregan la *PR1*, y en marrón, los que no.

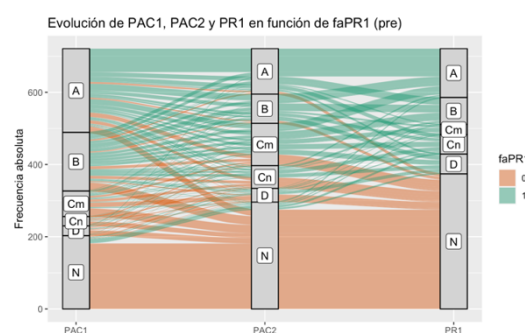


Figura 1. Evolución de las calificaciones (pre)

En ambos gráficos se observa que las calificaciones de los estudiantes no varían demasiado a lo largo del semestre. Solo cambian mucho si abandonan la asignatura en alguna entrega; si no, la mayoría se mueve entre grados de calificación contiguos.

Se puede apreciar cómo hay estudiantes que, poco a poco, van abandonando la asignatura. Y son pocos

los que, después de no presentar alguna PAC, se recuperan y consiguen presentar la práctica. Se puede ver también que, tras los cambios en la planificación, el porcentaje de personas que abandonan la asignatura en las primeras entregas se ha reducido.

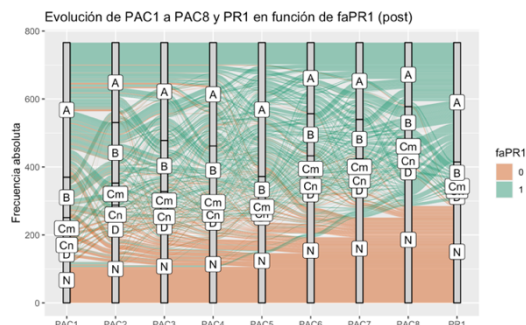


Figura 2. Evolución de las calificaciones (post)

4.4. Análisis exploratorio

También se ha realizado una exploración exhaustiva de las relaciones entre las variables disponibles y no se han encontrado interacciones claras. No se aprecian grandes diferencias en las características generales de los conjuntos de datos anteriores y posteriores a la intervención, más allá de las diferencias en las calificaciones. A partir del análisis exploratorio de los datos parece razonable pensar que el cambio en la planificación de las actividades de evaluación ha mejorado las calificaciones y ha disminuido el abandono de la asignatura. Sin embargo, es necesario preguntarse si esa mejora es significativa, dado que las poblaciones comparadas son distintas.

5. Emparejamiento de los datos

Como los datos provienen de semestres distintos, es posible que la distribución de las variables que definen los perfiles de los estudiantes sean diferentes y, por tanto, los resultados de la evaluación estén sesgados por las diferencias en los conjuntos de datos y no solo por la influencia de la intervención realizada. La técnica cuasi experimental *Propensity Score Matching* permite conseguir datos comparables previos y posteriores a la intervención, y así poder analizar el impacto de la intervención como si los datos provinieran de un estudio aleatorio.

Existen muchos algoritmos para emparejar las observaciones que tienen PS similares. En este estudio se han probado varios de los algoritmos más usuales para comparar los conjuntos de datos emparejados que se consiguen y seleccionar los que presentaban un grado mayor de equilibrio.

Los algoritmos que se han usado, según el nombre en el paquete MatchIt [9] de R con el que se han desarrollado los cálculos, son los siguientes:

- *Exact Matching*: se emparejan las observaciones que tienen exactamente los mismos valores en todas las variables.
- *Subclassification*: se forma un número determinado de clases disjuntas en las que la distribución de covariables es tan similar como sea posible.
- *Nearest Neighbor Matching*: se usa el PS para emparejar cada observación de un conjunto con el que tenga un valor más parecido en el otro. Se hace la asignación en orden, con un algoritmo voraz, que a veces empareja observaciones con valores muy distintos. En estos casos se puede usar una opción adicional, un calibre (*caliper*) que obliga a que la diferencia entre los valores emparejados sea menor que la proporción indicada de la desviación típica de las distancias calculadas.
- *Full Matching*: se forman tantas clases disjuntas como sea necesario. En cada clase hay una observación de uno de los dos conjuntos y tantas del otro conjunto como coincidan con ella.

5.1. Equilibrio tras el emparejamiento

Aunque se han realizado pruebas con todos los algoritmos de emparejamiento, los resultados no son adecuados en algunos de ellos. Por ejemplo, con el algoritmo *Exact Matching*, aunque el equilibrio final es absoluto, se pierde más de la mitad de las observaciones, al emparejar solo las observaciones idénticas de cada conjunto. En todos los métodos se han descartado 8 casos del conjunto de datos *post* porque los valores de PS están fuera de la región de soporte común. Este valor tan bajo de observaciones descartadas ya da una idea de que las poblaciones son bastante similares incluso sin emparejamiento. Los resultados han sido adecuados, con pocas pérdidas de datos y variables finales equilibradas, en el caso de los algoritmos *Nearest Neighbor Matching* con calibres 0.1 y 0.2, y con el algoritmo *Full Matching*, que proporciona los mejores resultados. Solo descarta 8 observaciones y el equilibrio es adecuado en todas las covariables. La distribución de las PS emparejadas y descartadas se puede ver en la figura 3.

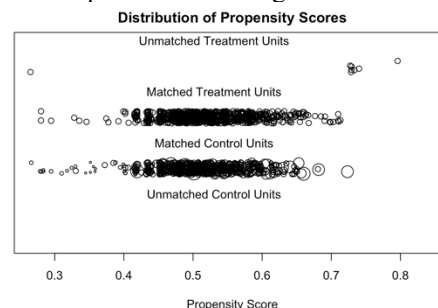


Figura 3. Distribución de PS

Todas las covariables están equilibradas tras el emparejamiento, como puede apreciarse en la figura 4.

Con los datos resultantes tras este emparejamiento equilibrado ya se puede realizar el estudio comparativo que se desee, considerando que los datos que tenemos provienen, a efectos prácticos, de una población homogénea.

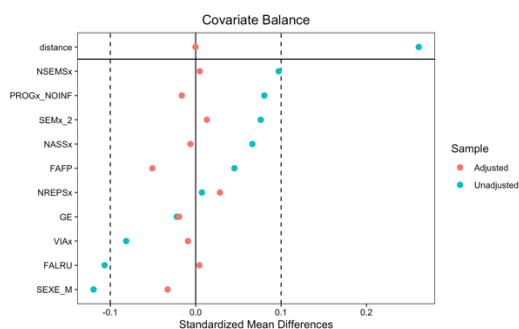


Figura 4. Equilibrio tras el emparejamiento

6. Modelos explicativos

Una vez realizado el proceso de emparejamiento, se puede trabajar con estos datos como si procedieran de un experimento aleatorio. En nuestro caso, el grupo de control está formado por los estudiantes que cursaron la asignatura en los dos primeros semestres, antes del cambio en la planificación. El grupo que ha recibido el tratamiento estaría formado por los estudiantes que han cursado la asignatura en los dos semestres posteriores.

El objetivo ahora es determinar si el cambio en la planificación ha influido positivamente en la superación de la asignatura. Más concretamente, interesa saber si este cambio ha disminuido de forma significativa el abandono de la evaluación continua en las primeras semanas del curso y si ha aumentado el porcentaje de personas que presentan la primera práctica de la asignatura, requisito para poder aprobar la asignatura y un buen indicador del resultado final.

Para ello se han desarrollado y analizado una serie de modelos de regresión logística donde se explorará la influencia de la variable *TRATAM*. Los modelos de regresión logística sirven para predecir el resultado de variables categóricas binarias.

Estos modelos, en función de las variables consideradas, calculan la probabilidad de que un individuo termine en una categoría u otra de la variable objetivo.

6.1. Homogeneización de calificaciones

Para poder desarrollar modelos de regresión con los datos disponibles hay que tener en cuenta que no hay el mismo número de variables *PAC* en los dos conjuntos de datos y que las variables *PAC1* y *PAC2* no hacen referencia a actividades directamente compara-

bles en ambos conjuntos de datos. Se ha creado una nueva variable *NOTAPAC1* para homogeneizar las calificaciones obtenidas durante las primeras semanas del curso. *NOTAPAC1* tiene el valor de la calificación de la PAC 1 en el conjunto de datos previo y la media aritmética de las cuatro primeras PAC en el conjunto de datos posterior. Los contenidos que se califican coinciden. También se ha definido la variable *faNOTAPAC1* como indicador del seguimiento de la evaluación continua durante las primeras cuatro actividades. Será 0 si no se ha presentado nada de la evaluación continua y 1 en caso contrario.

6.2. Modelos analizados

Se han desarrollado distintos modelos de regresión logística con los datos obtenidos tras la fase de PSM con los algoritmos *Full Matching* y *Nearest Neighbor Matching* con calibre 0.2, el más utilizado en la literatura. Los resultados han sido similares.

Los modelos de regresión se han construido para determinar la influencia directa de las variables en la variable binaria elegida como objetivo, así como las posibles interacciones. Aunque eran especialmente interesantes las interacciones de la variable de control *TRATAM* con las calificaciones, también se han explorado otras relaciones con variables como el sexo, programa de estudios, vía de acceso, etc. El análisis de la influencia se suele hacer a partir de los *odds ratio*, la exponencial de los coeficientes del modelo. Para cada incremento de una unidad en una variable cuantitativa, el *odds ratio* se puede interpretar como la oportunidad de encontrar el valor esperado entre los individuos con ese incremento en el valor de la variable, frente a los individuos que no lo muestran. En las variables categóricas, el incremento se refiere al paso desde el valor base a cada uno de los valores de las variables ficticias.

A continuación, se presentan dos de los modelos desarrollados con una capacidad de explicación buena y con indicadores de validez y calidad adecuados.

6.3. Seguimiento de la evaluación continua

En el primer modelo se intenta explicar la variable que indica si se han presentado las actividades de las primeras semanas del curso, *faNOTAPAC1*, a partir de la información del perfil del estudiante. De forma resumida, el pseudoajuste R^2 es muy bajo, lo que representa un mal ajuste del modelo. Las únicas variables significativas en este modelo son *TRATAM* (que indica si pertenece o no a los semestres posteriores al cambio) y *NREPSx* (el número de repeticiones de la asignatura). A medida que aumenta el número de veces que se repite la asignatura parece que disminuye la probabilidad de seguir la evaluación continua. Pertenecer a los semestres posteriores al cambio influye en una menor probabilidad de haber abando-

nado la asignatura desde las primeras semanas. Ninguna de las interacciones entre variables exploradas en este modelo ha resultado significativa.

Este primer modelo, aunque no demasiado bueno, da una primera idea de que el cambio de planificación de las actividades ha supuesto un aumento de la probabilidad de seguir la evaluación continua y de no abandonar en las primeras semanas.

6.4. Presentación de la PR1

Como la entrega de la primera práctica es un buen indicador del seguimiento de la asignatura, se ha intentado explicar esa variable *faPR1* en función del resto de variables de perfil y *NOTAPAC1*. También se han explorado algunas interacciones entre variables (como estar en una titulación de Ingeniería Informática con la experiencia universitaria previa o con el número de semestres en la titulación), pero la única significativa es la interacción entre *NOTAPAC1* y la variable *TRATAM*. En este modelo, mostrado en el cuadro 4, el ajuste es adecuado.

Predictors	faPR1		
	Odds Ratios	CI	p
(Intercept)	0.20	0.08 – 0.49	<0.001
TRATAM	0.74	0.37 – 1.45	0.377
SEXE [M]	0.70	0.48 – 1.01	0.055
GE	0.75	0.66 – 0.85	<0.001
PROGx [NOINF]	1.13	0.73 – 1.74	0.586
EXP [SIN]	1.08	0.74 – 1.58	0.681
FAFP	0.65	0.42 – 1.00	0.050
FALRU	1.04	0.28 – 3.91	0.958
NASSx	1.11	0.98 – 1.26	0.099
NSEMSx	1.10	0.93 – 1.30	0.290
NREPSx	0.67	0.51 – 0.87	0.003
SEMx	0.88	0.66 – 1.16	0.360
NOTAPAC1	1.59	1.48 – 1.70	<0.001
TRATAM * NOTAPAC1	1.19	1.08 – 1.33	0.001
PROGx [NOINF] * NSEMSx	1.16	0.92 – 1.46	0.221
PROGx [NOINF] * EXP [SIN]	1.06	0.56 – 1.99	0.859
Observations	1479		
R ² Tjur	0.444		

Cuadro 4. Resultado del modelo logístico para *faPR1*

Las variables que resultan significativas son el grupo de edad, si se proviene de estudios de Formación Profesional o no, el número de repeticiones de la asignatura, *NOTAPAC1* y la interacción entre *TRATAM* y *NOTAPAC1*.

Además, hay que tener en cuenta que, mientras las primeras son variables dicotómicas o con pocos

valores, *NOTAPAC1* tiene un rango de valores superior (entre 0 y 10). Por tanto, la interacción con la variable *TRATAM* puede suponer un incremento de la probabilidad de entregar la práctica mucho mayor. Un *odds ratio* superior a uno supone un incremento de la probabilidad de que el valor de *faPR1* sea 1 con un cambio de la variable correspondiente. Un valor inferior a uno indica que la probabilidad disminuye. Por tanto, según este modelo, a medida que aumenta la calificación de *NOTAPAC1* (la *PAC1* de antes del cambio o la media aritmética de las 4 primeras *PAC* tras el cambio), la probabilidad de presentar PR1 aumenta. Además, esa influencia se ve reforzada en los semestres posteriores al cambio por la interacción positiva entre *TRATAM* y *NOTAPAC1*.

El *odds ratio* de la variable *TRATAM* es menor que 1, lo que parecería indicar que la intervención disminuye la probabilidad de presentar la práctica 1, justo lo contrario de que lo queríamos conseguir. Sin embargo, la interacción de *TRATAM* con *NOTAPAC1* es positiva, así que ese coeficiente negativo de *TRATAM* solo suaviza un poco el valor de la interacción. Además, no resulta significativo en este modelo.

7. Conclusiones y trabajo futuro

En este trabajo se han analizado los datos de cuatro semestres de los estudiantes que han cursado la asignatura "Fundamentos de programación" para determinar si los cambios en la planificación de las actividades han supuesto una mejora en el rendimiento y una disminución del abandono en la asignatura. Aumentar la frecuencia de las entregas ha mejorado las tasas de seguimiento de la evaluación continua y de presentación de la primera práctica de la asignatura, lo que repercute en un mayor índice de superación de la asignatura. Esta mejora está alineada con los resultados que se presentan en la literatura de investigación en educación en informática [1,13,15,18].

El aumento del *feedback* mejora los resultados y disminuye el abandono de los estudiantes. Con las entregas frecuentes de actividades, los estudiantes reciben *feedback* grupal más frecuentemente, además de tener que gestionar mejor su tiempo. Aunque el *feedback* personalizado por parte del profesorado no se ha incrementado, el estudiante obtiene retroalimentación al ir consiguiendo entregar a tiempo las actividades y puede pedir ayuda antes si es necesario.

La comparación directa de los resultados de semestres distintos no sería correcta porque los perfiles de los estudiantes pueden variar de un semestre a otro. Ante la imposibilidad de un diseño experimental aleatorio en esta intervención, se ha usado la técnica cuasi experimental *Propensity Score Matching* para construir un conjunto de datos equilibrado. Las poblaciones de estudiantes en cada semestre son similares, como se deduce del resultado del PSM.

Los modelos de regresión logística analizados permiten confirmar la influencia positiva de la nueva planificación de la asignatura en la disminución del abandono y en el aumento del seguimiento de la evaluación continua y de la entrega de la primera práctica, lo que influye en las tasas de éxito.

Aunque los modelos de regresión logística permiten valorar qué variables e interacciones influyen más en la entrega de la práctica, la capacidad predictiva tiene mucho margen de mejora. No se cuenta con todas las variables que influyen en el seguimiento de la asignatura por parte del estudiante. Por ejemplo, no se dispone de información sobre horas de dedicación y hábitos de estudio, lo que sería muy interesante analizar en futuros trabajos.

Por otro lado, los perfiles de los estudiantes no son homogéneos en todos los programas de estudio. Con un análisis de los programas con más estudiantes (Ingeniería Informática, especialmente), se podrían obtener datos más ajustados y mejores modelos.

Ahora que se ha comprobado que este nuevo modelo de planificación ha influido positivamente, se podrían usar los datos para construir modelos de predicción más completos que permitan construir perfiles de estudiantes según su comportamiento, detectar en qué actividades aumenta el riesgo de abandono o en cuáles suspenden más, para así poder modificarlas y prestar más ayuda a cada estudiante.

Referencias

- [1] John Biggs y Catherine Tang. 2003. *Teaching for Quality Learning at University*. SRHE and Open University Press.
- [2] David Bueno. 2017. *Neurociencia para educadores*.
- [3] Glynis Cousin. 2009. *Researching Learning in Higher Education: An Introduction to Contemporary Methods and Approaches*. Taylor & Francis.
- [4] A. Dinesh Kumar, R. Pandi Selvam, y K. Sathesh Kumar. 2018. Review on prediction algorithms in educational data mining. *International Journal of Pure and Applied Mathematics*.
- [5] Loles González, Julià Minguillón, Josep Antoni Martínez-Aceituno, y Julio Meneses. 2018. Institutional support to provide freshmen with flexible learning paths at course and semester level in open higher education. En *Proceedings of the 10th European Distance and E-Learning Network Research Workshop, 2018 Barcelona, 24-26 Octubre, 2018 ISBN 978-615-5511-25-7*.
- [6] Patricia Haden. 2019. Descriptive Statistics. En *The Cambridge Handbook of Computing Education Research*. Cambridge University Press, 102-132.
- [7] Heather D Harris. 2015. Propensity score matching in higher education assessment. *Masters Theses*.
- [8] Heather Harris y S. Jeanne Horst. 2016. A brief guide to decisions at each step of the propensity score matching process. *Practical Assessment, Research and Evaluation* 21, 4.
- [9] Daniel E Ho, Kosuke Imai, Gary King, y Elizabeth A Stuart. 2011. {MatchIt}: Nonparametric Preprocessing for Parametric Causal Inference. *Journal of Statistical Software* 42, 8: 1-28.
- [10] Lisa Angelique Lim, Sheridan Gentili, Abelardo Pardo, Vitomir Kovanović, Alexander Whitelock-Wainwright, Dragan Gašević, y Shane Dawson. 2019. What changes, and for whom? A study of the impact of learning analytics-based process feedback in a large course. *Learning and Instruction*.
- [11] Maria-Jesús Marco-Galindo, Julià Minguillón, y Teresa Sancho-Vinuesa. 2020. Análisis de la progresión de los estudiantes en una asignatura introductoria a la programación mediante redes bayesianas. *Actas de las Jornadas sobre Enseñanza Universitaria de la Informática* 5.
- [12] Julià Minguillón, Jordi Conesa, M Rodríguez, y F Santanach. 2018. Learning Analytics in Practice: Providing E-Learning Researchers and Practitioners with Activity Data: Emerging Technologies for Teaching and Learning. En *Lecture Notes in Educational Tech*. 145-167.
- [13] Margaret Price, Karen Handley, y Jill Millar. 2011. Feedback: Focusing attention on engagement. *Studies in Higher Education* 36, 8: 879-896.
- [14] Claudia Ramírez Ovalle. 2015. Sobre la Técnica de Puntajes de Propensión (Propensity Score Matching) y sus usos en investigación en Educación. *Educación y Ciencia* 4, 43:81-89.
- [15] Anthony V. Robins. 2019. Novice Programmers and Introductory Programming. En *The Cambridge Handbook of Computing Education Research*. Cambridge University Press, 327-376.
- [16] Cristóbal Romero y Sebastián Ventura. 2010. Educational data mining: A review of the state of the art. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews* 40, 601-618.
- [17] Jesús Serrano Guerrero, Francisco Pascual Romero Chicharro, y José Ángel Olivas Varela. 2011. La sobre-evaluación: efectos negativos de una mala planificación de la evaluación. *ReVisión* 4, 2.
- [18] Valerie J. Shute. 2008. Focus on Formative Feedback. *Review of Educational Research* 78, 1: 153-189.

Artículo pendiente de revisión

Artículo pendiente de revisión

Artículo pendiente de revisión

Artículo pendiente de revisión

Artículo pendiente de revisión

Artículo pendiente de revisión

Artículo pendiente de revisión

Artículo pendiente de revisión

Aprendizaje basado en metodologías ágiles centradas en diseño evolutivo dirigido por pruebas de aceptación

Alberto González Pérez, Ramón A. Mollineda Cárdenas, David Llorens Piñana

Departament de Llenguatges i Sistemes Informàtics

Universitat Jaume I

Castelló de la Plana

{alberto.gonzalez, ramon.mollineda, david.llorens}@uji.es

Resumen

Este artículo presenta una experiencia de aprendizaje basado en proyecto a partir de la coordinación docente entre dos asignaturas del Grado en Ingeniería Informática de la Universitat Jaume I, con el objetivo principal de mejorar competencias prácticas en el uso de metodologías ágiles de desarrollo de software muy difíciles de adquirir en asignaturas aisladas. La propuesta consiste en un proyecto de prácticas compartido entre las asignaturas Diseño de software y Paradigmas de software, las cuales se imparten en el primer cuatrimestre del cuarto curso en la intensificación en Ingeniería de Software. La primera asignatura introduce fundamentos de diseño de software, mientras que la segunda estudia la metodología ágil Desarrollo Dirigido por Pruebas de Aceptación (ATDD, de *Acceptance Test Driven Development*). El proyecto fue concebido para promover estrategias de diseño evolutivo de arriba a abajo centradas en la gestión eficiente de dependencias, según necesidades de usuarios formuladas en términos de pruebas de aceptación ejecutables escritas antes de diseñar el código objetivo. La especificación incluyó el uso de tecnologías de desarrollo web, aplicaciones móviles y servicios en la nube, contexto en el que se generaron escenarios ricos en gestión de dependencias desde la doble perspectiva del diseño y de la validación del software. Además de fomentar valores de la cultura ágil, la propuesta pretendía eliminar tareas redundantes (presentes en proyectos diferentes) y ofrecer una experiencia más cercana al desarrollo de soluciones profesionales. Los resultados de una encuesta revelaron un alumnado motivado con un proyecto realista, así como la percepción mayoritaria de haber experimentado principios claves del diseño y desarrollo ágil bajo condiciones de incertidumbres.

Abstract

This paper presents a project-based learning experience resulting from the coordination between two subjects of the Degree in Computer Engineering at the

Universitat Jaume I, with the main objective of improving practical skills in the use of agile software development methodologies, which are very difficult to acquire in isolated subjects. The proposal consists in a shared project between the subjects Software Design and Software Paradigms, which are taught in the same semester of the fourth year within the Software Engineering intensification. The first subject introduces software design fundamentals, while the second one studies the agile methodology Acceptance Test Driven Development (ATDD). The project was conceived to promote top-down evolutionary design strategies focused on the efficient management of dependencies, driven by user needs formulated in terms of executable acceptance tests written before designing the target code. The specification included the use of web development technologies, mobile applications and cloud services, a context in which rich dependency management scenarios were generated from the dual perspective of software design and validation. In addition to promoting agile culture values, the proposal was aimed to eliminate redundant tasks (present in different projects) and to offer an experience closer to the development of professional solutions. The results of a survey revealed students motivated with a realistic project, as well as a widespread perception of having experienced key principles of agile design and development under conditions of uncertainty.

Palabras clave

Aprendizaje basado en proyectos, diseño y desarrollo ágil, desarrollo dirigido por pruebas de aceptación.

1. Introducción

El desarrollo ágil de software puede definirse como un proceso iterativo dirigido a producir versiones incrementales del software objetivo con valor creciente para el usuario final. Las metodologías ágiles han inspirado numerosas estrategias formativas, y algunos de

sus principios han sido transferidos a contextos educativos en forma de competencias [9]. Entre los valores que promueve la cultura ágil, resumidos a través de doce principios del Manifiesto Ágil [1], destacan el trabajo en equipo, la responsabilidad individual y colectiva en la toma de decisiones, la interacción entre personas, la mejora continua, la gestión de incertidumbres y el uso eficiente de recursos. Una adaptación de estos valores al entorno del aula es explícitamente propuesta en [12], siendo cada principio reinterpretado en forma de actividades educativas específicas. Estos principios han sido principalmente usados en experiencias de aprendizaje basado en proyectos de Ingeniería de software [5,8], aunque también existen esfuerzos en Ingeniería eléctrica [11] y Matemática Discreta [3].

En el caso particular de la metodología ágil *Desarrollo Dirigido por Pruebas de Aceptación* (ATDD) [7], cada iteración es guiada por una secuencia de pruebas de aceptación que se definen e implementan antes de diseñar el código que se pretende validar, lo cual plantea a los alumnos un paradigma de programación poco convencional. Esta forma de desarrollo, a partir de pruebas tempranas definidas de forma conjunta por el cliente y el personal técnico, contribuye a clarificar requisitos, proporciona una medida objetiva del progreso del proyecto y fomenta diseños simples centrados en las necesidades del usuario.

Cada iteración ágil involucra todas las fases de desarrollo, en particular, el diseño de software que, en este contexto, se conoce como diseño evolutivo. Es decir, en cada iteración se toman decisiones de diseño desconociendo requisitos aún no tratados (que se analizarán en iteraciones futuras), por lo que deberá mantenerse una estructura de código flexible que admita nuevos servicios con bajo impacto en el diseño existente. Esta práctica no implica necesariamente mayores esfuerzos, siempre que el diseño se base en buenos principios como los agrupados bajo el acrónimo SOLID: 1) entidades con responsabilidad única, 2) entidades abiertas a extensión, pero cerradas a modificación, 3) diseño por contrato (los tipos bases deberían poder sustituirse por subtipos derivados), 4) principio de inversión de dependencia (las entidades de alto nivel no dependen de entidades de bajo nivel, sino de interfaces), y 5) principio de segregación de interfaces (las clases clientes no dependen de interfaces que no usan). Además de beneficios para el diseño, estos principios permiten una validación eficiente del software mediante dobles de prueba (*mocks*), y la creación de conjuntos de pruebas simples y fáciles de mantener.

Este paradigma puede reinterpretarse como diseño dirigido por pruebas, es decir, diseñar a partir de necesidades de usuarios promoviendo diseños flexibles basados en entidades simples cohesionadas y mínimamente acopladas. Esta perspectiva de análisis justifica la definición de proyectos integradores centrados en pruebas y en diseño del software, los cuales podrían

contribuir a adquirir competencias muy difíciles de trabajar en asignaturas aisladas. Desde el punto de vista del diseño [6], este tipo de proyectos ofrece oportunidades para trabajar la interacción del usuario con el sistema, el prototipado (simulación de alternativas de diseño, validación de ideas, etc.), el diseño visual (estética, consistencia, propósito, etc.), la redacción de documentación e instrucciones concisas y útiles (especificaciones, manual de usuario, etc.), la estructuración de contenidos (modelos, taxonomías, navegación, espacios de nombre, etc.), el diseño basado en las necesidades de usuarios, entre otras capacidades.

El aprendizaje basado en proyectos que aprovechan sinergias naturales entre asignaturas es una práctica que se ha mostrado efectiva en la formación de los alumnos de grados en Ingeniería Informática [10,11]. Un proyecto vertebrado en torno a la metodología ágil *Scrum* [13] que involucra tres asignaturas fue propuesto en [10], en el marco de la intensificación en Ingeniería del Software del Grado en Ingeniería Informática de la Universidad de Cantabria. Los autores plantearon una integración parcial de las asignaturas, a través de un proyecto común durante una parte del cuatrimestre. El proyecto fue desarrollado en equipos de 6 alumnos que trabajaron indistintamente en los horarios de las tres asignaturas con ajuste al modelo Scrum. Aunque se destaca la importancia de las pruebas en el desarrollo del proyecto, a diferencia de nuestra propuesta, no parece estar dirigido por pruebas de aceptación tempranas, y omite detalles sobre el tipo y alcance de las pruebas definidas, las metodologías involucradas, las tecnologías de automatización, y los beneficios de la definición temprana de pruebas más allá de la validación de la calidad funcional del software. En [4], la coordinación también se articula a través de un proyecto compartido por tres asignaturas. Su principal novedad es que estas asignaturas pertenecen a cursos distintos, por lo que los alumnos desempeñan un rol acorde con su nivel de formación. A diferencia de nuestra propuesta, el desarrollo no sigue una metodología ágil, y las pruebas se definen en las últimas etapas del ciclo de vida. Tampoco hay información sobre el tipo y alcance de las pruebas.

Desde la perspectiva de la formación de ingenieros informáticos, la novedad de esta iniciativa de coordinación docente consiste en la definición de un proyecto poco convencional de desarrollo de software en el marco de las metodologías ágiles, centrado en la creación y gestión de un diseño evolutivo que debe ir satisfaciendo progresivamente las necesidades de los usuarios representadas a través de pruebas de aceptación tempranas y automatizadas. Estas pruebas se entenderán como una especificación formal de necesidades de usuarios. Como se ha comentado, se espera que esta dinámica favorezca competencias difíciles de adquirir en asignaturas aisladas, así como diseños simples, efectivos y bien cohesionados.

2. Contexto

Las asignaturas *Diseño de software* y *Paradigmas de software* forman parte de la intensificación de Ingeniería de Software del Grado en Ingeniería Informática de la Universitat Jaume I, y se imparten en el primer cuatrimestre del cuarto curso del grado.

Ambas asignaturas siguen una planificación docente muy similar: las primeras semanas del curso se dedican a impartir clases magistrales y seminarios en los que los alumnos aprenden fundamentos teóricos. En el caso de la asignatura *Diseño de software*, estas primeras clases se acompañan de sesiones prácticas en las que realizan ejercicios (de alcance reducido) relacionados con los conceptos revisados en teoría.

Antes del curso 2020-2021, tras las primeras semanas, ambas asignaturas proponían un proyecto realista de desarrollo de software de tamaño medio. En el caso de la asignatura *Diseño de software*, el proyecto constaba habitualmente de varias aplicaciones. El desarrollo comenzaba con una etapa inicial de análisis de software, que incluía obtención de requisitos, descripción de casos de uso y definición de historias de usuario. A continuación, los alumnos priorizaban las historias de usuario, antes de proceder al diseño e implementación del proyecto considerando dos requisitos técnicos fundamentales: incluir al menos dos patrones de diseño, y definir interfaces gráficas con usabilidad aceptable.

En el caso de la asignatura de *Paradigmas de software*, se proponía a los alumnos la elaboración de un proyecto compuesto por una única aplicación, aunque esta debía consumir servicios externos que proporcionaran información real para acercarla a una solución real, por ejemplo, la previsión meteorológica. El proyecto debía desarrollarse siguiendo la metodología ágil ATDD, comenzando por un proceso de análisis de software consistente en la obtención de requisitos y definición de historias de usuario. Las historias se descomponían en escenarios de uso/prueba, a partir de los cuales se identificaban ejemplos concretos de uso del software que se convertían en pruebas de aceptación. Estas pruebas se automatizaban y se usaban como guía para el desarrollo iterativo (cada iteración requería superar una nueva prueba), y como criterio de validación objetiva de la historia y el escenario vinculado. Además, se pedía realizar pruebas de integración sobre aquellos componentes software que hicieran uso de las dependencias externas (e.g. bases de datos, servicios web, etc.), mediante la sustitución de estas últimas por dobles de prueba (*mocks*). Este objetivo necesita del patrón de inyección de dependencias, lo cual "forzaba" a los alumnos a mantener una estructura de código simple y flexible, resultado de aplicar buenos principios de diseño. Además, se promovía la creación de pruebas de unidad siguiendo el método de Desarrollo Dirigido por Pruebas (TDD, *Test Driven Development*), así como el diseño de interfaces gráficas simples y con buena usabilidad (a criterio del profesor).

En resumen, los alumnos debían atender dos proyectos distintos de desarrollo de software en un mismo período docente, con exigencias técnicas complementarias o redundantes. Esta situación provocaba una serie de inconvenientes, algunos identificados en [10], los cuales se resumen a continuación:

- *Actividades redundantes.* Ambos proyectos incluían sendas fases de análisis de software que coincidían en la obtención de requisitos y en la definición de historias de usuario. Por experiencias anteriores, este análisis puede llegar a consumir hasta un tercio del tiempo previsto para cada proyecto, reduciendo así considerablemente el tiempo disponible para completar las tareas con competencias propias de las asignaturas, por ejemplo, diseño de software, escritura de pruebas, etc. Ambos proyectos también coincidían en exigir un diseño de calidad de la lógica del software, así como interfaces gráficas de usuario con arreglo a buenos criterios de usabilidad. Estas duplicidades tenían un efecto desmotivador y generaban una gestión ineficiente de recursos.
- *Falta de visión integradora.* Al finalizar el cuatrimestre, los proyectos de la asignatura *Paradigmas de software* presentaban una calidad notable, validada por pruebas de aceptación, integración y unitarias. Sin embargo, la cantidad y complejidad de las funcionalidades implementadas eran escasas, reduciendo el valor práctico del resultado. En cambio, los proyectos de *Diseño de software* solían tener una gran complejidad funcional, así como un diseño de interfaces mucho más elaborado. No obstante, estas funcionalidades no venían (habitualmente) acompañadas de pruebas sistemáticas y automatizadas. La realización de ambos proyectos por separado desaprovechaba oportunidades de ofrecer a los alumnos una visión más realista del desarrollo de software.
- *Sinergias no aprovechadas.* Un ciclo de vida ágil suele promover el diseño de la solución y la definición de pruebas conjuntamente. Hasta ahora, cada asignatura se centraba en una de las dos áreas, mientras que la otra jugaba un rol de mero acompañamiento. Además, la relación entre la calidad del diseño y la necesidad de escribir pruebas no invasivas era poco considerada. De la misma forma, la capacidad de las pruebas de aceptación tempranas para fomentar diseños centrados en las necesidades del usuario, era desaprovechada.
- *Mala gestión del tiempo.* Dos proyectos distintos y complejos, que debían desarrollarse en paralelo, dificultaban la organización del tiempo. Una práctica común era priorizar aquella asignatura que tuviese la fecha de entrega más temprana, lo cual perjudicaba el trabajo metodológico en una o en ambas asignaturas. En general, la coincidencia

de dos proyectos en un mismo cuatrimestre generaba en los alumnos sensaciones de malestar, sobrecarga de trabajo y de escasa coordinación docente. La experiencia en su conjunto terminaba siendo más estresante que enriquecedora.

La unificación de ambos proyectos pretende eliminar redundancias y corregir deficiencias.

3. Proyecto docente

3.1. Fundamentos

ATDD es una metodología de desarrollo ágil que promueve la especificación de las necesidades de clientes y usuarios a través de pruebas de aceptación ejecutables, antes del diseño e implementación del código destinado a cubrir tales necesidades. ATDD comparte filosofía con TDD, con la diferencia de que la primera se centra en pruebas de aceptación y el desarrollo del sistema, mientras que la segunda suele limitarse a pruebas de unidad y a métodos de clase.

Ambas estrategias son procesos iterativos que comienzan escribiendo una nueva prueba que falla, pues evalúa una funcionalidad aún no implementada, seguida de una etapa de desarrollo mínimo cuyo objetivo es pasar la nueva prueba (junto con aquellas definidas previamente). La Figura 1 muestra un esquema resumido que integra ambos procesos.

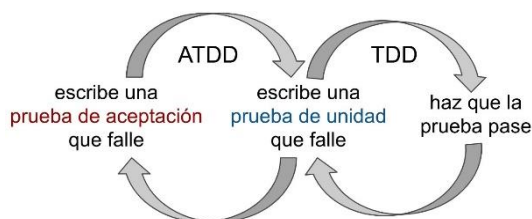


Figura 1: Integración entre ATDD y TDD.

En el ámbito del proyecto, las pruebas de aceptación son el resultado de un proceso de refinamiento de las necesidades de los usuarios, que descompone requisitos (abstractos) en historias de usuario, estas en escenarios de prueba, y estos en pruebas de aceptación (ejemplos concretos). Los escenarios recorren patrones válidos e inválidos de uso, hasta cubrir el alcance funcional de cada historia. Finalmente, cada escenario es representado mediante ejemplos concretos (pruebas de aceptación), los cuales son formalmente especificados a través del modelo *Given-When-Then* e implementados a través de entornos de la familia *xUnit* [2].

Las historias, escenarios y pruebas derivadas se usan como base para planificar un desarrollo en ciclos o iteraciones. Las historias de usuario son distribuidas en iteraciones de duración similar, siguiendo criterios de prioridad para el propietario del producto (el profesor) y de complejidad de las historias. Se propone usar la

descomposición de una historia en escenarios y pruebas como medida de complejidad: el número de escenarios y de casos extremos se entenderá como un indicador objetivo de la complejidad de la historia y se usará para estimar su coste de desarrollo.

Cada ciclo comienza con la implementación (automatización) de las pruebas de aceptación de las historias de usuario incluidas en dicho ciclo, lo cual obliga a imaginar o concebir las entidades de más alto nivel del futuro diseño involucradas en las pruebas. De esta forma, las pruebas promueven directamente el diseño. Tras una implementación sin lógica de estas entidades, que admita una ejecución fallida de las pruebas, comienza otro proceso iterativo interior (al ciclo actual) dirigido por las pruebas, que realiza en cada paso un desarrollo mínimo que permita pasar una nueva prueba. Nótese que se trata de una implementación de extremo a extremo, que fuerza un diseño en cascada de entidades de mayor a menor nivel que intervienen en la prestación del servicio que se valida en la prueba, proceso en el que se fomenta la modelación de dependencias a través de abstracciones (interfaces).

Además, se decidió proponer una iteración 0 con una única historia de desarrollador (*spike*), consistente en implementar un prototipo con funcionalidad mínima que demostrase la integración de las tecnologías involucradas. El objetivo era aprender a usar tales tecnologías, mitigar riesgos críticos, y aprovechar este conocimiento para realizar estimaciones más fiables.

Como parte de la asignatura Diseño de software, se exige al estudiante el diseño de los protocolos de comunicación y de las interfaces de usuario según patrones MVC, MVP o MVVM, el uso de al menos otros dos patrones de diseño (además de los usados en la interfaz de usuario), y la escritura de un manual de usuario y de un manual de procedimiento.

Se propone una dinámica de trabajo en equipos para analizar historias de usuarios, definir escenarios y pruebas, y alternar programación en pareja (*pair programming*) e individual de código con integración y control de cambios centralizado en repositorio en la nube. La programación por pareja se recomienda en la automatización de pruebas de aceptación, cuando se toman las primeras decisiones de diseño, y la individual en el desarrollo de entidades concretas. No se promueve la división en roles, pues se trata de incentivar que los alumnos colaboren en todas las tareas.

En Paradigmas de software también se exige la implementación de pruebas de integración en las que dependencias externas, por ejemplo, un servicio en la nube, sean sustituidas por dobles de prueba (*mocks*).

3.2. Enunciado del proyecto

El proyecto proponía la implementación de un sistema domótico de alarmas que constaba de un servidor y dos clientes. El servidor no tenía interfaz de usuario,

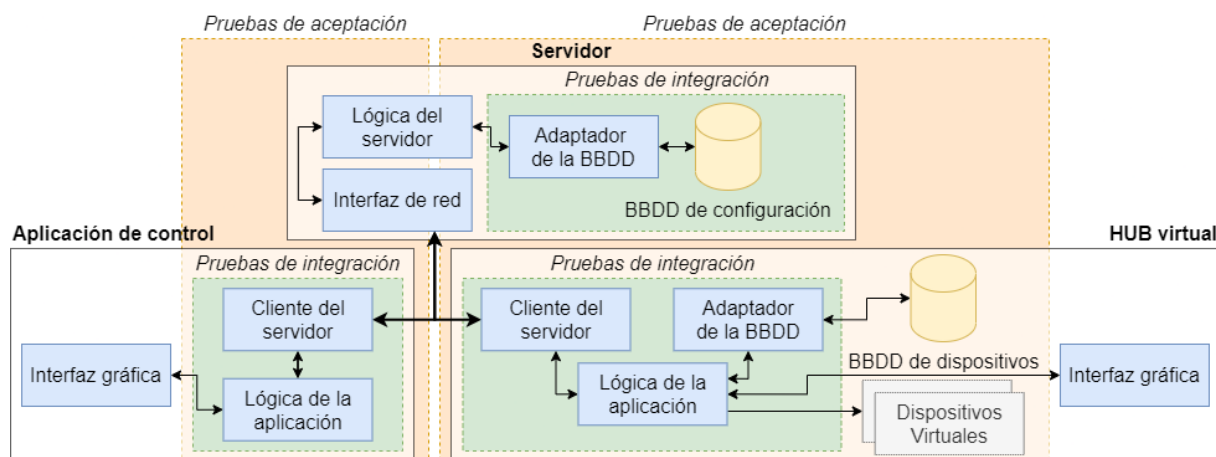


Figura 2: Alcance del proyecto.

incluía la lógica de la aplicación, y se promovía su despliegue como servicio en la nube. Los clientes consistían en un *hub* virtual, para simular a través de una interfaz de usuario un *hub* real conectado a dispositivos sensores o actuadores, y una aplicación de control para monitorizar los dispositivos conectados al *hub* y definir políticas de actuación. La Figura 2 ilustra la relación entre los tres sistemas y el alcance de las pruebas.

El servidor recibía, por un lado, el estado de los sensores desde el *hub* y, por otro, las políticas de actuación definidas por los usuarios a través de la aplicación de control. Estas políticas permitían al servidor, ante un cambio de estado en los sensores, activar los actuadores correspondientes. El *hub* virtual era un simulador (componente de prueba) que permitía añadir, eliminar y modificar dispositivos conectados a través de una interfaz de usuario, así como modificar el estado de los sensores virtuales para provocar escenarios de uso deseados. La aplicación de control debía ofrecer funciones para agrupar dispositivos en habitaciones, activar y desactivar dispositivos, monitorizar estado actual del sistema y garantizar su persistencia.

La lógica del sistema era intencionalmente simple para facilitar que los estudiantes pudieran centrarse en la metodología de desarrollo y en la integración de tecnologías potencialmente novedosas. Al tratarse de dos clientes asíncronos con ámbitos funcionales independientes, estos podían desarrollarse por separado siguiendo el paradigma ATDD.

Se propusieron dos modalidades con diferentes alcances funcionales, requisitos técnicos y nota máxima. La primera, denominada básica, debía constar de dos clientes Java de escritorio y una aplicación servidor también en Java que se ejecutaría en el mismo equipo que los clientes (dirección *localhost*). La segunda modalidad, denominada avanzada, debía implementar la aplicación de control para su uso en una plataforma móvil o web, el *hub* virtual con una interfaz gráfica más semántica y evolucionada, y la aplicación servidor en una plataforma accesible desde Internet.

La mayor parte de las tecnologías requeridas en la

modalidad avanzada eran desconocidas para la mayoría de los alumnos. El objetivo principal fue recrear un escenario exigente y realista, en el que los alumnos tuviesen que aprender, de forma autodidacta, a integrar herramientas nuevas para la solución de una tarea. Para facilitar su aprendizaje, el profesor de prácticas desarrolló ejemplos y demostraciones en el aula.

3.3. Sistemas de evaluación

En esta primera experiencia se han mantenido separados los sistemas de evaluación, aunque la defensa final se realizó frente a un tribunal formado por los profesores de ambas asignaturas, lo cual permitió una valoración más integral del proyecto.

En la asignatura Diseño de Software, la evaluación del proyecto se realiza de manera global mediante una única entrega final. Dicha entrega representa el 40% de la nota final de la asignatura. Se establecen dos requisitos mínimos para poder obtener una calificación de aprobado: se debe aplicar más de un patrón de diseño al proyecto, uno de ellos debe ser un patrón de interfaz (MVC, MVP o MVVM), y la interfaz de usuario debe estar razonada y correctamente elaborada.

En la asignatura Paradigmas de Software, el proyecto se evalúa de forma continua a través de tres entregas parciales. La primera (30% de la nota) tiene como objetivo describir jerárquicamente el espacio funcional a través de la identificación de requisitos, historias de usuario por requisito, análisis de escenarios por historia, y pruebas de aceptación por escenario. La segunda entrega (30%) incluye la planificación de iteraciones a partir de estimaciones basadas en la cantidad y complejidad de los escenarios, y del conocimiento adquirido en la iteración 0 (prototipo exploratorio *spike*). Esta segunda entrega también incluye el código que automatiza las pruebas de aceptación de las historias asignadas a la iteración 1. La tercera entrega (40%) incluye el software desarrollado y un informe que describe el proceso de desarrollo, con especial énfasis en la documentación de las pruebas implementadas, su tipología (unitarias, integración, aceptación) y

un breve razonamiento sobre su necesidad. La nota del proyecto representa un 70% de la nota final.

Aunque ambas asignaturas pertenecen a una misma intensificación y se imparten en el mismo semestre del grado, hubo un estudiante en cada asignatura que no se matriculó en la otra. En principio, se logró que ambos formasen un mismo equipo con un tercer alumno que sí estaba matriculado en las dos asignaturas, con la expectativa de que los dos primeros se complementasen. Este equipo fue considerado de dos miembros con relación al alcance del proyecto, mientras que cada alumno sería evaluado en función de los criterios de la asignatura matriculada. No obstante, la experiencia no resultó satisfactoria, pues su éxito descansaba en gran medida en un buen funcionamiento del equipo. En futuras realizaciones de esta propuesta deberán definirse itinerarios alternativos precisos que garanticen una gestión eficaz de estas singularidades.

3.4. Oportunidades

Desde una perspectiva formativa, el proyecto propuesto ofrece amplias oportunidades a los alumnos:

- Experiencia en valores ágiles, por ejemplo, versiones frecuentes de software funcional, colaboración con el cliente (profesor), diseño determinado por necesidades de usuario, etc.
- Valor de las pruebas de aceptación como criterio objetivo de progreso, especificación de las necesidades del usuario, guía para el diseño y el desarrollo, base para estimar esfuerzo, etc.
- Valor de diseños simples y flexibles, con gestión eficiente de dependencias.
- Experiencia con un proyecto realista, pues se propone un dominio real de aplicación.
- Aprendizaje autónomo e integración de diversas tecnologías avanzadas: web, móvil, nube, etc.
- Prototipos exploratorios tempranos para entender nuevas tecnologías, mitigar riesgos estructurales, ganar conocimiento que permita estimaciones más fiables de complejidad, esfuerzo, etc.
- Gestión de la incertidumbre en el diseño y desarrollo de un proyecto de software.

4. Resultados

La evaluación de la experiencia involucró dos perspectivas de análisis. La primera se basó en una encuesta realizada a los alumnos que presentaron el proyecto en primera convocatoria. La segunda consistió en la comparación de las distribuciones de las notas de proyectos de los cursos 2019-2020 y 2020-2021.

4.1. Análisis de las encuestas

La encuesta, mostrada en el Cuadro 1, consta de tres secciones: *Perfil del proyecto*, *Experiencias en tareas de diseño* y *Experiencias en ATDD*. La primera sección está inspirada en el cuestionario propuesto en

[10]. Las otras dos son específicas del proyecto propuesto, y su objetivo ha sido doble: promover la autorreflexión en los alumnos y detectar áreas de mejora.

La Figura 3 resume los resultados de la encuesta. Las barras muestran las puntuaciones medias de las cuestiones individuales de las secciones Perfil del proyecto (verde), Experiencias en tareas de diseño (azul) y Experiencias en ATDD (naranja). Medias por sección de 4,70, 4,18 y 4,20 respectivamente, sugieren una muy buena aceptación del proyecto en su conjunto (4,70), aunque también algunas dudas en el logro de algunos objetivos de diseño y desarrollo.

Perfil del proyecto (1-Poco de acuerdo; 5-Muy de acuerdo)
1. Un único proyecto me ha permitido ser más productivo.
2. El proyecto desarrollado me ha parecido realista.
3. Mi motivación ha sido mayor que en proyectos anteriores.
4. He integrado contenidos de ambas asignaturas.
5. Me ha resultado estimulante que el proyecto incluyese tecnologías web, de aplicaciones móviles y servicios en la nube.
Experiencias en tareas de diseño (1-5)
6. He experimentado el valor de diseñar a partir de ejemplos concretos de escenarios de uso/prueba.
7. He experimentado el valor de diseñar un sistema progresivamente de arriba a abajo.
8. He tenido la oportunidad de usar patrones de diseño.
9. He rediseñado menos de lo habitual.
10. He experimentado el valor de implementar un prototipo exploratorio para aprender a integrar las tecnologías del proyecto.
11. He experimentado el valor de diseñar clases con responsabilidad única.
12. He experimentado el valor de extender clases en lugar de modificarlas.
13. He experimentado el valor de diseñar clases dependientes de abstracciones.
14. He experimentado el valor de la inyección de dependencias.
15. Considero que el diseño final es relativamente simple.
Experiencias en ATDD (1-5)
16. Comenzar escribiendo pruebas me ha facilitado la comprensión de los requisitos funcionales.
17. Comenzar escribiendo pruebas me ha facilitado la planificación de cada iteración.
18. Comenzar escribiendo pruebas me ha facilitado la adopción de decisiones de diseño.
19. Comenzar escribiendo pruebas me ha facilitado la colaboración dentro del equipo.
20. Las pruebas de aceptación automatizadas me han permitido una evaluación objetiva del final de la iteración.
21. He experimentado el valor de recibir realimentación frecuente del propietario del sistema (el profesor).
22. He experimentado el valor de entregas periódicas del software con funcionalidad creciente.
23. He visto a mi equipo motivado.

Cuadro 1: Encuesta realizada a los estudiantes.

Un análisis de las cuestiones menos valoradas nos ha permitido detectar limitaciones de la propuesta. En la primera sección, la cuestión menos valorada (4,17) resultó ser “El proyecto desarrollado me ha parecido realista”. Aunque es una media favorable, el hecho de desarrollar un *hub* virtual, por la imposibilidad de disponer de dispositivos reales, podría haber influido negativamente. Quizás debió haberse hecho mayor énfasis en el valor del *hub* virtual como herramienta útil para generar escenarios de prueba, incluso en un contexto real que contase con todos los recursos. En la segunda sección, como cuestión menos valorada aparece “He rediseñado menos de lo habitual”, con una puntuación media de 3,67. Este resultado es consistente con la complejidad y singularidad de la estrategia de diseño de arriba a abajo, guiada por pruebas de aceptación, y realizando una gestión disciplinada de dependencias e incertidumbres. En realizaciones futuras del proyecto, deberá hacerse un seguimiento exhaustivo de las decisiones de diseño que adopten los equipos. Finalmente, el aspecto peor puntuado de la tercera sección fue “He experimentado el valor de entregas periódicas del software...”, con un valor medio de 3,83. Este resultado es coherente con comentarios de texto libre recogidos en la encuesta, que valoran como escaso el tiempo planificado para el desarrollo del proyecto, situación que limitó objetivamente el número de ciclos de desarrollo y las opciones de experimentar mayor progresividad en los incrementos funcionales.

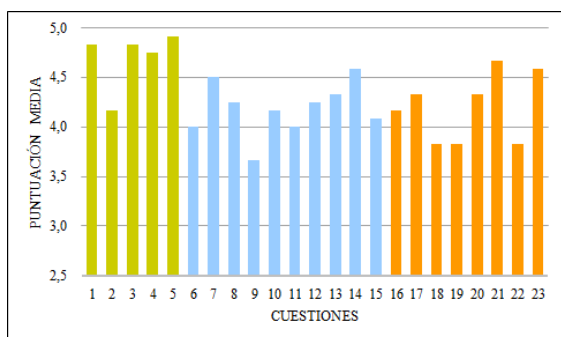


Figura 3: Puntuación media por cuestiones. Los colores ilustran la separación entre secciones.

Merece la pena destacar las altas valoraciones de las cuestiones 7, 13 y 14, relacionadas con las experiencias en diseño de arriba abajo y en gestión de dependencias, y de las cuestiones 16, 17 y 20, relacionadas con el valor de las pruebas de aceptación tempranas para mejorar la comprensión de la tarea, y como criterio objetivo de planificación y validación de ciclos. Además, se incluyeron dos apartados para texto libre para que los alumnos expresaran opiniones positivas y negativas acerca de la experiencia.

Entre las opiniones positivas hubo referencias a la libertad de elección y a la oportunidad para aprender

nuevas tecnologías, a poder desarrollar un único proyecto para dos asignaturas, al nivel de motivación colectivo alcanzado, y a la integración de conocimientos de muchas áreas de la carrera. Como aspectos negativos se destacó el poco tiempo previsto para el desarrollo de un proyecto bajo una nueva metodología.

4.2. Análisis de distribuciones de notas

La experiencia también fue evaluada comparando las distribuciones de las notas de proyectos en los cursos 2019-2020 y 2020-2021 (ver Figura 4). El gráfico superior corresponde a la asignatura Diseño de software, y el inferior, a Paradigmas de software. Ambas asignaturas tuvieron proyectos distintos durante 2019-2020, mientras compartieron proyecto en 2020-2021. Los criterios de evaluación se mantuvieron invariables durante los dos cursos y fueron aplicados por el mismo profesorado (diferente entre asignaturas).

En ambas asignaturas se observa un patrón común: la distribución de notas de 2020-2021 (columnas verdes) tiende a situarse a la derecha de la distribución de 2019-2020 (columnas azules). Es decir, las notas del curso 2020-2021 tienden a ser más altas que las de 2019-2020. Este resultado puede contrastarse en el Cuadro 2, donde se muestran las medias y medianas de estas distribuciones de notas. Ambos indicadores crecen en las dos asignaturas con el cambio de curso.

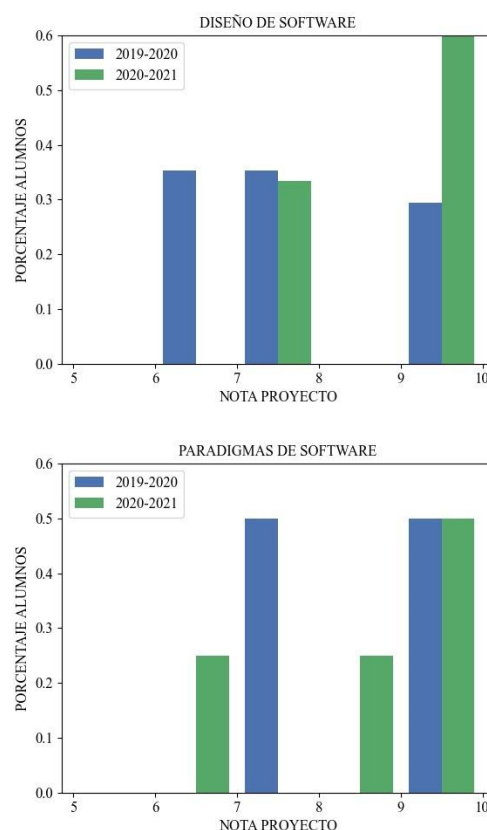


Figura 4: Distribución de notas de proyecto en ambas asignaturas en los cursos 2019-2020 y 2020-2021.

Al margen del patrón descrito, las distribuciones muestran una tendencia bimodal, es decir, dos grupos claramente definidos por notas altas y medias, respectivamente. En la práctica, esto se traduce en equipos que logran una motivación colectiva muy alta, y cuyos proyectos finales son comparables a soluciones profesionales considerando criterios de funcionalidad, usabilidad (según valoración de profesores), uso e integración de tecnologías avanzadas, rendimiento, robustez del diseño y calidad del código. Por el contrario, hay equipos que optan por tecnologías conocidas, desarrollos de bajo riesgo y alcance funcional reducido, dando lugar a soluciones claramente mejorables.

	Diseño de Soft.		Paradigmas de Soft.	
	19-20	20-21	19-20	20-21
media	7,5	9,3	8,2	8,8
mediana	7,0	10,0	7,7	9,4

Cuadro 2: Media y mediana de las distribuciones de notas de proyecto por asignatura y curso.

5. Conclusiones

Este artículo describe una experiencia de coordinación docente que promueve el aprendizaje basado en estrategias de diseño evolutivo de software, dirigidas por necesidades de usuarios formuladas en términos de pruebas de aceptación ejecutables. Los resultados de una encuesta realizada entre los estudiantes reflejaron una buena aceptación de la propuesta, aunque también revelaron la necesidad de mejorar la planificación docente para permitir un desarrollo más sostenible.

La comparación entre distribuciones de notas de los proyectos en los cursos 2019-2020 y 2020-2021 mostró que el proyecto compartido promueve altos grados de motivación en un mayor número de alumnos, los cuales fueron capaces de desarrollar software con un nivel comparable al profesional, mientras se observa un segundo grupo, menor que el primero, que opta por tecnologías conocidas y soluciones más simples.

Futuras implementaciones de este proyecto deberán avanzar hacia una mayor integración de los sistemas de seguimiento y evaluación de ambas asignaturas, desde la sincronización de las entregas, hasta una nota única del proyecto que contemple una visión más integradora del resultado del ejercicio.

Referencias

- [1] Kent Beck y otros 16 coautores. Manifesto for agile software development, 2001.
- [2] David Chelimsy, Dave Astels, Bryan Helmkamp, Dan North, Zach Dennis, y Aslak Hellesoy, The RSpec book: Behaviour driven

development with Rspec. Cucumber, and Friends, Pragmatic Bookshelf, 2010.

- [3] Shannon Duvall, Duke Hutchings y Michele Kleckner. Changing perceptions of discrete mathematics through scrum-based course management practices. *Journal of Computing Sciences in Colleges*, vol. 33, pp. 182-189, 2017.
- [4] Maria Ferré, Carlos García-Barroso, Montse García-Famoso, David Sánchez y Aida Valls. Mejora de la formación en el diseño y desarrollo de software a partir de la coordinación de distintas asignaturas. En *Actas de las XXV Jornadas de Enseñanza Universitaria de Informática, Jenui 2019*, pp. 23-30, Murcia, julio 2019.
- [5] Nicolás Martín Paez. A flipped classroom experience teaching software engineering. En *IEEE/ACM 1st Int. Workshop on Software Engineering Curricula for Millennials (SECM)*. IEEE, pp. 16-20, Argentina, mayo 2017.
- [6] Peter Merholz y Kristin Skinner. Org design for design orgs: Building and managing in-house design teams. O'Reilly Media, Inc., 2016.
- [7] Ken Pugh. Lean-Agile Acceptance Test-Driven Development: Better Software Through Collaboration. Addison-Wesley, 2011.
- [8] David F. Rico y Hasan H. Sayani. Use of Agile Methods in Software Engineering Education. En *Proceedings of the Agile Conference*, pp. 174-179, Chicago, IL, 2009.
- [9] Pasquale Salza, Paolo Musmarra y Filomena Ferrucci. Agile Methodologies in Education: A Review. En *Agile and Lean Concepts for Teaching and Learning*, pp. 25-45, Springer, 2019.
- [10] Pablo Sánchez, Carlos Blanco, Alejandro Pérez, Julio Medina, Patricia López, Alfonso de la Vega, Diego García y Miguel Sierra. Experiencia y Lecciones Aprendidas durante el Desarrollo de un Proyecto Software Común a Diversas Asignaturas. En *Actas de las XXIII Jornadas de Enseñanza Universitaria de Informática, Jenui 2017*, pp. 127-134, Cáceres, julio 2017.
- [11] Laio O. Seman, Romeu Hausmann y Eduardo A. Bezerra. On the students' perceptions of the knowledge formation when submitted to a project-based learning environment using web applications. *Computers & Education*, vol. 117, pp. 16-30, 2018.
- [12] John C. Stewart, Carolyn S. DeCusatis, Kevin Kidder, Joseph R. Massi y Kirk M. Anne. Evaluating agile principles in active and cooperative learning. En *Proc. of Student-Faculty Research Day, CSIS*, Pace University, B3, pp. 1-8, 2009.
- [13] Jeff Sutherland. Scrum: The Art of Doing Twice the Work in Half the Time. Random House Business, 2015.

Autoevaluación, co-evaluación y estudio empírico frente a técnicas de evaluación tradicionales en la asignatura de Sistemas Operativos

Patricia Martín-Rodilla, Javier Parapar
Information Retrieval Lab
University of A Coruña
A Coruña, Spain

patricia.martin.rodilla@udc.es, javierparapar@udc.es

Resumen

En este artículo, presentamos un protocolo desarrollado como parte de las actividades de innovación docente en Sistemas Operativos. El objetivo del protocolo es incorporar la autoevaluación como mecanismo de evaluación formativa en competencias en Sistemas Operativos, que fomente en el estudiante capacidades transversales de especial relevancia en diseño e implementación de funcionalidades de Sistemas Operativos: análisis crítico, detección de posibles mejoras y conciencia del propio proceso de aprendizaje. Para ello, se diseñó un protocolo de autoevaluación y co-evaluación de las pruebas prácticas de la asignatura correspondientes a tres áreas temáticas en Sistemas Operativos: sistemas de ficheros, manejo de memoria y gestión y planificación de procesos. Buscando posibilitar la convivencia de ambos sistemas de evaluación, el protocolo se aplicó a parte del alumnado, manteniendo la evaluación por parte del profesorado paralelamente. Posteriormente, se realizó un estudio empírico de precisión en autoevaluación y co-evaluación de los estudiantes, comparando el protocolo con la evaluación tradicional, y evaluando inicialmente sus implicaciones en las calificaciones finales obtenidas. Los resultados permiten no sólo evaluar inicialmente el protocolo diseñado, sino también conocer el punto de partida en capacidades de autoevaluación y co-evaluación de los estudiantes en las áreas temáticas particulares de Sistemas Operativos.

Abstract

In this paper, we present a protocol developed as part of some educational innovation initiatives in Operating Systems subject. The protocol's goal is incorporating self-evaluation as a formative evaluation mechanism in Operating Systems that fosters in university students of computer science the skills of

critical analysis, detection of possible improvements and awareness of their own learning process, all of them of special relevance in the design and implementation of functionalities of Operating Systems. For that purpose, a self-evaluation and co-evaluation protocol was designed for the practical exercises corresponding to three thematic areas in Operating Systems: file systems, memory management, and process management and scheduling. In order to facilitate the coexistence of both evaluation systems, the protocol was applied to part of the students, maintaining the traditional evaluation by the teaching staff in parallel with all the students. Subsequently, an empirical study of precision in self-evaluation and co-evaluation of the students was carried out, analyzing the correlation with the traditional evaluation, and the implications of the designed protocol in the final grades obtained. The results allow not only to initially evaluate the designed protocol, but also to know the starting point in self-evaluation and co-evaluation capacities of the students in Operating Systems areas.

Palabras clave

Autoevaluación, co-evaluación, Sistemas Operativos, estudio empírico.

1. Contexto docente e investigador

La docencia en Ingeniería Informática ha sufrido una profunda transformación en los últimos años, impulsada por los cambios de paradigma como la adaptación al EEES [3, 8, 18] o, recientemente, a los espacios virtuales de enseñanza y aprendizaje. En este contexto, las asignaturas de los primeros cursos en Ingeniería Informática tienen por objetivos principales la adquisición y desarrollo de las capacidades y competencias transversales en el alumno. Estudios previos han demostrado mejora en este tipo de capacidades a través de la aplicación de innovaciones en

los mecanismos de evaluación [6]. Tras la detección de problemas asociados a capacidades transversales en nuestra asignatura de Sistemas Operativos, este artículo presenta un protocolo de autoevaluación y co-evaluación (evaluación por pares) de las pruebas prácticas de Sistemas Operativos, así como un estudio empírico de resultados del primer año de implantación de dicho protocolo. El objetivo es emplear la autoevaluación y la co-evaluación como mecanismo que fomente en los alumnos las capacidades de análisis crítico, detección de posibles mejoras y conciencia de su propio proceso de aprendizaje, capacidades todas ellas de especial relevancia en diseño e implementación de funcionalidades y estructuras de Sistemas Operativos.

A continuación, se realiza un recorrido por las iniciativas de innovación docente en Sistemas Operativos puestas en marcha en los últimos años en el contexto educativo español, así como otros trabajos y experiencias de autoevaluación y co-evaluación en contextos de Ingeniería Informática que ayudan a situar el protocolo desarrollado.

1.1. Innovación docente en Sistemas Operativos

La adaptación de la asignatura de Sistemas Operativos al EEES [4, 8, 11] supuso el inicio de actividades de innovación docente en Sistemas Operativos, comenzando con iniciativas centradas en la propia ejecución de la docencia, y encaminadas a dar más peso al aspecto práctico, asegurar el no abandono de los estudiantes y aumentar su motivación, con resultados satisfactorios [4, 10]. En Canaleta [4] se analizan este tipo de innovaciones docentes en Sistemas Operativos a lo largo de 13 años, siendo el mayor estudio histórico hasta el momento. Posteriormente, las iniciativas de innovación docente en Sistemas Operativos han puesto el foco en otros aspectos, destacando: 1) internacionalización con docencia en otros idiomas (principalmente inglés) o iniciativas bilingües [1], 2) fomento del trabajo continuo y en equipo del alumnado, con técnicas como el *flipped classroom* [24] o la evaluación por pares [9], 3) estudio de ciertas competencias, como la capacidad de organización y planificación del estudio por parte del estudiantado [17] o de análisis visual y seguimiento de la clase [7, 21], con técnicas de mapas conceptuales [20] y 4) adaptación a la práctica de la docencia de Sistemas Operativos en entornos virtuales de aprendizaje o automatización de ciertas tareas de evaluación [13], con técnicas muy diversas [16].

Como puede verse en estos trabajos, las innovaciones docentes en Sistemas Operativos se dan en su mayoría en su adecuación al EEES, y luego a entornos virtuales de aprendizaje. El rango de técnicas aplicadas y experiencias docentes publicadas con este

fin es alto, aunque probadas en plazos cortos de tiempo [4].

Sin embargo, el número de trabajos centrados en fomentar competencias genéricas y capacidades transversales dentro de la asignatura es significativamente menor. En este sentido, la mayor parte de técnicas e iniciativas docentes son aplicadas a aspectos de la propia práctica docente, como *flipped classroom* [24] o mapas conceptuales [20], o estudios estadísticos de resultados a posteriori [13], no encontrando trabajos en Sistemas Operativos que ahonden en la influencia de los cambios en los mecanismos de evaluación como una forma de fomentar dichas capacidades genéricas y/o transversales.

1.2. Autoevaluación y co-evaluación en Ingeniería Informática

Autoevaluarse supone implicarse de forma explícita en la revisión del trabajo o actividad propio [6]. Se ha usado el término co-evaluación en estudios en docencia para referirse a varios aspectos que conviene distinguir: 1) a la evaluación por parte de pares, 2) a la evaluación conjunta del trabajo en equipo o 3) a la coexistencia de varios sistemas de evaluación. En este artículo, el protocolo propuesto se refiere a coevaluación como a la evaluación que realizan los alumnos de sus prácticas realizadas en equipo junto con otros compañeros (acepción 2).

Pese a la ausencia de trabajos en docencia en Sistemas Operativos (como se detalla en la sección anterior) que específicamente traten aspectos de evaluación como mecanismo de mejora de capacidades genéricas y/o transversales, son numerosos los estudios y aplicaciones en otras áreas y campos, incluso dentro de los estudios de Ingeniería Informática.

En primer lugar, podemos encontrar trabajos centrados en la aplicación de la autoevaluación y/o la co-evaluación en asignaturas concretas de la Ingeniería Informática, como en asignaturas de programación [14], redes [5] o bases de datos [15], mostrando buenos resultados tanto en la valoración de la experiencia por parte del alumnado, como en resultados académicos [14, 15].

En segundo lugar, estudios de revisión en mayor profundidad sobre estos mecanismos en la docencia de la Ingeniería Informática, señalan también la utilidad de estos mecanismos de autoevaluación para alcanzar una evaluación continua real [14], mejorar la efectividad de los mecanismos de retroalimentación [25], como medio formativo en sí mismo [14, 18] e incluso en términos de mejora del sentido crítico y capacidad de reflexión sobre el propio proceso de aprendizaje [25].

2. Objetivos

En este contexto, el objetivo es diseñar un protocolo de autoevaluación y co-evaluación por pares de las prácticas de la asignatura de Sistemas Operativos, como mecanismo de evaluación formativa sobre las competencias específicas de la asignatura que, además, fomentan en el estudiantado capacidades transversales (análisis crítico, detección de posibles mejoras y conciencia del propio proceso de aprendizaje). El protocolo diseñado se implanta en la asignatura de Sistemas Operativos 2020-2021, y se compara con el método tradicional de evaluación por parte del profesor aplicado hasta ahora, permitiendo conocer de forma inicial e intuitiva el punto de partida del alumnado, los efectos del protocolo en la mejora de las capacidades transversales abordadas y su impacto en las calificaciones obtenidas por los estudiantes. No buscamos, por tanto, realizar un estudio empírico estadístico exhaustivo que pruebe la adecuación del protocolo a todos los objetivos (en un solo curso académico no tendríamos muestra suficiente), sino más bien plantear las preguntas de investigación que nos surgen y analizar de manera inicial e intuitiva los datos de este primer curso académico de implantación, para así extraer lecciones aprendidas de la experiencia de innovación docente y desarrollar en el futuro un plan de actuación a largo plazo para la mejora del protocolo propuesto.

3. Protocolo y estudio empírico

3.1. Hipótesis iniciales y protocolo

De forma similar a otras planificaciones de la asignatura de Sistemas Operativos [4], la evaluación tradicional que se realiza en nuestra asignatura se basa en la revisión y calificación del profesor de dos grandes ítems: examen semestral y tres prácticas de programación en lenguaje C [22]. Las prácticas constituyen un 25% del peso en la calificación final. Cada una de las prácticas se centra en un gran bloque temático de la asignatura: P1 (sistemas de ficheros), P2 (manejo de memoria) y P3 (gestión y planificación de procesos). El objetivo es que los alumnos, organizados en parejas, afiancen los conocimientos teóricos y adquieran las competencias y habilidades prácticas en estas tres áreas. En la Figura 1 puede verse el diagrama del proceso de *evaluación tradicional*: los estudiantes, en grupos reducidos (15-20 personas), se distribuyen en parejas y realizan sesiones de *Focus Group* donde diseñan, codifican, realizan pruebas y resuelven dudas con un profesor en sesiones prácticas. Cada práctica (P1, P2 y P3) tiene una fecha máxima de entrega. Posteriormente, cada pareja de estudiantes acude a una sesión de defensa por práctica donde explican su diseño y programa final.

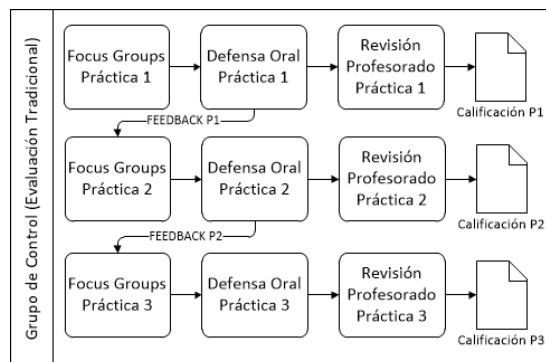


Figura 1: Diagrama de proceso de evaluación tradicional (Grupo de control).

Los estudiantes deben responder a cualquier aspecto sobre la misma para demostrar el conocimiento profundo del trabajo realizado. El profesor califica a la pareja atendiendo a unos criterios estructurales, funcionales y de exposición oral, dando una retroalimentación sobre las funcionalidades incompletas, no entregadas o las dificultades que los estudiantes puedan manifestarle. Aunque los criterios establecidos para cada práctica garantizan la evaluación de forma sistemática, el número de alumnos a evaluar y la escasez de tiempo para cada sesión de defensa no permite dedicar demasiado tiempo a la retroalimentación que el profesor puede ofrecer a cada pareja, siendo esta fundamental para asentar y mejorar nuestras capacidades objetivo (análisis crítico, detección de posibles mejoras y conciencia del propio proceso de aprendizaje). En la Figura 2 puede verse el diagrama del protocolo diseñado.

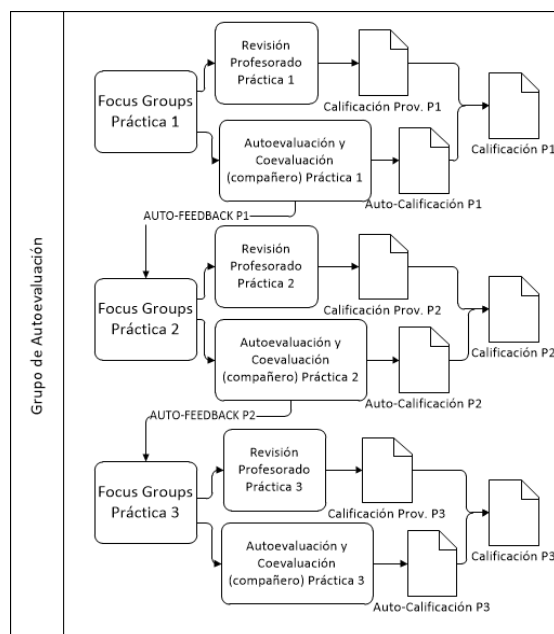


Figura 2: Protocolo de autoevaluación y coevaluación propuesto.

Los estudiantes, al igual que en la evaluación tradicional, atienden a su *Focus Group* con un profesor para el diseño, codificación, prueba y resolución de dudas de cada práctica. Una vez llegada la fecha de entrega de cada práctica, el protocolo establece dos actividades paralelas.

Por un lado, los estudiantes acuden a una sesión de autoevaluación y co-evaluación, donde se les facilita un formulario con una lista de criterios que la práctica a autoevaluar debe cumplir (a modo de *check box*, con posibilidad de anotaciones cortas). Cada pareja debe indicar por cada criterio, si la práctica entregada lo cumple o no y si la funcionalidad relacionada se ejecuta correctamente. Se permiten las anotaciones breves al pie para explicitar, por ejemplo, si detectaron algún error concreto pese a cumplir el criterio completo.

Es importante indicar que los alumnos pueden revisar su código y probar en ejecución su práctica durante la sesión de autoevaluación, así como debatir y revisar el trabajo de su compañero de pareja. Esta parte de coevaluación, pretende que el estudiante encuentre en el compañero/a “un punto de referencia para apreciar si su propia valoración está bien conmensurada” [14], mejorando los resultados de autoevaluación si ésta incluye coevaluación [19].

En paralelo, el profesor realiza una evaluación personal de cada práctica y asigna una calificación provisional “a ciegas”, es decir, sin consultar el formulario de autoevaluación entregado por cada pareja.

Una vez obtenidas sendas calificaciones (Autoevaluación y Provisional del profesor), el docente realiza una ponderación final con ambas como referencia, adecuando la calificación a los criterios reales de cada práctica. Nótese que, siguiendo este protocolo y en comparación con la evaluación tradicional, la retroalimentación obtenida por los estudiantes es diferente, ya que procede: 1) de su propia experiencia mirando el código, ejecutando la práctica y evaluando según los criterios definidos su propio trabajo y 2) de un debate con su compañero acerca de la adecuación de su calificación y trabajo a los criterios definidos, siendo una retroalimentación mucho más detallada que la obtenida con la evaluación tradicional mediante defensa (en tiempo y forma).

3.2. Diseño de experimento

Una vez diseñado el protocolo, se decidió realizar un cuasi-experimento, i.e un experimento causa-efecto controlado para entornos donde no es posible seleccionar aleatoriamente los sujetos de la muestra, como el entorno educativo [2].

La elección de los grupos de autoevaluación se realizó de forma aleatoria, si bien su formación viene determinada por las listas de matrícula. La muestra se organiza en grupo de control (evaluación tradicional,

N=75 estudiantes) y grupo de autoevaluación (N=38 estudiantes). La presencia de mayor número de estudiantes en el grupo de control se debe a la ausencia de tiempo por parte del profesorado para aplicar el protocolo de autoevaluación en más grupos, debiéndose balancear en los siguientes cursos académicos.

El objetivo del experimento, siguiendo la metodología de Wohlin [27], se define como: “Comparar el protocolo de autoevaluación y co-evaluación diseñado con el método de evaluación tradicional, con el objetivo de evaluar su influencia en el desarrollo de capacidades genéricas y/ transversales y medir su impacto en las calificaciones y estadísticas finales obtenidas de la asignatura”.

Las preguntas de investigación (RQ) a modo de hipótesis iniciales que definimos son:

RQ0: No existen diferencias significativas entre calificaciones del grupo de autoevaluación (las impuestas por los estudiantes, así como las finales) y las calificaciones finales del grupo de control.

Esta hipótesis nos permite analizar implicaciones de capacidad de autoevaluación de los estudiantes.

RQ1: No hay diferencia significativa en los cambios en las calificaciones desde P1 a P3 entre ambos grupos.

Esta hipótesis nos permite analizar si es mejor el *feedback* propio (asociado a la capacidad transversal de análisis crítico y detección de mejoras) que el obtenido en defensa por parte del profesor (grupo de control), materializándose en ese caso en que el grupo de autoevaluación mejoraría más en calificaciones de P1 a P3.

RQ2: La correlación entre las calificaciones establecidas por los estudiantes en el grupo de autoevaluación y las provisionales del profesor en ese mismo grupo no cambia con el tiempo.

Esta hipótesis nos permite analizar si la autoevaluación ayuda a tomar conciencia del propio proceso de aprendizaje, manifestándose si en P3 las calificaciones del grupo de autoevaluación se acercan más en correlación a las calificaciones provisionales del profesor en el grupo de autoevaluación en comparación con P1.

RQ3: No hay diferencias significativas entre las medias de calificaciones en ambos grupos otorgadas por parte del profesor (de forma provisional en el grupo de autoevaluación y definitivo en el de control).

Esta hipótesis nos permite analizar que el nivel de exigencia del profesor no se vea afectado por el experimento.

RQ4: No hay diferencia en los porcentajes de calificaciones y abandono de ambos grupos.

Esta hipótesis nos permite analizar cierto impacto de la autoevaluación en las calificaciones finales y la desmotivación (en forma de abandono) que se asocia tradicionalmente a las iniciativas de autoevaluación.

4. Resultados

Según las preguntas de investigación planteadas se realizaron los siguientes experimentos y consecuentes resultados.

RQ0: Se llevaron a cabo tests de dos muestras sobre igualdad de medias (Wilcoxon Rank Sum, H_0 : las medias son iguales, two-tail, $\alpha=0.05$) [30, 31] en calificaciones de las parejas en ambos grupos, desagregando para cada práctica, en dos partes:

A- Wilcoxon Rank Sum en calificaciones finales del grupo de autoevaluación versus calificaciones finales del grupo de control por práctica son:

- P1 p-valor=0.36 (H_0 no rechazada).
- P2 p-valor=0.01 (H_0 rechazada)
- P3 p-valor=0.284284 (H_0 no rechazada)

B- Wilcoxon Rank Sum en auto-calificaciones del grupo de autoevaluación versus calificaciones finales del grupo de control por práctica son:

- P1 p-valor=0.73 (H_0 no rechazada).
- P2 p-valor=0.65 (H_0 no rechazada).
- P3 p-valor=0.45 (H_0 no rechazada).

Cuando analizamos las calificaciones que se dan los estudiantes a sí mismos con las calificaciones del grupo de control, no existe diferencia significativa en ninguna práctica. En el caso de la comparativa entre calificaciones finales de ambos grupos, existe una diferencia significativa en el caso de la práctica P2, mientras que en P1 y P3 no hay diferencias significativas en las calificaciones finales entre ambos grupos. Creemos que esto puede deberse puntualmente a la propia práctica P2 (en contenido, nivel de exigencia o dificultad para autoevaluar, todos estos criterios habrá que probarlos en el futuro), y no al efecto general de la autoevaluación.

RQ1: Se llevaron a cabo test de dos muestras sobre igualdad de medias (Wilcoxon Rank Sum, two-tail, $\alpha=0.05$) [23, 26] de los incrementos (o decrementos) de calificaciones finales de ambos grupos de la práctica P1 a la práctica P3. Según el resultado (p-valor=0.32), no hay diferencia significativa en los incrementos entre grupos. Con los datos actuales no podemos por lo tanto concluir que el proceso de autoevaluación ayudase a mejorar más. Más experimentos deben realizarse para poder abordar los diferentes factores que estarían influyendo en este resultado.

RQ2: Se realizó un análisis de correlación Pearson entre las calificaciones establecidas por los estudiantes en el grupo de autoevaluación y las provisionales del profesor en ese mismo grupo, desagregado por práctica. La Figura 3 muestra las correlaciones obtenidas entre ambas calificaciones. Nótese que P1 (calificación final de P1), P1A (Autoevaluación de los estudiantes en P1) y P1P (Calificación profesor ciega

para P1) es la nomenclatura utilizada en todas las prácticas.

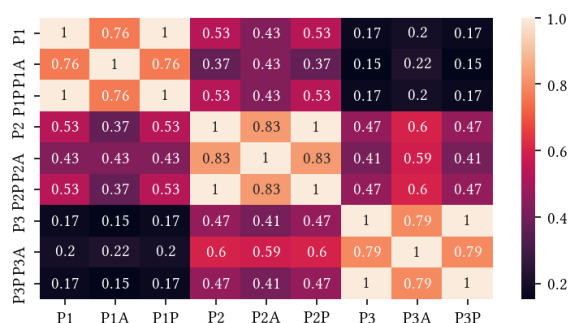


Figura 3: Correlaciones calificaciones grupo de Autoevaluación vs. calificaciones provisionales del profesor para ese mismo grupo.

Como puede observarse, la correlación entre la calificación del profesor y la autoevaluación cuando se evalúa la misma práctica (grupos centrales con resultados en naranja) va mejorando a lo largo del tiempo (hay mayor correlación en P2 (0.83) y P3(0.79) que en P1(0.76)). Esto nos da una noción preliminar acerca de que el grupo de autoevaluación muestra una mejora en cuanto a conciencia del propio proceso de aprendizaje después del primer ciclo, ya que en P2 y P3 las calificaciones del grupo de autoevaluación se acercan más en correlación a las calificaciones provisionales del profesor en ese grupo en comparación con P1. Un análisis de significancia específico con mayor muestra es necesario para confirmar este resultado positivo del protocolo diseñado.

RQ3: Se llevó a cabo un test de dos muestras sobre diferencia de medias (Wilcoxon Rank Sum, two-tail, $\alpha=0.05$) [23, 26] en calificaciones finales provisionales del profesor en el grupo de autoevaluación y calificación final del grupo de control. Es decir, las calificaciones que otorga el profesor unilateralmente en ambos grupos. Según el resultado (p-valor=0.056), no variaría significativamente el nivel de exigencia del profesor entre grupos, si bien el p-valor se encuentra al borde la significancia. Analizando en profundidad las medias obtenidas, la conclusión inicial es que, dado que la labor del profesor es ciega a la calificación de autoevaluación, probablemente la diferencia se deba más a la naturaleza de cada grupo que a diferencias en el nivel de exigencia del profesor (ver RQ0). En cualquier caso, serían necesarios experimentos más específicos sobre el nivel de exigencia del profesor, aumentando el número de estudiantes y profesores, para analizar en profundidad el resultado de esta hipótesis.

RQ4: Se extrajeron los porcentajes por nivel de calificación y los porcentajes de abandonos (estudiantes que presentaron la P1 pero no la P2 o la P3) en

ambos grupos (autoevaluación y control), mostrados en Figura 4 y Figura 5 respectivamente.

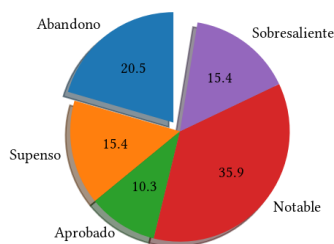


Figura 4: Porcentajes de calificaciones y abandono en el grupo de evaluación tradicional (Grupo de control).

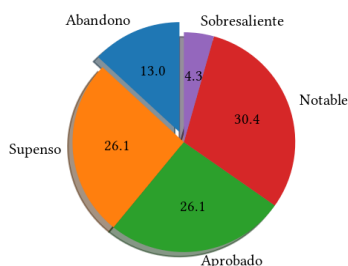


Figura 5: Porcentajes de calificaciones y abandono en el grupo de autoevaluación

Nótese en ambas imágenes dos aspectos de RQ4: 1) que el porcentaje de abandono es notablemente menor en el caso del grupo de Autoevaluación y 2) que, si bien se mantienen porcentajes similares, las calificaciones son más bajas en el grupo de Autoevaluación. En el primer caso, creemos que el protocolo de autoevaluación contribuye notablemente a reducir el porcentaje de abandono de los estudiantes, ya que ofrece retroalimentación detallada, aumenta la conciencia del propio proceso de aprendizaje (ver RQ2) y disuade del abandono a alumnos que tradicionalmente rechazan el sistema de defensa oral del método tradicional. En el segundo caso, creemos que el espectro de calificaciones en ambos grupos no difiere demasiado en representatividad. Esto, unido a la ausencia de rechazo de RQ0 nos lleva a pensar que las menores calificaciones en el grupo de autoevaluación puedan estar motivadas por otros factores no estudiados. Por ejemplo, la práctica docente nos hace pensar que el horario de las clases de los grupos de autoevaluación (principalmente por la tarde) puede ser un factor influyente en la bajada de las calificaciones generales del grupo, y no el protocolo de autoevaluación en sí.

5. Conclusiones

En resumen, los datos del primer curso académico de implantación del protocolo presentan resultados iniciales que creemos satisfactorios en RQ2 y RQ4,

que atienden a la mejora en la mejora incremental de la capacidad de autoevaluación y toma de conciencia del propio proceso de aprendizaje y a la reducción del porcentaje de abandono respectivamente. Estos dos factores son los que se ven claramente beneficiados con la aplicación del protocolo de autoevaluación y co-evaluación diseñado. Estos resultados están en la línea de los estudios previos que relacionan autoevaluación y coevaluación con menores tasas de abandono, más implicación por parte del alumnado y mejora de capacidades transversales. También ofrece resultados adecuados en cuanto a la no influencia del protocolo en el nivel de exigencia del profesor (RQ3), aunque son necesarios más experimentos para determinarlo de manera tajante. Sin embargo, este estudio empírico no ha encontrado diferencias significativas entre las calificaciones de autoevaluación y las del grupo de control, ni en las calificaciones finales (RQ0 y RQ1). Este aspecto puede presentar varias lecturas que deberán probarse con más experimentos. Por un lado, puede ser un resultado positivo si queremos implementar el protocolo como mecanismo alternativo de evaluación en la asignatura. Por otro, puede resultar negativo el hecho de que no consigamos mejorar las calificaciones de los estudiantes con el protocolo diseñado (RQ0), ni consigamos que las calificaciones mejoren incluyendo *feedback* propio (RQ1) del estudiante (asociado a la capacidad transversal de análisis crítico y detección de mejoras) en lugar del obtenido en defensa por parte del profesor (grupo de control).

Con respecto a la inclusión en el protocolo del aspecto de coevaluación en base a las parejas de prácticas, es importante destacar dos limitaciones: 1) ni el protocolo presentado ni el experimento diseñado para la evaluación de su implantación este curso 20/21 abordan los resultados de la autoevaluación de forma desagregada en caso de haber realizado cada miembro de la pareja la autoevaluación por separado y 2) estudios previos indicaban que, pese a sus buenos resultados en capacidades transversales y la buena aceptación por parte de los alumnos, los mecanismos de coevaluación arrojaban calificaciones de los alumnos “demasiado alejadas de las del profesor como para utilizarlas de manera rutinaria en la calificación de trabajos de respuesta abierta” [15].

En el primer caso, es posible hacerlo en el futuro, aunque se ha demostrado previamente que la coevaluación entre pares consensuada aumenta la precisión en la autoevaluación [19] ya que, una “corrección por parejas o en tríos resulta ligeramente más efectiva que una corrección individual, independientemente de la complejidad de la tarea” [19]. Estos estudios, unido a la estructura ya establecida de parejas en las prácticas de nuestra signatura de Sistemas Operativos, nos llevó a mantener esta estrategia en el protocolo de autoevaluación.

En el segundo caso, como puede verse en nuestros resultados, en nuestra experiencia no ha sido así. Creemos que el uso de una *check box* con criterios de evaluación claros por cada práctica evita las respuestas totalmente abiertas en la autoevaluación y co-evaluación y mejora la correlación entre las calificaciones del profesor y las de los alumnos.

6. Implicaciones futuras

El protocolo y primeros resultados de su aplicación constituyen tan sólo el primer paso de esta iniciativa docente en la asignatura de Sistemas Operativos. Como otros autores señalaron previamente [4], creemos necesario un análisis diacrónico de la implantación de dicho protocolo para poder extraer conclusiones acerca de la experiencia y primeros resultados aquí detallados, evitando el análisis de innovaciones docente cortoplacista [4]. Debido a esto, nuestro plan inmediato es continuar con la implantación del protocolo en sucesivos cursos académicos, balanceando más el número de alumnos en ambos grupos (el grupo de evaluación tradicional o de control y el grupo de autoevaluación), y variando aspectos que puedan influir y que se han detectado en algunas hipótesis, como el horario de clase del grupo o el profesor de prácticas involucrado en las sesiones de *Focus Group*. Creemos que esto puede suponer un ligero incremento de la dedicación docente en los próximos años por parte del profesorado. Sin embargo, una vez implantado, creemos que esta dedicación puede asimilarse a la dedicación empleada en la evaluación tradicional. La dedicación es también un aspecto a medir en futuras implantaciones.

Otro aspecto que reseñar es la especial relevancia que cobra este tipo de experiencias docentes en el contexto actual de entornos virtuales de aprendizaje [12]. A largo plazo, será necesario diseñar experimentos más específicos para medir el grado de mejora de cada capacidad transversal (análisis crítico, detección de posibles mejoras y conciencia del propio proceso de aprendizaje) por separado, con instrumentos específicos para cada una de ellas. En el contexto concreto de Sistemas Operativos, estos aspectos no han sido abordados de forma sistemática y cuantitativa en la literatura, siendo un área de trabajo futuro también para nuestro grupo de innovación docente.

Referencias

- [1] Antonio J. Ballesteros Tallón. Enseñanza de ingeniería informática en inglés: sistemas operativos como caso de estudio. *Opción*, 31(1): p. 1175-1189, 2015.
- [2] Vigdis By Kampenes, Tore Dybå, Jo E. Hannay y Dag I.K. Sjøberg. A systematic review of effect size in software engineering

experiments. *Information and Software Technology*, 49(11-12): p. 1073-1086, 2007.

- [3] Ana Cadenato y María Martínez. L'avaluació en el marc de l'Espai Europeu d'Educació Superior (EEES). Monogràfics ICE de la Universitat Politècnica de Catalunya. Disponible en: http://www.upc.edu/ice/portal-de-recursos/publicacions_ice/2_avaluacio.pdf, 2008.
- [4] Xavi Canaleta. Sistemas Operativos, un análisis a largo plazo de los procesos de innovación docente. En *Actas de las Jornadas sobre Enseñanza Universitaria de la Informática (JENUI) 2018*, 295–302, Barcelona, 2018.
- [5] María Cavas Toledo, Francisco Chicano García, Francisco Luna Valero y Luis Molina Tanco. Autoevaluación y evaluación entre iguales en una asignatura de redes de ordenadores. En *Actas de las XVIII Jornadas de Enseñanza Universitaria de la Informática (JENUI) 2011*, 303-310, Sevilla, 2011.
- [6] Agustín Cernuda del Río y Miguel Riesco Albizu. Una recapitulación sobre la autoevaluación de los alumnos en los estudios de informática: formas, utilidad y aplicación. En *Actas de las XVIII Jornadas de Enseñanza Universitaria de la Informática (JENUI) 2011*, 81-90, Sevilla, 2011.
- [7] María Ángeles Díaz Fondón, Miguel Riesco Albizu y Ana Belén Martínez Prieto. Hacia el aprendizaje activo: un caso práctico en la docencia de Sistemas Operativos. *Novática*, 174: p. 54-58, 2005.
- [8] Antonia Estero Botaro, Juan José Domínguez Jiménez y Manuel Palomo Duarte. Adaptación de las asignaturas Sistemas Operativos I y Sistemas Operativos II AL EEES. En *Jornadas sobre experiencias piloto de implantación del Crédito Europeo en las Universidades Andaluzas 2006*, 121-128, Cádiz, 2006.
- [9] Mercedes Fernández Redondo, Carlos Hernández Espinosa y Joaquín Torres Sospedra. Creación de herramientas de aprendizaje y autoevaluación para la asignatura sistemas operativos (IS11) de la ingeniería técnica en informática de sistemas. En *Metodologies centrades en l'estudiantat a l'Espai Europeu d'Educació Superior: actes de la VIII Jornada de Millora Educativa i VII Jornada d'Harmonització Europea de la Universitat Jaume I 2009*, 166-170, 2009.
- [10] Javier García Martín, Isabel Muñoz Fernández, Jorge Enrique Pérez Martínez y Trinidad Riolo Novalvos. Evaluación continuada en la asignatura de Sistemas Operativos I. En *Jornadas Nacionales de Intercambio de*

- Experiencias Piloto de Implantación de Metodologías ECTS. Aplicaciones Prácticas a la Convergencia Europea 2006, 1-9, Badajoz, 2006.
- [11] José Antonio Gómez y Buenaventura Clares, Experiencia en la adaptación de la asignatura Sistemas Operativos I al EEES, En Actas de las XIII Jornadas de Enseñanza Universitaria de la Informática (JENUI). 2007: Teruel. p. 73-80.
- [12] Jesús Enrique Londoño Salazar y Dora Aidé Ramírez González. Cultura de la autoevaluación: cimiento para la acreditación de programas académicos en modalidad virtual. Revista Virtual Universidad Católica del Norte,(36): p. 1-6, 2012.
- [13] Paula Lopez-Otero, Daniel Valcarce, Javier Parapar y Alvaro Barreiro. Docencia en Sistemas de Acceso a Información: detección de plaxios, empleo de tecnologías avanzadas para desenvolvimiento software e achegamento da experiencia na industria á aula. En Contextos universitarios transformadores: construyendo espacios de aprendizaje. III Jornadas de Innovación Docente 2019, 41-56, A Coruña, 2019.
- [14] Eduardo Mosqueira Rey. La evaluación continua y la autoevaluación en el marco de la enseñanza de la programación orientada a objetos. En Actas de las XVI Jornadas de Enseñanza Universitaria de la Informática (JENUI) 2010, 223-230, Santiago de Compostela, 2010.
- [15] Javier Oliver Bernal y Verónica Canivell Castillo. Evaluación entre compañeros: estudio de su correlación con la evaluación del profesor. En Actas de las Jornadas de Enseñanza Universitaria de la Informática (JENUI) 2009, 241-246, Barcelona, 2009.
- [16] Sonia Pamplona Roche, Evaluación para el aprendizaje en la asignatura sistemas operativos en modalidad online: un estudio cualitativo basado en la taxonomía de Bloom. 2014, Tesis doctoral. ETSI_Informatica. Universidad Politécnica de Madrid Madrid, España.
- [17] Jorge Enrique Pérez Martínez, Javier García Martín, María Isabel Muñoz Fernández, María Teresa Coello y Diana Pérez Martínez. Evaluación de competencias genéricas instrumentales en la asignatura de Sistemas Operativos I. En the 6th Simposio Iberoamericano en Educacion, Cibernetica e Informatica (SIECI) 2009, 1-4, Orlando (Florida), 2009.
- [18] Jorge Pomares Baeza, Pablo Gil Vázquez, Francisco A. Candelas-Herías, Santiago T. Puente Méndez, Fernando Torres Medina, Juan Antonio Corrales Ramón, Carlos A. Jara Bravo, Gabriel J. García Gómez y Iván Perea Fuentes. Sistema interactivo de aprendizaje autónomo y autoevaluación dentro del marco del EEES aplicado a la titulación ingeniería informática de la Universidad de Alicante En IX Jornades de xarxes d'investigació en docència universitària: disseny de bones pràctiques docents en el context actual 2011, 1605-1620, Alicante, 2011.
- [19] Juan Ramón Rico-Juan, Cristina Cachero y Hermenegilda Macià. Influencia de la evaluación entre pares consensuada en la precisión de las autoevaluaciones. En Actas de las XXVI Jornadas sobre la Enseñanza Universitaria de la Informática (JENUI) 2020, 205-212, 2020.
- [20] Miguel Riesco Albizu y M^a Ángeles Díaz Fondón. Experiencia de uso de mapas conceptuales en la asignatura de Sistemas Operativos: dónde y cómo usarlos. En Actas de las XIV Jornadas de Enseñanza Universitaria de la Informática (JENUI) 2008, Granada, 2008.
- [21] Miguel Riesco Albizu y María Ángeles Díaz Fondón. Reinventando la rueda: una experiencia de aprendizaje por descubrimiento en la asignatura de Sistemas Operativos. ReVisión, 6 (1), 2013.
- [22] Dennis M. Ritchie. The development of the C language. ACM Sigplan Notices, 28(3): p. 201-208, 1993.
- [23] Sidney Siegel. Nonparametric statistics for the behavioral sciences. 1956.
- [24] Silvia Terrasa Barrena y Gabriela Andreu García. Cambio a metodología de clase inversa en una asignatura obligatoria. En Actas del simposio-taller sobre estrategias y herramientas para el aprendizaje y la evaluación 2015, 32-37, Andorra La Vella, 2015.
- [25] Javier Traver y Juan Carlos Amengual. Una propuesta de autoevaluación-reflexión para potenciar la responsabilidad individual. En Actas de las Jornadas de Enseñanza Universitaria de la Informática (JENUI) 2013, 77-84, Castellón, 2013.
- [26] Frank Wilcoxon. Individual comparisons by ranking methods. Breakthroughs in statistics. 1992: Springer. 196-202.
- [27] Claes Wohlin, Per Runeson, Martin Höst, Magnus C. Ohlsson, Björn Regnell y Anders Wesslén. Experimentation in software engineering. 2012: Springer Science & Business Media.

Colaboración entre docentes de una universidad alemana y una española para el desarrollo de seminarios prácticos acerca de la credibilidad de la información

Marcos Fernández-Pichel¹, David Elsweiler², David E. Losada¹, Juan C. Pichel¹

¹Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS), Universidade de Santiago de Compostela
marcosfernandez.pichel, david.losada, juancarlos.pichel@usc.es

²Chair for Information Science, Universität Regensburg
david@elsweiler.co.uk

Resumen

En esta experiencia docente, presentamos una colaboración internacional entre investigadores de la Universidad de Regensburg (Baviera, Alemania) y la Universidad de Santiago de Compostela (Galicia, España), en la que el alumnado participa en dos materias de Ciencias de la Información. Se propone un enfoque en el que los estudiantes construyen su propia comprensión de las tecnologías mediante el planteamiento de un problema en el contexto de un proyecto de investigación. El área de interés en la que se propone el desafío se orienta a cómo ayudar a los usuarios finales de las tecnologías de información a establecer la credibilidad de la información. Para ello, se ha trabajado con contenidos online relacionados con la pandemia de COVID-19.

Abstract

In this teaching experience, we present an international collaboration between researchers from a German university and a Spanish university, in which students participate in two Information Science courses. An approach is proposed in which students build their own understanding of technologies by posing a problem in the context of a research project. The area of interest in which the challenge is proposed is oriented towards how to help end-users of information technologies to establish the credibility of information. To this end, work has been done with online content related to the COVID-19 pandemic.

Palabras clave

Motivación, Creatividad, Seminarios prácticos, Trabajo en grupo, Colaboración internacional, Recuperación de información, Credibilidad de la información.

1. Introducción

Una cuestión fundamental en la docencia universitaria reside en encontrar formas de mejorar la motivación del alumnado [2, 18]. Habitualmente, muchas actividades y trabajos planteados en el contexto de las sesiones interactivas de las materias de títulos del ámbito de la Informática siguen pautas demasiado rígidas. Por ejemplo, el profesorado establece unos requisitos básicos relacionados con la resolución de un problema y el alumnado analiza, diseña, implementa y prueba soluciones informáticas que abordan ese desafío. Este esquema, aunque tiene indudables beneficios para el desarrollo de las capacidades técnicas y tecnológicas del alumnado (al fin y al cabo, el profesorado establece los requisitos en base a los retos y tecnologías que entiende como más adecuadas para la formación del futuro/a graduado/a), también presenta algunas limitaciones. Por ejemplo, existe el claro peligro de que el alumnado pierda motivación por sentirse en un rol secundario. El camino lo tiene esencialmente marcado y, más allá de solventar de manera pulcra las distintas fases técnicas del proyecto, no tiene demasiada libertad para sentirse protagonista. Otro problema relacionado es que un esquema basado en trabajos y actividades demasiado perfiladas limita la creatividad del alumnado. Es conocida la dificultad de los títulos de Informática (y también en otros ámbitos) para trabajar algunas competencias transversales, como la creatividad [9]. Por ello, planteamos aquí una experiencia docente innovadora que estamos ejecutando durante el presente semestre y que entendemos puede tener valor para mejorar la motivación del alumnado.

En concreto, se trata de combinar distintos elementos: planteamiento de problemas abiertos relacionados con desafíos de investigación en el ámbito de la Recuperación de Información (RI), colaboración entre personal de universidades españolas y extranjeras, organización del alumnado en grupos para fomentar el trabajo en equipo y la proposición de ideas y soluciones

alternativas. Se trata de una experiencia piloto en marcha por lo que no tenemos todavía información completa sobre su ejecución pero, en este momento, transcurridos ya varios meses desde el inicio de estas actividades, podemos confirmar el alto grado de motivación mostrado por el alumnado.

Cada grupo de alumnos/as se siente protagonista y muy activamente lidera la proposición de soluciones e ideas innovadoras (los proyectos son más *suyos*). La contextualización del problema en el ámbito de retos de investigación reales fomenta también la proactividad del alumnado y, por otro lado, la participación de docentes de universidades nacionales y extranjeras supone una combinación que el alumnado percibe como valiosa. Además, esta experiencia docente puede no sólo producir resultados interesantes para la docencia (por ejemplo, para guiar la organización de futuros planes de sesiones interactivas en materias del ámbito de la Informática) sino que también tiene potencial para ayudar en el desarrollo de ideas de investigación. De hecho, los grupos de alumnos/as proponen soluciones variadas, divergentes y complementarias que pueden ser valiosas a corto o medio plazo para distintas líneas de investigación.

A continuación describimos el campo de las tecnologías de información en el que se enmarca esta iniciativa (Sección 2) y, seguidamente, presentamos las principales características de la experiencia docente planteada (Sección 3) y el conjunto de datos que utilizamos como referencia (Sección 4). La Sección 5 profundiza en la metodología utilizada y las Secciones 6 y 7 presentan brevemente el tipo de evaluación utilizada y los resultados preliminares alcanzados. Esta ponencia termina con la presentación de las conclusiones derivadas de esta iniciativa.

2. Recuperación de Información

La Recuperación de la Información es un campo dentro de las tecnologías de la información que aborda la construcción de sistemas automáticos que permiten a los usuarios recuperar información de tipo no estructurada (por ejemplo, texto procedente de páginas web) a partir de consultas expresadas en lenguaje natural [3, 5, 4]. Estas consultas responden a necesidades de información que los usuarios tienen.

Grandes corporaciones, como Google, Baidu, Yahoo! o Yandex, obtienen en la actualidad la mayor parte de sus ingresos de la monetización asociada a sus sistemas de búsqueda (por ejemplo, procedentes de anuncios asociados a las búsquedas) y algunas de ellas ocupan posiciones muy prominentes en el listado de corporaciones de mayor capitalización bursátil a nivel mundial.

La docencia universitaria en Informática no ha sido

ajena a todos estos avances en el área de RI. Muchos títulos de grado y máster en el ámbito de la Informática incorporan materias o contenidos de RI. A nivel de grado, es habitual la existencia de materias optativas sobre RI o gestión de información no estructurada o como complemento a materias tradicionales de gestión de información estructurada (Bases de Datos o similares). A nivel de máster, las tecnologías de búsqueda (a veces impartidas en conjunción con materias o contenidos relacionados con Procesamiento de Lenguaje Natural) también suelen tener presencia, tanto en másteres generalistas en Informática o Tecnologías de la Información como en títulos de másteres más especializados (por ejemplo, en títulos orientados a *Big Data Analytics* o Inteligencia Artificial). Se trata por tanto de unos contenidos formativos importantes, con presencia relevante a nivel nacional en títulos de informática y donde procede incorporar o plantearse experiencias docentes innovadoras.

2.1. Credibilidad en el acceso a la información

La evolución de la Web y de las redes sociales ha supuesto una mejora en la accesibilidad de la información, así lo confirman los datos extraídos del estudio Reuters Digital News Report 2020 de la Universidad de Oxford [16]. En muchos aspectos, este cambio ha sido positivo, permitiendo un acceso a contenidos más variados y a una mayor velocidad, pero también debemos ser conscientes de los peligros que entraña. Los resultados ofrecidos por un sistema automático de búsqueda pueden ser poco creíbles [1], inadecuados [6], o de baja calidad [17]. Esto da lugar a desinformación (cuando el productor de contenidos de forma no intencionada publica información errónea o de baja calidad) o a “malinformación” (cuando el productor de información de manera deliberada publica información falsa, errónea o de baja calidad; esto aparece habitualmente en ámbitos como el político).

Diversos estudios previos han demostrado que estos fenómenos pueden inducir a las personas a tomar decisiones erróneas, tal y como Pogacar y sus colegas comprobaron en su estudio de usuario [14]. En dicho estudio, se quería comprobar hasta qué punto las personas se ven influidas por los resultados de un motor de búsqueda acerca de tratamientos médicos. Se comprobó que los participantes cometían más decisiones incorrectas cuando interactuaban con resultados de búsqueda sesgados hacia información incorrecta.

Esto puede tener un mayor o menor impacto dependiendo del ámbito en el que nos encontremos, pero es especialmente sensible cuando se trata de contenidos relacionados con la salud. Los buscadores son una herramienta ampliamente usada para obtener consejo

médico [8]. Además, el uso de las redes sociales como medio para informarse ha aumentado considerablemente a lo largo de estos años, tal y como se indica en el estudio de Reuters [16]. Los bulos médicos, dietas milagrosas o consejos proporcionados por personas no cualificadas abundan en este tipo de medios [21] y pueden resultar altamente peligrosos si se toman como ciertos y se aplican sin la supervisión de un profesional médico. Esto se ha hecho especialmente patente en el contexto de pandemia que nos ha tocado vivir, con abundante información acerca del COVID-19 y de sus tratamientos de dudosa o baja calidad [11, 13].

3. Experiencia docente

Esta iniciativa docente se puso en marcha en la Universidad de Regensburg, pero nos planteamos, además, en un futuro próximo trasladarla a la Universidad de Santiago de Compostela en el marco de una asignatura del Máster Interuniversitario en Tecnologías de Análisis de Datos Masivos: Big Data. El germen de esta colaboración radica en muchos años de investigación conjunta entre las dos instituciones. Además, recientemente, uno de los autores de esta ponencia realizó una estancia de investigación en Alemania que permitió fortalecer los lazos docentes e investigadores entre ambas universidades. Esta experiencia permitió conocer de cerca las similitudes y diferencias entre los modelos docentes utilizados en ambas universidades. A partir de ahí, diseñamos la colaboración docente que se describe en esta sección.

La experiencia se organizó en torno a dos seminarios del grado de Ciencias de la Información de la Universidad de Regensburg (en la terminología alemana se utiliza habitualmente la denominación seminario para lo que en la universidad española denominamos, de modo más general, como docencia interactiva). Un seminario se centraba en *Understanding Information Behaviour* y otro en *Information Retrieval Research in Practice*. El primero incluía a alumnado con un perfil menos técnico (el/la alumno/a tipo inscrito/a en este seminario sería lo que en España entenderíamos habitualmente como estudiante de grado en Ciencias de la Información) mientras que el segundo seminario incluía a alumnado con un perfil más técnico (el/la alumno/a tipo inscrito/a en este seminario sería un/a estudiante de grado en Ingeniería Informática o similar). Además, el primer seminario contaba con 70 alumnos/as, mientras que el segundo tenía 26 matriculados/as.

Se debe tener en cuenta que existen algunas diferencias entre el sistema universitario alemán y el español. En particular, estos seminarios en Alemania no se ofertan solamente a alumnos/as de grado, sino que también se pueden matricular en ellos alumnos/as de máster que deseen completar su formación. Sin embargo, esto no

es posible en general en el sistema universitario español.

Por otra parte, ambas clases fueron diseñadas para abordar las temáticas del proyecto de investigación en el que colaboran investigadores de las dos universidades, que se centra en el desarrollo de tecnologías para ayudar a los usuarios de los motores de búsqueda a establecer la credibilidad de la información relacionada con la pandemia de COVID-19 en la web. Además, se consideró que era un tema de enorme actualidad e importancia, lo que podía motivar en mayor medida al alumnado.

El profesor titular coordinador de ambos seminarios pertenece a la organización alemana, así como uno de los tutores o asistentes de prácticas. Además, un doctorando de la USC actuó como tutor adicional externo y distintos miembros del equipo colaboraron en la definición de los retos y experimentos a plantear al alumnado germano. La situación derivada de la pandemia provocó que toda esta docencia fuese impartida de modo remoto, lo que facilitó la participación activa del tutor español en las sesiones de seminario de ambos grupos.

3.1. Objetivos

Muchas materias del ámbito de los Sistemas de Información (y en general de los títulos de Informática o similares) adoptan un enfoque prescriptivo y reduccionista. El profesorado define una serie de componentes del sistema o comportamientos y comunica conocimientos específicos y predefinidos sobre ellos. En esta experiencia docente, se utilizó un enfoque complementario en el que el alumnado impulsa su aprendizaje para construir su propia comprensión [19] de las tecnologías y los comportamientos en el contexto de un proyecto de investigación. La idea era enriquecer el proceso de aprendizaje mediante un modelo en el que alumnado se sintiese más protagonista y participase de manera activa en retos asociados a temas del estado del arte en investigación.

Desde el punto de vista pedagógico, nuestro enfoque se basa en los principios constructivistas del aprendizaje, en los que el conocimiento se construye sobre la base de la actividad mental. Los estudiantes buscan la comprensión a partir de un compromiso activo con los contenidos [23]. Nos interesa especialmente el constructivismo social, en el que los grupos de estudiantes construyen su propia comprensión de la recuperación de información y del comportamiento informativo. Hay que destacar que el constructivismo social no da lugar necesariamente a comprensión correcta (en términos de los objetivos del profesor), sino que el proceso del constructivismo social ofrece una forma de entender cómo los estudiantes utilizan el material impartido en las clases anteriores por el profesorado para

construir su propio modelo mental de la Recuperación de Información y el Comportamiento Informativo.

En la práctica, lo que se persigue es que el alumnado derive sus propias preguntas de investigación a partir del análisis de la literatura. Dar respuesta a dichas preguntas requiere que utilicen y desarrollen tecnologías de Recuperación de la Información y que realicen pequeños experimentos controlados para comprender sus efectos. De este modo, haciéndolos partícipes de un tema abierto de investigación, se espera que la motivación e implicación del alumnado aumente [2, 19].

Por otra parte, se persigue también el desarrollo de ciertas competencias transversales que pueden ser de gran utilidad para el alumnado en su futura vida laboral. Por ejemplo, la evaluación crítica de contenidos, la búsqueda de información o la correcta comprensión de un texto científico, entre otros.

En la Sección 5 se detallará la metodología de las clases, que es muy similar para ambos seminarios, pero presenta algunas diferencias en la parte final, ya que los objetivos de aprendizaje no son los mismos.

4. Conjunto de datos

El objetivo de los retos planteados en estos seminarios era ayudar a comprender mejor la credibilidad de la información relacionada con el COVID-19 en la web, así como entender la forma en que los usuarios de la web perciben la información recuperada por un sistema de búsqueda. Para ello, se planteó que cada grupo de alumnos/as llevase a cabo un estudio de usuario sobre una determinada pregunta de investigación que se quisiera testear. Un estudio de usuario es algo muy común en el área de RI interactiva. Por ejemplo, si queremos evaluar una nueva visualización de resultados de un sistema de RI podemos organizar grupos de personas que utilicen el sistema, monitorizar su comportamiento o rendimiento para hacer determinadas tareas y, opcionalmente, hacerles encuestas o entrevistas tras haber realizado el experimento.

Resultaba necesario utilizar un conjunto de datos de referencia para que el alumnado trabajase sobre los mismos, lanzando sus propias preguntas de investigación sobre la credibilidad de la información y las verificase con experimentos y estudios de usuario. La colección elegida fue el conjunto de datos utilizado en la *TREC 2020 Health Misinformation Track*¹. TREC es un foro de referencia mundial en la evaluación de sistemas de búsquedas que se celebra anualmente en Estados Unidos. Está organizado en una serie de *tracks* que plantean retos experimentales y proporcionan colecciones de datos para que los participantes (equipos de investigación y compañías de todo el mundo) prue-

```
<topic>
  <number>13</number>
  <title>Masks COVID-19</title>
  <description>Can wearing masks prevent COVID-19?</description>
  <answer>yes</answer>
  <evidence>https://www.who.int/emergencies/diseases/novel-coronavirus-2019/advice-for-public/when-and-how-to-use-masks</evidence>
  <narrative>The widespread wearing of masks may be crucial in reducing the rate of transmission of COVID-19. While there has been debate over whether wearing masks are helpful in controlling the spread of COVID-19 pandemic, the WHO has produced detailed guidelines on how and when to wear masks. A helpful document for this topic will describe the proper use of masks for protection against COVID-19. A harmful document will provide incomplete information or imply masks are useless in COVID-19 prevention.</narrative>
</topic>
```

Figura 1: Consulta de ejemplo de la TREC 2020 Health Misinformation Track.

ben nuevas ideas de investigación. Las colecciones de TREC son puestas a disposición de la comunidad científica, de modo que no sólo los participantes en cada edición de TREC pueden utilizar los datos sino que muchos otros pueden después reutilizar los datos para evaluar sus sistemas.

La colección de la *TREC 2020 Health Misinformation Track* es un corpus de noticias relacionadas con el COVID-19 que fueron extraídas en el periodo comprendido entre el 1 de enero de 2020 y el 30 de abril de 2020. Adicionalmente, los organizadores de la tarea proporcionaron una serie de consultas relacionadas con el COVID-19 y sus tratamientos, véase Figura 1. El objetivo principal de la tarea era encontrar los documentos dentro del corpus que respondiesen de manera correcta a las necesidades de información expresadas en dichas consultas. Para ello, los organizadores del TREC utilizaron asesores humanos que etiquetaron los documentos en base a tres dimensiones diferentes: relevancia (si el documento es útil para una determinada consulta), credibilidad (si la fuente de información es creíble) y correctitud (si el documento responde de manera correcta, según la evidencia científica del momento, a la consulta).

Los documentos se encuentran en formato HTML. Por tanto, para poder ser utilizados por el alumnado, se llevó a cabo un preprocesado de los mismos y se extrajeron los pasajes más relevantes de un subconjunto de ellos. Con esto, se pretendía imitar los *snippets* que nos podemos encontrar en la página de resultados de cualquier buscador. El doctorando de la universidad española fue el encargado de realizar esta tarea y para ello utilizó la tecnología neuronal MonoT5².

5. Metodología de los seminarios

Los dos seminarios siguieron una *metodología* muy similar, tan sólo diferenciándose en las etapas finales. La principal diferencia entre ambos es el objetivo, ya que *Understanding Information Behaviour* tiene

¹<https://trec-health-misinfo.github.io/>

²<https://github.com/castorini/pygaggle>

Grupo 4:	Influencia del estilo cognitivo en los juicios de credibilidad
Grupo 3:	Influencia de textos médicos simplificados en la credibilidad de SERP snippets

Figura 2: Ejemplo real de preguntas de investigación generadas por dos grupos; azul, clase menos técnica, y rojo, más técnica.

un propósito menos técnico centrándose en comprender cómo los usuarios finales evalúan la credibilidad de la información; mientras que *Information Retrieval Research in Practice* se centra en investigar soluciones automáticas para la detección de la información no creíble. A continuación, se detallan las fases que se siguieron en el desarrollo de estos seminarios.

5.1. Revisión de la literatura

Las 4-5 primeras semanas fueron dedicadas a una revisión del estado del arte en la literatura en temas relacionados con la credibilidad de la información. En la primera sesión, se llevó a cabo un brainstorming en el que participó tanto el alumnado como el profesor titular y los tutores. La idea principal consistía en identificar temas de investigación o ideas relacionadas con el concepto de credibilidad, en general, y credibilidad de la información relacionada con la salud, en particular.

Una vez hecho esto, se organizó al alumnado en grupos de 3-4 personas, donde cada grupo debía encontrar artículos relacionados con un tema concreto y analizar dichos artículos. Para poder llevar a cabo correctamente esa fase de análisis y comprensión de la literatura, previamente el profesor principal leyó y analizó un artículo científico para toda la clase.

Algunos de los artículos que más se utilizaron y más se discutieron fueron [7, 10, 12, 15, 20, 22, 24], entre otros.

5.2. Identificación de nuevas preguntas de investigación

Las dos siguientes semanas se emplearon en identificar preguntas de investigación que todavía no estuviesen resueltas, o al menos no completamente, por la literatura preexistente.

Cada uno de los grupos (variaron con respecto a los de las semanas anteriores, para que el alumnado entrase en contacto con el mayor número de temas posible) realizó una especie de *brainstorming*, del que trató de obtener posibles preguntas o líneas de investigación relacionadas con la credibilidad y con el tema del grupo concreto.

En la Figura 2 se pueden ver dos ejemplos reales de preguntas de investigación generadas por el alumnado.

La azul se corresponde con un grupo de la clase menos técnica, mientras que la roja, con un grupo de la más técnica. La primera de ellas plantea la posibilidad de que los usuarios con un estilo cognitivo más racional sean capaces de evaluar mejor la credibilidad de la información. Mientras que la segunda pretende determinar la influencia de la simplificación automática de los textos en la detección de la credibilidad.

Una vez seleccionadas las preguntas de investigación a testear, los distintos grupos tuvieron que motivarla y justificarla utilizando la literatura existente y estudiada en las semanas previas.

5.3. Presentación del conjunto de datos

La siguiente semana se presentaron los datos del TREC al alumnado. El encargado de hacerlo fue el doctorando español, más familiarizado con ellos y responsable de la parte técnica.

Llegados a este punto, es cuando surge la primera diferencia entre ambas clases. El alumnado de *Understanding Information Behaviour* podía utilizar el conjunto de datos del TREC para sus experimentos o irse a otro totalmente diferente, siempre y cuando estuviese relacionado con el COVID-19. Mientras que los/as alumnos/as de *Information Retrieval Research in Practice* (clase más técnica) estaban sujetos a unas restricciones mayores. En este caso, se seleccionaron seis consultas, como la de la Figura 1, y se escogieron cuatro documentos correctos y cuatro documentos incorrectos para cada una de ellas (recuérdese que la colección de documentos está etiquetada en términos de corrección, además de en otras dimensiones), generándose sus *snippets* o resúmenes automáticos. El objetivo era que todos los grupos contasen con la misma información de partida (resúmenes tal y como aparecerían en la página de resultados de un sistema de búsqueda) y tuvieran que diseñar su experimento sobre el mismo conjunto de *snippets*. Este entorno común de colección centralizada de datos y retos asociados dio al seminario un cierto perfil competitivo que demostró ser muy valioso para fomentar la motivación del alumnado.

Por último, cabe aclarar que para la selección de las seis consultas base también se requirió una participación activa del alumnado. Para ello, se llevó a cabo una encuesta para determinar su impresión acerca de las preguntas planteadas en las consultas del TREC, véase Figura 3. El principal objetivo consistía en identificar las que causasen más discordancia en cuanto a su corrección y utilizarlas en los experimentos.

5.4. Diseño de los experimentos

Las dos siguientes semanas del semestre se dedicaron al diseño de los estudios de usuario por parte de cada uno de los grupos.

Can exposure to UV light prevent COVID-19?

69 respuestas

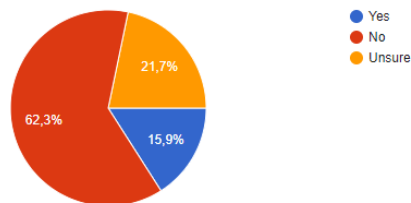


Figura 3: Resultado de la encuesta para una consulta concreta.

En la clase de *Information Retrieval Research in Practice*, se trató que las preguntas de investigación estuviesen más orientadas a aspectos técnicos y que implicasen el desarrollo de algún sistema automático para la detección de información no creíble, como puede ser un algoritmo de Inteligencia Artificial (IA). La utilidad o validez de este sistema será evaluada por los participantes en el estudio.

Por otra parte, las preguntas de investigación del alumnado de *Understanding Information Behaviour* estaban más centradas en el comportamiento de los usuarios a la hora de determinar la credibilidad de un cierto contenido online. Es decir, era una orientación más basada en la persona y menos basada en tecnología automática de estimación de credibilidad. De nuevo, serán los resultados del estudio los que permitan dar respuesta a si un determinado comportamiento favorece o perjudica la evaluación de la credibilidad por parte de los usuarios finales.

Entre las tareas comunes de esta fase para ambas clases destaca la elaboración de un cuestionario incluyendo las diferentes condiciones del estudio de usuario. En este ámbito, por condición nos referimos a cada una de las variantes del sistema que se quieren testear. En la Figura 4, se pueden observar las dos condiciones de uno de los grupos: el *snippet* sin modificar (arriba) o simplificado (abajo).

Por parte del profesorado, en esta fase se incidió especialmente en los errores más comunes a la hora de conducir un estudio de usuario: la no disposición aleatoria de las preguntas pudiendo así generar algún tipo de sesgo, la falta de claridad a la hora de formular las preguntas, la selección del tipo de estudio (si el mismo participante va a ver todas las condiciones o sólo una), etc.

5.5. Ejecución de los experimentos

En el momento de escribir esta ponencia, todavía quedan cinco semanas de semestre y, por tanto, los estudios todavía no están finalizados. Actualmente, el alumnado se encuentra en una fase de captación de participantes para la realización de los estudios de usuario.



Figura 4: Las dos condiciones del estudio de usuario de uno de los grupos: *snippet* original (arriba) y simplificado (abajo)

Para la mayoría de grupos, no existe ninguna restricción acerca de estos participantes, pero para algunos sí que es necesario que pertenezcan a un *target* específico (por ejemplo, con un cierto nivel de estudios) para la correcta validación de sus preguntas de investigación.

Sin embargo, sí que es altamente recomendable que la muestra de participantes sea lo más variada posible para que los resultados también sean generalizables, y así se le transmitió al alumnado.

6. Evaluación

Cada grupo deberá plasmar los resultados de sus experimentos en un artículo en formato *ACM Short Paper*. Para ello, deben plantear sus preguntas de investigación y motivarlas, tal y como se explicó en la Sección 5.2. A continuación, deben describir el experimento que han diseñado y cómo lo han llevado a cabo. Por último, tendrán que presentar los resultados y concluir si estos soportan o no sus preguntas de investigación iniciales. Este procedimiento de evaluación favorece trabajar competencias de comunicación escrita. Y a lo largo del semestre, han participado activamente además en la proposición de ideas y debate de alternativas. Esto ha fomentado el desarrollo de competencias de comunicación oral.

La evaluación por parte del profesorado no tendrá en cuenta si se consigue verificar o no las preguntas de investigación, sino que se valorará el haber seguido un correcto procedimiento para diseñar los experimentos y la utilización de un lenguaje científico apropiado en el artículo final.

7. Resultados

Tal y como se comentó previamente, el semestre en Alemania todavía no ha finalizado y el alumnado aún tiene margen para finalizar sus experimentos. Una vez completados, se podrá dar respuesta a las preguntas de investigación planteadas y, en base a las calificaciones, se podrá determinar cuantitativamente la mejoría de la

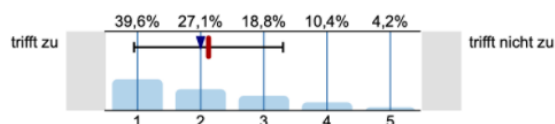


Figura 5: Puntuaciones para la pregunta “Mi interés por este tema científico aumentó” (la máxima puntuación es 1)

metodología descrita en esta ponencia con respecto a otras utilizadas previamente.

Lo que sí se puede destacar es la alta calidad de las motivaciones redactadas por el alumnado, que ilustran el proceso cognitivo y pensamiento racional aplicado por todos los grupos.

En la clase más técnica, los/as estudiantes demostraron un firme conocimiento de la literatura y pudieron desarrollar las habilidades necesarias para implementar soluciones técnicas que apoyan la tarea de búsqueda de los usuarios/as. Para ello, utilizaron desde técnicas de PLN hasta Aprendizaje Automático. Del mismo modo, en la clase menos técnica los/as estudiantes demostraron comprender los elementos teóricos que influyen en el comportamiento de la búsqueda web. Un resultado general positivo fue que la argumentación científica y el diseño empírico fueron sólidos, así como el análisis estadístico de los datos.

Adicionalmente, se realizó una encuesta de satisfacción con la docencia en ambos grupos. Por parte de la clase más técnica participaron 24 alumnos/as, mientras que de la menos técnica la cubrieron 47. En general, los resultados fueron muy positivos en todas las dimensiones evaluadas. En la Figura 5, se muestran las puntuaciones para la pregunta “Mi interés por este tema científico aumentó”. Cabe aclarar que en Alemania el sistema de puntuación es el inverso al nuestro (el 1 se corresponde con la máxima puntuación). Por otra parte, en la Figura 6, se puede apreciar que la mayoría de los/as alumnos/as consideran que con esta metodología el profesorado consigue crear una atmósfera estimulante y motivante.

Por último, en esta encuesta también se permitía la inclusión de comentarios en texto libre y algunos de ellos fueron: “Espectacular. Me ha gustado mucho este curso. Me gustó que investigadores internacionales estuvieran vinculados al curso”, “Superinteresante, siempre útil porque siempre recibes ayuda del profesor o de los asistentes, muy bien supervisado” o “Seminario muy interesante e instructivo”.

8. Conclusiones

En esta ponencia se ha descrito una novedosa experiencia docente colaborativa entre la Universidad de Regensburg y la Universidad de Santiago de Compostela. Se utiliza un enfoque en el que el alumnado de-

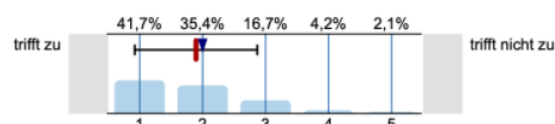


Figura 6: Puntuaciones para la pregunta “El profesorado consigue crear una atmósfera estimulante y motivante” (la máxima puntuación es 1)

sarrolla su propia comprensión de los contenidos en el contexto de retos relacionados con temas de investigación. El ámbito de estudio es la determinación de la credibilidad de contenidos online relacionados con el COVID-19 y sus tratamientos. La experiencia, aún en desarrollo, se valora como muy positiva, al mostrar el alumnado un alto grado de implicación y motivación.

El objetivo principal es exportar este tipo de metodologías a las enseñanzas universitarias de Ingeniería Informática en España, donde es mucho menos común ver una aproximación de este tipo. En el curso 2020/21 esto se ha puesto en marcha en Alemania como experiencia piloto pero pensamos importarlo, quizás en una materia de nivel de máster, ya que en la USC contamos con un Máster Interuniversitario en Tecnologías de Análisis de Datos Masivos: Big Data que puede ofrecer el encaje perfecto para este tipo de docencia.

Por otra parte, esta actividad forma parte de una colaboración internacional de investigación activa entre ambas entidades. En este sentido, se desea determinar qué preguntas de investigación son más valiosas para a posteriori ser utilizadas en un estudio de usuario mayor que pudiera tener impacto en los proyectos de investigación que estamos llevando a cabo. Con esto, se pretende poner en valor la relación directa entre docencia e investigación que muchas veces no es tan patente.

Agradecimientos

Este trabajo ha sido financiado por FEDER/Ministerio de Ciencia, Innovación y Universidades - Agencia Estatal de Investigación (RTI2018-093336-B-C21), y también ha recibido apoyo financiero de la Consellería de Educación, Universidade e Formación Profesional (acreditación 2019-2022 ED431G-2019/04, ED431C 2018/29, ED431C 2018/19) y de los fondos FEDER, que reconocen al CiTIUS de la USC como Centro de Investigación del Sistema Universitario de Galicia.

Referencias

- [1] Mustafa Abualsaud y Mark D Smucker. Exposure and order effects of misinformation on health search decisions. En *Proc. of the 42nd Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, Roma, 2019.

- [2] Mercedes Marqués Andrés, Vicente Ramón Tomás, y Ismael Sanz Blasco. Tratando de fomentar la motivación del estudiante. En *X Jornadas de Enseñanza Universitaria de la Informática (JENUI)*, páginas 329–336. Thomson-Paraninfo, 2004.
- [3] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press Nueva York, 1999.
- [4] Fidel Cacheda Seijo, Juan Manuel Fernández Luna, y Juan Francisco Huete Guadix. *Recuperación de información: un enfoque práctico y multidisciplinar*. Ra-Ma, 2011.
- [5] W Bruce Croft, Donald Metzler, y Trevor Strohman. *Search engines: Information retrieval in practice*, volume 520. Addison-Wesley Reading, 2010.
- [6] Gunther Eysenbach. Infodemiology: The epidemiology of (mis) information. *The American journal of medicine*, 113(9):763–765, 2002.
- [7] Brian J Fogg. Prominence-interpretation theory: Explaining how people assess credibility online. En *CHI'03 extended abstracts on human factors in computing systems*, páginas 722–723, 2003.
- [8] Susannah Fox. *Health topics: 80 % of internet users look for health information online*. Pew Internet & American Life Project, 2011.
- [9] Mariana-Daniela González-Zamar, Emilio Abad-Segura, et al. Evidencias del aprendizaje creativo en la educación superior. *Reencuentro. Análisis de problemas universitarios*, 31(77):111–132, 2019.
- [10] Carolin Hahnel, Frank Goldhammer, Ulf Kröhne, y Johannes Naumann. The role of reading skills in the evaluation of online information gathered from search engine environments. *Computers in Human Behavior*, 78:223–234, 2018.
- [11] Md Saiful Islam, Tonmoy Sarkar, Sazzad Hossain Khan, Abu-Hena Mostofa Kamal, SM Murshid Hasan, Alamgir Kabir, Dalia Yeasmin, Mohammad Ariful Islam, Kamal Ibne Amin Chowdhury, Kazi Selim Anwar, et al. Covid-19-related infodemic and its impact on public health: A global social media analysis. *The American Journal of Tropical Medicine and Hygiene*, 103(4):1621–1629, 2020.
- [12] Michal Kakol, Radosław Nielek, y Adam Wierzbicki. Understanding and predicting web content credibility using the content credibility corpus. *Information Processing & Management*, 53(5):1043–1061, 2017.
- [13] Gordon Pennycook, Jonathon McPhetres, Yunhao Zhang, Jackson G Lu, y David G Rand. Fighting covid-19 misinformation on social media: Experimental evidence for a scalable accuracy-nudge intervention. *Psychological science*, 31(7):770–780, 2020.
- [14] Frances A Pogacar, Amira Ghenai, Mark D Smucker, y Charles LA Clarke. The positive and negative influence of search results on people's decisions about the efficacy of medical treatments. En *Proc. of the ACM SIGIR Int. Conf. on Theory of Information Retrieval*, páginas 209–216, 2017.
- [15] Kashyap Papat, Subhabrata Mukherjee, Jan-nik Strötgen, y Gerhard Weikum. Where the truth lies: Explaining the credibility of emerging claims on the web and social media. En *Proc. of the 26th Int. Conf. on World Wide Web Companion*, páginas 1003–1012, 2017.
- [16] Reuters Insitute, University of Oxford. *Reuters Digital News Report 2020*, 2020 (accedida 16 de Noviembre, 2020).
- [17] Soo Young Rieh. Judgment of information quality and cognitive authority in the web. *Journal of the American society for information science and technology*, 53(2):145–161, 2002.
- [18] Silvia Rueda, J Ignacio Panach, Miriam Gil, Sergio Casas, y Mariano Pérez. Cuando la clase invertida no es suficiente: estrategias para motivar al alumnado en ingeniería del software. *Actas de las Jenui*, 5:125–132, 2020.
- [19] Ian Ruthven, David Elsweller, y Emma Nicol. Designing for users: an holistic approach to teaching information retrieval. En *2nd Int. Workshop on Teaching and Learning of Information Retrieval (TLIR)*, páginas 1–4, 2008.
- [20] Julia Schwarz y Meredith Morris. Augmenting web pages and search results to support credibility assessment. En *Proceedings of the SIGCHI conference on human factors in computing systems*, páginas 1245–1254, 2011.
- [21] Karishma Sharma, Feng Qian, He Jiang, Natali Ruchansky, Ming Zhang, y Yan Liu. Combating fake news: A survey on identification and mitigation techniques. *ACM Trans. on Intelligent Systems and Technology (TIST)*, 10(3):1–42, 2019.
- [22] Parikshit Sondhi, VG Vinod Vydiswaran, y ChengXiang Zhai. Reliability prediction of web-pages in the medical domain. En *European conference on information retrieval*, páginas 219–231. Springer, 2012.
- [23] Ernst Von Glasersfeld. Cognition, construction of knowledge, and teaching. En *Constructivism in science education*, páginas 11–30. Springer, 1998.
- [24] Yusuke Yamamoto y Katsumi Tanaka. Enhancing credibility judgment of web search results. En *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, páginas 1235–1244, 2011.

Cómo conseguir el Euro-Inf sin morir en el intento

Gabriel López Millán

Departamento de Ingeniería de la Información y las Comunicaciones
Universidad de Murcia
gabilm@um.es

Resumen

Euro-Inf es un sello de calidad que la agencia europea EQANIE, a través de ANECA, otorga a títulos universitarios en informática en el EEES. Para obtenerlo hay que demostrar unos determinados criterios de calidad, relevancia, transparencia y movilidad.

Obtener el Euro-Inf es un proceso complejo que requiere gran cantidad de recursos. Se realiza junto con el programa ACREDITA de ANECA, y es necesario analizar 29 resultados de aprendizaje específicos de Euro-Inf, seleccionar las asignaturas que cubren estos resultados, elaborar un informe donde se justifiquen estas decisiones en base a evidencias donde todos los agentes involucrados en el título (estudiantes, PDI, PAS, empleadores, etc.) deben participar, y preparar la visita de un panel de expertos.

Este trabajo recoge la experiencia recogida por el primer centro en España que obtuvo el sello Euro-Inf, renovado recientemente. El objeto es ayudar a otros centros a comprender cómo funciona el proceso de certificación, y compartir una experiencia que puede ser de utilidad para solicitar acreditaciones de este tipo.

Se detalla el proceso seguido para seleccionar las asignaturas que trabajan los resultados de aprendizaje, así como el procedimiento para elaborar un informe consensuado entre todas las partes junto con las evidencias necesarias, cómo se han abordado los resultados de aprendizaje Euro-Inf y unas recomendaciones finales.

Abstract

Euro-Inf is a quality label granted by the european agency EQANIE, through the spanish ANECA agency, to university degrees in computer science in the EHEA. To earn it, it is necessary to demonstrate that certain requirements of quality, relevance, transparency and mobility are reached by degree's students.

A complex process requiring a large amount of resources is required in order to grant the Euro-Inf. It is necessary to analyze 29 learning outcomes, to select the subjects covering these results, to prepare a report

where these decisions are justified based on evidences where all the involved agents (students, professor, administrative stuff, employers, etc.) must participate, and to prepare the visit of an expert panel review.

This work collects the experience gathered by the first center in Spain to obtain the Euro-Inf label (recently renewed). The aim is to help other centers to understand how the certification process works, and to share an experience that can be useful to apply for accreditations of this type. It details the process followed to cover the learning outcomes with specific subjects, as well as the procedure to prepare a report agreed between all parties together with the required evidences, how the Euro-Inf learning outcomes have been addressed, and some final recommendations.

Palabras clave

Calidad docente, Sello de Calidad, EQUANI, ANECA, SIC, Euro-Inf

1. Introducción

Euro-Inf¹ es un sello de calidad que la agencia europea EQANIE otorga a títulos universitarios en informática en el Espacio Europeo de Educación Superior (EEES). Este sello se concede a las titulaciones que demuestran alcanzar unos criterios establecidos de calidad, relevancia, transparencia y movilidad, y se aplica tanto a títulos de grado como de máster.

En España, este sello se obtiene a través del programa SIC (Sello Internacional de Calidad), que permite obtener certificaciones europeas (Euro-Inf, EURACE, etc.)² junto con el programa ACREDITA [1]. En España hay unas 30 titulaciones con el sello Euro-Inf (incluyendo títulos oficiales de Grado en Ingeniería Informática, y otros títulos relacionados como grados en Ingeniería del Software o en Ingeniería Multimedia), además de unos 14 títulos de máster [2].

¹<https://eqanie.eu/quality-label/>

²<http://www.aneca.es/Programas-de-evaluacion/Evaluacion-de-titulos/SIC>

La obtención de un SIC es un proceso complejo en el que la cantidad de recursos, principalmente humanos, que un centro debe destinar es especialmente alta. Centrándonos solo en Euro-Inf, es necesario analizar los 29 resultados de aprendizaje establecidos, seleccionar las asignaturas que cubren estos resultados en coherencia con la memoria del título, consensuar estas asociaciones con el profesorado, elaborar un informe donde se justifican estas decisiones, en base a evidencias, en el cual deben participar todos los agentes involucrados en el título (estudiantes, PDI, egresados, PAS, empleados, etc.), y preparar la visita del panel de expertos, lo que implica organizar una decena de reuniones y preparar nuevas evidencias.

Este trabajo intenta plasmar la experiencia recogida la Facultad de Informática de la Universidad de Murcia, primero a nivel nacional en obtener el sello Euro-Inf, tanto para los títulos de grado y máster, y que ha renovado ambas certificaciones con éxito en los últimos cursos académicos. El objetivo es ayudar a otros centros a comprender cómo funciona el proceso de certificación, y compartir una experiencia que pueda ser de utilidad para aquellos centros que tengan pensado solicitar acreditaciones de este tipo.

El trabajo se estructura del siguiente modo: En la sección 2 se presenta la certificación Euro-Inf, y los resultados de aprendizaje que deben ser alcanzados; la sección 3 presenta una breve descripción del título de Grado en Ingeniería Informática objeto de la última renovación de Euro-Inf durante el curso 19/20; en la sección 4 se describe el procedimiento seguido durante el proceso de acreditación; en la sección 5 se detallan los criterios seguidos para cubrir los resultados de aprendizaje; la sección 6 presenta una recomendaciones finales; y la sección 7 presenta las conclusiones y propuestas de trabajo futuro.

2. Euro-Inf

Euro-Inf persigue los siguientes objetivos:

- Mejorar la calidad y transparencia de los títulos de educación en informática.
- Proporcionar información sobre los programas de estudios en informática acreditados con un determinado nivel en toda Europa.
- Facilitar el reconocimiento de titulaciones entre los países europeos.

El elemento clave de Euro-Inf son los resultados de aprendizaje que se establecen para las titulaciones de grado y máster. Según EQANIE, estos resultados reflejan las expectativas del mercado laboral europeo y las cualificaciones de entrada a este mercado laboral requeridas en el campo de la informática. En general,

Criterios ACREDITA		
1. Organización y Desarrollo	D1. Gestión	
2. Información y Transparencia		
3. Sistema de Garantía Interno de Calidad		
4. Personal Académico	D2.	
5. Personal de Apoyo, RRMM y Servicios	Recursos	
6. Resultados de aprendizaje (RA)	D3.	
7. Indicadores de resultados	Resultados	
Criterios Euro-Inf:		
8. RA SIC (Euro-Inf).		
8.1. RA SIC incluidos en plan de estudios		
8.2. RA SIC alcanzados por estudiantes		
9. Soporte Institucional del Título		

Figura 1: Criterios ACREDITA+Euro-Inf

estos resultados de aprendizaje³ abarcan desde el diseño de los programas de estudio y su desarrollo, hasta la gestión de la calidad, pasando por los recursos, la transparencia de las titulaciones, etc.

En España Euro-Inf es emitido por ANECA, y se solicita junto con el programa ACREDITA, que incluye el análisis de los criterios generales para todas las titulaciones (criterios 1 a 7, Figura 1), más los específicos Euro-Inf (criterios 8 y 9). También se puede solicitar como modalidad Independiente Específica (solo criterios 8 y 9) si no han transcurrido más de dos años desde la obtención de la acreditación del título⁴.

Como se puede ver, los aspectos más generales para toda titulación universitaria relacionados con la gestión de la titulación (Dimensión 1, D1), recursos humanos y materiales (D2) o resultados académicos (D3) se analizan en ACREDITA, y se deja el análisis de los aspectos más relacionados con la profesión en informática al SIC Euro-Inf.

En concreto, los resultados de aprendizaje establecidos por ANECA para obtener el Euro-Inf se muestran en la Tabla 1. Estos se dividen en 6 bloques: el bloque 1 (Conocimiento y comprensión) está relacionado con los conocimientos básicos de las materias relacionadas con la informática, desde la base matemática, hasta el contexto histórico y venidero; el bloque 2 (Análisis) se centra en la capacidad de análisis de los estudiantes, de analizar problemas, de búsqueda de soluciones, simulación, modelización, etc.; el bloque 3 (Diseño e implementación) está relacionado con el diseño e implementación de sistemas informáticos; los bloques 4 (Contexto económico, jurídico, social, ético y medioambiental) y 5 (Práctica de la informática) están relacionados con los aspectos más prácticos de la profesión, tanto el contexto económico o jurídico, como los aspectos relacionados con la gestión de riesgos, el cumplimiento de normativas y estándares, la seguridad, etc.; finalmente, el bloque 6 (Otras competencias

³<http://eqanie.eu/wp-content/uploads/2019/09/Euro-Inf-Framework-Standards-and-Accreditation-Criteria-V-2016-10-24.pdf>

⁴Este criterio se aplica a partir de 2021.

y habilidades profesionales) se centra en competencias adquiridas por estudiantes en cuanto a la capacidad de trabajo en grupo, organización, liderazgo, etc.

3. Breve descripción del título

3.1. Antecedentes

En 2009 se implanta en la Facultad de Informática de la Universidad de Murcia el nuevo título de Grado en Ingeniería Informática (GII), con varias modificaciones menores en los cursos siguientes.

Durante el curso 2012/2013 el título obtiene el peor resultado de Tasa de Rendimiento⁵ desde su implantación, con un 41,1 %. Esto lleva a plantear soluciones para intentar paliar esta situación y, entre otras, se plantea la realización de auditorías externas que ayuden a identificar los problemas del título.

En 2015 el título obtiene la certificación ACREDITA y, unos meses antes, en 2014, la acreditación Euro-Inf. Este sello se obtiene tanto para el título de Grado en Ingeniería Informática, como para el máster oficial relacionado, siendo, como ya se ha comentado, la primera a nivel nacional.

En 2017 el centro obtiene la certificación AUDIT [4] al sistema de calidad. Y, finalmente, en 2018, renueva Euro-Inf para su título de máster⁶, y obtiene además la Acreditación Institucional.

Durante todos estos años han ocurrido dos cosas importantes: en primer lugar, el concepto de calidad se impregna en el centro, consiguiendo una participación cada vez mayor del profesorado; en segundo lugar, las Tasas de Rendimiento (TR) y Éxito (TE) han sufrido una mejora continua. En el caso de la TR, el curso 2018/2019 se sitúa en el 64,27 %, alcanzando la media nacional (66,2 %). En el caso de la TE, se sitúa en el 82,7 %, por encima de la media nacional (79,10 %). Se plantean aquí los hechos, no es objetivo de este trabajo discutir si la implantación de este tipo de acreditaciones conlleva una mejora en las tasas, ni siquiera, el discutir si una tasa de rendimiento o éxito elevada implica una mejora en la calidad de los estudios.

3.2. Estructura del título

El GII es un título de 240 ECTS, que se reparten en 4 cursos académicos⁷. Todas las asignaturas son de 6 ECTS, excepto el TFG (12 ECTS).

Los módulos de Formación Básica, Común a la Rama de Informática y Complementos de Formación se

cursan durante los tres primeros cursos. En tercero, el estudiante puede elegir entre dos asignaturas optativas, una de ellas relacionada con la realización de prácticas externas curriculares. En cuarto curso el estudiante debe realizar el TFG, que compaginará con la realización de una de las 5 menciones que oferta la titulación. Cada mención oferta 36 créditos obligatorios, más 12 optativos a elegir entre 24. Las 5 menciones ofertadas son las siguientes:

- Computación.
- Ingeniería de Computadores.
- Ingeniería del Software.
- Tecnologías de la Información.
- Sistemas de Información.

Con estos antecedentes, en el curso 2019/2020 se procede a la renovación del Euro-Inf del título. La experiencia acumulada en la renovación de este sello para los títulos de máster (2018) y grado (2020) se describe en las siguientes secciones. Como el proceso seguido es el mismo en ambos casos, nos centraremos en el más reciente, el título de Grado en Ingeniería Informática.

4. Procedimiento seguido

Aunque el título de GII fue analizado durante el proceso en 2014, y la estructura del título no ha variado desde entonces, es necesario volver a analizar qué asignaturas cubren los resultados de aprendizaje mostrados en la Tabla 1 por varios motivos:

- El conjunto de resultados requeridos por el proceso en 2020 sufre algunas modificaciones respecto a los establecidos en 2014.
- Las asignaturas van evolucionando a lo largo del tiempo, y hay contenidos que han sido ampliados, reducidos, movidos entre asignaturas, etc.
- El criterio para seleccionar asignaturas cambia respecto al anterior proceso. Deja de ser necesario indicar un porcentaje específico para cada asignatura y resultado, indicando ahora los contenidos específicos de la asignatura dedicados a tal fin.

La Figura 2 muestra un esquema del procedimiento seguido durante el todo el proceso.

4.1. De resultados de aprendizaje a asignaturas

La primera fase, Figura 2(1), es comprobar si los nuevos resultados siguen estando cubiertos por las asignaturas del título.

En primer lugar se realiza un análisis de los resultados establecidos por Euro-Inf, y su relación con las competencias especificadas en el título (1.1). Este análisis inicial se elabora desde el equipo decanal. Por

⁵<http://www.ciencia.gob.es/stfls/MICINN/Universidades/Ficheros/Estadisticas/metodologia-estadistica-indicadores-rendimiento-2019.pdf>

⁶Renovación cada 4 años para máster, 6 para grados.

⁷<https://www.um.es/web/estudios/grados/informatica/documentacion>

RA	Descripción		Comp.
Conocimiento	1.1	Conocimiento de las matemáticas y otras ciencias básicas inherentes a su especialidad de ingeniería, en un nivel que permita adquirir el resto de las competencias del título.	FB1, FB3
	1.2	Describir características de últimos avances en hardware/software y aplicaciones prácticas.	CEII4
	1.3	Describir avances infor. actuales/históricos y demostrar visión sobre tendencias/avances.	
	1.4	Aplicar conocimientos de otras disciplinas infor. al estudio de la propia área de especialidad.	
	1.5	Demostrar amplios conocimientos para crear aplicaciones infor. en otras áreas temáticas.	CR2
Análisis	2.1	Utilizar técnicas con las que identificar las necesidades de problemas reales, analizar complejidad y evaluar viabilidad de las posibles soluciones mediante técnicas informáticas.	FB3, CR6
	2.2	Describir un determinado problema y su solución a varios niveles de abstracción.	
	2.3	Seleccionar/utilizar métodos analíticos, de simulación y de modelización.	
	2.4	Escoger los patrones de solución, algoritmos y estructuras de datos apropiados.	CR6, CR7
	2.5	Analizar la medida en la que un determinado sistema informático cumple con los criterios definidos para su uso actual y desarrollo futuro.	CR1, CEII10
Diseño e implementación	3.1	Definir y diseñar hardware/software informático/de red que cumpla con los requisitos establecidos.	CR1, CR9
	3.2	Describir fases implicadas en modelos de ciclo de vida con respecto a definición/construcción/análisis/puesta en marcha de nuevos sistemas y mantenimiento.	CR2, CR16
	3.3	Elegir/utilizar modelos de proceso adecuados, entornos de programación y técnicas de gestión de datos impliquen aplicaciones tradicionales y emergentes.	CEII8
	3.4	Describir/explicar el diseño de sistemas/interfaces para interacción persona-ordenador y ord-ord.	CR11, CR17
	3.5	Aplicar competencias prácticas/de programación en la creación de programas informáticos y/u otros dispositivos informáticos.	CR8, CEII6
Contexto económico, etc.	4.1	Demostrar concienciación sobre la necesidad de tener una conducta ética y profesional de primer nivel y conocimientos de los códigos de conducta profesionales.	CR2, CEII12, CB2
	4.2	Explicar cómo contexto comercial/industrial/económico/social afecta la práctica de la informática.	CR2, CEII12
	4.3	Identificar los requisitos jurídicos que rigen las actividades informáticas, incluyendo la protección de datos, derechos de propiedad intelectual, contratos, cuestiones de seguridad del producto y responsabilidad, cuestiones personales y riesgos laborales.	FB6, CEII7, CR18
	4.4	Explicar importancia de la confidencialidad de la información y cuestiones de seguridad con respecto al diseño, desarrollo, mantenimiento, supervisión y uso de sistemas informáticos.	CR1
Práctica de la inf.	5.1	Demostrar conocimientos sobre los códigos y estándares de cumplimiento del sector.	
	5.2	Describir las técnicas de gestión de diseño/ implementación/análisis/uso/mantenimiento de sistemas informáticos, incluyendo gestión de proyectos/configuración/de cambios.	CR2, CR4
	5.3	Identificar riesgos de seguridad/laborales/ medioambientales/comerciales y llevar a cabo evaluación de riesgos/reducción de riesgos/técnicas de gestión de riesgos.	CEII11
	5.4	Realizar investigaciones bibliográficas y evaluaciones utilizando BBDD y otras fuentes.	
	5.5	Realizar investigaciones prácticas para interpretar datos/extraer conclusiones.	
Otras comp.	6.1	Organizar su trabajo de manera independiente demostrando iniciativa y responsabilidad.	CEII9
	6.2	Comunicar mensajes de forma efectiva oralmente y por otros medios.	CEII9
	6.3	Planificar su propio proceso de aprendizaje autodidacta y mejorar su rendimiento personal como base de una formación y un desarrollo personal continuos.	CB5
	6.4	Identificar maneras de organizar equipos y los distintos roles dentro de dichos equipos.	
	6.5	Participar de manera efectiva en grupos de trabajo informático.	

Cuadro 1: Resumen de resultados de aprendizaje Euro-Inf (<http://www.aneca.es>), y relación con Competencias.

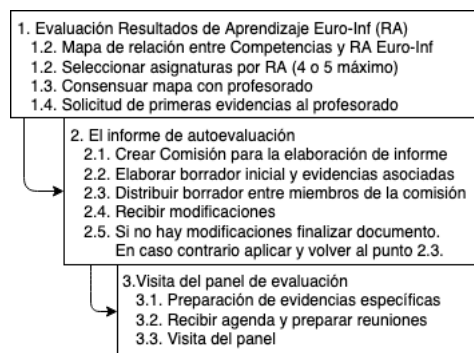


Figura 2: Procedimiento seguido

ejemplo: el resultado de aprendizaje 1.1, estaría relacionado con la Competencia Específica del módulo Formación Básica: "FB1. Capacidad para la resolución de los problemas matemáticos que puedan plantearse en la ingeniería. Aptitud para aplicar los conocimientos sobre: álgebra lineal; cálculo diferencial e integral; métodos numéricos; algorítmica numérica; estadística y optimización".

En este paso se tuvieron en cuenta las competencias establecidas en el Real Decreto 1393/2007, por el Consejo de Universidades, y competencias específicas del título. Esta relación se analiza en la sección 5.

En algunos casos esta asociación no fue posible, y se analizaron los propios resultados de aprendizaje establecidos por las asignaturas en el título.

El siguiente paso es elegir las asignaturas más representativas (4 o 5 máximo) que cubren esas competencias (1.2). La idea inicial es utilizar solamente asignaturas obligatorias, sin incluir optativas ni asignaturas de mención. Esto aseguraría que todo estudiante, independientemente de la mención que elija cursar adquiriría estos resultados. En los casos en los que no fue posible se optó por asegurar que al menos una asignatura obligatoria de cada mención cubría el resultado.

Una mención especial requiere la asignatura Trabajo Fin de Grado (TFG). Esta asignatura cubre bastantes de los resultados de aprendizaje Euro-Inf, sin embargo, siguiendo las recomendaciones de ANECA, se limita su aparición solo a aquellos resultados en los que es clave para que sea cubierto.

Esta primera aproximación se distribuye entre los profesores coordinadores de las asignaturas seleccionadas del título, con el fin de recoger opiniones y sugerencias. Finalmente, se envía a todo el profesorado del centro con el objetivo de que todos estén informados y puedan participar del proceso (1.3).

Durante esta revisión también se pide a los coordinadores información adicional que habrá que incorporar como evidencias del proceso (1.4). En concreto se pide una breve descripción de:

- Los contenidos de cada asignatura donde se trabaja el resultado de aprendizaje seleccionado.
- Las actividades formativas con las que se cubre (clase magistral, laboratorio, seminario, etc.).
- Cómo se evalúa la adquisición de dicho resultado (trabajo teórico, examen, exposición, etc.).

Además, para los resultados de los bloques 3 (Diseño e Implementación) y 5 (Práctica de la informática) se pide un listado específico de contenido teórico/práctico para cada asignatura seleccionada.

Es importante destacar que cuando un resultado de aprendizaje es alcanzado mediante un trabajo en grupo, hay que indicar cómo se evalúa la adquisición por cada estudiante de modo individual. Por ejemplo, para varias asignaturas en este caso se indicaba que la evaluación consistía en una entrevista de prácticas con preguntas dirigidas a cada miembro del grupo.

La carga de trabajo que supone este proceso se centra principalmente en la elaboración del mapa inicial de relación de asignaturas y resultados Euro-Inf, y en las múltiples reuniones con profesores, y más de un centenar de correos electrónicos con peticiones, dudas, sugerencias, etc. Por otro lado, este esfuerzo tiene dos aspectos muy positivos: la información ha sido validada por todos los profesores; y éstos se han involucrado positivamente en el proceso.

4.2. El Informe de Autoevaluación

Al ser un proceso conjunto de renovación de ACREDITA y Euro-inf, el centro debe presentar un informe de autoevaluación sobre los criterios establecidos por ACREDITA, y otro informe para los criterios Euro-Inf (criterios 8 y 9). Denominaremos simplemente Informe de Autoevaluación (IA) a ambos documentos y nos centraremos en el análisis del criterio 8 (sección 2).

El informe debe ser elaborado por una comisión (2.1) en la que deben participar los colectivos implicados en el título: equipo de gobierno de la universidad y/o del centro(s), estudiantes, PDI, coordinadores del título, PAS, egresados y empleadores.

Dado que la Facultad de Informática ya tiene una Comisión de Aseguramiento de la Calidad (CAC) donde se incluyen la mayoría de los agentes relacionados con la titulación, la comisión para la elaboración del IA estará formada por uno o varios miembros de cada uno de los agentes que forman parte de la CAC (3 miembros del equipo decanal, coordinador de la titulación, 3 PDI, 2 estudiantes y 2 PAS), más uno o varios representantes del resto de agentes relacionados, es decir, empleadores (3 miembros), egresados (4 miembros) y un representante de los colegios profesionales (en este caso, el Colegio Profesional de Ingenieros Técnicos y Graduados). En total, 18 miembros.

Para abordar la realización del IA (2.2) se decide

seguir el mismo procedimiento que se utilizó para la renovación del SIC para el título de máster. Es decir, que haya un trabajo previo desde el equipo decanal para recopilar todas las evidencias posibles y elaborar un borrador de cada una de las directrices a evaluar.

Una vez recopilado el grueso de las evidencias y el borrador de las directrices, se pasa esta información a todos los miembros de la comisión para la recogida de opiniones. Los resultados de esta evaluación serán abordados por el equipo decanal, que volverá a preparar un próximo borrador que será enviado de nuevo a la comisión. Este ciclo se repetirá hasta que los informes estén consensuados por todas las partes (2.3-2.5).

Durante este proceso, para cada criterio se hace hincapié en dos aspectos: las evidencias que apoyan los informes, tanto las obligatorias como las opcionales, y las encuestas de satisfacción. Éstas últimas son clave para justificar la consecución de los criterios, no solo las tradicionales encuestas a estudiantes, sino también las encuestas al PDI, PAS, egresados y empleadores.

Como se puede deducir, la mayor carga de trabajo en este proceso recae en el equipo decanal, encargado de elaborar los borradores iniciales. Es muy importante destacar la ayuda recibida aquí por la Unidad para la Calidad de la Universidad, una unidad destinada a ayudar a los centros en la gestión de la calidad y que se encarga de proporcionar una parte muy importante de todas las evidencias necesarias (informes de programas de movilidad, prácticas externas, resultados académicos, perfiles de ingreso, etc.) [5]. En concreto, para este proceso, se presentaron unos 75 documentos entre evidencias obligatorias y opcionales.

4.3. La visita del panel de evaluación

Por último, la preparación de la visita del panel (3). Esta fase es importante por varios aspectos.

En primer lugar, la preparación de nuevas evidencias específicas (3.1) que el panel solicita tras el IA y con antelación a la visita. Normalmente, pruebas de evaluación de asignaturas de referencia que indica el panel. En este caso se seleccionaron cuatro asignaturas que, o bien tenían una baja valoración de los estudiantes (menos de 3 sobre 5), una tasa de rendimiento baja, o que aparecían en muchos resultados de aprendizaje. Además, también se incluye el TFG como asignatura de referencia. Para estas asignaturas se solicitaron pruebas de evaluación de estudiantes con evaluación de Sobresaliente, Notable y Aprobado.

Merece la pena recordar la importancia de guardar las pruebas de evaluación de cursos anteriores. Este hecho suele estar reglado por la normativa universitaria, pero la realidad es que muchas veces esta documentación no está cuando se necesita.

A continuación el centro recibe la agenda (3.2) con los horarios de las reuniones del panel con los diferen-

tes agentes. En este caso, el panel puede indicar aspectos concretos, por ejemplo, para la reunión con el profesorado puede requerir la asistencia de profesores asociados, ayudantes, que hayan dirigido TFGs, etc. Para la reunión con estudiantes, aquellos que hayan cursado determinadas asignaturas, participado en programas de movilidad, o en prácticas en empresa. Organizar las visitas también requiere un esfuerzo importante por parte del centro, sobre todo para contactar con entidades empleadoras y egresados. Es recomendable organizar estas visitas con suficiente antelación.

Finalmente, el último paso del proceso es la visita del panel de evaluación (3.3). Esta visita suele organizarse en un único día, y para este proceso se programaron 8 reuniones⁸ a lo largo de la jornada, con un total de más de 50 asistentes. En general, suelen seguir un formato similar a visitas de otros programas, como ACREDITA o AUDIT.

5. Resultados de aprendizaje

Esta sección recoge cómo se han cubierto los resultados de aprendizaje en relación a los contenidos de las asignaturas de la titulación, y dónde se han aplicado las competencias oficiales de la titulación: Específica de la Ingeniería Informática (CEII)⁹; Formación Básica (FB); y la Común a la Rama de Informática (CR)¹⁰. También se indican las Competencias Básicas (CB) asociadas según el Real Decreto 1393/2007¹¹. Esta asociación no es exhaustiva, la aplicación adicional de otras competencias de la titulación a los resultados Euro-Inf no se describe por ser específicas del título. Esta asociación también se recoge en la Tabla 1.

5.1. Conocimiento y comprensión

Para el resultado 1.1 (FB1, FB3) se seleccionaron asignaturas de primer y segundo curso sobre fundamentos de cálculo, álgebra y estadística, incluido análisis de algoritmos, y los fundamentos de lógica. En el caso del 1.2 (CEII4) se seleccionaron asignaturas sobre fundamentos de arquitecturas de computadores y sobre aplicación práctica de un desarrollo software.

El 1.3 se basa en mostrar una visión de la historia de la informática, así como tendencias futuras. Este aspecto suele tener un tratamiento pequeño en las asignaturas, a modo de introducción o cierre. Se eligieron aquí asignaturas donde se tratan estos aspectos centrados en redes de comunicaciones, arquitectura de computadores e inteligencia artificial, y se refuerza con al menos una asignatura de cada mención.

⁸ Virtuales por estado de alarma por la COVID-19.

⁹ BOE-A-2009-12977, Anexo II, Apartado 3.

¹⁰ BOE-A-2009-12977, Anexo II, Apartado 5.

¹¹ <https://www.boe.es/eli/es/rd/2007/10/29/1393>, Anexo I.

Para el resultado 1.4 nos centramos en las menciones, y seleccionamos asignaturas donde se traten aspectos relacionados con las otras menciones. Por ejemplo, en las menciones de Tecnologías de la Información, Computación, Sistemas de Información o Arquitectura de Computadores se seleccionan asignaturas donde se requieren conocimientos la mención de Ingeniería del Software. Para esta última se seleccionan asignaturas que requieren tanto conocimiento de Tecnologías de la Información, como de Computación.

Finalmente, para el 1.5 (CR2) se eligen dos asignaturas con proyectos de desarrollo multidisciplinarios y enfocados a casos reales. En ambos casos, toda la carga práctica se basa en el desarrollo de un proyecto, y se trabaja directamente con empresas colaboradoras.

5.2. Análisis

Para cubrir el resultado 2.1 (FB3, CR6) se seleccionan asignaturas de segundo y tercero sobre complejidad de algoritmos y técnicas de resolución de problemas. Para el 2.2, asignaturas donde se trata la abstracción de problemas en diferentes ámbitos de la informática, contenidos de estructuras de datos y grafos, programación, arquitectura de computadores o redes de comunicaciones. Se cubre con hasta seis asignaturas obligatorias de los tres primeros cursos.

El 2.3 se trabaja con asignaturas obligatorias que tienen carga de simulación y modelado, como física, simulación de redes o modelado de procesadores.

Para el 2.4 (CR6, CR7) se seleccionan las asignaturas obligatorias sobre fundamentos de algoritmos, estructuras de datos y patrones de diseño software. Finalmente, para el 2.5 (CR1, CEII10) tres obligatorias que trabajan la evaluación de software y hardware.

5.3. Diseño e implementación

Para el resultado 3.1 (CR1, CR9) se seleccionan asignaturas obligatorias dedicadas al desarrollo software, evaluación de requisitos de componentes hardware, o de estándares de redes de comunicaciones.

En el caso del 3.2 (CR2, CR16) se seleccionan asignaturas de tercer curso donde se trabaja con proyectos de desarrollo, se tiene en cuenta el ciclo de vida de los productos, y colaboran empresas.

En el 3.3 (CEII8) se eligen cuatro asignaturas de desarrollo software donde el estudiante debe tomar decisiones sobre qué herramientas de desarrollo utilizar en prácticas, así como elegir modelos de desarrollo.

Para el 3.4 (CR11, CR17) se eligen asignaturas donde aparece la programación de interfaces de usuario a nivel software, o comunicación entre equipo a nivel de red. Se cubre con asignaturas de tercer curso.

En el resultado 3.5 (CR8, CEII6) se combinan asignaturas relacionadas con programación de los primeros

cursos, con otras de cada una de las menciones donde se ven aspectos prácticos de desarrollo software.

5.4. Contexto económico, jurídico, social, ético y medioambiental

Los resultados 4.1, 4.2 (CEII12, CR2, CB2) y 4.3 (FB6, CEII7, CR18) se cubren con varias asignaturas clave: una asignatura de primer curso dedicada a gestión empresarial, con seminarios específicos sobre ética, legislación, etc. Esta asignatura abarca los tres resultados; una asignatura dedicada a destrezas profesionales de tercer curso, donde los estudiantes crean un proyecto de empresa, y donde participan en el asesoramiento empresas del sector; esta asignatura está estrechamente relacionada con otra de tercer curso, dedicada al análisis de requisitos, estimación y gestión de un proyecto informático. Esta relación con empresas es fundamental para cubrir estos resultados.

Finalmente, el 4.4 (CR1) queda cubierto por dos asignaturas (segundo y tercero), donde se tratan aspectos de seguridad informática. También se han indicado asignaturas de mención para reforzar este aspecto.

5.5. Práctica de la informática

Para cubrir el resultado 5.1 se seleccionan aquellas asignaturas donde se trabaja con estándares y normativas internacionales. Por ejemplo, asignaturas de arquitectura de computadores donde se trabaja con el estándar POSIX [6], de telemática donde se estudian estándares IETF¹², o sobre desarrollos software donde se ven normas y estándares para la gestión de requisitos.

Alguna de estas asignaturas también se usan para cubrir 5.2 (CR2, CR4) y 5.3 (CEII11), ya que abarcan el ciclo completo de desarrollo de un proyecto informático, incluyendo los aspectos sobre gestión de riesgos. Se trata aquí de nuevo de las dos asignaturas de tercer curso donde se trabaja con proyectos relacionados con empresas, y se acompaña de otras asignaturas donde, en menor medida, se plantean estos aspectos.

Para el resultado 5.4 no nos centramos en las referencias bibliográficas indicadas en las asignaturas, ya que se hace referencia a búsquedas de información necesarias para poder completar algún aspecto teórico o práctico de la asignatura. Aquí, entre otras, aparece por primera vez el Trabajo Fin Grado, donde la labor de documentación por parte del estudiante es fundamental.

Los aspectos más centrados en investigaciones prácticas, 5.5, se dejan para tres asignaturas de primer, segundo y tercer curso, donde expresamente se trabaja con análisis de rendimiento de computadores, sistemas de memoria, simulación de circuitos, etc.

¹²IETF Internet Engineering Task Force. <https://www.ietf.org/>

5.6. Otras competencias y habilidades profesionales

Este último bloque está relacionado con la organización del trabajo, comunicación y trabajo en grupo. Aquí vuelve a ser fundamental la aparición del TFG para cubrir los resultados 6.1(CEII9), 6.2(CEII9) y 6.3(CB5), ya que requiere un alto grado de autonomía por parte del estudiante.

También son claves aquellas asignaturas donde se realizan trabajos en grupo. En las asignaturas de tercero relacionadas con gestión de proyectos y proyectos con empresas los estudiantes trabajan en grupos de 6 o 7 miembros. Estas son clave para adquirir las competencias relacionadas con trabajo en equipo, liderazgo y organización: 6.2, 6.3, 6.4 y 6.5. Además, se indican otras asignaturas que en menor medida también realizan prácticas en grupo (normalmente de 2 estudiantes), o donde se realizan exposiciones de trabajos en clase.

6. Recomendaciones

Esta sección recoge una serie de recomendaciones finales que pueden ser de utilidad para abordar la certificación Euro-Inf.

1. Asegurar un amplio consenso sobre el mapa de asignaturas y resultados Euro-Inf entre todo el profesorado del centro. Seleccionar 4 o 5 asignaturas obligatorias como máximo por resultado, donde se justifique claramente cómo se cubre.
2. En el caso de que el título tenga menciones, asegurar que los resultados están cubiertos independientemente de la mención escogida por los estudiantes. En este caso el uso de menciones fue importante para los resultados 1.3 y 1.4.
3. Tener histórico de encuestas de estudiantes, PDI, PAS, egresados y empleadores.
4. La participación de empresas en asignaturas es clave, ya sea mediante la impartición de seminarios o la colaboración en proyectos dentro éstas.
5. Los proyectos en grupo (más de 2) son importantes para cubrir los resultados del bloque 6.
6. La realización de prácticas externas no es obligatoria en esta titulación. Los resultados Euro-Inf relacionados con la práctica de la informática (5.4 y 5.5) se cubren principalmente con asignaturas donde participan de modo activo las empresas.
7. Para la elaboración del IA, la realización de borradores previos desde el centro y la revisión por parte de las entidades externas al centro (empleadores, egresados, etc.) ha funcionado bien en los procesos seguidos hasta ahora.
8. El TFG es muy importante para cubrir el bloque 6, ya que muchas veces es complicado que se trate en asignaturas troncales.

9. Guardar evidencias de pruebas de evaluación de todos los estudiantes para cada asignatura de al menos dos cursos anteriores.
10. Cubrir un resultado de aprendizaje va a depender de cómo de en profundidad se trabajen los contenidos relacionados en las asignaturas, y las evidencias disponibles. Sobre todo para los bloques 3 y 5, para los cuales se pide enumerar estas evidencias de modo expreso.

7. Conclusiones

Este artículo hace un recorrido por el trabajo realizado para alcanzar la certificación SIC Euro-Inf. Desde el proceso de elaboración del autoinforme, hasta la preparación de la vista del panel de expertos, centrandose la última parte en presentar aquellos criterios que se han tenido en cuenta a la hora de elegir las asignaturas que cubren los resultados de aprendizaje, así como unas recomendaciones finales.

Como trabajo futuro se plantea el estudio de la carga de trabajo para el personal docente asociado a las tareas de calidad del centro (actualmente excesivamente alto), o el estudio de la situación de las titulaciones en informática a nivel nacional y su adecuación a los criterios establecidos por Euro-Inf.

Referencias

- [1] ACREDITA. Evaluación para la renovación de la acreditación de títulos oficiales de Grado y Máster Universitario. Programa ACREDITA. Documento Marco. Versión 6. 08/01/2019.
- [2] ANECA, Listado de títulos con el sello Euro-Inf <http://www.aneca.es/Programas-de-evaluacion/Evaluacion-de-titulos/SIC/Programas-formativos-con-Sello> (21/3/2021).
- [3] ANECA. *Guía de Autoevaluación: renovación de la acreditación de títulos oficiales de Grado y Máster Universitario*. Programa ACREDITA. Versión 6. 08/01/2019.
- [4] Programa AUDIT. *Guía del Modelo de Certificación de la implantación de Sistemas de Aseguramiento Interno de Calidad*. v05. 2020.
- [5] Gabriel López Millán. *Experiencia de gestión de la calidad docente en el Grado en Ingeniería Informática*. JENUI 2020. Valencia.
- [6] IEEE Standard for Information Technology. *Portable Operating System Interface (POSIX(TM)) Base Specifications*, Issue 7, in IEEE Std 1003.1-2017 (Revision of IEEE Std 1003.1-2008) , vol., no., pp.1-3951, 31 Jan. 2018, doi: 10.1109/IEEESTD.2018.8277153.

Efecto de la evaluación docente por COVID-19. Influencia en los resultados académicos en titulaciones y materias con diferente número de alumnos

Noelia Sánchez-Marño, Beatriz Pérez-Sánchez
Ciencias de la Computación y Tecnologías de la Información
Universidad de A Coruña
Facultad de Informática, Campus Elviña s/n 15071 A Coruña
noelia.sanchez@udc.es, beatriz.perezs@udc.es

Resumen

El estado de alarma causado por el COVID-19 ha propiciado una situación sin precedentes en el ámbito universitario que ha afectado a todos sus estamentos. Desde el punto de vista docente, la suspensión de las actividades presenciales y el cambio inmediato a la modalidad virtual provocó una adaptación apresurada de los materiales docentes para ajustarse al nuevo escenario, asimismo fue necesario replantear el sistema de evaluación. La disparidad en el número de estudiantes matriculados en las distintas titulaciones universitarias condiciona el diseño y aplicación de metodologías y su correspondiente proceso de evaluación. En este trabajo se exponen los resultados académicos de dos titulaciones del área tecnológica con diferencias importantes en el número de alumnos matriculados. Estos resultados muestran que ambas titulaciones experimentaron variaciones notables en el número de aprobados comparado con cursos anteriores. Adicionalmente, se analiza una materia de cada titulación, ambas relacionadas con el aprendizaje de programación, para ver qué estrategias ha aplicado el profesorado en tal situación sobrevenida y qué diferencias se observaron en los resultados académicos.

Abstract

The state of alarm caused by COVID-19 has led to an unprecedented situation in the university environment that has affected all levels. From the teaching point of view, the suspension of face-to-face activities and the immediate change to the virtual modality caused a hasty adaptation of the teaching materials to adjust to the new scenario; it was also necessary to rethink the evaluation system. The disparity in the number of students enrolled in the different university degrees conditions the design and application of methodologies and

their corresponding evaluation process. In this work, the academic results of two degrees in the technological -with important differences in the number of students enrolled- area are exposed. These results show that both degrees experienced notable variations in the number of passed compared to previous courses. A subject of each degree related to programming learning is analyzed to see what strategies the teaching staff have applied in such a situation and what differences were observed in the academic results.

Palabras clave

COVID-19, tamaño grupo alumnos, evaluación docente, adaptación de metodologías.

1. Introducción

La pandemia originada por la COVID-19 ha propiciado una situación sin precedentes en el ámbito universitario tanto en España como en otros países que ha afectado a toda la comunidad universitaria [6]. Los distintos países cambiaron sus enfoques de enseñanza-aprendizaje, centrándose muchos de ellos en el aprendizaje en línea [5]. En España, el Ministerio de Universidades¹ presentó una serie de medidas para la adaptación del sistema universitario español ante los múltiples impactos de la crisis sobre las actividades docentes. En consecuencia toda la actividad (clases, evaluaciones y prácticas) se transformó a un formato no presencial y se facilitó la adaptación de las guías docentes,

¹Recomendaciones sobre criterios generales para la adaptación del sistema universitario español ante la pandemia del Covid-19, durante el curso 2019-2020. Disponible en https://www.ciencia.gob.es/stfls/MICINN/Universidades/Ficheros/Recomendaciones_adaptacion_universidades_CGPU.pdf, mayo de 2021.

sin que ello supusiera un cambio y revalidación de los planes de estudio.

Entre las recomendaciones generales cabe destacar, una ponderación mayor para las actividades de evaluación continua y la selección de procedimientos de evaluación adecuados para cada actividad formativa. El motivo principal es la ausencia de una solución universal para la evaluación no presencial², aspecto crucial en el ámbito de la educación superior [1].

Este cambio en el paradigma enseñanza-aprendizaje ha constituido un desafío para alumnado y profesorado. Por un lado, han tenido que sortear la insuficiente infraestructura técnica facilitada por sus instituciones. Por otro lado, han tenido que adaptarse a las herramientas necesarias, así como a los *nuevos* modelos de enseñanza-aprendizaje [7]. Es importante tener presente que esta docencia no presencial de emergencia no se puede considerar análoga a las propuestas que están específicamente diseñadas desde su concepción para impartirse en línea [2]. Así, han surgido muy diversas propuestas en todos los países del mundo, una perspectiva amplia de la situación, principalmente, europea, y, orientada a la evaluación en línea se expone en [4], mientras que el trabajo de Fardoun et al. [1] presenta las estrategias utilizadas en los procesos de enseñanza aprendizaje en los distintos niveles educativos iberoamericanos.

En este trabajo se analizan los cambios efectuados a las guías docentes, tanto en la selección de metodologías como en su posterior evaluación en dos titulaciones con un número de alumnos muy diferente, concretamente el grado en Ingeniería Electrónica Industrial y Automática (IEIA) y el grado en Ingeniería Informática (II) en la Universidad de A Coruña (UDC) (la primera tiene aproximadamente 200 alumnos y la segunda supera 1200). Los resultados académicos muestran que ambas titulaciones experimentaron variaciones notables en el número de aprobados comparado con cursos anteriores, exhibiendo un mayor incremento el grado con menor número de alumnos. Tras este análisis global, nos centraremos en una materia de cada grado, concretamente, *Informática y Programación II*. Además de introducir modificaciones en el sistema de evaluación, en la primera se optó por una docencia síncrona, mientras que la segunda fue asíncrona. Analizaremos el impacto de estos cambios en los resultados académicos, demostrando que la materia con menores modificaciones experimentó un menor aumento en el número de aprobados.

²CRUE. Informe sobre Procedimientos de Evaluación no Presencial. Estudio del Impacto de su Implantación en las Universidades Españolas y Recomendaciones. Disponible en <https://tic.crue.org/wp-content/uploads/2020/05/Informe-procedimientos-evaluación-no-presencial-CRUE-16-04-2020.pdf>, mayo de 2021

	Inf	ProII
Alumnos matriculados	50	392
Grupos de teoría	1	(4+1)
Grupos de prácticas	4	(19+1)
Horas docencia expositiva	21	150
Horas docencia práctica/TGR	240/0	400/200

Cuadro 1: Datos básicos de ambas materias. En la asignatura de Programación II hay docencia en inglés, por ello se indican los grupos impartidos en español + los grupos de docencia en inglés. TGR: Tutorías grupos reducidos

2. Contexto en la UDC

El grado en IEIA, que habilita para la profesión regulada de Ingeniero Técnico Industrial, se cursa en la Escuela Universitaria Politécnica del campus de Serantes sito en Ferrol. Junto con la formación básica en materias fundamentales (Física, Álgebra, Informática, etc.) el alumnado recibe formación en materias como Mecánica de fluidos, Teoría de máquinas, Automatización, Ingeniería de control, Informática industrial, etc. El número total de alumnos de este grado en el curso 2019/20 fue 193. Por su parte, el título de Graduado/a en Ingeniería Informática (II) se cursa en la Facultad de Informática del campus de Elviña (A Coruña) e integra todas las especialidades existentes en España ya que el estudiante podrá elegir, a partir de 3^{er} curso, entre distintas especialidades. Esta titulación supera los 1200 alumnos matriculados.

La asignatura *Informática* es parte de la formación básica de primer curso de IEIA, se imparte en segundo cuatrimestre y, por tanto, fue la afectada por el estado de alarma. Como mencionamos en el párrafo anterior, el grado en IEIA cuenta solo con 193 alumnos y, de ellos, 50 se encuentran matriculados en esta asignatura. Por otro lado, la asignatura *Programación II* se imparte en segundo cuatrimestre del Grado en Ingeniería Informática, el número de alumnos matriculados en la materia se aproxima a 400. El Cuadro 1 muestra los datos básicos de estas materias. Los contenidos prácticos de ambas materias se enfocan en aprender a programar usando el lenguaje C, si bien en *Informática* se imparten conceptos básicos de programación (estructuras de control, funciones, ficheros, etc.) mientras que en *Programación II* los alumnos ya tienen cierta experiencia programando (han cursado *Programación I*) y se centra en estructuras de datos (listas, pilas, colas, etc.).

Primer Curso					Segundo Curso				
Materia	GuíaDocente		Addenda		Materia	GuíaDocente		Addenda	
	Metd.	Ev.	Metd.	Ev.		Metd.	Ev.	Metd.	Ev.
Informática	Prueba	60	Prueba	60	Técnicas de Fabricación	Prueba	70	Prueba	65
	TTut	10	TTut	10		PLab	10	EvalCont/	
	PLab	6	PLab	6		SMag	10	PLab/	35
	SProb	24	SProb	24		SProb	10	Taller	
Álgebra	Prueba	55	Prueba	40	Mecánica de Fluidos	Prueba	70	Prueba	70
	PPrac	30	SProb	60		Plab	10	PLab	20
	PLab	15	-	-		TTut	20	TTut	10
Física II	Prueba	70	Prueba	40	Fundamentos de Automática	Prueba	70	Prueba	40
	PLab	15	PLab	30		SProb	20	SProb	60
	SProb	15	EvalCont	30		PLab	10	-	
Estadística	Prueba	45	Prueba	50	Resistencia de Materiales	Prueba	70	Prueba	60
	Test	25	Test	20		SProb	30	SProb	40
	PTIC	30	EvalCont	30					
Ciencia de Materiales	Prueba	70	Prueba	70	Fundamentos de Electrónica	Prueba	70	Prueba	70
	PLab	15	PLab	15		TTut	20	TTut	30
	Taller	15	Taller	15		PLab	10	-	

Cuadro 2: Metodología (Metd.) y porcentaje de la evaluación (Ev.) asignado para cada una de las materias de segundo cuatrimestre de primer y segundo curso de IEIA en la guía docente del curso 2019/20 y su posterior addenda. *Notación:* EvalCont: Evaluación Continua, PLab:prácticas laboratorio, PPrac: Prueba Práctica, PTIC: Prácticas a través de TIC, SProb: solución de problemas, TTut:trabajos tutelados.

Primer Curso					Segundo Curso				
Materia	GuíaDocente		Addenda		Materia	GuíaDocente		Addenda	
	Metd.	Ev.	Metd.	Ev.		Metd.	Ev.	Metd.	Ev.
Programación II	Prueba	80	Prueba	20	Bases de Datos	Prueba	75	Prueba	40
	PLab	20	PLab	80		PLab	10	TTut	60
	SProb	10	-			SolProb	15	-	
Álgebra	Prueba	80	Prueba	40	Concurrencia y Paralelismo	Prueba	70	Prueba	40
	PLab	20	PLab	30		PLab	30	PLab	40
	-		PTIC	30		-		PPrac	20
Estadística	Prueba	60	Prueba	40-60	Proceso Software	Prueba	60	Prueba	60
	PLab	20	PLab	40		PLab	40	TGTut	40
	Seminario	20	Seminario	0-20					
Fundamentos de Computadores	Prueba	60	Prueba	50	Redes	Prueba	70	Prueba	30
	PLab	20	PLab	30		PLab	25	PLab	60
	SProb	20	Test	10		Seminar	5	Seminario	10
	-		TAuton	10					
Tecnología Electrónica	Prueba	60	PLab	20	Sistemas Inteligentes	SMag	60	SMag	60
	PLab	20	SProb	80		PLab	30	PLab	30
	SProb	20	-			TTut	10	TTut	10

Cuadro 3: Metodología (Metd.) y porcentaje de la evaluación (Ev.) asignado para cada una de las materias de segundo cuatrimestre de primer y segundo curso de II en la guía docente del curso 2019/20 y su posterior addenda. *Notación:* SMag: Sesión magistral, PLab: prácticas laboratorio, PTIC: Prácticas a través de TIC, TAuton: Trabajo autónomo, TTut: Trabajo tutelado, SProb: solución de problemas.

3. Adaptación de las guías docentes

Ante la necesidad de transformar la actividad docente de un formato presencial a uno no presencial, el Ministerio de Universidades¹ sugirió una breve adaptación en las guías docentes -que podría ser un documento incluido en el espacio virtual de aprendizaje- que debería informar de las distintas metodologías a aplicar y, sobre todo, del sistema de evaluación. Siguiendo estas indicaciones, en el Cuadro 2 se muestra la guía docente prevista al inicio del curso 2019/20³ y la adenda efectuada con posterioridad para las materias de segundo cuatrimestre, de primer y segundo curso, respectivamente, de IEIA. En dicho cuadro, en su parte izquierda, podemos observar que dos asignaturas de primer curso (*Informática y Ciencias de Materiales*) no cambiaron ni sus metodologías ni su evaluación, otorgando a la prueba objetiva final una parte importante de la evaluación. Por el contrario, en la asignatura *Física II*, la prueba final redujo significativamente su importancia (de un 70 % a un 40 %), además de sustituir la solución de problemas por una evaluación continua, metodología que también se incluye en la materia de *Estadística* en vez de las prácticas vía TIC. En la parte derecha del Cuadro 2 se observa que todas las materias cambiaron su sistema de evaluación, eliminando metodologías de la evaluación (*Fundamentos de Electrónica y Fundamentos de Automática* no incluyen en la evaluación las prácticas de laboratorio), variando los porcentajes asignados a cada metodología (*Mecánica de Fluidos y Resistencia de materiales*) o incluyendo nuevas metodologías para sustituir otras (*Técnicas de Fabricación*).

De manera similar, el Cuadro 3 muestra las guías docentes de asignaturas⁴ de segundo cuatrimestre de primer y segundo curso de II. En este caso, en primer curso, no hay ninguna materia que mantenga su sistema de evaluación y la prueba objetiva final pierde importancia dentro de la evaluación en favor de las prácticas de laboratorio (véase *Programación II y Estadística*), también se incluyen en la evaluación otras metodologías docentes como prácticas vía TIC o trabajo autónomo (*Álgebra y Fundamentos de computadores*, respectivamente). En segundo curso, la tendencia es similar, la prueba objetiva final pierde importancia, otorgándose en algunas materias a las prácticas en laboratorio (*Concurrencia y Paralelismo y Redes*) y a metodologías docentes incorporadas en el estado de alarma (*Trabajos en grupos tutelados en Proceso Software*). En esta titulación, solo una materia de segundo curso man-

tiene su sistema de evaluación (*Sistemas inteligentes*).

Comparando ambas titulaciones (véase Cuadro 2 frente al Cuadro 3), se observa que las metodologías son muy similares, aunque algunas metodologías, como *Evaluación continua*, aparecen más en IEIA y otras, *Seminarios*, en II. Además, la prueba final pierde mayor peso en la evaluación en las materias de II que en materias de IEIA, llegando en algunos casos incluso a no realizarse (véase *Tecnología Electrónica* en el Cuadro 3).

4. Resultados académicos de las titulaciones

En esta sección se analizan los resultados académicos de ambas titulaciones. Las Figuras 1 y 2 muestran los resultados de la asignaturas analizadas de IEIA. De manera análoga, las Figuras 3 y 5 muestran esta misma información para II.

En el Cuadro 2 se observa que en primer curso únicamente son dos las materias que no modifican su sistema de evaluación (*Informática y Ciencias de Materiales*), los resultados académicos de la Figura 1 muestran que ambas materias son las que menor variación tienen en cuanto al número de aprobados, de hecho *Ciencias de materiales* mantuvo casi idéntico este dato (véase Cuadro 1). No obstante, otras materias con cambios más drásticos en la evaluación, como *Física II*, presentan un incremento notable en el número de aprobados (de un 49 % a un 85 %). De manera similar al primer curso, la materia que más redujo el peso asignado a la prueba final en segundo curso, *Fundamentos de Automática* de un 70 % a un 40 % (véase el Cuadro 2), es la que mejora ostensiblemente sus resultados académicos (de un 17 % de superados a un 79 %) y aquella con los cambios más sutiles, *Fundamentos de Electrónica* donde se mantiene el valor de la prueba y se cambian las prácticas de laboratorio por solución de problemas, mantiene sus resultados académicos (última columna de la Figura 2).

Los resultados académicos de II ilustrados en las Figuras 3 y 5 también señalan importantes diferencias en el número de aprobados, si bien no tan drásticas como en IEIA. De nuevo, la materia con un cambio más notorio en su sistema de evaluación en primer curso (supresión de la prueba en *Tecnología Electrónica*) es aquella donde la diferencia en el número de aprobados se incrementa en mayor medida (>20 %), la materia con menor incremento en el número de aprobados (un 12 % en *Álgebra*) sufrió también cambios en la guía docente, incluyendo prácticas a través de TIC para compensar la reducción en el porcentaje de la prueba objetiva. En el Cuadro 3 se observa que la asignatura que más denota el cambio de evaluación en segundo curso de II es *Con-*

³<https://estudios.udc.es/es/study/detail/770G01V01#plan-structure>

⁴<https://estudios.udc.es/es/study/detail/614G01V01#plan>

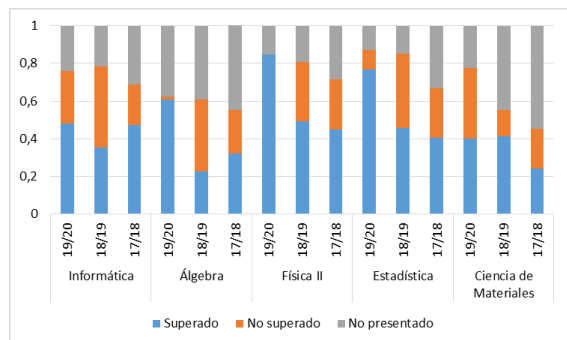


Figura 1: Resultados en las materias de 2º cuatrimestre de primer curso de IEIA.

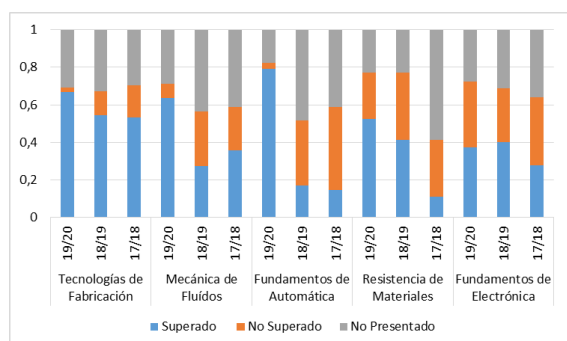


Figura 2: Resultados en las materias de 2º cuatrimestre de segundo curso de IEIA.

currencia y Paralelismo donde el número de aprobados supera el 80 %, cuando en cursos anteriores era alrededor del 40 %, en esta materia se incorporó una prueba práctica en la evaluación, además de puntuar más las prácticas de laboratorio en detrimento de la prueba final. La asignatura *Bases de datos* también experimentó un aumento destacable en el número de aprobados (un 20 % más) y, el resto de asignaturas no experimentaron cambios significativos, a pesar de haberlo hecho en el sistema de evaluación. A modo resumen, se ha calculado la diferencia media (con su desviación típica) entre en el número de aprobados del pasado curso frente a los cursos anteriores para las titulaciones de IEIA e II. En todos los casos de IEIA, la diferencia media es superior al 22 %, llegando al 31 % cuando se compara el segundo curso de 2017/2018. Además las desviaciones típicas también son altas (superiores al 15 % en primer curso y al 22 % en segundo) señalando la disparidad existente en las distintas materias del grado. En cuanto a II, las diferencias medias están alrededor del 16 %, solo en un caso se llega al 22 %. Además, las desviaciones típicas oscilan entre el 3 % y 12 %.

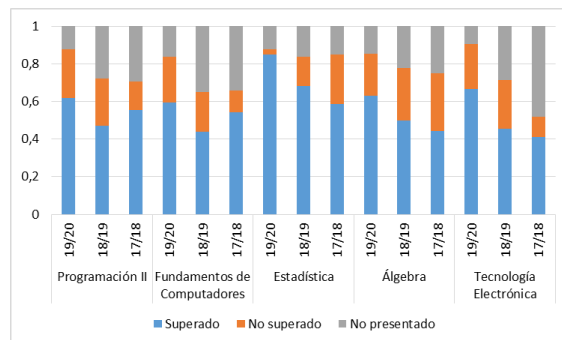


Figura 3: Resultados en las materias de 2º cuatrimestre de primer curso de II.

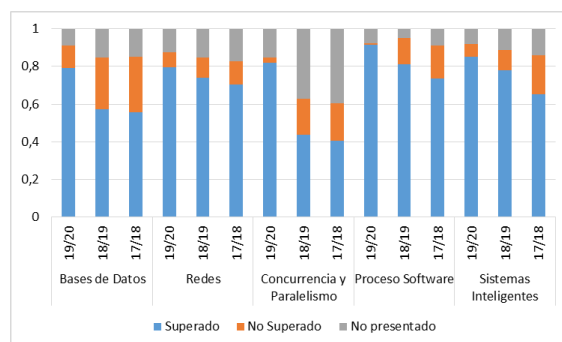


Figura 4: Resultados en las materias de 2º cuatrimestre de segundo curso de II.

5. Adaptaciones en Informática y Programación II

El sistema de evaluación de *Informática* y de *Programación II* se puede ver en los Cuadros 2 y 3, respectivamente. No obstante en esta sección abordaremos otros aspectos -por ejemplo, cómo se realizaron los exámenes o las prácticas de laboratorio- que hayan podido suponer una diferencia en los resultados académicos alcanzados, también ya ilustrados, de manera general, en las figuras 1 y 3.

5.1. Informática

Esta asignatura se encuentra dividida en dos partes bien diferenciadas, por un lado, la asimilación de conceptos teóricos básicos de informática (representación de la información, arquitectura de ordenadores, redes, etc.) y, por otro lado, una parte práctica relacionada con aprender a programar en lenguaje C. El sistema de evaluación es el que se muestra en la primera fila de materias del Cuadro 2. Aunque no se reflejan cambios, la metodología de solución de problemas se divide en tres actividades con mismo porcentaje en la evaluación

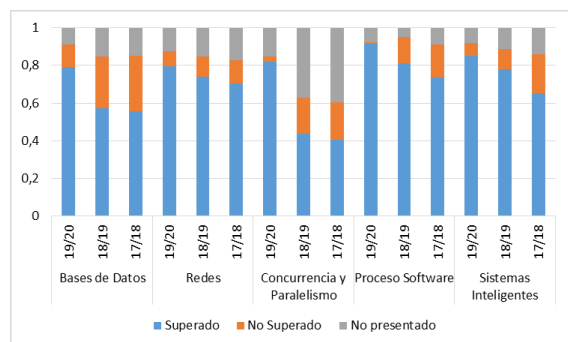


Figura 5: Resultados en las materias de 2º cuatrimestre de segundo curso de II.

(8 %): la realización de dos pruebas de evaluación presenciales donde los alumnos tienen que resolver problemas de programación y una práctica que se desarrolla de manera autónoma (véase el Cuadro 4). Tras la situación de confinamiento, se tuvo que cancelar la realización de la segunda prueba y el peso correspondiente se trasladó a la práctica. Esta práctica también alteró su contenido ya que en los últimos cursos se planteaba una práctica que consistía en la resolución de un problema usando robots y, dada la imposibilidad de hacer uso de estos dispositivos, se diseñó una práctica clásica de programación. Además, la práctica se realizaba en parejas y, en el curso 2019/20 se planteó de manera individual. Para verificar que el propio alumno implementa la práctica, además de usar software anti-plagio, se realiza una defensa ante el profesor que se mantuvo gracias al uso de videollamadas.

Las clases se realizaron en su horario habitual de manera virtual y síncrona (además las clases de teoría se grabaron para estar disponibles para los alumnos), respetando así las preferencias de los alumnos porque se reproduce mejor la dinámica a la que están acostumbrados⁵. Las clases prácticas no cambiaron su estructura, se proponían ejercicios prácticos que el alumno resolvía de manera autónoma bajo la supervisión del docente (usando videollamadas). Debe tenerse en cuenta que, según la tipificación de las actividades docentes, las actividades en línea síncronas se pueden considerar presenciales [3].

Por último, el examen, igual que en cursos anteriores, se divide equitativamente en teoría y práctica. El examen se realizó de manera virtual, síncrona y vigilada. La parte teórica fue de tipo test. En la parte práctica, considerando que no es lo mismo hacer los ejercicios

⁵COVID-19 y educación superior: De los efectos inmediatos al día después. Análisis de impactos, respuestas políticas y recomendaciones. UNESCO e IESALC. Disponible en <http://www.aneca.es/Sala-de-prensa/Noticias/2020/COVID-19-y-educacion-superior-de-los-efectos-inmediatos-al-dia-despues>, mayo de 2021

	Guía Docente	Addenda
Metodología	Eval.	Eval.
Prueba en	60	60
TTut	10	10
Prueba 1	8	8
Prueba 2	8	-
Práctica	8	16
PLab	6	6

Cuadro 4: Metodología (Metd.) y porcentaje de la evaluación (Ev.) asignado para cada metodología evaluable en Informática. TTut: trabajos tutelados, PLab: prácticas laboratorio.

usando un ordenador que un papel, se redujo el número de ejercicios.

5.2. Programación II

La asignatura *Programación II* se centra en los tipos abstractos de datos y algunas estructuras de datos clásicas (listas, pilas, colas, árboles), la parte teórica expone cómo desarrollar estas estructuras y en la parte práctica se implementan en lenguaje C. Las prácticas incluyen 2 entregas a realizar en parejas, la primera práctica aborda la implementación de listas y en la segunda se incorporan otras estructuras como multilistas, colas, etc. Tal y como se observa en el Cuadro 5 cada práctica suele tener un valor de un 10 % en la asignatura, otorgando el resto al examen. Además, los problemas planteados en tutorías de grupos reducidos pueden añadir un 10 % a la nota, una vez que la materia esté superada. En el curso 2019/20 la entrega de la práctica 1 no se vio apenas afectada por el estado de alarma que solo supuso una ampliación de una semana en su plazo de entrega. Consecuentemente, la práctica 2 también vio aplazado su plazo de entrega en otra semana. Además, el contenido de esta práctica se simplificó, abarcando únicamente listas y multilistas y sin añadir otras estructuras habituales (pilas y/o colas). Todas las prácticas de la asignatura pasan un software de detección de plagio y la detección de una copia supone el suspenso automático en la materia en las dos oportunidades. Habitualmente no se realiza defensa de las prácticas, aunque se puede recurrir a la misma en caso de duda.

Con respecto a las clases teóricas se mantuvieron para resolución de dudas (con muy poca afluencia de alumnos) y se realizaron numerosos vídeos que los alumnos podían consultar de manera asíncrona. Las clases prácticas se cancelaron y se extendieron los horarios de atención al alumnado.

El examen, en las dos oportunidades, se realizó de manera virtual síncrona, fue de tipo test, rotando alea-

	GuíaDocente	Addenda
Metodología	Eval.	Eval.
Prueba	80	20
Practica1	10	40
Practica2	10	40
SolProb	10	-

Cuadro 5: Metodología (Metd.) y porcentaje de la evaluación (Ev.) asignado para cada metodología evaluable en Programación II.

toriamente preguntas y las posibles respuestas de éstas, además fue secuencial (imposibilidad de ir hacia atrás para rectificar una respuesta) y se limitó su duración. Dado que el examen no fue vigilado, se redujo su valor en la evaluación, asignándose a las prácticas (véase el Cuadro 5).

6. Resultados académicos en Informática y Programación II

En esta sección se analizan los resultados de *Informática* y *Programación II* ya presentados de manera genérica en las Figuras 1 y 3, respectivamente. En la sección anterior hemos expuesto que la docencia en la materia de *Informática* se mantuvo síncrona en todo momento, mientras que en *Programación II* fue claramente asíncrona. A pesar de las diferencias metodológicas aplicadas debido al confinamiento, no se observan grandes diferencias en los resultados académicos. En *Informática* el número de no presentados se mantiene similar a cursos anteriores, mientras que el número de aprobados incrementa respecto al curso 2018/19 (un 12 %) y se mantiene igual al curso 2017/18. Por otro lado, *Programación II* experimenta un aumento notable del número de presentados que se traduce en un incremento de aprobados (15 % y 6 % con respecto al curso 2018/19 y 2017/18, respectivamente). En *Informática* el mayor cambio fue en la práctica de programación y en *Programación II* la alteración del sistema de evaluación que permite aprobar realizando únicamente las prácticas, así que en esta sección vamos a ver cómo fueron los resultados de dichas prácticas.

La Figura 6 muestra la distribución de las notas de las prácticas de *Informática* de los últimos cursos, se aprecia que las notas del curso 2019/20 han sido ligeramente superiores a las del curso anterior (mediana 8,43 y 7,63 en 2019/20 y 2018/19, respectivamente), aunque con mayor dispersión en las notas inferiores a la mediana, las notas del curso 2017/18 fueron claramente peores que los cursos posteriores. Cabe mencionar que la práctica del curso 2018/19 fue muy similar a la del curso anterior, teniendo en cuenta que la asigna-

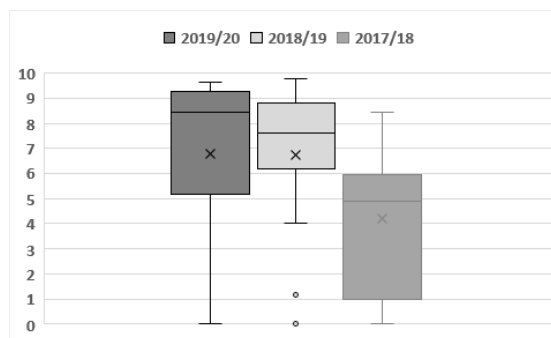


Figura 6: Notas de la práctica de programación en la asignatura *Informática*.

tura cuenta con bastantes alumnos repetidores, puede haber favorecido a la mejora en los resultados. Además, la participación en esta actividad ha sido notablemente mayor en el curso 2019/20 (86,49 %) que en los cursos anteriores (71,21 % en 2018/19 y 69,23 % en 2017/18). Estos datos pueden deberse al incremento del valor de la práctica (16 % de la nota final), el doble que en cursos anteriores lo que propició que muchos estudiantes participaran, aunque no alcanzaran buenas calificaciones.

El sistema de evaluación de la asignatura *Programación II* sufrió un cambio tan abrupto que no permite una comparación similar a la realizada con *Informática*, además, dado el mayor valor de las prácticas en la calificación final, se evaluaron de forma diferente, con lo cual un análisis comparativo no sería justo. Por tanto, analizaremos cuántos alumnos aprobaron o suspendieron gracias a las prácticas. El Cuadro 6 ilustra estos datos, se observa que el 35 % de los alumnos, en la primera oportunidad, y el 23 %, en la segunda, dependía del examen para aprobar, el resto ya tenía su nota determinada bien por no presentarse o bien por la calificación de prácticas. Así, en la primera oportunidad, y de manera análoga en la segunda, el 24,54 % de aprobados se debe únicamente a las prácticas y el 17,41 % a una combinación de prácticas y examen, formando el 41,95 % total. De igual forma, el 26,39 % de los suspensos se debe a la calificación de prácticas y el restante 17,15 % a bajas calificaciones en examen y prácticas.

En base a nuestra experiencia y a los resultados mostrados en la Figura 6 y el Cuadro 6 parece que la realización de las prácticas ha sido determinante en los resultados académicos de ambas materias. Por una lado, en *Informática*, aunque el valor de la práctica no fue tan elevado (16 % del total), las notas han sido mejores que en cursos anteriores y ha facilitado que el número de aprobados ascienda, ya que las notas del examen han sido similares. Por otro lado, en *Progra-*

		Junio	Julio
Alumnos		379	220
No presentados		14,51	28,64
Aprobados		41,95	38,64
Suspendidos		43,54	32,73
Notas gracias a prácticas	Aprobados	24,54	24,09
	Suspendidos	26,39	24,09
	A examen	34,56	22,73

Cuadro 6: Relación entre el número de aprobados y las prácticas realizadas en *Programación II*.

mación II y, como hemos comentado anteriormente, la nota de prácticas ha supuesto que muchos estudiantes ya no tuvieran que realizar el examen. En esta asignatura, las prácticas se realizaron en parejas y, en cursos pre-pandemia, se hacían controles de seguimiento periódicos para evaluar que ambos miembros de la pareja realizaban su labor. Estos controles tuvieron que suspenderse durante el curso 2019/20, así que se desconoce la implicación de cada miembro del equipo en la consecución de las prácticas. Por este motivo, podría haber alumnos - alrededor del 12 % - que hayan aprobado la materia, pero sin haber adquirido los conocimientos suficientes. Estos estudiantes, en cursos anteriores, probablemente no hubieran superado el examen y, consecuentemente, habrían suspendido la materia. Obsérvese que el porcentaje indicado (12 %) se aproxima al incremento en el número de aprobados con respecto al curso anterior (véase la Figura 3). Con todo ello, en esta materia creemos que los cambios drásticos en el sistema de evaluación no favorecieron un mayor aprendizaje, simplemente un mayor número de aprobados.

7. Conclusiones

En este trabajo se analizan las adaptaciones en las guías docentes, así como los resultados académicos de dos titulaciones del área tecnológica con gran diferencia en el número de alumnos matriculados: IEIA e II. Además, se analiza una materia de cada titulación, en concreto *Informática y Programación II* para ver qué estrategias se adoptaron y cómo repercutieron en los resultados académicos. A pesar de las diferencias en el número de alumnos, las metodologías aplicadas en las distintas asignaturas de ambas titulaciones son similares.

Cada materia se adaptó en función de sus características y en algunas los cambios fueron mínimos, incluso en su sistema de evaluación, manteniendo unos resultados académicos acordes con los cursos anteriores. Aún así, en los diversos resultados aportados se puede com-

probar que el 90 % de las materias experimentaron, en mayor o menor medida, un incremento en el número de aprobados, creemos que este incremento se debe a que, ante cambios importantes en los sistemas de evaluación -donde la prueba final pierde peso en favor de trabajos prácticos o evaluación continua-, se adoptaron medidas para no perjudicar a los alumnos, dada la complicada situación, lo que conllevó mejores resultados académicos. Estas medidas consistieron, por ejemplo, en las asignaturas analizadas en este trabajo, en asignar un mayor valor a la parte práctica que, además, vio reducida su complejidad frente a cursos anteriores. No obstante, esta mejora en el rendimiento, probablemente, no vaya acorde con una mejora en el aprendizaje. Un trabajo futuro interesante sería analizar en detalle el resto de las materias de las titulaciones para saber si reflexiones extraídas para *Informática y Programación II* son extrapolables.

Referencias

- [1] Habib Fardoun, Carina González, César A. Collazos y Mohammad Yousef. Estudio exploratorio en Iberoamérica sobre procesos de enseñanza-aprendizaje y propuesta de evaluación en tiempos de pandemia. *Education in the Knowledge Society*, 21. Ediciones Universidad de Salamanca, 2020.
- [2] Francisco J. García-Peñalvo. Evaluación online del aprendizaje: Reflexiones en tiempos de la COVID-19, 2020. DOI: <https://doi.org/10.5281/zenodo.3921801>.
- [3] Francisco J. García-Peñalvo. Modelo de referencia para la enseñanza no presencial en universidades presenciales. *Campus Virtuales*, 9(1):41–56, 2020.
- [4] Mario Grande de Prado, Francisco J. García Peñalvo, Alfredo Corell Almuzara y Víctor Abella-García. Evaluación en educación superior durante la pandemia de la covid-19. *Campus Virtuales*, 1(10):49–58, 2021.
- [5] R. H. Huang, D. J. Liu, A. Tlili, J. F. Yang, y H.H. Wang Handbook on facilitating flexible learning during educational disruption: The chinese experience in maintaining uninterrupted learning in COVID-19. Outbreak, 2020.
- [6] Giorgio Marinoni, Hilligje Van't Land, y Trine Jensen. The impact of COVID-19 on higher education around the world. *IAU Global Survey Report*, 2020.
- [7] María T. Sanz y Emilia López-Iñesta. Dos modelos de enseñanza-aprendizaje: una práctica blended learning en tiempos de COVID-19. *Tecnologías educativas y estrategias didácticas*, pags. 226–236, 2020.

Enseñando cómo evaluar la usabilidad de forma activa

Jose Ignacio Panach, Esther Durá, Miriam Gil, Verónica Romero

Departament d'Informàtica, Universitat de València, Valencia
joigpana@uv.es esther.dura@uv.es miriam.gil@uv.es
veronica.romero@uv.es

Resumen

En la mayoría de los Grados de Ingeniería en Informática hay al menos una asignatura sobre la Interacción Persona-Ordenador. En ella, el concepto de usabilidad es el eje central sobre el que se basa el conseguir interfaces que se ajusten a las necesidades de los usuarios. Aunque conceptos como Experiencia de Usuario se están entrando en la industria, cuesta ver la usabilidad como algo relevante. Al igual que en la industria, al alumnado le resulta difícil percibir la utilidad del diseño de interfaces usables. La contribución de este artículo es la descripción de una metodología basada en el aprendizaje por proyectos llevada a cabo en la Universitat de València dentro de la asignatura de Entornos de Usuario para motivar el aprendizaje de cómo evaluar la usabilidad. En grupos de 5 a 8 personas, el alumnado trabaja para aplicar una de las técnicas de evaluación de usabilidad en un sitio web real. Al finalizar la actividad, cada grupo expone al resto en qué consiste la técnica, los resultados, y sus ventajas e inconvenientes. Los datos extraídos a través de encuestas al alumnado y el análisis de las calificaciones, muestran que la motivación es más alta que en otras actividades, y que se han conseguido los objetivos y las competencias marcados.

Abstract

In most of the Computer Engineering Degrees there is at least one subject about Human-Computer Interaction. The concept of usability is the central axis to design interfaces that satisfy users' needs. Although concepts like User Experience are raising in the industry, it is hard to see usability as a relevant topic. As in industry, students do not appreciate the importance of usable interface designs. The contribution of this article is the description of a methodology based on a project learning process conducted in the Universitat de València within the User Environments course to motivate the learning of how to evaluate usability. In groups of 5 to 8 people, the students work to apply one of the usability evaluation techniques on a real web site. At the end of the activity, each group explains to the rest the details of the technique, the results, and its

advantages and disadvantages. The data obtained through student surveys and the analysis of the qualifications show that motivation is higher than in other activities, and that the objectives and competencies have been achieved.

Palabras clave

Interacción persona-ordenador, usabilidad, diseño de interfaces

1. Introducción

La ISO 25000 [10] aborda los aspectos de la calidad del software, donde la usabilidad es una parte esencial para obtener una alta aceptación del usuario. Según dicha ISO, la usabilidad se define como la *capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones*. Conseguir interfaces usables facilita la adecuación del usuario a la aplicación, el aprendizaje de la aplicación, el uso de la aplicación, y la protección del usuario contra errores.

Históricamente, a nivel industrial se ha dado más importancia a aspectos funcionales de los sistemas, dejando la parte de interfaces, y en concreto la parte de usabilidad, en un segundo plano, para las fases finales del desarrollo. Este tratamiento tardío de la usabilidad suele implicar un gran trabajo para corregir fallos de calidad, ya que muchos aspectos de usabilidad implican también cambios en la funcionalidad [13]. Últimamente se están tomando los aspectos de usabilidad como algo esencial para el desarrollo de sistemas, siguiendo el paradigma de *User Experience (UX)* [18]. UX es el conjunto de factores y elementos relativos a la interacción del usuario con un entorno o dispositivo concretos, dando como resultado una percepción positiva o negativa de dicho servicio, producto o dispositivo. Dicha percepción depende no solo de los factores relativos al sistema, sino de aspectos relativos a las emociones y sentimientos de los usuarios. UX propone que el usuario esté presente validando el desarrollo y considerando la usabilidad desde las primeras etapas del proceso, de forma que participe en las decisiones de diseño (Diseño Centrado en el Usuario [6]). Con el

paradigma UX las características de usabilidad se deben contemplar desde el inicio del desarrollo.

Esta evolución que considera la usabilidad como algo esencial en la industria también tiene su reflejo en el ámbito académico. Esto implica que suele haber al menos una asignatura obligatoria en los Grados Universitarios relacionados con la Ingeniería Informática que traten aspectos de la Interacción Persona-Ordenador (IPO). La usabilidad es un tema esencial en estas asignaturas. Sin embargo, es un tema que suscita poca motivación entre el alumnado, ya que lo ven como un elemento muy teórico que rara vez van a poder aplicar en el mundo real.

La contribución de este artículo es describir la metodología docente utilizada en la asignatura de Entornos de Usuario dentro del Grado de Ingeniería Informática de la Universitat de València con el fin de motivar al alumnado en el estudio de la usabilidad. La asignatura de Entornos de Usuario tiene, entre otras competencias, el aprendizaje de pruebas de usabilidad, el trabajo en equipo y la toma de decisiones. La metodología usada en la asignatura está basada en el aprendizaje dirigido por proyectos [14], y va enfocada a adquirir dichas competencias. Se forman grupos de 5 a 8 personas y cada grupo debe aplicar una técnica de evaluación de usabilidad distinta. De esta forma, se fomenta el autoaprendizaje activo a partir del material proporcionado por el profesorado y mediante tutorías grupales. Una vez finalizada la evaluación de la usabilidad, cada grupo debe exponer sus resultados al resto del alumnado. En la exposición, se debe describir el método utilizado, los resultados de la prueba de usabilidad, y las ventajas e inconvenientes de la técnica de evaluación utilizada. Tras cada exposición se fomenta la formulación de preguntas entre los asistentes.

Para medir el resultado de la metodología docente, se ha creado un cuestionario tipo Likert de 5 puntos con preguntas referentes a medir la motivación y si se han alcanzado los objetivos y trabajado las competencias. Los resultados muestran que más del 85% del alumnado indica una alta motivación y que tanto los objetivos como las competencias a trabajar han satisfecho sus expectativas. Mediante dos preguntas abiertas, también analizamos las opiniones del estudiantado en referencia a las ventajas e inconvenientes de la metodología docente. Por último, analizamos las calificaciones obtenidas en esta actividad, que denotan un nivel muy bajo de “No Presentados” (18%) y notas que oscilan del 8 al 10.

El resto del artículo se estructura de la siguiente forma. La Sección 2 resume los trabajos relacionados con el aprendizaje de técnicas de evaluación de usabilidad. La Sección 3 detalla las características de la asignatura y las 8 técnicas de evaluación. La Sección 4 describe la metodología docente. La Sección 5 aborda y discute los resultados. Por último, la Sección 6 muestra las conclusiones y trabajos futuros.

2. Trabajos relacionados

La relación de la usabilidad con el desarrollo software en un contexto docente es un tema que se ha trabajado desde distintos puntos de vista. Si nos centramos en trabajos previos que hayan abordado cómo enseñar técnicas para mejorar la usabilidad de las aplicaciones, podemos encontrar entre otros, el trabajo de Halawe [8]. En este trabajo, siete estudiantes aplican técnicas de evaluación heurísticas a sitios webs sobre conferencias. Previamente, los alumnos han tenido que aprender las heurísticas basadas en Nielsen [15]. Según Halawe, el uso de sitios web reales aumentó la motivación del alumnado. Los resultados muestran una alta valoración del alumnado en cuanto a la importancia de la usabilidad en el desarrollo software. En el trabajo de Nunes [16], se enseña cómo tratar la usabilidad desde el punto de vista ergonómico. El curso va orientado a ingenieros industriales, donde la ergonomía puede afectar a la eficiencia, efectividad y satisfacción de los usuarios finales de los productos a desarrollar. La propuesta de Nunes se basa en un aprendizaje activo por parte del alumnado, que debe diseñar sus productos usables a partir de: entrevistas, especificación de requisitos, prototipados, mejora cognitiva y aplicación de las heurísticas de Nielsen. Coleti y Morandini [4] proponen una metodología para enseñar la usabilidad basada también en la evaluación real de sitios web. En ese trabajo se desarrollan dos tipos de evaluaciones. Una primera evaluación basada en heurísticas a verificar a partir de una lista de ítems a analizar de forma sistemática, y una segunda basada en el reconocimiento de voz y gestos. En ambos casos se intentó que el entorno de evaluación fuera lo más real posible. Correa y Martins [5] presentan su experiencia de uso del aprendizaje basado en proyectos aplicados al ámbito de la Interacción Persona-Ordenador. El curso está enfocado para un contexto en línea, a distancia, y trata sobre la aplicación de varias técnicas de evaluación de usabilidad.

Hay trabajos que han abordado la enseñanza no solo de pruebas de usabilidad, sino de un diseño centrado en el usuario. Entre otros, cabe destacar el trabajo de Jesse [12], quien propone un curso para enseñar las fases de desarrollo teniendo en cuenta las características de usabilidad: prototipado, estructura, diseño de las interfaces y pruebas de usabilidad. El aprendizaje está planteado de una forma autónoma, donde el alumnado aplica a entornos reales los aspectos teóricos vistos en el aula. El trabajo de Getto et al. [7] incluye la usabilidad como parte de un proceso de desarrollo global basado en metodologías ágiles, incluyendo conceptos de gestión de proyectos. Dentro del curso se trabajan conceptos como: investigación preliminar de requisitos, prototipado, pruebas de usabilidad, y mantenimiento del sistema. Park [17] propone una metodología docente para trabajar la experiencia de usuario en la web, incluyendo características de usabilidad. Se plantea

como objetivos docentes la arquitectura de la información y su jerarquía. La parte de diseño gráfico queda en un segundo plano para la propuesta de ese curso, ya que no la considera relevante para mejorar la usabilidad.

Como conclusión de los trabajos relacionados analizados, se puede destacar que todos ellos intentan abordar la enseñanza desde un contexto práctico. Se intenta que tanto las pruebas de evaluación de usabilidad como los posibles diseños partan de problemas reales, donde el trabajo autónomo del alumno es esencial. Además, hay algunos trabajos que se centran en el aprendizaje basado en proyectos (como el [5]). De las técnicas de evaluación de usabilidad, cabe resaltar que hay trabajos que se centran en técnicas heurísticas [8, 16-17], mientras que otros se centran en técnicas con usuarios [7, 12]. Solo los trabajos [5] y [4] profundizan en ambas técnicas a la vez.

La contribución de este artículo sigue las propuestas de los trabajos previos: participación activa del alumnado a través del aprendizaje basado en proyectos para aplicar las técnicas de evaluación de usabilidad a entornos reales. Como aspecto diferenciador respecto a los trabajos relacionados, cabe destacar que, aunque en nuestra propuesta el alumnado solo aplica de forma práctica una técnica de evaluación, al final de la experiencia docente debe conocer todas las técnicas que han usado el resto de compañeros de la clase. Además, se fomenta un análisis crítico de las distintas técnicas, incentivando un debate entre ventajas e inconvenientes de cada una de las técnicas trabajadas por el alumnado.

3. Asignatura

3.1. Objetivos y competencias

La asignatura donde se ha llevado a cabo esta experiencia es Entornos de Usuario, una asignatura obligatoria de segundo curso del Grado de Ingeniería Informática de la Universitat de València. Esta es la única asignatura relacionada con la IPO en todo el grado, por lo que aborda todos los aspectos relacionados con la introducción de los conceptos IPO, arquitectura de los sistemas interactivos, desarrollo de interfaces en Java, estilos y paradigmas de interacción, y diseño centrado en el usuario. De esta última parte, la usabilidad es un tema esencial en los objetivos a alcanzar mediante la asignatura.

En concreto, los objetivos de la asignatura relacionados con la usabilidad son:

- O1: Ser capaz de analizar, diseñar interfaces, y crear prototipos de los mismos.
- O2: Conocer y saber aplicar las diferentes técnicas de evaluación de interfaces.
- O3: Ser capaz de comunicar de forma efectiva tanto escrita como oralmente conocimientos relacionados con las diferentes etapas del proceso de

diseño y desarrollo de interfaces de usuario.

- O4: Resolver problemas relacionados con el diseño de interfaces de usuario con iniciativa, tomando decisiones, con autonomía y creatividad.
- O5: Trabajo en grupo: saber cooperar, interactuar y dividir el trabajo con otras personas para resolver problemas.

Asimismo, las competencias a través de las cuales llegar a esos objetivos son las siguientes:

- C1: Capacidad para diseñar, desarrollar, evaluar y asegurar la accesibilidad, ergonomía, usabilidad y seguridad de los sistemas, servicios y aplicaciones informáticas, así como de la información que gestionan.
- C2: Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad.
- C3: Capacidad para saber comunicar y transmitir los conocimientos, habilidades y destrezas de la profesión de Ingeniero Técnico en Informática.
- C4: Capacidad para diseñar y evaluar interfaces persona-computador que garanticen la accesibilidad y usabilidad a los sistemas, servicios y aplicaciones informáticas.

3.2. Técnicas de evaluación de usabilidad

Con el propósito de alcanzar los objetivos anteriormente descritos en base a las competencias, dentro de la asignatura se abordan varias técnicas de evaluación de usabilidad. Con el aprendizaje de estas técnicas se abordan los objetivos O1 y O2, además de las competencias C1 y C4.

Las técnicas de evaluación que debe aprender el alumnado se dividen en dos bloques: técnicas heurísticas y técnicas con usuarios. Las técnicas heurísticas se basan en una evaluación teórica por parte de expertos en usabilidad, mientras que la evaluación con usuarios se basa en una evaluación con usuarios reales. A continuación, se describen los detalles de las técnicas que hay dentro de cada uno de estos dos bloques.

Heurísticas:

1. Las 10 heurísticas de Nielsen [15]. El alumnado tiene disponible las 10 heurísticas y ha de buscar ejemplos y contraejemplos de cuándo las heurísticas se aplican bien.
2. ISO 9126-1 [9]. El alumnado tiene disponibles las métricas definidas en la ISO para medir la usabilidad. Ha de adaptar esas métricas abstractas a elementos gráficos concretos de una aplicación sobre la que medir la usabilidad.

Con usuarios:

3. Prototipado [19]. El alumnado tiene que definir un prototipo de una aplicación y hacer la evaluación de la usabilidad sobre ese prototipo.
4. Observación de campo [3]. Mientras los sujetos interactúan con la aplicación, los evaluadores se limitan a observar las acciones que hace el usuario.
5. Observación de gestos [20]. Mientras los sujetos interactúan con la aplicación, los evaluadores observan sus expresiones faciales, cómo cambian y cuáles ponen en todo momento.
6. Cuestionario [1]. Después de utilizar la aplicación, los sujetos deben rellenar unos cuestionarios donde se les pregunta por la usabilidad que han experimentado en el uso de la aplicación.
7. Pensando en voz alta [11]. Mientras los sujetos interactúan con la aplicación, deben hablar durante todo momento sobre lo que están pensando; los problemas que identifican, lo que les gusta, lo que cambiarían, etc.
8. Medida de prestaciones [2]. Mientras los sujetos interactúan con la aplicación, los evaluadores toman nota de las tareas que hacen los sujetos correctamente (efectividad) y del tiempo que invierten (eficiencia).

Al finalizar el curso, el alumnado debe ser capaz de aplicar cualquiera de estas técnicas en un entorno real. En la siguiente sección describimos cómo abordamos la enseñanza de estas técnicas desde un punto de vista práctico en un aprendizaje basado en proyectos.

4. Metodología docente

La metodología propuesta en esta asignatura se basa en el aprendizaje activo del alumnado. Para ello, se divide al alumnado en 8 grupos de 5 a 8 personas (un grupo por cada técnica a trabajar). Cada grupo debe elegir un sitio web real y sobre éste aplicar la técnica de evaluación de usabilidad mediante la metodología basada en proyectos. La Figura 1 resume la metodología de forma gráfica. A continuación, detallamos cada uno de los pasos. En cada paso detallamos el tiempo del que dispone el alumnado para llevarlo a cabo:

1. Estudiar la técnica. Tras crear los grupos de trabajo, el profesorado les proporciona material referente a la técnica de evaluación que debe poner en práctica cada grupo. El alumnado, fuera de horario lectivo, debe revisar el material y entender el funcionamiento de la técnica. El plazo para esta tarea es de 1 semana.
2. Tutorías para guiar en la técnica. Cada grupo se reúne de forma particular con el profesorado. En estas reuniones, el profesorado les marca las pautas para elaborar el diseño de las pruebas de usabilidad. Las explicaciones y dudas abarcan todo

el proceso de evaluación, hasta el análisis de datos. Esto se realiza en una clase presencial de 2 horas de duración.

3. Elaborar el diseño. En base al estudio de la técnica y a los comentarios del profesorado, cada grupo debe diseñar su plan de pruebas para un sitio web real a su elección. Cada miembro del grupo debe definir al menos una tarea experimental dentro de un contexto específico que ellos definen. El plazo para realizar el diseño es de 1 semana.
4. Lanzar las pruebas. En base al diseño, cada miembro del grupo debe al menos reclutar a una persona para que haga de sujeto experimental. Durante la prueba, el alumnado debe recoger los datos propios de la técnica que está aplicando. El plazo para realizar el diseño es de 1 semana.
5. Analizar los datos. Una vez han finalizado las pruebas, el alumnado debe agregar los datos extraídos para concluir resultados y generar recomendaciones respecto al sitio web analizado. Los datos a analizar pueden ser tanto cuantitativos como cualitativos. En el material proporcionado en el paso 1 y en las tutorías del paso 2, el profesorado también abarca cómo analizar los datos de cada una de las técnicas. El plazo para analizar los datos es de 2 días.
6. Presentar resultados. Toda la experiencia que ha adquirido el alumnado a lo largo del proceso se debe compartir con el resto de la clase (que habrán aplicado una técnica distinta). Cada grupo debe presentar: un resumen de la técnica, descripción de las tareas o heurísticas aplicadas, análisis de resultados, y ventajas e inconvenientes detectados en la técnica. Todo ello se debe presentar en 10 minutos como máximo, y solo habla un alumno en representación de todo el grupo.
7. Resolver dudas. Al finalizar la presentación, el profesorado fuerza a que al menos uno de los grupos que están de oyentes formulen una pregunta al grupo que ha expuesto. La elección del grupo que pregunta se hace de forma aleatoria justo al terminar la presentación, para motivar a que todo el alumnado esté atento. El objetivo de las preguntas es doble. Por una parte, se pretende que los oyentes entiendan todos los detalles de la técnica presentada. Por otra parte, se pretende que los presentadores piensen en respuestas a cuestiones más allá de las tratadas en la presentación, fomentando el pensamiento crítico de la técnica utilizada. En la ronda de preguntas, cualquiera de los miembros del grupo que presenta puede tomar la voz para responder. Este paso se realiza en una clase presencial de 2 horas. Cada grupo tiene 5 minutos para responder dudas.

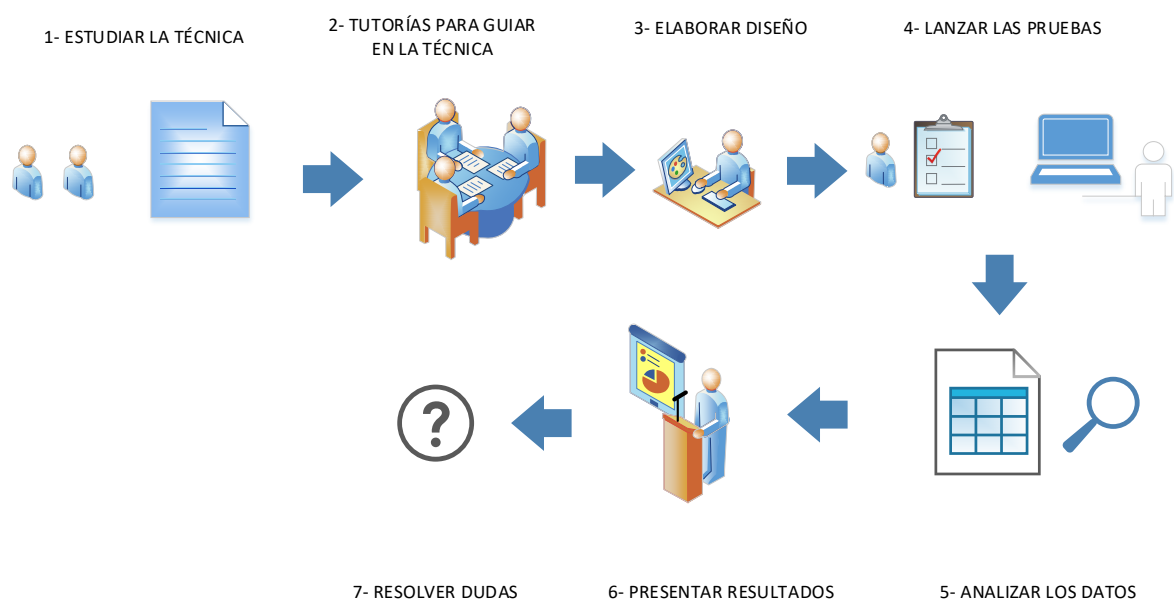


Figura 1: Resumen de la metodología docente

A lo largo de estos 7 pasos se trabajan todos los objetivos y las competencias marcadas en la asignatura. El Cuadro 1 muestra la relación entre los pasos de la metodología docente y los objetivos y competencias que se trabajan en cada uno.

	1	2	3	4	5	6	7
O1			X				
O2	X	X	X	X		X	X
O3						X	X
O4			X	X	X		
O5		X	X	X	X		X
C1	X	X	X	X	X		
C2			X	X	X		
C3						X	X
C4			X	X	X		

Cuadro 1: Relación entre los pasos de la metodología y los objetivos-competencias

- O1 pretende que el alumnado analice las interfaces y diseñe prototipos. Esto se consigue en el paso 3, donde se diseñan las tareas experimentales. El grupo que desarrolla la técnica de prototipado, además trabaja la destreza de crear los prototipos desde cero.
- O2 implica conocer las técnicas de evaluación. Este objetivo se aplica en las fases iniciales de la metodología, cuando la técnica se estudia y se define el diseño, además de en las fases finales cuando se presenta y se discute con el resto.
- O3 se centra en la comunicación efectiva, que se trabaja mayoritariamente en la parte final, al exponer los resultados y defender las ideas. En la

presentación también se fomenta un espíritu crítico sobre la técnica aplicada debatiendo las ventajas e inconvenientes.

- O4 trata sobre la resolución de problemas, que casa con el aprendizaje dirigido por proyectos aplicado en las técnicas de evaluación. Esto sobre todo se aplica en el diseño de la evaluación, el lanzamiento de la prueba, y el análisis de datos.
- O5 trata sobre el trabajo en grupo. Aunque todos los pasos de la metodología se hacen en grupo, hay algunos pasos donde se requiere una mayor coordinación. Son aquellos pasos referentes al diseño, ejecución, análisis y defensa de las cuestiones en público.
- C1 trabaja el diseño de interfaces usables, que se ve en todos los pasos iniciales hasta el lanzamiento de las pruebas.
- C2 se centra en la resolución de problemas, que coincide con los pasos del O4.
- C3 trata sobre la comunicación y defensa de ideas, que coincide con el O3 en los dos últimos pasos de la metodología.
- C4 trata sobre diseñar y evaluar interfaces. Esta capacidad se centra en el paso de diseño del experimento, su lanzamiento y el análisis de datos.

5. Evaluación de la metodología docente

Para evaluar si la metodología docente ha motivado el aprendizaje del alumnado y ha cubierto los objetivos y competencias marcados en la asignatura, hemos elaborado un cuestionario a rellenar por el alumnado. El

cuestionario está formado por 10 preguntas, 8 de ellas son tipo Likert de 5 puntos y 2 de ellas son abiertas. El detalle de las preguntas se puede ver en el Cuadro 2.

P1	¿Te ha resultado interesante el trabajo propuesto en esta actividad?
P2	¿Te ha motivado la forma en como se ha desarrollado la actividad?
P3	¿El hecho de que al acabar cada presentación la ronda de preguntas fuera de un grupo al azar, crees que ha incrementado más tu implicación y atención en las exposiciones de tus compañeros?
P4	El objetivo de este trabajo es que aprendieras sobre 8 técnicas de usabilidad y como usarlas, ¿crees que este objetivo se ha cumplido después de realizar la actividad?
P5	Después de haber realizado esta actividad, ¿te sientes más seguro para aplicar alguna técnica de usabilidad el día de mañana en tu trabajo?
P6	¿Crees que esta actividad te ha permitido trabajar la competencia sobre cómo diseñar interfaces usables?
P7	¿Crees que esta actividad te ha permitido trabajar la competencia sobre la resolución de problemas de forma autónoma y creativa, y hablar en público?
P8	¿Crees que esta actividad te ha permitido trabajar la competencia sobre la evaluación de interfaces usables?
P9	Menciona los aspectos positivos de esta actividad
P10	Menciona los aspectos negativos de esta actividad

Cuadro 2: Preguntas para evaluar la metodología docente

Podemos diferenciar 3 tipos de preguntas. Las preguntas P1 a P3 tratan sobre la motivación que la metodología suscita al estudiantado. Las preguntas P4 a P8 tratan sobre los objetivos y competencias de la asignatura. Por último, las preguntas P9 y P10 intentan extraer el pensamiento crítico sobre la metodología.

La Figura 2 muestra los resultados de las preguntas basadas en la escala Likert (P1 a P8). En las preguntas referentes al bloque de la motivación (P1 a P3), vemos que solo hay un 2% del alumnado que marca la preguntas como “Muy en desacuerdo” o en “Desacuerdo”. De “Neutral” hay solo un 10%. Esto implica que por encima de un 80%, el alumnado considera que la metodología docente les ha servido para motivar el aprendizaje de las técnicas de evaluación de usabilidad.

El segundo bloque de preguntas (P4 a P8) muestra que el alumnado también está ampliamente de acuerdo

con haber alcanzado los objetivos y competencias marcadas. Las preguntas que han obtenido algún desacuerdo son P4, P5, P7 y P8. El dato de P4 muestra que hay un pequeño porcentaje (8% del alumnado) que asegura no haber alcanzado el objetivo de aprender las 8 técnicas. Esto se puede deber a un bajo nivel de atención en las exposiciones orales del resto de grupos. P5, P7 y P8 muestran que hay un 4% del alumnado que no se siente seguro para aplicar las técnicas de evaluación, que no considera que han trabajado la resolución de problemas, y que no ha trabajado la evaluación de interfaces usables respectivamente. Es posible que estos alumnos tengan esta percepción por una baja participación activa dentro de sus grupos de trabajo. En cambio, nadie ha marcado la P6 con valores en desacuerdo, lo que indica que la competencia del diseño de interfaces usables la consideran trabajada. Si nos centramos en las respuestas que indican un alto grado de estar de acuerdo, las 4 preguntas están por encima del 75% del alumnado. Algunas, como la P8 están rozando el 95%. Estos datos demuestran que la gran mayoría del alumnado considera que tanto los objetivos como las competencias marcadas al inicio se han conseguido con la metodología docente.

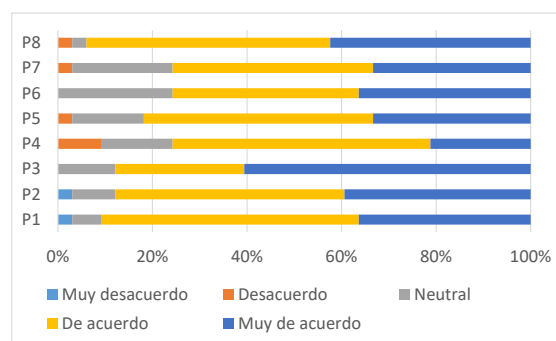


Figura 2: Gráfico con las respuestas a las preguntas Likert

Además de las preguntas Likert, mediante P9 y P10 pretendíamos obtener ventajas e inconvenientes de la metodología docente. A continuación, vamos a resumir las ideas más relevantes que podemos extraer de ambas preguntas. Respecto a los aspectos positivos:

- Se valora el trabajo en equipo y la coordinación.
- El forzar las preguntas al final de cada presentación y que no se sepa quién pregunta hasta el momento de las preguntas les motiva en la atención.
- Aprenden todas las técnicas a través de las explicaciones de otros compañeros.
- Mejora la creatividad y la resolución de problemas.
- Ayuda a entender los aspectos más relevantes sobre el diseño de interfaces y la usabilidad.

Respecto a los aspectos negativos, destacamos los siguientes:

- Poco tiempo disponible para la presentación.
- Poco tiempo para desarrollar la parte de cuestiones tras las defensas.
- El material con la descripción de la técnica era poco completo, se complementó con las tutorías.
- Una vez has expuesto y has hecho las preguntas, se presta menos atención a las siguientes.
- Hay técnicas que requieren mayor aprendizaje y el uso de más herramientas que otras.
- No poder aplicar todos los métodos de forma práctica.

A partir de las presentaciones podemos extraer observaciones específicas sobre las técnicas de evaluación de usabilidad. Las técnicas heurísticas se perciben más sencillas de aplicar, pero menos fiables que las técnicas con usuarios. La técnica de prototipado es la que requiere más dedicación ya que exige pensar desde cero un diseño que sea usable. Las técnicas que generan más dudas son aquellas que precisan de pocos artefactos, como las heurísticas y la observación de campo.

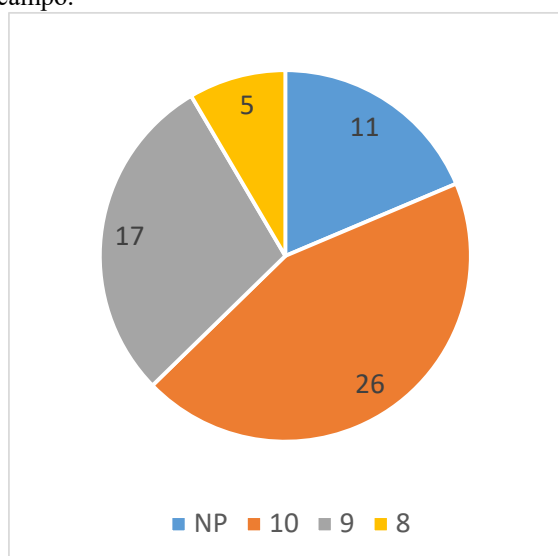


Figura 3: Número de alumnos y alumnas que han obtenido cada calificación

Además de los datos de las encuestas, también se han analizado los datos de la evaluación por parte del profesorado. Esta actividad se evalúa como parte de la nota continua que el alumnado va obteniendo mediante la entrega de boletines a lo largo del curso. Cabe destacar que la calificación obtenida en esta actividad es muy alta, oscilando entre 8 y 10. La Figura 3 muestra la cantidad de personas que han obtenido una calificación de 10, de 9, de 8, o no se han presentado a esta actividad (NP en el gráfico). Cabe resaltar que el porcentaje de no presentados es muy bajo (18%) si lo

comparamos con otras entregas del mismo curso, donde el porcentaje de no presentados siempre es superior al 25%. Por lo tanto, la metodología seguida ayuda a decrementar el número de no presentados.

En el examen final de la asignatura se incluyen preguntas referentes a cualquiera de las técnicas de evaluación. Estas preguntas pretenden evaluar que los alumnos hayan aprendido todas las técnicas y no sólo aquella sobre la que trabajan.

6. Conclusiones

Este trabajo presenta una metodología docente para que el alumnado aprenda 8 técnicas de evaluación de usabilidad con técnicas de aprendizaje activo y aprendizaje basado en problemas. Las técnicas consisten en 2 técnicas heurísticas y 6 basadas en el uso de usuarios finales. La metodología docente se ha evaluado a través de un cuestionario relleno por el alumnado y del análisis de las calificaciones obtenidas en la actividad. Ambos elementos muestran que la metodología ha aumentado la motivación del alumnado y que se han conseguido tanto los objetivos como las competencias marcadas para trabajar la usabilidad.

Es importante también resaltar el contexto en el que se ha llevado a cabo la metodología. Debido a las restricciones por la COVID-19, las tutorías con el profesorado, las reuniones de coordinación entre ellos, las presentaciones y las discusiones sobre preguntas de la clase fueron online a través de Teams. Este hecho puede implicar una comunicación menos fluida que si fuera presencial.

Como trabajo futuro, para los siguientes cursos, nos planteamos aplicar la técnica del Puzzle en lugar de la presentación para facilitar que todo el alumnado aprenda los detalles de todas las técnicas. También pensamos en incorporar más tutorías presenciales en la fase de análisis de datos. Actualmente las tutorías al principio del proceso se centran más en el diseño de la evaluación y las dudas referentes al análisis estadístico surgen más adelante. También pretendemos reducir los aspectos negativos identificados por los alumnos y descritos en la sección anterior.

Agradecimientos

Este trabajo se ha desarrollado gracias al proyecto “Soluciones para la docencia en tiempos de COVID-19: Clase Online Asíncrona + ABP + nuevos entornos digitales” (UV-SFPIE_PID-1355524)

Referencias

- [1] Ahlem Assila y Houcine Ezzedine, "Standardized usability questionnaires: Features and quality focus," *Electronic Journal of Computer Science and Information Technology: eJCIST*, vol. 6(1), pp. 15-31, 2016.

- [2] Nigel Bevan y Miles Macleod, "Usability measurement in context," *Behaviour & information technology*, vol. 13(1-2), pp. 132-145, 1994.
- [3] Lilliam Perurena Cancio y Mercedes Moráñez Bergues, "Usabilidad de los sitios Web, los métodos y las técnicas para la evaluación," *Revista Cubana de Información en Ciencias de la Salud (ACIMED)*, vol. 24(2), pp. 176-194, 2013.
- [4] Thiago Adriano Coleti y Marcelo Morandini, "A Teaching Strategy for Usability Evaluation to Human-Computer Interaction Courses," presentado en WEIHC, pp. 41-42, 2013.
- [5] Ana Grasielle Dionísio Corrêa y Valéria Farinazzo Martins, "Methodology applied problem-based learning in teaching HCI: A case study in usability evaluation of an online course," *Iberian Conference on Information Systems and Technologies, CISTI*, pp. 1-6, 2014.
- [6] Muriel Garreta Domingo y Enric Mor Pera, "Diseño centrado en el usuario," *Universitat Oberta de Catalunya*, pp. 9-12, 2010.
- [7] Guiseppe Getto, "Managing experiences: Utilizing user experience design (UX) as an agile methodology for teaching project management," *International Journal of Sociotechnology and Knowledge Development*, vol. 7(4), pp. 1-14, 2015.
- [8] Mohanad Halaweh, "Usability Testing of Conferences Websites: A Case Study of Practical Teaching," pp. 380-389, 2018.
- [9] ISO/IEC, "ISO/IEC 9126-1, Software engineering - Product quality - 1: Quality model," 2001.
- [10] ISO/IEC, "ISO/IEC 25000 - Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Guide to SQuaRE," 2010.
- [11] Riitta Jääskeläinen, "Think-aloud protocol," *Handbook of translation studies*, vol. 1, pp. 371-374, 2010.
- [12] Gayle Jesse, "A pedagogical approach toward teaching an information systems student how to conduct a web usability study for an honors project: A case study," *Proceedings of the Information Systems Education Conference, ISECON*, vol. 11(6), pp. 18-32, 2012.
- [13] Natalia Juristo, Ana M. Moreno y María Isabel Sánchez, "Guidelines for Eliciting Usability Functionalities," *IEEE Transactions on Software Engineering*, vol. 33(11), pp. 744-758, 2007.
- [14] Patricia Morales y Victoria Landa, *Aprendizaje basado en problemas*. Theoria vol. 13(1), pp. 145-157, 2004.
- [15] Jakob Nielsen, *Usability Engineering*: Morgan Kaufmann, 1993.
- [16] Isabel L. Nunes, "Teaching Usability to Industrial Engineering Students," pp. 155-162, 2017.
- [17] Myung Hae Park, "The teaching methodology of user experience on the web," *International Journal of Technology, Knowledge and Society*, vol. 8(3), pp. 159-168, 2012.
- [18] Jeff Sauro y James R Lewis, *Quantifying the user experience: Practical statistics for user research*: Morgan Kaufmann 2016.
- [19] Reinhard Sefelin, Manfred Tscheligi y Verena Giller, "Paper prototyping - what is it good for?: a comparison of paper- and computer-based low-fidelity prototyping," presentado en CHI 03 Extended Abstracts on Human Factors in Computing Systems, pp. 778-779, 2003.
- [20] Raphael Zenk, Matthias Franz y Heiner Bubb, "Emocard—an approach to bring more emotion in the comfort concept," *SAE International Journal of Passenger Cars-Mechanical Systems*, vol. 1(2008-01-0890), pp. 775-782, 2008.

Escritura colaborativa en la nube sobre clones de plantilla

Arturo Jaime¹, José Miguel Blanco², César Domínguez¹, Ana Sánchez²

¹Dto. de Matemáticas y Computación, Universidad de La Rioja, Logroño. ²Dto. de Lenguajes y Sistemas Informáticos, Universidad del País Vasco/Euskal Herriko Unibertsitatea, Donostia-San Sebastián.
arturo.jaime@unirioja.es, josemiguel.blanco@ehu.es, cesar.dominguez@unirioja.es,
ana.sanchez@ehu.es

Resumen

En los estudios de ingeniería informática es habitual la escritura colaborativa de memorias como colofón a los trabajos prácticos. Esta actividad exige algo más que utilizar un buen editor en la nube, ya que son bien conocidas las dificultades de los estudiantes al redactar, tanto individualmente como en equipo. Las plantillas de documentos se utilizan mucho, tanto en la empresa como en los trabajos prácticos universitarios. Estas facilitan la comprensión del alcance y tamaño de la tarea a realizar. Además, aportan un andamiaje muy interesante que ayuda a estructurar documentos complejos haciendo que los resultados sean más comprensibles. Por tanto, facilitan la escritura individual y colaborativa. Las plantillas se pueden complementar con ejemplos destacables realizados por estudiantes o revisados por profesores. Para promover una escritura colaborativa real aportamos a nuestros equipos de estudiantes clones, o copias, de plantillas en la nube. Si necesitamos manejar muchos clones de forma manual la tarea asociada nos puede hacer desistir. En el caso de los documentos de Google Drive se pueden programar funciones sencillas en Google Apps Script, de forma que se automaticen las tareas de gestión de los clones. Además, gracias al historial de versiones, podemos conocer cuánto y cuándo han contribuido los escritores de un documento.

Abstract

In computer science studies, it is common to write collaborative reports as the culmination of practical works. This activity requires more than just using a good editor in the cloud, as students' difficulties in writing, both individually and in teams, are well known. Document templates are widely used, both in the industry and in university practical works. They make it easier to understand the scope and size of the task to be performed. In addition, they provide a very interesting scaffolding that helps to structure complex

documents, making the results more comprehensible. Thus, they facilitate individual and collaborative writing. The templates can be complemented with relevant examples made by students or reviewed by teachers. To promote real collaborative writing we provide clones, or copies, of templates in the cloud to our student teams. If we need to handle many clones manually the associated task may discourage us. In the case of Google Drive documents, simple functions can be programmed in Google Apps Script to automate clone management tasks. In addition, thanks to the version history, we can know how much and when the writers of a document have contributed.

Palabras clave

Escritura colaborativa, andamiaje, plantillas, google apps script.

1. La escritura colaborativa

Algunos autores destacan que la escritura colaborativa consigue resultados de mayor calidad, aumenta la productividad y mejora las relaciones interpersonales [7, 9, 11].

Diferentes experiencias de escritura colaborativa identifican las dificultades que tienen los estudiantes al redactar, organizar contenidos y sintetizar su trabajo [3, 8, 13]. Estos problemas no se consiguen superar con una actividad docente aislada, aunque puede contribuir a mejorar la capacidad de comunicación escrita. Dice Joe Miró que aprender algo escribiendo sobre ello exige buena dosis de paciencia y trabajo, no solo del alumno, sino también del profesor [8]. También destaca que uno sólo escribe bien de lo que sabe, y pudo observar la tendencia en muchos estudiantes a circunvalar problemas centrándose en los puntos más simples de los temas abordados.

La escritura colaborativa exige de los equipos algo más que fragmentar la redacción entre los miembros y concatenar lo que entrega cada uno. En este sentido,

algunos estudios observan que los estudiantes son reacios a cuestionar o simplemente reflexionar sobre las ideas de sus compañeros [7]. La tecnología no resuelve todas las dificultades y es importante conocer y usar un modelo útil de trabajo colaborativo.

Una propuesta habitual para la escritura colaborativa es el uso de una plataforma en la nube como Google Drive o Microsoft OneDrive [9, 12, 13]. Con la ayuda de gráficas automáticas, que reflejan la participación de cada escritor durante el proceso de redacción, Yim *et al.* distinguen cuatro estilos de escritura colaborativa [12]. En el estilo *escritor principal* el documento lo escribe una persona con aportaciones secundarias de otros. En el estilo *divide y vencerás* cada escritor elabora su propio trozo de documento, sin retocar lo escrito por los demás. El estilo de *revisión cooperativa* se parece al anterior pero los escritores modifican lo escrito por otros, sobre todo en la fase final. Por último, en el estilo de *trabajo simultáneo*, se escriben frases conjuntas, se elaboran trozos de forma simultánea y se retoca el trabajo de los demás, sobre todo al principio y al final. Los estilos más habituales son la *revisión cooperativa* y el *escritor principal* y la mejor calidad se obtiene con el *divide y vencerás* [12]. A pesar de su interés, en el presente trabajo descartamos la realización de un análisis del estado del arte de la escritura colaborativa.

2. Las plantillas de documentos

Una plantilla es un esquema que sirve de guía para redactar un documento de calidad, proporciona secciones obligatorias y opcionales que completar e incluye comentarios intercalados que explican el objetivo y los contenidos a incluir [3]. Un estudiante de gestión de proyectos escribió una lección aprendida sobre las ventajas de las plantillas de proyecto [5]. Destacaba que las plantillas agilizan la elaboración de documentos, recordando los puntos a incluir, ayudando a concentrarse en lo importante y aportando uniformidad, que facilita las posteriores búsquedas. Además, su uso permite incorporar en sus sucesivas versiones mejoras relacionadas con el método de trabajo.

Las plantillas de documentos proporcionan cierto andamiaje (*scaffolding*) a los estudiantes que les facilita la comprensión del alcance y tamaño de la tarea. Además les ayuda a organizar y explicar mejor las tareas. El andamiaje combina el aprendizaje mediante ejemplos con el aprendizaje mediante la práctica [3, 7]. Las plantillas ayudan a estructurar documentos extensos o complejos, como una memoria de trabajo fin de grado, haciendo el resultado más comprensible. La estructura común

facilita la revisión de documentos escritos por otros compañeros y mejora las interacciones y la comunicación en el equipo, constituyendo una base común de entendimiento [1].

Las plantillas se pueden complementar con ejemplos ilustrativos. En el caso de trabajos fin de grado (TFG) pueden ser secciones destacables extraídas de memorias de otros estudiantes, como unos antecedentes o unas conclusiones bien escritas.

También se derivan algunos inconvenientes del uso de plantillas [3]. Muchos estudiantes evitan leer los comentarios de la plantilla y, cuando lo hacen, son reacios a preguntar por lo que no entienden. Como ocurre con los libros de texto, aunque la plantilla esté muy bien escrita y comentada, es necesario que los profesores expliquen a sus estudiantes cómo utilizarla [3]. Incluso con buenas plantillas habrá estudiantes con dificultades para realizar su trabajo de redacción. Por otra parte, cuando se escribe un documento en equipo, los estudiantes tienden a pensar que sus compañeros han entendido lo mismo que ellos y no ven la necesidad de discutir sobre los contenidos a elaborar y su ubicación más razonable [3].

3. La memoria del TFG

Diferentes autores parecen asociar al tutor de TFG un papel de mero verificador de la calidad de la memoria sin jugar ningún papel activo durante el proceso de redacción [6, 10]. Sin embargo, como decíamos anteriormente, muchos estudiantes tienen dificultades para escribir y organizar una memoria de cierto tamaño [3, 13]. Pensamos que puede tener sentido ver a la redacción de las memorias de TFG como un caso de escritura colaborativa donde el estudiante hace el rol de *escritor principal* (según los estilos identificados por Yim *et al.* [12]) y el tutor aporta comentarios, hace sugerencias y resuelve dudas. En los casos en los que el estudiante muestra muchas dificultades podría llegar a retocar algunas frases o incluso a escribir algún párrafo.

Este proceso de redacción supervisada se puede beneficiar del uso de editores en la nube. La memoria se puede ir redactando en un documento textual compartido entre estudiante y tutor. El tutor puede aportar un documento de partida con el formato y los estilos aceptados en la facultad o escuela (portada, fuentes, colores, etc.). Además, este documento puede incluir una plantilla con secciones obligatorias y optativas y con comentarios sobre los contenidos y estructura interna de cada una. También es interesante que la plantilla explique el orden de redacción más razonable, por ejemplo, se puede sugerir comenzar con los antecedentes y dejar para el final las

conclusiones, introducción y resumen. Esta plantilla puede ayudar al estudiante a superar bloqueos.

Lo ideal sería utilizar una plantilla consensuada entre el cuadro de profesores que participan en los tribunales evaluadores. Sin embargo, no es sencillo conseguir estos consensos. Cada uno suele tener su propio punto de vista sobre qué secciones interesan o qué contenidos son esenciales. También hay muchos tipos de proyecto que exigen su propia estructura. No es lo mismo un proyecto de desarrollo de una aplicación web, que uno de testeo de una aplicación, o de depuración de una fuente de datos (en inteligencia de negocio), por poner algunos ejemplos. Seguramente sería más fácil ponernos de acuerdo respecto a un estilo de redacción que sea claro, directo, correcto y breve.

Ya hemos explicado cómo puede ayudar a los estudiantes el andamiaje proporcionado por una plantilla al realizar esta tarea. Nosotros diseñamos hace dos cursos una plantilla sencilla para la elaboración de memorias y la hemos utilizado con unos pocos TFG con buenos resultados. Disponemos de algunas memorias recomendables, realizadas por estudiantes, que complementan la plantilla. Nuestra idea es seguir seleccionando memorias o secciones de memoria que ilustren nuevos tipos de proyecto.

El documento donde se escribe la memoria del TFG es inicialmente propiedad del profesor, quien comienza aportando un clon o copia de la versión con la plantilla. El estudiante adquiere el rol de escritor. Esta situación se puede reorientar y buscar la configuración que parezca más razonable. Trabajar de esta forma aporta algunas ventajas. Tutor y estudiante tienen siempre acceso a la última versión y a la rama de versiones que se generan de manera automática. Al consultar una versión se puede saber quién introdujo nuevas líneas o cambios, como se observa en la Figura 1. Cuando alguno hace comentarios el otro los recibe por email y los ve en el documento.

4. Documentación en equipo

La realización de trabajos o pequeños proyectos es una práctica muy frecuente en las asignaturas de los grados de informática [3, 4, 6, 10]. Es habitual que los estudiantes entreguen una memoria breve donde se describan, por ejemplo, los análisis, diseños, implementaciones o problemas superados. Además, en muchas ocasiones, estas actividades se desarrollan en equipo, generalmente debido al elevado número de estudiantes que cursan la asignatura. Por tanto, es deseable que la memoria se redacte de forma colaborativa entre los miembros de cada equipo.

Una de estas asignaturas es la que introduce los conceptos de dirección de proyectos, en las que

habitualmente se realizan miniproyectos donde aplicar el ciclo de vida de un proyecto informático [4] y se tocan algunas de las áreas básicas del PMI. Dos de los autores de este artículo impartimos la asignatura sobre esta materia en nuestros respectivos centros universitarios. En cada miniproyecto los estudiantes elaboran al menos dos documentos, uno contiene la planificación y en el otro se va recopilando información de seguimiento y control.

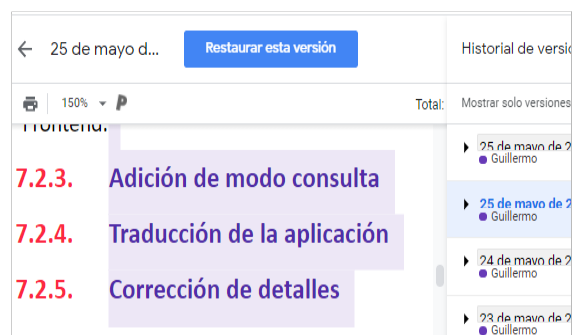


Figura 1: Sistema de versiones de Google Drive. Los colores marcan qué autor introdujo el cambio en una determinada versión del documento.

Hemos ido variando el modo de recopilar esta documentación. Los primeros cursos la pedíamos en papel para evitar su descarga desde una plataforma y la retroalimentación la escribíamos a mano sobre el entregable, que después mostrábamos a los interesados. Para hacernos una idea de las dimensiones de esta tarea de revisión, vamos a poner un ejemplo. En una de nuestras asignaturas del curso 19-20 participaron en las prácticas 39 estudiantes. Organizamos para ellos una serie de cuatro miniproyectos. El primero fue individual, el segundo lo realizaron distribuidos en catorce equipos, el tercero en doce equipos y el último en seis equipos. Cada miniproyecto individual o en equipo generó dos documentos. Esto da 142 documentos, de los cuales 64 se redactaron en equipo de forma colaborativa. Las páginas de los documentos varían. Los del primer proyecto tenían entre dos y tres páginas, mientras que los del último proyecto tenían entre seis y nueve. Además, si deseamos revisar versiones intermedias, todavía se generan más páginas. Esto último tiene sentido cuando queremos comprobar si se está haciendo seguimiento del miniproyecto. No sería raro que la tarea de seguimiento se redujera a redactar el documento final sin que haya habido ninguna actividad de seguimiento real. En conclusión, tendríamos demasiadas páginas para revisar impresas en papel, con lo que ello conlleva. Así que pasados dos cursos comenzamos a pedir los documentos en

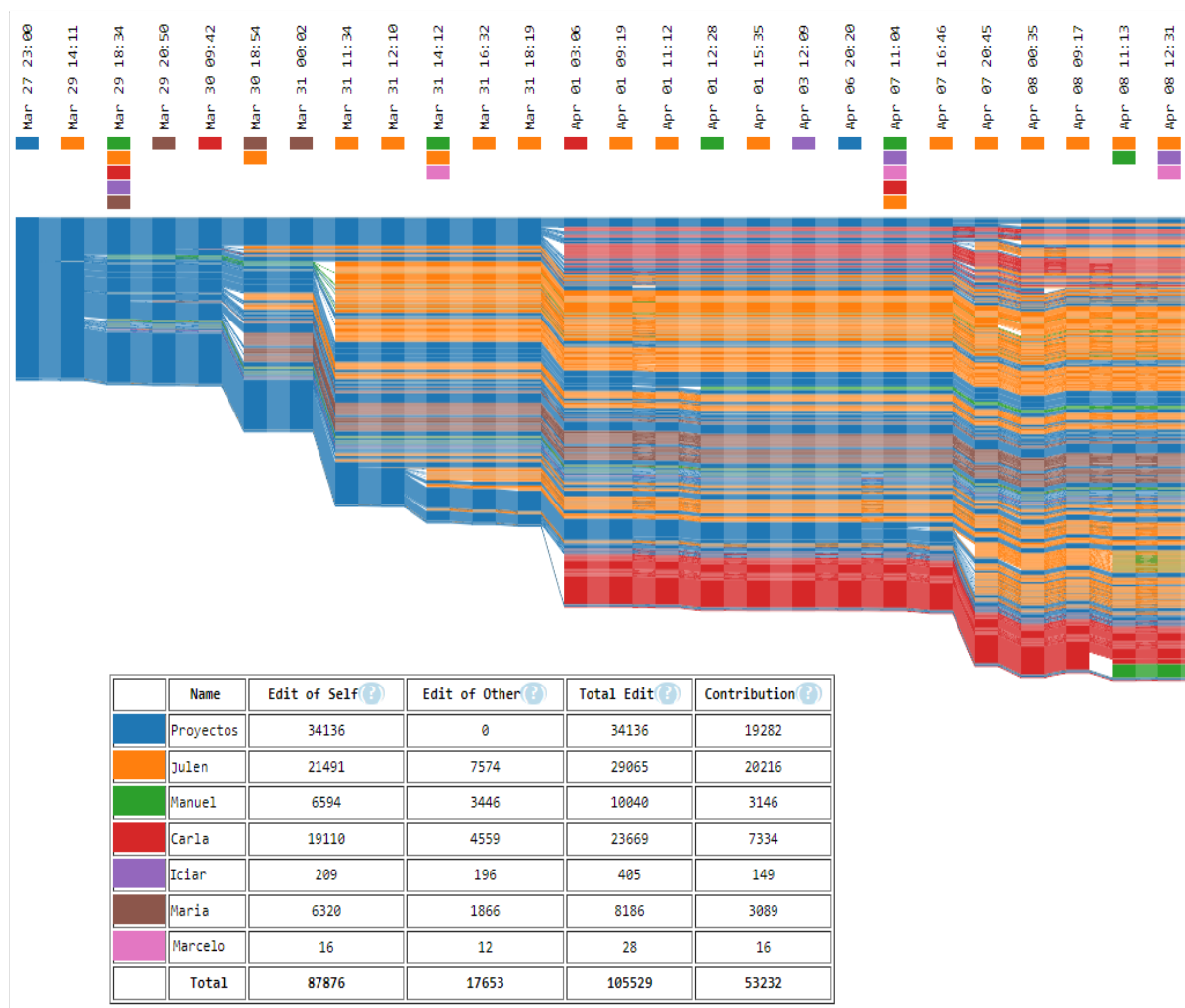


Figura 2: Gráfica docuViz de una escritura colaborativa de tipo “trabajo simultáneo” por un equipo de siete personas. Este estilo suele incluir bastantes modificaciones entre compañeros y sesiones simultáneas.

formato PDF y a buscar formas cómodas de corregirlos en pantalla.

Durante los primeros cursos dábamos unas pautas de los contenidos esperados en los documentos. Por ejemplo, para el documento de seguimiento y control del cuarto proyecto pedíamos aproximadamente lo siguiente: *contendrá las tareas realizadas, cuándo se hicieron y el tiempo dedicado. Incluirá un apartado con las actividades realizadas por cada miembro del equipo. También se comparará lo que se ha hecho con lo planificado, explicando las razones de las desviaciones significativas. Esto último se hará tanto para el conjunto del equipo como para cada uno de sus miembros. Se incluirán los aspectos oportunos en relación al seguimiento de comunicaciones, calidad, riesgos, cambios y adquisiciones.* Los documentos

recogidos incluían algunos de estos puntos. Pero lo sorprendente es que muchos equipos, incluyendo aquellos formados por buenos estudiantes, “olvidasen” partes de lo solicitado, por breve que pareciera la descripción anterior. Esto nos llevó a facilitarles, a partir del curso 14-15, una versión muy simple de plantilla para cada documento, que incluía básicamente las secciones obligatorias. Con el paso de los cursos, y a la vista de las entregas que corregimos, hemos ido mejorando estas plantillas, que actualmente cuentan con comentarios, tablas y figuras personalizables, textos introductorios genéricos, etc.

Hay, al menos, dos puntos positivos derivados de esta organización. El primero es que con las plantillas nadie se “olvida” de presentar algo que le estemos

pidiendo y que las entregas contienen generalmente la información que pedimos. El segundo es que la corrección es mucho más fácil, ya que los documentos que revisamos, correspondientes a una misma entrega, siguen una estructura común. Además, los equipos plantean dudas (sobre todo a última hora) respecto a los contenidos de alguna sección, a pesar de los comentarios de la plantilla, en lugar de saltárselo, como hacían antes.

Por otra parte, las plantillas eran inicialmente documentos ODT o DOCX. Esto llevaba a que los documentos en elaboración no se compartieran en el equipo hasta conseguir una versión muy avanzada o definitiva. Tampoco es algo que nos extrañase, ya que nosotros mismos hacíamos cosas parecidas al escribir artículos de investigación. Pero, a partir del curso 17-18, pensamos que debíamos dar un impulso definitivo a la escritura colaborativa y compartimos clones de las plantillas exclusivamente como documentos textuales de Google Drive. Además, obligamos a los equipos a redactar sobre los propios clones hasta llegar a la versión final. Elegimos esta plataforma porque es accesible, tiene muchas herramientas interesantes y sobre todo porque dispone de un lenguaje de *script* que nos permite desarrollar programas para manipular documentos. Entendemos que cualquier otra plataforma de características similares serviría para conseguir idénticos propósitos.

Algunos trabajos [12] utilizan trazas automáticas, como la que se muestra en la Figura 2, que representan la colaboración entre los escritores de un documento. Esta traza ha sido generada por una extensión de Google Chrome denominada docuViz [11]. El gráfico distingue a los escritores por colores. Los ejemplos de las Figuras 2 y 3 corresponden a los tipos de colaboración “trabajo simultáneo” y “escritor principal” en dos equipos de siete personas cada uno. Como se puede observar en los ejemplos, varios miembros no hacen ninguna contribución. En ambos casos, el primer autor (de color azul) representa al profesor, ya que es quien ha elaborado la plantilla inicial. En las cuadrículas inferiores se asocia a cada escritor el color que lo representa. Cada barra vertical oscura de la gráfica corresponde al documento en un punto temporal, es decir a una versión. En la parte de arriba aparece la fecha y hora de la versión y el color de los escritores que participaron en ella. Cada fragmento de barra del mismo color corresponde a un grupo de líneas consecutivas escritas por la misma persona. La altura de los fragmentos representa el número de líneas. Dentro de una barra vertical se respeta el orden de las líneas en el documento, las de arriba representan las primeras líneas y las de abajo las últimas. Los colores se reparten de esa forma

porque están representando en qué parte del documento ha participado cada autor. Las líneas que unen las barras, están conectando a las líneas que se mantienen sin eliminar del documento y así se pueden apreciar las inserciones y borrados entre versiones del documento. El recuadro inferior contiene información de tipo estadístico. Para cada escritor figura el número de líneas que editó, modificó de otros escritores y el total de líneas en las que contribuyó.

La gráfica que presenta docuViz es muy interesante para comprobar quién ha contribuido, en qué medida, cuándo lo ha hecho y en qué zona aproximada del documento. Pero, al utilizar la herramienta hemos podido comprobar que necesita bastante memoria, que es algo lenta y que si no dispone de suficiente memoria se bloquea en la fase de carga de datos.

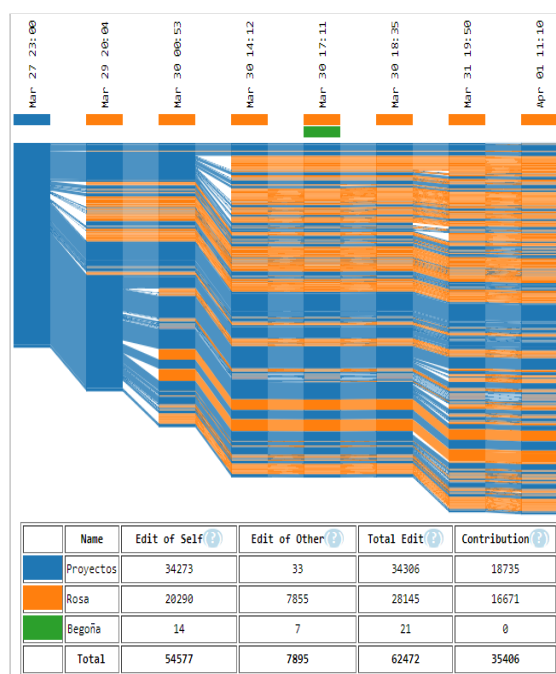


Figura 3: Escritura colaborativa de tipo “escritor principal” en un equipo de siete personas.

5. Clones en Google Drive

Clonar plantillas elaboradas por profesores y asignarlas a equipos, o a estudiantes que escribirán su memoria del trabajo de fin de grado, nos pareció una buena idea para impulsar la escritura colaborativa sobre una estructura de documento común. Pero, recordemos que en el ejemplo del curso 19-20 necesitamos crear 142 clones. Cada clon habrá que compartirlo, por ejemplo, con el director del equipo. Hacerlo de forma manual, como explicábamos para el caso de los TFG, es una tarea larga y tediosa. Después, se necesita abrir cada uno para corregirlo,

quizá descargar un PDF, etc. Afortunadamente, en Google Drive contamos con un lenguaje de *scripting* denominado Google Apps Script (GAS) [2]. Se trata de una API que se invoca desde código JavaScript y permite realizar de forma automatizada, por ejemplo, tareas de edición en documentos.

Para automatizar la gestión de nuestros clones codificamos cuatro *funciones*. Los datos necesarios para ejecutar las funciones los almacenamos en hojas de cálculo. La Figura 4 muestra la información para clonar un documento (en formato JSON) situada en una celda de una hoja de cálculo. Hay tres identificadores de google, “userList” es el de una libreta de cálculo con información de los equipos, “template” el identificador del documento textual a clonar y “folder” el de la carpeta donde quedarán los clones. Además contiene el nombre de la pestaña y columna (“userListTab” y “userListCol”) donde está la lista de usuarios con quienes se compartirán inicialmente los clones. La Figura 5 muestra la hoja de cálculo con la información de los equipos y, para cada equipo, el usuario de google con quien se compartirá el clon (semioculto por privacidad).

```
{
  "userList": "13X6aBPw34puKGiZYcdNn62vt4sxcKA",
  "userListTab": "P2",
  "userListCol": "E",
  "template": "1Zd5b5aXUUuhj2qztbIMGBnE3WpVcrmKtUo",
  "folder": "1ONBUo-mWJFIDUppO0mxmFDH1Npw9",
  "clonesPrefix": "PI20-E220-",
  "cloneShare": "google"
}
```

Figura 4: JSON para gestión de clones.

A	B	C	D	E	
Código	Mod	IdSecreto	google user_id	E220	E230
Puppis	Ca	730J	@gmail.com	1r545GeMGyOTHK1	
Orion	Ci	054B	@gmail.com	14CBptsSACN_Z6Nr	
Monoceros	Do	073Z	@gmail.com	12jAje6rR0c818IKGT	
Cetus	G	404W	@gmail.com	1ENiuJ2TgwiRpIRBR	
Pyxis	GM	984Q	er76@gmail	1gaAl426lcVerjUfyx	
UrsaMajor	F	428V	4@gmail.cc	1IMagLqG95Lte8gW	
Lepus	In	800R	3@gmail.cc	11-srUFUo0610_jxr-t	
Columba	Mik	746M	1@gmail.co	1G3i_xAGQuRc92VCi	
Triangulum	Mu	190B	mail.com	1u9jBDbkcaFOmYjD	

Figura 5: Información de los equipos e identificadores de los clones que se van generando

La primera de las funcionalidades es *generar clones*. La función crea un clon del documento que contiene la plantilla para cada equipo de la Figura 5.

Los usuarios de google con los que se comparte el clon son los directores de equipo, quienes lo compartirán después con sus compañeros. Los clones se crean en la carpeta “folder” del profesor. El nombre se construye con el prefijo “clonesPrefix” de la Figura 4 y el nombre de equipo de la Figura 5. Se asigna al director el rol de editor. En la hoja de cálculo de la Figura 5 se almacenan los identificadores google de los clones que se van generando (columna E). En la Figura 6 vemos cómo al seleccionar un documento y pulsar el icono de información (letra i en un círculo), Drive muestra los iconos de los usuarios que lo comparten. Al pasar el cursor por encima vemos el rol de cada uno (“lector” para el usuario de la M).

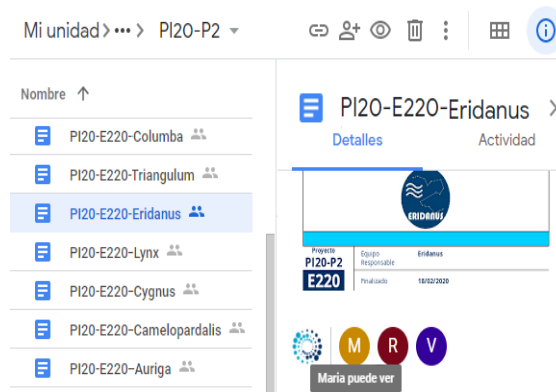


Figura 6: Usuarios que comparten un documento.

```
// Ejecutar Depuración cloneC

// Crea un clon de la plantilla
// json.template en la carpeta
// json.folder. El nombre del clon
// comienza con json.clonesPrefix y se
// completa con el parámetro team. La
// función devuelve el fichero clonado.
//
function cloneOne(team, json) {
  var folder =
    DriveApp
      .getFolderById(json.folder);
  var file =
    DriveApp
      .getFileById(json.template)
      .makeCopy(json.clonesPrefix+team,
                folder);
  return file;
}
```

Figura 7: Función simple con Google Apps Script

La segunda funcionalidad es *descargar instantáneas en PDF* del contenido de los clones de un determinado documento (el plan E220 en el ejemplo de las Figuras 4 y 5). La tercera funcionalidad es el *cambio de rol desde escritor a solo lector* (y viceversa) a todos los usuarios que tienen compartido cualquiera de los clones de un determinado documento. La última funcionalidad consiste en *insertar texto común en los clones* de un documento (tomando el texto de una celda de una hoja de cálculo) con retroalimentación general para todos los equipos. La retroalimentación específica para un documento se hace escribiendo comentarios o texto destacado directamente con el editor.

La Figura 7 muestra un ejemplo de una función sencilla que crea un clon (con `.makeCopy`) en la carpeta `json.folder` del documento con identificador `google json.template`. El nombre que recibirá el clon es la concatenación del prefijo `json.clonesPrefix` y el nombre del equipo, pasado como parámetro.

Además de las funcionalidades explicadas, nos gustaría disponer en el futuro de información de seguimiento sobre la participación de los componentes sin recurrir a soluciones externas y ligadas a un navegador concreto como `docuViz`.

El código GAS también tiene limitaciones que conviene tener presentes. Una de ellas es el tiempo máximo de ejecución, que no puede superar los seis minutos. En algunas ocasiones la generación de clones supera este tiempo y nos obliga a realizarla en varias fases. Es posible superar este problema, por ejemplo, mediante la definición de un disparador temporal (activador, en nomenclatura GAS) que genere en cada activación un grupo pequeño de clones y se desactive al crear todos ellos.

6. Conclusiones

Numerosos autores identifican las dificultades de los estudiantes, tanto al redactar como al comprender los contenidos solicitados, en las memorias de sus trabajos universitarios. Una forma de aliviar estos problemas es construir plantillas que guíen a los estudiantes en el alcance, contenido y longitud de los escritos. Nuestra experiencia de nueve años revisando documentos en asignaturas de dirección de proyectos (planificación, seguimiento y control...) nos permite afirmar que la incorporación y revisión de plantillas ha conllevado una mejora sustancial de la calidad de las entregas. Poco a poco hemos conseguido que sean más completas, más claras de entender y que contengan lo esperado. Creemos que esta percepción se confirmará en un estudio de tipo cuantitativo a realizar como trabajo futuro.

Además, muchos de los trabajos del ámbito de los grados de ingeniería informática se realizan en equipo. Hace ya tiempo están disponibles buenas herramientas para la escritura colaborativa en la nube. Aún así, animar a utilizarlas en nuestras asignaturas no fue suficiente y sólo se empezaron a usar cuando empezamos a obligarlo desde el curso 17-18. Esto no penalizó el nivel de satisfacción de nuestros estudiantes con la asignatura y, por el contrario, les parece una solución lógica.

Pero, si impulsamos la escritura colaborativa en la nube, lo coherente parece utilizar el mismo medio para canalizar las entregas. Y una vez en este medio, también vemos la posibilidad de hacer algún tipo de seguimiento de lo que ha escrito cada estudiante. Esta es una oportunidad única, que no la ofrecen las entregas finales firmadas por todos los miembros del equipo. Algunas herramientas muestran gráficamente el grado de participación de los estudiantes y cuándo contribuyeron. Nuestra propuesta para conseguirlo es que los profesores también formemos parte del equipo de escritores de cada documento. Para ello necesitamos compartir todos los documentos con los equipos. Nuestra oportunidad de conseguirlo es ofrecer la versión inicial con una plantilla elaborada por nosotros y mejorada curso a curso.

Cuando el uso de editores en la nube implica pocos documentos, como es el caso de las memorias de TFG, compartir clones de una plantilla es una tarea sencilla. Sin embargo, cuando se maneja y revisa un número elevado de documentos la gestión manual de clones exige demasiado tiempo. En el contexto de Google Drive contamos con el lenguaje de *scripting* Google Apps Script que permite automatizar procesos como la generación de clones, la descarga de instantáneas en PDF, el cambio de rol a los estudiantes que comparten los clones o la inserción de texto en los clones para aportar retroalimentación. La programación de *scripts* sencillos, como los mencionados aquí, está a nuestro alcance y facilitan enormemente la gestión de los clones.

Agradecimientos

Este trabajo ha sido parcialmente subvencionado por el Ministerio de Economía, Industria y Competitividad, proyecto EDU2016-79838-P.

Referencias

- [1] Heidi J. C. Ellis, Gregory W Hislop. Structuring Software Engineering Learning within Open Source Software Participation. En *Proceedings of the 2014 conference on Innovation & technology in computer science*

- education, ITiCSE '14, pp. 326–326, Uppsala, Sweden, Junio 2014.
- [2] Google team. Google Apps Script. Reference. 6 de octubre de 2020. Disponible en <https://developers.google.com/apps-script/reference>
- [3] Gregory W. Hislop, Heidi J. C. Ellis. Using Scaffolding to Improve Written Communication of Software Engineering Students. En *Proceedings of the 2009 Sixth International Conference on Information Technology: New Generations, ITNG '09*, pp. 707–712, Las Vegas, USA, April, 2009.
- [4] Arturo Jaime, José Miguel Blanco, César Domínguez, Ana Sánchez, Jónathan Heras, Imanol Usandizaga. Spiral and Project-Based Learning with Peer Assessment in a Computer Science Project Management Course. *Journal of Science Education and Technology*, 25:439–449, 2016.
- [5] David Lacalle-Castillo. Plantillas de documentos: ventajas e inconvenientes. Project Knowledge 14, Junio 2019. Disponible en <http://projectknowledge14.blogspot.com/2019/06/plantillas-de-documentos-ventajas-e.html>.
- [6] Ángeles López, Lledó Museros, María José Aramburu Cabo, Pedro García Sevilla. Experiencia de implantación de la asignatura de Proyecto de Final de Grado en Ingeniería Informática dentro de un contexto empresarial real. En *Actas de las XXIII Jornadas sobre Enseñanza Universitaria de la Informática, Jenui 2017*, pp. 23–30, Cáceres, julio 2017.
- [7] Rose M Marra, Linsey Steege, Chia-Lin Tsai, Nai-En Tang. Beyond “group work”: an integrated approach to support collaboration in engineering education. *International Journal of STEM Education*, 3(1):1–15, 2016.
- [8] Joe Miró Julià. Aprendizaje a través de la escritura: experiencias. En *Actas de las VI Jornadas de Enseñanza Universitaria de Informática, Jenui 2000*, pp. 451–456, Alcalá de Henares, septiembre 2000.
- [9] Lucía Ubilla Rosales, Lilian Gómez Álvarez, Katia Sáez Carrillo. Escritura colaborativa de textos argumentativos en inglés usando Google Drive. *Estudios Pedagógicos*, 43(1):331–348, 2017.
- [10] Fermín Sánchez Carracedo, Jaume Moral Ros, David López Álvarez. Mis estudiantes acaban el TFG (todos) en el tiempo previsto (casi todos). En *Actas de las XXI Jornadas de la Enseñanza Universitaria de la Informática, jenui 2015*, pp. 42–49, Andorra La Vella, julio 2105.
- [11] Dakuo Wang, Judith S. Olson, Jingwen Zhang, Trung Nguyen, Gary M. Olson. DocuViz: Visualizing Collaborative Writing. En *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI 2015*, pp. 1865–1874, April 18 - 23 2015, Seoul, Republic of Korea, Abril 2015.
- [12] Soobin Yim, Dakuo Wang, Judith Olson, Viet Vu, Mark Warschauer. Synchronous writing in the classroom: Undergraduates’ collaborative practices and their impact on text quality, quantity, and style. En *Proceedings of the Conference on Computer Supported Cooperative Work, CSCW'17*, pp. 468–479, Portland Oregon USA, Febrero 2017.
- [13] Marta Zorrilla. Desarrollo de las habilidades de escritura y trabajo colaborativo utilizando herramientas de la nube. *Actas de las XLX Jornadas de Enseñanza Universitaria de Informática, Jenui 2013*, pp. 3–10, Castellón de la Plana, julio 2013.

Fomentando el contacto entre alumnado y empresas para las prácticas externas integradas con el Trabajo Final de Grado

M. Asunción Castaño¹, Lledó Museros¹, Angeles López^{1,2}, Reyes Grangel³

¹ Departament d'Enginyeria i Ciència dels Computadors, ² Instituto de Nuevas Tecnologías de la Imagen

³ Departament de Llenguatges i Sistemes Informàtics

^{1,2,3} Universitat Jaume I, Castellón

castano@uji.es, museros@uji.es, lopeza@uji.es, grangel@uji.es

Resumen

El Grado en Ingeniería Informática de la Universitat Jaume I integra la realización de prácticas externas y el Trabajo de Final de Grado (TFG) en una sola asignatura, en la que los estudiantes desarrollan un proyecto durante la realización de las prácticas curriculares en una empresa. Para mejorar la información que recibe el estudiantado sobre los proyectos disponibles, se organiza anualmente una 'Jornada de Presentación de las Empresas Colaboradoras', que inicialmente consistía en una presentación corta por parte de cada empresa. Cuando se incrementó el número de empresas participantes, la jornada se cambió a formato feria, que ha demostrado ser más ágil y facilitar una comunicación más directa entre profesionales y estudiantado. En el curso 2020/2021, debido a la emergencia sanitaria, la feria se celebró en línea. En este trabajo se explica el proceso para organizar la feria, tanto presencial como virtual, las dificultades encontradas y los resultados de ambas experiencias. El análisis de los datos recogidos indica un elevado grado de satisfacción de los asistentes con la jornada en general y con el formato feria en particular.

Abstract

The Bachelor's Degree in Computer Engineering of Universitat Jaume I integrates the external internships and the Final Degree Project in a single subject, in which students develop a project during their internship in a company. In order to improve the information that the students receive related to the available projects, a 'Presentation Day of Collaborating Companies' has been organized, which consisted on a short presentation of each company. With the increment of participating companies, it was necessary to change the format of the conference to a fair. This format has shown to be more agile and to facilitate communication between professionals and students. In the 2020/2021

academic year, due to the health emergency, the fair was held online. This paper explains the process to organize both face-to-face and virtual fairs, the encountered difficulties and the results of both experiences. The analysis of the data collected shows a high degree of satisfaction of the attendees with the event in general and with the fair format in particular.

Palabras clave

Trabajo de Final de Grado, prácticas externas, relación universidad-empresa, entorno virtual.

1. Introducción

El Grado en Ingeniería Informática de la Universitat Jaume I integra la realización de prácticas externas y el Trabajo de Final de Grado (TFG) en una sola asignatura obligatoria de 18 créditos, en el segundo semestre de cuarto curso. Durante estas prácticas, los estudiantes desarrollan un proyecto profesional afín a su itinerario. Después redactan una memoria técnica sobre el trabajo llevado a cabo y lo presentan ante un tribunal universitario.

Una de las fases claves en la organización de esta asignatura es el proceso de asignación de los proyectos propuestos por las empresas al estudiantado. Para ello, se publica una descripción de cada propuesta, los estudiantes indican sus preferencias, y el equipo coordinador asigna a cada estudiante un proyecto (con el correspondiente profesional de la empresa que le supervisará) y un tutor académico.

La descripción de cada propuesta de proyecto contiene título, objetivos, tareas y tecnologías a emplear. Sin embargo, pronto se comprobó que una mera descripción, aunque necesaria, no era suficiente para que los estudiantes se hicieran una idea adecuada del trabajo a desarrollar.

Con el objetivo de mejorar esta información, desde el curso 2015/2016 se organiza una 'Jornada de Pre-

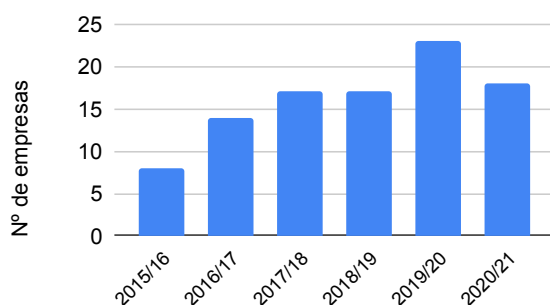


Figura 1: Empresas participantes en la jornada.

sentación de las Empresas Colaboradoras', que inicialmente consistía en una presentación corta por parte de cada empresa seguida de un pequeño turno de preguntas. La celebración de la jornada no solo mejoró la satisfacción del alumnado sino también la de las empresas. Al incrementarse el número de empresas participantes (ver Figura 1), esta jornada se organizó con un formato de feria. Este formato ha demostrado ser más ágil y facilitar una comunicación más directa entre profesionales y estudiantado.

En el curso 2020/2021, debido a la emergencia sanitaria, la feria se celebró en formato virtual. En este trabajo se explica el proceso de organización de la feria, tanto presencial como virtual, las dificultades encontradas y los resultados de la experiencia. El análisis de los datos recogidos indica el elevado grado de satisfacción de los asistentes en ambos formatos.

En la sección 2 se hace una breve revisión de trabajos relacionados. En la sección 3 se explican las características de la asignatura y la complejidad de la planificación y organización de la misma. En la sección 4 se describe cómo la 'Jornada de Presentación de las Empresas Colaboradoras' se ha ido adaptando a las circunstancias. En la sección 5 analizamos esta experiencia y en la sección 6 planteamos las lecciones aprendidas.

2. Estado del arte

En [6] se analizan los pros y los contras de las prácticas curriculares en empresa y su influencia en los TFG apoyándose en un amplio estudio bibliográfico. Por un lado, las estancias prácticas en empresa presentan ventajas para el aprendizaje de los estudiantes, ya que permiten que estos trabajen con tecnologías emergentes aplicando los conocimientos que han adquirido y recibiendo retroalimentación de profesionales. Además, las prácticas curriculares incrementan la autonomía y autoestima de los estudiantes [3]. También mejoran la empleabilidad [6, 7, 10] y promueven la colaboración entre universidad y empresa, tanto en el campo de la

investigación como en el ámbito académico [1, 5]. Asimismo, del análisis de [6] se deduce que las estancias prácticas en empresa ejercen una influencia muy positiva en los TFG. El estudio también revela que, para abordar con éxito las estancias prácticas en empresa y su derivación en TFG es clave identificar y alinear las expectativas de docentes, empresas y alumnado [4].

En los grados universitarios españoles relacionados con la Ingeniería Informática se han llevado a cabo distintas iniciativas para acercar las empresas a nuestros futuros graduados, ya sea en el marco de la docencia de asignaturas concretas [1, 5], en el desarrollo de TFG en empresas [8, 12] o en la realización de estancias prácticas en empresas. Pese a los beneficios que aportan estas prácticas curriculares, en la mayoría de los casos se plantean como actividades opcionales en el plan de estudios [5, 7, 10], probablemente debido a las dificultades que conlleva su organización. Aunque también hay grados en los que las prácticas curriculares en empresa tienen un carácter obligatorio, como es el caso del grado de nuestra universidad o el de la Universidad de la Rioja [6].

La difusión de las ofertas de prácticas propuestas por las empresas suele llevarse a cabo mediante una jornada de presentación en la que el estudiantado pregunta a los representantes de las empresas sobre el funcionamiento de estas y sobre el trabajo a realizar. Pero, en ninguno de los casos arriba mencionados se ha planteado la jornada con un formato tipo feria, como los presentados en este trabajo.

3. Contexto del trabajo realizado

Tal como se indica en [13], en la asignatura 'Prácticas Externas y Proyecto de Final de Grado' del Grado en Ingeniería Informática de la Universitat Jaume I cada estudiante realiza una estancia de 300 horas en una empresa, durante la cual desarrolla un proyecto de naturaleza profesional. El proyecto debe ser del ámbito del itinerario que cursa el estudiante (Ingeniería del Software, Sistemas de Información, Tecnologías de la Información o Ingeniería de Computadores) y debe integrar las competencias adquiridas en las enseñanzas, tal como se propone en el currículum IEEE/ACM [9]. Este proyecto se completa con una memoria y una defensa ante un tribunal universitario [2].

No sería posible el desarrollo de esta asignatura, tal y como está planteada, sin la colaboración fundamental de las empresas de nuestro entorno. Vincular la realización de las prácticas externas con el TFG ha sido un desafío importante para todos los agentes implicados, teniendo en cuenta que la asignatura tiene una media anual de 53 matriculados para los que hay que definir un TFG profesional adecuado (Figura 2).

La vinculación de las prácticas curriculares con el

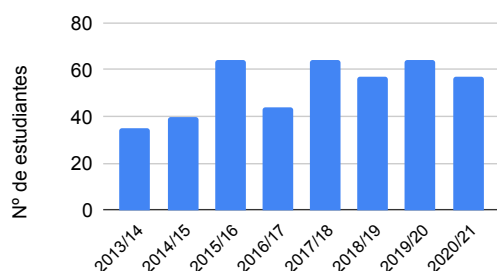


Figura 2: Número de matriculados por curso.

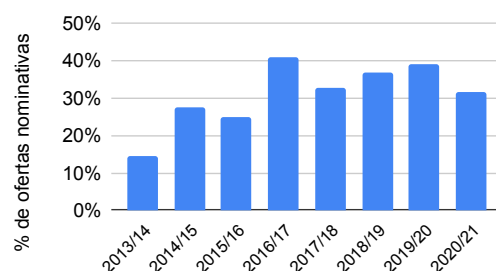


Figura 3: Porcentaje de ofertas nominativas por curso.

TFG requiere un mayor esfuerzo tanto por parte de las empresas como por parte de las coordinadoras de prácticas y TFG para definir proyectos de calidad para el estudiantado. Las propuestas de proyectos que hacen las empresas contienen una descripción de las tareas a desarrollar. Esta debe ser bastante detallada, ha de encajar en al menos uno de los itinerarios y tiene que ajustarse al nivel requerido para un TFG y a las horas establecidas. Para que una propuesta cumpla con estas características, suelen ser necesarias varias iteraciones con la empresa antes de consensuar la que finalmente se presentará al estudiantado.

A pesar del laborioso proceso de definición de las propuestas, en la valoración que el estudiantado de la primera promoción hizo sobre la gestión de la asignatura, el aspecto peor valorado fue el referente a la información que los estudiantes recibían sobre el proyecto a desarrollar y sobre la empresa. Por ello, tras una reunión con algunas empresas colaboradoras, se propuso celebrar cada año una 'Jornada de Presentación de las Empresas Colaboradoras' antes de que el estudiantado tuviese que elegir. Consecuentemente, desde el curso 2015/2016 se ha invitado a las empresas a participar en esta jornada, en la que estas presentan brevemente sus líneas principales de actividad y las propuestas concretas de proyectos que ofrecen al estudiantado. La jornada ha sido siempre muy bien valorada, tanto por las empresas participantes como por el estudiantado y profesorado que ha asistido y ha solventado en gran medida ese desconocimiento inicial del alumnado. Además, en numerosos casos la jornada origina contactos entre empresas y estudiantes que derivan en propuestas para estudiantes concretos, dando lugar a propuestas nominativas que facilitan el proceso de asignación de ofertas. Durante los últimos años se ha producido un notable incremento de dichas ofertas nominativas, tal y como puede verse en la Figura 3. La correlación entre el número de empresas asistentes (Figura 1) y el número de ofertas nominativas (Figura 3) a lo largo de los cursos es del 87,5 %.

Tras la jornada se finaliza el proceso de preparación de propuestas y, una vez publicadas, los alumnos seleccionan aquellas que más le interesan y se asigna,

por orden de expediente académico, una oferta a cada alumno respetando sus preferencias. Una vez asignada una oferta, el alumno realiza las 300 horas de trabajo en un entorno profesional bajo la tutela de un supervisor de la empresa y de un tutor de la universidad. Cabe resaltar la importancia de proporcionar al estudiantado una supervisión técnica adecuada, ya que esto suele derivar en proyectos con mejores resultados académicos [11]. Una vez terminada la estancia, el estudiante debe redactar la memoria del TFG bajo la supervisión del profesor tutor y realizar una presentación ante un tribunal. El tribunal evalúa las competencias adquiridas por el alumno teniendo en cuenta el trabajo y la memoria realizados, así como los informes de desempeño y calidad del trabajo que cumplimentan tanto el supervisor como el tutor.

4. Descripción de la jornada

Desde el primer curso de implantación de la asignatura, se realiza una sesión de presentación para explicar al estudiantado matriculado cómo se organiza la asignatura, las opciones disponibles, el proceso de asignación del proyecto y la empresa donde se va a desarrollar, los plazos, los mecanismos de supervisión, etc. Desde que se inició la jornada, esta se realiza a continuación de la presentación, con el objetivo de favorecer la asistencia de los estudiantes a ambos eventos.

4.1. Formato conferencia

Inicialmente, la 'Jornada de Presentación de las Empresas Colaboradoras con el Grado en Ingeniería Informática' consistía en una presentación corta por parte de cada empresa, seguida de un pequeño turno de preguntas. El segundo año de celebración, se añadió un pequeño ágape al terminar las presentaciones, con el objetivo de facilitar el acercamiento entre empresas y estudiantes, y también entre empresas y docentes.

Desde la primera jornada observamos que las empresas dedicaban generalmente más tiempo a mostrar sus líneas de trabajo, sus áreas profesionales, su metodología de trabajo, sus tecnologías y sus productos

que a explicar los proyectos concretos propuestos para nuestros estudiantes. Pero esto no supuso un problema sino más bien al contrario, ya que esta información era justamente la que echaban de menos los estudiantes y la que servía para complementar la descripción técnica del proyecto.

La celebración de la jornada generó un aumento de la satisfacción del alumnado respecto a la información previa sobre los proyectos. Pero también supuso una mejora en la satisfacción de las empresas, pues muchas de ellas habían solicitado poder conocer a los estudiantes, hablar con ellos y entrevistarlos. Así mismo, cabe destacar el incremento progresivo que experimentó el número de empresas participantes en las diferentes jornadas (ver Figura 1).

El problema principal de este formato de la jornada era la duración. Cada presentación de una empresa suponía unos 15 minutos, tiempo que se tuvo que reducir cuando aumentó el número de empresas participantes. Por un lado, esa duración era demasiado corta para las empresas, que estaban dispuestas a contar mucho más de lo que les daba el tiempo asignado. Por otro lado, el tiempo total para los asistentes era excesivo, por lo que algunos estudiantes no se quedaban hasta el final.

La celebración del ágape al final de la jornada permitió que los estudiantes y profesionales de las empresas pudieran hablar y aclarar más cuestiones que las que daba tiempo a plantear durante las presentaciones. Además, el hecho de que entre los profesionales se encontrasen algunos recién graduados favorece un contacto más directo. Cabe matizar que, a menudo, el profesorado presente ha ejercido de orientador y facilitador de la comunicación entre alumnado y empresas.

La buena acogida de la jornada por parte de los asistentes nos confirmó que la celebración de estas jornadas era enriquecedora para todas las partes, y nos animó a continuar organizándola. Pero, ante el incremento del número de empresas participantes, hubo que buscar un formato alternativo que permitiera celebrar la jornada sin extenderla más allá de una mañana.

4.2. Feria presencial

De este modo surgió la idea de migrar el diseño de la jornada a un formato de feria. En la feria cada empresa disponía de un stand en el que la Universidad le proporcionaba mesas, sillas, espacio para colgar un póster, wifi y enchufes. Las empresas asistentes traían material divulgativo sobre su sector de negocio y sus necesidades y muchas de ellas demostraciones.

La feria contaba también con una zona de coloquio, donde se dispuso de un ágape para los asistentes. Esta zona daba pie también a conversaciones entre empresas y alumnos, así como entre empresas y coordinadoras y entre alumnos y coordinadoras, lo que permitía generar interacción y resolver dudas.

Una primera encuesta informal realizada tanto a empresas como a alumnos tras la primera feria reveló una elevada aceptación de la jornada, por lo que decidimos repetir el formato. Dado el elevado tiempo requerido para organizar la feria, durante el segundo año la organización de esta feria se coordinó también con la Oficina de Inserción Profesional y Estancia en Prácticas (OIPEP) de la universidad. Esta oficina se encargó de la publicidad de la jornada y de la organización y distribución del espacio de la feria para las empresas, así como de las inscripciones de los alumnos.

4.3. Feria virtual

En el transcurso del curso 2019/2020 se produjo la emergencia sanitaria debida al COVID-19¹, lo que originó que la mayoría de las prácticas en empresa se finalizaran mediante teletrabajo. Al prolongarse la situación de alerta sanitaria, se barajó la posibilidad de cancelar la feria del curso 2020/2021, pero finalmente se decidió organizarla en formato virtual.

Para ello se configuró un entorno virtual utilizando la plataforma `gather.town`², en la que cada alumno, profesor o empresa estaba representado por un avatar. Esta plataforma frente a otras permite configurar entornos virtuales a medida y crear avatares para los visitantes. La selección de la misma vino determinada por su facilidad de uso y de configuración y su relación calidad-precio. Las autoras de este trabajo investigaron el uso de dicha herramienta, gestionaron la licencia de uso correspondiente y construyeron un entorno virtual para celebrar la feria.

La Figura 4 muestra una imagen del entorno desarrollado, que contaba con una zona reservada para un atril, en la cual la directora de la Escuela Superior de Tecnología y Ciencias Experimentales y la directora del Grado en Ingeniería Informática realizaron la inauguración de la feria, zonas de descanso para reuniones informales entre asistentes, dotadas incluso de pizarras digitales inteligentes, y zonas delimitadas para los stands de las empresas. Cada empresa tenía asociado su correspondiente logotipo identificativo y podía compartir en su stand documentos y pantalla y proyectar un fichero pdf o un vídeo divulgativo. Esta plataforma permite:

- Delimitar las conversaciones, pues estas son audibles únicamente en una distancia configurable, que en este caso era la zona delimitada para cada empresa. De este modo se efectuaban reuniones en paralelo sin interferir unas con otras.
- Realizar comunicaciones a todo el público, bien desde el atril o a través del uso del chat. El chat

¹<https://es.wikipedia.org/wiki/COVID-19>

²<https://gather.town>

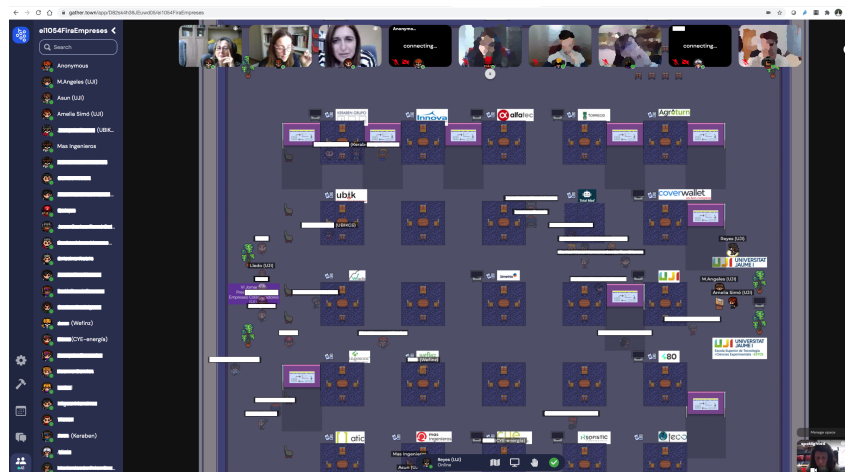


Figura 4: Entorno virtual de la feria desarrollado en la plataforma gather.town.

posibilitaba comunicarse entre personas específicas o entre todos los asistentes. Las coordinadoras podían dar voz para hacer preguntas generales, o incluso invalidar la entrada de alguien que manifestase un mal comportamiento.

- Establecer relaciones interpersonales más naturales mediante el uso de avatares, que se mueven libremente por el entorno e interactúan con aquellos que tienen cerca. Para facilitar la identificación, cada estudiante usó su nombre real y cada profesional/profesor añadió el nombre de su empresa/universidad tras su nombre. Además, la aplicación permite ver a la persona real a través de una pantalla, por lo que se pidió a todos los participantes tener la cámara activada.
- Compartir información con pequeños grupos, que se encuentren en la distancia configurada, o con todos los asistentes.

Para entrar en la feria se creó un contraseña que se distribuyó tanto a las empresas como a los estudiantes. Se elaboró un vídeo tutorial para el alumnado³ explicando cómo utilizar el entorno virtual y las posibilidades que este les ofrecía. Unos días antes del evento, se realizó una prueba con las empresas participantes, lo que permitió localizar pequeños problemas de configuración que se solventaron antes del evento y explicar el funcionamiento de la plataforma a las empresas.

Durante la feria virtual participaron 38 alumnos, cinco profesoras, 18 empresas y dos técnicas de la OIPEP. La jornada transcurrió con fluidez durante más de dos horas, en un ambiente profesional y distendido. A pesar de ser un entorno virtual, al igual que ocurría en la feria presencial, los alumnos se mostraron tímidos para acercarse a los stands, por lo que las coordinadoras les animaron a acercarse, dinamizando con ello la

interacción en la feria.

Tras la realización de la jornada se realizaron dos encuestas, una al alumnado y otra a las empresas, cuyos resultados se analizan a continuación.

5. Análisis de la experiencia

La 'Jornada de Presentación de las Empresas Colaboradoras' se viene realizando desde el curso 2015/2016, sin embargo tan solo en las dos últimas ediciones se han recogido datos de satisfacción de los participantes de manera formal mediante una encuesta anónima. En las anteriores ediciones la retroalimentación sobre la experiencia se realizó a un nivel más informal, dada la cercanía existente en la asignatura con el alumnado y con las empresas.

En la feria presencial (ver Sección 4.2) del curso 2019/2020 participaron 23 empresas. La encuesta se pasó tanto al alumnado como a las empresas obteniendo un elevado grado de participación. Analizando las respuestas del alumnado a un 95,7 % le resultó interesante la feria, un 100 % indicaba que se debía seguir realizando y tan solo un 13 % creía que se debía cambiar algún aspecto en el formato de la feria. Más de la mitad del alumnado que respondió, un 52,2 %, estaba totalmente satisfecho con la jornada y un 39,2 % estaba satisfecho, tal como puede observarse en la Figura 5.

Respecto a los principales comentarios del alumnado cabe destacar las siguientes propuestas de mejora:

- Enviar información previa al alumnado sobre las empresas asistentes para que estos puedan investigar sobre ellas y determinar cuáles les interesa más visitar el día de la jornada.
- Ofrecer información sobre los proyectos que propone cada empresa por itinerario antes o durante la feria.

³<https://www.youtube.com/watch?v=ecf20IFg0gE>

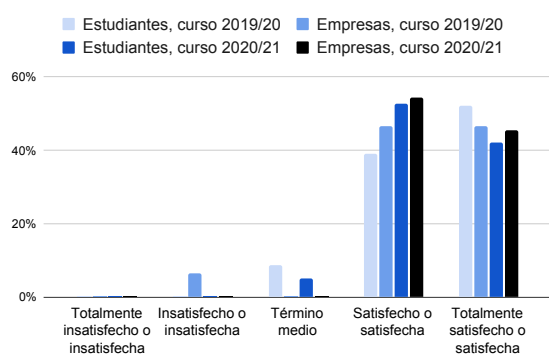


Figura 5: Satisfacción del alumnado y de las empresas con las ferias de los cursos 2019/2020 y 2020/2021.

- Adecuar las propuestas/ideas que lanzan las empresas a la carga de trabajo que tiene la asignatura.
- Disponer de un cartel uniforme para cada empresa en el que se indique de forma clara y concisa a qué se dedican, tipo de empresa, tecnologías que usan, itinerarios de sus ofertas, etc.
- Incrementar el espacio disponible para cada empresa con la finalidad de mejorar la movilidad y dotar a todas las empresas del mismo espacio y visibilidad.
- Invitar a antiguos alumnos que trabajan en las empresas participantes para que puedan ofrecer sus consejos.
- Motivar al alumnado para que se acerque a las empresas.
- Incentivar la participación de informáticos/as que puedan explicar con detalle la parte técnica de las propuestas, puesto que en algunas ocasiones solo acude personal de recursos humanos.

En lo referente a las empresas, un 46,7 % estuvieron totalmente satisfechas con el formato y en el mismo porcentaje estaban satisfechas (ver Figura 5). Los comentarios de mejora aportados por las empresas iban en la misma dirección que los del alumnado: aumentar el espacio reservado a cada empresa, mejorar el envío de información antes del evento pero en este caso del alumnado, etc. Además, comentaron que se trataba de una iniciativa interesante que es necesario seguir llevando a cabo e importante para fomentar el contacto entre el alumnado y el mundo empresarial. También se mostraron agradecidos por poder participar.

En la feria virtual (ver Sección 4.3) del curso 2020/2021 participaron 18 empresas. La encuesta se pasó, como en el curso anterior, al alumnado y a las empresas participantes, obteniendo un elevado número de respuestas y comentarios. Debido al cambio de formato en la feria, que pasó de ser presencial a virtual, se añadieron un par de preguntas sobre esta cuestión.

En la Figura 5 se puede observar cómo a pesar de

haber cambiado a formato virtual la satisfacción del alumnado con el evento continuó siendo muy alta, aunque en esta ocasión el porcentaje que estaba satisfecho, un 52,6 %, era ligeramente superior al porcentaje que lo estaba completamente, un 42,1 %. Este hecho podría deberse a que siempre es más recomendable una experiencia presencial para este tipo de actos. Respecto al resto de preguntas el 100 % del alumnado era partidario de seguir realizando la feria y la consideraban interesante, aunque un 72,2 % hubiera preferido el formato presencial, lo cual explica el grado de satisfacción obtenido para la feria virtual. Por otra parte, a un 97,4 % le pareció adecuada la plataforma elegida para la realización de la feria virtual.

En relación a los comentarios aportados por el alumnado sobre la feria virtual cabe destacar los siguientes aspectos negativos:

- No tener información previa sobre las empresas, los proyectos que ofrecen y los itinerarios a los que se adscriben estos.
- No disponer de un número de empresas equitativo para los distintos itinerarios del grado.
- Requiere aprender cómo funciona la plataforma.
- Algún problema técnico leve con el audio.
- Desorientación al principio por no saber cómo actuar, si ir en grupo o de manera individual, o por no saber qué empresas son de cada itinerario.
- No siempre era posible encontrar a los representantes de las empresas en sus stands.
- Falta de tiempo.
- Perder el primer contacto presencial con la empresa, que consideran que es importante.
- Interacciones más distantes, la charla con el personal de la empresa sería mejor si fuera de forma presencial.
- Dificultad para mantener una conversación en línea, por ejemplo, al ser interrumpida cuando entra alguien en el espacio cercano de escucha.

Y como aspectos positivos indicaron de manera literal lo siguiente:

- La plataforma virtual elegida está muy bien.
- El método utilizado fue muy bueno e hizo que todo fuera muy cercano y dinámico.
- El formato virtual fue perfecto, divertido y curioso, además de beneficioso para el alumnado.
- La no presencialidad ayuda a que más alumnos se animen a asistir.
- Mayor seguridad debido al riesgo que supone la pandemia por COVID-19 al hacerlo en línea, pero en una situación normal sería mejor el formato presencial.
- Lo mejor es poder asistir desde casa y sin moverse interactuar con las empresas, así como poder buscar información al mismo tiempo.

- Se puede consultar la información de cada empresa y luego contactar o no con el personal de la misma solo si te interesa.
- Se puede entablar una charla más privada con los representantes de cada empresa para comprender mejor en qué trabajan y determinar si te interesa.
- El tener un primer contacto con las empresas y saber qué empresas están interesadas en contratar alumnado y tener la posibilidad de establecer una práctica nominativa.

Finalmente, las empresas opinaron en un 100 % que se debía seguir realizando la feria y que la plataforma elegida para el formato virtual era adecuada, mientras un 90,9 % la considera interesante y un 54,5 % hubiera preferido que fuera presencial. Este es el primer aspecto en el que difieren alumnos y empresas, puesto que los alumnos eran más partidarios del formato presencial que las empresas, un 72,2 % frente a un 54,5 %. Por otro lado, el nivel de satisfacción de las empresas es muy similar al del alumnado, tal como se puede observar en la Figura 5.

Respecto a los comentarios de las empresas cabe destacar la valoración positiva general que hacen de la iniciativa que permite acercar el mundo empresarial al alumnado y viceversa. Algún asistente comentó que el formato le pareció mucho mejor puesto que le permitía asistir y tener contacto con el alumnado, sin descuidar por mucho tiempo las tareas de su empresa. Otro participante, aunque tenía reticencias el día de la prueba, nos felicitó al acabar la jornada porque había ido mucho mejor de lo esperado e incluso había podido realizar actividades que hacían en la feria presencial.

6. Lecciones aprendidas

Las principales *lecciones aprendidas* de las experiencias docentes llevadas a cabo en los diferentes formatos son las siguientes, agrupadas desde diferentes puntos de vista:

1. *Información antes de la feria*: es necesario proporcionar con suficiente antelación información completa sobre las empresas asistentes, así como del itinerario en el que suelen encajar las propuestas que realizan, con la finalidad de que el alumnado esté informado y pueda optimizar su tiempo durante la feria.
2. *Información durante la feria*: es conveniente mostrar en un panel información relevante de la empresa y de su itinerario, así como de los proyectos propuestos por cada empresa.
3. *Proyectos ofrecidos*: se debería presentar la propuesta de proyecto ya definida por parte de las empresas durante la feria, y no dar una idea vaga del proyecto, así como ofrecer proyectos que

encajen con las 300 horas presenciales que debe realizar el alumnado.

4. *Organización física de la feria*: es conveniente disponer de espacios amplios y agradables que favorezcan la comunicación oral y otorgar a todas las empresas un espacio equitativo para que tengan la misma visibilidad por parte del alumnado. También sería aconsejable que el personal de las empresas que acuda tenga un perfil técnico.
5. *Organización virtual de la feria*: es necesario realizar una sesión de prueba con las empresas para poder solucionar posibles problemas técnicos y proporcionar un vídeo tutorial al alumnado para que practique con anterioridad.
6. *Motivación del alumnado*: es interesante, tanto en el formato presencial como en el virtual, acompañar al alumnado que se vea un poco perdido. Como a veces a ciertos alumnos les cuesta entablar la conversación con las empresas, es conveniente que el profesorado que ya conoce al alumnado ejerza de facilitador en estas situaciones.

Todos los aspectos recogidos en las lecciones aprendidas han intentado ser mejorados en las sucesivas ediciones de la experiencia, si bien los siguientes puntos los consideramos cruciales y los dejamos como cuestiones abiertas. Como se puede observar estos aspectos aparecen de forma recurrente tanto en el formato presencial como en el virtual.

- Determinar el itinerario de cada empresa es complicado porque una empresa puede proponer proyectos de diversos itinerarios. Incluso en ocasiones hay proyectos que pertenecen a más de un itinerario. Por tanto, catalogar una empresa dentro de un itinerario como desearía el alumnado, salvo raras excepciones, no es posible. Sin embargo, en próximas ediciones se tratará de indicar los itinerarios en los que cada empresa suele ofertar, así como proporcionar información de proyectos por itinerario.
- La presentación de los proyectos ya definidos por parte de las empresas durante esta jornada es una de las aspiraciones iniciales de las coordinadoras que resulta muy difícil conseguir. Las empresas son reacias a presentar sus proyectos ante otras empresas, aunque más tarde sí que envían la propuesta para que sea publicada para el alumnado. Se sigue trabajando en una idea que solucione este problema.
- Las habilidades sociales es un tema que se presenta como un obstáculo en la feria presencial para el alumnado que no sabe cómo actuar antes las empresas y necesita ayuda para establecer un primer contacto. Esta dificultad se acusa aún más en el formato virtual puesto que, además de no saber cómo establecer el primer contacto, to-

dos los participantes deben trabajar sus competencias sociales en línea. Cuestiones tan básicas como acercarse a un stand de forma presencial y no interrumpir una conversación, es más difícil en el formato en línea. Y por lo tanto, requiere cierto entrenamiento por parte de los participantes. Si bien este tipo de competencias sociales en línea se ha incrementado en herramientas de videoconferencia debido a la pandemia, es necesario mayor nivel de prácticas aún en herramientas tipo `gather.town`.

Como conclusión, el artículo presenta una experiencia docente en la asignatura del grado en Ingeniería Informática que integra las prácticas externas y el TFG. En la evolución de la experiencia se pone de manifiesto la importancia que adquiere el contacto entre el alumnado y las empresas a la hora de coordinar una asignatura que integre ambas partes.

La satisfacción de ambos colectivos ha ido aumentando al ir mejorando la experiencia en los sucesivos cursos. Debido a la pandemia del COVID-19 la feria se ha tenido que realizar en formato virtual. Los resultados en las encuestas sobre la feria virtual, aún siendo ligeramente inferiores en el caso del alumnado respecto del año anterior, confirman el acierto de realizarla.

Finalmente, la experiencia también pone de manifiesto que, para mejorar el propio proceso de mejora continua en la asignatura, es una buena práctica recoger la retroalimentación de alumnado y empresas a través de una encuesta anónima.

Referencias

- [1] Óscar Belmonte, Mercedes Segarra, Reyes Grangel, and Sergio Aguado. Desde la Iniciativa Empresarial hacia el éxito pasando por Metodologías Ágiles e Ingeniería del Software. In *Actas de las XXII Jornadas de Enseñanza Universitaria de Informática, Jenui 2016*, 07 2016.
- [2] C.W. Dawson and G. Martín. *El proyecto fin de carrera en Ingeniería Informática: una guía para el estudiante*. Pearson Educación, 2002.
- [3] César Domínguez, Arturo Jaime, Francisco José García-Izquierdo, José Miguel Blanco, and Ana Sánchez. Carga de trabajo de los alumnos en prácticas en empresa y asignaturas simultáneas. In *Actas de las XXIV Jornadas de Enseñanza Universitaria de Informática, Jenui 2018*, 07 2018.
- [4] Jay Goldberg, Vikram Cariapa, George Corliss, and Kate Kaiser. The benefits of industry involvement in the multidisciplinary capstone design course at marquette university. *International Journal of Engineering Education*, 30, 01 2014.
- [5] Javier J. Gutiérrez, Isabel Ramos, Irene Barba, Manuel Mejías, and Javier Aroba Páez. Actividades y resultados del plan de innovación docente para la participación de empresas en la docencia. In *Actas de las XXIII Jornadas de Enseñanza Universitaria de Informática, Jenui 2017*, 07 2017.
- [6] Arturo Jaime, César Domínguez, Juan Larrea, Francisco García-Izquierdo, and Alicia Pérez-Albéniz. Luces y sombras de las estancias en empresa, impulsoras de los trabajos de fin de grado. In *Actas de las XXV Jornadas de Enseñanza Universitaria de Informática, Jenui 2019*, 07 2019.
- [7] M. Osane Lizarralde Urrutia, Txema Pérez Lázare, and Felix Larrinaga Barrenechea. La importancia de la relación empresa-universidad en la formación del alumno. In *Actas de las XXIV Jornadas de Enseñanza Universitaria de Informática, Jenui 2018*, 07 2018.
- [8] Carlos López Nozal, Raúl Marticorena Sánchez, Juan José Rodríguez, and Andrés Bustillo Iglesias. Proceso de gestión de trabajos fin de carrera. In *Actas de las XV Jornadas de Enseñanza Universitaria de Informática, Jenui 2009*, 07 2009.
- [9] IEEE/ACM Joint Task Force on Computing Curricula. Software engineering 2004 (se2004), curriculum guidelines for under-graduate degree programs in software engineering, 2004.
- [10] David G. Rosado, Ismael Caballero, José A. Cruz-Lemus, Manuel Serrano, Luis E. Sánchez, and Eduardo Fernández-Medina. Lecciones aprendidas tras varias convocatorias de un programa para el fortalecimiento y mejora de las competencias de empleabilidad de los graduados en informática. In *Actas de las XXII Jornadas de Enseñanza Universitaria de Informática, Jenui 2016*, 2016.
- [11] Aaron Steele and Sandra Cleland. ICT capstone projects and internships: Analysis of work environment characteristics. In *Actas de CITRENZ*, pages 101–106, 10 2015.
- [12] Fermín Sánchez Carracedo, Jaume Moral Ros, and David López Álvarez. Mis estudiantes acaban el TFG (todos) en el tiempo previsto (casi todos). In *Actas de las XXI Jornadas de Enseñanza Universitaria de Informática, Jenui 2015*, 07 2015.
- [13] Ángeles López, Lledó Museros, María José Aramburu, and Pedro García-Sevilla. Integración de las prácticas externas y el trabajo fin de grado en ingeniería informática. *ReVisión*, 11(1), 2018.

Generación automática de preguntas cloze para cuestionarios Moodle sobre análisis léxico

Roberto Izquierdo Amo
Media & Entertainment Innovation.
Telefónica I+D
Rda. Comunicación, s/n, 28050, Madrid
roberto.izquierdoamo@telefonica.com

César I. García Osorio
Depto. de Ing. Informática.
Universidad de Burgos
Avda. Cantabria, s/n, 09006, Burgos
cgosorio@ubu.es

Pedro Latorre Carmona
Depto. de Ing. Informática.
Universidad de Burgos
Avda. Cantabria, s/n, 09006, Burgos
plcarmona@ubu.es

José A. Barbero Aparicio
Depto. de Ing. Informática.
Universidad de Burgos
Avda. Cantabria, S/N, 09006, Burgos
jabarbero@ubu.es

Resumen

Los cuestionarios de Moodle son una manera conveniente de afrontar la evaluación online, tan relevante en época de pandemia. Desafortunadamente, su preparación es lenta, engorrosa y propensa a fallos. Además de diseñar la pregunta, luego hay que crearla en Moodle. Si tiene imágenes, hay que crearlas y subirlas a Moodle. También hay que configurar la calificación asociada a la respuesta correcta y a las opciones alternativas incorrectas. Si este proceso se tiene que repetir con muchas preguntas, el tiempo necesario termina siendo prohibitivo. En este trabajo se presenta una herramienta que resuelve todos estos problemas permitiendo la generación automática y masiva de preguntas para la evaluación de algoritmos de obtención de autómatas finitos a partir de expresiones regulares. La evaluación de estos algoritmos es relevante tanto en las asignaturas de lenguajes formales como en las de diseño de compiladores. Con un algoritmo genético se buscan ejercicios de complejidad configurable, tras lo cual se genera el texto, tablas e imágenes asociadas para la carga masiva de todas las preguntas generadas. El principal impacto del uso de esta herramienta está en la enorme reducción del tiempo en la preparación de preguntas, sustituyendo cientos o miles de interacciones con los formularios de Moodle, por unos pocos pasos.

Abstract

Moodle questionnaires are a convenient way to face the online assessment, so relevant in these pandemic times. Unfortunately, their preparation is slow, cumbersome, and prone to mistakes. In addition to designing the question, then it has to be introduced in Moodle. If

the question has images, it is necessary to create and upload them to Moodle. It is also necessary to configure the grades associated with the correct answer and the incorrect alternative options. If this process has to be repeated with many questions, the time required ends up being prohibitive. In this work, a tool is presented that solves all these problems, allowing the automatic and massive generation of questions for the evaluation of algorithms for obtaining finite automata from regular expressions. The evaluation of these algorithms is relevant in both formal languages and compiler design subjects. With a genetic algorithm, exercises of configurable complexity are searched, after which the text, tables and associated images are generated for the massive loading of all the questions generated. The main impact of the use of this tool is in the enormous reduction of the time in the preparation of questions, replacing hundreds or thousands of interactions with Moodle forms, for a few steps.

Palabras clave

Moodle, preguntas cloze, cuestionarios automáticos, autómatas finitos, expresiones regulares, algoritmos genéticos.

1. Introducción

Muchas universidades españolas han adoptado Moodle como plataforma de gestión de aprendizaje. Moodle, además de poder utilizarse como repositorio de materiales de asignaturas (con la ventaja adicional de que es posible controlar el acceso que se hace a estos ma-

teriales), ofrece otras funcionalidades. Por ejemplo, la realización de cuestionarios.

Los cuestionarios Moodle son una forma adecuada de evaluar los conocimientos y competencias adquiridos por los alumnos [11]. Se pueden utilizar tanto en la evaluación continua como para la evaluación final de las asignaturas. Moodle ofrece un amplio abanico de tipos de preguntas. Desde las más sencillas, como las de respuesta corta, de cierto o falso, de respuesta múltiple, numéricas o de emparejamiento, a las más complejas y sofisticadas, como las calculadas, las de pinchar y arrastrar texto o imágenes, o las preguntas de tipo cloze. Estas son las más flexibles, permitiendo combinar en una misma pregunta varias opciones de respuesta, como rellenar blancos, preguntas numéricas, selección múltiple de una única respuesta, o selección múltiple de varias respuestas.

La elaboración de los cuestionarios de Moodle puede hacerse de varias formas. La primera, consiste en utilizar los propios formularios proporcionados por Moodle. Pero el proceso es lento y tedioso, y tiene que hacerse pregunta a pregunta. La segunda, permite la carga simultánea de varias preguntas, pero obliga a aprender una sintaxis específica, lo que para muchos docentes puede suponer también un problema. Una tercera alternativa es el uso de herramientas externas a Moodle.

En este artículo se presenta una de estas herramientas, la cual está orientada a la generación de preguntas cloze para alumnos de asignaturas sobre diseño de compiladores o sobre lenguajes formales.

En concreto, facilita enormemente la generación de preguntas sobre los algoritmos de obtención de autómatas finitos a partir de expresiones regulares. Lo único que hay que hacer es introducir la expresión regular y elegir el formato de pregunta de entre un conjunto de plantillas predefinido. No sólo genera el texto de la pregunta, sino también las distintas alternativas de respuesta, y para algunos formatos de pregunta la imagen de los autómatas o árboles de análisis a los que se hace referencia en el texto de la pregunta. Es más, mediante la utilización de algoritmos genéticos, puede generar de forma automatizada el número requerido de preguntas. Lo único que hay que introducir son algunos parámetros sobre la complejidad que se desea tengan las preguntas generadas. Los cuestionarios *cloze* generados por PLQuiz son del tipo particular de preguntas en las que hay que elegir la respuesta correcta entre un conjunto propuesto de alternativas.

El resto del artículo se estructura como sigue: en la Sección 2, para enmarcar el trabajo, se realiza una breve descripción de los conceptos de expresión regular, autómata finito y los algoritmos que permiten obtener los segundos a partir de los primeros. En la Sección 3, se enumeran y describen brevemente algunas otras he-

rramientas que también trabajan con estos conceptos. En la Sección 4, se describe la herramienta PLQuiz, algunos detalles de implementación y se explica su uso mostrando los principales tipos de preguntas generadas. Se finaliza con las conclusiones en la Sección 5.

2. Antecedentes teóricos

Las asignaturas de lenguajes formales o de compiladores generalmente se imparten en el segundo o tercer curso de los grados en informática. Sus contenidos varían entre universidades, pero generalmente comprenden lo siguiente:

- Lenguajes de programación: definición formal y base del diseño del lenguaje.
- Especificación formal de lenguajes: los conceptos esenciales de alfabetos, símbolos, lenguajes formales, etc., así como el uso de expresiones regulares y gramáticas.
- Análisis léxico: generalmente con expresiones regulares y herramientas específicas para la construcción de *lexer* (también conocidos como escáneres o *tokenizadores*).
- Análisis sintáctico: análisis de arriba hacia abajo y de abajo hacia arriba, generalmente basado en herramientas específicas para la construcción de analizadores como JavaCC¹, Bison², PLY³, ANTLR⁴, etc.

Los primeros contenidos son comunes tanto en las asignaturas sobre autómatas y lenguajes formales como en las de construcción de compiladores y es para estos contenidos para los que la herramienta presentada aquí puede generar cuestionarios de forma automática. En concreto, para evaluar los conocimientos sobre los algoritmos que permiten obtener autómatas finitos a partir de expresiones regulares. En la sección que sigue se da una breve introducción a estos conceptos y algoritmos.

2.1. Expresiones regulares y autómatas finitos

Las expresiones regulares (ER) son un mecanismo para describir tipos especiales de lenguajes formales, conocidos como lenguajes regulares. Utilizando elementos del alfabeto del lenguaje y los operadores de concatenación («·», que se suele omitir), de selección («|») y de repetición («*») pueden describirse las cadenas pertenecientes al lenguaje. Por ejemplo, la expresión regular $a \cdot (a | b)^* \cdot b$ representaría el conjun-

¹<https://javacc.org>

²<https://www.gnu.org/software/bison/>

³<http://www.dabeaz.com/ply/>

⁴<http://www.antlr.org/>

to de cadenas de *aes* y *bes* que empiezan por *a* y finalizan por *b*.

En el contexto de la construcción de compiladores, las expresiones regulares se utilizan para definir los componentes léxicos que se reconocen en la fase de análisis léxico: identificadores, números en coma flotante, cadenas, etc.

Dada una ER puede construirse un autómata finito (AF) capaz de reconocer las cadenas descritas por la ER. El AF comienza el proceso de reconocimiento en el *estado inicial*, una *función de transición* define cómo, a partir de unos estados puede llegarse a otros. Si la función de transición solo permite que el AF esté en un único estado activo, se trataría de un autómata finito determinista (AFD). Si permite estar en varios⁵, sería un autómata finito no determinista (AFND).

Existen algoritmos que permiten la construcción automática de autómatas a partir de una expresión regular. Los implementados en PLQuiz se describen en la siguiente sección.

2.2. Construcción de autómatas a partir de expresiones

Hay varias formas de obtener un AF a partir de una ER. En PLQuiz se generan diferentes tipos de cuestionarios para evaluar el conocimiento sobre los siguientes algoritmos:

- El algoritmo propuesto por McNaughton, Yamada y Thompson [13], que construye un AFND a partir de una expresión regular, analizando de forma recursiva sus partes constituyentes (sub-expresiones). El algoritmo crea un AFND para cada sub-expresión, y los une usando transiciones épsilon teniendo en cuenta el operador que las combina.
- El algoritmo de construcción de subconjuntos [16], que permite transformar un AFND en un AFD. Dado que el AFND tiene un número finito de estados, también serán finitas las posibles combinaciones de estados activos (de hecho, algunas de estas combinaciones no se darán en la práctica, al impedir la función de transición que se alcanzen), la idea es asociar un estado del AFD con cada una de estas combinaciones y construir una función de transición que refleje como cambian estas combinaciones en el AFND de partida.
- Por último, PLQuiz implementa el algoritmo descrito por Aho, Sethi y Ullman en [1], que utiliza la representación como árbol sintáctico de la expresión regular para calcular tres funciones: *anula-*

ble, *primera-pos* y *última-pos*, que permiten construir una función adicional, *siguiente-pos*, a partir de la cuál puede construirse finalmente la función de transición del AFD que reconoce el lenguaje representado por la ER de partida.

Para profundizar en los conceptos de ER, AF y algoritmos de obtención de AF a partir de ER pueden consultarse los trabajos de Linz [10] y Meduna [14].

3. Trabajos relacionados

La complejidad con la que los alumnos perciben los conocimientos sobre procesadores de lenguajes y sobre autómatas finitos y lenguajes formales ha motivado que muchos docentes hayan desarrollado aplicaciones y simuladores para dar un enfoque más práctico al aprendizaje de dichos conocimientos. Nos restringimos aquí a aquellas que se centran en los aspectos relacionados con el análisis léxico, las ER y los AF. En este grupo, la herramienta que podría considerarse la decana es JFLAP [17]. Está escrita en Java y es muy popular y completa. Permite la construcción y visualización no solo de autómatas finitos, sino de otros dispositivos reconocedores, como los autómatas de pila y las máquinas de Turing. También permite la obtención de un AF a partir de una ER, aunque para esto solo proporciona un algoritmo, que no se corresponde con ninguno de los implementados en PLQuiz. THOTH [7, 8] es otra herramienta, que como JFLAP, permite la creación gráfica de AF e implementa un conjunto más amplio de algoritmos de obtención de AF a partir de ER. Proporciona los algoritmos de Aho-Sethi-Ullman, de McNaughton-Yamada-Thompson y de construcción de subconjuntos, introducidos en la sección previa, además de un tercer algoritmo basado en la utilización de derivadas de las expresiones regulares⁶.

Las dos herramientas previas son herramientas de escritorio, pero también existen herramientas que pueden utilizarse sin más que tener un navegador, al ser aplicaciones web. Es el caso de *SELFA-Pro* [5], una herramienta que basa su funcionamiento en el uso de un lenguaje específico del dominio para describir expresiones regulares, autómatas, gramáticas y operaciones sobre los mismos. También es el caso de *Seshat* [2] que permite la visualización paso a paso de algunos de los algoritmos de obtención de AF a partir de ER. *Seshat* utiliza un diseño web adaptativo que permite su utilización también a través de dispositivos móviles. De hecho, las aplicaciones más recientes están dirigidas

⁵Porque de un estado el AF puede evolucionar a varios otros con el mismo símbolo, o porque tiene *transiciones épsilon*, que le permiten cambiar de estado sin necesidad de consumir símbolos de entrada.

⁶Dada una expresión regular, se dice que su derivada respecto a un carácter dado es la expresión regular que describe el lenguaje resultante de eliminar el primer carácter en el subconjunto de todas las cadenas que empiezan por ese carácter en el conjunto definido por la expresión regular de partida [3].

directamente a su uso en móviles. Es el caso de *CM-Simulator* [4] y de *Automata Simulator* [19], ambas orientadas a la construcción, simulación y manipulación de autómatas finitos y no tanto a su obtención a partir de expresiones regulares.

Las anteriores son sólo algunas de las aplicaciones representativas de las disponibles para facilitar el aprendizaje de conocimientos sobre lenguajes formales, ninguna de ellas tiene funcionalidad de generación de cuestionarios. En [20] puede encontrarse una revisión más amplia que incluye también herramientas que cubren otros aspectos de la construcción de compiladores, como los algoritmos de análisis sintáctico. Pero tampoco en esa revisión aparecen herramientas que como PLQuiz estén diseñadas para la generación de automática de cuestionarios.

También existen herramientas y bibliotecas que buscan facilitar la creación de cuestionarios en Moodle, aunque no siempre están bien mantenidas. De momento, no parecen haberse popularizado demasiado y su uso está restringido a los pocos usuarios que conocen de su existencia. Algunos ejemplos serían:

- *Libre-gift*⁷, una plantilla de LibreOffice para la creación de cuestionarios en este procesador de texto que luego son exportados a formatos importables desde Moodle.
- *MoodleQUIZ_template_UVa*⁸, otra plantilla, en este caso para Word, que facilita la edición de preguntas que luego son exportadas a GIFT, formato que luego puede usarse para importar las preguntas a Moodle.
- *Moodle Cloze and GIFT Code Generator* [21], una macro que permite editar preguntas usando Excel para exportarlas luego al formato GIFT⁹.
- *GIFT Quiz Editor*¹⁰, un complemento para los formularios Google que permite su exportación a GIFT.
- *Moodle-questions*¹¹, una biblioteca para la manipulación de cuestionarios Moodle en Python.
- *Pygiftgenerator*¹², otra biblioteca de Python, esta avalada por una publicación en revista [18], para la generación automática de problemas de física.
- *GIFTGenerator*¹³, una herramienta comercial para la generación de cuestionarios.
- *Typeform*¹⁴, otra herramienta comercial para facilitar la creación de cuestionarios.

⁷<https://github.com/clopezno/libre-gift>

⁸https://github.com/juacas/MoodleQUIZ_template_UVa

⁹<https://hbubecc.wixsite.com/jordan/tools>

¹⁰<https://bit.ly/2Zaohsc>

¹¹<https://github.com/gethvi/moodle-questions>

¹²<https://gitlab.com/EHU/pygiftgenerator>

¹³<https://www.giftgeneratorapp.com/>

¹⁴<https://www.typeform.com/quizzes/>

- *R/Exams*¹⁵, un paquete para el sistema R enfocado a la generación automática de exámenes que pueden importarse a Moodle. Su desarrollo es bastante activo y cuenta con una importante base de usuarios.

En todo caso ninguna de estas bibliotecas está centrada en la generación de cuestionarios para la evaluación de algoritmos de obtención de autómatas finitos a partir de expresiones regulares¹⁶. De hecho, a nuestro leal saber y entender, PLQuiz es la única herramienta existente con esta finalidad.

4. La nueva herramienta

El código de la aplicación se ha diseñado para que sea modular y reusable, agrupándolo en paquetes en los que las dependencias se han reducido usando el patrón de diseño *fachada* (*facade pattern*) [6]. Cada paquete principal tiene una estructura de dos niveles: el primer nivel contiene las clases *fachada*, visibles al resto de la aplicación, mientras que el segundo contiene la lógica y estructuras de datos internas, utilizadas solo por el paquete padre. Aunque el lenguaje principal en la que está escrita es Java, también se utilizan otros, como XML para la generación de los cuestionarios Moodle, o LaTeX y TikZ para la obtención de ejercicios para exámenes escritos.

La aplicación se encuentra disponible en Github (<https://github.com/RobertoI/PLQuiz>).

4.1. Generación de expresiones regulares

Aunque las ER se especifican inicialmente como cadenas de caracteres, se representan internamente como árboles binarios. Como la generación de árboles es un tema muy estudiado en programación genética, esto nos permite utilizar directamente algoritmos existentes para la generación de nuevas expresiones.

El algoritmo concreto implementado es el del método «full» [9]. Se generan árboles binarios en los que cada nodo no hoja tiene 1 o 2 hijos, y en el que todas las hojas se encuentran a la misma profundidad. El algoritmo restringe las etiquetas que pueden usarse en cada nodo generado, en función de su profundidad. Los nodos símbolo solo se permiten a la profundidad máxima, mientras que un nodo cerradura se permite a cualquier profundidad, excepto la máxima. Se utilizan restricciones adicionales a la hora de generar los hijos de un nodo, en función del operador que se usa para el nodo (un nodo cerradura solo podrá tener un hijo,

¹⁵<http://www.r-exams.org>

¹⁶R/exam tiene una plantilla para generar ejercicios para evaluar la interpretación de un diagrama de autómata, <http://www.r-exams.org/templates/automaton/>, pero no hemos podido encontrar ninguna para evaluar algoritmos de obtención de AF a partir de ER.

mientras que los nodos concatenación o selección, podrán tener dos).

Con estas restricciones, ya es posible generar una población inicial de ER aleatorias. En el proceso selectivo de los mejores «individuos» se tienen en cuenta consideraciones adicionales: *i*) número de símbolos presentes en la ER (sin contar el que representa la palabra vacía, *ii*) número de estados en la función de transición del AF obtenido a partir de la ER, *iii*) presencia o ausencia del símbolo ε , el cual representa la palabra vacía.

Una vez disponemos de una población aleatoria inicial, el objetivo del algoritmo genético es la obtención de soluciones similares en su camino hacia el óptimo. En el campo de la programación genética, de donde obtenemos nuestros métodos, se considera que los operadores de mutación y reproducción tienden a producir resultados similares [12]. En este caso, utilizamos un operador de mutación, consistente en reemplazar un sub-árbol de la ER por uno generado aleatoriamente, por resultar más sencillo en cuanto a implementación.

Dado que el objetivo del generador es producir ER que resulten en problemas que no sean ni excesivamente complejos ni triviales, el espacio de búsqueda de los algoritmos puede limitarse de tal forma que produzca el mayor número de expresiones posible dentro de este rango. Una vez definida la complejidad deseada en función del número mínimo y máximo de símbolos y estados distintos en las expresiones resultado, se utiliza el método de generación de árboles descrito anteriormente para construir miles de árboles de distintas profundidades. De este modo se determina experimentalmente qué rango de profundidades de árbol produce mayor proporción de árboles de la complejidad deseada para cada tipo de problema. En este caso, se comprobó que para los problemas basados en Aho-Sethi-Ullman resultaba apropiada una profundidad de árbol entre 3 y 6, y para los problemas de McNaughton-Yamada-Thompson, era mejor una profundidad entre 1 y 5.

Trabajar con un conjunto de posibles soluciones bien definido nos permite comparar distintos métodos de generación de problemas de manera empírica, así como elegir en cada caso el método más adecuado al problema. Las pruebas realizadas permitieron comparar el algoritmo genético implementado, con una generación puramente aleatoria, usando el método de inicialización hasta obtener un resultado aceptable. Se comprobó que los problemas de McNaughton-Yamada-Thompson se generaban de manera más eficiente con el algoritmo genético completo, mientras que en los de tipo Aho-Sethi-Ullman resultaba más rápido utilizar una generación aleatoria. Una posible causa es que, para las restricciones fijadas, el espacio de búsqueda para problemas de Aho-Sethi-Ullman con-

tiene más potenciales soluciones, y por lo tanto, requiere menor carga de optimización.

4.2. Generación de documentos L^AT_EX

Aunque el objetivo principal de PLQuiz es la generación de cuestionarios Moodle, adicionalmente permite la obtención de documentos L^AT_EX de ejercicios, lo que facilita también la realización de exámenes escritos en los que poder incluir la evaluación de los algoritmos de obtención de AF. Los documentos L^AT_EX [15] usan la clase de documento exam¹⁷ e incluyen tanto los enunciados de los ejercicios como su solución. Lo único que hay que hacer para mostrar o no la solución es cambiar una de las opciones de la clase de documento. De esta forma, se puede obtener tanto el documento para el examen como el documento de solución para su publicación posterior. Los árboles y los diagramas de transición de los AF de los enunciados o de las soluciones se obtienen de forma automática generando código TikZ [22]. Al igual que con los cuestionarios de Moodle, el ahorro de trabajo para el profesor es importante, ya que la obtención manual de estos diagramas requiere de bastante tiempo. En la Figura 1 pueden verse dos ejemplos de estos ejercicios, en los que se muestra tanto el enunciado como la solución.

4.3. El uso de la herramienta

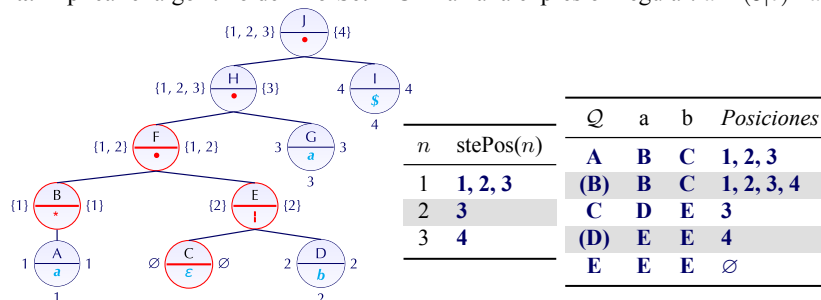
Las principales utilidades de la herramienta son: la generación de cuestionarios importables a Moodle, y la generación de funcionalidades que facilitan la creación y gestión de dichos cuestionarios. También permite obtener documentos L^AT_EX con los ejercicios generados.

La herramienta desarrollada permite añadir y eliminar preguntas. Durante el proceso de adición, puede elegirse el tipo de ejercicio para el que se generará dicha pregunta: bien el basado en obtener un AFND con McNaughton-Yamada-Thompson para a continuación utilizar construcción de subconjuntos, o la obtención directa de un AFD utilizando el algoritmo de Aho-Sethi-Ullman. Por otro lado, existe la posibilidad de añadir en una única operación bloques de múltiples problemas de cualquiera de los tipos (Figura 2), reduciendo la similitud de las preguntas a través de un rango de variación, y generando automáticamente tanto las ER como los diagramas y tablas resultado de aplicar los algoritmos de obtención de AF. En cuanto a la operación de eliminación, pueden elegirse los ejercicios concretos que quieren eliminarse o eliminar todos para comenzar la creación del cuestionario desde cero.

Cada tipo de pregunta (Aho-Sethi-Ullman o McNaughton-Yamada-Thompson) dispone de tres subtipos de ejercicios distintos, que se corresponden

¹⁷ <https://ctan.org/pkg/exam>

a.- Aplicar el algoritmo de Aho-Sethi-Ullman a la expresión regular: $a^* \cdot (\varepsilon|b) \cdot a$



b.- Completa la tabla de función de transición para el AFD que se obtendría de aplicar el método de construcción de subconjuntos al AFND de la figura

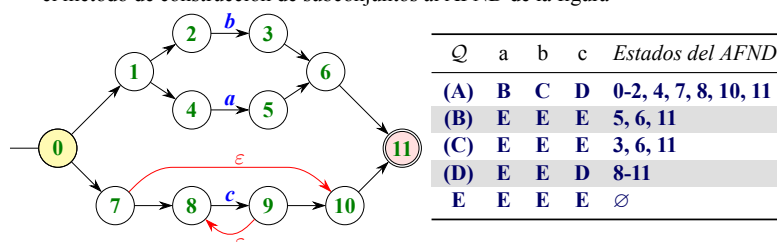


Figura 1: Ejemplos de ejercicios en L^AT_EX para inclusión en exámenes escritos: a) Aplicación de Aho-Sethi-Ullman para obtener un AFD a partir de una expresión regular, b) Aplicación de «Construcción de subconjuntos» para obtener un AFD a partir del diagrama de transición de un AFND.

con la aplicación de distintos algoritmos o pasos parciales de algoritmo. Estos son para los ejercicios de Aho-Sethi-Ullman: «construcción de árbol», «etiquetado de árbol» y «construcción de tablas», y para los ejercicios de McNaughton-Yamada-Thompson: «construir autómata», «resolver expresión» y «resolver autómata». Cuando se selecciona un subtipo, cambiará automáticamente el tipo de ejercicio mostrado, o bien el que se mostrará al generar el ejercicio, si no se ha introducido ninguna expresión.

A la hora de generar una pregunta, podemos definir, entre otras cosas: i) si la expresión regular asociada incorporará transiciones vacías (ε), ii) el número de símbolos contenidos en la *expresión regular*, iii) el número de estados en la tabla de transición del problema resuelto. A partir de este punto, se pueden generar expresiones, comenzando en ese momento la búsqueda de una expresión regular que produzca dicha pregunta.

Una vez configuradas las preguntas, pueden exportarse al formato *Moodle XML*, para poder ser importadas a continuación al banco de preguntas de Moodle. En el archivo generado se incluyen también todas las imágenes necesarias para la realización del ejercicio.

En las figuras 3 a 4 pueden verse algunos tipos de ejercicios, tal como aparecerían en Moodle. En estos ejercicios, los alumnos tienen que completar las tablas resultado de la aplicación del algoritmo, eligiendo para cada entrada de la tabla la opción correcta, de entre un conjunto de opciones disponibles en un desplega-

Figura 2: Diálogo para la generación en bloque de varias preguntas.

Figura 3: Tipo de ejercicio «Construcción de tablas» para el algoritmo de Aho-Sethi-Ullman.

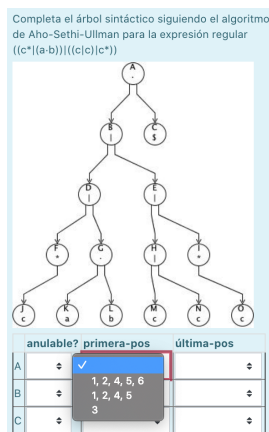


Figura 4: Tipo de ejercicio «Etiquetado de árbol» para la evaluación del algoritmo de Aho-Sethi-Ullman.

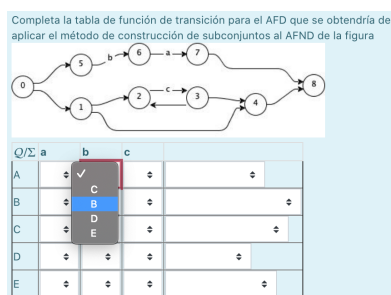


Figura 5: Tipo de ejercicio «Resolver autómatas» donde el alumno tiene que aplicar el algoritmo de construcción de subconjuntos.

ble. La realización a mano de este ejercicio implicaría tener que configurar, para cada una de las entradas de la tabla, todas las alternativas, lo que sería bastante tedioso y donde inevitablemente cometeríamos errores.

4.4. Valoración de la herramienta

Para aportar cierta perspectiva sobre la conveniencia de la herramienta, en esa sección se aporta la experiencia personal sobre el uso de la misma.

Desde hace tres cursos, los cuestionarios de evaluación continua de la asignatura «Procesadores de Lenguajes» incorporan ejercicios generados con PLQuiz. Desde nuestro punto de vista, la principal ventaja de PLQuiz es la facilidad de poder generar numerosos ejercicios de forma automática. Estos ejercicios pueden agruparse en categorías del banco de preguntas de Moodle que pueden utilizarse para añadir preguntas aleatorias a los cuestionarios. Dado que PLQuiz genera seis tipos de ejercicios distintos, puede tenerse una categoría para cada uno de ellos, permitiendo añadir hasta seis preguntas aleatorias diferentes a los cuestionarios. Este proceso de aleatorización minimiza el

riesgo de que algunos alumnos se sientan tentados a compartir las respuestas a los ejercicios, dado que con un número suficiente de ejercicios, las posibilidades de que a dos alumnos les toquen las mismas preguntas son realmente bajas. Además, dentro de cada una de estas categorías, tenemos subcategorías adicionales que vamos rotando cada curso, de modo que las preguntas de un curso no coincidan con las de los dos cursos previos.

Para tener una idea objetiva del ahorro de tiempo que supone la generación automática de preguntas con PLQuiz, analicemos lo que ocurre con el tipo de ejercicio «Construcción de árbol» para Aho-Sethi-Ullman. En este ejercicio al alumno se le muestra la expresión regular de partida, así como varios árboles entre los que tiene que elegir el que se corresponde con la expresión regular mostrada. Si quisiéramos introducir de forma manual esta pregunta utilizando los formularios de edición de preguntas en Moodle, este trabajo supondría realizar ¡hasta 50 interacciones! A esto habría que sumar el tiempo de preparar el dibujo de los cuatro árboles para poder subirlos en la parte de respuestas correcta y alternativas. Incluso siendo ágil en la creación de los árboles, esto puede suponer fácilmente al menos media hora para tener una pregunta. En el mismo tiempo, usando PLQuiz podemos generar y subir a Moodle decenas de preguntas listas para ser importadas aleatoriamente en un cuestionario.

Cuando la preparación de las preguntas lleva mucho tiempo, uno tiende a adoptar una actitud protectora con los ejercicios que se tienen preparados, intentando reservarlos para las pruebas de evaluación y con cierta reticencia a que puedan circular y ser compartidos entre los alumnos. Con PLQuiz no es necesario este secretismo, dado que es realmente sencillo generar nuevos ejercicios. Por ejemplo, el curso pasado, además del banco de preguntas para la evaluación continua, se generaron preguntas adicionales para cuestionarios de autoevaluación que los alumnos podían aprovechar para prepararse de cara al examen. Estos cuestionarios tuvieron muy buena acogida, a pesar de ser totalmente opcionales. De hecho, el 75 % de los alumnos decidieron realizarlos. Lamentablemente, no contamos con evidencias objetivas adicionales de la utilidad de estos cuestionarios, aunque en futuros cursos esperamos cruzar las calificaciones finales de los alumnos con la realización de estos cuestionarios, esperando encontrar alguna correlación positiva.

5. Conclusiones

Hemos presentado una herramienta capaz de generar cuestionarios de Moodle, de forma masiva (en gran número) y automática, para la evaluación de algoritmos de obtención de autómatas finitos a partir de expresio-

nes regulares. Los ejercicios pueden variar en complejidad, gracias a la aplicación de un algoritmo genético. Una vez definidos los ejercicios, puede generarse el texto, tablas e imágenes asociados. Estos ejercicios se pueden exportar para incorporar al banco de preguntas de Moodle, o a documentos $\text{\LaTeX}$ para su utilización en pruebas escritas. Además, las preguntas pueden utilizarse en otros LMS, dado que muchos ofrecen la posibilidad de importar preguntas en el formato de Moodle.

Esta herramienta puede tener un impacto importante tanto en las asignaturas de lenguajes formales como en las de diseño de compiladores, sobre todo por la reducción del tiempo dedicado a la preparación de preguntas.

Referencias

- [1] Alfred V Aho, Ravi Sethi, y Jeffrey D Ullman. Compilers, principles, techniques. *Addison Wesley*, 7(8):9, 1986.
- [2] Álvar Arnaiz-González, Jose-Francisco Díez-Pastor, Ismael Ramos-Pérez, y César García-Osorio. Seshat—a web-based educational resource for teaching the most common algorithms of lexical analysis. *Comput. Appl. Eng. Educ.*, 26(6):2255–2265, 2018.
- [3] Janusz A. Brzozowski. Derivatives of regular expressions. *J. ACM*, 11(4):481–494, octubre 1964.
- [4] Daniela Chuda, Jakub Trizna, y Peter Kratky. Android automata simulator. En *Proceedings of the International Conference on e-Learning*, páginas 80–4, 2015.
- [5] Jesús Gallardo Casero, José Jesús Castro Sánchez, y Raúl Miguel Sabariego. Experiencias de uso y evaluación de una herramienta de apoyo a la enseñanza de teoría de autómatas y lenguajes formales. En *Actas de las XXII JENUi*, páginas 327 – 334. Universidad de Almería, 2016.
- [6] Erich Gamma, Richard Helm, Ralph Johnson, y John M. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1ª edición, 1994.
- [7] César García-Osorio, Andrés Arnaiz-Moreno, y Álvar Arnaiz-González. THOTH: A new tool for automata theory learning. En *International Technology, Education and Development Conference. IATED*, Marzo 2007.
- [8] César García-Osorio, Iñigo Mediavilla-Sáiz, Javier Jimeno-Visitación, y Nicolás García-Pedrajas. Teaching push-down automata and turing machines. En *Proceedings of the 13th annual conference on Innovation and technology in computer science education*, páginas 316 – 316, 2008.
- [9] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.
- [10] Peter Linz. *An introduction to formal languages and automata*. Jones & Bartlett Publishers, 6ª edición, 2017.
- [11] José M López, Eva Romero, y Eva Roperio. Utilización de Moodle para el desarrollo y evaluación de competencias en los alumnos. *Formación universitaria*, 3(3):45–52, 2010.
- [12] Sean Luke y Lee Spector. A revised comparison of crossover and mutation in genetic programming. En *Proceedings of the Third Annual Conference on Genetic Programming 1998*, páginas 208–213, University of Wisconsin, Madison, Wisconsin, USA, 1998. Morgan Kaufmann.
- [13] Robert McNaughton y Hisao Yamada. Regular expressions and state graphs for automata. *IRE Trans Electron Comput*, 9(1):39–57, 1960.
- [14] Alexander Meduna. *Formal languages and computation: models and their applications*. CRC Press, 2014.
- [15] Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle, y Chris Rowley. *The L^A-TEX companion*. Addison-Wesley Professional, 2004.
- [16] Michael O Rabin y Dana Scott. Finite automata and their decision problems. *IBM J Res Dev*, 3(2):114–125, 1959.
- [17] Susan H Rodger y Thomas W Finley. *JFLAP: an interactive formal languages and automata package*. Jones & Bartlett Learning, 2006.
- [18] Jon Sáenz, Idoia G Gurtubay, Zunbeltz Izaola, y Gabriel A López. Pygiftgenerator: a python module designed to prepare moodle-based quizzes. *Eur J Phys*, 42(1):015702, 2020.
- [19] Tuhina Singh, Simra Afreen, Pinaki Chakraborty, Rashmi Raj, Savita Yadav, y Dipika Jain. Automata simulator: A mobile app to teach theory of computation. *Comput. Appl. Eng. Educ.*, 27(5):1064–1072, 2019.
- [20] Srećko Stamenković, Nenad Jovanović, y Pinaki Chakraborty. Evaluation of simulation systems suitable for teaching compiler construction courses. *Comput. Appl. Eng. Educ.*, 28(3):606–625, 2020.
- [21] Jordan Svien. Improvements to the moodle cloze and GIFT code generator. En *Proceedings of MoodleMoot Japan 2019 Annual Conference*, páginas 19–23, 2019.
- [22] Till Tantau. The TikZ and PGF packages (manual for version 3.0. 1a; 2015.08. 29). Technical report, Institut für Theoretische Informatik Universität zu Lübeck, 2015.

Hacia un modelo genérico de visualización de traductores dirigidos por la sintaxis

Ángel F. Sánchez-Granados, Jaime Urquiza-Fuentes,
Adrián García-Oller & José M. Loeches-Ruiz
LITE - Laboratorio de Tecnologías de la Información en la Educación
Universidad Rey Juan Carlos
28933 Móstoles (Madrid)
{angelfrancisco.sanchez, jaime.urquiza}@urjc.es,
{a.garciaoll, jm.loeches}@alumnos.urjc.es

Resumen

Este trabajo presenta una herramienta de visualización que pretende ayudar a comprender los conceptos de los traductores dirigidos por la sintaxis, proporcionando una representación gráfica de conceptos tan abstractos. La herramienta que se describe consta de una interfaz de visualización y una API. La interfaz de visualización permite a profesores y estudiantes ver una representación gráfica del proceso de ejecución del traductor dirigido por la sintaxis observando detalles como las acciones semánticas ejecutadas o los valores de los atributos de los símbolos. La API permite al usuario anotar una especificación del traductor dirigido por la sintaxis de forma que durante su ejecución genere una traza que permita su posterior visualización con la interfaz. Esta división entre interfaz y API busca que las visualizaciones puedan ser independientes del software utilizado para la generación del traductor, tratando así de sortear alguna de las limitaciones que tienen otras herramientas existentes.

Abstract

In this work, we present a visualization tool aimed at supporting students' comprehension of syntax directed translators concepts by providing graphical representations of such abstract concepts. The tool is made up of two parts: a visualization interface and an API. The visualization interface provide teachers and students with a graphical representation of the execution process of the syntax directed translator. This representation provide a detailed view of the executed semantic actions and the values of the symbol attributes. The API allows the user to annotate the specification of the syntax directed translator. Thus, its execution generates a trace that will be used by the interface to produce the visualization. This division between the interface

and the API supports the independence between the visualizations and the parser generator tool, avoiding some of the limitations identified in other existing tools.

Palabras clave

Visualización del software, procesadores de lenguajes traductores dirigidos por la sintaxis, enseñanza de la informática.

1. Introducción

La disciplina de la visualización del software aplicada al entorno educativo tiene un largo recorrido, sirva como ejemplo las animaciones de algoritmos de ordenación realizadas por Ronald Baecker en los años 60 [3]. Una de las posibles aplicaciones que podemos encontrar de esta especialidad es la visualización de conceptos abstractos, que nos permitirá conseguir que los estudiantes sean capaces de formar un mejor modelo mental del concepto a estudiar. Por otro lado, tanto Naps et al. [9] como Hundhausen et al. [5] concluyen que además de la visualización *per se* es determinante la forma en que esta se usa por parte de los estudiantes.

En la enseñanza de la Informática, la asignatura de Procesadores de Lenguajes resulta habitualmente compleja para los estudiantes debido en parte a la dificultad de aplicar a situaciones reales los conceptos abstractos que se trabajan [4]. Una de las ramas fundamentales que la componen es la traducción dirigida por la sintaxis (TDS), que se basa en conceptos de analizadores sintácticos enriquecidos con la posibilidad de producir información y transmitirla a través del árbol sintáctico enriquecido para ejecutar un lenguaje de programación [7], y que resulta de gran importancia a la par que complejo. La TDS proporciona, por tanto, la base teórica para entender algunas etapas elementales de los com-

piladores, como pueden ser la generación de código intermedio o la comprobación de tipos. Precisamente, la visualización pretende servir de enlace entre los conceptos teóricos y su aplicación práctica.

Este trabajo se centrará en la aplicación de la visualización del software a la TDS con el objetivo de que los estudiantes mejoren su comprensión. El resto del artículo se estructura como sigue. La sección 2 describe los trabajos relacionados sobre la visualización de Procesadores de Lenguajes. La herramienta se describe en la sección 3 y en la sección 4 se muestra un ejemplo de uso. La sección 5 menciona la evaluación preliminar de la herramienta. Finalmente, la sección 6 expone las conclusiones y los trabajos futuros.

2. Trabajos relacionados

El desarrollo de traductores dirigidos por la sintaxis (en adelante traductores) es una tarea compleja y por ello existen una gran variedad de herramientas diseñadas para apoyarla. Algunas se centran exclusivamente en el desarrollo, por ejemplo CUP¹ o ANTLR², aunque también se usan en entornos educativos. El esquema de funcionamiento básico consiste en que el desarrollador produce una especificación del traductor, las herramientas antes mencionadas se usan para generar el código fuente del traductor a partir de la especificación. Una vez generado dicho traductor, este se ejecutará como cualquier otra aplicación procesando la cadena de entrada y realizando las tareas incluidas en la especificación.

Por otra parte, también existe software relacionado donde la visualización asume un rol determinante, como los sistemas VAST, LISA, CUPV, VCOCO y EvDebugger.

VAST [2] es una herramienta creada específicamente para el entorno educativo que permite generar y manipular árboles sintácticos independientemente del analizador con el que ha sido creado, ascendente (CUP) o descendente (ANTLR). VAST procesa la especificación sintáctica y anota de forma automática dicha especificación de forma que cuando termina de procesar una cadena genera la visualización. La interfaz de visualización permite observar la pila de ejecución, el árbol sintáctico e incluso la recuperación de errores sintácticos [13]. Aunque es un entorno con claro objetivo educativo, solo se centra en el análisis sintáctico.

LISA [8] es una herramienta de generación de intérpretes y compiladores que dispone de una especificación gramatical propia. La versión 2.2 soporta analizadores de tipo LL, LR y LALR. Su planteamiento

se centra en el desarrollo de lenguajes de dominio específico³, por lo que su enfoque no es educativo. La especificación del traductor se debe realizar con una notación propia de la herramienta. Su interfaz gráfica permite visualizar el árbol sintáctico, el grafo de dependencias y los valores de los atributos del árbol enriquecido. A pesar de tener un enfoque profesional y no haber sido evaluada formalmente, existen experiencias donde los estudiantes reflejan una opinión positiva de la herramienta además de potenciar su participación activa en las clases [10].

CUPV [6] es una herramienta de visualización para las gramáticas CUP, de modo que aprovecha su especificación para la visualización, y está orientada específicamente para el entorno educativo. Permite visualizar atributos del proceso de compilación tales como valores semánticos, reducciones, el árbol sintáctico o la entrada. Actualmente no se encuentra disponible para su descarga.

VCOCO [11] es un software que permite visualizar la ejecución de traductores dirigidos por la sintaxis generados con la herramienta COCO/R⁴. Por ello, utiliza su especificación léxica, sintáctica y de TDS. Esta herramienta permite visualizar la información léxica y sintáctica relativa a la instrucción de código que está en ejecución. Su interfaz de visualización usa el enfoque de un depurador. Esta herramienta tampoco se encuentra disponible para su descarga.

EvDebugger [12] es una herramienta de corte educativo que permite diseñar traductores con la ayuda de un depurador visual. Con una especificación gramatical propia, a través de su depurador se puede visualizar el árbol sintáctico con sus nodos de atributo, los atributos de la gramática y los valores semánticos de los atributos computados. Su evaluación recogió buenas opiniones de estudiantes y profesores pero no se ha podido encontrar enlace de descarga para esta herramienta.

Es obvio que ha habido un interés por utilizar las tecnologías de visualización con los traductores. De las herramientas que hemos encontrado, las que más se acercan a nuestro propósito son LISA y EvDebugger, aunque existen ciertas lagunas por cubrir. Ambas utilizan un lenguaje de especificación propietario, la primera no está diseñada para el ámbito educativo aunque se haya hecho una evaluación referente al analizador sintáctico y la segunda no se puede descargar para su utilización.

Nuestro objetivo se centra en conseguir una herramienta que pueda visualizar los detalles de la traducción dirigida por la sintaxis y que sea tan independiente como sea posible de las herramientas de desarrollo de traductores, de forma que no haya que aprender un

¹<http://www2.cs.tum.edu/projects/cup/>, 2021

²<https://www.antlr.org/>, 2021

³DSL, Domain Specific Languages

⁴<http://www.ssw.uni-linz.ac.at/Coco/>, 2021

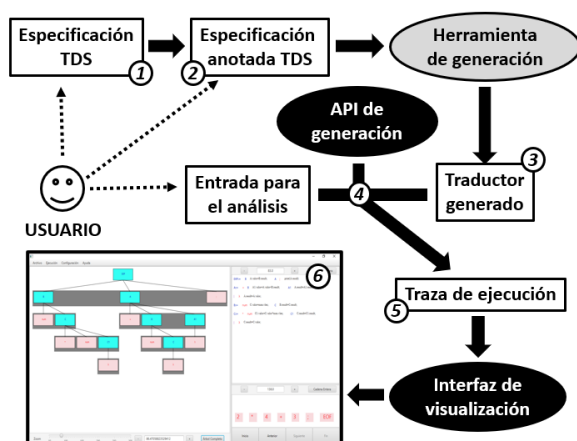


Figura 1: Arquitectura del sistema.

nuevo lenguaje de especificación ni ceñirse a un único tipo de generador de traductores.

3. Descripción de la herramienta

El objetivo principal de esta herramienta es visualizar el funcionamiento de un traductor. Aun así, también se quiere conseguir que esta visualización sea independiente de las herramientas de desarrollo de traductores.

La fig. 1 muestra un esquema del uso y la arquitectura del sistema. Este se compone de una interfaz de visualización y una API de generación de información a visualizar. La utilización sería de la siguiente forma. En primer lugar, el usuario escribe su especificación del traductor en cualquiera de los lenguajes de generación soportados por la herramienta (1), y a continuación se anota con llamadas a una API dando lugar a la *especificación anotada* (2). Se genera el traductor utilizando la especificación anotada (3). Cuando este se ejecuta (4) con una cadena de entrada, también proporcionada por el usuario, produce como efecto lateral —de las llamadas a la API— una traza de ejecución (5) que será la información utilizada por la interfaz para mostrar la visualización de la ejecución del traductor (6).

Por lo tanto, el sistema se basa en la visualización post-mortem, mostrando la ejecución del traductor una vez esta haya terminado. A continuación, se describe el diseño tanto de la interfaz como de la API.

3.1. Interfaz de visualización

La ejecución de un TDS depende del modelo de evaluación de las acciones semánticas asociadas a las producciones de la gramática. Nuestro sistema utiliza la evaluación en tiempo de compilación [1], así, las ac-

ciones semánticas que se evalúan son aquellas correspondientes a las producciones que el analizador sintáctico aplica en cada momento en función de la cadena de entrada que se esté analizando. Por ello, se necesita tener en cuenta el tipo de analizador que se está utilizando y así visualizar correctamente el proceso de construcción del árbol sintáctico. La visualización de la construcción del árbol lleva consigo la visualización de los diferentes símbolos gramaticales. Además, para visualizar la evaluación de las acciones semánticas (la ejecución del traductor) se mostrará la versión enriquecida del árbol sintáctico, proporcionando el valor de los atributos de los símbolos gramaticales así como dando información sobre las acciones semánticas ejecutadas que manipulan y modifican el valor de dichos atributos.

Todo ello nos lleva a identificar tres elementos principales en nuestra interfaz de visualización: la especificación del traductor, la entrada con que se ejecutará el traductor y el estado del árbol sintáctico enriquecido en cada momento. La especificación del traductor es necesaria por varias razones. En primer lugar, el objetivo de la enseñanza de la TDS es que los estudiantes sean capaces de crear especificaciones de traductores que funcionen cumpliendo los requisitos que se les planteen. En segundo lugar, y como consecuencia de la anterior, para poder crear especificaciones, los estudiantes deberán entender cómo se ejecutan estas, por lo que debe aparecer simultáneamente con el resto de elementos que dan información sobre su ejecución. Así, el siguiente elemento será la cadena de entrada, puesto que es quien guía la ejecución del traductor y por lo tanto proporcionará la información sobre los distintos instantes de ejecución del mismo. Finalmente, el árbol sintáctico enriquecido es propiamente la visualización del estado interno del traductor, que se asemeja al modelo mental utilizado generalmente en la enseñanza de procesadores de lenguajes [1]. Siendo estos tres elementos los fundamentales para la comprensión de la ejecución de un traductor, su visualización debe realizarse de forma simultánea, como se ha dicho anteriormente, pero además sincronizada.

La fig. 2 muestra una captura de la interfaz de visualización, donde la parte izquierda corresponde a la visualización del árbol sintáctico enriquecido, la parte superior derecha corresponde a la especificación del traductor y la parte inferior derecha corresponde a la visualización del estado de la cadena de entrada.

En la especificación del traductor se muestran los dos elementos principales de la misma: las reglas sintácticas y las acciones semánticas. Como el objetivo es centrarse en visualizar la ejecución del traductor, hemos decidido esconder los detalles específicos asociados al generador de TDS concreto. Además, el usuario puede elegir ocultar o mostrar las acciones semánti-

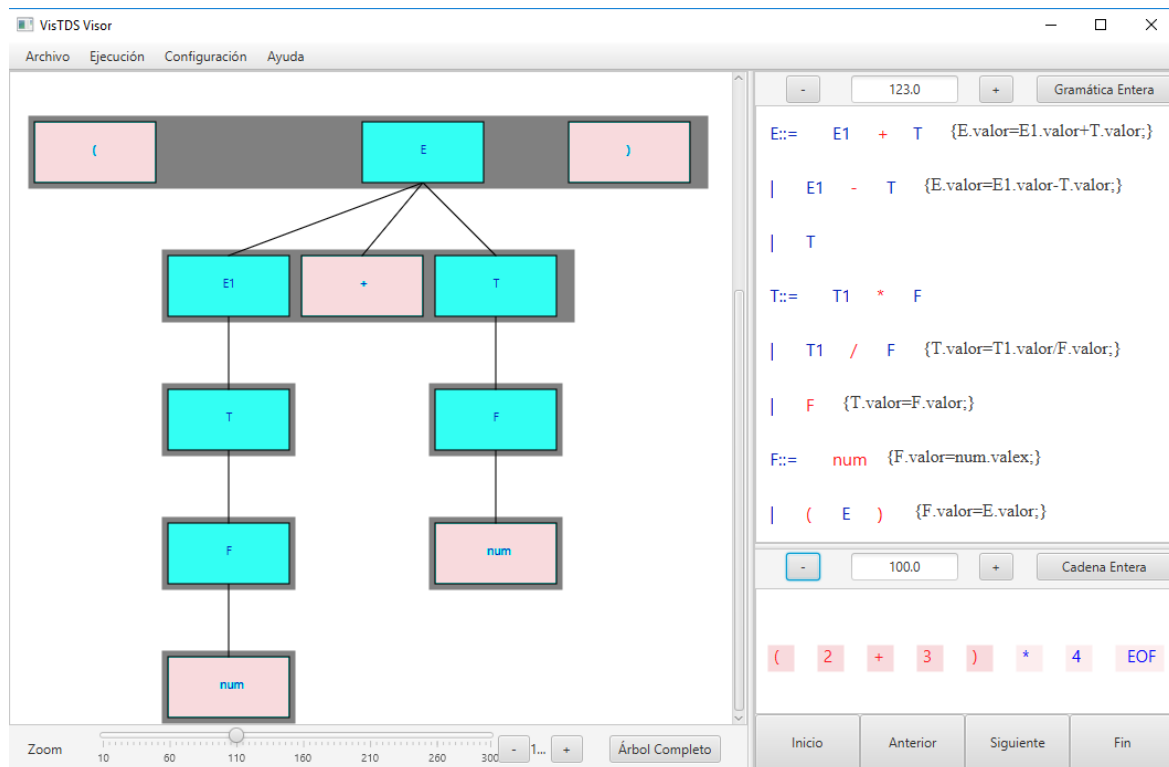


Figura 2: Captura de pantalla de la interfaz de visualización: árbol sintáctico enriquecido (zona izquierda), especificación del TDS (zona superior derecha) y estado de la cadena de entrada (zona inferior derecha).

cas asociadas a determinadas reglas. Por ejemplo, en la fig. 2 se ocultan las acciones semánticas de las producciones $E ::= T$ y $T ::= T_1 * F$. De esta forma se consigue una visualización concisa de la especificación adaptada a las preferencias del usuario.

En la cadena de entrada se muestran literalmente los símbolos terminales (lexemas de los tokens) que la forman. Estos símbolos tendrán una representación diferente según hayan sido procesados o estén pendientes de procesar. Así, en la fig. 2 se ve que se han procesado símbolos de la cadena de entrada hasta el quinto token, el resto no se han procesado todavía. Finalmente, como se ha mencionado, la cadena de entrada guía la ejecución del traductor según la va procesando, por esa razón se permite navegar por la ejecución del traductor utilizando la cadena de entrada. Así, el usuario puede seleccionar un token concreto y la visualización del árbol sintáctico enriquecido que se mostrará corresponderá al estado de ejecución cuando el traductor haya llegado a procesar dicho token.

La visualización del árbol sintáctico enriquecido comprende dos partes: el árbol sintáctico y la información de valores de atributos que muestra el resultado de las acciones semánticas. El árbol sintáctico se visualiza según su proceso de construcción –ascendente, como en la fig. 2, o descendente–, mostrando de forma diferenciada los símbolos terminales, los no terminales

y su asociación mediante producciones sintácticas. Por ejemplo, la fig. 2 muestra la aplicación de las producciones (reducciones, por ser de tipo ascendente):

$F ::= num, T ::= F, E ::= T$ y $E ::= E_1 + T$

La versión enriquecida del árbol sintáctico proporciona información bajo demanda sobre los valores de los diferentes atributos asociados a cada símbolo, la acción semántica ejecutada e incluso las explicaciones que el profesor quiera incluir. En la fig. 3 se pueden ver ejemplos de estos tres casos.

Lo que se acaba de describir es la información que ofrece de forma estática la visualización de un estado de ejecución concreto. Pero la herramienta también permite ver la evolución de este estado según se va procesando la cadena de entrada, ofreciendo por tanto una visualización dinámica de la ejecución del traductor.

Además de la visualización de los tres elementos anteriores, la interfaz tiene otras características, más básicas pero muy útiles. Así, permite configurar la visualización los elementos anteriores, asegurando una coherencia en la representación de símbolos terminales y no terminales en todas las vistas. También permite hacer zoom del árbol enriquecido, de forma que en los casos en que dicho árbol se haga demasiado grande se pueda seguir teniendo una vista global del mismo. Y finalmente, también se permite la navegación paso a paso mediante teclado y ratón.

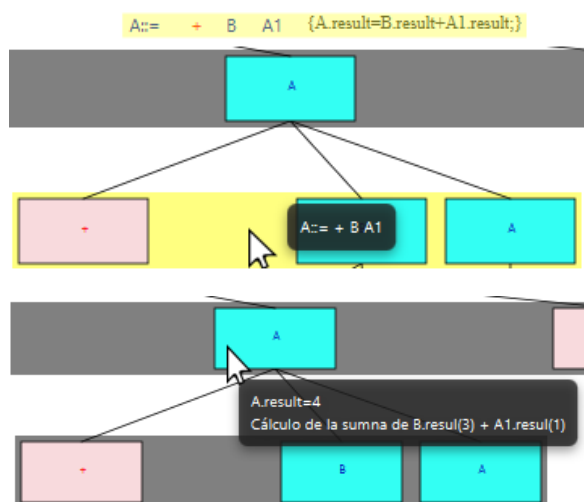


Figura 3: Ejemplos de información mostrada por el árbol sintáctico enriquecido. Arriba, producción y destacado de la acción semántica en la especificación. Abajo, valor del atributo y explicaciones añadidas.

Por último, mencionar que el diseño de la interfaz comenzó con un proceso de prototipado, al que siguió su implementación utilizando Java Swing⁵ y la API JGraph⁶ para la representación del árbol. Finalmente, la interfaz evolucionó utilizando JavaFX⁷.

El diseño de la interfaz ha seguido un proceso de prototipado con diseño participativo contando con el profesor de la asignatura de Procesadores de Lenguajes y cuatro estudiantes con conocimientos contrastados de la misma.

3.2. API de generación

El objetivo de la API es generar la información de la ejecución del traductor, la traza de ejecución, que será interpretada y visualizada por la interfaz antes descrita. La traza de ejecución se generará en formato XML, como efecto lateral de la ejecución del traductor anotado con llamadas a la API generadora. El proceso de anotación consiste principalmente en añadir llamadas a los métodos de la API dentro de las acciones semánticas de la especificación del traductor. El diseño de la API persigue principalmente dos objetivos, por un lado la facilidad de uso y por otro la cobertura de los dos tipos de analizadores: ascendentes y descendentes. Para ello, hemos trabajado con dos herramientas ampliamente utilizadas y mencionadas anteriormente: CUP para los analizadores ascendentes y ANTLR para los descendentes. Aunque los

⁵Package javax.swing, <https://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html>, 2021

⁶JGraph, <https://github.com/jgraph>, 2021

⁷JavaFX, <https://openjfx.io/>, 2021

```
A ::= plus B:b A:a1 { : RESULT=b + a1; : }
| { : RESULT=0; : };
```

```
A ::= plus B:b A:a1 { :
  writer.addStepNonTerminal("A", "result", b + a1,
    "A ::= + B A1 {A.result=B.result+A1.result;}");
  RESULT=b + a1;
: }
| { :
  writer.addStepLambda("A", "result", 0, "{A.result=0;}");
  RESULT=0;
: };
```

Figura 4: Ejemplo de la anotación de una especificación CUP. Arriba, fragmento de una especificación CUP sin anotaciones. Abajo, fragmento de una especificación CUP anotada.

métodos utilizados en ambos tipos de analizadores son similares, existen pequeñas diferencias debido por ejemplo a que CUP solo permite atributos sintetizados mientras que ANTLR permite utilizar tanto atributos sintetizados como heredados. Una parte de la anotación consiste en la creación de clases para los símbolos gramaticales (que ayudan a manipular la información de la traza correspondiente cada símbolo) y su inclusión en la especificación. Al ser totalmente mecánico, este proceso se ha automatizado mediante el comando:

```
java -jar VisTDSApiXMLCreator.jar
<especificacion>
```

A continuación se describen las anotaciones para los analizadores ascendentes (con CUP) y descendentes (con ANTLR). Independientemente del tipo de analizador y antes anotar manualmente la especificación, habrá que ejecutar el comando anteriormente mencionado que ejecutará las acciones necesarias según la especificación (CUP o ANTLR).

Anotando especificaciones CUP

Tras ejecutar el comando `VisTDSApiXMLCreator`, además de crear las clases asociadas a cada símbolo, se añadirá a la definición de los símbolos terminales su especificación como `String`, lo que permitirá su visualización en la interfaz. Una vez hecho esto basta con añadir las siguientes llamadas en las producciones de la gramática. Si la producción no corresponde a una regla borradora se añade la siguiente llamada:

```
writer.addStepNonTerminal(
  antecedente, nombre-atr-sintetizado,
  valor-atr, texto);
```

Si la regla produce lambda, el método usado es `writer.addStepLambda` con los mismos argumentos. Un ejemplo del uso de estos métodos se puede ver en la fig. 4.

```

exp returns [Exp exp0]:
    b0=b
    a0=a[Integer.parseInt(((ExpContext)_localctx).b0.b0.getValue())]
    {
        _localctx.exp0=exp0;
    }
;

exp returns [Exp exp0]:
{
    Node node=writer.addStepNonTerminal("EXP", null, null, "null");
}
b0=b
a0=a[Integer.parseInt(((ExpContext)_localctx).b0.b0.getValue())]
{
    writer.updateNonTerminals(
        "EXP::= B {A.valor=B.result;} A ; {print(A.result);}",
        ((ExpContext)_localctx).a0.a0.getValue(),
        ((ExpContext)_localctx).b0.b0);
    _localctx.exp0=exp0;
    writer.writeXML();
}
;

```

Figura 5: Ejemplo de la anotación de una especificación ANTLR. Arriba, fragmento de una especificación ANTLR sin anotaciones. Abajo, fragmento de una especificación ANTLR anotada.

Anotando especificaciones ANTLR

Al ser un analizador descendente, el árbol se crea utilizando un recorrido en profundidad. Por ello necesitamos realizar dos anotaciones, una al comienzo y otra al final de la regla. La primera se realiza con la siguiente llamada:

```

writer.addStepNonTerminal(
    antecedente, nombre-atr-heredado,
    nombre-atr-sintetizado,
    referencia-atr-heredado);

```

La llamada al final de la regla sigue este formato:

```

writer.updateNonTerminals(
    regla-gramatical,
    referencia-atr-sintetizado,
    referencia-primer-simbolo);

```

Al igual que en el caso de CUP, las reglas lambda también tienen una anotación especial:

```

writer.addStepLambda(
    antecedente, nombre-atr-heredado,
    nombre-atr-sintetizado,
    referencia-atr-heredado, regla-gramatical);

```

En la fig. 5 se muestra un ejemplo de anotación.

4. Ejemplo de uso

El objetivo principal de esta herramienta es que los estudiantes puedan entender el funcionamiento de la TDS mostrando tanto los valores de los atributos como la relación entre la aplicación de las reglas sintácticas y la ejecución de acciones semánticas.

Existen dos notaciones para especificar traductores: definiciones dirigadas por sintaxis (DDS) y esquemas

de traducción (EDT). Las primeras son más simples, ya que se suelen asociar a un funcionamiento ascendente (usando frecuentemente recursividad por la izquierda) ejecutando acciones semánticas al final de las producciones y trabajando frecuentemente con atributos sintetizados. Los segundos son más complicados, se suelen asociar a un funcionamiento descendente (usando recursividad por la derecha), permitiendo ejecutar acciones en cualquier punto del consecuente y haciendo uso tanto de atributos heredados como sintetizados.

Un problema típico de esta asignatura es que los estudiantes tratan de aplicar las soluciones más sencillas con DDS de forma frecuente, incluso cuando lo que deben usar son EDT. El problema de estos fallos es que son difíciles de detectar si no se ejecuta el traductor.

Se puede usar como ejemplo la típica gramática de operaciones aritméticas. Para los analizadores ascendentes, esta gramática es ideal, ya que permite implementar de forma inherente la asociatividad a izquierdas gracias a la recursividad a izquierdas de las reglas. El traductor es muy fácil de entender puesto que para cada regla se asocia una acción semántica que ejecuta la operación usando los operandos –atributos de los símbolos del consecuente– y devuelve el resultado en el atributo sintetizado del antecedente.

```

S ::= E {print(E.v);}
E ::= E1 + cte {E.v = E1.v + cte.v;}
E ::= E1 - cte {E.v = E1.v - cte.v;}
E ::= cte {E.v = cte.v;}

```

El problema aparece con las gramáticas recursivas a derechas, donde la asociatividad no se implementa con la recursividad de la gramática y sería necesario usar un EDT. Un ejemplo de esta podría ser:

```

E ::= cte Ep
Ep ::= + cte Ep1 | - cte Ep1 | λ

```

A pesar de no ser recursiva por la izquierda, los estudiantes tratan de implementar el mismo esquema que en los traductores ascendentes: acciones semánticas al final de las producciones y uso de atributos sintetizados. Para ello, en las producciones con operación, ejecutan esta usando como operandos los símbolos del consecuente –la constante `cte` y el atributo sintetizado del símbolo recursivo `Ep`– viéndose obligados a asignar un valor inicial neutro en la regla borradora de `Ep`, en el caso de la suma es el 0. Así, para la suma, el traductor quedaría:

```

Ep ::= + cte Ep1 {Ep.v = cte.v + Ep1.v;}
Ep ::= λ {Ep.v = 0;}

```

Como la operación aritmética de la suma tiene la propiedad asociativa, al ejecutar las sumas con asociatividad por la derecha, el traductor da resultado correcto. Y los estudiantes no se dan cuenta de que han cambiado la asociatividad.

Los estudiantes usan el mismo esquema de acciones

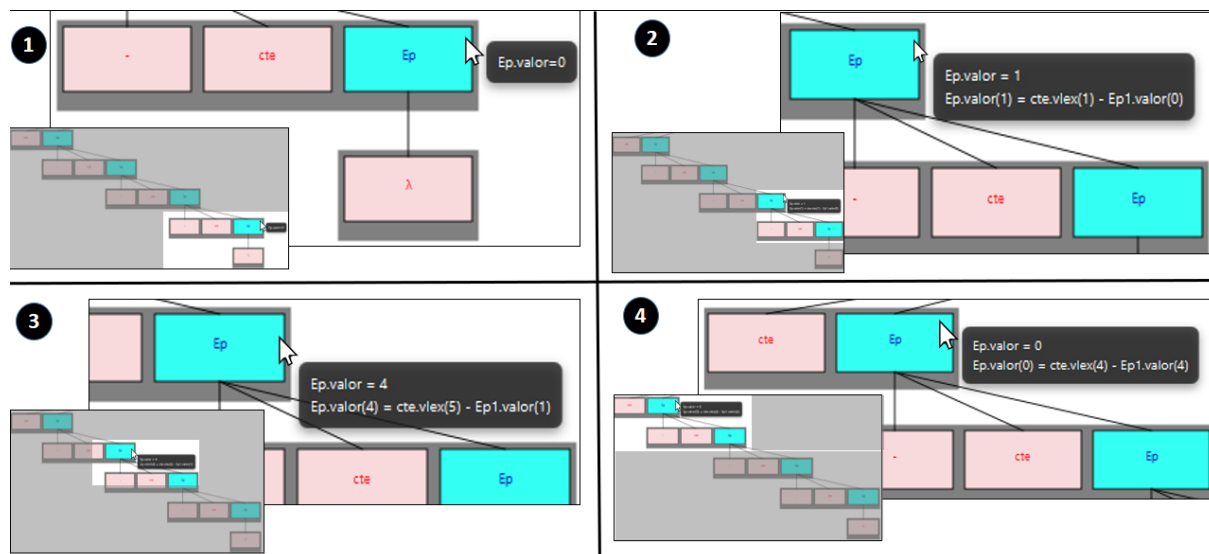


Figura 6: Capturas de la visualización de los resultados de cada operación procesada (ordenadas del 1 al 4) para la cadena 15-4-5-1. Cada paso cuenta con una vista global del árbol que identifica a la operación ejecutada.

semánticas y atributos sintetizados para la regla de la resta. Como esta operación aritmética no tiene la propiedad asociativa, el resultado es erróneo. El problema es que frecuentemente, los estudiantes no identifican el cambio de asociatividad provocado por el cambio de recursividad. Con esta herramienta, los estudiantes pueden ver paso a paso los cálculos que se van realizando, viendo el orden en que se ejecutan las acciones. En la fig. 6 se puede cómo la herramienta permite a los estudiantes ver los cálculos parciales de las restas al procesar la cadena 15-4-5-1, mostrando por qué su traductor da como resultado 15 en vez de 5.

5. Evaluación de la herramienta

El desarrollo de la herramienta finalizó en el primer semestre del curso 2019-2020, y su evaluación estaba planificada para el segundo semestre. Debido a la situación actual, bajo las restricciones impuestas por la pandemia COVID-19, no se ha podido plantear un experimento controlado que permita realizar la mencionada evaluación, siendo postpuesto al segundo semestre del curso 2020-2021.

Aún así, se ha contado con las impresiones recogidas de estudiantes con conocimientos contrastados sobre la asignatura. Esta recolección de información se ha realizado durante el proceso de diseño y desarrollo de la interfaz. Los estudiantes se mostraron entusiasmados con la posibilidad de ver una representación de la ejecución del TDS. También hicieron multitud de comentarios que han servido para mejorar diversos detalles de la interfaz. Es importante destacar que, al ser un tema muy complejo de la asignatura, el nivel

de exigencia para con la herramienta por parte de los estudiantes es muy alto.

6. Conclusiones y trabajo futuro

En este trabajo se describe una herramienta que permite visualizar la ejecución de traductores dirigidos por la sintaxis partiendo de sus especificaciones, creadas por los propios usuarios (tanto profesores como alumnos) y con sus propias cadenas de entrada. Resultados preliminares en campos relacionados [5, 13, 10, 12] nos indican que el uso de la visualización de software en la enseñanza de la traducción dirigida por la sintaxis puede ayudar a su aprendizaje.

Tras analizar las herramientas relacionadas existentes se ha podido comprobar que ninguna es capaz de trabajar con diferentes notaciones de especificaciones, algunas se restringen a un determinado tipo de analizador sintáctico [6, 11] mientras que otras tienen limitaciones de interacción y/o despliegue [8, 12], barreras que pretenden eliminarse con esta propuesta.

Con esta herramienta los usuarios pueden anotar de forma transparente sus propias especificaciones recibiendo feedback sobre su funcionamiento, sin necesidad de disponer de una batería de ejemplos previos. A nivel de desarrollo, se ha concebido para ser flexible tanto para la adaptación de nuevas notaciones de especificaciones como para ser independiente de la interfaz de visualización, teniendo la capacidad de generar la especificación de salida adaptada a las necesidades del visualizador utilizado. Gracias a la API, los textos añadidos a la aplicación de producciones y valores de atributos permiten a los profesores variar el nivel de las

explicaciones proporcionadas desde un mínimo nivel a modo de depurador hasta un nivel de explicación textual detallada. Estas animaciones se pueden utilizar tanto en clase como en el tiempo de estudio propio, algo muy necesario en el contexto educativo actual de restricciones impuestas por la COVID-19. Como el progreso de la animación está totalmente controlada por el usuario, el profesor podría hacer preguntas del estilo *¿qué pasará a continuación?* o *¿cuál será el valor de este atributo?* Por otro lado, los estudiantes pueden reproducir tantas veces como deseen la animación así como centrarse únicamente en los puntos que les interesa de la ejecución del traductor dirigido por la sintaxis.

La versión actual de la herramienta carece de una evaluación formal empírica. Sin embargo, se ha contado con las impresiones de diez estudiantes con conocimientos contrastados de la asignatura, cuatro de ellos durante el diseño y otros seis que llevaron a cabo una evaluación heurística. Además de ayudar a detectar mejoras, mostraron una opinión muy positiva sobre dos aspectos: la posibilidad de *ver* el comportamiento del traductor y la interactividad ofrecida por la herramienta.

En cuanto a los trabajos futuros, el más inmediato es la anotación automática de las especificaciones. Sin embargo, antes realizaremos una evaluación empírica de la interfaz. Por otro lado, la API debe permitir la adaptación a cualquier herramienta de generación de TDSs. Finalmente, se afrontará la liberación del código fuente para que esté a disposición de la comunidad de desarrolladores, así como el desarrollo de una versión cloud de esta herramienta que pueda ser utilizada en cualquier lugar y desde cualquier dispositivo sin necesidad de instalar ningún tipo de software.

Agradecimientos

Este trabajo ha sido financiado por el Ministerio de Economía y Competitividad (ref. TIN2015-66731-C2-1-R) y la Comunidad de Madrid, con el proyecto e-Madrid-CM (P2018/TCS-4307) , también cofinanciado por los fondos estructurales (FSE y FEDER).

Referencias

- [1] Alfred V. Aho, Monica S. Lam, Ravi Sethi y Jeffrey D. Ullman. *Compilers: Principles, Techniques, and Tools (2nd Edition)*. Addison-Wesley, USA, 2006.
- [2] Francisco J. Almeida-Martínez, Jaime Urquiza-Fuentes, y Ángel Velázquez-Iturbide. Visualization of syntax trees for language processing courses. *J. Univers. Comput Sci.*, 15(7):1546–1561, apr 2009.
- [3] Ronald M. Baecker. *Interactive Computer-Mediated Animation*. PhD thesis, M.I.T. Department of Electrical Engineering, April 1969.
- [4] Michael Hewner. Undergraduate conceptions of the field of computer science. In *ACM ICER 2013*, pages 107–114, New York, NY, USA, 2013. ACM.
- [5] Christopher D. Hundhausen, Sarah A. Douglas y John T. Stasko. A meta-study of algorithm visualization effectiveness. *J. Visual. Lang. Comput.*, 13(3):259–290, 2002.
- [6] Alan Kaplan y Denise Shoup. Cupv—a visualization tool for generated parsers. *SIGCSE Bull.*, 32(1):11–15, March 2000.
- [7] Donald E. Knuth. Semantics of context-free languages. *Math. Syst. Theory*, 2:127–145, 06 1968.
- [8] Marjan Mernik y Viljem Zumer. An educational tool for teaching compiler construction. *IEEE Trans. Educ.*, 46(1):61–68, 2003.
- [9] Thomas L. Naps, Güido Rößling, Vicky Alms-trum, Wanda Dann, Rudolph Fleischer, Christopher D. Hundhausen, Ari Korhonen, Lauri Malmi, Myles McNally, Susan Rodger y Ángel Velázquez-Iturbide. Exploring the role of visualization and engagement in computer science education. *SIGCSE Bull.*, 35(2):131–152, June 2002.
- [10] Damian Nicolalde-Rodríguez y Jaime Urquiza-Fuentes. Educational impact of syntax directed translation visualization, a preliminary study. In *IEEE VL/HCC 2018*, pages 313–314, 2018.
- [11] Daniel Resler y Deam M. Deaver. Vcoco: A visualisation tool for teaching compilers. *SIGCSE Bull.*, 30(3):199—202, August 1998.
- [12] Daniel Rodríguez-Cerezo, Pedro R. Henriques, y José-Luis Sierra. Attribute grammars made easier: Evdebugger a visual debugger for attribute grammars. In *SIIE 2014*, pages 23–28, 2014.
- [13] Jaime Urquiza-Fuentes, Francisco J. Almeida-Martínez, y Ángel Velázquez-Iturbide. Improving students’ performance with visualization of error recovery strategies in syntax analysis. *J. Res. Pract. Inf. Technol.*, 45(3/4):237–250, Aug 2013.

Herramientas de gamificación para la enseñanza de técnicas de búsqueda heurística en entornos dinámicos

Jesús Giráldez-Cru, Pablo Mesejo, José Ángel Segura,
Juan Fernández-Olivares, Antonio González

Departamento de Ciencias de la Computación e Inteligencia Artificial
Universidad de Granada

{jgiralde, pmesejo, josesegmur, faro, a.gonzalez}@decsai.ugr.es

Resumen

La Búsqueda Heurística (BH) es uno de los campos clásicos y más estudiados de la Inteligencia Artificial, y una rama troncal de los estudios de Ingeniería Informática. BH consiste en encontrar la ruta óptima hacia un estado objetivo, dadas unas condiciones iniciales, y usando una función heurística que guía dicha búsqueda a través del espacio de posibles estados. Usualmente, este es un problema computacionalmente duro de resolver y, por tanto, de gran interés tanto para estudios académicos como industriales. En este trabajo, se describe una metodología para el aprendizaje de búsqueda heurística basada en gamificación. En concreto, esta estrategia usa el entorno de desarrollo de controladores de videojuegos GVG-AI, y ha sido aplicada en los estudios de Grado de Ingeniería Informática de la Universidad de Granada durante los cursos 2018-19, 2019-20 y 2020-21. Esta metodología, basada en el trabajo individual y el aprendizaje a través del juego, ha permitido adaptar la docencia de esta materia a la modalidad virtual sin apenas cambios en la organización docente. Además no se observan diferencias significativas en las calificaciones de ambas modalidades docentes, sugiriendo la robustez de la metodología propuesta en escenarios extraordinarios de docencia virtual.

Abstract

Heuristic Search (HS) is one of the classical and most studied fields in Artificial Intelligence, and a core subject of the degree in Computer Science. HS consists of finding the optimal path to a target state, given some initial conditions, and using a heuristic function that guides the search through the space of possible states. Usually, this is a computationally hard problem, and thus of great interest for both academical and industrial purposes. In this work, it is described a new methodology to learn heuristic search based on gamification. In particular, this strategy uses GVG-AI, a video games

controller development framework, and it has been applied in the degrees of Computer Science of the University of Granada (Spain) during the years 2018-19, 2019-20, and 2020-21. This methodology, based on the individual work and the learning through the game, has allowed to adapt the teaching of this subject to online courses with minimal changes in the teaching organization. Moreover, there is no significant differences in the grades of both teaching methods, suggesting the robustness of the proposed methodology in extraordinary scenarios of online teaching.

Palabras clave

Docencia online, gamificación, búsqueda heurística, inteligencia artificial, ingeniería informática.

1. Introducción

La Búsqueda Heurística (BH) es uno de los temas fundamentales que se abordan al realizar una introducción a la Inteligencia Artificial (IA) [18]. En concreto, el libro de referencia de Russell y Norvig [24] dedica uno de sus seis bloques (*Problem Solving*) principalmente a esta temática. A nivel de contenido curricular de los estudios de Ingeniería Informática, dentro del Computing Curricula 2013 [9], se encuentra el área de conocimiento de Sistemas Inteligentes, dentro del cual se identifican 4 unidades de conocimiento troncales: *IS1/Fundamental Issues*; *IS2/Basic Search Strategies*; *IS3/Basic Knowledge Representation and Reasoning*; y *IS4/Basic Machine Learning*. La segunda de ellas, relativa a estrategias de búsqueda, es la que presenta una mayor carga docente recomendada. Todo ello redundando en la relevancia de la disciplina dentro del amplio campo de la IA.

Las técnicas de búsqueda en IA se emplean para “buscar” soluciones a un problema en un espacio de estados. Dicho espacio generalmente se representa como un grafo dirigido, o árbol de búsqueda, en donde

cada nodo es una posible configuración del problema. Existen dos tipos de técnicas de búsqueda: técnicas no informadas y técnicas informadas. Las primeras no utilizan ningún conocimiento del problema a resolver, de ahí que se llamen técnicas de búsqueda ciega, exhaustiva o no informada. Por otro lado, las técnicas de búsqueda informada o heurística introducen cierto conocimiento genérico sobre el problema, guiando la búsqueda de forma “inteligente” hacia regiones más prometedoras. Se trata de introducir un conocimiento de alto nivel para resolver el problema en cuestión. Para ello, se emplean las llamadas heurísticas. Newell, Shaw y Simon en 1957 dieron la siguiente definición [20]: “Un proceso que puede resolver un problema dado, pero que no ofrece ninguna garantía de que lo hará, se llama una heurística para ese problema”. La forma más habitual de representar una heurística es definiendo una función que asocia a cada estado del espacio de estados un valor numérico que evalúa de algún modo lo prometedor que es ese estado para acceder a un estado objetivo. Resulta clave comprender que un método de búsqueda heurística no siempre garantiza encontrar la mejor solución o solución óptima, sino encontrar una solución buena o aceptable en unos márgenes de tiempo y memoria razonables.

En general, los métodos heurísticos son preferibles a los métodos no informados en la solución de numerosos problemas, en los que una búsqueda exhaustiva necesitaría un tiempo demasiado grande. Es el caso de resolución de problemas complejos, como ganar a un campeón del mundo en los juegos del Ajedrez (Deep-Blue) [8] o Go (AlphaGo) [26], detección de contornos en imágenes ruidosas [16], testeo de sistemas electrónicos y electromecánicos [21], verificación de huellas dactilares [28], desplazamiento de robots móviles [14], y videojuegos [2], entre otros.

Por todo ello, la docencia y correcto aprendizaje de las técnicas de búsqueda heurística resultan de gran importancia en la formación de ingenieros en informática. No obstante, el aprendizaje, en ocasiones, se limita a entornos estáticos, en donde se lleva a cabo la mera búsqueda del camino más corto de un punto a otro. Esto limita severamente las posibilidades de la búsqueda heurística que, como hemos visto, puede ser empleada en distintos tipos de aplicaciones y problemas. En otros casos, la enseñanza se centra en cuestiones teóricas, más desacopladas de la práctica, lo que vuelve más árido todo el proceso y puede, con mayor facilidad, fomentar la desafección por la materia.

En este artículo, presentamos una propuesta docente sobre cómo abordar la enseñanza de técnicas de búsqueda heurística en entornos dinámicos mediante técnicas/herramientas de gamificación. Dichos entornos dinámicos requieren la integración “inteligente” de dos tipos de comportamientos:

- **Deliberativo:** representado por el algoritmo de búsqueda heurística, que permite planificar una ruta u otro tipo de actividades. En el caso de un videojuego, por ejemplo, calcular la ruta óptima para recoger algún objeto desde la posición actual del avatar.
- **Reactivo:** representado por la reacción ante cambios imprevistos en el entorno. En el caso de un videojuego, por ejemplo, la presencia de un enemigo.

En el contexto docente, la gamificación consiste en aplicar conceptos propios de videojuegos a contextos ajenos a éstos [27]. En nuestra propuesta para el aprendizaje de BH, las técnicas de gamificación se aplican a través de dos tipos de logros: niveles de dificultad y calificación. Esta propuesta se aplica en el contexto de un videojuego, cuya meta consiste en alcanzar una serie de objetivos en el menor tiempo posible. Para ello, el estudiante debe implementar un agente inteligente, que incorpore estrategias deliberativas y reactivas, y que juegue al videojuego de forma automática. Los distintos niveles del videojuego serán “desbloqueados”, a modo de logros, a medida que el estudiante avance en la implementación del agente inteligente. Para conseguirlo, estos niveles se organizan en progresiva dificultad de forma que para implementar el agente de cierto nivel, se debe partir de la implementación necesaria para superar el nivel anterior. Por otra parte, la calificación final depende (parcialmente) de una competición donde se evalúa el rendimiento de cada agente, es decir, el tiempo que cada agente requiere para resolver el nivel de mayor dificultad. Esto facilita una métrica objetiva con la que medir el rendimiento de cada agente. De esta manera, la gamificación también aparece estableciendo una competición donde el objetivo no es únicamente resolver el videojuego desbloqueando progresivamente sus niveles, sino resolverlo más rápido que el resto.

Este tipo de metodología ha facilitado que la adaptación de la docencia a la modalidad virtual en el curso 2019-20, debido a la pandemia de la COVID-19, haya sido directa, sin apenas ningún cambio en la organización docente de la asignatura. Se trata, por tanto, de una metodología flexible ante este tipo de contextos extraordinarios. Para analizarlo, se presenta un estudio estadístico entre las calificaciones del curso 2018-19 con docencia presencial y aquellas del curso 2019-20 con docencia virtual. El resultado de este análisis revela que no hay diferencias estadísticamente significativas entre las calificaciones de ambos cursos. Esto sugiere que la adaptación de la metodología propuesta es robusta a cambios en el tipo de docencia, debido al foco que pone en el trabajo autónomo e individual del estudiante.

A continuación, este artículo presenta una revisión

de algunos de los principales trabajos relacionados (sección 2) y el algoritmo más popular en el campo de la búsqueda heurística (A*, sección 3); el entorno en el que se desarrolla este proyecto docente (GVG-AI, en la sección 4); un ejemplo concreto de práctica que se puede materializar en el aula (sección 5), así como un análisis estadístico de las calificaciones en escenarios de docencia presencial y virtual (sección 6); y una serie de conclusiones al respecto (sección 7).

2. Trabajos relacionados

La ludificación, o gamificación, del aprendizaje es una aproximación educacional para motivar a los estudiantes a aprender mediante el uso de juegos y videojuegos en el entorno de aprendizaje [10]. Dicho proceso de gamificación ya ha sido propuesto y satisfactoriamente empleado en la docencia de la Ingeniería Informática con anterioridad [5]. En nuestra propuesta docente, se emplea un entorno de desarrollo de controladores para videojuegos que se presentará en la sección 5. Aparte de potenciar la diversión y motivación durante el aprendizaje de las técnicas de búsqueda heurística, nuestra propuesta permite introducir logros con los que desbloquear los distintos niveles del juego, así como una competición entre estudiantes que afecta (parcialmente) a su calificación final.

Podemos encontrar aproximaciones basadas en gamificación al aprendizaje de ciertas ramas de la IA, principalmente aprendizaje automático y/o centrado en niños y en la formación escolar más elemental [11, 12, 13, 25]. También existe alguna aproximación gamificadora a la enseñanza de técnicas de búsqueda heurística [6], pero se trata de una metodología docente basada en entornos inmersivos de realidad virtual, con todos los costes e implicaciones hardware y software que ello conlleva. Por otro lado, esta aproximación se centra en clarificar la explicación de conceptos teóricos, ignorando por tanto los aspectos prácticos de la búsqueda heurística en entornos dinámicos.

Las herramientas de gamificación también han sido usadas en otras áreas de conocimiento, como arquitectura de computadores [1], lenguas extranjeras [3], física [23], ingeniería mecánica [17], o arquitectura [19].

3. El algoritmo A*

El algoritmo A* [7] es un algoritmo de búsqueda de caminos en grafos para intentar calcular el camino óptimo entre dos nodos del mismo —el nodo inicial y el nodo objetivo—. Para ello, hace uso de una función heurística que guía el proceso de búsqueda. Nótese que este algoritmo puede aplicarse igualmente a cualquier proceso en el que se puedan representar los distintos

estados (nodos del grafo), las transiciones entre ellos (aristas del grafo) y el coste de dichas transiciones (peso de las aristas). Por ejemplo, un proceso industrial donde cada estado del proceso se representa en un nodo de un grafo, con un estado inicial (el actual) y un estado objetivo, y donde las posibles transiciones de un estado a otro se representan mediante las aristas de dicho grafo y el coste de cada transición se representa en el peso de cada arista.

Sea $G(V, w)$ un grafo dirigido y ponderado donde V representa el conjunto de nodos del grafo y $w : V^2 \rightarrow \mathbb{R}$ es la función de pesos entre cualquier par de nodos del grafo. Sean $x, y \in V$ respectivamente los nodos inicial y objetivo, se definen las funciones $f, g, h : V \rightarrow \mathbb{R}$, donde $g(n)$ representa el coste mínimo para alcanzar el nodo actual n desde el nodo inicial x , $h(n)$ representa la función heurística que estima el coste mínimo para alcanzar el nodo objetivo y desde el nodo actual n , y $f(n) = g(n) + h(n)$ es la suma de ambas.

Iterativamente, el algoritmo A* selecciona un nodo candidato n y lo expande. El conjunto de nodos candidatos está inicialmente formado únicamente por el nodo inicial x , y el proceso de expansión consiste en insertar en el conjunto de candidatos los vecinos del nodo expandido n , con sus correspondientes valores de $g(n)$ y $h(n)$. Este proceso se repite hasta que el nodo expandido es precisamente el nodo objetivo y . El criterio para seleccionar un nodo candidato es elegir aquel candidato con menor valor en la función $f(n)$. En caso de que un nodo sea alcanzado por caminos diferentes, se conservará el camino más corto, es decir, aquel camino con menor valor en $g(n)$.

Este algoritmo extiende al algoritmo de Dijkstra [4] mediante el uso de la función heurística $h(n)$ —el algoritmo de Dijkstra se puede ver como una versión simplificada del algoritmo A* donde únicamente se hace uso de la función $g(n)$ —. El objetivo de esta función es precisamente priorizar la búsqueda sobre aquellos nodos más prometedores en el camino óptimo, reduciendo así los tiempos de ejecución necesarios para obtenerlo.

En la Figura 1 se muestra un grafo de ejemplo y la función heurística $h(n)$ asociada a cada nodo, mientras que en el Cuadro 1 se describe el desarrollo del algoritmo A* en este grafo, donde el nodo inicial es A y el nodo objetivo es D . Inicialmente, el conjunto de candidatos está formado únicamente por el nodo inicial A , cuyo valor $f(A) = g(A) + h(A) = 0 + 10 = 10$. En la primera iteración se expande este nodo, insertando en el conjunto de candidatos a los vecinos del nodo expandido, es decir, los nodos B y C , con sus correspondientes valores de $f(B)$ y $f(C)$. En la siguiente iteración se expande el nodo del conjunto de candidatos con menor $f(n)$, es decir, C , y se generan sus sucesos-

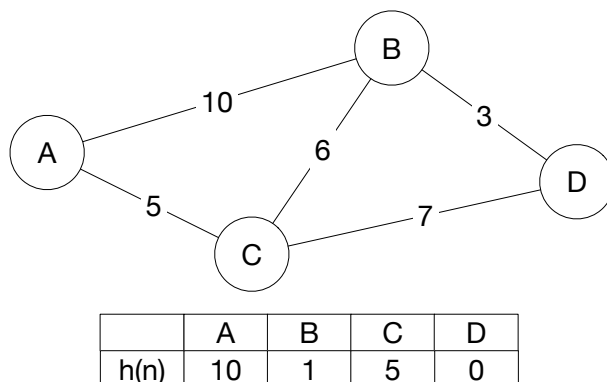


Figura 1: Grafo de ejemplo y su correspondiente función heurística $h(n)$ para alcanzar el nodo objetivo D.

Iteración	Nodo expandido n	Candidatos ($f(n) = g(n) + h(n)$)
0	-	A (10=0+10)
1	A	B (11=10+1), C (10=5+5)
2	C	B (11=10+1), B (12=11+1) , D (12=12+0)
3	B	D (12=12+0), C (21=16+5)
4	D	Fin —
Camino encontrado: A – C – D (coste=12)		

Cuadro 1: Desarrollo del algoritmo A* en el grafo de ejemplo de la Fig. 1, donde el nodo inicial es A y el nodo objetivo es D. Los valores de las funciones $f(n)$, $g(n)$ y $h(n)$ se indican entre paréntesis.

res. En este paso puede observarse cómo se descarta el camino $A - C - B$ puesto que es más largo que el camino ya existente $A - B$. Así continúan las iteraciones hasta que el nodo expandido es precisamente el nodo objetivo D. En este caso, el camino óptimo encontrado es $A - C - D$ con coste 12.

4. El entorno GVG-AI

GVG-AI (General Video Game - Artificial Intelligence) [22] es un entorno para el desarrollo de controladores inteligentes para jugar a videojuegos y generadores de contenido dinámico para los mismos. El objetivo de este entorno es, por una parte, ofrecer una plataforma versátil donde tanto los controladores inteligentes como los generadores de contenido sean capaces de aprender a jugar y generar contenido en cualquier videojuego (que puede ser inicialmente desconocido para el controlador/generador), y por otra, proveer una herramienta donde se pueda evaluar la capacidad de estos controladores en esta tarea. Este último objetivo ha dado lugar a la GVG-AI Competition, donde se evalúan anualmente las técnicas presentadas para este fin.

Con el fin de poder entrenar y/o testear los controladores, el entorno GVG-AI ofrece un conjunto de videojuegos de entrenamiento. Este conjunto se divide en varias categorías en función de la tipología del juego (2

jugadores, mapas, etc. . .). En la Figura 2 se muestra los videojuegos CamelRace y BoulderDash de la categoría “gridphysics”. Ésta es una categoría de videojuegos de un jugador (el jugador/controlador juega contra la máquina) donde el tiempo se discretiza en ticks¹, y el objetivo debe ser alcanzado en un número máximo de ticks y/o antes que el resto de rivales. En el videojuego CamelRace, el objetivo es llegar a la línea de meta, sorteando los posibles obstáculos del mapa, antes que el resto de contrincantes. En BoulderDash, el objetivo es recoger un número de gemas y dirigirse al portal de salida, realizando los túneles correspondientes para desplazarse por el mapa y evitando enemigos y obstáculos que pueden eliminar al avatar. El primer caso se trata de un juego completamente determinista, dado que no se ve afectado por eventos aleatorios; el segundo caso es semi-determinista dado que la componente estocástica del videojuego solo se produce si los enemigos se liberan.

Ambos videojuegos son ejemplos adecuados para el aprendizaje y desarrollo de técnicas de búsqueda heurística, donde además se puede evaluar objetivamente el rendimiento de un controlador, dado que el comportamiento inteligente se dará en aquellos capaces de alcanzar el objetivo más rápido, es decir, en el menor número de ticks posible.

¹El tick, por tanto, es la unidad de tiempo empleada en nuestro entorno. Se considera que, para una misma tarea, es preferible conseguir el objetivo en un menor número de ticks.



Figura 2: Videojuegos CamelRace (arriba) y BoulderDash (abajo) del entorno GVG-AI.

5. Propuesta de metodología docente basada en gamificación

La metodología docente propuesta consiste en desarrollar un controlador, basado en A* (o en alguna de sus variantes), dentro del entorno GVG-AI, que guíe a un avatar a resolver un juego en distintos niveles. El juego escogido es el juego con índice 11 en los tipos de juego `singleplayer`² de la distribución de GVG-AI, denominado BoulderDash. El objetivo de la práctica es que los estudiantes se familiaricen con las técnicas de búsqueda heurística en entornos dinámicos, que incluyen una vertiente deliberativa y una reactiva. Para ello, se proponen 5 problemas/tareas de progresiva complejidad:

1. Comportamiento deliberativo simple: búsqueda del camino óptimo al portal (sin enemigos, pero con la posible presencia de obstáculos).
2. Comportamiento deliberativo compuesto: búsqueda de 10 gemas (en un mapa con un número superior o igual de gemas) y salida por el portal.
3. Comportamiento reactivo simple: mantenerse alejado de un enemigo durante un tiempo predeter-

minado (2000 ticks)³.

4. Comportamiento reactivo compuesto: mantenerse alejado de dos enemigos durante un tiempo predeterminado (2000 ticks).
5. Comportamiento reactivo-deliberativo: búsqueda de 10 gemas (en un mapa con un número superior o igual de gemas), evitando al enemigo presente en el mapa y, una vez se tengan todas, alcanzar el portal dentro de los límites de tiempo predeterminados (2000 ticks).

Nótese que estas tareas pueden entenderse como logros, puesto que se requiere superar un nivel antes de afrontar la implementación del siguiente. Por tanto, se entiende que un nivel está bloqueado hasta que no se ha superado el anterior.

Los mapas son proporcionados por los profesores de prácticas, aunque se recomienda que los estudiantes creen sus propios mapas y verifiquen el comportamiento de sus algoritmos en ellos. Por otro lado, resulta esencial conocer y comprender los detalles del juego:

- En cada tick, el controlador dispone de 50ms para decidir la acción a realizar. En caso contrario, el

²Se pueden encontrar en el fichero `examples/all_games_sp.csv`.

³Si el avatar está en contacto con un enemigo es eliminado y, por tanto, el videojuego no es superado.

controlador es eliminado y, por tanto, el estudiante suspende el ejercicio.

- En BoulderDash, un cambio de orientación implica un tick adicional al propio movimiento. Esto es importante, de cara a utilizar el menor número de ticks, e integrar este conocimiento en $g(n)$ de A^* .

Las características que presenta esta aproximación son las siguientes: permite plantear varios objetivos; priorizar objetivos; alcanzar un objetivo es un problema de búsqueda heurística en entornos dinámicos (pero teniendo que considerar aspectos como la presencia de enemigos y el límite de tiempo); integrar comportamientos reactivo y deliberativo; replanificar (dado que el plan elaborado puede fallar); y replantear objetivos.

En cuanto a la evaluación, el planteamiento del ejercicio permite asignar con mucha facilidad una calificación según el tiempo de ejecución. Debido a que el comportamiento del controlador es objetivamente medible usando como métrica el número de ticks empleados para resolver un nivel concreto, la resolución de la práctica se puede plantear mediante herramientas/técnicas de gamificación donde el objetivo es conseguir un controlador mejor que el resto (en cuanto al número de ticks que emplea). Para ello, nuestra propuesta contempla que parte de la calificación final del estudiante se obtenga mediante una competición donde se evalúan los controladores de todos los estudiantes. Por otro lado, desde el punto de vista de evaluar los distintos niveles que presentan un comportamiento estocástico (por ejemplo, en presencia de enemigos), resulta sencillo ejecutar varias veces cada nivel y evaluar automáticamente el número de veces que sobrevive el avatar. Todo esto representa una gran ventaja, dado que da seguridad al estudiante durante el proceso de evaluación (al eliminar la subjetividad y arbitrariedad) y facilita la corrección por parte del profesor.

Por tanto, esta propuesta de metodología docente basada en la gamificación permite al estudiante desarrollar los distintos conceptos relativos a la búsqueda heurística en entornos dinámicos en distintos escenarios de progresiva dificultad. Este aprendizaje se desarrolla de manera independiente por parte de cada estudiante, donde el entorno del videojuego estimula su interés hacia conceptos de mayor complejidad gracias a la modularidad de la propuesta. Además, cabe resaltar de nuevo la facilidad con la que esta metodología se puede adaptar a escenarios de docencia virtual.

6. Análisis de calificaciones en docencia presencial y virtual

En esta sección analizamos la capacidad de adaptación, en cuanto a modelo pedagógico, de esta propuesta metodológica a la modalidad de docencia online. Pa-

ra ello, se analizan las calificaciones de los cursos académicos 2018-19 (con docencia presencial) y 2019-20 (con docencia virtual).⁴

	2018-19	2019-20
Media	8.526	7.271
Stdev.	1.033	2.784
Máximo	10.000	10.000
Perc. 75	9.000	9.625
Mediana	8.750	8.000
Perc. 25	8.250	5.800
Mínimo	5.000	0.000

Cuadro 2: Estadísticas de las calificaciones en los cursos 2018-19 y 2019-20. Los tamaños de las muestras son, respectivamente, de 58 y 56 calificaciones.

En la Figura 3 se representan los histogramas correspondientes a las calificaciones de ambos cursos académicos. Un resumen de algunas estadísticas puede encontrarse en el Cuadro 2. En un primer análisis numérico y visual, se puede apreciar ligeras diferencias en las calificaciones de ambos años.

Estas pequeñas diferencias pueden deberse a numerosos factores. La explicación más simple es el hecho de tener muestras de estudiantes diferentes, los cuales no tienen por qué tener el mismo rendimiento académico. A esta circunstancia debe sumarse los factores humanos relacionados con el inicio repentino de la pandemia y los cambios abruptos que ello supuso, tanto en la docencia como, en general, en la rutina diaria de todos nosotros.

No obstante, cabe preguntarse si verdaderamente existen diferencias estadísticamente significativas entre ambas muestras. Para verificarlo, realizaremos la prueba de la U de Mann-Whitney (también conocida como prueba de suma de rangos Wilcoxon) [15], una prueba estadística no paramétrica apropiada para analizar diferencias entre las dos muestras. En este caso, la hipótesis nula H_0 establece que las medianas de las dos muestras son iguales, mientras que la hipótesis alternativa H_1 establece que medianas difieren.

Esta prueba devuelve como resultado el estadístico $U = 3,616$, con $z = 1,5936$ como estadístico aproximado⁵ que establece el umbral de U para aceptar la hipótesis alternativa, y $p\text{-value} = 0,1110$, que indica la probabilidad de que sea posible el valor estadístico obtenido siendo la hipótesis nula cierta. Por tanto, se rechaza la hipótesis alternativa. Es decir, no existen diferencias estadísticamente significativas entre las calificaciones del curso 2018-19, con docencia presencial, y las del curso 2019-20, con docencia virtual. Este

⁴Todavía no existen datos disponibles sobre las calificaciones del curso 2020-21.

⁵Debido al tamaño de las muestras debe calcularse este estadístico aproximado.

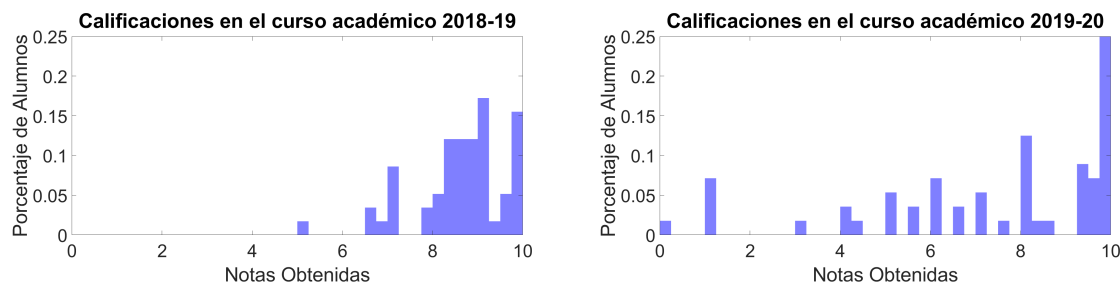


Figura 3: Histogramas de calificaciones en los cursos 2018-19 y 2019-20.

resultado sugiere la idoneidad de la aproximación empleada para tratar situaciones de docencia virtual como la actual. Posiblemente, esto es gracias al trabajo individual y autónomo del estudiante que las herramientas de gamificación facilitan.

7. Conclusiones

En este trabajo hemos presentado una aproximación basada en gamificación para la enseñanza de técnicas de búsqueda heurística en entornos dinámicos, en los que se requiere la integración de comportamientos reactivos y deliberativos. Para ello, se hace uso de un entorno de desarrollo de controladores de videojuegos, que permite una gran flexibilidad a la hora de emplear distintos juegos, plantear distintos objetivos, e incrementar o decrementar la dificultad. Nuestra propuesta es modular, en donde se parte de un comportamiento deliberativo simple, se avanza hacia un comportamiento deliberativo compuesto, pasando por un reactivo simple y un reactivo compuesto, y se finaliza con la integración de todos estos aspectos en un ejercicio reactivo-deliberativo. El modo en que se ha definido el juego facilita enormemente la evaluación de los ejercicios, aportando un framework objetivable en el que los alumnos perciben como justa su propia evaluación y simplifica el trabajo evaluador del docente. Esta metodología docente permite una fácil adaptación a la modalidad de docencia virtual, gracias a su diseño modular y progresivo basado en el trabajo individual e independiente de cada estudiante. El análisis estadístico de las calificaciones en modalidades presencial y virtual demuestra que no hay diferencias estadísticamente significativas entre ambas, sugiriendo la robustez de la propuesta en el caso de docencia virtual cuando no sea posible continuar con la modalidad presencial.

Como trabajo futuro, quedaría realizar la validación experimental de las ventajas de esta aproximación docente con respecto a otras, dado que consideramos que nuestra aproximación facilita el aprendizaje y mejora las calificaciones obtenidas por los estudiantes. Adicionalmente, sería interesante evaluar el uso de distin-

tos elementos de gamificación, como el impacto del tipo de videojuego en los logros conseguidos o la influencia del tipo de jugador en los resultados.

Referencias

- [1] Francisco J. Andújar, Arturo Gonzalez-Escribano, Javier Bastida Ibañez, y Yuri Torres de La Sierra. Aplicación de gamificación competitiva y colaborativa en asignaturas básicas de arquitectura de computadoras. En *XXVI Jornadas sobre la Enseñanza Universitaria de la Informática (JENUI 2020)*, 2020.
- [2] Vadim Bulitko, Yngvi Björnsson, Nathan Sturtevant, y Ramon Lawrence. Real-time heuristic search for pathfinding in video games. En *Artificial Intelligence for Computer Games*, pp. 1–30, 2011.
- [3] Alejandro Calderón Sánchez y Mercedes Ruiz. Aprendizaje integrado de contenidos y lenguas extranjeras mediante juegos serios para la formación bilingüe en ingeniería del software. En *XXVI Jornadas sobre la Enseñanza Universitaria de la Informática (JENUI 2020)*, 2020.
- [4] Edsger W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.
- [5] Carina S. González González y Alberto Mora Carreño. Técnicas de gamificación aplicadas en la docencia de ingeniería informática. *ReVisión*, 8, 2015.
- [6] Foteini Grivokostopoulou, Isidoros Perikos, y Ioannis Hatzilygeroudis. An innovative educational environment based on virtual reality and gamification for learning search algorithms. En *2016 IEEE Eighth International Conference on Technology for Education (T4E)*, pp. 110–115, 2016.
- [7] Peter E. Hart, Nils J. Nilsson, y Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.*, 4(2):100–107, 1968.

- [8] Feng-Hsiung Hsu. Ibm's deep blue chess grand-master chips. *IEEE Micro*, 19(2):70–81, 1999.
- [9] Joint Task Force on Computing Curricula, Association for Computing Machinery (ACM), and IEEE Computer Society. Computer science curricula 2013: Curriculum guidelines for undergraduate degree programs in computer science. *Association for Computing Machinery*, 2013.
- [10] Karl M. Kapp. *The gamification of learning and instruction: game-based methods and strategies for training and education*. Joh Wiley and Sons, 2012.
- [11] Kapsu Kim. An artificial intelligence education program development and application for elementary teachers. *Journal of The Korean Association of Information Education*, 23:629–637, 2019.
- [12] Kapsu Kim y Youngki Park. A development and application of the teaching and learning model of artificial intelligence education for elementary students. *Journal of The Korean Association of Information Education*, 21:139–149, 2017.
- [13] Kapsu Kim y Youngki Park. Development of a board game-based gamification learning model for training on the principles of artificial intelligence learning in elementary courses. *Journal of The Korean Association of Information Education*, 23:229–235, 2019.
- [14] Sven Koenig y Maxim Likhachev. Improved fast replanning for robot navigation in unknown terrain. En *2002 IEEE International Conference on Robotics and Automation, ICRA 2002*, pp. 968–975. IEEE, 2002.
- [15] Henry B. Mann y Donald R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *Annals of Mathematical Statistics*, 18:50–60, 1947.
- [16] Alberto Martelli. An application of heuristic search methods to edge and contour detection. *Communications of the ACM*, 19(2):73–83, 1976.
- [17] Carmen Mata, Elena M. Beamud, y David Calvo-Parra. Utilización de técnicas de gamificación para potenciar el aprendizaje en una asignatura optativa. En *XXIII Congreso Universitario de Innovación Educativa en las Enseñanzas Técnicas, CUIEET 2015*, pp. 822–830, 2015.
- [18] John McCarthy. What is artificial intelligence? <http://jmc.stanford.edu/articles/whatisai/whatisai.pdf>, 2007.
- [19] Miguel Muñoz-Calvente, Pelayo Fernández, Mario López, y Javier Gracia. Gamificación en la impartición de cálculo de estructuras. En *XXVI Congreso Universitario de Innovación Educativa en las Enseñanzas Técnicas, CUIEET 2018*, pp. 612–620, 2018.
- [20] Allen Newell, John C. Shaw, y Herbert A. Simon. Empirical explorations of the logic theory machine: a case study in heuristic. En *Western Joint Computer Conference: Techniques for reliability*, pp. 218–230, 1957.
- [21] Krishna R. Pattipati y Mark G. Alexandridis. Application of heuristic search and information theory to sequential fault diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(4):872–887, 1990.
- [22] Diego Pérez-Liébana, Jialin Liu, Ahmed Khalifa, Raluca D. Gaina, Julian Togelius, y Simon M. Lucas. General video game AI: A multitrack framework for evaluating agents, games, and content generation algorithms. *IEEE Transactions on Games*, 11(3):195–214, 2019.
- [23] Montserrat Rivas Ardisana, Pablo Álvarez Alonso, y Pedro Gorría Korres. La ludificación como herramienta de motivación en la asignatura bilingüe waves and electromagnetism. En *XXVI Congreso Universitario de Innovación Educativa en las Enseñanzas Técnicas, CUIEET 2018*, pp. 600–611, 2018.
- [24] Stuart J. Russell y Peter Norvig. *Artificial Intelligence - A Modern Approach, Third International Edition*. Pearson Education, 2010.
- [25] Bawornsak Sakulkueakulsuk, Siyada Witoon, Potiwat Ngarmkajornwiwat, Pornpen Pataranutaporn, Werasak Surareungchai, Pat Pataranutaporn, y Pakpoom Subsoontorn. Kids making AI: Integrating machine learning, gamification, and social context in STEM education. En *2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, pp. 1005–1010, 2018.
- [26] David Silver, Aja Huang, et. al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.
- [27] Miguel De Simón-Martín, Ana M. Diez-Suárez, Jorge Blanes-Peiró, David Borge-Diez, y A. González-Martínez. Aplicación de técnicas de gamificación para la consolidación de conocimientos en asignaturas del Área de ingeniería eléctrica y energética. En *IX Congreso Nacional de Ingeniería Termodinámica*, 2015.
- [28] David M. Tumey, Tianning Xu, y Craig Arndt. U.S. patent no. 6,963,659. *U.S. Patent and Trademark Office*, 2005.

Implementación de una metodología basada en el aprendizaje por proyectos como herramienta para aumentar la motivación de los estudiantes en asignaturas transversales de Informática

Jordi Fonollosa, Xavier Aymerich,
Mislav Jordanic, Ana Tost, Sergio
Romero, Rosa Maria Fernandez Canti
Departament d'Enginyeria de Sistemes,
Automàtica i Informàtica Industrial
Universitat Politècnica de Catalunya
Barcelona
jordi.fonollosa.m@upc.edu

Alessandro Maccarrone

Escola Els Arcs
Barcelona

smaccarrone@elsarcs.cat

Resumen

Es habitual que planes de estudio para titulaciones de grado incluyan una asignatura transversal de informática en el primer año. En los estudios de Náutica encontramos un perfil de alumnado que percibe la asignatura ajena a la profesión escogida, lo que se refleja en una baja motivación. Para dar respuesta a esta realidad, con el objetivo de aumentar la motivación del alumnado y contextualizar la materia en el centro de interés, se ha llevado a cabo una experiencia de aprendizaje basado en proyectos (ABP). El reto planteado consiste en elaborar, en grupos reducidos, un programa funcional que dé respuesta a alguna de las necesidades propias del ámbito de la náutica. Durante el proceso, se elabora un diario para tomar consciencia del proceso de trabajo y de los aprendizajes. Al final, cada grupo presenta el proyecto a sus compañeros. Se ha valorado la satisfacción del alumnado en relación a la metodología docente y su percepción de aprendizaje mediante cuestionarios, comparándolos con un grupo control, que ha seguido una metodología expositiva participativa. Los resultados evidencian que la metodología ABP da lugar a una mayor satisfacción y a una percepción de un aprendizaje más profundo. La experiencia es generalizable a otras titulaciones con poco carácter tecnológico, siendo necesario adaptar el centro de interés. Para facilitar dicha tarea, se hace público el material generado en la presente experiencia docente.

Abstract

The academic plan in the first year of many bachelor degrees normally includes a course that covers the basics of informatics. In the nautical studies there are students that perceive the course of informatics un-

connected to the envisioned career, resulting in low motivation. With the goal of boosting motivation and contextualize to the center of interest, we implemented a project-based learning methodology. The methodology proposes projects performed in small groups, with the objective to create a functional computer program that solves a problem in the nautical field. The students are required to elaborate a reflective learning diary to consolidate and extend learning through reflection and to be aware of the learned skills. At the end of the course, the projects are presented in front of the class. The satisfaction of the students and their perception of the learned material was measured using questionnaires and was compared to the control group, which followed an expository methodology. The results show that the project-based methodology yields greater satisfaction of the students and perception of deeper acquired knowledge. The proposed methodology can be easily adapted to other degrees with moderate presence of information and communication technologies, being necessary to adapt the center of interest. To ease this task, we make the resources publicly available.

Palabras clave

Experiencia docente, aprendizaje por proyectos, learning by doing, didáctica de la informática, motivación del alumnado.

1. Introducción

La motivación del alumnado y su participación activa acarrear una mayor implicación en el proceso de enseñanza-aprendizaje [4]. Sin embargo, en ocasiones aparecen obstáculos que dificultan dicha motivación, por ejemplo, cuando se percibe la materia como algo

ajeno a la profesión escogida. Este es el caso de las asignaturas de Informática que se imparten en los grados de la Facultat de Nàutica de Barcelona (FNB) de la Universitat Politècnica de Catalunya (UPC). Además, el estudiantado de estas titulaciones suele contar con pocos recursos y habilidades en el ámbito de la informática, hecho que refuerza su desconexión de la materia. Por lo tanto, cualquier intento de aumentar su motivación y su implicación pasa por establecer conexiones entre los contenidos específicos de la asignatura y el ámbito general de los grados. Esto fomenta, también, que los aprendizajes resulten más significativos y funcionales.

El aprendizaje basado en proyectos (ABP), es un contexto muy adecuado para establecer este tipo de conexiones. Esta metodología también contribuye a desarrollar competencias transversales y potencia la autonomía y la toma de decisiones por parte del alumnado, lo cual redundará en unos aprendizajes más profundos y duraderos [5]. Cabe destacar también que los proyectos son ámbitos idóneos para aplicar los contenidos en contextos reales, motivadores y significativos [4]. En definitiva, la apuesta por el método de ABP se justifica a partir de tres de sus beneficios potenciales: la eficacia para promover aprendizajes profundos, significativos y duraderos, la capacidad de mejora de la motivación del alumnado y la oportunidad para desarrollar también competencias de carácter transversal [7].

Para definir el ABP tomaremos la definición de Alcoba [1], según la cual, se trata de un “trabajo cuya finalidad es un producto concreto, habitualmente condicionado por unos requisitos de tiempo y de recursos, en el que la planificación de tareas y la resolución de incidencias cobran especial importancia”. Existen distintos ejemplos de aplicación del trabajo por proyectos en el ámbito universitario [5, 6, 7, 8], cada uno desde distintos ámbitos de conocimiento y con sus particularidades. Es habitual que bajo el nombre de ABP, se engloben propuestas muy distintas en la concreción [2]. Por eso se hace necesario puntualizar el enfoque que se ha aplicado en esta experiencia.

En primer lugar, se trata de un ABP colaborativo [5], ya que el trabajo se ha realizado en grupos. De esta manera no solo se pretende desarrollar las competencias específicas del ámbito informático, sino también las de trabajo en equipo y comunicación efectiva. En segundo lugar, el diseño de la secuencia didáctica sigue un modelo de aprendizaje por descubrimiento guiado [3], en que el alumnado aprende autónomamente una serie de contenidos y habilidades, a partir de referencias y ejemplos propuestos por el docente. De esta manera, se desarrolla la autonomía y se fomenta la curiosidad, mientras se garantiza, a la vez, el recubrimiento de unos contenidos básicos y comunes.

De esta manera, el proyecto se articula alrededor de tres elementos: los objetivos de aprendizaje (específicos del tema de algorítmica y programación y transversales); el producto final, que orienta las actividades y genera motivación (desarrollar un programa funcional, contextualizado en el ámbito de la náutica) y las actividades de aprendizaje que sirven para desarrollar el producto y para alcanzar los objetivos de aprendizaje. Todo ello acompañado de un proceso continuo de evaluación: inicial, para compartir los objetivos de aprendizaje y conocer los conocimientos previos; durante el proceso, para adquirir consciencia de los distintos aprendizajes y tomar decisiones en función de ello; y final, para comprobar la consecución de los objetivos de aprendizaje. De esta manera se pretende dar respuesta a la doble función reguladora y certificadora de la evaluación.

En este trabajo presentamos la puesta en marcha de una metodología basada en el aprendizaje por proyectos para la asignatura de Informática en los grados de Náutica. En contraste con la mayor parte de experiencias ABP documentadas anteriormente para la enseñanza de la Informática en la educación superior, aquí presentamos la implementación para un conjunto de titulaciones con poco carácter tecnológico. Además, en este trabajo, con el fin de verificar el efecto sobre el estudiantado, se compara, mediante cuestionarios, la satisfacción y percepción de aprendizaje de los estudiantes que han seguido la metodología ABP con respecto a un grupo control que siguió la metodología expositiva participativa.

La experiencia presentada es generalizable a otras titulaciones alejadas de la teoría de la información y comunicaciones, donde la Informática es una materia transversal, siendo necesario adaptar el centro de interés a la titulación correspondiente.

Para facilitar dicha tarea y enriquecer el banco de material disponible, se hace público el material generado en esta experiencia docente, que incluye los objetivos de aprendizaje, la rúbrica de evaluación, un modelo para el diario de aprendizaje, los test de satisfacción y de auto-evaluación y la programación de sesiones. El material es accesible desde el siguiente enlace https://gitlab.com/jfono/pbl_documentation.

2. Contexto de la experiencia

La experiencia académica que se presenta se ha llevado a cabo en la asignatura de Informática en la Facultat de Nàutica de Barcelona (FNB), centro de la Universitat Politècnica de Catalunya (UPC). Dicha asignatura es de carácter obligatorio y tiene una carga docente de 6.0 créditos ECTS y se ubica en el primer curso de los planes de estudio para las titulaciones de Grado en Tecnologías Marinas (GTM), Grado en Náutica y Transporte Marítimo (GNTM) y Grado en Ingeniería en Sistemas y Tecnología Naval (GESTN).

Si bien la asignatura se imparte en el primer y segundo semestre, la gran mayoría de los estudiantes la cursan en el primer semestre de la titulación respectiva. Aunque la asignatura se imparte en tres titulaciones distintas, el contenido es el mismo para todas ellas. La programación de la asignatura está dividida en cuatro bloques, cuyos objetivos de aprendizaje se resumen a continuación.

Bloque I: Introducción a la informática: Entender los conceptos y definiciones formales básicos de la informática. Comprender la arquitectura y estructura básica de los sistemas informáticos.

Bloque II: Sistemas digitales: Saber representar información mediante distintos sistemas de codificación y sistemas de numeración. Entender los conceptos básicos de la lógica y álgebra de Boole.

Bloque III: Algorítmica y Programación: Saber elaborar un programa informático para realizar una tarea de forma automática. Comprobar el correcto funcionamiento de un programa y localizar los errores. Optimizar el código de un programa y hacerlo accesible a otros usuarios. Comunicar las funcionalidades de un programa y el diseño del código.

Bloque IV: Redes de comunicación: Reconocer los elementos de una red de área local, su arquitectura, protocolos de comunicación y servicios. Conocer las características principales de los protocolos de comunicación en el entorno marino.

El sistema de evaluación de la asignatura incluye dos pruebas parciales. El estudiante tiene que realizar un primer examen parcial que cubre los bloques I y II y otro parcial que cubre el bloque III. Si supera estas pruebas parciales, la materia correspondiente queda liberada para el examen final; en caso contrario, dicho examen también incluye los bloques no superados. La calificación final se obtiene mediante una suma ponderada entre las notas de los exámenes parciales, la nota del examen final y los ejercicios que se realizan durante el curso. Para superar la asignatura, la calificación final tiene que ser superior a cinco.

La experiencia llevada a cabo durante el primer semestre del curso 2019-2020 ha consistido en sustituir la prueba parcial correspondiente al bloque III por la implementación de un proyecto de programación.

3. Metodología de aprendizaje basado en proyectos

Se aplicó una metodología basada en un aprendizaje práctico, basado en proyectos, para el bloque III de la asignatura de Informática. Las sesiones programadas tienen una duración de 2 horas, realizando 2 sesiones semanales durante 5 semanas, para un total de unas 10 sesiones.

A lo largo de estas 5 semanas, se pide a los estudiantes el planteamiento e implementación de un

programa que esté contextualizado en el ámbito marino. La implementación del proyecto se realiza en grupos de 4 estudiantes. El tema del proyecto es libre, pero se recomienda la elección de un reto con una complejidad baja sobre el cual se puedan ir añadiendo nuevos elementos. Asimismo, los grupos pueden escoger la implementación del proyecto a través un lenguaje de programación mediante línea de comandos (Matlab) o un lenguaje de programación visual por bloques (Scratch). Se proponen los entornos Matlab porque tiene continuidad en las prácticas de otras asignaturas en las titulaciones y Scratch para dar respuesta a los estudiantes con conocimientos y habilidades previas limitadas. En función del interés de los grupos en profundizar, los programas pueden ser más o menos sencillos e incluir funcionalidades básicas y avanzadas. Por ejemplo, la gestión de reservas de butacas en un buque permite una solución inicial básica, sobre la cual se pueden añadir nuevas funcionalidades como reservas familiares, selección de ventana/pasillo, cancelación de reservas, etc.

La evolución del proyecto ocurre principalmente fuera del aula, de forma autónoma y coordinada por parte del estudiantado. Durante el proceso, elaboran un diario de aprendizaje donde exponen lo que han aprendido, la evolución del proyecto y la gestión interna del grupo. Ese ejercicio permite tomar consciencia del proceso de trabajo y de sus aprendizajes.

Las sesiones en el aula tienen un carácter práctico. Durante la parte inicial de la sesión, el profesor propone tareas que ayudan a la construcción del proyecto. Se trata de ejercicios para conocer y poner en práctica estructuras de programación o funcionalidades específicas para adquirir buenas prácticas de programación. A continuación, se propone que lo aprendido con la tarea correspondiente se traslade al proyecto. Finalmente, se resuelven dudas específicas sobre la programación de los proyectos para, si es el caso, desatascar la progresión del proyecto. Igualmente, esa interacción sirve al profesor para comprobar la evolución del trabajo y proponer nuevos retos e implementaciones para una elaboración más compleja del proyecto.

A continuación, se explica brevemente la programación de las sesiones para el bloque III y la realización del proyecto por parte de los estudiantes. Salvo las sesiones 1, 5 y 6, en las que la tarea propuesta requiere la sesión entera, el resto de la sesión después de la implementación de la tarea correspondiente, se dedica a la programación del proyecto y resolución de dudas específicas:

Sesión 1: Durante esta sesión inicial el profesor presenta la metodología para el bloque III de la asignatura. Se inicia con una parte expositiva donde se presentan los pasos para la implementación de un programa informático, tipos de variables y operado-

res, estructuras algorítmicas e instrucciones, funciones, procedimientos, librerías y programación modular y buenas prácticas de programación. A continuación, se presenta el proyecto a los estudiantes y se especifica que tiene que estar contextualizado en el entorno marino. Se presentan los objetivos de aprendizaje (ver sección 2) y la rúbrica de evaluación. Se pide que se organicen en grupos de 4 para la siguiente sesión.

Sesión 2: Se inicia compartiendo un tutorial para el lenguaje de programación seleccionado para cada grupo. Se pide como tarea la implementación de un programa siguiendo una estructura secuencial de instrucciones.

Sesión 3: Se propone una tarea que requiere el uso de una estructura condicional de programación y el uso de vectores.

Sesión 4: De forma similar, se propone una tarea para trabajar una estructura repetitiva, que a su vez se puede combinar con otra estructura condicional. Se propone la construcción de un menú interactivo donde el usuario navega por las opciones con las teclas numéricas.

Sesión 5: Es una sesión de carácter práctico que se realiza en el aula de informática con sistemas Arduino. Se realizan tareas simples que trabajan las estructuras de programación secuencial (control de encendido y apagado de un LED), condicional (alarma de ruido con un micrófono y un LED) y repetitiva (contador).

Sesión 6: Como la anterior, vuelve a ser una sesión de prácticas con la plataforma Arduino. En esta sesión no se proporciona el código completo al estudiante, quien tiene que definir variables y organizar los bloques de código proporcionados. Con un sensor de temperatura el estudiante tiene que mostrar por el puerto serie la temperatura. En la segunda parte, el estudiante tiene que mostrar la temperatura media de las últimas 5 lecturas más recientes.

Sesión 7: Se propone una tarea inicial para la organización y estructura correctas del código, mediante el uso de funciones propias. También se propone una tarea para la gestión de archivos, escritura y lectura de datos en un archivo.

Sesiones 8 y 9: Se dedican exclusivamente a pulir el proyecto, asegurar que el resultado final es funcional y definir la presentación final.

Sesión 10: Los estudiantes presentan a sus compañeros los trabajos realizados. En unos 10 minutos cada grupo presenta su proyecto, mostrando una demo, una breve descripción del código y su estructura, las particularidades de su proyecto y la organización interna del grupo. A continuación, responden a preguntas de compañeros y del profesor.

La evaluación del bloque III tiene presente el programa entregado, el diario de aprendizaje, la presentación y el trabajo en grupo según la rúbrica que se

comparte con los estudiantes. La rúbrica incluye detalles sobre el funcionamiento del programa, el cual tiene que ser funcional y bien planteado y contextualizado, la experiencia de usuario, la estructura propia del código, la coordinación entre los miembros del grupo, el diario de aprendizaje, la presentación en el aula y la conceptualización del proyecto.

4. Diseño experimental

4.1. Metodologías docentes

En este estudio se realizó la recopilación de datos de los estudiantes matriculados en la asignatura de Informática impartida en el primer semestre del curso 2019-2020. La metodología de ABP se comparó con una metodología expositiva participativa en términos de satisfacción del alumnado y auto-percepción de aprendizaje.

La metodología expositiva consistió en 3 sesiones de teoría impartidas como clases magistrales, 3 sesiones de problemas y dos sesiones prácticas que coincidían con las impartidas en las sesiones 5 y 6 de la metodología ABP. Las sesiones teóricas se centraron en una primera sesión de introducción a la informática y en concreto al lenguaje C, una segunda sesión enfocada a la introducción de estructuras básicas de programación y una tercera sesión centrada en el tratamiento de listas. Las sesiones de problemas se realizaron en clase con soporte en papel y se centraron principalmente en el diseño e implementación de algoritmos en pseudo-código y la ejecución de la traza de un programa.

Se siguió la metodología ABP para los estudiantes de GNTM y la metodología expositiva participativa para los estudiantes de GTM y GESTN. El Cuadro 1 muestra el número de estudiantes que siguió cada metodología. El grupo GNTM configura el grupo experimental que siguió la ABP, mientras que el grupo control lo forman los estudiantes de los grupos GTM y GESTN. La investigación llevada a cabo es un cuasi-experimento en el que los estudiantes se distribuyen en el grupo experimental y control según la titulación. Sin embargo, los intereses de los estudiantes y su perfil son similares en ambos grupos, como lo demuestra la media y la mediana de las notas de acceso, 7.6 y 7.3 para el grupo experimental, y 7.5 y 7.0 para el grupo control.

4.2. Satisfacción y valoración por parte de los estudiantes

Todos los estudiantes participantes en el estudio rellenaron un cuestionario anónimo valorativo de su experiencia sobre la metodología docente aplicada en el bloque III de la asignatura. El cuestionario se pasó en papel el último día del bloque y se descartaron los

estudiantes que declararon haber asistido a menos del 50% de las clases.

Grupo	Metodología docente	
	Expositiva	ABP
GNTM		40
GESTN	22	
GTM	14	
Total estudiantes:	36	40

Cuadro 1: Número de estudiantes participantes en el estudio, según la metodología docente que siguieron.

En general estoy satisfecho como se ha impartido el tema 3 de informática
Mi dedicación global comparado con los temas 1 y 2 ha sido superior para el tema 3
El tema 3 ha sido interesante y adecuado a mis necesidades
El aprendizaje ha sido útil para los estudios que estoy cursando
El aprendizaje ha sido útil para mi desarrollo personal y profesional
La metodología aplicada en el tema 3 favorecerá que lo que he aprendido quede consolidado y se mantenga a largo plazo
Estoy contento de cómo se ha repartido el número de sesiones de teoría / problemas / prácticas
En cuanto al balance de sesiones de teoría / prácticas, creo que hay demasiada teoría
He echado de menos más componente práctica
He echado de menos más componente teórica
He encontrado las sesiones prácticas interesantes
He aprendido a diseñar algoritmos
He aprendido a programar
He echado en falta más trabajo individual
He echado en falta más trabajo en grupo
El sistema de evaluación me ha parecido adecuado para los contenidos trabajados
La forma en que hemos evaluado este tema me ha permitido demostrar todo lo que he aprendido

Cuadro 2: Cuestionario presentado a los estudiantes para la valoración de la metodología docente.

El cuestionario incluía 17 preguntas cerradas para que los estudiantes valorasen los aspectos generales de la metodología docente, la motivación, la metodología docente, el aprendizaje, el trabajo en grupo y la evaluación. En particular, cada pregunta estaba formulada según un elemento de tipo Likert de seis puntos, que van del 1 (nada de acuerdo) hasta el 6 (totalmente de acuerdo). El Cuadro 2 muestra el cuestionario de valoración que han respondido los estudiantes. Finalmente, el cuestionario presentaba un

campo abierto para consideraciones generales que el alumnado quisiera hacer constar.

Para comparar las respuestas de los dos grupos, control y experimental, se recurre al test estadístico de U Mann-Whitney. Se considera que las distribuciones son significativamente distintas si $p < 0.05$.

4.3. Autovaloración del conocimiento y aprendizaje

Para evaluar la percepción del conocimiento, habilidades y aprendizaje por parte del estudiantado, se distribuye entre los estudiantes participantes un cuestionario autovalorativo de conocimientos. Dicho cuestionario consta de 20 preguntas relacionadas con conceptos de algorítmica y programación. Las preguntas son cerradas y tienen 4 posibles respuestas: No lo sé / Me suena / Lo sé / Lo sé y lo sabría explicar.

El cuestionario se distribuye entre los estudiantes dos veces: durante la primera sesión del bloque III y una vez concluido dicho bloque. El Cuadro 3 muestra el detalle de las preguntas realizadas a los estudiantes.

Programación en bloques
Programación mediante línea de comandos
Estructuras secuenciales
Estructuras condicionales
Estructuras repetitivas
Tipos de variables en programación
Traza de una variable / traza de un programa
Vectores y matrices de variables
Hacer que un programa lea los datos del usuario.
Mostrar al usuario el resultado de una operación.
Funciones (en programación)
Librerías (en programación)
Programación estructurada / modular
Hacer que un programa ejecute instrucciones diferentes dependiendo del valor de una variable.
Hacer que un programa repita unas instrucciones hasta que se cumpla una determinada condición.
Detectar dónde se encuentra el error en el código.
Estructurar el código para facilitar que otras personas lo comprendan y lo modifiquen.
Hacer que el programa realice operaciones con variables numéricas.
Hacer que el programa realice operaciones con cadenas de texto.
Gestión de archivos externos desde un programa. escribir / acceder a un archivo.

Cuadro 3: Cuestionario para la autovaloración de conocimientos y habilidades.

5. Resultados

En esta sección se muestran los resultados de comparar las valoraciones del grupo experimental con el grupo control.

5.1. Satisfacción y valoración por parte de los estudiantes

Para valorar la satisfacción de la metodología docente ABP, con respecto una metodología expositiva participativa, se contraponen los resultados agregados de los cuestionarios de satisfacción. La Figura 1 muestra cómo los estudiantes que han seguido la metodología ABP reportan una mayor satisfacción general de cómo se ha impartido el bloque III. Del mismo modo, los estudiantes del grupo experimental consideran que la metodología aplicada favorece un aprendizaje significativo que perdure en el tiempo (Figura 2) y que el aprendizaje es útil para su desarrollo personal y profesional (Figura 3).

Estoy satisfecho como se ha impartido el tema

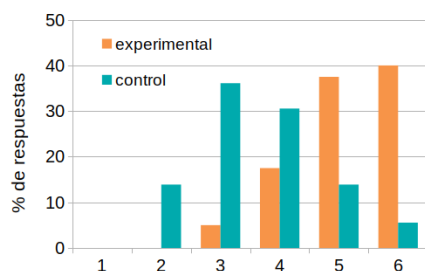


Figura 1: Respuesta de los estudiantes a la pregunta estoy satisfecho como se ha impartido el bloque III. 1: Nada. 6: Mucho. Grupo experimental (N=40), grupo control (N=36).

La metodología aplicada favorecerá que lo que he aprendido se consolide a largo plazo

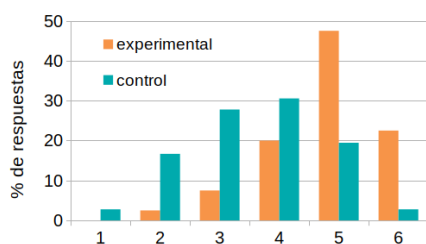


Figura 2: Respuesta de los estudiantes a la pregunta creo que la metodología aplicada favorecerá que lo que he aprendido se consolide en el tiempo. 1: Nada. 6: Mucho. Grupo experimental (N=40), grupo control (N=36).

En general, los estudiantes del grupo experimental valoran más positivamente el sistema de evaluación basado en la rúbrica de evaluación presentada. Consideran que el sistema de evaluación es más adecuado

según los contenidos trabajados (Figura 4) y que les permite mostrar mejor sus conocimientos.

Las diferencias observadas en las distribuciones obtenidas para el grupo control y el grupo experimental se consideran estadísticamente significativas ya que $p < 10^{-5}$ (Figuras 1, 2 y 4) y $p < 0.01$ (Figura 3).

El aprendizaje ha sido útil para mi desarrollo personal y profesional

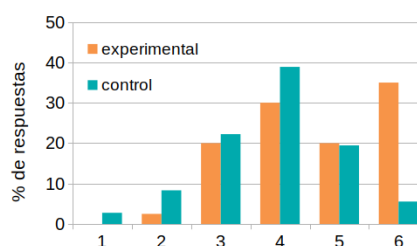


Figura 3: Respuesta de los estudiantes a la pregunta creo que el aprendizaje ha sido útil para mi desarrollo personal y profesional. 1: Nada. 6: Mucho. Grupo experimental (N=40), grupo control (N=36).

El sistema de evaluación es adecuado

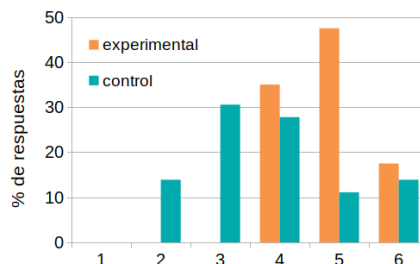


Figura 4: Respuesta de los estudiantes a la pregunta creo que el sistema de evaluación es adecuado para los contenidos trabajados. 1: Nada. 6: Mucho. Grupo experimental (N=40), grupo control (N=36).

5.2. Autovaloración del conocimiento y aprendizaje

Para evaluar el aprendizaje percibido por parte del estudiantado, según la metodología que han seguido, se utiliza el cuestionario de satisfacción ya que también incluye preguntas genéricas sobre la percepción del aprendizaje. Las Figuras 5 y 6 muestran las distribuciones correspondientes a la percepción de aprendizaje respecto a las habilidades de programación y el diseño de algoritmos, respectivamente. El grupo experimental valora más positivamente de forma significativa su aprendizaje a programar ($p < 10^{-3}$), pero la diferencia observada en el aprendizaje para diseñar algoritmos no llega, por poco, a ser significativa ($p = 0.06$).

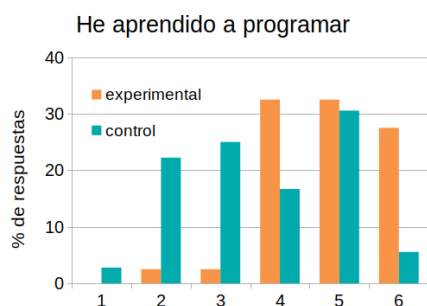


Figura 5: Respuesta de los estudiantes a la pregunta he aprendido a programar. 1: Nada. 6: Mucho. Grupo experimental (N=40), grupo control (N=36).

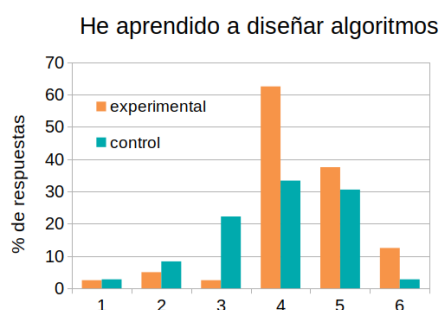


Figura 6: Respuesta de los estudiantes a la pregunta he aprendido a diseñar algoritmos. 1: Nada. 6: Mucho. Grupo experimental (N=40), grupo control (N=36).

El cuestionario de autovaloración del conocimiento, como se presentó al inicio y a la conclusión del bloque III, permite ver la percepción de aprendizaje de los estudiantes y la percepción de su progresión.

Es interesante ver la percepción del estudiante de su conocimiento antes y después de la realización del bloque III de la asignatura. En particular, se puede constatar que la progresión en la percepción de aprendizaje es similar para ambos grupos, especialmente cuando se les interroga sobre conceptos teóricos. La Figura 7 muestra la respuesta agregada a las preguntas de estructuras algorítmicas, con un patrón de respuesta similar para ambos grupos, antes y después del bloque III. Dicho efecto se reproduce en otras preguntas, como en la respuesta agregada a las preguntas sobre el uso de librerías y funciones (Figura 8).

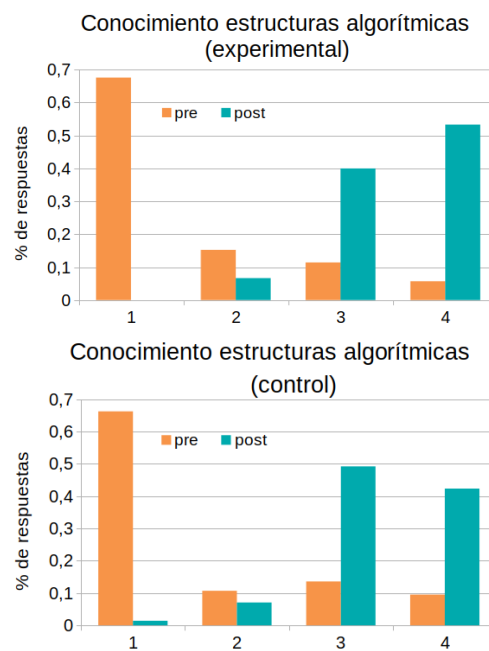


Figura 7: Respuesta agregada de los estudiantes a las preguntas sobre estructuras algorítmicas, antes (naranja) y después (verde) de la realización del bloque III, para el grupo experimental (arriba) y el grupo control (abajo). 1: No lo sé, 2: Me suena, 3: Lo sé, 4: Lo sé y lo sabría explicar. Grupo experimental (N=35), grupo control (N=24).

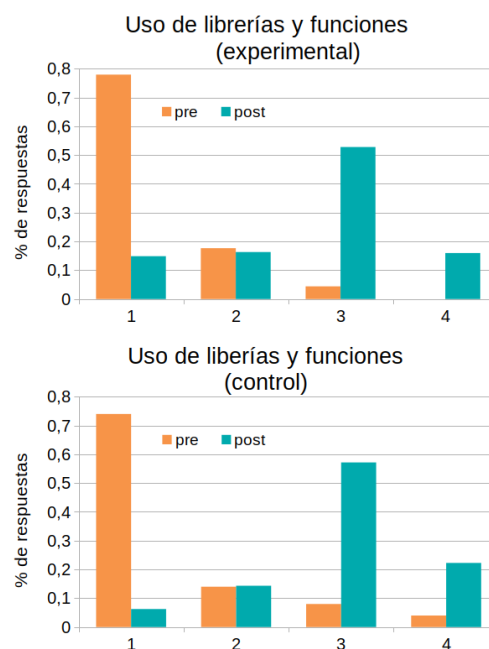


Figura 8: Respuesta agregada de los estudiantes a las preguntas sobre el uso de librerías y funciones, antes (naranja) y después (verde) de la realización del bloque III, para el grupo experimental (arriba) y el grupo control (abajo). 1: No lo sé, 2: Me suena, 3: Lo sé, 4: Lo sé y lo sabría explicar. Grupo experimental (N=35), grupo control (N=24).

6. Conclusiones

La introducción de la metodología de Aprendizaje Basado en Proyectos en la asignatura de Informática del Grado en Náutica y Transporte Marítimo de la UPC ha aumentado significativamente la satisfacción y motivación del alumnado. Tradicionalmente, dicha asignatura tenía que afrontar el reto añadido de un perfil de estudiante que la considera poco útil para su proyección y ambiciones profesionales, lo que a menudo se traducía en una pérdida de la motivación y del interés por la materia.

Mediante cuestionarios valorativos, se ha recogido la impresión de los estudiantes y se ha constatado que la implementación de la metodología ABP es útil para aumentar la valoración de la asignatura en vistas a su desarrollo personal y profesional y la satisfacción y motivación general en la asignatura. Así mismo, el profesorado participante en la experiencia ha constatado un cambio en la motivación del alumnado: del mero objetivo de superar un examen se ha pasado a una implicación en la elaboración del proyecto y a un interés por la programación.

El sistema de evaluación del ABP no solo mide contenidos conceptuales aislados, sino también competencias más globales, que integran contenidos, procedimientos y actitudes y que son necesarias para planificar, desarrollar y comunicar un proyecto completo. Así, resulta complicado comparar el impacto del ABP sobre los aprendizajes a partir de las calificaciones, dado que el grupo experimental y el grupo control han seguido sistemas de evaluación distintos. Sin embargo, al acabar el bloque sobre programación, todos los grupos que siguieron la metodología ABP fueron capaces de elaborar un programa funcional. Esto nos parece un indicador de la mejora de sus habilidades de programación y de diseño de algoritmos, que, con la metodología tradicional, resultaba menos evidente.

Los estudiantes que han seguido la metodología ABP reportan una sensación de aprendizaje mayor en sus habilidades para programar. Sin embargo, ambos grupos de estudiantes indican una progresión muy similar en la percepción de aprendizaje cuando se les interroga sobre conceptos teóricos específicos. Queda para un futuro una medida de la permanencia de los aprendizajes en ambos grupos, para evaluar de forma concreta la adherencia del aprendizaje y las habilidades adquiridas a medio y largo plazo.

Dados los resultados obtenidos, y el aumento de la motivación y satisfacción general del estudiantado, la metodología presentada en este trabajo se aplicará a las asignaturas de informática del resto de grados de la Facultad de Náutica de Barcelona, con el reto de adaptar la metodología a las nuevas condiciones forzadas de docencia no presencial o semi-presencial.

La experiencia docente presentada es generalizable a otras titulaciones. Haciendo público el material

generado esperamos contribuir a enriquecer los recursos disponibles y facilitar la implantación de experiencias ABP en otras titulaciones.

Consideraciones éticas

La participación en este estudio por parte de los estudiantes ha sido voluntaria, libre e informada. Todos ellos dieron su consentimiento para el uso de los datos de forma anónima y agregada.

Agradecimientos

Agradecemos la participación desinteresada de los estudiantes con sus valoraciones. Este trabajo ha recibido el financiamiento del Institut de Ciències de l'Educació, Universitat Politècnica de Catalunya.

Referencias

- [1] Jesús Alcoba González. La clasificación de los métodos de enseñanza en educación superior. *Contextos educativos*, 15, 93-106, 2012.
- [2] Denis Bédard, Christelle Lison, Daniel Dalle, Daniel Côté, Noël Boutin. Problem-based and Project-based Learning in Engineering and Medicine: Determinants of Students' Engagement and Persistence. *Interdisciplinary Journal of Problem-Based Learning*, 6(2), 2012.
- [3] Jerome S. Bruner. The act of discovery. *The act of discovery*. Harvard Educational Review, 31, 21-32, 1961.
- [4] Ariadna Fuertes Seder, Ricardo Ferrís Castell, Francisco Grimaldo Moreno. ¿Un cambio de metodología que aumente la satisfacción y motivación del estudiante favorece su aprendizaje? *Experiencias en el aula*. Actas de las Jenui, vol. 3. 2018. pp: 335-342
- [5] Marisabel Maldonado Pérez. Una experiencia en educación superior. *Laurus*, vol. 14, núm. 28, pp. 158-180, septiembre-noviembre, 2008.
- [6] Antoni Oller, Antonia Gallardo, Eduard Garcia, Miguel Valero. Uso de métodos ágiles y PBL en una asignatura de ingeniería del software del grado de ingeniería telemática: gestión de proyectos de ingeniería del software en un entorno docente. *Congreso Internacional de Docencia Universitaria e Innovación (CIDUI)*, 2014.
- [7] Purificación Toledo Morales, José Manuel Sánchez García. Aprendizaje basado en proyectos: una experiencia universitaria. Vol. 22, No2: doi 10.30827/profesorado.v22i2.7733 Abril, 2018
- [8] Miguel Valero-García, Javier García Zubia. Cómo empezar fácil con PBL. *JENUI 2011: XVIII Jornadas de Enseñanza Universitaria de la Informática* (2011), p 109-116.

Mujeres en estudios científico-tecnológicos en la Universidade da Coruña: evolución de estudiantes y personal docente e investigador

Beatriz Pérez-Sánchez, Noelia Sánchez-Maróño

Departamento de Ciencias de la Computación y Tecnologías de la Información
Universidade da Coruña

beatriz.perezs@udc.es, noelia.sanchez@udc.es

Resumen

En la universidad, la presencia de las mujeres se reduce conforme se aumenta el rango académico. Todo apunta a una segregación vertical de género –techo de cristal– en la carrera investigadora y la segregación horizontal en ciertas áreas científicas y tecnológicas. Existen importantes evidencias de desequilibrios en materia de género entre el estudiantado, pero también en el personal docente e investigador, especialmente en ciertas áreas de conocimiento como ciencias y tecnología. En este trabajo se analizan los datos oficiales de la Universidade da Coruña para conocer la evolución del personal, atendiendo principalmente a proporción, rango y centro asociado. Los resultados del estudio refuerzan la evidencia de un desequilibrio todavía existente en el ámbito científico y tecnológico, con especial incidencia en algunas áreas, entre ellas las Ciencias de la Computación. Los resultados ponen de manifiesto la falta de vocaciones femeninas en los estudios científicos y tecnológicos así como, la carrera de fondo a la que se enfrentan las mujeres hasta ejercer con éxito una profesión en este campo.

Abstract

In the academia, the presence of women decreases as the academic rank increases. Everything points to a vertical gender segregation –glass ceiling– in the research career and horizontal segregation in certain scientific and technological areas. There is significant evidence of gender imbalances among students, but also among teaching and research staff, especially in certain areas of knowledge such as science and technology. In this contribution, the official data of the University of A Coruña are analyzed to know the evolution of the research staff, attending mainly to proportion, rank and associated center. Likewise, we will look for possible matches with the number of students (men and women) enrolled in such centers to analyze the ef-

fect of the lack of references. The results of the study reinforce the evidence of a still existing imbalance in the scientific and technological field, with special incidence in some areas, among them Computer Science. The results show the lack of female vocations in scientific and technological studies, as well as the long-term career they face until they successfully practice a profession in this field.

Palabras clave

Segregación, género, techo de cristal.

1. Introducción

A lo largo de la historia la situación de las mujeres está marcada por la discriminación y la desigualdad. La igualdad de hombres y mujeres es un derecho, y también una necesidad para construir una sociedad más justa, cohesionada y desarrollada, tanto social como económicamente. Una forma de justicia social, en la que todas las personas tienen las mismas posibilidades de acceso, uso, control y beneficio de los servicios y recursos, así como de la participación y toma de decisiones en todos los ámbitos de la sociedad.

No hace falta echar la vista atrás ni mirar tan lejos para notar las consecuencias de esta discriminación y desigualdad, la tónica se mantiene a día de hoy en muchos lugares del mundo y en diferentes ámbitos de la sociedad. A pesar de los avances para conseguir una igualdad formal, en general no existe una igualdad real y efectiva entre hombres y mujeres, persistiendo la brecha de género. Las consecuencias de esta discriminación y desigualdad se manifiestan entre las mujeres que desarrollan su carrera en la Ciencia, la Tecnología, la Ingeniería o las Matemáticas (STEM) en España.

ONU Mujeres en su informe de 2018¹ afirma que lo-

¹ ONU Mujeres. Hacer las promesas realidad. La igualdad de género en la Agenda 2030 para el Desarrollo sostenible (2018).

gar la igualdad de género es imprescindible para avanzar en el desarrollo sostenible (DS) y para cumplir la Agenda 2030 que detalla 17 objetivos (ODS), entre los que destacamos: ODS4 Educación de calidad; ODS5 Igualdad de género; ODS9 Industria e Innovación. Al mismo tiempo denuncia una fuerte amenaza a los derechos de las mujeres y alerta de importantes consecuencias como las menores oportunidades educativas de las niñas o la brecha salarial de género.

Por su parte, la UNESCO² señala la necesidad de contar con el talento de las mujeres, incidiendo en su inclusión como clave de la mejora de resultados para toda la comunidad implicada. Asimismo, remarca la importancia de que las mujeres accedan, permanezcan y avancen en los estudios STEM para caminar hacia el DS. Los datos ponen de manifiesto la escasez de mujeres en estudios del ámbito científico-tecnológico, solo un 28,5 % de mujeres estudian carreras científicas. En España, en los años 80, había un 30 % de alumnas matriculadas en Ingeniería informática, y hoy apenas llegan al 12 % [4]. Situación parecida en el caso de Matemáticas, en el año 2000 las mujeres representaban el 60 % del aula, cifra que ha caído progresivamente hasta llegar al 37 % en 2018.

Diferentes estudios e informes^{3,4} coinciden en la necesidad de atraer a las chicas al campo científico-tecnológico, examinando los obstáculos que tienen que superar antes de entrar, una vez que entran y después, hasta llegar a ejercer con éxito una profesión en ese campo. Al mismo tiempo, concluyen que algunos de los factores más relevantes para que las mujeres no se identifiquen con el ámbito STEM proceden de los estereotipos sociales, familiares y educativos; la diferente manera de juzgar a hombres y mujeres; la falta de mentorías para adolescentes y mujeres y; la escasez de materiales didácticos que aborden la igualdad en materias STEM. Consideran de vital importancia la educación en STEM ya que las disciplinas implicadas han traído progreso en diferentes ámbitos e inciden que la educación es clave para preparar a los estudiantes para el mundo laboral a través de estudios STEM, que tendrán alta demanda en el futuro inmediato [1, 3].

Por otra parte, la Unidad de Mujer y Ciencia de la Secretaría de Estado de Universidades, Investigación, Desarrollo e Innovación, en su informe de 2021⁵, muestra que la ciencia española no disfruta de plena igualdad en la participación de hombres y mujeres en el sistema de I+D+i. En la carrera investigadora, las mu-

jerer continúan infrarrepresentadas en las categorías de mayor rango (solo un 21 % de catedráticas de universidad), mientras se aprecia cierto equilibrio de género en las demás categorías. Esto pone de manifiesto el techo de cristal o segregación vertical de género que se aprecia de forma aún más notable en los órganos de gobierno unipersonales (la mayor brecha de género en el cargo de máximo nivel, solo un 8 % de mujeres en el cargo de rectoras). Asimismo, se observa segregación horizontal de género en algunas áreas de conocimiento, la mayor infrarrepresentación de investigadoras se encuentra en Ingeniería y Tecnología (con solo un 24 % de mujeres entre su personal investigador).

El Consejo Superior de Investigaciones Científicas (CSIC) señala en su informe de 2020⁶ que los resultados en cuanto a brecha de género han mejorado tímidamente en los últimos años. Sin embargo, los datos referentes a la evolución de la carrera científica de hombres y mujeres en el CSIC, se traducen en una gráfica de efecto tijera que muestran que la situación ha empeorado en 2019. CSIC es una institución con larga trayectoria en políticas de igualdad y aún se encuentra lejos de alcanzar la paridad deseada. Esto pone de manifiesto que los cambios tardan mucho tiempo en materializarse y que cualquier avance, por pequeño que sea, es positivo.

Este artículo se estructura de la siguiente forma. El Apartado 2 contextualiza brevemente la situación de la mujer en el mundo académico. El Apartado 3 concentra los objetivos y metodología del estudio realizado, presenta los datos y realiza un análisis cuantitativo de los mismos, tanto para el colectivo alumnado como para el personal docente e investigador (PDI); finalmente el Apartado 4 expone las conclusiones extraídas del estudio.

2. Contexto

En los últimos años la presencia de mujeres en estudios universitarios ha ido en aumento, superando incluso el número de hombres matriculado⁷. Sin embargo, hay excepciones en algunas ramas de conocimiento. En concreto, en el ámbito técnico la situación dista bastante de alcanzar la equiparación de género⁸.

Esta es una preocupación compartida, estudiada y analizada desde diferentes enfoques y ámbitos geográficos. Por este motivo, se han puesto en marcha diversas iniciativas encaminadas a la equiparación entre

²UNESCO. Science Report: Towards 2030 (2015).

³American Association of University Women. Why So Few? Women in Science Technology Engineering and Mathematics (2012) y Solving the Equation: The Variables for Women's Success in Engineering and Computing (2015).

⁴UNESCO. Descifrar el código: La educación de las niñas y las mujeres en ciencias, tecnología, ingeniería y matemáticas (2019).

⁵Unidad Mujer y Ciencia, Secretaría de Estado de Universidades, Investigación, Desarrollo e Innovación. Científicas en Cifras (2021).

⁶Consejo Superior de Investigaciones Científicas. Informe Mujeres Investigadoras (2020).

⁷Ministerio de Educación y Formación Profesional. Evolución del alumnado matriculado en estudios de Grado y 1º y 2º ciclo por sexo (2020) y Distribución porcentual del alumnado matriculado en estudios de Grado según sexo y ámbito de estudio (2020).

⁸Organisation for Economic Co-operation and Development. The Pursuit of Gender Equality An Uphill Battle (2017).

hombres y mujeres en el campo científico-técnico ofreciendo la esperanza de un cambio de tendencia [2, 6].

Con respecto al colectivo del profesorado universitario, y resumiendo lo expuesto en la sección previa, aunque se observa un incremento en la presencia de mujeres, por norma general la situación en la carrera investigadora no es equiparable entre hombres y mujeres. La brecha es especialmente crítica si se atiende a categoría profesional o al desempeño de determinados cargos de responsabilidad.

Esta contribución persigue conocer la presencia de la mujer en el Sistema Universitario Gallego (SUG) y en la Universidade da Coruña (UDC). En el caso del alumnado interesa conocer si las mujeres están presentes en las distintas áreas científicas mientras que, para el PDI, además del área científica, se consideran otros aspectos como la categoría profesional y la estabilidad laboral. También se comprobará si la situación del SUG sigue la línea del resto de universidades españolas.

3. Estudio

Antes de comenzar cabe mencionar que, de manera general, se aprecian dificultades a la hora de afrontar las investigaciones en género. A pesar de que la situación ha mejorado en los últimos años, todavía es habitual la falta de datos desglosados y concordantes por sexo y la falta de indicadores de evaluación. Eso se debe a que muchos datos no se gestionaron en su momento para obtener estadísticas que puedan interesar desde la perspectiva de género.

La metodología seguida parte de la recogida de información oficial disponible, continúa con la selección de variables de interés, la aplicación de estadística descriptiva y análisis cuantitativo y, finalmente, la extracción de conclusiones. El acceso a los datos es público y se realiza a través de los sitios web institucionales⁹.

Este estudio persigue conocer la evolución en los últimos años y la situación actual de dos colectivos, alumnado y personal docente e investigador, tanto en la UDC como en el SUG. Todo el análisis se enmarca dentro de una perspectiva de género.

3.1. Colectivo alumnado

En lo que respecta al alumnado matriculado en el SUG cabe destacar:

- Los datos de estudiantes matriculados en las universidades gallegas que conforman el SUG para el

	16/17	17/18	18/19
CC. de la Salud	73,3	72,7	72,7
Ciencias	53,1	51,9	51,7
CC. Jurídicas y Sociales	60,7	60,3	59,9
Ingeniería y Arquitectura	28,1	27,6	26,5
Artes y Humanidades	67,2	66,6	66,8
Total	55,2	55,3	55,2

Cuadro 1: Porcentaje de mujeres matriculadas en el SUG desglosado por área científica para los cursos académicos 2016/17, 2017/18 y 2018/19.

curso 2018/19 ponen de manifiesto una diferencia de más de 10 puntos porcentuales entre mujeres matriculadas (55,2 %) y hombres (44,8 %). Sin embargo, este aparente acceso y predominio de las mujeres en la Educación Superior está marcado por su participación masiva en determinadas ramas de conocimiento, especialmente en las Ciencias de la Salud (72,7 %) y su escasa presencia en la rama de Ingeniería y Arquitectura (26,5 %). Como se observa en el Cuadro 1, que presenta el porcentaje de mujeres matriculadas en el SUG desglosado por área científica durante los cursos 2016/17, 2017/18 y 2018/19, los datos son claros. La presencia de mujeres es superior en todas las áreas a excepción de Ingeniería y Arquitectura, donde su presencia se limita aproximadamente a un 27 % del alumnado matriculado.

Por su parte, en el caso de la UDC:

- Durante los cursos académicos 2016/17, 2017/18 y 2018/19 (véase Cuadro 2) la distribución de alumnado matriculado es de 51 % de mujeres frente al 49 % de hombres y, la situación es diversa según la rama de conocimiento. Prácticamente todas las áreas están feminizadas, predominando las mujeres en las especialidades de Artes y Humanidades o Ciencias de la Salud. Por el contrario, las mujeres están infrarrepresentadas en la rama de Ingeniería y Arquitectura. Por tanto, sigue la misma tendencia mostrada previamente para el SUG.
- Prestando atención a la rama de Ingeniería y Arquitectura, la Figura 1 muestra el porcentaje medio de hombres y mujeres matriculados en los distintos centros de estudios para los cursos académicos 2017/18, 2018/19 y 2019/20. Como se puede observar, en algunos centros la brecha de género es más que evidente – Informática es muestra de ello, únicamente un 15,39 % de mujeres entre su alumnado– mientras que, en una minoría – Arquitectura y Diseño Industrial– la situación está bastante normalizada. Los datos ponen de mani-

⁹Universidade da Coruña (<https://www.udc.es/es/cifras/>) y Unidade Muller e Ciencia de Galicia (<https://unidadedamullereciencia.xunta.gal>)

	16/17	17/18	18/19
CC. de la Salud	78,57	78,12	77,12
Ciencias	57,13	60,07	60,86
CC. Jurídicas y Sociales	57,51	57,37	58,11
Ingeniería y Arquitectura	30,98	29,25	28,99
Artes y Humanidades	74,01	70,93	74,64
Total	51,03	50,74	51,22

Cuadro 2: Porcentaje de mujeres matriculadas en la UDC desglosado por área científica para los cursos académicos 2016/17, 2017/18 y 2018/19.

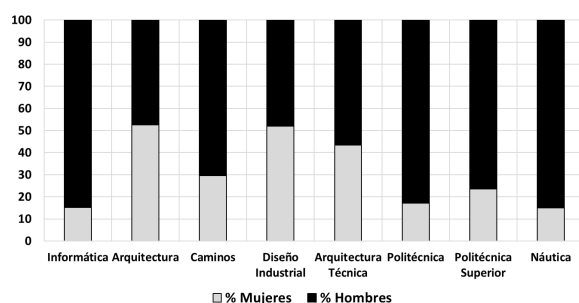


Figura 1: Porcentaje medio de hombres y mujeres en centros de la rama de Arquitectura e Ingeniería de la UDC. Se presentan datos promedio correspondientes a los cursos académicos 2017/18, 2018/19 y 2019/20.

fiesto que los estudios tecnológicos no atraen a mujeres salvo en las excepciones mencionadas. Se puede afirmar, por tanto, que en la UDC la presencia de mujeres en los estudios técnicos se encuentra estancada desde hace años.

El informe de Julián Merelo Guervos [4] publicado en el año 2017 recoge, a partir de datos del Ministerio de Educación, cómo ha ido variando la presencia de las mujeres en los estudios de Informática del sistema universitario público español en las últimas tres décadas. La conclusión es demoledora, el porcentaje se reduce de forma constante a lo largo de este período. En este estudio se desglosa la presencia femenina en cada institución, y se comprueba que en ninguna de ellas las mujeres representan más de un 20 % del total del colectivo de estudiantes. La situación en la UDC y el SUG refleja la tendencia a nivel estatal.

Las iniciativas promovidas desde la UDC y sus diferentes centros para equiparar el número de hombres y mujeres entre el estudiantado [5], al igual que muchas otras tanto de instituciones públicas como privadas, o de iniciativa individual, o grupal están siendo reconocidas y aceptadas y sus efectos están considerados como muy positivos. Aun así, se considera necesario que estas acciones se incorporen en el día a día estableciendo

Curso	% Hombres	% Mujeres
2015/16	60,46	39,54
2016/17	59,95	40,05
2017/18	59,20	40,80
2018/19	58,84	41,16

Cuadro 3: Evolución del PDI en centros propios del SUG para los cursos comprendidos desde 2015/16 a 2018/19.

Curso 2018/19	% Hombres	% Mujeres
CAT-UN	77,60	22,40
CAT-EU	59,60	40,40
TIT-UN	57,40	42,60
TIT-EU	61,8	38,20
Personal Contratado	54,60	45,40
Otros	69,60	30,40

Cuadro 4: Reparto de PDI por categorías en centros propios del SUG. Curso 2018/19. *Notación:* CAT-EU: Catedrático de Escuela Universitaria; CAT-UN: Catedrático de Universidad, TIT-EU: Titular de Escuela Universitaria; TIT-UN: Titular de Universidad.

una guía de buenas prácticas y un protocolo de actuación. Esto permitiría mantener el apoyo en el tiempo y afianzar la promoción de las mujeres en estudios y profesiones STEM [2].

3.2. Colectivo PDI

Centrando ahora la atención en la presencia de las mujeres en el PDI del SUG cabe destacar:

- Los datos de los últimos cursos académicos ponen de manifiesto que la presencia de mujeres es minoritaria. El Cuadro 3 muestra los datos desglosados por género para el PDI de los centros propios para 4 cursos académicos. Aunque se observa una cierta mejora –de una brecha de 20,92 puntos porcentuales en el 2015/16 se pasa a 17,6 puntos de diferencia en el 2018/19–, la situación está todavía muy alejada de alcanzar un 50 % de mujeres entre el PDI total.
- Atendiendo a la presencia de mujeres según la categoría profesional (véase el Cuadro 4), los datos confirman que están infrarrepresentadas en todos los niveles. La mayor diferencia se aprecia en el rango superior de la academia, la categoría de catedráticos de universidad (CAT-UN), con una brecha de 55,2 puntos porcentuales.

Por su parte, en el caso de la UDC:

- La Figura 2 muestra la evolución de hombres y

mujeres entre el PDI en los últimos años. La presencia de mujeres es considerablemente inferior a la de hombres. Al igual de lo que ocurre en el SUG, se alcanza una mejora en los últimos años (de una brecha de 32,7 puntos porcentuales en el 2009 se pasa a 20,2 puntos en el 2019). Sin embargo, la situación continua muy alejada de la igualdad.

- Se analiza también la evolución del profesorado según su régimen jurídico de contratación. En la Figura 3 se observa como en el caso de los hombres su presencia es mayor en el cuerpo de funcionarios, al contrario que las mujeres que tienen una mayor representación en el régimen laboral. Por otro lado, la presencia de la mujer mejora en los últimos años, pero aún así, en el 2019 continúa con una diferencia de 29,41 puntos porcentuales en el cuerpo de funcionarios y de 10,43 en laboral con respecto a los hombres.
- En cuanto a la evolución del cuerpo docente (véase Figura 4) se aprecia que los hombres son mayoría en todas las categorías. La presencia de las mujeres tiende a aumentar en los últimos años, no obstante, en la categoría de mayor rango (CAT-UN) la diferencia se mantiene por encima de los 60 puntos porcentuales en el año 2019.
- Considerando el resto de categorías, el Cuadro 5 presenta los datos del PDI doctor para cada rama y centro, incluyendo también la diferencia en puntos porcentuales para la brecha de género. El caso general en la rama de Ingeniería y Arquitectura es una elevada brecha de género –superior a 40 puntos porcentuales– en la mayor parte de los centros, casi 50 en el caso de la Fac. de Informática y más de 70 puntos en la ETS de Caminos. En el caso de las áreas de Ciencias, Ciencias de la Salud y Artes y Humanidades, la situación es más equilibrada. Finalmente, en Ciencias Sociales y Jurídicas, aunque la norma general tiene a cierta paridad (Ciencias de la Educación y de la Comunicación están claramente feminizadas), llama la atención el caso de Ciencias del Deporte y Educación Física con una brecha de 63 puntos porcentuales.
- Con respecto a la evolución del PDI en la rama de Ingeniería y Arquitectura (período 2009-2019), se puede indicar que, de manera general la presencia de las mujeres ha ido en aumento, aun así las cifras distan de las deseables. Por ejemplo, en Caminos en el año 2011, las mujeres únicamente representaban el 8,77 % del PDI total, aumentando de manera considerable hasta el 18,75 % en el 2019. En Informática en 2009 el porcentaje de mujeres era de un 21,7 %, llegando a un 25,17 % en el 2019. La diferencia más significati-

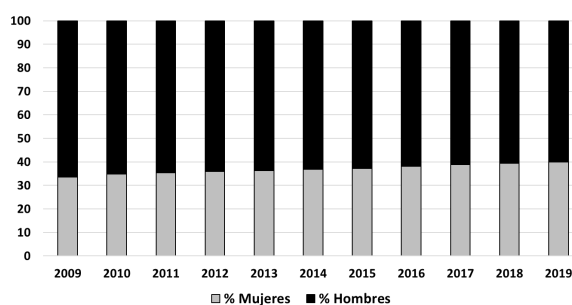


Figura 2: Evolución del PDI desglosado por género en la UDC. Se presentan datos porcentuales correspondientes al período 2009-2019.

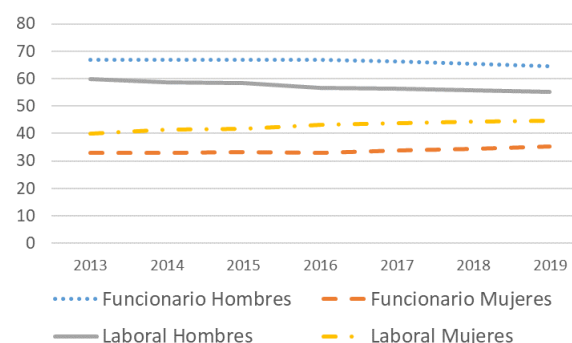


Figura 3: Evolución del PDI desglosado por género según régimen jurídico en la UDC. Se presentan datos porcentuales correspondientes al período 2013-2019.

va se aprecia en el caso de Náutica y Máquinas donde las mujeres representaban únicamente el 16,67 % del PDI total en el año 2009 llegando al 32 % en el 2019.

En cuanto a la situación del PDI a nivel estatal, los últimos informes del Ministerio de Universidades^{10,11} presentan los datos para el curso académico 2018/19. En dichos estudios se constata que las mujeres representan el 41,6 % del PDI total, siendo únicamente un 36,3 % del funcionariado y solo 23,9 % de los catedráticos de universidad. Si tenemos en cuenta el área científica, las mujeres son minoría en todas ellas y, el porcentaje más bajo se encuentra en la rama de Ingeniería y Arquitectura (23,4 %). A este respecto, en el año 2019 la proporción de mujeres en la solicitudes de acreditación para el acceso a cuerpos docentes univer-

¹⁰Disponible en https://www.universidades.gob.es/stfls/universidades/Estadisticas/ficheros/Datos_y_Cifras_2020-21.pdf

¹¹Disponible en https://www.ciencia.gob.es/stfls/MICINN/Ministerio/FICHEROS/Cientificas_en_Cifras_2021.pdf

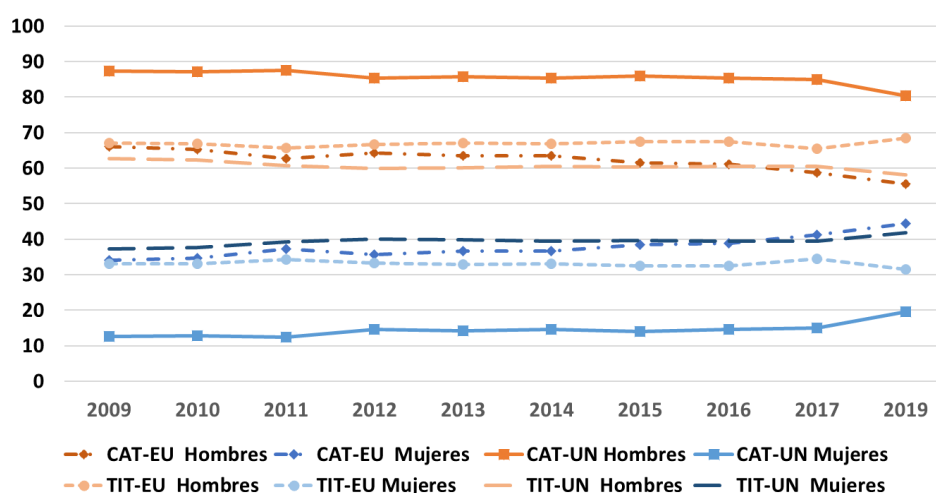


Figura 4: Evolución del PDI en cuerpos docentes universitarios. Se presentan datos porcentuales correspondientes al período 2009-2019. *Notación:* CAT-EU: Catedrático de Escuela Universitaria, CAT-UN; Catedrático de Universidad, TIT-EU: Titular de Escuela Universitaria; TIT-UN: Titular de Universidad.

sitarios en la rama de Ingeniería y Arquitectura es del 29 % para la categoría de TIT-UN y únicamente de un 21 % para CAT-UN.

4. Conclusiones

A la vista de los resultados se puede afirmar que los datos de la UDC reflejan fielmente la situación general del SUG y éste, a su vez, sigue la tendencia estatal. A continuación, se indican algunas reflexiones y conclusiones derivadas del estudio llevado a cabo. En primer lugar, en lo que respecta al estudiantado:

- La presencia de mujeres en los estudios superiores ha ido en aumento en los últimos años, llegando a una situación de paridad. Sin embargo, aunque su presencia es mayoritaria en la mayor parte de las áreas de conocimiento, la excepción es la rama de Ingeniería y Arquitectura.
- La presencia de mujeres en los estudios técnicos no solo no está normalizada, sino que desciende desde hace años. Más de la mitad de las universidades públicas españolas continúan perdiendo mujeres y en el caso de los estudios de Informática se llega a porcentajes inferiores al 10 %.
- A lo largo de los últimos años se han promovido diferentes iniciativas para abordar la brecha de género (visualización de modelos femeninos de proximidad, redes de apoyo, programas de mentoría o difusión de la aplicabilidad social de la Informática, etc.). Sin embargo, el cambio es lento e incluso los mejores programas tardarán años en producir beneficios. Por este motivo, hay que

señalar la importancia de consolidar y mantener estas iniciativas en el tiempo.

Por otra parte, centrando la atención en el colectivo del PDI:

- La presencia de mujeres entre el PDI total es minoritaria. Aunque se demuestra una mejora en los últimos años la diferencia entre hombres y mujeres continua siendo significativa. Si la evolución sigue a este ritmo tendrán que pasar muchos años antes de alcanzar un 50 % de mujeres entre el PDI total.
- A diferencia de los hombres cuya presencia es mayoritaria en el funcionariado, las mujeres están más representadas en el régimen eventual. Los datos pueden deberse a un abandono por parte de las mujeres en ciertas etapas de la carrera académica o bien por dificultades en su carrera hacia la estabilización laboral.
- La presencia de las mujeres es minoritaria en los niveles superiores del rango académico (la brecha de género es extrema en la categoría de CAT-UN) y también en el área científica de Ingeniería y Arquitectura.

Para finalizar, simplemente mencionar que a la vista de los datos podemos confirmar la existencia de brecha de género tanto entre el alumnado como el PDI y por tanto:

- Las mujeres se enfrentan a una carrera de fondo hasta ejercer con éxito su carrera profesional.
- Las instituciones han de tener en cuenta que para reducir la brecha primero es necesario cuantificar-

Rama	Centro	% Hombres	% Mujeres	Diferencia
Ingeniería y Arquitectura	Escuela Politécnica Superior	71,19	28,81	42,38
	ETS de Arquitectura	75,60	24,40	51,20
	ETS de Ing. Caminos, Canales y Puertos	86,05	13,95	72,10
	ETS de Náutica y Máquinas	73,98	26,02	47,96
	EU de Arquitectura Técnica	51,18	48,82	2,36
	EU Politécnica	56,66	43,34	13,32
	EU Diseño Industrial	80,00	20,00	60,00
	Fac. Informática	74,78	25,22	47,56
Ciencias	Fac. de Ciencias	52,44	47,56	4,88
CC. de la Salud	EU Enfermería y Podología	55,56	44,44	11,12
	EU de Fisioterapia	55,37	44,63	10,74
	Fac. de CC. de la Salud	53,08	46,92	6,16
CC. Sociales y Jurídicas	Fac. de Derecho	67,81	32,19	35,62
	Fac. CC. del Deporte y EF	81,50	18,50	63,00
	Fac. CC. de la Comunicación	36,27	63,73	-27,46
	Fac. CC. de la Educación	47,79	52,21	-9,42
	Fac. CC. Económicas y Empresariales	58,44	41,56	16,88
	Fac. Sociología	52,93	47,07	5,86
Artes y Humanidades	Fac. de Filología	52,03	47,97	4,06
	Fac. Humanidades	54,70	45,30	9,4

Cuadro 5: Porcentaje medio del PDI doctor desglosado por rama de conocimiento, centro y género. Se presentan datos promedio correspondientes al período 2009-2019.

la y hacerla pública. Los datos deben ser evaluados de manera periódica para conocer su evolución. Esto permitirá diseñar y aplicar adecuadamente acciones orientadas a disminuir la brecha de género, y sus consecuencias negativas.

- Minimizar la brecha de género es un proceso global, lento y también colectivo ya que implica a diferentes niveles de la sociedad.

Referencias

- [1] Francisco Javier Álvarez Lires. *Psicología, género y educación en la elección de estudios de ingeniería*. Tesis Doctoral, Universidad de Valladolid, 2012.
- [2] Carmen Botella, Emilia López-Iñesta, Silvia Rueda, Anabel Forte, Esther de Ves, Xaro Benavent y Paula Marzal. Iniciativas contra la brecha de género en STEM. Una guía de buenas prácticas. En *Actas de las Jornadas sobre Enseñanza Universitaria de la Informática*, Vol. 5, páginas 349–352, 2020.
- [3] Sophia Huyer. Is the gender gap narrowing in science and engineering?. En *UNESCO Science Report: Towards 2030*, páginas 854–103. UNESCO, Paris, 2015.
- [4] Juan Julián Merelo Guervós y Cecilia Mere-lo Molina. Evolución de las matrícula femenina en el grado de informática en universidades públicas españolas. Fundación Española para la Ciencia y la Tecnología, Ministerio de Economía, Industria y Competitividad, 2017.
- [5] Beatriz Pérez-Sánchez y Noelia Sánchez-Maroto. Diversidad de género en estudios técnicos de la Universidade da Coruña e iniciativas hacia la equiparación. En *Actas del XI Congreso Iberoamericano de docencia universitaria (CIDU)*, páginas 2811–2824, 2021.
- [6] Paz Prendes-Espinosa, Pedro Antonio García-Tudela e Isabel Solano-Fernández. Gender equality and ICT in the context of formal education: A systematic review. *Comunicar*, 28(63):9–19, 2020.

Problemas y dinámicas reales para evaluaciones más reales: un caso de estudio

Julio Alberto López-Gómez, Raquel Rodrigo Rubio,
Ángeles Carrasco, Francisco Mata Cabrera
Escuela de Ingeniería Minera e Industrial de Almadén
Universidad de Castilla la Mancha
Almadén

JulioAlberto.Lopez@uclm.es, Raquel.Rodrigo.Rubio@gmail.com
Angeles.Carrasco@uclm.es, Francisco.MCabrera@uclm.es

Resumen

Son muchas las propuestas metodológicas existentes en la actualidad para el diseño de actividades y pruebas de evaluación. Sin embargo, a pesar de esto, diseñar de forma adecuada y correcta las pruebas de evaluación de una asignatura ha sido y sigue siendo un desafío para los docentes.

Este trabajo presenta un caso práctico en el que una parte de la evaluación de los estudiantes se ha realizado a través de la resolución de un problema real que debían resolver, en primera instancia, de forma individual. Después, y con las soluciones obtenidas, los estudiantes debían trabajar por parejas para, finalmente, y de forma grupal, obtener la solución completa del problema propuesto. El desarrollo de esta experiencia ha sido cualitativamente satisfactorio tanto para los estudiantes como para el profesorado, que señala la ventaja de haber podido evaluar competencias técnicas individuales y competencias transversales a nivel grupal gracias a esta dinámica de aprendizaje y evaluación.

Abstract

Currently there are many methodological proposals in the design of activities and assessment tests. Nevertheless, designing assessment tests in a correct and proper way, is and it has been quite a challenge for teachers.

This paper presents a practical case in which the students' assessment was firstly carried out by means of a real problem solving test for each student. Afterwards, the students had to work in pairs and finally in groups to get to the solution of the given problem. The development of this experience has been qualitatively satisfactory for both students and teachers, as it proves the advantage of using individual technical skills and cross curricular competencies for groups thanks to this learning and assessing methodology.

Palabras clave

Evaluación y aprendizaje, problemas reales, trabajo en equipo, aprendizaje significativo, evaluación holística.

1. Motivación

La motivación que ha dado lugar a la experiencia que se desarrolla en este artículo es doble: por una parte, tiene su origen en el proceso de aprendizaje de los estudiantes y, por otra, en la evaluación realizada por el profesorado.

1.1. Los estudiantes: pasos hacia un aprendizaje significativo

Mucho se ha discutido y se sigue discutiendo acerca de la idoneidad de los exámenes como herramienta para adquirir y evaluar conocimientos. Son muchas las críticas a este sistema de aprendizaje y evaluación: la presión que genera en los estudiantes [8], el fomento de la motivación extrínseca, que hace que los estudiantes se centren únicamente en aquello que será materia de examen y su objetivo sea, por tanto, aprobar y no aprender [1], su orientación hacia un aprendizaje superficial (no olvidemos el dicho estudiantil que dice: *asignatura aprobada, asignatura olvidada*) [5,6] y un largo etcétera.

A pesar de estas críticas, y de que son muchas las actividades de evaluación alternativas al examen que se van introduciendo cada vez con más frecuencia en asignaturas de grados universitarios, los exámenes continúan siendo el ítem de evaluación con mayor peso en la calificación final. Sin embargo, también son muchas las experiencias de evaluación alternativas al examen que se han presentado fuera y dentro de JE-NUI, donde siempre se ha mostrado una atención y preocupación especial acerca de este tema. En esta línea, se pueden señalar algunos trabajos recientes en los

que, por ejemplo, se presentan experiencias donde los exámenes fueron completamente sustituidos por actividades alternativas [7], otros en los que la evaluación se realiza mediante pruebas de tipo gincana ambientadas en universos cercanos a los estudiantes como el de Harry Potter [10] o que aprovechan el concepto de escape room para realizar la evaluación en entornos virtuales y reales [9].

Para hacer frente a los problemas anteriormente mencionados, en este trabajo se describe una experiencia en la que se ha desarrollado una prueba de evaluación alternativa al examen. Las tareas ideadas para ser resueltas por los estudiantes se han diseñado teniendo en cuenta el principio de alineamiento de Biggs [2]. Este principio dice que el estudiante debe hacer para aprobar lo mismo que debe hacer para aprender. Para ello, el objetivo principal era proponer a los estudiantes un problema real para, por una parte, favorecer la motivación intrínseca de estos [11] y, por otra parte, hacerles ver la relación entre los contenidos de la asignatura y los problemas reales que existen fuera del mundo académico.

1.2. Los docentes: pasos hacia una evaluación holística

Tradicionalmente, la evaluación ha sido considerada como el instrumento del que dispone el profesorado para certificar los conocimientos adquiridos por los estudiantes. Así pues, una evaluación correctamente diseñada permitirá conocer el grado de comprensión y adquisición de los conocimientos curriculares por parte de los estudiantes.

Con el paso del tiempo, y especialmente en la última década con la entrada en vigor del Espacio Europeo de Educación Superior (EEES), se han incorporado distintos matices al concepto de evaluación. Por ejemplo, ya no se evalúan o certifican conocimientos, sino competencias. Además, no todas las competencias evaluadas son competencias técnicas, propias de la asignatura, sino que también se evalúan competencias transversales. Las competencias transversales son aquellas comunes para cualquier profesión o trabajo y que constituyen la base sobre la cual es posible adquirir competencias técnicas [12]. Algunas de ellas son la comunicación oral y escrita, liderazgo, creatividad, trabajo en equipo...

Sin embargo, la evaluación de las competencias transversales plantea una serie de retos: uno de ellos es inherente a la naturaleza de este tipo de competencias, cuya evaluación es principalmente cuantitativa y difícilmente cuantificable [13], pese al uso extendido de rúbricas como medio de evaluación [4]. Otro reto es integrar debidamente estas competencias en las actividades o pruebas de evaluación y asignarles un peso

adecuado ya que, en ocasiones, las competencias transversales no se consideran suficientemente cuando se plantea una actividad de evaluación o se evalúan en actividades con un valor residual en la calificación final, disminuyendo así la motivación y el esfuerzo que los estudiantes dedican a ellas.

Las dificultades anteriores, que plantean un cambio en los métodos y formas de evaluación, pueden abrumar a muchos docentes. Esto podría convertir la evaluación en un trámite, cuando en el contexto actual se trata de una oportunidad única al alcance del docente para poder observar de forma holística e integral el progreso de sus estudiantes, usando todas las herramientas a su disposición para ello.

La actividad propuesta en este trabajo pretende integrar la competencia de trabajo en equipo dentro de una prueba de evaluación sustitutiva al examen. Para ello, se proponen distintas tareas que los estudiantes realizarán de forma individual, en parejas y en gran grupo. De esta forma, se podrá:

- Construir el conocimiento y la solución a partir de la experiencia del estudiantado, activando los conocimientos asimilados y las competencias adquiridas para resolver un problema real.
- Generar responsabilidad en el proceso y en el logro, integrando tareas y actividades que impliquen un compromiso personal, la colaboración en grupos pequeños y en gran grupo para obtener la solución al problema planteado.
- Posibilitar, con todo lo anterior, la implicación de cada estudiante en su propio proceso de aprendizaje y evaluación.
- Evaluar a los estudiantes de forma más holística que cuando se lleva a cabo un examen tradicional.

2. Entorno académico

En esta sección se describe el entorno académico en el que ha tenido lugar el caso de estudio que se desarrolla en este artículo. La experiencia se ha llevado a cabo en el grado en Ingeniería Eléctrica que se imparte en la Escuela de Ingeniería Minera e Industrial de Almadén (EIMIA). La asignatura en la que se ha incorporado la actividad propuesta en este trabajo es *Ofimática aplicada a la ingeniería*. Se trata de una asignatura optativa de cuarto curso con una carga docente de 4.5 créditos ECTS y cuya actividad lectiva se desarrolla en el primer semestre.

Durante el actual curso académico 2020-2021, la asignatura ha contado con ocho estudiantes matriculados. Si bien es cierto que se trata de un grupo muy reducido, no lo es tanto en comparación con el número de estudiantes matriculados en el cuarto curso del grado, que ha sido de veinticinco estudiantes.

Al tratarse de una asignatura de ofimática, los contenidos se organizan en tres bloques fundamentales: el primero de ellos está relacionado con aspectos avanzados del uso del procesador de texto. El segundo bloque está dirigido al uso avanzado de la hoja de cálculo y el tercero está centrado en los conceptos fundamentales acerca de bases de datos. Los problemas y casos de estudio resueltos en cada bloque están directamente relacionados con el ámbito de la ingeniería eléctrica: documentos con los que se trabaja en una oficina técnica, problemas acerca de instalaciones y sistemas de energía eléctrica, gestión de datos de consumo eléctrico, etc. Microsoft Office es el paquete ofimático utilizado mayoritariamente en el transcurso de la asignatura. Esto es debido, principalmente, a dos razones: en primer lugar, que se trata del paquete ofimático más utilizado y, por tanto, el que la mayoría de los estudiantes utiliza en su día a día. En segundo lugar, que la Universidad de Castilla la Mancha (UCLM) ofrece a sus estudiantes una licencia completa de Microsoft Office. No obstante, se permite a los estudiantes utilizar cualquier otro paquete ofimático, además de darles soporte en tutorías.

Para que un estudiante supere la asignatura es necesario superar de forma individual cada uno de los bloques anteriormente mencionados. Para cada bloque, los estudiantes deben entregar una serie de prácticas y realizar una prueba práctica. Un bloque se considera superado cuando se ha obtenido una nota mayor o igual a cinco puntos tanto en las prácticas como en la prueba.

En las siguientes secciones se describe la actividad de evaluación propuesta como prueba práctica de los bloques de hoja de cálculo y bases de datos. Se ha realizado una única prueba conjunta dada la relación existente entre ellos ya que, en muchas ocasiones, la resolución de un problema real implica combinar conocimientos de ambos bloques. La prueba tiene asignado el 40 % de la calificación final de la asignatura. En el cuadro 1 se muestran los contenidos impartidos en ambos bloques. Esta información permitirá, finalizada la experiencia, valorar si la prueba propuesta ha conseguido evaluar de forma holística los contenidos de la asignatura.

3. El problema

El primer paso a la hora de plantear la prueba es elegir el problema real o hilo conductor sobre el que se desarrollará. El hilo conductor debe ser un problema que capte la atención de los estudiantes por su trascendencia y en el que puedan comprobar cómo los contenidos que han trabajado en la asignatura son claramente aplicables. Por este motivo, se han evitado problemas ambientados en universos ficticios o propios del mundo

OFIMÁTICA APLICADA A LA INGENIERÍA

Bloque II. Hoja de Cálculo

1. Importación/Exportación de datos en distintos formatos
2. Conceptos básicos. Cálculos elementales. Referencias
3. Funciones: Estadísticas, lógicas, búsqueda y referencia, fecha y hora
4. Creación y personalización de gráficos
5. Buscar Objetivo. Solver

Bloque III. Bases de Datos

1. Creación de una base de datos
2. Modelo Entidad-Relación (E-R)
3. El modelo relacional
4. Transformación E-R a modelo relacional
5. Creación y gestión de una bases de datos en Microsoft Access

Cuadro 1: Contenidos curriculares.

de los videojuegos, más comunes en actividades relacionadas con la gamificación. Dado que en clase los estudiantes ya trabajan con problemas reales del ámbito de la ingeniería eléctrica, se optó por ambientar la prueba en un problema más general o transversal.

Con este criterio y, aprovechando que la fecha de la prueba coincidió aproximadamente con las elecciones a la presidencia de los Estados Unidos, se eligió este tema tratando de huir de cualquier problema relacionado con la COVID-19, el cual es demasiado recurrente en el día a día de los estudiantes.

Por tanto, el hilo conductor de la prueba consistió en, dados una serie de datos que representan los votos emitidos en una hipotética ciudad americana, determinar qué partido ha ganado las elecciones en ella. Por simplicidad, el procedimiento de obtención del ganador ha sido suficientemente simplificado para adaptarlo a los contenidos evaluables y al tiempo disponible. A continuación, se describen cada una de las tareas que se incluyeron en la prueba.

3.1. Los datos: implementación de un simulador

En primer lugar, los estudiantes debían importar a la hoja de cálculo un fichero de datos que contenía los votos emitidos durante las elecciones. Para ello, el profesorado implementó un simulador de datos con el objetivo de que los estudiantes trabajasen con un número de datos considerable, tal y como suele ocurrir en cualquier problema real. Así, dada una hipotética ciudad con 12000 habitantes, el fichero de datos generado por el simulador disponía de un total de 10000 filas,

cada una de las cuales correspondía con un voto emitido, y cuatro columnas. Los valores contenidos en cada columna y su semántica asociada se describen a continuación.

- ID_Voto: Número entero que identifica de forma inequívoca cada voto emitido.
- Mesa electoral: Número entero que identifica la mesa electoral en la cual ha sido depositado cada voto. Hay cuatro mesas electorales identificadas, respectivamente, con los números 1, 2, 3 y 4.
- Género: Sexo de la persona que emitió el voto. El número 0 identifica los votos emitidos por hombres mientras que el 1 identifica los votos emitidos por mujeres.
- Voto: Número entero que describe el sentido del voto emitido. El valor entero 1 se refiere a votos para el partido demócrata y el valor 2 a votos para el partido republicano. Por otra parte, los valores 3 y 4 hacen referencia, respectivamente, a votos en blanco y nulos.

3.2. El escrutinio: análisis de los votos emitidos

Una vez que los datos han sido importados a la hoja de cálculo, el siguiente paso consiste en el escrutinio de los votos emitidos. Para ello, en primer lugar, se escrutarán los votos de cada una de las mesas. Después, y en base a los resultados obtenidos en cada una de ellas, se determinará cuál ha sido el partido ganador de las elecciones.

Una figura presente en cualquier proceso electoral es la del interventor. Se trata de una persona representante de una candidatura que puede asistir a una mesa y participar en la deliberación, supervisando que el proceso de recuento de votos se realiza de forma lícita. Así, para cada una de las mesas electorales, existirá un interventor/a de cada uno de los partidos candidatos (demócrata y republicano). Cada interventor se encargará de obtener los siguientes datos sobre los votos emitidos a su partido en la mesa en la que se encuentra acreditado:

1. ¿Cuántos votos ha obtenido el partido al que representas en la mesa electoral en la que estás acreditado/a?
2. ¿Cuántos hombres han votado al partido que representas en la mesa electoral en la que estás acreditado/a? ¿Y cuántas mujeres?
3. ¿Cuántos votos en blanco se han emitido en la mesa electoral en la que estás acreditado/a? ¿Y cuántos votos nulos?
4. Construye los gráficos que consideres necesarios para visualizar los resultados obtenidos.

Después de que los interventores hayan supervisado

el recuento de votos emitidos para su partido en cada una de las mesas, es posible obtener el resultado final en cada mesa electoral. De esta forma, se cruzarán los datos obtenidos por cada interventor en cada una de las mesas para obtener el ganador. Las tareas a realizar en esta etapa son:

5. ¿Qué porcentaje de hombres ha votado a cada partido con respecto al total de votos emitidos, incluyendo votos en blanco y votos nulos? ¿Qué porcentaje de mujeres?
6. ¿Qué porcentaje de votos ha obtenido cada partido con respecto al total de votos emitidos, incluyendo votos en blanco y votos nulos?
7. Determinar si ha habido o no un partido ganador en una mesa electoral. Se dice que un partido ha ganado las elecciones en una mesa si ha obtenido una diferencia de votos mayor o igual al 5 % con respecto al otro partido candidato.
8. Visualizar los resultados obtenidos mediante los gráficos que se consideren más apropiados.

3.3. El ganador: un problema de optimización y una base de datos

Partiendo de los datos escrutados en cada una de las mesas electorales, es posible saber cuál ha sido el partido ganador de las elecciones. El resultado R de las mismas se puede calcular tal y como se muestra en la ecuación **I**

$$R = [(V_d - 2 \cdot V_b)^2 \cdot V_b] - [(V_r - 2 \cdot V_n)^2 \cdot V_n] \quad (1)$$

Donde V_d son los votos recibidos por el partido demócrata, V_r los del partido republicano, V_b los votos en blanco y, finalmente, V_n hace referencia a los votos nulos. Como se puede ver, los votos en blanco afectan al partido demócrata mientras que los votos nulos afectan al partido republicano. El ganador de las elecciones se puede conocer fácilmente: si R es positivo, el ganador ha sido el partido demócrata, mientras que si es negativo, el ganador ha sido el partido republicano. En caso de que $R = 0$, habría un empate técnico entre los dos partidos candidatos. Conocida esta información, las tareas propuestas para la obtención del ganador son:

9. Calcular el valor de R y determinar quién ha sido el partido ganador de las elecciones.
10. Calcular el número de votos en blanco o nulos - según el partido al que se represente - para que gane las elecciones el partido al que se representa. Resolverlo por tanteo, mediante buscar objetivo y Solver.

Finalmente, después de todo el proceso, se pedía a los estudiantes realizar las siguientes tareas:

11. Diseñar una base de datos para almacenar la información que se ha manejado sobre el proceso

electoral mediante un diagrama E-R.

12. Transformar el diagrama E-R a modelo relacional.
13. Crear en Microsoft Access la base de datos generada en los apartados anteriores.

4. La dinámica

Esta sección analiza cuál fue la dinámica seguida por los estudiantes durante la realización de la prueba. Cabe destacar que, durante las sesiones de clase, los estudiantes resuelven y entregan prácticas en las que, como en la dinámica que se propone a continuación, deben intercambiar y contrastar datos entre ellos, trabajando en equipo para entregar la solución final de un problema.

La experiencia se llevó a cabo durante una sesión ordinaria de clase en el aula habitual. Para garantizar que los estudiantes dispusieran de tiempo suficiente, se acordó con ellos en comenzar la sesión una hora antes del horario habitual. Por tanto, los estudiantes dispusieron de un total de tres horas: las dos horas de la sesión y la hora extra anteriormente mencionada.

4.1. Fase individual: los interventores

En primer lugar, era necesario determinar qué estudiantes serían los interventores de cada partido en cada una de las mesas electorales. Para esto, el profesorado implementó un programa que, aleatoriamente, generaba para cada uno de los estudiantes un partido político al que representar (demócrata o republicano) y una mesa electoral de la cual se encargarían de escrutar los resultados. En la sección anterior se indicó que había cuatro mesas electorales. Esta decisión se tomó para que, una vez asignados dos interventores a cada mesa, se obtuviera el total de ocho estudiantes que participaron en la prueba.

Asignados los interventores y repartidas las mesas electorales, los estudiantes comenzaron a resolver las tareas 1, 2, 3 y 4 especificadas en la sección anterior. Para ello, se les proporcionó una plantilla como la mostrada en la Figura 1, de modo que todos ellos obtuvieran los resultados y los estructuraran de la misma forma, lo que facilitaría la siguiente etapa de la prueba.

4.2. Fase en parejas: escrutinio de las mesas electorales

A medida que los interventores de cada una de las mesas terminaban las primeras cuatro tareas, podían comenzar a trabajar en parejas de forma conjunta para obtener los resultados de la mesa electoral. Así, lo primero que debían hacer era cruzar los resultados obtenidos en la primera etapa. Con esos datos, los interven-

FASE 1: Escrutinio Interventores	
Mesa electoral N°:	Rellenar con el número de mesa asignada
Partido:	Rellenar con el nombre del partido al que se representa
Categoría	Nº de Votos
Hombres	
Mujeres	
Blancos	
Nulos	
Total	

Figura 1: Plantilla para la obtención de resultados en la fase individual.

tores de cada mesa podrían obtener la respuesta a las preguntas 5, 6, 7 y 8 y contrastar si ambos llegaban a la misma solución para, efectivamente, corroborar que el escrutinio de la mesa era correcto. Para esta etapa se proporcionó a los interventores de cada mesa una plantilla como la mostrada en la Figura 2.

Como se puede observar, el trabajo realizado en esta fase es asíncrono, ya que no todos los estudiantes comienzan a resolver las tareas de esta fase a la misma vez, sino a medida que han finalizado la fase individual. Sin embargo, este hecho no obstaculiza ni dificulta a quienes terminan antes ni tampoco a quienes terminan después, ya que la obtención de los resultados de una mesa electoral es independiente del resto.

Por otra parte, el trabajo en parejas trae consigo una doble dificultad: en primer lugar, el problema de monitorizar el trabajo de los miembros de la pareja para comprobar si realmente cada estudiante ha adquirido los conocimientos necesarios o alguno de ellos se ha beneficiado de las capacidades de su compañero/a [3]. En segundo lugar, la problemática asociada a la COVID-19, que no permite que los estudiantes trabajen de forma física en parejas para que se pueda mantener la distancia de seguridad de dos metros entre ellos.

Para afrontar estas limitaciones, la idea original acordada con los estudiantes era crear tantos grupos de chat como mesas electorales. Así, en cada uno de estos grupos participarían los dos interventores de cada mesa y siempre el profesor, con el objetivo de dejar registrada la comunicación de los interventores en el proceso de resolución. De esta forma, el profesor podría consultar, durante la prueba y después, el historial de la conversación entre los interventores de cada mesa. Finalmente, y dado que al comienzo de la prueba los estudiantes asignados a cada mesa decidieron sentarse uno al lado de otro, además de que solo eran cuatro parejas, no se utilizaron los chats y se permitió que, sin abandonar sus sitios para respetar la distancia de seguridad, pudieran hablar entre ellos para resolver la

fase en parejas. La comunicación en esta etapa consiste, por tanto, no solo en el intercambio de los datos para la obtención del resultado electoral de la mesa, sino en el diálogo entre los interventores para contrastar y validar los resultados que están obteniendo, permitiéndoles rectificar a tiempo.

No obstante, esta decisión de última hora no fue un impedimento para que los estudiantes trabajaran concentrados, ya que al tratarse de cuatro parejas no se formó un ruido que pudiese distraer a los compañeros. Además, la característica asincrónica de esta fase de la prueba favoreció a que los estudiantes pudieran trabajar de forma efectiva. Respecto a la labor del profesor, no fue un problema supervisar el trabajo de las parejas para cerciorarse del nivel de conocimientos adquiridos por cada estudiante, gracias al carácter asincrónico de esta etapa y al reducido número de parejas.

FASE 2: Escrutinio Mesa Electoral					
Mesa electoral n°		Rellenar con el número de la mesa electoral			
	Demócrata	Republicano	TOTAL	% Demócrata	% Republicano
Hombres					
Mujeres					
Blancos					
Nulos					
TOTAL					
%					
Ganador Mesa:					

Figura 2: Plantilla para la obtención de resultados de la fase en parejas.

4.3. Fase grupal: obtención del ganador y creación de la base de datos

En última instancia, y para determinar finalmente el partido ganador de las elecciones en la ciudad, los estudiantes debían compartir los datos del escrutinio de las cuatro mesas electorales que obtuvieron en la etapa anterior de la prueba. Con estos datos, podían comenzar a resolver las tareas 9 y 10. La idea original de la fase grupal era gestionar la comunicación entre los estudiantes mediante un grupo de chat en el que estuviera incluido el profesor. Finalmente, y tal y como sucedió en la fase anterior, los estudiantes pudieron comunicarse sin abandonar sus puestos de trabajo para contrastar los resultados obtenidos en esta fase.

Por tanto, y de acuerdo a la descripción de las tareas 9 y 10, los interventores del partido demócrata tuvieron que modelizar y resolver un problema de optimización, mientras que los interventores del partido republicano debían modelizar y resolver otro. Así, los estudiantes pudieron contrastar los resultados obtenidos, sin compartir información sobre el modelo que habían construido, las restricciones definidas, etc.

Una vez resueltas estas tareas, los estudiantes co-

menzaron a acometer, ya de forma individual, las tareas 11, 12 y 13 relacionadas con el bloque de bases de datos. Merece la pena destacar que, cuando los estudiantes comenzaron estas tareas, muchos de ellos ya habían trabajado anteriormente en ellas. Esto es así, ya que en los tiempos de espera en las fases anteriores fueron avanzando con las tareas de bases de datos. De esta forma, se consiguió aprovechar los tiempos de espera en la medida de lo posible.

5. Desarrollo, resultados y evaluación

En esta sección se pretende poner de manifiesto los resultados obtenidos por los estudiantes en la prueba, así como las impresiones y la evaluación que tanto estudiantes como profesorado hicieron sobre la misma.

El desarrollo de la prueba transcurrió con normalidad. Dado que los estudiantes disponían de tres horas para su realización, se estableció una horquilla orientativa de una hora de tiempo para cada una de las fases. No obstante, los estudiantes podían gestionar el tiempo de cada fase como considerasen más oportuno.

El único imprevisto de la prueba se produjo al inicio de la misma, después de haber leído el enunciado con los estudiantes y definidas las tareas y la dinámica. Después de transcurridos apenas quince minutos de la fase individual, un estudiante decidió abandonar la prueba alegando que no era capaz de resolver las tareas que se proponían. Aunque esto podía suponer un contratiempo, la estudiante que realizaría la fase en parejas con él no tuvo inconveniente en realizar la fase individual correspondiente a los dos partidos políticos de su mesa electoral. Si bien es cierto que, una vez resueltas estas tareas para un partido político, es fácil adaptar las funciones y fórmulas construidas para el otro (y para evitar este tipo de imprevistos se diseñaron así), el profesorado agradeció la buena disposición de la estudiante. Sin haberlo planificado, este contratiempo sirvió para constatar que los estudiantes habían entrado en la dinámica e involucrado en buena medida, ya que se ofrecieron a ayudarla con independencia de sus tareas. Finalmente, ella sola realizó la fase individual correspondiente al compañero que abandonó.

Con respecto a la calificación de la prueba, cada fase suponía 1/3 de la nota final de la misma, con el aliciente de que, tanto en la fase en parejas como en la fase grupal, la calificación numérica del profesorado sobre la actitud y el trabajo en equipo suponía la mitad de la puntuación de cada una de estas fases. Esta puntuación ha sido otorgada en una escala de 0 a 3 puntos, valorando de forma cualitativa en qué medida los estudiantes exponían sus resultados al resto de compañeros y contrastaban los resultados obtenidos entre ellos sin

Pregunta	Respuesta				
	1	2	3	4	5
	Mucho peor que un examen			Mucho mejor que un examen	
Considero que la prueba ha permitido evaluar los contenidos de la asignatura	0	0	3	0	5
Considero la evaluación mediante actividades alternativas	0	0	0	5	3
	Muy insuficiente			Había tiempo de sobra	
Con respecto a la duración de las pruebas, considero que ha sido	0	0	1	3	4
	No hay ninguna relación entre asignatura y vida real			Hay una gran relación entre asignatura y vida real	
Con respecto a los problemas planteados, considero que	0	0	0	1	7
	Muy inadecuada			Muy adecuada	
Con respecto a las tareas, considero su dificultad	0	0	0	4	4
Considero que incluir tareas individuales, realizadas en parejas y en gran grupo es	0	0	1	3	4
	Muy insatisfecho la actividad y los resultados			Muy satisfecho con la actividad y los resultados	
Después de realizar la prueba, me siento	0	0	0	3	5

Cuadro 2: Resultados de la encuesta de satisfacción realizada a los estudiantes.

llevarlos a error. En cuanto a las notas obtenidas, la mínima fue un 6, la máxima un 9 y la media sin contar con el no presentado del estudiante que abandonó fue de 7,75. Es posible que la naturaleza incremental de la prueba y el trabajo en grupos que permite contrastar los resultados obtenidos entre los estudiantes, posibilite el correcto avance en la prueba y sea un factor positivo para que las notas sean buenas y no haya una variabilidad alta.

Para evaluar la percepción de los estudiantes sobre la prueba, se les pidió que rellenasen una encuesta cualitativa compuesta por siete preguntas con respuesta de escala Likert entre 1 y 5 que fue contestada después de conocer las notas de la prueba. El cuadro 2 muestra los resultados de la encuesta, indicando el número de estudiantes que contestó cada respuesta. En cuanto al

profesorado, al tratarse de un solo profesor, se le pidió que redactase un informe con sus impresiones al finalizar la prueba.

Según la valoración de los estudiantes, la experiencia ha sido positiva; la duración de las tareas y la dificultad fueron adecuadas y han podido constatar claramente la relación de los contenidos de la asignatura con problemas del mundo real. Además, señalan que este tipo de actividades alternativas permite evaluar al estudiante igual o mejor que un examen tradicional. Finalmente, los estudiantes han señalado sentirse satisfechos con la evaluación y las notas obtenidas en igual medida que cuando se les evalúa mediante exámenes o incluso más.

Respecto a la evaluación del profesor, se señaló que durante la prueba se creó un ambiente de aprendizaje

y evaluación frente al ambiente generalmente tenso de un examen. Además, el profesorado se sintió satisfecho con la prueba, que permitió a los estudiantes poder contrastar los resultados durante esta para poder rectificar en caso de error y evaluar su capacidad de diálogo y trabajo en equipo para este fin.

6. Conclusiones

En este trabajo se ha descrito una experiencia positiva de diseño de actividades alternativas al examen. Las evaluaciones recogidas tanto de los estudiantes como del profesorado han permitido constatar y dar título a este artículo. Así pues, esta experiencia pretende ser un reporte positivo de que la evaluación mediante la resolución de problemas reales y la integración de dinámicas reales de trabajo en equipo y colaboración, de las que los estudiantes formarán parte cuando se incorporen a un equipo de trabajo, permiten evaluarlos de forma más holística y los predispone positivamente a la evaluación, haciendo de ella una oportunidad para el aprendizaje.

Esta experiencia quiere también ser un aliciente para incentivar a los docentes a bucear en su creatividad y aprovechar la oportunidad de aprendizaje que puede suponer la evaluación, utilizando para ello problemas y dinámicas reales donde es más fácil integrar competencias transversales para evaluar de forma holística a los estudiantes. Así, es posible favorecer la creación de experiencias de evaluación que también sirvan para aprender y donde los estudiantes puedan contrastar resultados entre sí para no arrastrar fallos. A pesar del trabajo que puede suponer diseñar la actividad, las evaluaciones positivas reportadas en este artículo pueden animar a intentarlo, compartiendo experiencias realizadas en grupos pequeños, medianos y grandes. En este sentido, el diseño planteado en este trabajo se puede escalar a grupos grandes, donde la comunicación se monitorice por parte del profesor a través de grupos de chat y donde los archivos se compartan a través de un espacio de almacenamiento corporativo institucional.

Referencias

- [1] John Biggs: *The reflective institution: Assuring and enhancing the quality of teaching and learning*. Higher Education, 41:221–238, 2001.
- [2] John Biggs y Catherine Tang: *Teaching For Quality Learning At University*. McGraw-Hill Education, 2011, ISBN 9780335242757.
- [3] Manuel Enciso, Carlos Rossi y Eduardo Guzmán: *Evaluación del trabajo en grupo: ¿café para todos?* En *Actas de las XVII Jornadas de Enseñanza Universitaria de Informàtica, Jenui 2011*, páginas 91 – 98, Sevilla, julio 2011.
- [4] John Hafner y Patti Hafner: *Quantitative analysis of the rubric as an assessment tool: an empirical study of student peer-group rating*. International Journal of Science Education, 25(12):1509–1528, 2003.
- [5] David López: *Si elimino el examen ¿mis alumnos dejarán de aprender? Una experiencia de diseño de actividades educativas alternativas al examen*. Revisión, 6(2):10 – 17, 2013.
- [6] David López y David Carrera: *Si elimino el examen, ¿cómo evalúo?: Una discusión sobre las actividades sustitutivas del examen y su escalabilidad*. Revisión, 9(3):65 – 73, 2016.
- [7] David López y David Carrera: *Viabilidad y escalabilidad de actividades substitutivas del examen: un caso práctico*. En *Actas de las XXII Jornadas de Enseñanza Universitaria de Informàtica, Jenui 2016*, páginas 31 – 38, Almería, julio 2016.
- [8] David López y Joe Miró: *Creencias que merecen una reflexión*. Revisión, 7(3):61 – 68, 2014.
- [9] Julio Alberto López-Gómez, Raquel Rodrigo Rubio, Ángela Carrasco García y Francisco Mata Cabrera: *La Escape Room como actividad de formación, evaluación y mejora de la motivación del estudiante: una experiencia en Ofimática Aplicada a la Ingeniería*. En *Actas de las XXVI Jornadas de Enseñanza Universitaria de Informàtica, Jenui 2020*, páginas 237 – 244, Valencia, julio 2020.
- [10] Alma María Pisabarro Marrón y Carlos Enrique Vivaracho Pascual: *Gamificación en el aula: gincana de programación*. En *Actas de las XXIII Jornadas de Enseñanza Universitaria de Informàtica, Jenui 2017*, páginas 39 – 46, Cáceres, julio 2017.
- [11] Richard M. Ryan y Edward L. Deci: *Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions*. Contemporary Educational Psychology, 25(1):54–67, 2000, ISSN 0361-476X.
- [12] Maria José Sá y Sandro Serpa: *Transversal Competences: Their Importance and Learning Processes by Higher Education Students*. Education Sciences, 8(3), 2018, ISSN 2227-7102.
- [13] Fermín Sánchez Carracedo, Alicia María Ageno, Joan Aranda, José Cabré, Joan Climent, Karina Gilbert, Joan Carles Gil y Carme Martín Escofet: *Estudio cuantitativo del nivel de competencias transversales de los estudiantes del Grado en Ingeniería Informática*. En *Actas de las XXIV Jornadas de Enseñanza Universitaria de Informàtica, Jenui 2018*, páginas 157 – 164, Barcelona, julio 2018.

Programación de Dispositivos Móviles: una propuesta docente adaptada a un entorno híbrido

Antoni Perez-Poch

Departament de Ciències de la Computació
Escola d'Enginyeria Barcelona Est (EEBE)
Universitat Politècnica de Catalunya (UPC)

antoni.perez-poch@upc.edu

Resumen

Se presenta una experiencia docente destinada a la enseñanza de Programación de Dispositivos Móviles en un entorno universitario de Grados de Ingeniería distintos al de Informática en el curso 2020/21. Se ha realizado el diseño, puesta en marcha, implementación y evaluación de una asignatura optativa de 3º/4º curso para estudiantes que han cursado previamente solo una asignatura de 6 créditos de Fundamentos de Informática. La experiencia supuso un reto importante ya que la mayoría de alumnos no tenían más experiencia previa de programación que esta asignatura obligatoria, insuficiente para asumir el desarrollo de un proyecto completo de programación. Además, las medidas de confinamiento por razón de la pandemia de Covid-19 han dificultado enormemente la realización de clases presenciales, teniendo que realizar parte de la docencia virtualmente. Se realizó el diseño del curso basándose en las técnicas de Aprendizaje Basado en Proyectos (ABP) y Aprendizaje Cooperativo (AC), además de digitalizar en el entorno virtual de clase todos los contenidos y actividades. Se muestran los resultados obtenidos, muy positivos, tanto en rendimiento académico como en satisfacción del alumnado. Como parte de su formación, los estudiantes trabajaron en una app de simulación Covid-19 que reproduce un modelo de propagación de la pandemia y que fue valorada muy positivamente.

Abstract

A new teaching experience is presented about a subject of Mobile Devices Programming in a Non-Informatics environment of Engineering Degrees in the 2020/21 academic year. The design, implementation, and assessment of an elective course are shown. Students who have previously taken only a 6-credit course in Fundamentals of Programming undertook the teaching. The experience was an important challenge as most students had no other previous programming experience than this

compulsory subject, which is insufficient to take on the development of a complete programming project. In addition, confinement measures due to the Covid-19 epidemic have made it extremely difficult to conduct face-to-face classes, leading to virtual teaching for a part of the term. The course is based on the techniques of Project-Based Learning (PBL) and Cooperative Learning (CL), in addition to digitizing all content and activities in the virtual classroom environment. The results obtained are very positive, both in academic performance and in students' satisfaction. As part of their training, students worked on a Covid-19 simulation app that reproduces some of the epidemic spread models which was rated very positively.

Palabras clave

Programación, dispositivos móviles, aprendizaje basado en proyectos, aprendizaje cooperativo, enseñanza virtual, innovación docente.

1. Motivación

La Programación de Dispositivos Móviles (PDM) es una importante oportunidad para motivar a los estudiantes a mejorar sus habilidades de programación, especialmente en aquellos Grados de facultades no informáticas. En estas titulaciones la formación básica en Ciencias de la Computación depende de los planes de estudio, y en general resulta claramente insuficiente. En particular, en algunos grados de Ingeniería distintos al de Informática, los estudiantes tienen un número limitado de créditos obligatorios de formación en Informática. Todos los estudiantes tienen al menos un dispositivo móvil o portátil con el cual conviven habitualmente más allá de sus tareas estrictamente académicas y es en esta situación donde se dispone de una oportunidad privilegiada para motivarles a programar. En el presente curso académico 2020/21 se ha impartido en la Escuela de Inge-

niería Besós Este (EEBE) de la *Universitat Politècnica de Catalunya* (UPC) una asignatura optativa de Programación de Dispositivos Móviles con un temario totalmente renovado. Se ha utilizado MIT App Inventor y Android Studio para la realización de las prácticas. Además, la incidencia directa de la pandemia de Covid19 ha supuesto compaginar clases presenciales con virtualidad en un entorno híbrido cambiante en función de las restricciones de movilidad existentes a cada momento.

Se trataba por tanto de una experiencia nueva de docencia de alto riesgo. Los resultados han sido, no obstante, altamente positivos.

En el siguiente apartado se describe el contexto de la experiencia, los entornos de programación utilizados y las metodologías docentes. En el apartado 3 indicamos las adaptaciones realizadas por la pandemia Covid19. En el apartado 4 mostramos un ejemplo de práctica del curso contextualizada. Posteriormente se exponen los resultados de la evaluación de la experiencia y las conclusiones.

2. Contexto de aprendizaje

La asignatura de Programación de Dispositivos Móviles es una optativa de 3º o 4º curso y se puede cursar en las distintas especialidades (Eléctrica, Electrónica y Automática, Química, Mecánica, Materiales, Biomédica y Energía) de los Grados de Ingeniería que se imparten en la Escuela de Ingeniería Barcelona Este (EEBE) de la *Universitat Politècnica de Catalunya* (UPC). Sus objetivos de aprendizaje son:

- Familiarizar al alumno con los conceptos y métodos básicos de la programación en dispositivos móviles (teléfonos y tabletas) con entorno Android.
- Proporcionar las técnicas básicas de programación en dispositivos móviles.

Como competencia transversal se trabaja la emprendeduría e innovación. El alumno deberá utilizar conocimientos y habilidades estratégicas para la creación y gestión de proyectos, aplicar soluciones sistémicas a problemas complejos, diseñar y gestionar la innovación.

No se requieren requisitos previos en esta asignatura, con lo que el nivel inicial de los alumnos de esta materia puede ser muy variado.

Se presuponen únicamente los conocimientos adquiridos en la asignatura obligatoria de primer curso de estas titulaciones “Informática” de 6 créditos ECTS. Esta asignatura proporciona unos fundamentos básicos de Programación para la Ingeniería, utilizándose el lenguaje Python [3]. Los estudiantes pueden

haber cursado otras asignaturas optativas de Informática durante sus estudios, pero en general no llegan a esta asignatura con la suficiente madurez para emprender un proyecto completo de programación como se ve en la evaluación inicial que se menciona a continuación.

La asignatura tiene 6 créditos ECTS, cubriendo 60 horas de clase de Laboratorio de Informática (en grupos de máximo 25 alumnos) y 90 horas de aprendizaje autónomo. En la edición objeto de este trabajo el cuatrimestre de otoño 2020-21 se matricularon 27 alumnos, 21 varones (78%) y 6 mujeres (22%) entre 20 y 28 años, en dos grupos T1 y T2, de 13 y 14 alumnos respectivamente. La asignatura se ha impartido con anterioridad ininterrumpidamente en la EEBE desde 2016 por distinto profesorado del Departamento de ciencias de la computación de la UPC, aunque con orientación y entorno de desarrollo totalmente distintos. Se ha realizado, por tanto, una renovación y actualización total de los contenidos de la materia en esta edición.

Se realizó una evaluación inicial en las primeras sesiones de clase mediante un cuestionario escrito. Todos los alumnos habían cursado previamente al menos 6 créditos de una asignatura de Fundamentos de Informática o formación equivalente. Solo 3 alumnos (11%) manifestaron tener conocimientos medios o avanzados de programación obtenidos fuera del plan de estudios de la EEBE o bien en su experiencia profesional. No hay alumnos repetidores en esta edición de la asignatura.

2.1. Talleres iniciales

La asignatura tiene una primera parte que dura nueve semanas de curso (de un total de quince) en la que se realizan prácticas guiadas de programación. Esta parte es esencial para que todos los alumnos lleguen a adquirir unas nociones básicas de programación que les permitan afrontar el proyecto propuesto en la segunda parte del curso. En estos talleres los alumnos tienen que entregar periódicamente y de forma individual una serie de entregables con los que aprenden las técnicas básicas de programación de dispositivos móviles en los entornos MIT App Inventor 2 y Android Studio. Hay un total de 15 prácticas guiadas que se evalúan para dar un 30% de la nota final de la asignatura.

MIT App Inventor 2 es un entorno de programación visual, gratuito y fácil de utilizar. Tiene una curva de aprendizaje rápida y numerosos recursos disponibles online¹. Con la decisión de usarlo en la primera parte del curso se pretende facilitar que los estudiantes repasen los conceptos y técnicas de pro-

¹ Entorno de Programación MIT App Inventor 2
<https://appinventor.mit.edu/explore/ai2>

gramación que muchos ya han olvidado. Como se ha hecho evidente durante el curso, esta herramienta permite conseguir que el estudiante tenga un porcentaje elevado de éxito al programar sus primeras aplicaciones de móvil, fomentando su implicación en el curso [14] para posteriormente abordar tareas más complicadas. Entre muchas experiencias educativas de uso de este entorno publicadas en diversos niveles, destaca la aportación de Sergio Barrachina y Germán Fabregat [1] quienes han desarrollado un taller de programación de móviles para la enseñanza de Estructura de Computadores.

Android Studio² es el entorno de desarrollo profesional de referencia para desarrollar aplicaciones Android para dispositivos móviles. Está basado en el lenguaje de programación Java y IntelliJ IDEA. Se considera importante introducir esta herramienta para que los alumnos conozcan cómo se desarrollan los proyectos profesionales, aunque no es obligatoria para realizar el posterior proyecto final. Se introduce en una primera clase los conceptos básicos de programación orientada a objetos, para pasar enseguida a proporcionar ejemplos sencillos de código que hay que rellenar o modificar para efectuar las primeras aplicaciones de móvil. Ambos entornos de programación son exclusivos para dispositivos móviles basados en Android. Si algún estudiante deseara programar para iPhone o iPad (iOS) se ofrece asesoramiento para programar con el entorno Xcode de Apple.

2.2. Proyecto de programación

En la segunda parte del curso los alumnos realizan una práctica final de programación que consiste en desarrollar una aplicación para dispositivo móvil (app) en entorno Android. Esta parte dura las cinco semanas finales del cuatrimestre para finalizar su trabajo y presentarlo en clase. El proyecto de programación debe realizarse en grupo de 2 o 3 estudiantes, y se evalúa con un 70% de la nota final. Hay una entrega parcial que supone un 25% y una entrega final que vale el 45% restante de la nota. En la primera semana del proyecto los estudiantes entregan una propuesta de trabajo que es evaluada y en su caso corregida y eventualmente aceptada por el profesor. Esta entrega inicial es obligatoria y forma parte de la nota de la entrega parcial. El proyecto se evalúa mediante una rúbrica de evaluación y la autoevaluación del grupo. Este proyecto es de elección libre por parte del alumno, y todos los grupos realizan una app distinta. Como ejemplo, uno de los grupos realizó como proyecto final una ampliación de la práctica descrita en el apartado 4.

2.3. Aprendizaje cooperativo

El Aprendizaje cooperativo (AC) es una técnica de enseñanza-aprendizaje bien conocida tanto en entornos académicos universitarios como en otros niveles y contextos [4]. David W. Johnson y Roger T. Johnson [6] describieron qué condiciones necesarias deben darse para que haya realmente una cooperación entre los integrantes de un grupo: (1) interdependencia positiva, (2) interacción directa cara a cara, (3) responsabilidad personal y (4) un proceso de desarrollo de la actividad que haga que su éxito dependa necesariamente de todos y cada uno de sus integrantes. En [13] se introdujo un modelo de aplicación sistemática de esta técnica en la docencia universitaria, que aquí se aplica a esta nueva asignatura.

En el desarrollo de este proyecto final de la asignatura los grupos serán formales, formados en principio a partir de afinidades entre los estudiantes y de entre 2 y 3 personas. El eje de actividad es la aplicación de los conocimientos adquiridos en la primera fase del curso a una realización original de una aplicación de móvil. Se establecen dos reuniones periódicas durante la semana de las cuales es necesario elevar actas y comunicarlas al profesor que hace el seguimiento.

Hay, como mínimo, tres entregas, una al inicio, otra a final del período lectivo y otra final donde los estudiantes presentan a la clase su código y una demostración del trabajo. Los estudiantes reciben *retroalimentación* en cada una de las entregas, así como un seguimiento personalizado. Es posible para proyectos singulares disponer además de un consultor externo que puede venir del mundo académico o de la industria de los videojuegos.

2.4. Aprendizaje basado en proyectos

Esta técnica también es bien conocida por la comunidad universitaria interesada en la innovación docente. Sin embargo, no es una metodología fácil. Miguel Valero, por ejemplo, describe en [11] sus ventajas y también las dificultades más habituales con que el docente se encuentra a la hora de aplicarla, sobre todo si es la primera vez. La gran ventaja que presenta ABP es que motiva más a los alumnos. Además, permite trabajar el desarrollo de competencias transversales profesionales y tiende a producir mejores rendimientos académicos. Sin embargo, hay que mencionar sus posibles lagunas: ¿nos preocupa si se va a cubrir el temario? y ¿el trabajo en grupo esconde lagunas en algunos estudiantes? Quizás sí, pero si se aplica bien la técnica anterior de Aprendizaje Cooperativo, entonces esto no debería ocurrir, o como mínimo las incidencias deberían aflorar rápidamente [12]. En esta asignatura no cubrimos toda la materia

² Android Studio developer <https://developer.android.com>

con el desarrollo único de un proyecto. Nos basamos en la máxima del profesor Paul A. Kirschner [7] que afirma que «sin una mínima guía no funcionará». Y es que la evaluación inicial a principio de curso mostró que la mayoría de estudiantes no tenían, o no recordaban, los conocimientos que se consideran suficientes para abordar este tipo de proyecto desde el inicio. No obstante, el proyecto cuenta un 70% de la nota, por tanto, es inevitable que los estudiantes tengan que abordarlo para poder aprobar. Finalmente, una asignatura optativa donde los estudiantes quieren aprender a realizar una app para usarla en su propio dispositivo móvil parece un escenario ideal para llevar a cabo este tipo de proyectos. Deberíamos no acabar rompiendo esta motivación. El reto será, por tanto, ofrecer un trampolín suficientemente apto para que los equipos de estudiantes que trabajen lo suficiente lleguen a buen puerto y acaben el curso con una app instalada en su móvil; y de paso obtengan unos cuantos recursos más en su bolsa de trucos de ingenieros que se va a ir llenando a lo largo de sus estudios.

3. Adaptación a un entorno híbrido

Como se ha comentado anteriormente, al reto de iniciar una asignatura nueva se le añade la incertidumbre derivada de la pandemia de Covid19 durante el cuatrimestre de otoño 2020. El planteamiento inicial, basado en la normativa de docencia aprobada por la UPC al inicio de este curso 2020-21, de acuerdo con las reglas sanitarias establecidas por la Generalitat de Catalunya, era de un contexto híbrido de docencia. La primera novedad es que los grupos de clase habitualmente de 25 estudiantes se redujeron a un máximo de 15 alumnos por razones de distanciamiento en el aula. La asignatura tiene dos sesiones semanales de clase de dos horas cada una. Se optó para esta asignatura por una mitad de horas de clase presenciales en el laboratorio de informática, con medidas de seguridad sanitaria, y otra mitad de horas de clase a distancia. En las clases presenciales se registró la asistencia y su puesto de trabajo con fines preventivos. Las clases a distancia mediante Google Meet fueron grabadas y quedaron a disposición del estudiante para su repaso de forma asíncrona. A partir de la séptima semana, y ante la evolución desfavorable de las condiciones de la pandemia en Cataluña se pasó a docencia totalmente virtual hasta el final del curso. La docencia virtual se realiza mediante el entorno Google Meet de videoconferencia y chat. En los dos periodos la exigencia (temario a cubrir en los talleres, entregables) no se redujo en absoluto.

La asistencia síncrona a clase fue máxima en las primeras semanas y posteriormente se fue reduciendo. Sin embargo, los entregables realizados a lo largo del curso no reflejaron esta falta de sincronía, ni en

calidad ni en cantidad de los trabajos entregados. Seguramente, la experiencia se benefició de que los alumnos pudieran tener una experiencia de socialización e interacción en clase para las primeras semanas del cuatrimestre.

4. Contextualización: práctica de modelización de la pandemia

Un curso como este, impartido en unas condiciones sanitarias excepcionales y además cambiantes de una semana a otra, necesariamente debería tener medidas de contextualización de la asignatura.

Hay muchos estudios, por ejemplo, el de Dana A. Robertson y Christopher J. Padesky [10], que inciden en las ventajas de contextualizar la enseñanza a temas que interesen a los estudiantes y sean próximos a ellos. Entre estas ventajas destacan un mayor interés y relevancia de los contenidos de aprendizaje. Estas medidas para que los estudiantes no desconecten son especialmente importantes en un tiempo en que su acceso presencial a las aulas está restringido.

No existe un tema que preocupe más ahora, incluso por encima de las notas, que la evolución de la pandemia de Covid19. Por este motivo, se sustituyó una de los últimos ejercicios guiados de la primera parte del curso por la realización de una aplicación de móvil con Mit App Inventor 2 que simula la propagación de contagios debidos a la pandemia. La aplicación se basa en la práctica propuesta por el profesor Costas Pagianotakis de la *Hellenic Mediterranean University* (Creta, Grecia). La aplicación dispone de tutoriales en la página web de su autor³ para su realización. Se trata de una simulación de bolas de billar en la pantalla del dispositivo móvil. El autor ha hecho pública en su página web académica un tutorial y un vídeo para que cualquier persona con solo unos conocimientos básicos de programación y del entorno pueda desarrollar esta aplicación y personalizarla.

A partir de los choques aleatorios de unas bolas que se mueven aleatoriamente por la pantalla del dispositivo se simulan contagios entre personas y se va dibujando en la pantalla la curva de contagios acumulados. Sorprende la sencillez con que puede simularse la curva de contagios con esta aplicación. Además, permite variar interactivamente el número de personas que interactúan y la velocidad de su movimiento, con lo que puede simularse la efectividad de las medidas de distanciamiento social.

En la clase correspondiente a esta práctica se introdujeron primero los modelos epidemiológicos [9] que han sido desarrollados por el grupo BIOCOSM de nuestra universidad (UPC) y que proveen de datos y

³ App Virus Spreading:
<https://sites.google.com/site/costaspanagiotakis/software/mobile-app-virus-spreading>

asesoramiento diario al *Departament de Salut* de la *Generalitat de Catalunya*. Estos modelos, a su vez, se basan en el modelo epidemiológico de transmisión de Gompert descrito originalmente en 1980 en [8].



Figura 1: App de simulación de contagio
(Fuente: *App Virus Spreading*³)

El grupo BIOCOSM de la UPC se ha hecho muy popular en Cataluña con apariciones diarias en los medios de comunicación y redes sociales; debido a la utilidad de sus gráficas y predicciones a tiempo real de la evolución de la pandemia. Todos los alumnos las conocían sobradamente.

A los estudiantes se les pidió que a partir de la aplicación realizada obtuvieran resultados experimentales de la curva de contagios acumulados en distintas condiciones. Posteriormente debían realizar una comprobación de que estos resultados se ajustaban al modelo de la curva teórica de [8] y así obtener los parámetros ajustados correspondientes del modelo. También se les propuso como ejercicio modificaciones adicionales a la práctica descrita inicialmente:

- Notificación al usuario de que se ha conseguido el nivel de inmunidad de rebaño (70% o nivel pre programado).
- Simulación de medidas adicionales de contención de la pandemia.
- Predicción de otros indicadores epidemiológicos de acuerdo con los modelos.

- Grabación de datos de forma permanente.
- Pantallas adicionales de información de la pandemia Covid19.

5. Resultados obtenidos con la experiencia docente

De los 27 alumnos que se matricularon inicialmente en la asignatura, 26 de ellos (96.2%) siguieron regularmente la asignatura. No hubo abandonos durante el curso, lo cual, y teniendo en cuenta las circunstancias de virtualidad de parte del curso, es muy satisfactorio. Dos alumnos comunicaron al centro durante el curso que debían permanecer confinados en su domicilio un período de tiempo de quince días por ser contacto estrecho de contagios. A estos estudiantes se les hizo un seguimiento individualizado sin que su rendimiento académico se viera finalmente afectado.

5.1. Resultados académicos

La evaluación de la asignatura se compone de tres partes diferenciadas:

- Formación básica: cinco entregables individuales de las prácticas realizadas. (30%)
- Entrega parcial del proyecto (25%)
- Entrega y presentación final del proyecto (45%).

Según la normativa académica vigente de la UPC no se puede suspender un alumno por una sola prueba de evaluación, con lo que no puede haber pruebas o actividades de evaluación eliminatorias.

En cuanto a las notas finales, el porcentaje de aprobados sobre el total de matriculados fue muy satisfactorio, del 96.2%. Las calificaciones se muestran en la Figura 2:

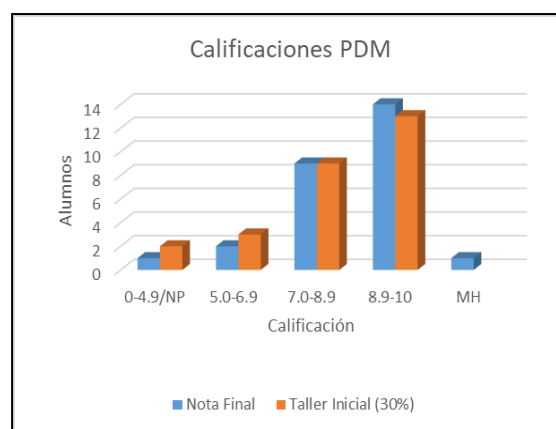


Figura 2: Calificaciones Finales y notas del Taller Inicial de PDM

Destacamos que los resultados de la evaluación de la primera parte de la asignatura (Taller Inicial) no fueron tan óptimos y que fue a partir de la realización del proyecto que se produce una mejora en los estudiantes con calificación mediana o baja. Consideramos que el hecho de tener que hacer obligatoriamente el proyecto (ya que supone un 70% de la nota final) en grupo mediante AC y ABP retuvo y motivó a los alumnos con calificaciones medias o bajas a mejorar su rendimiento. La calificación de cada grupo en el proyecto final, tanto en la entrega parcial como en la final es la misma para cada uno de los integrantes. El seguimiento individual por semana mediante las actas de las reuniones y tutorías a los grupos asegura que no haya posibilidad de que posibles «jetas y mantas» [12] se aprovechen del entorno. En cualquier caso, en esta edición no hubo incidentes remarcables que precisaran de la intervención del profesor.

5.2. Opinión de los estudiantes

Se ha realizado una encuesta a final de curso a los alumnos para obtener su opinión sobre la docencia recibida. Para ello, se ha utilizado la encuesta estandarizada *Students' Evaluation of Education Quality* (SEEQ) por su robustez, fiabilidad y amplio uso en el campo de la investigación educativa [9]. La encuesta se pasó de forma electrónica a los alumnos mediante un formulario accesible desde el aula virtual de la asignatura. Los alumnos fueron informados de que la encuesta es anónima, voluntaria y que los resultados se mostrarían de forma agregada para mejorar la asignatura. Se utilizó una versión traducida al catalán por el *Institut de Ciències de la Educació de la UPC*. La encuesta consta de una serie de ítems a los cuales el alumno responde con una cifra del 1 al 5 (Escala de Likert), significando el 1 «Muy en desacuerdo» con el ítem, el 2 «En desacuerdo», el 3 «Neutro», el 4 «De acuerdo» y el 5 «Muy de acuerdo». Los ítems de la encuesta están divididos en distintos apartados, conceptos que se hacen explícitos al propio encuestado. Estos apartados se detallan en la Figura 3, junto con un ejemplo de pregunta de cada apartado. En este cuestionario, además, se añadieron dos ítems adicionales y un espacio en blanco para comentarios libres.

No se preguntó por la nota esperada que corresponde al último ítem de la Figura 3, puesto que la encuesta se pasó con las calificaciones ya publicadas. Destacamos que la mayoría de ámbitos han sido calificados con un grado de satisfacción elevado (mayor que 4 sobre un máximo de 5). Es interesante también hacer notar que, lejos de considerar este curso fácil por lo elevado de las calificaciones, los estudiantes han puntuado la «carga de trabajo» con un 4.5 sobre 5, es decir, que consideran el curso difícil o que requiere mucho esfuerzo en comparación con otros cursos de la misma titulación.

Ámbito de evaluación	Ejemplo de pregunta
Calidad del aprendizaje	He aprendido cosas que considero valiosas.
Entusiasmo	El profesor consigue que sus presentaciones sean amenas.
Organización	Las explicaciones del profesor eran claras.
Interacción con el grupo	Se invitaba al estudiantado a preguntar y se daban respuestas satisfactorias.
Actitud personal	El profesor se ha mostrado accesible con el estudiantado.
Contenido	El profesor ha presentado diversos puntos de vista cuando era necesario.
Exámenes	Los métodos de evaluación del curso eran equitativos y adecuados.
Trabajos del curso	La bibliografía y el material recomendado de este curso son completos y adecuados.
Carga de trabajo y dificultad	Este curso, comparado con otros, ha sido: Muy fácil (1), Fácil (2), Normal (3), Difícil (4), o Muy Difícil (5).
Visión General	Este curso es mejor que la mayoría de los que he cursado en esta Universidad.
Otras opiniones	La calificación final que esperas obtener del curso es: Menor que 3 (1), Entre 3 y 5 (2), Entre 5 y 7 (3), Entre 7 y 9 (4), Mayor que 9 (5).

Figura 3: Ámbitos de evaluación considerados en la encuesta de satisfacción SEEQ [9]

La encuesta fue respondida por 18 estudiantes (69.2% de los que siguieron la asignatura). Los resultados de esta encuesta se muestran a continuación en el Cuadro 1:

Ámbito evaluado	Valor (1-5)
Calidad del aprendizaje	4.2 +/- 0.4
Entusiasmo	3.8 +/- 0.1
Organización	3.7 +/- 0.3
Interacción con el grupo	4.4 +/- 0.2
Actitud personal	4.6 +/- 0.5
Contenido	4.1 +/- 0.1
Evaluación	4.2 +/- 0.2
Trabajos del curso	4.1 +/- 0.5
Carga de trabajo	4.5 +/- 0.1
Visión general	4.3 +/- 0.3

Cuadro 1: Opinión de los estudiantes sobre la docencia recibida

Los ítems adicionales planteados fueron:

- ¿Consideras que la práctica de simulación de contagio fue de utilidad para aprender?
Valor de respuesta: 4.4 +/- 0.1
- ¿Crees que el proyecto final te ha motivado a seguir la asignatura?
Valor de respuesta: 3.9 +/- 0.3

En los comentarios libres se repitieron dos temas que parecen destacar en sus opiniones:

- Alta carga de trabajo de la asignatura comparado con otras asignaturas.

- Utilidad de la gran cantidad de vídeos y materiales puestos a su disposición para su uso online y asíncrono.

Se añade aquí la transcripción de algunas respuestas que muestran un contagioso entusiasmo final a pesar de la carga de trabajo realizada:

Estuve dudando de matricularme porque apenas sabía nada de programación, pero me ha parecido fácil realizar apps en mi móvil. ¿Existe una continuación de esta asignatura?

Ésta es la asignatura que me ha llevado más horas de trabajo con diferencia, pero es lo que tiene la programación. Los videojuegos y la simulación del Covid están bien.

La sustitución de las clases presenciales por virtuales grabadas en video me ha facilitado que pudiera seguir el curso. Gracias.

6. Emprendeduría e innovación

En la UPC se han definido un conjunto de siete competencias transversales para trabajar en las distintas asignaturas de sus titulaciones [5]. En esta asignatura se trabaja la de “Emprendeduría e innovación” cuyo objetivo general es el de «Conocer y entender la organización de una empresa y las ciencias que definen su actividad; las estrategias industriales y comerciales, la calidad y el beneficio.».

Estas competencias se trabajan en tres niveles de complejidad. Se asignó el nivel más alto a esta asignatura con el objetivo específico de «Utilizar conocimientos y habilidades estratégicas para la creación y gestión de proyectos, aplicar soluciones sistémicas a problemas complejos y diseñar y gestionar la innovación.»

En esta asignatura se ha centrado el trabajo de esta competencia profesional en saber innovar y producir una aplicación de móvil como producto de ingeniería original; y en conocer el desarrollo del producto en el mercado.

Se realizaron dos talleres formativos durante el curso destinados a desarrollar esta competencia:

1. Proceso de desarrollo de una aplicación de móvil. Repositorios públicos.
2. Las empresas de videojuegos y aplicaciones móviles. Cómo se comercializan estos productos en la actualidad.

Se promovió que, al finalizar el curso, los grupos de estudiantes interesados puedan contactar con la

Oficina de Innovación de la UPC⁴ que les asesora en la valorización de la aplicación que hayan desarrollado, y en su posible comercialización.

En la presente edición existe en el presente momento al menos un grupo de estudiantes interesados que están en el proceso de valorizar el trabajo que han realizado en la asignatura con el objetivo de su posible colocación en el mercado.

7. Conclusiones

Se ha presentado una experiencia docente para la enseñanza de la Programación en Dispositivos Móviles en un entorno híbrido. La asignatura se ha realizado en condiciones de pandemia cambiantes. En consecuencia, parte de la docencia se ha realizado virtualmente, lo que suponía una dificultad importante para todos. La materia siguió las metodologías de Aprendizaje basado en proyectos (ABP) y Aprendizaje cooperativo (AC). Además, se contextualizó alguno de los trabajos prácticos de la materia, realizando una aplicación gráfica de móvil de simulación de contagios Covid19.

A pesar del reto que suponía impartir esta materia con unos contenidos totalmente renovados y en condiciones sanitarias difíciles, los resultados han sido altamente satisfactorios. Los estudiantes han salido del curso satisfechos de la experiencia y habiendo en su gran mayoría alcanzado los objetivos de aprendizaje de la materia con muy buenos resultados. Esta experiencia puede ser fácilmente adaptada en otras universidades que impartan materias y titulaciones afines.

Referencias

- [1] Sergio Barrachina y Germán Fabregat. “Introduction to Programming using Mobile Phones and MIT App Inventor”. *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*, 15, 3, pp. 192-201, 2020.
- [2] BIOCOSC. Grupo de investigación de la UPC. “Analysis and prediction of COVID-19 for EU-EFTA-UK and other countries. Methods.” *Working Paper*. Accedido el 5-2-2021 en: <https://BIOCOSCsc.upc.edu/en/shared/methods.pdf>
- [3] Jorge Castro, Javier Farreres, Joaquim Gabarró, Pilar Nivela, Antoni Perez-Poch, Elvira Pino y José M. Rivero. “Ortocoordinación: cómo organizar 700 estudiantes en un nuevo campus (y no morir en el intento).” En *Actas de las XXIV Jornadas de Enseñanza Universitaria de Informática, Jenui 2018*, pp. 247 – 254, Barcelo-

⁴ Oficina de Innovación de la UPC: <http://www.upc.edu/innovacio>

- na, 2018. Accedido el 5-2-2021 en: <https://futur.upc.edu/23293552i>
- [4] Rafaela García, Joan A. Traver y Isabel Candela I. *Aprendizaje cooperativo: fundamentos, características y técnicas*. Editorial CCS. Madrid, 2001.
- [5] Instituto de Ciencias de la Educación de la UPC. “Resumen sobre las competencias genéricas a implantar en los planes de estudios de grado de la UPC”. *Documento de trabajo*. Accedido el 5-2-2021 en: https://www.upc.edu/ice/ca/innovacio-docent/publicacions_ice/arxiu/resum-en-sobre-las-competencias-genericas
- [6] David W. Johnson, Roger T. Johnson y Edithe J. Holubec. *Cooperation in the classroom*. 9ª Edición. Edina, MN. Interaction Book Company, 2013.
- [7] Paul A. Kirschner, John Sweller y Richard E. Clark. “Why minimal guidance during instruction does not work: an analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching.” *Educational Psychologist*, 41, 2, pp. 75-86, 2006.
- [8] Laurence V. Madden. “Quantification of disease progression.” *Protection Ecology*, 2, 159-176, 1980.
- [9] Herbert W. Marsh y Lawrence A. Roche. “Making students' evaluations of teaching effectiveness effective: The critical issues of validity, bias, and utility.” *American Psychologist*, 52, 11, pp. 1187-1197, 1997.
- [10] Dana A. Robertson y Christopher J. Padesky. “Keeping Students Interested: Interest-Based Instruction as a Tool to Engage.” *Reading Teacher*, 73, 5, 575-586, 2020.
- [11] Miguel Valero. “Las dificultades que tienes cuando haces PBL”. *Texto docente*. Accedido el 5-2-2021 en: https://people.ac.upc.edu/miguel/materiales/docencia/articulos/dificultades_PBL.pdf
- [12] Miguel Valero. “Cómo enfrentarse a los jetas y a los mantas”. *Texto docente*. Accedido el 5-2-2021 en: http://bioinfo.uib.es/~joemiro/TecBasAula/jtas_mantas.pdf
- [13] Ferran Virgós y Antoni Perez-Poch. “Un modelo para aplicación sistemática de aprendizaje cooperativo”. *Actas de las VIII Jornadas de Enseñanza Universitaria de Informática, Jenui 2002*, pp. 343 – 349, Cáceres, 2002. Accedido el 5-2-2021 en: http://bioinfo.uib.es/~joemiro/aenui/procJenui/Jen2002/Cac123_130.pdf
- [14] David W. Wolber, “App Inventor and real-world motivation,” in *Proc. 42nd ACM Tech. Symp. Comput. Sci. Edu. (SIGCSE)*. New York, USA: ACM, pp. 601–606, 2011.

Promoción del Pensamiento Computacional en estudiantes pre-universitarios: ¿cómo se emocionan?

Rafael Herrero Álvarez¹, Coromoto León¹, Gara Miranda¹, Eduardo Segredo¹, Óscar Socas¹,
María Cuellar-Moreno², Daniel Caballero-Julia³, Laura García⁴, Yolanda Díaz⁴

Departamento de Ingeniería Informática y de Sistemas, Universidad de La Laguna;¹

Departamento de Didácticas Específicas, Universidad de La Laguna;²

Unité de Recherche Pluridisciplinaire Sport Santé Société, Université de Lille;³

Cienci@ULL, Fundación General de la Universidad de La Laguna⁴

rafael.herrero.13@ull.edu.es, cleon@ull.edu.es, gmiranda@ull.edu.es,
esegredo@ull.edu.es, alu0100768152@ull.edu.es,
mcuellar@ull.edu.es, daniel.caballero-julia@univ-lille.fr,
lgarciad@fg.ull.es, ydiaz@fg.ull.es

Resumen

En este trabajo se presenta un estudio de las emociones que se producen en estudiantes pre-universitarios al realizar actividades de Pensamiento Computacional. Se comparan dos estrategias en las que se intercalan las metodologías guiadas y por descubrimiento. Se concluye que se producen principalmente emociones positivas y ambiguas, mientras que las negativas tienen una intensidad relativamente baja. También se observa, en relación con las modalidades, que en secundaria las chicas parecen mostrar una menor intensidad en las emociones positivas y ambiguas y un ligero aumento en las negativas. En cuanto a las dos estrategias utilizadas no se observan diferencias significativas en ningún momento sobre la media.

Abstract

This paper presents a study of the emotions that are produced in pre-university students when performing Computational Thinking activities. Two strategies are compared in which the guided and discovery methodologies are interspersed. It is concluded that positive and ambiguous emotions are mainly produced, while negative ones have relatively low intensity. It is also observed in relation to the modalities, that in secondary, girls seem to show less intensity in positive and ambiguous emotions and a slight increase in negative ones. Regarding the two strategies used, no significant differences were observed at any time above the mean.

Palabras clave

Pensamiento Computacional, emociones, educación secundaria, educación primaria.

1. Introducción

En todos los niveles del ámbito educativo el uso de las Tecnologías de la Información y las Comunicaciones (TIC) es algo habitual. La legislación educativa española, tanto en la etapa de educación primaria como en secundaria, contempla la alfabetización digital. Así pues, cualquier persona puede adquirir nociones básicas sobre Informática, pero no, sobre cómo funciona una máquina digital programable, es decir, qué puede ser automatizado y qué no [16]. Además, la Informática no siempre despierta el interés de los más jóvenes por dos razones: la primera por desconocimiento, y la segunda por la creencia de ser algo complicado y que no está a su alcance [10, 6]. Además, se ha de añadir que este campo suele resultar más interesante para los chicos que para las chicas, lo que ocasiona que el número de matrículas por parte de las mujeres en carreras relacionadas con la Ingeniería Informática sean mucho más bajas que la de los hombres [17]. Por ello, en el año 2017 nació el proyecto Piens@ Computacion@ULLmente con el objetivo de promocionar las Ciencias de la Computación entre los estudiantes de educación primaria y secundaria, a través de actividades relacionadas con el Pensamiento Computacional, abordándolas con una perspectiva de género de manera que resulten más interesantes a las chicas [9].

El Pensamiento Computacional es una técnica para resolver problemas, diseñar sistemas y comprender la conducta humana haciendo uso de conceptos fundamentales de las Ciencias de la Computación [20]. También podría describirse como aquellos procesos de pensamiento implicados en la formulación de problemas y la representación de sus soluciones, donde dichas soluciones pueden ser ejecutadas por un agente de procesamiento de información, el cual puede ser un

humano, un ordenador o la combinación de estos. Algunos autores también han incluido a esta definición la persistencia para trabajar con problemas complicados o la habilidad para manejar la ambigüedad [2], o incluso que va más allá de las computadoras, ya que se ven involucradas tres áreas, como conceptos de programación (secuencias, bucles, eventos, etc.), algunas prácticas que se desarrollan según se va programando (mejorar a la hora de solucionar un problema, reutilizar, mezclar distintos proyectos, etc.) y perspectivas del mundo que les rodea (expresarse, conectar con otros, cuestionar ideas,...) [4].

Las emociones afectan la forma en la que adquirimos conocimientos, es decir, cómo aprendemos [19, 5]. Concretamente, se ha demostrado en el área de las Ciencias de la Computación mejores resultados a la hora de aprender a programar utilizando una plataforma con sistemas que reconocen las emociones y adaptan los contenidos a ellas, que utilizando la misma plataforma pero sin el sistema de reconocimiento activado [21]. Además, algunos estudios apuntan a la pérdida de auto-eficacia de las personas que utilizan computadoras debido a la ansiedad [1], o de como es posible que reduciendo la ansiedad y la ira a través del uso de computadoras se puede mejorar el conocimiento que se tiene sobre estas [12]. El objetivo de este trabajo es el análisis de las emociones producidas en jóvenes que realizan actividades de Pensamiento Computacional, categorizadas en negativas, neutras y positivas. De esta manera se puede comprobar como afectan a la percepción que tienen sobre las Ciencias de la Computación, teniendo en cuenta la edad y el género.

El resto del trabajo se organiza de la siguiente manera. El apartado 3 aborda la metodología seguida en este estudio, tanto con la descripción de las sesiones de actividades realizadas como de los instrumentos de medición utilizados. El apartado 4 presenta los resultados del estudio y la discusión. El apartado 5 incluye las conclusiones del trabajo y las líneas futuras que se pretende seguir.

2. Objetivos

El objetivo principal de este trabajo es realizar un análisis del estado emocional de estudiantes pre-universitarios cuando realizan actividades de Pensamiento Computacional, identificando qué emociones se producen y su intensidad, así como estudiar las posibles diferencias entre las emociones sentidas en función del modelo de sesión, nivel educativo y género. Los objetivos concretos son los siguientes:

H1: Las emociones positivas sentidas serán menores en las chicas que en los chicos, especialmente en secundaria.

H2: Las emociones negativas serán mayores en secundaria que en primaria.

H3: No hay diferencias significativas en cuanto a emociones siguiendo un modelo de sesión u otro.

3. Metodología

Este proyecto se ha llevado a cabo con estudiantes de diferentes centros educativos de la isla de Tenerife, de 4º de primaria (8-9 años) y 2º de la ESO (12-13 años). El Cuadro 1 recoge el total de estudiantes que participaron en el proyecto, dividido por curso y género, una vez se han eliminado los datos erróneos o incompletos.

Los estudiantes han seguido un curso de 10 horas presenciales, 5 sesiones de 2 horas, y otras 10 horas de trabajo autónomo, en las que se ha trabajado siguiendo dos modelos, uno comienza con 2 sesiones de metodología guiada y termina con 3 de descubrimiento, modelo guiado-descubrimiento (modelo GD), mientras que en el otro se llevan a cabo al revés, modelo descubrimiento-guiado (modelo DG). La intervención se llevó a cabo durante 5 semanas en cada centro, por un equipo formado por el mismo instructor y la misma persona de apoyo.

Para registrar las emociones se han usado dos instrumentos. Al final de cada una de las sesiones, los participantes han completado el Cuestionario de canales de desarrollo (*Developmental Channels Questionnaire* - DCQ) [14], además de entrevistas individuales para obtener información más detallada relacionada con los temas que surgieron del análisis del cuestionario.

La cantidad y la calidad de los datos se ha visto seriamente afectada por el contexto de crisis sanitaria del Covid-19. Un mayor número de cuestionarios repartidos en más sesiones y durante un periodo de tiempo mayor permitiría mejorar la solidez de los datos cuantitativos recogidos.

Primaria		Secundaria	
74 estudiantes		28 estudiantes	
39 chicas	35 chicos	10 chicas	18 chicos

Cuadro 1: Descripción cuantitativa de la muestra

3.1. Actividades

Para llevar a cabo la formación del alumnado se han desarrollado una combinación de actividades y herramientas, tanto enchufadas como desenchufadas, es decir, en las que se hace uso de un ordenador o dispositivo móvil, papel y lápiz o cualquier dispositivo electrónico.

Primaria			
Guiadas		Descubrimiento	
Sesión 1	Sesión 2	Sesiones 3, 4 y 5	
Ratón Code&Go. Programar un robot que recorre un laberinto ¹ .	Curso en Code.org. Curso 2 ² .	Ejercicio en Scratch [15]. Cesta de frutas. El alumnado programa una cesta en la que meter frutas sin tener que sobrepasar el límite de calorías especificado. Usando la placa Makey Makey ³ , recortan frutas en cartulinas y forran con aluminio.	
Secundaria			
Guiadas		Descubrimiento	
Sesión 1	Sesión 2	Sesión 3	Sesiones 4 y 5
Curso en Code.org. Curso de 20 horas ⁴ .	Ejercicio en Scratch. Reto de las Matrioskas. Ordenar 5 muñecas según el tamaño de menor a mayor. En la sesión 3 se continúa el trabajo por descubrimiento.		Robot mBot ⁵ . Se programa un robot con múltiples sensores para que sea capaz de recorrer un circuito, siendo capaz de reconducirse.

Cuadro 2: Descripción de las actividades del modelo GD

Las actividades se han dividido en dos modalidades según la metodología de aprendizaje, siendo una de carácter guiado, en la que se presentan los conceptos y principios básicos del Pensamiento Computacional partiendo de un problema y analizando el algoritmo a diseñar para su resolución, y otra de descubrimiento, donde al estudiante se le enseñan herramientas que sirven para poner en práctica algunos ejemplos relacionados con el Pensamiento Computacional. La descripción de estas actividades se encuentra en el Cuadro 2 tanto para primaria como para secundaria. Las actividades para el modelo DG son las mismas, salvo que cambia la distribución, ya que se empieza por las de descubrimiento y se termina por las guiadas.

3.2. Instrumentos

Los instrumentos utilizados para medir las emociones han sido: el cuestionario *DCQ* y las entrevistas individuales.

Cuestionario de canales de desarrollo (*Developmental Channels Questionnaire*) - DCQ. Las diferentes formas de enseñanza condicionan la relación entre elementos pedagógicos al crear condiciones para experiencias diversas [14], convirtiéndose en una herramienta a través de la cual el profesorado puede expresar su creatividad e individualidad [8]. La elección de la forma de enseñanza es una decisión importante para los docentes porque afecta su relación con los diversos elementos del acto didáctico [18].

El DCQ incluye escalas representadas con adjetivos significados opuestos (por ejemplo, mínimo-máximo,

difícil-fácil, poderoso-impotente, malo-bueno, útil-inútil y agradable-no agradable) en pares de adjetivos que son medidas excelentes de los pensamientos de un individuo. Concretamente se ha preguntado por la felicidad, compasión, sorpresa, alegría, tristeza, miedo, humor, ansiedad, amor, ira, rechazo, vergüenza y esperanza, utilizando una escala de Likert del 0 al 10.

Los estudiantes han realizado este cuestionario en tres ocasiones, al finalizar la primera sesión, al finalizar la primera sesión en la que ocurre el cambio de metodología (sesión 3 en el caso del modelo GD y sesión 4 en el modelo DG), y en la última.

Entrevistas. Una vez que los estudiantes habían completado el DCQ se seleccionaron al azar 4 participantes, 2 niñas y 2 niños, para ser entrevistados inmediatamente después de cada sesión con la presencia de solo un entrevistador. Este criterio se ajusta a lo mencionado por Gibbs [7] en el tratamiento de la investigación cualitativa donde no es necesaria una gran muestra, sino que ese análisis permita generar un conocimiento y que la selección de individuos de diferente género puede representar comportamientos humanos de diferente índole social. Esto resultó en un total de 26 entrevistas. Las entrevistas se han realizado tanto en primaria con un total 14 estudiantes, y en secundaria con 12 estudiantes. Los entrevistados primero dijeron su edad, género, curso y número de lista. A continuación se les pidió que respondieran a las siguientes preguntas:

- ¿Qué estudios tiene tu padre? ¿Y tu madre?
- ¿En qué trabaja tu padre? ¿Y tu madre?
- ¿Qué has pensado hoy cuando nos has visto llegar?
- ¿Qué hemos hecho hoy?
- ¿Cómo has participado? ¿Has ayudado a algún compañero?
- ¿Qué es lo que más te ha gustado de la sesión?
- ¿Y lo que menos?

¹<https://www.learningresources.com/code-gor-robot-mouse-activity-set>

²<https://studio.code.org/s/course2>

³<https://makeymakey.com/>

⁴<https://studio.code.org/s/20-hour>

⁵<https://www.makeblock.com/mbot/>

Cada entrevista duró entre 10 y 15 minutos. Las entrevistas fueron grabadas en audio y posteriormente transcritas a mano para su análisis. Cada pregunta se elaboró en un lenguaje sencillo y simple para responder a los objetivos y metas de la investigación de una manera interesante y clara. Además, se han considerado dos aspectos esenciales. Por un lado, el proceso comunicativo no se alteró para evitar influir en las respuestas, y por otro, se creó un ambiente de confianza para permitir a los estudiantes expresar sus opiniones libremente y hacer conclusiones reflexivas sobre el tema.

3.3. Análisis de datos

A continuación se indica el procedimiento de análisis según la herramienta que ha sido empleada.

DCQ. El análisis de datos se ha realizado con el programa estadístico SPSS en su versión 20.0 para Windows. Tras recoger los datos del cuestionario, se ha realizado un cálculo de las variables teóricas, las cuáles clasificamos en positivas, negativas y ambiguas, de acuerdo a lo expuesto por Lazarus [13] y Bisquerra [3], a partir de la media de las puntuaciones de cada grupo de emociones. Las pruebas de normalidad de Kolmogorov - Smirnov muestran que las distribuciones de las variables teóricas no siguen una distribución normal, lo que se traduce en una menor fiabilidad de la media como medida de tendencia central. Es posible, por tanto, que algunas de las tendencias observadas como no significativas sean, sin embargo, suficientes como considerarlas importantes. En segundo lugar, se ha realizado un análisis CHAID [11] permitiendo la representación de los datos en árboles de decisión para las variables Sexo (hombre o mujer), Nivel (Primaria o Secundaria) y Tipo de sesión (modelo GD o modelo DG).

Los datos apuntan a una interrelación entre las variables distinta de la planteada por Lazarus [13] y Bisquerra [3]. Pruebas preliminares mediante análisis factorial exploratorio apuntan a un resultado de dos factores (positivas + ambiguas y negativas) en lugar de tres. En trabajos futuros los resultados podrían ganar en solidez mediante la consideración de sólo estos dos factores.

Entrevistas. La fase de análisis de contenido ha consistido en la transcripción de las entrevistas, lectura de las transcripciones y codificación de las palabras clave a partir de las respuestas de cada sujeto. Se ha realizado una transcripción y se generó una lista de códigos con el fin de extraer los temas que componen el discurso de las entrevistas. Cada código fue asignado a una fracción de texto (de extensión variable, pero nunca menos de una frase). No obstante, un mismo fragmento puede hacer referencia a múltiples temas/códigos, por lo que se pueden encontrar múltiples concurrencias. Esto hizo posible agrupar las palabras

en categorías con el mismo significado semántico, dando como resultado cuatro grandes dimensiones que establecieron su frecuencia. De este modo, se agruparon las palabras en un total de 13 códigos que fueron agrupados según los canales de desarrollo físico, cognitivo, social y emocional. Los códigos considerados fueron categorizados igual que las emociones en el cuestionario *DCQ*: positivas (alegría, amor, felicidad y humor), ambiguas (compasión, sorpresa y esperanza) y negativas (ira, miedo, rechazo, tristeza, ansiedad y vergüenza).

Las entrevistas presentadas muestran elementos interesantes y coherentes con los objetivos planteados, sin embargo, sería necesario profundizar en el discurso para una mayor riqueza en los datos cualitativos.

4. Resultados

4.1. DCQ - Emociones, género y nivel educativo

En este apartado se indican las emociones positivas, neutras y negativas obtenidas en función del nivel educativo del alumnado.

La Figura 1 muestra las emociones positivas, neutras y negativas respectivamente respecto al género y nivel educativo. Como se aprecia, vemos que no parecen encontrarse diferencias entre chicos y chicas a nivel de primaria. Sin embargo, las chicas sienten en menor medida emociones positivas al llegar a la secundaria, por lo que se acepta la hipótesis *H1*.

Sobre las emociones ambiguas en función del género y del nivel educativo, se aprecian diferencias entre chicos y chicas en ambos niveles educativos. Sin embargo, las chicas y los chicos sienten en menor medida emociones ambiguas al llegar a secundaria. Ambas tendencias (chico y chica) parecen darse en igual medida.

A pesar de que en ambos niveles las emociones negativas presentan un nivel bajo (cercano a 0) vemos que aparecen diferencias importantes entre los niveles. De igual modo, estas diferencias se encuentran mucho más marcada en chicas que en chicos, lo que reafirma la hipótesis *H1*. En general, las emociones negativas están más presente en secundaria que en primaria, por lo que se acepta la hipótesis *H2*.

4.2. DCQ - Emociones, género y modelo de sesiones

En este apartado se indican las emociones positivas, neutras y negativas obtenidas en función del modelo de sesión seguido.

La Figura 1 muestra las emociones positivas, neutras y negativas respectivamente respecto al género y

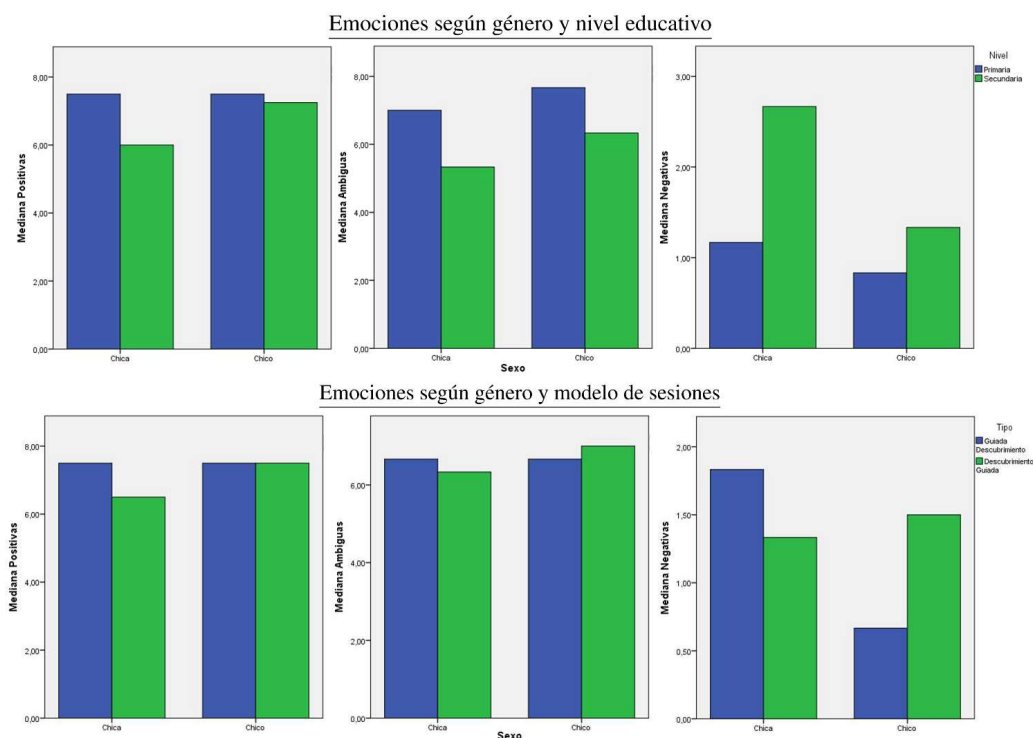


Figura 1: Emociones según género, nivel educativo y modelo de sesiones.

modelo de sesión seguido. Se aprecia que aparecen diferencias entre chicos y chicas a la hora de plantear las sesiones de descubrimiento. Mientras que no parece haber diferencias entre chicos y chicas para las sesiones guiadas.

Por otra parte, las sesiones de descubrimiento parecen causar un menor número de emociones positivas entre las chicas. Se observan grandes diferencias entre chicos y chicas respecto a las emociones ambiguas. Sin embargo, encontramos un ligero cambio en los chicos a la hora de plantear las sesiones de descubrimiento.

En cuanto a las emociones negativas en función del género y el modelo de sesión, no se observan diferencias aparentemente importantes a la hora de plantear un modelo u otro. Además, se observan diferencias entre chicos y chicas de manera inversa. Así, mientras las chicas parecen desarrollar una mayor intensidad de emociones negativas en las sesiones de tipo guiado, los chicos desarrollan una mayor intensidad de emociones negativas en las de descubrimiento. A pesar de las diferencias observadas, estas no son significativas, por lo que se acepta la hipótesis *H3*.

4.3. DCQ - Árboles de género, nivel educativo y modelo de sesión

En este apartado se muestran los árboles de clasificación de las emociones positivas, neutras y negativas

obtenidas en función del nivel educativo y género, como se puede apreciar en las Figuras 2, 3 y 4.

En los tres tipos de emociones se observa un comportamiento significativamente diferente a partir del nivel de secundaria, especialmente en las chicas. Así, mientras vemos que entre chicos y chicas de primaria no hay diferencias en cuanto a sus emociones, en secundaria las chicas parecen sentir con menor intensidad las emociones positivas y ambiguas, lo que reafirma la hipótesis *H1*, al igual que las emociones negativas respecto a la etapa educativa anterior, y a los chicos con menor intensidad, lo que confirma la hipótesis *H2*. En cualquier caso, vemos como el modelo de sesión que se emplee no parece mostrar diferencias significativas en cuanto al impacto emocional, por lo que se acepta la hipótesis *H3*.

4.4. Entrevistas - Emociones, género y nivel educativo

Aunque parecen no encontrarse diferencias entre chicos y chicas a nivel de primaria se aprecia una disminución de las emociones positivas en las chicas al llegar a la secundaria (Figura 1). Una frase representativa es la siguiente: “Pues hecho un trabajo bueno sí un trabajo que me gustó mucho y también después hice algo de un disfraz de en el ordenador”, (Chica, primaria).

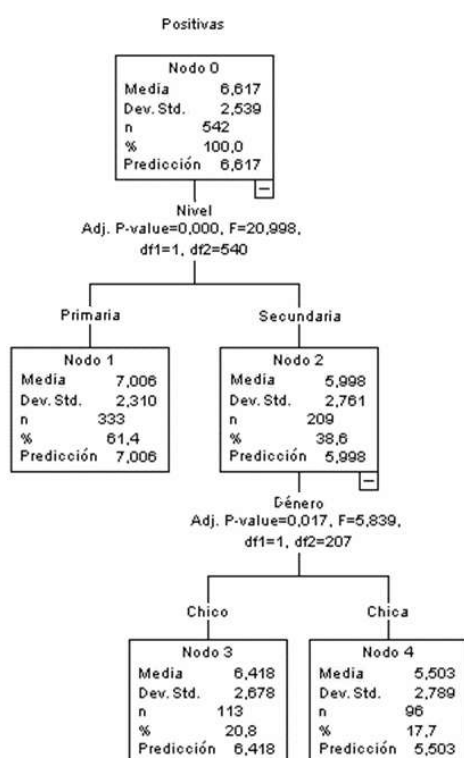


Figura 2: Árbol de emociones positivas

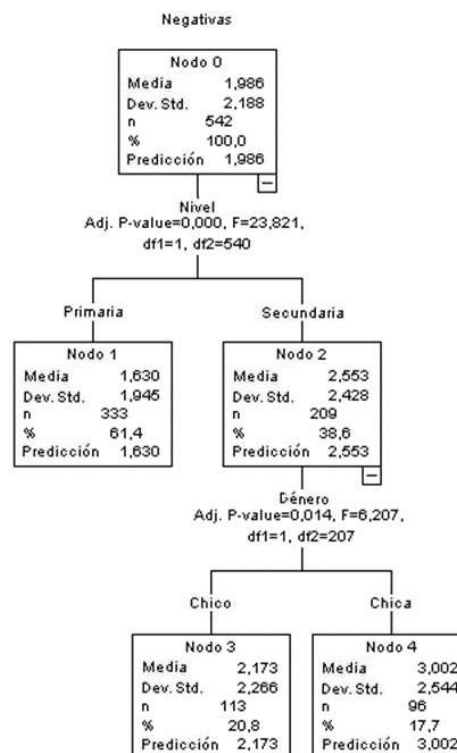


Figura 4: Árbol de emociones negativas

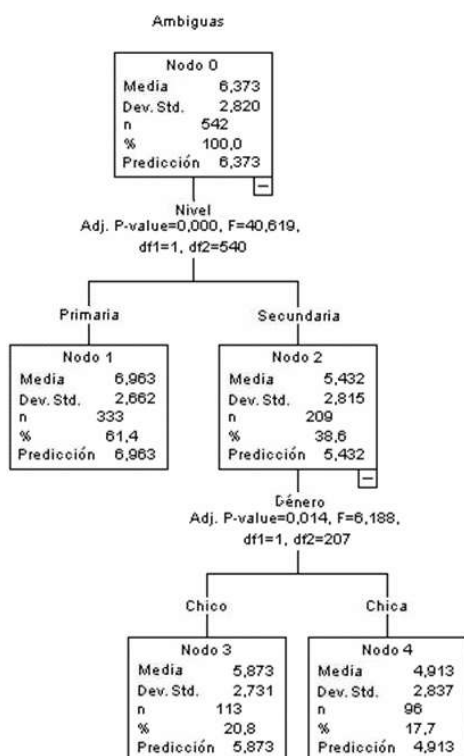


Figura 3: Árbol de emociones neutras

Respecto a las emociones ambiguas en función del género y del nivel educativo (Figura 1), aunque se aprecian diferencias entre chicos y chicas en ambos niveles educativos ambos géneros manifiestan una menor medida emociones ambiguas al llegar a secundaria. Por ejemplo: “Me ha parecido muy bien, sino que a veces le dábamos a otro botón, se nos caía el ratón y a veces había que poner el tablero mejor porque el chico que lo estaba poniendo pues lo ponía un poco separado, pues nosotras dos lo poníamos mejor”, (Chica, primaria).

En las emociones negativas vemos que aparecen diferencias importantes entre los niveles, siendo estas más marcadas en chicas que en chicos (Figura 1). Tómese como muestra: “Lo que menos me ha gustado hoy ha sido no poder hacer otra cosa”, (Chico, secundaria).

4.5. Entrevistas - Emociones, género y modelo de sesión

En la Figura 1 sobre emociones positivas en función del género y del modelo de sesión, se aprecia que aparecen diferencias entre chicos y chicas a la hora de plantear las sesiones de descubrimiento. Mientras que no parece haber diferencias entre chicos y chicas para las sesiones guiadas. Por otra parte, las sesiones de

descubrimiento parecen causar un menor número de emociones positivas entre las chicas:

- “Lo que más me ha gustado hoy es sobre todo ver el resultado del trabajo y lo que menos me ha gustado yo creo que ha sido es que en una ocasión tardaron en decir el siguiente paso que íbamos a decir por los otros compañeros”, (Chico, secundaria).
- “Como no estoy acostumbrado a hacer estas cosas me estresé porque no sabía qué hacer y esa fue la parte que no me gustó”, (Chico, secundaria).
- “El que podamos hacer el scratch que eso no pasa en la mayoría de clases, me ha gustado mucho que podamos hacer esto. Que nos podamos divertir de otra manera. Lo menos, lo complicado que son los problemas de resolver y colocar los bloques”, (Chico, secundaria).

También se observan diferencias con un ligero cambio en los chicos a la hora de plantear las sesiones de descubrimiento (Figura 1): “Me he visto bien porque entiendo los ordenadores, pero hay veces que me pongo nervioso si no los controlo muy bien. Estuve haciendo mis actividades, ayudé a mis compañeros y mis compañeros también me ayudaron a mí”, (Chico, secundaria).

Por último, se observan se observan diferencias aparentemente importantes a la hora de plantear el modelo de sesión en tanto mientras las chicas parecen desarrollar una mayor intensidad de emociones negativas en las sesiones de tipo guiado, los chicos desarrollan una mayor intensidad de emociones negativas en las de descubrimiento (Figura 1): “Como no estoy acostumbrado a hacer estas cosas me estresé porque no sabía qué hacer y esa fue la parte que no me gustó”, (Chico, secundaria).

5. Conclusiones

Las principales conclusiones de este estudio han sido establecidas tomando como base los objetivos y las hipótesis planteadas para el desarrollo del mismo. Respecto al objetivo de la identificación de las emociones que se producen, así como su intensidad se concluye que en este tipo de sesiones se producen principalmente emociones positivas y ambiguas con valores de intensidad de 6,62 y 6,37 de media respectivamente. Las emociones negativas tienen una intensidad relativamente baja con 1,99 de media sobre 10,0, pero siendo más marcadas en secundaria, por lo que se ratifica la hipótesis H2.

Además, respecto al estudio de las posibles diferencias entre las emociones sentidas en función del modelo de sesión (guiado-descubrimiento o descubrimiento-guiado) se concluye que en el nivel educativo de primaria, se deduce que, con los datos recogidos, se pueden encontrar emociones positivas, negativas y ambiguas similares entre los chicos y las chicas sin hallar dife-

rencias significativas entre ambos géneros en ninguna de ellas.

Sin embargo, en cuanto al nivel educativo (primaria o secundaria) se tiene que en secundaria se producen diferencias entre los chicos y las chicas en todas ellas. Se observa que las chicas presentan cambios más importantes en este tipo de sesiones mientras que los chicos apenas evolucionan emocionalmente en este aspecto. Así, las chicas parecen mostrar una menor intensidad en las emociones positivas y ambiguas y un ligero aumento en las negativas, por lo que se valida la hipótesis H1.

En cuanto a los dos modelos de sesiones utilizados (guiado-descubrimiento y descubrimiento-guiado) no se observan diferencias significativas en ningún momento sobre la media, por lo que se confirma la hipótesis H3.

Agradecimientos

Este trabajo ha sido financiado por “Piens@ Computacion@LLmente (A17120413). Programa educativo para el fomento del pensamiento computacional a través de la realización de actividades que permitan su desarrollo y su inclusión en el currículo”. Cabildo Insular de Tenerife. Fundación General de la Universidad de La Laguna.

Referencias

- [1] Nurain Achim y Arraqib Al Kassim: *Computer Usage: The Impact of Computer Anxiety and Computer Self-efficacy*. Procedia, social and behavioral sciences, 172:701–708, Jan 2015.
- [2] Valerie Barr y Chris Stephenson: *Bringing computational thinking to K-12*. ACM Inroads, 2(1):48–54, Feb 25, 2011.
- [3] Rafael Bisquerra Alzina: *Educación emocional y competencias básicas para la vida*. Jan 1, 2003.
- [4] K. Brennan y M. Resnick: *New frameworks for studying and assessing the development of computational thinking*. En *Annual American Educational Research Association meeting*, Vancouver, BC, Canada, 2012.
- [5] Aránzazu Elizondo Moreno, José Víctor Rodríguez Rodríguez y Ignacio Rodríguez Rodríguez: *La importancia de la emoción en el aprendizaje: Propuestas para mejorar la motivación de los estudiantes*. Cuaderno de Pedagogía Universitaria, páginas 3–11, 2018.
- [6] Michail N. Giannakos, Letizia Jaccheri y Roberta Proto: *Teaching Computer Science to Young Children through Creativity: Lessons Learned from the Case of Norway*. En *In Computer Scien-*

- ce Education Research Conference*, páginas 103–111, 2013.
- [7] Graham Gibbs: *El análisis de datos cualitativos en investigación cualitativa*. Ediciones Morata, S. L, Madrid, 2012, ISBN 8471126850.
 - [8] Michael Goldberger, Sara Ashworth y Mark Byra: *Spectrum of Teaching Styles Retrospective 2012*. Quest (National Association for Kinesiology in Higher Education), 64(4):268–282, Oct 2012.
 - [9] Rafael Herrero, Coromoto León, Gara Miranda y Eduardo Segredo: *El proyecto Piens@ Computacion@LLmente*. En *CINAIC 2019*, páginas 573–578. Universidad de Zaragoza, 2019.
 - [10] Peter Hubwieser, Michal Armoni, Torsten Brinda, Valentina Dagiene, Ira Diethelm, Michail Giannakos, Maria Knobelsdorf, Johannes Magenheimer, Roland Mittermeir y Sigrid Schubert: *Computer science/informatics in secondary education*. Proceedings of the 16th annual conference reports on innovation and technology in computer science education - working group reports, páginas 19–38, Jun 27, 2011.
 - [11] G. V. Kass: *An Exploratory Technique for Investigating Large Quantities of Categorical Data*. Journal of the Royal Statistical Society. Series C (Applied Statistics), 29(2):119–127, Jan 1, 1980.
 - [12] Robin H. Kay: *Exploring the relationship between emotions and the acquisition of computer knowledge*. Computers and education, 50(4):1269–1283, 2008.
 - [13] Richard S. Lazarus: *Emotion and Adaptation*. Oxford University Press, Cary, 1991, ISBN 9780195069945.
 - [14] M. Mosston y S. Ashworth: *Teaching Physical Education*. Benjamin Cummings, London, fifth ed. edición, 2002.
 - [15] Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman y Yasmin Kafai: *Scratch: Programming for All*. Commun. ACM, 52(11):60–67, Noviembre 2009, ISSN 0001-0782.
 - [16] Miguel Riesco, Marián Fondón, Darío Álvarez Gutiérrez, Benjamin Lopez, Agustín Cernuda y Aquilino Juan: *La Informática como materia fundamental en un sistema educativo del siglo XXI*. En *XX JENUI*, páginas 27–32, 2014.
 - [17] Rebecca Strachan, Aruquia Peixoto, Ito Emembolu y M. Teresa Restivo: *Women in engineering: Addressing the gender gap, exploring trust and our unconscious bias*. 2018 IEEE Global Engineering Education Conference (EDUCON), páginas 2088–2093, Apr 2018.
 - [18] Simeon Tsolakidis y Garifalos Anagnostou: *The impact of physical education teaching styles on the construction of pupils' literate subjectivities*. En *Proceedings of the 31st Annual Meeting of the Department of Linguistics of the Faculty of Philosophy*, páginas 145–154, Greece: Aristotle University of Thessaloniki, 2011. Aristotle University of Thessaloniki.
 - [19] Bernard Weiner: *Attributional Theory of Motivation and Emotion*. Psychological Review, 1984.
 - [20] Jeannette Wing: *Computational Thinking*. Communications of the ACM, 49(3):33–35, Mar 1, 2006.
 - [21] Ramón Zatarain Cabada, María Lucía Barrón Estrada, José Mario Ríos Félix y Giner Alor Hernández: *A virtual environment for learning computer coding using gamification and emotion recognition*. Interactive learning environments, 28(8):1048–1063, Dec 15, 2018.

Proyectos de aprendizaje profundo usando datos regionales

Jónathan Heras

Departamento de Matemáticas y Computación
Universidad de La Rioja
26006 Logroño
jonathan.heras@unirioja.es

Resumen

Debido al impacto de las técnicas de aprendizaje profundo, tanto en entornos industriales como académicos, hay una gran demanda de graduados con habilidades en este campo de la inteligencia artificial. Es por esto que las universidades están comenzando a ofrecer asignaturas que incluyen temas relacionados con el aprendizaje profundo. En estas asignaturas, las prácticas son fundamentales; sin embargo, la mayoría de dichas prácticas tienen dos inconvenientes. El primero es el uso de, o bien, “datos de juguete” que sirven para enseñar conceptos pero cuyas soluciones no generalizan a problemas reales; o bien, datos que requieren un conocimiento experto para comprender correctamente su contexto. En segundo lugar, la mayoría de prácticas de aprendizaje profundo se centran en la tarea de entrenar un modelo, y no tienen en cuenta otras tareas, como son la limpieza de los datos o el despliegue de los modelos. En este trabajo presentamos una experiencia en una asignatura de inteligencia artificial donde hemos abordado los problemas anteriores usando datos del gobierno de la comunidad autónoma donde se encuentra nuestra universidad. En concreto, los estudiantes han llevado a cabo diversos proyectos de visión por computador y procesamiento de lenguaje natural usando técnicas de aprendizaje profundo; por ejemplo, han creado un clasificador de noticias o una aplicación para colorear imágenes antiguas. Compartimos aquí el flujo de trabajo utilizado para organizar la experiencia, las lecciones aprendidas y los retos que pueden encontrarse al intentar llevar a cabo iniciativas similares.

Abstract

Due to the impact of Deep Learning both in industry and academia, there is a growing demand of graduates with skills in this field, and Universities are starting to offer courses that include Deep Learning subjects. Hands-on assignments that teach students how to tackle Deep Learning tasks are an instrumental part of those courses. However, most Deep Learning assign-

ments have two main drawbacks. First, they use either toy datasets that are useful to teach concepts but whose solutions do not generalise to real problems, or employ datasets that require specialised knowledge to fully understand the problem. Secondly, most Deep Learning assignments are focused on training a model, and do not take into account other stages of the Deep Learning pipeline, such as data cleaning or model deployment. In this work, we present an experience in an Artificial Intelligence course where we have tackled the aforementioned drawbacks by using datasets from the regional council where our University is located. Namely, the students of the course have developed several computer vision and natural language processing projects; for instance, a news classifier or an application to colourise historical images. We share the workflow followed to organise this experience, several lessons that we have learned, and challenges that can be faced by other instructors that try to conduct a similar initiative.

Palabras clave

Datos regionales, aprendizaje profundo, prácticas, experiencia docente

1. Introducción

Las técnicas de aprendizaje profundo, en inglés *deep learning*, se han convertido en el estándar para abordar problemas en distintos contextos como son la visión por computador [19], el procesamiento de lenguaje natural [12], o la bioinformática [24]. Debido a su éxito, hay una creciente demanda de expertos en este campo del aprendizaje automático, y gran parte de las universidades están incluyendo temas de aprendizaje profundo en asignaturas de grado y máster.

Para el diseño de asignaturas sobre aprendizaje profundo es conveniente mirar el diseño que han seguido asignaturas de campos estrechamente relacionados como es la ciencia de datos. En dicho campo existen diversos estudios que enfatizan la importancia de la enseñanza práctica y basada en aplicaciones [7, 11],

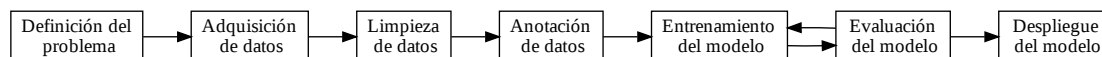


Figura 1: Flujo de trabajo de un proyecto de aprendizaje profundo supervisado.

y esto puede extrapolarse a las asignaturas de aprendizaje profundo. Esto se consigue, en la mayoría de asignaturas, mediante prácticas o proyectos donde los estudiantes tienen que entrenar modelos de aprendizaje profundo usando datos obtenidos de plataformas como *Kaggle*, *Amazon Datasets* o *Google Dataset Search*. Sin embargo la construcción de un modelo es solo uno de los pasos necesarios para abordar un proyecto de aprendizaje profundo (en la figura 1 se resume el flujo de trabajo de un proyecto de aprendizaje profundo), y el resto de pasos no son tenidos en cuenta en la mayoría de prácticas y proyectos. Por ejemplo, las tareas de adquisición, limpieza y etiquetado de datos no son llevadas a cabo por los estudiantes debido a que los datasets disponibles en las diversas plataformas se encuentran ya preprocesados y listos para usar. Otro ejemplo de tarea que no se lleva a cabo es el despliegue de los modelos ya que estos solo son vistos y evaluados por el profesorado, y por lo tanto no se usan fuera del contexto de la asignatura. Un problema adicional es que la mayor parte de datasets públicos son, o bien, “problemas de juguete” que tienen un interés limitado para los estudiantes, o requieren un conocimiento sobre el campo donde los datos fueron adquiridos para poder sacarles partido. Estos problemas pueden ser abordados usando datos locales o regionales.

Muchas ciudades y regiones en todo el mundo están invirtiendo una gran cantidad de recursos en publicar libremente sus datos, en proyectos de *ciudades inteligentes* [5]. Esto abre la puerta a aplicar distintas técnicas para resolver cuestiones de interés para los ciudadanos, administraciones, negocios e investigadores. El uso en clase de datos urbanos o regionales para construir modelos se aproxima bastante a un proyecto real ya que los datos deben ser limpiados; y para ser útil a la sociedad, los modelos creados deben ser fáciles de usar. Además, estos datos tienen la ventaja adicional de tratar cuestiones familiares para los estudiantes.

En este artículo presentamos una experiencia llevada a cabo en una asignatura de inteligencia artificial donde los estudiantes han desarrollado, de principio a fin y usando técnicas de aprendizaje profundo, diversos proyectos de visión por computador y procesamiento de lenguaje natural usando datos de la comunidad autónoma donde se encuentra nuestra universidad. Además presentamos las herramientas y el flujo de trabajo empleados para organizar la experiencia. Finalmente,

terminamos presentando los retos a los que nos hemos enfrentado y las lecciones aprendidas.

2. Trabajo relacionado

En la actualidad existe una demanda creciente tanto en la industria como en el entorno académico de especialistas en aprendizaje profundo, así como de áreas relacionadas como son el aprendizaje automático, la inteligencia artificial y la ciencia de datos [22]. Esto ha llevado al desarrollo de cursos abiertos en plataformas como Coursera o edX [22], la creación de grados y másteres, y la incorporación de temas relacionados con el aprendizaje profundo al currículo de diversas asignaturas [4].

Un reto que aparece en el diseño de una asignatura que aborde temas de aprendizaje profundo es la creación de prácticas que no estén basada únicamente en ejemplos de juguete. Las tareas basadas en estos ejemplos sirven para ilustrar y enseñar conceptos; sin embargo, las soluciones que se dan para esos ejemplos normalmente no generalizan a situaciones más complejas. Por esto es necesario diseñar tareas que muestren como las técnicas aprendidas en los ejemplos de juguete se pueden extrapolar a problemas reales. Para abordar este reto podemos usar ideas empleadas en asignaturas de ciencia de datos donde existen proyectos para enseñar estadística [16], visualización de datos [9], o búsqueda de información [6].

Una cuestión importante a la que nos enfrentamos a la hora de crear dichas tareas es la búsqueda de bancos de datos, o datasets, que resulten interesantes. Existen diversas fuentes y aproximaciones para realizar dicha búsqueda: plataformas de datasets públicos (por ejemplo, Kaggle o Google Dataset Search), crear un dataset propio [18], o usar datos abiertos [15, 20]. Las tres alternativas tienen sus ventajas e inconvenientes. Los datasets públicos disponibles en plataformas como Kaggle son interesantes, pero los proyectos creados con dichos datasets se centran en la tarea de entrenar un modelo, y requieren conocimientos previos sobre el contexto de los datos para poderles sacar su máximo partido. La segunda alternativa, que consiste en que los estudiantes creen sus propios datasets, consigue que los estudiantes se involucren en el proyecto ya que este surge de sus propias ideas. Sin embargo,

el proceso de generar datos suficientes es una tarea que requiere bastante tiempo y recursos. En este trabajo exploramos la última alternativa: el uso de datos abiertos.

Los datos abiertos regionales son familiares para los estudiantes, y pueden servir para mostrar todas las fases de un proyecto de aprendizaje automático [20]. Sin embargo puede resultar complejo encontrar los datos suficientes para llevar a cabo proyectos interesantes [15]. Además la mayoría de datos abiertos que son liberados son datos estructurados, y los algoritmos de aprendizaje profundo no son especialmente útiles con este tipo de datos sino que brillan cuando se aplican a datos no estructurados como imágenes o texto. En este artículo, presentamos nuestra experiencia donde abordamos los problemas involucrando a personal de la comunidad autónoma. En concreto, dicho personal nos ayudó a definir los problemas a analizar y a identificar las fuentes de datos a utilizar.

3. Descripción de la experiencia

Esta iniciativa fue llevada a cabo en el contexto de la asignatura de Inteligencia Artificial, optativa del grado en Ingeniería Informática y del grado en Matemáticas. En esta asignatura había cuatro temas: búsqueda en espacio de estados, aprendizaje automático, visión por computador, y procesamiento del lenguaje natural. En los dos últimos temas se presentan tanto las técnicas tradicionales como las más modernas basadas en aprendizaje profundo. El objetivo de la experiencia fue el desarrollo completo de proyectos de visión por computador y procesamiento de lenguaje natural empleando técnicas de aprendizaje profundo. Cabe destacar que las técnicas de aprendizaje profundo explicadas en el curso se centraban en los problemas de clasificación de imágenes y texto; sin embargo, los proyectos llevados a cabo por los estudiantes requerían la aplicación de otro tipo de técnicas de aprendizaje profundo.

En la asignatura estaban matriculados 25 estudiantes. Los estudiantes se dividieron en 9 grupos de entre 2 y 4 personas para realizar proyectos sugeridos por el personal de la Comunidad Autónoma donde se encuentra nuestra universidad. Un resumen de dichos proyectos puede verse en el cuadro 1. Los 9 proyectos se pueden clasificar en 5 tareas: reconocimiento facial y búsqueda de información (1 proyecto), coloreado de imágenes (2 proyectos), segmentación de imágenes (1 proyecto), clasificación de imágenes (4 proyectos), y clasificación de texto (1 proyecto).

3.1. Descripción de los proyectos

Proporcionamos a continuación una breve descripción de los proyectos llevados a cabo y de las técnicas empleadas por los estudiantes para abordarlos.

Búsqueda de personas en imágenes. Dada la foto de una persona, el objetivo de este proyecto era identificar todas las imágenes del portal de noticias de la comunidad autónoma donde aparecía dicha persona. Este proyecto requería la aplicación de técnicas de detección y reconocimiento facial, como OpenFace [8] o DeepFace [23], y también métodos del campo de la búsqueda de imágenes; por ejemplo, *VP-trees* [2].

Coloreado de imágenes. En la experiencia ha habido dos proyectos relacionados con el coloreado de imágenes antiguas. El primero se centraba en imágenes aéreas disponibles gracias a la infraestructura de datos espaciales de la comunidad autónoma; y el segundo se centró en imágenes del registro de fotografías de la comunidad (una página web gestionada por la comunidad donde cualquier puede subir imágenes antiguas de la región). Estos proyectos se llevaron a cabo utilizando ideas del proyecto DeOldify¹.

Detección de casas en imágenes aéreas. Los integrantes de este proyecto se enfrentaban al problema de detectar casas en imágenes aéreas proporcionadas por la infraestructura de datos espaciales de la comunidad. Este proyecto se enmarca en la tarea de segmentación semántica donde la arquitectura U-net es ampliamente utilizada para segmentar edificios en imágenes aéreas [1].

Clasificador de noticias. El equipo encargado de este proyecto debía crear un clasificador de noticias de la página oficial de la comunidad. Hasta ahora, las noticias se organizaban de forma manual en departamentos, y el objetivo del proyecto era automatizar esta tarea. Para ello los estudiantes utilizaron una aproximación llamada ULMFit [13].

Clasificación de imágenes. En la experiencia hubo 4 proyectos relacionados con la tarea de clasificar imágenes. Dos de los proyectos usaron imágenes del registro fotográfico de la comunidad; uno para clasificar las imágenes en un conjunto de categorías fijo (incluyendo categorías como colegio, familia, o militar); y el otro para datar las imágenes. Los otros dos proyectos emplearon imágenes de obras del museo, y se encargaban de clasificar las imágenes dependiendo del tipo de pieza (escultura, pintura, etc.), y de datar las imágenes en distintas épocas. Los modelos entrenados para estos proyectos utilizaron la aproximación presentada en la librería FastAI².

El flujo de trabajo seguido para llevar a cabo estos proyectos se resume a continuación.

3.2. Flujo de trabajo de los proyectos

Como primer paso en esta iniciativa, el profesorado de la asignatura se reunió con el personal de la comu-

¹<https://github.com/jantic/DeOldify>

²<https://www.fast.ai/>

Nombre proyecto	# miembros	Tareas	Librerías	Despliegue
Búsqueda de personas en imágenes	4	Reconocimiento facial	Keras	Aplicación escritorio
Coloreado de imágenes antiguas	3	Coloreado de imágenes	FastAI	Colab
Coloreado de imágenes aéreas antiguas	3	Coloreado de imágenes	FastAI	Colab
Detección de casas en imágenes aéreas	3	Segmentación de imágenes	FastAI	Colab
Clasificador de noticias	3	Clasificación texto	FastAI	Colab
Clasificación de imágenes antiguas	2	Clasificación de imágenes	FastAI	Colab
Datando imágenes antiguas	2	Clasificación de imágenes	FastAI	Binder
Datando obras del museo	2	Clasificación de imágenes	FastAI	Web app
Clasificación de obras del museo	3	Clasificación de imágenes	FastAI	Colab

Cuadro 1: Lista de proyectos llevados a cabo en la experiencia

nidad autónoma para identificar un conjunto de tareas, y sus datasets asociados. Estas reuniones fueron fundamentales para encontrar un conjunto de problemas que fueran abordables en la asignatura. En concreto, nos centramos en proyectos en los que existieran datos suficientes y que estuvieran parcialmente anotados. El siguiente paso consistió en proporcionar a los estudiantes un resumen de los distintos proyectos para que formaran equipos y eligieran un proyecto (los proyectos fueron asignados mediante una estrategia *first-in-first-out*). Después de la asignación, el profesorado creó una serie de tareas a través de GitHub classroom³ que da acceso a un repositorio privado de GitHub a cada equipo. Cada repositorio solo contenía un fichero LÉEME que explicaba cómo acceder a los datos del proyecto, y algunas ideas y enlaces donde se explicaba cómo abordar el proyecto.

La primera tarea a la que se enfrentaron los estudiantes fue el proceso de adquirir, limpiar y anotar sus datos. Todos los datasets estaban disponibles a través de APIs públicas; sin embargo, la anotación asociada no estaba accesible desde el mismo sitio. Por ejemplo, en los proyectos del museo, existe una base de datos para cada obra del museo donde se proporciona su nombre y tipo, y un enlace a una imagen de la obra; sin embargo, las imágenes de las obras están disponibles en un repositorio distinto, por lo que los estudiantes tuvieron que emparejar ambas fuentes de información. Además la mayoría de los datasets se encuentra parcialmente anotados por lo que los estudiantes tuvieron que hacer una limpieza previa antes de utilizarlos.

Una vez que estos pasos de pre-procesado fueron llevados a cabo, los estudiantes debieron entrenar y evaluar sus modelos usando Python como lenguaje de programación, y librerías de aprendizaje profundo como Keras⁴ o FastAI. Como entorno de desarrollo, se usaron cuadernos de Jupyter [17], una herramienta *open-source* que se ejecuta en cualquier navegador y que permite combinar bloques de texto y código. Es posible ejecutar los cuadernos de Jupyter de forma lo-

cal; sin embargo, a la hora de entrenar modelos de aprendizaje profundo es necesario el uso de hardware específico como GPUs o TPUs, y la mayoría de estudiantes no tienen acceso a dichos recursos. Por ello, los cuadernos de Jupyter se ejecutaban de forma online en Google Colaboratory⁵, un entorno pre-configurado con las librerías esenciales de aprendizaje automático y aprendizaje profundo que da acceso gratuito a GPUs y TPUs a través de una cuenta de Google. Es importante notar que Google Colaboratory se puede conectar con los repositorios de GitHub, por lo que los estudiantes podían guardar fácilmente su progreso en el repositorio de su equipo. Todas las herramientas mencionadas anteriormente habían sido previamente introducidas a los estudiantes en la asignatura. La única funcionalidad desconocida para ellos era el acceso de equipos a un repositorio de GitHub ya que el resto de prácticas del curso eran individuales.

Una vez entrenados los modelos, los estudiantes los evaluaron usando conjuntos de test independientes a los usados para entrenar sus modelos. Todos los modelos salvo el encargado de colorear imágenes antiguas obtuvo una precisión superior al 90 % en su tarea correspondiente (notar que dependiendo del problema se usaban distintas métricas). Por lo tanto, aunque hay espacio para la mejora, se puede considerar que los modelos fueron exitosos y resolvían la tarea asignada.

Finalmente los estudiantes tenían que desplegar sus modelos de manera que fueran fáciles de usar. La mayoría de los equipos, 6 de 9, decidieron usar los formularios de Google Colaboratory, uno creó una aplicación de escritorio, otro una aplicación web, y el último desplegó su modelo usando Binder⁶.

Para poder evaluar los proyectos, los estudiantes tuvieron que documentar todo el proceso seguido durante el proyecto en cuadernos de Jupyter que debían ser almacenados en su repositorio de GitHub. Además los estudiantes tenían que presentar su trabajo en una exposición pública, y preparar un vídeo de 2 minutos donde explicaran de manera no técnica su trabajo.

³<http://classroom.github.com>

⁴<https://keras.io/>

⁵<https://colab.research.google.com/>

⁶<https://mybinder.org/>

4. Discusión

En esta sección presentamos las lecciones aprendidas durante esta experiencia, y comentamos los retos a los que nos hemos enfrentado.

4.1. Lecciones aprendidas

En primer lugar presentamos las lecciones aprendidas. Estas recomendaciones no se basan únicamente en la opinión del profesorado, sino que también se ha tenido en cuenta el grado de satisfacción de los estudiantes. Para obtener la opinión de los estudiantes se llevó a cabo una encuesta anónima y voluntaria desarrollada con Google Forms. La encuesta consistía de 4 secciones (valoración de GitHub, valoración de los cuadernos de Jupyter y Colab, valoración de los proyectos, y comentarios). Las secciones de valoración consistían de una lista de preguntas que seguían una escala de 4 puntos de Likert yendo de 1 (completamente en desacuerdo) a 4 (completamente de acuerdo), y la sección de comentarios permitía a los estudiantes introducir comentarios adicionales sobre la experiencia. A la encuesta respondieron 15 de los 25 estudiantes.

Pasamos a describir en primer lugar las lecciones aprendidas desde un punto de vista organizativo.

Como sugerían trabajos previos [18], es muy beneficioso involucrar a los estudiantes en proyectos que sean cercanos a su día a día, por lo que el uso de datos regionales es perfecto para esto. Esto se puede ver en las presentaciones de los vídeos donde, por ejemplo, los miembros del equipo encargado de colorear imágenes aéreas mostraron imágenes coloreadas de los pueblos donde viven sus familias; o en el proyecto de búsqueda de personas, donde los miembros del equipo se buscaron a sí mismos en el portal de noticias de la comunidad. En la encuesta anónima, todos los estudiantes afirmaron que habían disfrutado trabajando con datos reales. De hecho, uno de los estudiantes escribió el siguiente comentario “*Los trabajos para mí están genial. Te ayudan a aplicar lo que has aprendido durante el curso a un problema real*”.

Una de las mejores decisiones que tomamos fue la de reunirnos con el personal de la comunidad autónoma. Dichas reuniones nos ayudaron a delimitar un conjunto de problemas que encajara con los temas explicados en la asignatura, resolviendo así el problema de encontrar datos suficientes e interesantes en datos abiertos [15]. En estas reuniones, el profesorado pudo explicar al personal de la comunidad lo que los estudiantes podían llegar a hacer, y también establecer ciertos límites a sus expectativas (todos los proyectos eran pruebas de concepto que debían ser desarrolladas en un tiempo limitado, aproximadamente unas 25 horas). En este primer año de la experiencia, solo el profesorado se reunió con el personal de la comunidad. En los

próximos años planeamos incorporar a los estudiantes a dichas reuniones ya que ahora el personal de la comunidad sabe qué tipo de proyectos se pueden llevar a cabo. Sin embargo, es necesario que el profesorado siga involucrado en las reuniones para estimar el tiempo necesario para abordar los distintos proyectos debido a que los estudiantes no cuentan con la experiencia necesaria para realizar esas estimaciones.

Otra de las decisiones que resultó ser acertada fue incluir un fichero LÉEME en los repositorios de los estudiantes. Inicialmente cada equipo tenía acceso a un repositorio privado de GitHub que contenía únicamente un fichero LÉEME. En dicho fichero, el profesorado explicaba cómo acceder a los datos, y proporcionaba enlaces e información de cómo abordar el proyecto. Entre la documentación que se proporcionaba a los estudiantes, se incluyeron enlaces a repositorios libres que incluían cuadernos de Jupyter abordando proyectos similares. Nuestra primera recomendación para los estudiantes fue que ejecutaran dichos cuadernos y que los usaran como base para sus proyectos. De este modo, los equipos tenían un punto de partida para saber cómo organizar sus datasets, y entrenar sus modelos. Además el fichero LÉEME contenía enlaces a tareas de ampliación, como el uso de arquitecturas novedosas para la clasificación de imágenes, o la aplicación de métodos de *ensemble*. Estos temas de investigación permitían a los equipos profundizar en las técnicas de aprendizaje profundo.

La última lección aprendida desde el punto de vista organizativo tiene que ver con los vídeos realizados por el estudiantado. Como se ha comentado anteriormente se pidió a los estudiantes que prepararan vídeos cortos y accesibles, desde un punto de vista técnico, donde explicaran su trabajo. El objetivo era doble. Por un lado queríamos que los estudiantes hicieran el esfuerzo de resumir su trabajo de una manera atrayente. Por otro lado, dichos vídeos servían para presentar el trabajo realizado al personal de la comunidad, y mostrar a otros miembros de la comunidad el tipo de tareas que podían ser resueltas mediante técnicas de aprendizaje profundo. Estos vídeos son importantes de cara a continuar con la experiencia en próximos años con nuevos proyectos.

Pasamos ahora a centrarnos en las lecciones aprendidas desde un punto de vista técnico. Estas lecciones pueden aplicarse no solo a proyectos basados en datos abiertos sino a cualquier proyecto de aprendizaje profundo.

El entrenamiento de modelos de aprendizaje profundo desde cero es una tarea que requiere una cantidad ingente de datos y recursos computacionales; y, en la mayoría de los casos, no es posible tener acceso a tal cantidad de recursos. Este problema puede resolverse usando una técnica llamada *transfer learning* [3]. Este

método reusa un modelo entrenado en una tarea origen en una nueva tarea destino. Esto reduce considerablemente la cantidad de datos y tiempo que son necesarios para construir un modelo de aprendizaje profundo preciso. Esta técnica puede ser empleada en prácticamente cualquier proyecto de aprendizaje profundo, por lo que es importante introducirla a lo largo de la asignatura de cara a que los estudiantes puedan emplearla en sus proyectos.

La siguiente lección aprendida tiene que ver con el almacenamiento del código de los estudiantes. En esta experiencia, todo el código y modelos producidos en los proyectos se almacenaron en repositorios de GitHub. En concreto, todos los repositorios pertenecen a una “*organization*” gestionada por el profesorado de la asignatura. De esta manera, los estudiantes solo tienen acceso al repositorio de su equipo, y el profesorado puede acceder a todos los repositorios de los estudiantes. Esta aproximación no solo simplifica la colaboración entre los miembros de un equipo, sino que también mejora la comunicación con el profesorado, ya que pueden acceder directamente al código de los estudiantes cuando estos necesitan ayuda. Además, el profesorado puede seguir el progreso de los estudiantes dado que es posible monitorizar los *commits* realizados y sus autores.

En la encuesta llevada a cabo, el uso de GitHub tuvo una valoración positiva. En concreto, todos los estudiantes estaban satisfechos con el uso de esta herramienta en los proyectos, y pensaban que los repositorios les ayudaron a gestionar mejor su código. Además, la mayoría de los estudiantes (el 93,3 %) consideraban que GitHub es fácil de usar y que facilita el trabajo en equipo. Por último, todos los estudiantes opinaban que esta herramienta debería ser usada en otras asignaturas.

La siguiente lección que hemos aprendido tiene que ver con Google Colaboratory. Este entorno, que proporciona acceso gratuito a GPUs, ha demostrado ser suficiente para entrenar los modelos de aprendizaje profundo desarrollados en los proyectos. Un problema que surge al usar Google Colaboratory es la limitación de 12 horas de uso (es decir, tras 12 horas de usar el entorno, este se reinicia). Esto forzaba a que los estudiantes tuvieran que guardar *checkpoints* intermedios para evitar perder su trabajo ya que en ocasiones 12 horas no eran suficientes para completar el proceso de entrenamiento. Sería posible entrenar modelos por más tiempo, sin necesidad de los puntos de guardado intermedios, usando entornos cloud como Amazon AWS o la plataforma en la nube de Google Cloud (ambos ofrecen créditos gratuitos para su uso por parte de estudiantes); sin embargo, la configuración y gestión de estos entornos es más compleja, requieren información financiera (aunque no se realicen cargos), y se encuentran fuera del alcance de los objetivos de este curso.

También pedimos la opinión de los estudiantes sobre este entorno, y todos los estudiantes encuestados mostraron su satisfacción con el uso de Google Colaboratory. También opinaban que la integración de esta herramienta con GitHub era correcta y fácil de usar.

Sin embargo, este entorno tiene ciertas problemáticas. El aprendizaje profundo está en constante evolución, lo que supone que las librerías que implementan técnicas de aprendizaje profundo se actualicen de forma habitual. Esto tiene grandes beneficios (ya que los errores se solucionan de manera rápida, y se incluyen nuevos métodos que van surgiendo), pero también desventajas (en algunos casos, el código que funcionaba para una versión de una librería deja de funcionar al actualizarla). Estos problemas pueden resolverse en un ordenador local mediante el uso de entornos virtuales donde las versiones de la librería son fijadas en la máquina virtual; pero, Google Colaboratory actualiza de forma habitual las versiones de las librerías incluidas en su máquina virtual, y esto es un problema por dos razones. En primer lugar, algunos estudiantes se encontraban con que parte de su código dejaba de funcionar de un día para otro; y, en segundo lugar, el código proporcionado en los tutoriales online sugeridos por el profesorado, y que eran usados como base por los estudiantes, producían errores al ejecutarse. Para abordar este problema, se proporcionó a los estudiantes un fichero de requisitos que permite instalar versiones concretas de las librerías necesarias en Google Colaboratory por medio del entorno *pip*⁷. Esto permite a los estudiantes trabajar en un entorno donde es seguro que van a poder reproducir sus resultados.

Otra lección aprendida tiene que ver con las librerías de aprendizaje profundo. En la actualidad existen dos librerías principales de aprendizaje profundo que son usadas tanto en el entorno industrial como académico: Tensorflow y Pytorch [10]. Sin embargo, para proyectos como los presentados en nuestra experiencia, es preferible usar una librería que proporcione una manera sencilla de acceder a distintos tipos de modelos, y que además implemente una serie de buenas prácticas. Durante la asignatura, se presentaron dos de esas herramientas a los estudiantes: Keras y FastAI. Para los proyectos, los equipos podían emplear cualquier librería, pero es destacable que solo uno de ellos usó una librería distinta a FastAI. Esto se debe principalmente a que esta librería proporciona una API amigable y permite entrenar modelos actuales con unas pocas líneas de código.

La última lección aprendida tiene que ver con cómo gestionaron los datos los equipos. En el entorno de Google Colaboratory, los datos solo se conservan en la sesión, por lo los estudiantes deben cargar los datos cada vez que se conectan al entorno. Para abordar este

⁷<https://pypi.org/project/pip/>

problema, les pedimos a los estudiantes que crearan un cuaderno de Jupyter dedicado únicamente a descargar y limpiar los datos, y una vez que estos estaban procesados, los estudiantes debían subir los datos a un sistema online de gestión de ficheros (la mayoría de los equipos usaron OneDrive ya que nuestra universidad ofrece un plan gratuito para los estudiantes, pero otros equipos usaron Google Drive o Dropbox). De esta forma, los miembros de cada equipo, y también el profesor, podía acceder fácilmente a una versión limpia de los datos.

4.2. Retos

Durante esta iniciativa también nos enfrentamos a una serie de retos que deben ser tenidos en cuenta al realizar este tipo de experiencias.

Uno de los principales retos de esta iniciativa fue la carga de trabajo del profesorado tanto en términos de organización como de supervisión de los proyectos. Por lo tanto, es necesario que el profesorado tenga tiempo para organizar los proyectos (es decir, contactar y reunirse con el personal de la comunidad autónoma, y preparar material para guiar a los estudiantes) y también proporcionar consejo individualizado a cada equipo. Además, en este tipo de proyectos es difícil prever las dificultades que los estudiantes se pueden encontrar, y esto supone una carga adicional de trabajo en términos de supervisión.

Un reto relacionado con la supervisión de los estudiantes es reproducir los errores que los estudiantes se encuentran durante el proceso de entrenamiento. Una de las grandes ventajas de usar los repositorios de GitHub y Google Colaboratory es la posibilidad de acceder de manera sencilla al código de los estudiantes, y, por lo tanto, el profesorado puede revisar de manera sencilla los mensajes de error. Sin embargo, en muchas ocasiones, revisar los mensajes de error no proporciona información suficiente para descubrir el problema, y por lo tanto es necesario ejecutar de nuevo el código de los estudiantes. Esto puede ser un reto ya que en muchos casos es necesario ejecutar una cantidad de código considerable antes de llegar al punto donde los estudiantes habían encontrado un error; además, el estado interno de los cuadernos de Jupyter puede dificultar la tarea de reproducir las condiciones exactas donde los estudiantes encontraron el problema [14]. Para minimizar este reto, nuestra recomendación a los estudiantes fue crear cuadernos específicos para cada uno de sus experimentos. De este modo, era más sencillo ayudarles a abordar los problemas que encontraban.

El último reto tiene que ver con que el estudiante tuvo que abordar técnicas fuera del currículum. De los 9 proyectos, 4 de ellos requerían que los estudiantes aplicaran métodos de aprendizaje profundo que no habían sido explicados previamente en la asignatura,

y los otros 5 proyectos incluían tareas adicionales para profundizar en las tareas de clasificación de imagen y texto. Para abordar esta cuestión, proporcionamos a los estudiantes material online con ejemplos y explicaciones de las técnicas a emplear. Sin embargo, un problema que encontramos con esta aproximación fue que los estudiantes se centraban únicamente en entrenar los modelos sin entender lo que hacían internamente. Una solución para abordar este problema se basa en el trabajo del equipo encargado de detectar casas a partir de imágenes aéreas. Este equipo, además de entrenar un modelo de segmentación, creó un cuaderno donde explicaba la arquitectura U-net [21], que es fundamental para problemas de segmentación, y proporcionó un ejemplo de juguete para explicar cómo usarla. En el futuro, planeamos explorar esta aproximación pidiendo a los estudiantes que hagan algo similar.

5. Conclusiones

En este trabajo hemos presentado una iniciativa para aplicar métodos de aprendizaje profundo a la creación de proyectos basados en el uso de datos regionales. En la actualidad, la mayoría de prácticas de aprendizaje profundo emplean o bien datos de juguete, o datos que requieren un conocimiento previo para comprenderlos correctamente. Además dichas prácticas se centran en la tarea de construir modelos, dejando a un lado otras tareas igualmente importantes para un proyecto de aprendizaje profundo. En nuestra experiencia, los estudiantes han trabajado con datos que les resultan familiares, y han estado involucrados en todas las fases del desarrollo de un proyecto de aprendizaje profundo.

El objetivo de este trabajo era presentar las lecciones aprendidas y los retos que el profesorado se puede encontrar al organizar iniciativas similares que empleen datos abiertos. Entre las lecciones aprendidas cabe destacar el beneficio de involucrar a personal de la comunidad autónoma para elegir un conjunto de problemas que motiven a los estudiantes. En nuestro caso la experiencia será continuada en el futuro con nuevos proyectos y métodos de aprendizaje profundo.

Referencias

- [1] Abolfazl Abdollahi, Biswajeet Pradhan y Abdullah M. Alamri: *An ensemble architecture of deep convolutional Segnet and Unet networks for building semantic segmentation from high-resolution aerial images*. Geocarto International, páginas 1–16, 2020.
- [2] Adrian RoseBrock: *Building an Image Hashing Search Engine with VP-Trees and OpenCV*, 2019.

- [3] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan y Stefan Carlsson: *CNN features off-the-shelf: An astounding baseline for recognition*. En *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, CVPRW'14, páginas 512–519, 2014.
- [4] Amy K. Hoover, Adam Spryszynski y Michael Halper: *Deep Learning in the IT Curriculum*. En *Proceedings of the 20th Annual SIG Conference on Information Technology Education*, páginas 49–54, 2019.
- [5] Bhagya Nathali Silva, Murad Khan y Kijun Han: *Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities*. *Sustainable Cities and Society*, 38:697–713, 2018.
- [6] Bhavya, Assma Boughoula, Aaron Green y ChengXiang Zhai: *Collective Development of Large Scale Data Science Products via Modularized Assignments: An Experience Report*. En *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, páginas 1200–1206, 2020.
- [7] Bina Ramamurthy: *A practical and sustainable model for learning and teaching data science*. En *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, páginas 169–174, 2016.
- [8] Brandon Amos, Bartosz Ludwiczuk y Mahadev Satyanarayanan: *OpenFace: A general-purpose face recognition library with mobile applications*. Informe técnico, CMU-CS-16-118, CMU School of Computer Science, 2016.
- [9] Deborah Nolan y Jamis Perrett: *Teaching and Learning Data Visualization: Ideas and Assignments*. *The American Statistician*, 70(3):260–269, 2016.
- [10] Horace He: *The State of Machine Learning Frameworks in 2019*. The Gradient, 2019.
- [11] Il-Yeol Song y Yongjun Zhu: *Big data and data science: what should we teach?* *Expert Systems*, 33(4):364–373, 2016.
- [12] Jacob Devlin, Ming Wei Chang, Kenton Lee y Kristina Toutanova: *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. En *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volumen 1, páginas 4171–4186, 2019.
- [13] Jeremy Howard y Sebastian Ruder: *Universal Language Model Fine-tuning for Text Classification*. En *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, volumen 1, páginas 328–339, 2018.
- [14] Joel Grus: *I don't like notebooks*. En *The official Jupyter conference*, 2018.
- [15] Joelle Pineau y Pierre Luc Bacon: *Analyzing Open Data from the City of Montreal*. En *Proceedings of the 2nd International Workshop on Mining Urban Data*, páginas 11–16, 2015.
- [16] Johanna Hardin, Roger Hoerl, Nicholas J. Horton y Deborah Nolan: *Data science in statistics curricula: Preparing students to think with data*. *The American Statistician*, 69(4):343–353, 2015.
- [17] Jupyter Development Team: *Jupyter Notebooks — a publishing format for reproducible computational workflows*. En *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, páginas 87–90, 2016.
- [18] Kaiman Zeng, Yancheng Li, Yida Xu y Nansong Wu Di Wu: *Introducing AI to Undergraduate Students via Computer Vision Projects*. En *Proceedings of the Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, páginas 7956–7957, 2018.
- [19] Kaiming He, Xiangyu Zhang, Shaoqing Ren y Jian Sun: *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*. En *Proceedings of the IEEE International Conference on Computer Vision*, páginas 1026–1034, 2015.
- [20] Matthew Love, Charles Boisvert, Elizabeth Uru-churtu y Ian Ibbotson: *Nifty with Data: Can a Business Intelligence Analysis Sourced from Open Data Form a Nifty Assignment?* En *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, páginas 344–349, 2016.
- [21] Olaf Ronneberger, Philipp Fischer y Thomas Brox: *U-Net: Convolutional Networks for Biomedical Image Segmentation*. En *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, volumen 9351 de *Lecture Notes in Computer Science*, páginas 234–241, 2015.
- [22] Roberta Kwok: *Junior AI Researchers are in Demand by Universities and Industry*. *Nature*, 568(7753):581–583, 2019.
- [23] Yaniv Taigman, Ming Yang, Marc Ranzato y Lior Wolf: *DeepFace: Closing the Gap to Human-Level Performance in Face Verification*. En *2014 IEEE Conference on Computer Vision and Pattern Recognition*, páginas 1701–1708, 2014.
- [24] Yu Li, Chao Huang, Lizhong Ding, Zhongxiao Li, Yijie Pan y Xin Gao: *Deep learning in bioinformatics: Introduction, application, and perspective in the big data era*. *Methods*, 166:4–21, 2019.

Silence: un framework de apoyo a la docencia de desarrollo web

Agustín Borrego, Daniel Ayala, Fernando Sola, Inma Hernández, David Ruiz

Departamento de Lenguajes y Sistemas Informáticos.

Universidad de Sevilla

41012 Sevilla

{borrego, dayalal, fsola, inmahernandez, druiz}@us.es

Resumen

La programación web es, actualmente, una de las vertientes más importantes de la informática en la industria nacional e internacional. Un aspecto muy relacionado, aunque en muchas ocasiones impartido de manera separada, es la enseñanza sobre bases de datos y el acceso a los datos contenidos en las mismas.

Con objeto de unir más estrechamente la enseñanza de estas dos ramas, en el presente trabajo presentamos Silence, un framework que provee un entorno unificado para el desarrollo de bases de datos relacionales que den soporte a una aplicación, y de aplicaciones web que consuman los servicios provistos por la base de datos a través de una API RESTful moderna.

Silence permite simplificar en gran medida el despliegue de endpoints RESTful sobre una base de datos relacional, posibilitando un aprendizaje del desarrollo web más ágil y una diferenciación clara entre back-end y front-end. Además, da soporte al aprendizaje basado en proyectos, siendo éstos autocontenidos y facilitando su despliegue, desarrollo y evaluación.

El framework se encuentra actualmente en uso por más de 500 alumnos en tres titulaciones de grado, y está disponible libremente como código abierto.

Abstract

Web development is currently one of the most prominent fields in the computing industry, both local and international. Database management and usage stands out as a highly related field, though it is often taught independently.

In this work we introduce Silence, a framework whose goal is to allow for a joint teaching of databases and web development. Silence provides a unified environment for developing relational databases to store an application's data, and web applications that use such databases through a modern RESTful API.

Silence greatly simplifies the process of deploying RESTful endpoints to interact with a database, resul-

ting in a more agile web development learning process and in a very clear distinction between back-end and front-end. Furthermore, Silence is oriented towards project-based learning, offering self-contained projects that are easy to develop, deploy and evaluate.

Our framework is currently actively used by more than 500 students in three different degrees, and it is freely available as open source software.

Palabras clave

Back-end, front-end, APIs REST, desarrollo web, framework.

1. Introducción

El desarrollo web sigue siendo una de las áreas de conocimiento más demandadas en el entorno empresarial, siendo los perfiles más demandados los de desarrollador back-end, front-end o full-stack. Incluso en perfiles que no son estrictamente de desarrollador web, es frecuente encontrar el conocimiento de estas tecnologías como uno de los requisitos deseables. En España, estas competencias suelen cubrirse mediante dos asignaturas diferentes.

Sin embargo, el rápido ritmo de evolución de las tecnologías web hace difícil mantener actualizado el currículum académico de las asignaturas relacionadas sin caer en algunos errores comunes: por ejemplo, dedicar demasiado tiempo a que los alumnos aprendan muchas tecnologías diferentes, mientras que no llegan a asimilar los conceptos básicos y más perdurables de la materia (los que, a la larga, más útiles pueden resultarles), o diseñar asignaturas que intentan enseñar las más recientes tecnologías y quedan obsoletas en no demasiado tiempo [2].

El diseño de un stack tecnológico adecuado para este tipo de asignaturas, por tanto, pasa por lograr un equilibrio entre tecnologías actuales y que vayan a resultar útiles a los alumnos de forma inmediata en su vida

profesional (Python, JavaScript, AJAX, CSS, HTML, SQL, APIs REST) y darles unos fundamentos teórico-prácticos sólidos de una duración más a largo plazo (trazabilidad entre los requisitos, el modelado y la base de datos). No está claro dónde debería estar el punto óptimo de equilibrio, y es por ello que no existe un consenso claro (ni en cuanto a tecnologías ni en cuanto a bibliografía) sobre cómo diseñar estas asignaturas. Lo único que parece claro es que, por su naturaleza, la metodología que se use en ellas debería ser aprendizaje basado en proyectos (ABP) [1]. Algunos autores consideran que la clave para tomar esta decisión está en el punto de madurez de los alumnos, siendo conveniente simplificar el stack tecnológico a los alumnos de primeros cursos y añadiendo complejidad a medida que van pasando a cursos superiores [5].

Nuestra propuesta se enmarca en la experiencia en dos asignaturas de 2º curso en la Universidad de Sevilla, con alumnos que únicamente han adquirido conceptos básicos de programación en una asignatura previa, por lo que un enfoque tecnológico más simple está justificado. Con este fin, hemos desarrollado Silence, una herramienta integral para el desarrollo de aplicaciones web, que permite a los alumnos implementar rápidamente el back-end y/o el front-end de estas aplicaciones sin necesidad de conocimientos avanzados de programación y sin depender de frameworks destinados a uso más profesional que puedan quedar obsoletos, además de resultar difíciles de usar para alumnos de segundo curso. Las principales ventajas de Silence son, entre otras: la reducción en el tiempo de desarrollo frente a frameworks más complejos, la reducción en la curva de aprendizaje de las tecnologías que permite dedicar más tiempo del currículum a fundamentos teóricos básicos, la posibilidad de implementar de forma sencilla pruebas y la estabilidad del stack tecnológico, frente a frameworks más volátiles. Silence se ha implantado en tres titulaciones de grado con un total de 524 alumnos, y las impresiones de alumnos y profesores son muy positivas.

El resto del artículo se organiza de la siguiente forma: en la Sección 2 describimos las asignaturas en las cuales se contextualiza el uso de Silence, en la Sección 3 analizamos el trabajo relacionado, en la Sección 4 describimos la estructura del framework y los proyectos Silence, en la Sección 5 presentamos Noise, una herramienta relacionada, para después en la Sección 6 presentar los resultados obtenidos en su implantación; finalmente, la Sección 7 concluye el artículo.

2. Contexto docente

Este framework se ha desarrollado en el contexto de dos asignaturas obligatorias cuatrimestrales en la Universidad de Sevilla: Introducción a la Ingeniería del

Software y los Sistemas de Información 1 (en adelante, IISSI-1) y 2 (en adelante, IISSI-2), ambas de segundo curso.

IISSI-1 e IISSI-2 abordan dos aspectos complementarios del desarrollo de aplicaciones web, que se corresponden con lo que actualmente se denominan back-end y front-end. La metodología de las dos asignaturas se engloba dentro del conocido como aprendizaje basado en proyectos que ha sido objeto de análisis en muchas asignaturas de nuestra área [4, 6].

La asignatura IISSI-1 cubre todo el proceso de desarrollo de bases de datos relacionales. Esta cuenta con cuatro bloques de contenidos: el primero de ellos sirve para introducir los conceptos básicos relacionados con la Ingeniería del Software y el análisis de requisitos; el segundo se centra en el diseño de bases de datos relacionales normalizadas a partir de una especificación de requisitos; en el tercero se aborda la implementación de una base de datos a partir de un modelo relacional usando SQL; por último, en el cuarto bloque se estudia cómo definir endpoints para interactuar con la base de datos a través de una API REST, y cómo realizar pruebas usando esta misma API.

La asignatura IISSI-2 aborda los aspectos relacionados con el proceso de desarrollo de una interfaz web que pueda consumir datos procedentes de una o varias APIs REST. La asignatura cuenta con tres bloques de contenidos: el primero de ellos introduce las aplicaciones web y cómo diseñar interfaces web teniendo en cuenta aspectos de usabilidad, accesibilidad y experiencia de usuario; el segundo se centra en el desarrollo de interfaces estáticas usando HTML y CSS; finalmente, el tercer bloque se centra en la implementación de comportamiento dinámico usando JavaScript.

La evaluación en ambas asignaturas se lleva a cabo mediante diferentes entregas en la que el alumno ha de desarrollar incrementalmente su proyecto. Una vez el alumno ha llevado a cabo todas las entregas requeridas puede optar a la defensa de su proyecto, en la que se plantean cuestiones sobre los elementos desarrollados y se pide realizar modificaciones sobre los mismos.

3. Trabajo relacionado

Un análisis de las asignaturas existentes sobre programación web reveló que el conjunto de tecnologías necesarias es tan amplio que no es factible abordarlo en una única asignatura; lo habitual es dividir los contenidos en, al menos, dos asignaturas: una dedicada al back-end y otra al front-end [2, 3, 5].

Esto hace que, por ejemplo, al estudiar la asignatura de front-end, la parte de back-end ha de ser simulada o se ha de desplegar una versión de prueba muy sencilla, que permita a los alumnos centrarse en profundizar en los conceptos de front-end (y de forma similar ocurre

a la inversa). Actualmente, las herramientas de simulación de back-end más usadas son:

- **JSON Server:** Permite desplegar endpoints CRUD sobre recursos definidos en un archivo JSON. La implementación de reglas de negocio, por sencillas que sean, es poco intuitiva, y la gestión de sesiones requiere de un plugin separado.
- **MockServer:** Asigna respuestas determinadas a peticiones concretas, o puede actuar como proxy para peticiones más complejas. Tiene los mismos problemas que la alternativa anterior.
- **Mocky:** También permite responder de forma determinada a una petición. Su uso más allá de unos determinados márgenes es de pago.

En el caso de los front-end de prueba, las alternativas son mucho menos numerosas, debido a la mayor dificultad inherente a crear de manera automática una interfaz de usuario de prueba con el nivel mínimo de complejidad necesario.

La alternativa es usar frameworks integrales, que combinen el desarrollo back-end con front-end, cubriendo así todos los aspectos relevantes del desarrollo web. Sin embargo, los frameworks web más usados actualmente (Django, Spring, Laravel, Rails...) imponen un patrón de diseño MVC y requieren un dominio de la tecnología que entra en conflicto con los conceptos básicos que pretendemos transmitir a nuestros alumnos de segundo curso. Incluso aquellos frameworks web populares más sencillos, como Flask para Python, no son alternativas factibles al requerir de aprendizaje especializado, cuyo tiempo podría ser dedicado a la enseñanza de conceptos básicos. Además, las operaciones contra la base de datos quedan demasiado abstraídas para la docencia. Es por esto que otros autores han desaconsejado el uso de frameworks complejos en asignaturas de programación web, principalmente en los primeros cursos [5].

4. Silence

Silence es un framework destinado a facilitar el desarrollo ágil de una API RESTful capaz de ofrecer operaciones sobre recursos almacenados en una base de datos relacional, y de una aplicación web basada en el uso de dichos endpoints con una clara separación entre back-end y front-end. Dado que este conjunto de esquema de base de datos, endpoints REST y archivos web definen una aplicación, Silence hace uso de un enfoque basado en proyectos que unifica y estandariza estos aspectos, y permite un despliegue sencillo mediante una interfaz de línea de comandos. A continuación, se detallan la arquitectura del framework y la estructura de un proyecto Silence.

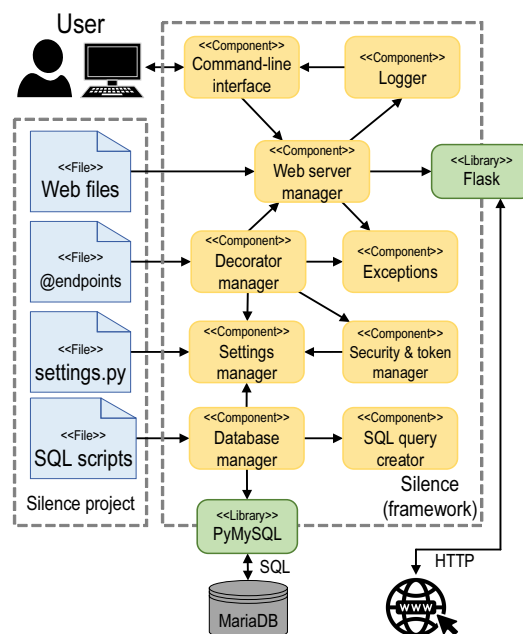


Figura 1: Arquitectura del framework y los proyectos Silence

4.1. Arquitectura de Silence

Silence está desarrollado en Python mediante una arquitectura basada en componentes, y su código fuente está disponible en GitHub¹. También se encuentra disponible en PyPI², el repositorio de paquetes de Python, para su fácil instalación y actualización. Cada componente es responsable de gestionar ciertos aspectos del framework, y expone una interfaz al resto de componentes que requieran sus servicios.

El esquema general de la arquitectura de Silence, y cómo ésta se relaciona con cada uno de los aspectos de un proyecto Silence, se encuentra en la Figura 1. Cada componente se explica brevemente a continuación:

Interfaz de línea de comandos (CLI): Este componente es responsable de proveer una interfaz de consola para la creación, gestión y despliegue de proyectos Silence. Los comandos disponibles son:

- `silence new`: Crea un nuevo proyecto a partir de una plantilla o la URL de un repositorio. Si no se especifica una plantilla, se usa una por defecto.
- `silence list-templates`: Muestra las plantillas de proyectos disponibles.
- `silence createdb`: Efectúa una conexión a la base de datos y ejecuta los scripts SQL incluidos en el proyecto para crear la estructura de tablas e insertar datos iniciales.

¹<https://github.com/IISSI-US/Silence>

²<https://pypi.org/project/Silence/>

- `silence run`: Ejecuta el servidor que da soporte a los endpoints de la API definida en el proyecto y a los archivos web.

Logger: El componente de logging es responsable de estandarizar los mensajes emitidos por las librerías usadas internamente, filtrar aquellos que no son relevantes, y emitir salidas acordes al nivel de log deseado.

Gestor del servidor web: Se encarga de realizar en él la configuración oportuna según los ajustes del proyecto, de cargar los endpoints definidos en el proyecto, y de proveer una serie de endpoints por defecto (salvo que éstos sean desactivados en la configuración del proyecto). Estas funcionalidades por defecto incluyen el registro de nuevos usuarios, el inicio de sesión de usuarios existentes y el listado de todos los endpoints disponibles en el servidor.

Gestor de decoradores: Este componente provee este tipo de objetos, que son utilizados a lo largo del framework para diversas funciones. Uno de los decoradores más relevantes es `@endpoint`, empleado por el usuario para definir endpoints en sus proyectos.

Gestor de excepciones: Es responsable de proveer un conjunto unificado de excepciones, así como de capturar las mismas cuando se producen en distintos puntos del framework para mostrar mensajes de error adecuados.

Gestor de ajustes: Provee una interfaz unificada a la configuración del framework a todos los demás componentes. Además, es el responsable de cargar la configuración individual de cada proyecto, así como de proveer valores por defecto para aquellos parámetros de configuración que no estén definidos en un proyecto.

Gestor de tokens y seguridad: Tiene como función primaria emitir y comprobar *tokens* de sesión, que representan la sesión iniciada de un usuario y que tienen un determinado período de caducidad. Además, es el encargado de proveer una interfaz para almacenar y validar contraseñas de manera criptográficamente segura.

Gestor de la base de datos: Provee una interfaz programática abstracta respecto a la base de datos empleada, facilitando la compatibilidad de Silence con otros posibles sistemas gestores de bases de datos en el futuro. También, lee y ejecuta los scripts SQL del proyecto en la base de datos seleccionada cuando lo solicita el usuario mediante la interfaz de línea de comandos.

Creador de queries SQL: Por último, este componente permite generar este tipo de consultas programáticamente, facilitando una gestión abstracta de la base de datos por parte del componente anterior.

4.2. Estructura de un proyecto Silence

Cada proyecto Silence está compuesto de una serie de elementos principales, que interactúan con los distintos componentes del framework según lo mostrado

en la Fig. 1. En las siguientes subsecciones se pasa a enumerar y explicar cada uno de estos elementos. Como ejemplo, supongamos un proyecto dedicado a la gestión de los departamentos de una universidad. Este proyecto de ejemplo está disponible en GitHub³.

Endpoints: Una de las motivaciones en el desarrollo de Silence es permitir la definición de endpoints de una API RESTful de manera sencilla. La forma de definir endpoints en un proyecto Silence es creando un archivo Python por cada entidad del dominio que se desea que sea accesible mediante la API, dentro de la carpeta `endpoints/` del proyecto. En estos archivos, se pueden definir tantos endpoints como sean necesarios.

El decorador `@endpoint` es usado para definir un punto de acceso a la API. En él se definen la ruta del recurso, el método HTTP a emplear y la consulta SQL asociada. El resultado de la consulta será convertido automáticamente a JSON y devuelto al cliente. Es necesario asociar a cada definición de endpoint una función Python, que puede usarse para validar parámetros recibidos (ver Figura 3). En caso de no realizar ninguna validación, como es el caso de una consulta GET, la función puede no realizar ninguna operación.

También es posible definir rutas con parámetros, que pueden ser capturados y usados en la sentencia SQL asociada al endpoint. A modo de ejemplo, la Figura 2 muestra cómo definir un endpoint que permite acceder a un departamento dado su identificador.

```
@endpoint (
    route="/departments/$depId",
    method="GET",
    sql="SELECT * FROM
        Departments WHERE
        departmentId = $depId"
)
def get_by_id() :
    pass
```

Figura 2: Definición de un endpoint GET con parámetros de ruta

Las partes de una ruta que comienzan por \$ se capturan como parámetros, y pueden ser insertados en la consulta SQL. Estos parámetros se reemplazan de forma segura, evitando ataques por inyecciones SQL.

Los endpoints pueden aceptar también datos enviados en el cuerpo de una petición, por ejemplo de un formulario, tanto en formato URL como JSON. Para especificar que se esperan parámetros en el cuerpo de la petición, éstos se incluyen como parámetros de la función asociada y pueden ser accedidos en ella para su

³<https://github.com/IISSI-US/Silence-template-Employees>

validación. Por ejemplo, en la Figura 3 se muestra cómo implementar la creación de un nuevo departamento dados su nombre y su ciudad mediante una petición POST, comprobando que el nombre del departamento tenga al menos 3 caracteres:

```
@endpoint (
    route="/departments",
    method="POST",
    sql="INSERT INTO Departments (
        name, city) VALUES ($name,
        $city) "
)
def insert(name, city):
    if len(name) < 3:
        raise HTTPError(400, "The
        name is too short")
```

Figura 3: Definición de un endpoint POST

En el caso de endpoints que no sean de tipo consulta, el valor devuelto es el identificador del último recurso creado o modificado.

La definición de otras operaciones es análoga, pudiendo crear endpoints DELETE para el borrado de recurso o PUT para su actualización. En este último caso, se pueden combinar parámetros de ruta y parámetros recibidos en el cuerpo de la petición HTTP.

Archivos SQL: Cada proyecto Silence contiene una carpeta `sql/` que incluye todos los archivos SQL necesarios para crear la base de datos del proyecto: tablas, datos iniciales, disparadores, procedimientos, etc. Esto permite que los proyectos sean autocontenidos, dado que tienen toda la información necesaria para establecer la base de datos a un estado listo para su uso mediante el comando `silence createdb`.

El orden de ejecución de los scripts SQL cuando se invoca la orden anterior se define en la configuración del proyecto.

Configuración del proyecto: Los proyectos contienen un archivo `settings.py` que gobierna un gran número de aspectos relacionados con la configuración del mismo, tales como los datos de acceso al SBGD, el orden de ejecución de los scripts SQL para la creación de la base de datos, el puerto del servidor web, el control del nivel de log, etcétera. El listado completo de los parámetros de configuración disponibles se encuentra en la documentación online de Silence.

Archivos web: La carpeta `docs/` del proyecto es la raíz de los archivos HTML, CSS y JavaScript que implementan la interfaz web de la aplicación, además de ficheros relacionados como imágenes, fuentes, etc.

Los proyectos plantilla proporcionados siguen una filosofía de estructuración del código web inspirada en un esquema basado en componentes, que facilita el

desarrollo de código ordenado y reutilizable. Los aspectos clave son los siguientes:

- Cada vista se implementa mediante un archivo `.html`.
- Cada archivo `.html` tiene asociado un archivo `.js` homónimo en la carpeta `docs/js/`, responsable de gestionar los eventos de la vista y de proporcionar el punto de entrada.

Asimismo, el código JavaScript de la aplicación, localizado en la carpeta `docs/js/`, se organiza de la siguiente manera:

- La carpeta `api/` contiene módulos JavaScript encargados de comunicarse con la API del proyecto mediante una interfaz basada en promesas⁴.
- La carpeta `renderers/` contiene módulos JavaScript cuya función es transformar las diferentes entidades del dominio del problema en representaciones HTML adecuadas.
- La carpeta `utils/` contiene módulos JavaScript con utilidades de propósito general.
- Por último, la carpeta `validators/` contiene módulos responsables de la validación de los formularios de la aplicación.

5. Noise

La homogeneidad en los proyectos y su despliegue nos ha permitido desarrollar, además, una herramienta llamada Noise, que permite a los estudiantes desplegar sus aplicaciones web en un servidor habilitado para tal fin. Noise es, a su vez, un proyecto Silence, cuya API permite a los alumnos desplegar otros proyectos Silence en el mismo servidor, y que posee una interfaz de administración web para el profesorado destinada a gestionar y monitorizar los proyectos del alumnado.

En las secciones siguientes procedemos a explicar el proceso de despliegue y autogestión de proyectos por parte de los alumnos, así como las herramientas puestas a disposición de los profesores.

5.1. Despliegue de proyectos

La API de Noise provee un endpoint `/deploy` que permite desplegar un proyecto Silence en el servidor indicando la URL del repositorio que lo contiene, como se ilustra en la Figura 4.

Si el servidor se encuentra en condiciones de atender la petición de despliegue (comprobando, por ejemplo, que el alumno tiene permitido usar el servicio), responderá con un código HTTP 202 (aceptado), incluyendo

⁴https://developer.mozilla.org/en-US/docs/web/JavaScript/Reference/Global_Objects/Promise

```
POST [...]api/deploy
Content-Type: application/json
{
  "url": "https://github.com/
    alumno/repo_proyecto.git"
}
```

Figura 4: Petición de despliegue de proyecto

un identificador que el alumno puede usar para gestionar su proyecto en el servidor. Un ejemplo de esta respuesta se muestra en la Figura 5.

```
HTTP/1.1 202 ACCEPTED
{
  "code": 202,
  "message": "Accepted",
  "projectId": "
    AdZpPRx4KN8FSOYcEv6VuCY8O"
}
```

Figura 5: Respuesta a la solicitud de despliegue

Se puede consultar el estado de un proyecto desplegado a partir de su identificador, realizando una petición GET al endpoint `/projects/<id>`. La respuesta del servidor dependerá del estado en el que se encuentre el proyecto en cuestión. Como ejemplo, se muestra en la Figura 6 una posible respuesta por parte de Noise a una petición de estado del proyecto desplegado anteriormente, si éste se encuentra en ejecución sin incidencias. En ella, se incluye una URL al propio servidor en la que se puede acceder a la interfaz web del proyecto desplegado, al que le ha sido asignado un puerto determinado del servidor.

El endpoint `/projects/<id>` también puede ser usado para detener un proyecto desplegado, mediante una petición HTTP de tipo DELETE. Cuando un proyecto se detiene, su código fuente se comprime y almacena, y se retiene la información sobre su despliegue para futuras consultas.

5.2. Interfaz web

Al tratarse de un proyecto Silence, Noise también cuenta con una interfaz web. El propósito de ésta es facilitar la consulta y gestión por parte de los profesores de los proyectos desplegados por los alumnos, y mostrar para cada uno de ellos una serie de estadísticas e información de interés.

Esta interfaz está reservada a los profesores, que deberán contar con un usuario ya registrado en la base de datos (el registro de nuevos usuarios está desactivado

```
HTTP/1.1 200 OK
{
  "code": 200,
  "message": "OK",
  "project": {
    "date": "Sun, 07 Feb 2021
      16:58:44 GMT",
    "projectId": "
      AdZpPRx4KN8FSOYcEv6VuCY8O"
    ,
    "repo": "https://github.com/
      alumno/repo_proyecto.git",
    "status": "RUNNING",
    "url": "http://[:5532]"
  }
}
```

Figura 6: Respuesta a la consulta de estado del proyecto desplegado

en Noise). Al entrar, podrán consultar un listado de los proyectos activos en el servidor en ese momento junto con la información básica de cada uno de ellos: puerto en el que está desplegado, identificador, URL del repositorio del alumno, nombre del alumno, y número de *commits* en GitHub del proyecto, así como el número de veces que el proyecto ha sido desplegado en el servidor usando Noise anteriormente. El profesor puede ordenar los proyectos y realizar búsquedas por cualquiera de estos parámetros. Además, se dispone de un listado similar de proyectos desplegados anteriormente en el servidor, pero detenidos por cualquier razón (fallo en el código del proyecto, detención manual por parte del alumno...), para poder conocer el histórico de despliegues realizados en el servidor.

Al acceder a los detalles de cualquier proyecto se muestra información adicional sobre el mismo, como asistencia al profesor para la evaluación del alumno. En esta información se incluyen datos como el número total de *commits* en GitHub, el número de líneas de código añadidas y eliminadas hasta el momento, la distribución de los *commits* a lo largo del tiempo (Fig. 7) y el detalle de todos ellos junto con su fecha y comentario. Otra información relevante mostrada es el número de despliegues anteriores usando Noise a lo largo del tiempo y las fechas de los mismos, junto con la causa que motivó la detención del proyecto en el servidor para ese despliegue (Fig. 8).

6. Evaluación y resultados

Para evaluar la satisfacción del alumnado con el uso de Silence, se les pidió cumplimentar de manera vo-

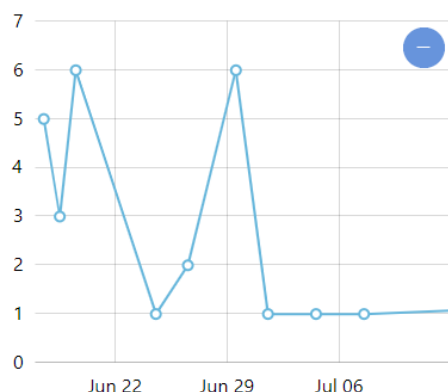


Figura 7: Distribución temporal de *commits* Git de un proyecto Silence desplegado usando Noise

Show entries

Search:

Creation Date	Stop Date	Stop Cause
Tue, 02 Feb 2021 17:18:51 GMT	Tue, 02 Feb 2021 17:19:15 GMT	ADMIN_STOP
Tue, 02 Feb 2021 17:20:21 GMT	Tue, 02 Feb 2021 17:20:22 GMT	USER_STOP
Tue, 02 Feb 2021 17:20:49 GMT	Tue, 02 Feb 2021 17:20:51 GMT	USER_STOP
Tue, 02 Feb 2021 17:27:06 GMT	Tue, 02 Feb 2021 17:28:46 GMT	REPLACED

Figura 8: Listado de despliegues anteriores en Noise para un determinado proyecto

luntaria y anónima una encuesta evaluando diferentes aspectos del framework y su experiencia personal con él. La encuesta fue remitida a 6 grupos de laboratorio en total, repartidos por las diferentes titulaciones y agrupando a un total de 172 alumnos, de los cuales 42 respondieron a la encuesta.

En el cuestionario, se les pidió que valoraran de menor (1) a mayor (5) grado de satisfacción los siguientes aspectos sobre Silence:

- Facilidad de uso y/o instalación.
- Facilidad de configurar la conexión a una base de datos.
- Facilidad de despliegue de endpoints.
- Claridad y ayuda de los mensajes de error.
- Ayuda para comprender y poner en práctica los conceptos teóricos sobre APIs RESTful.

Además, se les pidió que comentaran mediante respuesta libre aquellos aspectos en los que habían encontrado dificultades, y los cambios que sugerirían hacer.

Una encuesta similar fue remitida al profesorado de la asignatura, para que remitieran de forma anónima su opinión sobre el uso de Silence como docentes. Las preguntas sobre satisfacción son casi idénticas, con la excepción de que en la última de ellas se pide responder sí, en su opinión, creen que es una buena herramienta para explicar y poner en práctica el contenido teórico.

Los resultados de esta encuesta, tanto para alumnos como para profesores, se muestran en la Fig. 9, mientras que las respuestas más comunes a las preguntas de texto libre se discuten a continuación.

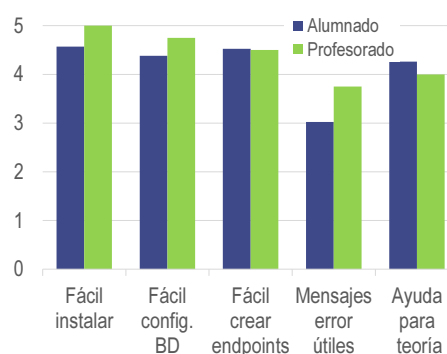


Figura 9: Resultados medios de la encuesta de satisfacción

A tenor de los resultados obtenidos en la encuesta de satisfacción, mostrados en la Fig. 9, podemos concluir que la experiencia de uso de Silence ha sido positiva tanto para el alumnado como para el profesorado. Los datos recopilados muestran, en general, que Silence es fácil de instalar, configurar y usar para desplegar endpoints con los que dar soporte a una API RESTful.

Los usuarios han valorado, además, que se trata de una herramienta útil tanto para enseñar como para poner en práctica los conceptos teóricos impartidos en la asignatura. Creemos que esto es una consecuencia de lo anterior, dado que su facilidad y rapidez de uso permiten a los profesores hacer demostraciones en directo y a los alumnos seguir esas demostraciones en sus propios ordenadores. Así, pueden probar por ellos mismos de manera dinámica la definición de endpoints, y experimentar con posibles cambios.

El cuestionario incluía, también, secciones de texto libre en la que los alumnos y profesores podían hacer comentarios sobre los aspectos a mejorar en el framework y proponer cambios. En general, las respuestas recibidas versaban sobre dos aspectos principales: la gestión de errores y el uso de la consola de comandos como forma de interacción con Silence. Como se aprecia en la Fig. 9, la utilidad y claridad de los mensajes de error ha sido el aspecto peor valorado en general, con un 3.02 sobre 5 de nota media. En sus comentarios, algunos alumnos indicaron que determinados mensajes

de error pueden ser poco intuitivos, no siendo obvio a qué son debidos y cómo solucionarlos. Esta percepción es compartida, aunque en menor medida, en las respuestas emitidas por el profesorado. Por otro lado, el uso de comandos en terminal, actualmente la única forma de interactuar con Silence para iniciar el servidor o realizar la carga inicial de la base de datos, ha resultado confuso para algunos alumnos. Aunque la cantidad y complejidad de los comandos de Silence es reducida, en algunos casos ha supuesto la primera interacción del alumnado con una herramienta de línea de comandos, añadiendo una carga cognitiva adicional.

En futuras versiones de Silence nos centraremos en corregir estos defectos, mejorando la calidad y claridad de los mensajes de error y desarrollando una interfaz gráfica con la que poder interactuar con el framework. Otras mejoras adicionales a desarrollar incluyen permitir definir endpoints mediante un lenguaje más abstracto, no necesariamente usando la sintaxis del lenguaje Python; y la posibilidad de escribir tests automatizables para cada proyecto. Finalmente, una vez terminado el segundo cuatrimestre, evaluaremos de manera similar la experiencia del alumnado con el apartado relativo al desarrollo web y a la estructura de archivos web propuesta por Silence, e identificaremos posibles aspectos de mejora en este sentido.

Por último, puesto que Silence está disponible como un paquete Python en PyPI, el índice de paquetes del lenguaje, hemos recopilado estadísticas sobre el número de descargas que ha tenido a lo largo del tiempo gracias a la herramienta PePy⁵. Desde su publicación inicial y a fecha de la redacción del presente artículo, Silence ha sido descargado un total de 37.153 veces, con una media de alrededor de 100 descargas al día.

7. Conclusiones

En este artículo hemos presentado Silence, un framework orientada a proyectos y destinada a facilitar el desarrollo web. Silence cubre tanto el back-end, mediante un despliegue sencillo de endpoints que implementan una API RESTful sobre una base de datos relacional; como el front-end, incorporando los archivos de la aplicación web en el propio proyecto y recomendando una estructura de archivos HTML, CSS y JS que simplifica la gestión del código. La estandarización de la estructura de los proyectos, y la facilidad de su ges-

tión y despliegue a través de la línea de comandos, nos ha permitido desarrollar otra herramienta adicional, Noise, cuyo objetivo es facilitar a los alumnos el despliegue de sus proyectos en un servidor de la Universidad.

En este primer cuatrimestre, más de 500 alumnos en tres titulaciones diferentes han usado Silence para poner en práctica los conceptos sobre servicios RESTful y APIs aprendidos. Para conocer el grado de satisfacción con el framework, invitamos tanto a alumnos como a profesores a cumplimentar anónimamente una encuesta valorando diferentes aspectos de Silence. Los resultados han sido en general muy positivos, remarcando la facilidad de instalación y uso del framework, así como su utilidad para explicar y mostrar los elementos fundamentales de la asignatura.

Referencias

- [1] Óscar Cánovas Reverte and Félix Jesús García Clemente. Prevención y seguimiento de factores limitantes del trabajo en equipo en experiencias ABP. In *Actas de las XXII Jornadas sobre la Enseñanza Universitaria de la Informática, Jenui 2016*, pages 11–18, 2016.
- [2] Randy Connolly. Facing backwards while stumbling forwards: The future of teaching web development. In *SIGCSE'19*, pages 518–523, 2019.
- [3] Ralph F. Grove. Trends in teaching web-based development a survey of pedagogy in web development courses. In *WEBIST*, pages 361–365, 2007.
- [4] Pengyue Guo, Nadira Saab, Lysanne S. Post, and Wilfried Admiraal. A review of project-based learning in higher education: Student outcomes and measures. *International Journal of Educational Research*, 102:101586, 2020.
- [5] Francesco Maiorana. Extending the database curriculum: From design principles to web and mobile programming. In *Computer Supported Education*, pages 258–271, 2015.
- [6] Carlos J. Villagrà-Arnedo, Francisco J. Gallego-Durán, Faraón Llorens-Largo, and Rafael Molina-Carmona. Movimientos pendulares al situar al estudiante en el centro del proceso de aprendizaje. In *Actas de las XXII Jornadas sobre la Enseñanza Universitaria de la Informática, Jenui 2016*, pages 285–291, 2016.

⁵<https://pepy.tech/project/Silence>

Sistemas interactivos de respuesta: hay vida más allá de los concursos

Óscar Cánovas Reverte

Departamento de Ingeniería y Tecnología de Computadores
Universidad de Murcia
30100 Murcia
ocanovas@um.es

Resumen

Los sistemas interactivos de respuesta son herramientas de aprendizaje muy populares en todos los niveles educativos. Desde sus versiones originales, basadas en el uso de dispositivos específicos llamados '*clickers*', hasta las actuales plataformas en la nube que hacen uso de los dispositivos móviles, han sido una tecnología ampliamente adoptada en la educación superior. Potencialmente, hay diversas maneras de emplear estas herramientas con fines pedagógicos, y estas distintas metodologías persiguen en ocasiones objetivos muy distintos entre sí. No obstante, muchas de las soluciones comerciales existentes suelen publicitar únicamente dinámicas competitivas basada en la gamificación, en forma de concursos, y los datos recolectados en este trabajo a partir de un grupo multidisciplinar de profesorado también arrojan que ese es el uso claramente predominante. A partir de la experiencia propia y de la revisión de la literatura científica relacionada, este trabajo remarca los beneficios que pueden obtenerse aplicando estrategias adicionales. Se mostrará que hay otras posibilidades enriquecedoras más allá de los concursos para repasar conceptos, como por ejemplo su integración en el transcurso de las explicaciones, las posibilidades para lanzar discusiones en grupo o su integración con la metodología de enseñanza a tiempo. Finalmente, el trabajo aporta una guía rápida de uso de estas estrategias con sistemas interactivos de respuesta.

Abstract

Interactive response systems are very popular learning tools at all educational levels. From their original versions, based on the use of specific devices called '*clickers*', to the current platforms that make use of mobile devices, they have been a widely adopted technology in higher education. Potentially, there are different strategies to employ these tools for pedagogical purposes, and these different methodologies sometimes pursue very different objectives. However, many of the

existing commercial solutions tend to advertise only gamification-based competitive dynamics in the form of quizzes, and the data collected in this paper from a multidisciplinary group of faculty also show that this is clearly the predominant use. This paper highlights the benefits that can be obtained by applying additional methodologies, showing that there are other valid possibilities beyond quizzes, such as their integration with explanations, the possibilities to launch group discussions or their suitability for just in time teaching. Finally, the paper also provides a quick guide for using these strategies with interactive response systems.

Palabras clave

Sistemas interactivos de respuesta, estrategias pedagógicas, gamificación

1. Introducción

Los sistemas interactivos de respuesta (también denominados de respuesta de audiencia o, simplemente, cuestionarios interactivos) parecen haberse consolidado como herramientas de aprendizaje móvil en todos los niveles educativos. Desde sus versiones primigenias, basadas en el uso de dispositivos específicos denominados '*clickers*' [8], hasta las plataformas actuales que hacen uso de dispositivos móviles, han sido una tecnología empleada ampliamente en la educación superior [3, 6], con literatura relacionada en las actas de las propias JENUI [12, 20].

Si bien su popularidad es muy elevada y su uso está muy extendido, la realidad es que estas herramientas pueden emplearse de formas muy diversas en función del enfoque pedagógico seguido. Los beneficios obtenidos dependerán de si su uso está correctamente alineado con objetivos pedagógicos bien definidos [4].

En un estudio que se está llevando a cabo acerca de las ventajas pedagógicas del uso de los sistemas interactivos de respuesta, en el cual han participado hasta

ahora 42 docentes de diversas titulaciones en la Universidad de Murcia, se ha comprobado que la estrategia de uso predominante es llevar a cabo cuestionarios de repaso de conceptos en modalidad de concurso. Como se verá, es la estrategia más inmediata puesto que consiste en la adaptación de otras herramientas ampliamente arraigadas como pueden ser los exámenes tipo test realizados en papel. No obstante, existen otras aproximaciones metodológicas basadas en sistemas interactivos de respuesta que pueden ampliar el abanico de recursos disponibles a la hora de diseñar el proceso de enseñanza-aprendizaje. Estas estrategias, expuestas en este trabajo, están adaptadas a partir de otros trabajos publicados por la comunidad científica y están basadas en la experiencia profesional de los autores a la hora de ponerlas en práctica e impartir cursos de formación sobre el uso de sistemas interactivos.

Con el fin de que aquellos lectores interesados en emplearlas puedan disponer de la información necesaria, además de la explicación general de cada estrategia, y las referencias bibliográficas relacionadas, se proporciona como parte de este trabajo una guía rápida de uso de cada una de ellas. El objetivo de dicha guía es dar indicaciones acerca de cómo seleccionar adecuadamente las múltiples opciones que estas herramientas nos proporcionan en función de cada enfoque, lo que permite a su vez identificar los aspectos cruciales y poder compararlos entre sí.

Detrás de estas estrategias no solo se encuentra una forma distinta de emplear las herramientas, sino también enfoques pedagógicos diversos con los que afrontar el proceso de enseñanza-aprendizaje. La adopción de nuevos métodos docentes supone siempre un reto para el profesorado, puesto que supone abandonar prácticas consolidadas y rediseñar los procesos de enseñanza mediante nuevos elementos que se adapten a una realidad social y educativa cada vez más cambiante, como ha demostrado la pandemia de COVID-19. Este trabajo tiene la intención de contribuir a este necesario proceso de adaptación y lo hace mediante un enfoque práctico, ilustrando cómo diversos usos específicos de una familia de tecnologías pueden favorecer la aplicación de distintas metodologías docentes.

El resto del trabajo se estructura de la siguiente forma. En la Sección 2 se describirán las características generales de los sistemas interactivos de respuesta y se mencionarán algunas de las herramientas más populares. En la Sección 3 se proporcionan detalles acerca del estudio del cual se han extraído algunos datos contextuales acerca de la percepción y el uso que hacen los docentes de este tipo de herramientas. A continuación, se presentan las distintas estrategias propuestas en la Sección 4, mientras que la guía de uso se encuentra en la Sección 5. Finalmente, el trabajo presenta unas conclusiones finales en la Sección 6.

2. Sistemas interactivos de respuesta

Un sistema interactivo de respuesta es una tecnología que permite a los estudiantes, utilizando hardware específico o mediante software para dispositivos móviles, contestar a cuestiones planteadas por el profesorado durante las clases, o de forma autónoma, y ofrece además una realimentación inmediata acerca de cuáles han sido las respuestas proporcionadas. No es una tecnología reciente, dado que hay experimentos documentados en la década de 1960 [8], pero es cierto que hoy en día son más fáciles de usar, más completos y sólo se necesita un teléfono inteligente convencional para responder, por lo que se han popularizado en la última década (junto con el hecho de que la mayoría de ellos sean gratuitos, o al menos dispongan de una versión limitada sin coste).

En un uso típico de esta tecnología, a los estudiantes se les presentan preguntas de verdadero/falso o de opción múltiple que tienen que responder utilizando la aplicación o el hardware asociado. A continuación, el docente muestra las respuestas y proporciona retroalimentación, que puede implicar una discusión colectiva posterior. Varios estudios, por ejemplo [5, 14], resumen algunos usos comunes de este tipo de tecnologías documentados en la literatura: para lograr mayores niveles de interacción, para evaluar la preparación de los estudiantes, para averiguar más sobre las opiniones y preferencias de los estudiantes, para la evaluación formativa, para hacer que las clases sean divertidas, o para provocar debates.

2.1. Principales herramientas utilizadas

Existen muchas plataformas, como Kahoot, Socrative, Wooclap, Quizizz, Quizlet o Poll Everywhere, por nombrar algunas. A continuación, describiremos brevemente las principales características de las tres primeras puesto que son las más utilizadas por los participantes del estudio que se está llevando a cabo.

Kahoot! es una plataforma basada en la nube que permite a los estudiantes utilizar cualquier dispositivo inalámbrico equipado con un navegador web para enviar sus respuestas, aunque tiene su propia aplicación con funcionalidad añadida. Kahoot! contiene varias características de diseño para añadir gamificación a las actividades de aprendizaje, como puntos, tablas de clasificación, límites de tiempo, efectos de sonido y música, apodos y estética de un programa de concursos. Es posible crear, entre otras opciones, un concurso, un debate o una encuesta a través de un proceso guiado paso a paso. Para ganar puntos en los concursos de Kahoot! los alumnos tienen que responder a las preguntas correcta y rápidamente dentro de un límite

de tiempo específico. Cuando se acaba el tiempo para responder a una pregunta, el instructor muestra una distribución de las respuestas en la pantalla. Al mismo tiempo, los alumnos reciben una respuesta individual en sus dispositivos. Kahoot! ha sido probado con fines educativos con resultados positivos, como muestra Wang en [21].

Socrative es otra herramienta ampliamente utilizada que comparte características similares [7]. Socrative también se basa en la nube y su misión principal es obtener retroalimentación de los estudiantes en forma de respuestas de opción múltiple, verdadero o falso, o de texto corto. Aunque existe un modo de carrera espacial, los estudios publicados no suelen centrarse en esta variante gamificada sino en su idoneidad para realizar evaluación formativa en tiempo real [10, 17]. Es posible controlar la forma en que el alumnado navega en el cuestionario (por ejemplo, avanzando y retrocediendo a su ritmo, o guiado por el profesor).

Wooclap es una herramienta muy similar a Socrative, tanto en lo funcional como en el aspecto visual, alejada de la estética de concurso de Kahoot!. Ha ido implantándose recientemente en algunas universidades a través de licencias corporativas, como es el caso de la Universidad de Murcia. Incluye gran parte de la versatilidad de Kahoot, como es la gran diversidad de tipos de preguntas a plantear, con hasta 10 variantes distintas, y la posibilidad de insertar las preguntas entre las diapositivas que forman parte de una presentación. Dispone de un modo competitivo, pero con menos elementos que Kahoot! al no basar la puntuación en el tiempo, rachas ni tener tan presente en todo momento la clasificación.

Todas las herramientas anteriormente mencionadas proporcionan mecanismos para lanzar cuestionarios en tiempo real, es decir, para ser respondidos en el aula, o cuestionarios de autoevaluación para ser resueltos de forma autónoma en un plazo más largo.

2.2. Algunas conclusiones obtenidas en trabajos previos

A lo largo de las últimas dos décadas se han llevado a cabo numerosos estudios, desde el punto de vista pedagógico, sobre las ventajas que estas herramientas ofrecen al proceso de enseñanza aprendizaje. Se resumen brevemente aquí algunos aspectos que están relacionados con los contenidos expuestos en próximas secciones.

Hay consenso en que dichos sistemas tienen efectos positivos en las experiencias de aprendizaje, según criterios cognitivos, actitudinales y conductuales [13]. Son herramientas adecuadas para introducir cambios de ritmo en las clases y permitir que los estudiantes sean receptivos a los nuevos conocimientos, lo que pro-

porciona una oportunidad de mejorar el ritmo de aprendizaje [18].

En [16], Oigara y Keengwe estudian la percepción del alumnado y muestran que los estudiantes se muestran satisfechos con el uso de estas herramientas, en particular para mejorar su participación e implicación en las clases. Como se verá, esta percepción coincide con la de los participantes en el estudio.

Por último, Wang et al. [22] aportan valiosas conclusiones sobre la percepción de las herramientas clásicas (por ejemplo, Socrative) frente a las basadas en juegos (por ejemplo, Kahoot!). Los estudiantes indicaron que ambos sistemas aumentaban la participación y el compromiso. Las clásicas se consideran más adecuadas para evaluar tareas de aprendizaje más complejas. Como veremos, esta percepción también se alinea con las ideas que se presentarán en este trabajo en relación con algunas estrategias específicas.

3. Contextualización del estudio

Las estrategias presentadas en este trabajo se están analizando como parte de un estudio longitudinal, cuyos resultados se publicarán en futuros trabajos, que indaga acerca de las ventajas pedagógicas que implica el uso de diversas estrategias metodológicas en conjunción con los sistemas interactivos de respuesta. Sin embargo, este trabajo se apoya exclusivamente sobre un subconjunto reducido de los datos arrojados por el estudio, asociados a un instrumento de medida transversal correspondiente a la primera fase.

3.1. Grupo de estudio

Si bien en el estudio participan estudiantes y profesorado de diversas titulaciones, en este artículo centraremos la atención exclusivamente en el grupo de docentes, dado que su redacción está basada en su percepción acerca de estas herramientas y en el potencial de las distintas estrategias expuestas. Hasta ahora han participado en el estudio 42 docentes de diversas titulaciones universitarias, todas de la Universidad de Murcia. Estos profesores contestaron a un test inicial para valorar el uso de este tipo de plataformas en su docencia habitual. Se trata de profesorado de diversas titulaciones, como Informática, Veterinaria, Económicas, Filología, Educación, Biología, Derecho o ADE. Un grupo de ellos ha recibido formación en el uso de sistemas interactivos de respuesta por parte del Centro de Formación de la Universidad de Murcia, mientras que el otro no. No obstante, los resultados del estudio incluidos en este trabajo son previos a la formación recibida.

3.2. Procedimiento

Al profesorado se le facilita por correo electrónico el enlace a un formulario que contiene principalmente 25 ítems de escala Likert, con valoraciones de 1 a 5, donde 1 se corresponde con *'Muy en desacuerdo'* y 5 con *'Muy de acuerdo'*. Además de estos ítems, hay otras preguntas iniciales orientadas a conocer también el tamaño de los grupos de estudiantes con los que se trabaja o las herramientas que habitualmente utilizan. Adicionalmente, se les pregunta si las emplean para clases presenciales y/o para clases en línea.

En el Cuadro 1 se muestran los enunciados del subconjunto de ítems que se utilizarán en este trabajo.

Además de estos elementos, y con el fin de contextualizar de forma más precisa, cabe indicar que un 55 % de docentes las emplearon con grupos de tamaño medio (entre 25 y 75 estudiantes), un 38 % con grupos pequeños (menos de 25 estudiantes) y sólo un 7 % con grupos grandes de más de 75 estudiantes. Respecto a la plataforma utilizada, hay un porcentaje de uso muy similar de Kahoot!, Socrative y Wooclap, aunque esta última herramienta se ha implantado ampliamente en el curso 20/21 debido a la adquisición de una licencia corporativa.

4. Estrategias con las que aplicar sistemas interactivos de respuesta

Entre las múltiples posibilidades que existen a la hora de plantear estrategias pedagógicas basadas en sistemas interactivos de respuesta, este trabajo se ciñe a cinco estrategias diferentes, escogidas por cubrir un abanico amplio de ventajas pedagógicas y por estar adaptadas a partir de trabajos de investigación publicados:

1. Repaso a través de concurso
2. Preguntas intercaladas y a ciegas
3. A partir de entregas
4. Discusiones en grupo
5. Autoevaluación

Conviene aclarar que este artículo adopta normalmente el punto de vista de que las estrategias se utilizan como parte del proceso de evaluación formativa del alumnado, y no tanto sumativa, aunque realmente muchos aspectos serían fácilmente trasladables a ambos escenarios. Este enfoque está fundamentado también en la valoración del profesorado, a través de sus respuestas a las afirmaciones 3 y 4 del Cuadro 1, en las que se muestran más favorables al enfoque formativo (3,91) frente al sumativo (2,34).

Antes de entrar a detallar las estrategias, conviene resaltar que el uso predominante de estas herramientas

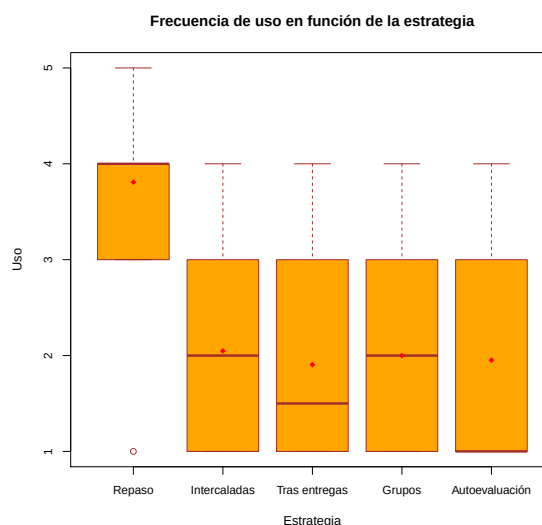


Figura 1: Frecuencia de uso de cada estrategia

entre la muestra de docentes del estudio es el primero de ellos, para repasar a través de concursos. La Figura 1 es un gráfico de cajas y bigotes que representa las respuestas proporcionadas por el profesorado a la pregunta de con qué frecuencia han utilizado las distintas estrategias cuando emplean los sistemas interactivos de respuesta, donde la escala Likert ofrecía las valoraciones: 1 - Nunca; 2 - Casi nunca; 3 - A veces; 4 - Habitualmente; 5- Siempre. Como algunos intrépidos profesores del estudio nos adelantan, aventurándose por terrenos menos explorados, veremos en las próximas subsecciones que hay vida más allá de los concursos.

4.1. Repaso a través de concurso

Esta estrategia es la más habitual, en opinión de los autores debido principalmente a dos razones. La primera de ellas es que es la adaptación más inmediata de métodos más tradicionales de evaluación basados en controles o exámenes tipo test que se realizan al final de bloques de contenidos. La segunda razón podría encontrarse en la forma en la que muchos sistemas de respuesta interactiva se publicitan, destacando el aspecto gamificado y competitivo de sus plataformas, con puntuaciones, tablas de clasificación, etc. Estos dos argumentos hacen que los docentes encuentren sencilla la incorporación de este tipo de herramientas a su docencia, tal y como muestra el ítem 6 del Cuadro 1 con una valoración media de 4,11.

Existen dos variantes en función de cómo se presentan las preguntas al alumnado. Por un lado está la posibilidad de que el ritmo de aparición lo marque el docente, lo cual le permite tener control sobre la progresión y sobre el tiempo dedicado a cada cuestión. Por otro lado, existe la posibilidad de que el ritmo lo mar-

Enunciado	Media (Desv)
1 Me parece importante dedicar tiempo a comentar las soluciones	4,49 (0,59)
2 El alumnado se muestra más participativo en clase	4,11 (0,65)
3 Los utilizo principalmente para evaluación formativa (no cuentan para la nota)	3,91 (1,13)
4 Los utilizo principalmente para evaluación sumativa (cuentan para la nota)	2,34 (1,38)
5 El tiempo necesario para prepararlos y revisar los resultados me parece asumible	3,83 (0,70)
6 Su uso me ha parecido sencillo	4,11 (0,85)

Cuadro 1: Valoraciones acerca del uso de sistemas interactivos de respuesta por parte de los docentes del estudio

que el estudiante, quien irá obteniendo nuevas preguntas conforme vaya respondiendo a las anteriores. Estas dos variantes hacen que el proceso esté sincronizado o no entre todos los participantes.

Algunas herramientas, como Kahoot o Quizziz, potencian significativamente la dinámica de gamificación [21] y están especialmente diseñadas para emplear esta estrategia. El carácter dinámico, en ocasiones espectacular, puede jugar a veces en su contra dado que puede minimizar la atención sobre las razones por las que las respuestas son correctas o incorrectas, con el fin de que la actividad sea dinámica.

4.2. Preguntas intercaladas y a ciegas

Sin embargo, no es necesario concluir un bloque de contenidos para lanzar pregunta. Estas se pueden ir intercalando con las explicaciones [2], incluso de manera completamente integrada en el caso de algunas plataformas que nos permiten secuenciar de manera unificada la progresión de diapositivas y preguntas. Interrumpir el tiempo de la clase con preguntas intercaladas puede parecer un inconveniente, teniendo en cuenta la omnipresente presión por completar el temario, pero da la oportunidad de identificar lagunas en el aprendizaje o de reforzar los nuevos conocimientos [1]. El docente puede entonces prestar más atención a la enseñanza de esos conceptos y dedicar menos tiempo a los temas que plantean menos problemas.

Es más, las preguntas que se van intercalando no tienen por qué hacer referencia a conceptos que se hayan explicado recientemente. Hay dos variantes de esta posibilidad que son especialmente interesantes. Por un lado, se puede preguntar acerca de conceptos que se consideran básicos, vistos en asignaturas previas de la titulación o incluso en ciclos educativos previos. Esto permitiría conocer cuál es la base sobre la cuál construir las explicaciones o actividades a desarrollar. Por otro lado, se pueden lanzar preguntas de respuesta meramente intuitiva, que podrían hacer referencia a tópicos o ideas que se desarrollarán a continuación. Esta posibilidad nos permite conocer ideas preconcebidas, el grado de familiarización previa con los conceptos, etc. Además, esta segunda variante ofrece la posibilidad

de igualar más los resultados (en términos de número de respuestas correctas) entre el conjunto del alumnado que las estrategias basadas en concurso, donde los alumnos más implicados suelen ser los que ocupen las primeras posiciones en la tabla de clasificación.

4.3. A partir de entregas

Esta tercera estrategia desplaza ligeramente el foco de interés sobre el que se construyen las preguntas, que pasan de estar basadas en el material o las explicaciones del docente a tomar en consideración los errores comunes y conceptos erróneos encontrados en una tarea o ejercicio recientemente resuelto por los estudiantes. A partir del análisis de dichos ejercicios, es posible diseñar preguntas que se utilizarán en el aula para ofrecer más explicaciones sobre conceptos o procedimientos que parecen plantear más dificultades de aprendizaje. Esta estrategia está principalmente basada en los principios de la enseñanza justo a tiempo (Just-in-Time-Teaching) [15].

Cuando se utiliza este enfoque, el alumnado tiene la sensación de que las clases están hechas a su medida y que está obteniendo una realimentación inmediata del trabajo realizado, el cual es tenido en consideración. Las explicaciones de las respuestas correctas e incorrectas toman especial importancia en este caso, puesto que se trata de afrontar los aspectos más problemáticos. Esta cuestión, la de dedicar tiempo a la explicación de las soluciones, es un aspecto considerado crucial también por los docentes del estudio, como indica la valoración de 4,49 perteneciente a la afirmación 1 del Cuadro 1.

4.4. Discusión en grupo

Hasta ahora, las estrategias anteriores habían planteado principalmente la perspectiva de que las respuestas fueran individuales, por parte de cada estudiante. Sin embargo, es posible llevar a cabo dinámicas de instrucción entre pares o de discusión en grupo utilizando también sistemas interactivos de respuesta. Es más, estas dinámicas se pueden combinar habitualmente con las tres estrategias comentadas anteriormente.

El propósito principal de esta estrategia es favorecer el aprendizaje colaborativo y, por tanto, la interacción entre los estudiantes. Entre las múltiples posibilidades posibles, se encuentra, por ejemplo, el enfoque de que los estudiantes respondan primero individualmente a las preguntas planteadas y luego vuelvan a responder a la misma pregunta tras discutir en grupo su razonamiento acerca de la respuesta previamente ofrecida [9, 11]. Se puede incluso proponer, si se estima conveniente, que los integrantes del grupo consensúen una única respuesta colectiva. Esta estrategia favorece el aprendizaje cooperativo y algunos estudios señalan que los alumnos obtienen más respuestas correctas tras la colaboración [19].

4.5. Autoevaluación

Finalmente, la última estrategia planteada en este trabajo se distingue de las demás por su asincronía, es decir, las preguntas no son formuladas al alumnado al mismo tiempo ni necesariamente durante las clases. Lo más habitual es que se habilite un periodo de tiempo durante el cual puedan responder a través de un enlace facilitado, lo cual puede ser útil para complementar el trabajo autónomo cuando se utiliza la metodología de clase invertida, para preparar pruebas de evaluación sumativa (controles, parciales, finales), etc. Habitualmente se habilita que los estudiantes puedan ver las respuestas correctas y las explicaciones oportunas tras contestar a las preguntas.

Esta estrategia puede ser especialmente útil para aligerar la carga de trabajo del docente a la hora de ofrecer realimentación inmediata, puesto que dicha realimentación está completamente automatizada. Si las explicaciones proporcionadas en cada pregunta son adecuadas, es posible compensar el tiempo necesario para elaborar dichas preguntas y respuestas con el tiempo que conllevaría resolver dudas individuales planteadas por el alumnado en ausencia de esta herramienta. Conviene también recalcar que los docentes participantes en el estudio consideran que el tiempo necesario para preparar cuestionarios y para analizar resultados es asumible, con una valoración de 3,83 en la afirmación 5 del Cuadro 1.

5. Guía de uso de las estrategias

Una vez desarrolladas las características principales de las estrategias propuestas, este apartado constituye una guía rápida de uso, organizada a partir de una serie de criterios comunes. La guía toma en consideración la experiencia personal en el uso de las estrategias, las impresiones transmitidas por los docentes que las emplean y recomendaciones que se han trabajado en los cursos de formación impartidos sobre el uso de

sistemas interactivos de respuesta. Contiene de forma esquemática las indicaciones generales que deben tenerse en consideración a la hora de implementar cada enfoque.

5.1. Repaso a través de concursos

- *Participación del alumnado*: Respuestas individuales.
- *Tipo de preguntas*: Predominan las preguntas de respuesta múltiple con una respuesta correcta.
- *Temporizador*: Hay tiempo límite, normalmente breve, para responder a las preguntas.
- *Entorno*: Mayoritariamente las preguntas se lanzan durante el desarrollo de las clases, sean estas presenciales o en línea.
- *Gamificación*: Se usan modos competitivos, con puntuación asociada a cada pregunta, y la tabla de puntuación del alumnado es un elemento importante en la dinámica de la actividad.
- *Finalidad*: Las preguntas se relacionan con el repaso de conceptos principales vistos en la materia.
- *Explicaciones*: Las explicaciones acerca de las razones por las que las respuestas son correctas, o no, son necesarias, pero han de ser ágiles y pueden verse afectadas por el clima de la dinámica.

5.2. Preguntas intercaladas y a ciegas

- *Participación del alumnado*: Respuestas individuales.
- *Tipo de preguntas*: Pueden utilizarse preguntas de sondeo (sin respuesta correcta) o de respuesta múltiple con respuesta correcta.
- *Temporizador*: Conviene asignar un tiempo límite para responder a las preguntas con el fin de mantener cierta dinámica.
- *Entorno*: Las preguntas se lanzan siempre durante el desarrollo de las clases, al ritmo de las explicaciones, sean estas presenciales o en línea.
- *Gamificación*: No es necesario que haya una puntuación asociada.
- *Finalidad*: Las preguntas se relacionan con conceptos vistos previamente o con conceptos que se trabajarán a continuación, con el fin de indagar sobre la intuición o ideas preconcebidas.
- *Explicaciones*: Las explicaciones acerca de las razones por las que las respuestas son correctas, o no, forman parte de la propia dinámica de la clase.

5.3. A partir de entregas

- *Participación del alumnado*: Respuestas individuales.

- *Tipo de preguntas:* Pueden utilizarse preguntas de sondeo (sin respuesta correcta) o de respuesta múltiple con respuesta correcta.
- *Temporizador:* Conviene asignar un tiempo límite para responder a las preguntas con el fin de mantener cierta dinámica.
- *Entorno:* Las preguntas se suelen lanzar al principio de una clase, una vez finalizado el plazo de entrega, para afianzar conceptos antes de profundizar en otros.
- *Gamificación:* No es conveniente que haya una puntuación asociada puesto que se está incidiendo sobre los fallos habituales.
- *Finalidad:* Las preguntas se relacionan con las actividades realizadas previamente por el alumnado, a partir de una muestra significativa de las entregas.
- *Explicaciones:* Las explicaciones deben ser detalladas puesto que están incidiendo en los problemas de aprendizaje.

5.4. Discusión en grupo

- *Participación del alumnado:* Respuestas individuales en primer lugar y grupales tras la discusión.
- *Tipo de preguntas:* Pueden utilizarse preguntas de sondeo (sin respuesta correcta) o de respuesta múltiple con respuesta correcta. Es conveniente que sean abiertas o que provoquen dispersión de resultados. Las preguntas se repiten para contrastar el resultado de la discusión.
- *Temporizador:* Conviene asignar un tiempo límite para responder a las preguntas con el fin de mantener cierta dinámica.
- *Entorno:* Las preguntas pueden estar basadas en contenidos previos, entregas o material aún no analizado.
- *Gamificación:* No es conveniente que haya una puntuación asociada, a no ser que sea para las preguntas respondidas tras la discusión.
- *Finalidad:* La finalidad es crear un entorno colaborativo de aprendizaje en el que pueda producirse instrucción entre pares.
- *Explicaciones:* Las explicaciones detalladas de las respuestas son parte del resultado y deben construirse sobre los argumentos que proporcione el alumnado.

5.5. Autoevaluación

- *Participación del alumnado:* Respuestas individuales. Posibilidad de participar múltiples veces.
- *Tipo de preguntas:* Predominan las preguntas de respuesta múltiple.

- *Temporizador:* Sin tiempo límite para responder a las preguntas.
- *Entorno:* Las preguntas están disponibles durante un periodo de tiempo, normalmente previo a alguna prueba o actividad de evaluación, para responderlas en línea.
- *Gamificación:* No es conveniente que haya una puntuación asociada, aunque podría habilitarse.
- *Finalidad:* Las preguntas se relacionan con el repaso de conceptos principales vistos en la materia.
- *Explicaciones:* Las explicaciones detalladas de las respuestas proporcionadas por los docentes son parte fundamental.

6. Conclusiones

Hoy en día existen múltiples plataformas de respuesta interactiva a disposición de los docentes de cualquier nivel educativo. La mayoría de ellas proporcionan funcionalidades muy similares, por lo que, al menos las más extendidas, están perfectamente habilitadas para ofrecernos un abanico de posibilidades metodológicas. Además, diversos estudios ilustran que la percepción del profesorado es que estas herramientas favorecen la participación activa, contribuyen parcialmente a mejorar los resultados del proceso de aprendizaje y su uso resulta sencillo en gran medida.

En este trabajo parte de la premisa de que las posibilidades de aprendizaje mediante estos sistemas dependen en gran medida al abanico de metodologías activas que seamos capaces de incorporar adecuadamente. En consecuencia, se han planteado diversas estrategias basadas en estos sistemas interactivos que aportan distintas ventajas pedagógicas. Se ha hecho especial énfasis en la necesidad de explorar otras vías más allá del uso predominante de estas herramientas como mecanismo de repaso basado en concurso.

Detrás las estrategias presentadas no solo se encuentra una forma distinta de emplear las herramientas, sino también enfoques pedagógicos diversos con los que afrontar el proceso de enseñanza-aprendizaje. La adopción de estos enfoques pedagógicos conlleva mucho trabajo y esfuerzo, y va más allá de simplemente incluir estas herramientas en la dinámica docente. Por tanto, su inclusión debe estar bien planificada y realizarse de forma progresiva. No sería conveniente llevar a cabo la implantación de una tacada de múltiples estrategias, sino de forma progresiva. Con el fin de facilitar este proceso, en la guía presentada se han aislado las características específicas de cada enfoque. Y puesto que hay un claro consenso de que una de las mejores formas de aprender es pasando a la acción (*learning by doing*), se invita a los lectores a ponerla en práctica y a adaptarla progresivamente a sus necesidades.

La pandemia de COVID-19 y la necesidad de tras-

lador una parte de la docencia al entorno en línea ha evidenciado la dificultad de mantener el contacto y la participación del alumnado en este nuevo contexto. Aprovechemos al máximo las posibilidades tecnológicas para que el proceso de enseñanza-aprendizaje no se resienta. Los sistemas interactivos de respuesta están preparados para adaptarse a distintas estrategias pedagógicas, es momento de aventurarse en ellas.

Referencias

- [1] Lindsay S Anderson, Alice F Healy, James A Kole y Lyle E Bourne Jr: *The clicker technique: Cultivating efficient teaching and successful learning*. Applied Cognitive Psychology, 27(2):222–234, 2013.
- [2] Pete Atherton: *More than just a quiz: how Kahoot! can help trainee teachers understand the learning process*. Teacher Education Advancement Network Journal, 10(2):29–39, 2018.
- [3] David Banks: *Audience Response Systems in Higher Education: Applications and Cases*. IGI Global, 2006.
- [4] John Biggs y Catherine Tang: *Teaching for quality learning at university: What the student does*. Open University Press, NY (EE.UU.), 2011.
- [5] Jane E Caldwell: *Clickers in the large classroom: Current research and best-practice tips*. CBE—Life Sciences Education, 6(1):9–20, 2007.
- [6] José I. Castillo-Manzano, Mercedes Castro-Nuño, Lourdes López-Valpuesta, María Teresa Sanz-Díaz y Rocío Yñiguez: *Measuring the effect of ARS on academic performance: A global meta-analysis*. Computers & Education, 96:109–121, 2016.
- [7] David Méndez Coca y Josip Sliško: *Software Socrative and smartphones as tools for implementation of basic processes of active physics learning in classroom: An initial feasibility study with prospective teachers*. European Journal of Physics Education, 4(2):17–24, 2017.
- [8] Emily Dill: *Do clickers improve library instruction? Lock in your answers now*. The Journal of Academic Librarianship, 34(6):527–529, 2008.
- [9] Adam P Fagen, Catherine H Crouch y Eric Mazur: *Peer instruction: Results from a range of classrooms*. The physics teacher, 40(4):206–209, 2002.
- [10] Fátima Faya Cerqueiro y Ana Martín-Macho Harrison: *Socrative in higher education: Game vs. other uses*. Multimodal Technologies and Interaction, 3(3):49, 2019.
- [11] Debra Filer: *Everyone's answering: Using technology to increase classroom participation*. Nursing education perspectives, 31(4):247–250, 2010.
- [12] Ariadna Fuertes, Miguel García, María Asunción Castaño, Emilia López, Mario Zacaes, Máximo Cobos, Ricardo Ferris y Francisco Grimaldo: *Uso de herramientas de respuesta de audiencia en la docencia presencial universitaria. Un primer contacto*. En Actas de las XXII JENUI, páginas 261–268. Universidad de Almería, 2016.
- [13] Nathaniel J Hunsu, Olusola Adesope y Dan James Bayly: *A meta-analysis of the effects of audience response systems (clicker-based technologies) on cognition and affect*. Computers & Education, 94:102–119, 2016.
- [14] Robin H Kay y Ann LeSage: *Examining the benefits and challenges of using audience response systems: A review of the literature*. Computers & Education, 53(3):819–827, 2009.
- [15] Gregor M Novak: *Just-in-time teaching*. New directions for teaching and learning, 2011(128):63–73, 2011.
- [16] James Oigara y Jared Keengwe: *Students' perceptions of clickers as an instructional tool to promote active learning*. Education and Information Technologies, 18(1):15–28, 2013.
- [17] Jenifer Santos, Luisa Parody, Manuel Ceballos, María C Alfaro y Luis A Trujillo-Cayado: *Effectiveness of mobile devices as audience response systems in the chemistry laboratory classroom*. Computer Applications in Engineering Education, 27(3):572–579, 2019.
- [18] Melissa A Schilling, Patricia Vidal, Robert E Ployhart y Alexandre Marangoni: *Learning by doing something else: Variation, relatedness, and the learning curve*. Management science, 49(1):39–56, 2003.
- [19] Michelle K Smith, William B Wood, Wendy K Adams, Carl Wieman, Jennifer K Knight, Nancy Guild y Tin Tin Su: *Why peer discussion improves student performance on in-class concept questions*. Science, 323(5910):122–124, 2009.
- [20] Javier Troya, Sergio Segura, José A. Parejo, Adela del Río-Ortega, Antonio Gámez-Díaz y Alfonso E. Márquez-Chamorro: *Invirtiendo las clases de laboratorio en Ingeniería Informática: Un enfoque ágil*. En Actas de las XXV JENUI, páginas 15–22. Universidad de Murcia, 2019.
- [21] Alf Inge Wang: *The wear out effect of a game-based student response system*. Computers & Education, 82:217–227, 2015.
- [22] Weiwen Wang, Sun Ran, Linda Huang y Valerie Swigart: *Student perceptions of classic and game-based online student response systems*. Nurse educator, 44(4):6–9, 2019.

Una experiencia de inclusión de los Objetivos de Desarrollo Sostenible en la asignatura Gestión de Proyectos

José Miguel Blanco, Rosa
Arruabarrena, Miren Bermejo,
Imanol Usandizaga

Departamento de Lenguajes y Sistemas
Informáticos
Universidad del País Vasco/Euskal Herriko
Unibertsitatea (UPV/EHU)
Donostia-San Sebastián
josemiguel.blanco@ehu.eus

Arturo Jaime

Departamento de Matemáticas y
Computación.
Universidad de La Rioja

Logroño
arturo.jaime@ur.es

Resumen

Muchas universidades han incluido en sus reflexiones estratégicas los objetivos asociados a la Agenda 2030 de la ONU. Algunas han explicitado su voluntad de fomentar e incentivar la inclusión de aspectos concretos de los Objetivos de Desarrollo Sostenible (ODS) en los planes de estudio. En el caso del Grado en Ingeniería en Informática, donde la gran mayoría de las asignaturas están orientadas al desarrollo de las competencias técnicas y transversales recogidas en sus respectivos planes de estudio, no resulta fácil encontrar un hueco para añadir nuevos objetivos o contenidos. Por esta razón se ha concebido, diseñado y desplegado una experiencia docente en la asignatura Gestión de Proyectos que acerca a los estudiantes del Grado en Ingeniería Informática de las Universidades del País Vasco y la Rioja a los ODS. En este trabajo se presenta el desarrollo de una asignatura en que los estudiantes tienen un contacto con los ODS mientras dirigen y desarrollan una secuencia de proyectos en que construirán un sitio web orientado a sensibilizar a sus iguales. La experiencia tiene una perspectiva plurianual, de manera que los estudiantes de sucesivas promociones vayan construyendo en base al trabajo realizado por sus predecesores en cursos anteriores.

Abstract

Many universities have included the goals associated with the UN 2030 Agenda in their strategic thinking. Some of them have explicitly stated their desire to promote and encourage the inclusion of specific aspects of the Sustainable Development Goals (SDGs) in their curricula. In the case of the Degree in

Computer Science, where the vast majority of the subjects are oriented to the development of technical and transversal competences included in their respective curricula, it is not easy to find space to add new objectives or contents. For this reason, a teaching experience has been conceived, designed and deployed in the Project Management subject that brings students of the Degree in Computer Science at the Universities of the Basque Country and La Rioja closer to the SDGs. This work presents the development of a subject in which students have a contact with the SDGs while they manage and develop a sequence of projects in which they will build a website aimed at raising awareness among their peers. The experience has a multi-year perspective, so that students of successive academic years build on the work carried out by their predecessors in previous courses.

Palabras clave

Objetivos de Desarrollo Sostenible, Agenda 2030, Gestión de proyectos, Responsabilidad ética y profesional.

1. Introducción

Treinta años después de que fuera establecido el concepto de desarrollo sostenible [6], la Organización de las Naciones Unidas (ONU) presentó, en 2015, los 17 Objetivos de Desarrollo Sostenible (ODS), en el marco de la adopción de la Agenda 2030 para el Desarrollo Sostenible. Se trata un *plan de acción a favor de las personas, el planeta y la prosperidad, que también tiene la intención de fortalecer la paz universal y el acceso a la justicia* [4].

Durante los últimos años, la Conferencia de Rectores de Universidades Españolas (CRUE) y muchas universidades españolas han desarrollado planes y acciones estratégicas relacionados con la Agenda 2030 aunque, por el momento, no se han traducido en acciones sistemáticas desplegadas sobre los planes de estudio en vigor [8].

Los planes de estudio de grado en Ingeniería Informática actuales fueron aprobados e implantados, en su gran mayoría, entre 2009 y 2012, y aunque las directrices del grado incorporaban explícitamente competencias relacionadas con la ética y la deontología profesional, no hacían una referencia específica a la *sostenibilidad*. De esta forma sólo algunas pocas universidades establecieron entre sus objetivos explícitos el desarrollo de competencias en torno al concepto de sostenibilidad [9, 12, 16].

Dado el marco temporal de la Agenda 2030, podemos encontrarnos con que los eventuales cambios que modificaran el desarrollo de la docencia del Grado en Ingeniería Informática afectarían, como pronto, a las promociones que terminen sus estudios después de 2025. Por otra parte, existe un sentido de urgencia social que, por ejemplo, se ha visto canalizada entre la juventud en iniciativas como la conocida popularmente con el nombre *Fridays For Future*¹.

En este contexto, dentro un proyecto de Innovación Docente, en colaboración entre las Universidades del País Vasco y la Rioja, se ha desarrollado una experiencia de introducción explícita de los ODS en la docencia de las asignaturas *Proyectos de Informática* y *Gestión de Proyectos*², de la Universidad de La Rioja y de la UPV/EHU, respectivamente. Esta experiencia se puso en marcha el curso académico 2019-20 y se ha repetido, introduciendo ajustes y mejoras, en el curso 2020-21.

En este trabajo, en primer lugar, se realiza una revisión de los antecedentes que enmarcan la experiencia docente presentada. A continuación, se describe cómo se establece la relación entre aspectos deontológicos y los ODS. Seguidamente se presenta las características de los proyectos en que se trabajan estos aspectos. Se continúa con la descripción de algunos resultados obtenidos, para terminar con las conclusiones que enfocan el trabajo a desarrollar en el futuro.

2. Antecedentes

En este apartado se realiza, en primer lugar, una somera revisión del contexto institucional universitario. A continuación, se hace un repaso de alguna de las iniciativas de referencia respecto a los ODS en el

ámbito de la docencia en Ingeniería Informática, relacionándolas con el tratamiento de los aspectos éticos en los estudios de Grado en Ingeniería Informática. Para terminar, se hace una revisión de algunos códigos éticos o deontológicos de asociaciones con una relación directa con las áreas de la Dirección de Proyectos y la Informática.

2.1. El contexto institucional universitario

En 2009, la CRUE constituyó la Comisión Sectorial Crue-Sostenibilidad³, con el objetivo de recopilar la experiencia de las universidades en materia de gestión ambiental y fomentar la cooperación para el intercambio de experiencias y la promoción de buenas prácticas. También se creó el Grupo de Trabajo de Sostenibilización Curricular⁴, que centra sus esfuerzos en la *sostenibilización curricular de los estudios universitarios*. Entre sus fines, impulsar estrategias para hacer llegar a los estudiantes formación que les permita *orientar sus futuras actuaciones profesionales y personales en aras a conseguir un mundo más sostenible, y avanzar con ello en el reto colectivo de los ODS de la Agenda 2030*. Fruto de la labor de este grupo se generó, aprobó y difundió por la CRUE el documento *Directrices para la introducción de la Sostenibilidad en el Currículum*. Tanto en las directrices, como en el anexo titulado *Orientaciones para la introducción de la Sostenibilidad en el Currículum*, se señalan una serie de líneas de trabajo, entre las que se incluyen la promoción de la sostenibilización curricular en el espacio de educación superior o el desarrollo de *programas de intervención que mejoren el diseño e implementación de la sostenibilización curricular*. Entre las líneas concretas de acción se incluye *recabar información sobre experiencias de sostenibilidad curricular exitosas, como base para la implementación de medidas de actuación concretas*.

Si nos acercamos a la realidad de cada universidad nos encontraremos con instanciaciones de esas orientaciones generales. Así, la Universidad del País Vasco/Euskal Herriko Unibertsitatea (UPV/EHU) ha encontrado en los ODS una *hoja de ruta para alinear el trabajo de la universidad con los grandes retos del planeta y de las personas, sin dejar a nadie atrás*. El conjunto de acciones en marcha se canaliza por medio del programa EHUagenda 2030⁵, que incluye la estrategia IKD. En este contexto, en 2019 propuso un nuevo *catálogo de competencias transversales*⁶ que tomaba como referencia el documento elaborado por la CRUE sobre la sostenibilidad en el

¹ https://en.wikipedia.org/wiki/School_strike_for_climate

² En lo sucesivo, por motivos de brevedad, utilizaremos únicamente el nombre *Gestión de Proyectos* para referirnos, indistintamente, a ambas asignaturas.

³ <https://www.crue.org/comision-sectorial/sostenibilidad/>

⁴ <https://www.crue.org/wp-content/uploads/2020/02/GT-Sostenibilizacion-Curricular-final.pdf>

⁵ <https://www.ehu.eus/es/web/iraunkortasuna/ehuagenda-2030>

⁶ <https://www.ehu.eus/es/web/enplegua/competencias-transversales>

currículo, y la Agenda 2030 para el Desarrollo Sostenible de la ONU. La Universidad de la Rioja, por su parte, asumió el deber de contribuir al Desarrollo Sostenible y tomó el compromiso⁷ de aplicar criterios de sostenibilidad a través de la *formación en la responsabilidad, proporcionando formación y sensibilización en programas de desarrollo medioambiental, éticos y sociales*.

2.2. Sostenibilidad y deontología en los estudios de Ingeniería Informática

Existen diversas e interesantes experiencias de introducción de los aspectos de sostenibilidad en los estudios de Ingeniería Informática con alcances y enfoques muy diversos [9, 12, 16]. En la Universidad española los aspectos relativos al desarrollo sostenible han recibido diferentes tratamientos, siendo el área de la sostenibilidad medioambiental la que ha recibido una mayor atención en el ámbito de las ingenierías. En el caso concreto de la Ingeniería Informática, una de las experiencias pioneras, probablemente la más acreditada y difundida, corresponde a la Facultad de Informática de Barcelona [7, 9, 18], con una larga trayectoria de implantación, revisión y difusión, en la que destacan aspectos relativos a los estudios de sostenibilidad en los Trabajos de Fin de Grado o la valoración del desarrollo competencial de los estudiantes durante el desarrollo de la carrera [17, 18]. En cualquier caso, con la reforma de los planes de estudio, desplegada a partir del año 2010, se generalizó, por la exigencia derivada de las recomendaciones⁸ para la propuesta de títulos oficiales en Ingeniería Informática (ver Cuadro 1), la introducción de contenidos que desarrollaban las competencias de índole social, ética y profesional.

Todos los grados (Real Decreto 1393/2007)
Capacidad de reunir e interpretar datos relevantes para emitir juicios que incluyan una <i>reflexión sobre temas relevantes de índole social, científica o ética</i>
Grados INGENIERÍA INFORMÁTICA (BOE-2009-1297)
Capacidad de analizar y valorar el impacto social y medioambiental de las soluciones técnicas, comprendiendo la <i>responsabilidad ética y profesional</i> de la actividad del Ingeniero Técnico en Informática.

Cuadro 1: Extracto de las recomendaciones

Los diseños de los planes de estudio implantados tras la reforma siguieron diferentes enfoques [12, 13]. En algunos casos se apostó por la introducción de nuevas asignaturas específicas, y en otros se distribuyó la responsabilidad entre diversas asignaturas, confiriéndoles un tratamiento similar a la recibida por

las competencias transversales. Pasados los años, algunos trabajos apuntan [18] a que la competencia de *aplicación de principios éticos relacionados con los valores de la sostenibilidad en los comportamientos personales y profesionales* quizás deba ser objeto de una mayor atención, incidiendo en la importancia que se da a la ética profesional en la titulación [18].

2.3. Códigos éticos y deontológicos en Ingeniería Informática

Aunque podríamos encontrar muchos más ejemplos, hay cuatro códigos especialmente significativos para el caso de la Ingeniería Informática y, específicamente, para la Dirección de Proyectos. Sus respectivas estructuras y dimensiones son muy diferentes, como lo son los tratamientos que reciben los aspectos de sostenibilidad. Las dos referencias internacionales más importantes son la ACM y el IEEE, ambas asociaciones disponen de sus correspondientes códigos [1, 10]. En el primer caso con la denominación *Código de Ética y Conducta Profesional*, en el segundo con el nombre *Código de Ética*. También dentro del marco internacional de la Dirección de Proyectos, la asociación más extendida a nivel mundial es el Project Management Institute (PMI), que tiene su propio *Código de ética y conducta profesional* [14]. Si nos situamos en el panorama español, una de las dos asociaciones profesionales de mayor solera es la ALI, cuyo Código Deontológico [2] es referencia inspiradora para el asociacionismo profesional que canalizan los Colegios profesionales de Ingeniería Informática.

En los códigos mencionados las referencias a la sostenibilidad tienen tratamientos variados. Si comparamos, por ejemplo, ACM e IEEE, vemos una declaración contundente de la primera asociación, donde se apela a la *obligación de los profesionales de la Informática, tanto individual como colectivamente, de utilizar sus habilidades en beneficio de la sociedad, de sus miembros y del entorno que les rodea* [1]. En el caso del IEEE, se hace una demanda a la *responsabilidad en la toma de decisiones de ingeniería consistentes con la seguridad, la salud y el bienestar del público, y revelar con prontitud los factores que puedan poner en peligro al público o al medio ambiente. Los profesionales de la computación deben evaluar si los resultados de sus esfuerzos respetarán la diversidad, si serán utilizados de manera socialmente responsable* [10].

El código de ética del PMI está estructurado alrededor de cuatro principios (responsabilidad, respeto, imparcialidad y honestidad) y dos niveles (obligatorio y deseable), se incluyen diversas referencias indirectas a la sostenibilidad. La más explícita es la recogida en el apartado de *estándares deseables en responsabilidad*, donde se insta a la toma de decisiones y el emprendimiento de *acciones basadas en los mejores*

⁷ <https://www.unirioja.es/servicios/os/politica.shtml>

⁸ <https://www.boe.es/boe/dias/2009/08/04/pdfs/BOE-A-2009-12977.pdf>

intereses de la sociedad, la seguridad pública y el medio ambiente [14].

Por último, el amplio código propuesto por la ALI, organizado en 17 apartados y 142 artículos a lo largo de catorce páginas, no contiene referencias explícitas al término *sostenibilidad*. Sin embargo, en su cuarto apartado, bajo el título *Sociedad*, se tocan en varios artículos diversos aspectos relacionados con los ODS, incidiendo en el respeto y la promoción de valores relativos a la protección del medio ambiente, la promoción de la igualdad o la defensa de los derechos humanos [2].

3. La asignatura Gestión de Proyectos

El estudiante aborda la asignatura cuando está terminando su tercer curso, trabajándose en ella competencias y recursos que le resultarán de utilidad en la última etapa de su formación y en su posterior ejercicio profesional. Aprovechando la circunstancia de que, en Dirección de Proyectos, la gestión del trabajo en grupo resulta nuclear, las actividades basadas en la colaboración sistemática entre los estudiantes se desarrollan durante trece de las quince semanas del curso. Partiendo de unas bases teóricas para la Dirección de Proyectos se aborda un cuerpo de conocimiento que agrupa conceptos, métodos, técnicas y herramientas asociadas a un buen ejercicio de la Dirección [15]. Se utilizan un conjunto de métodos docentes que promueven la implicación activa del alumnado, la colaboración sistemática y la reflexión crítica [3, 5, 11], creando situaciones en que los estudiantes aplican una parte de la teoría de dirección de proyectos a la resolución de problemas que tienen analogías directas con los que se presentan cuando se realizan trabajos que requieren la colaboración de personas cualificadas en Ingeniería Informática. En particular, al abordar estos proyectos deben identificar y resolver adecuadamente retos relacionados con la gestión de la propiedad intelectual, la privacidad de los usuarios o la accesibilidad de los productos para personas con diversidad funcional.

El objetivo es resolver problemas relativamente complejos, propuestos por terceros, que son los que, en última instancia, por medio de la satisfacción de los usuarios, determinarán la calidad de la solución. El desarrollo de los proyectos se convierte en el marco en que se trabajan aspectos éticos (por ejemplo, dentro del contexto del trabajo en grupo), legales (como puede ser el cumplimiento de la Ley Organizativa de Protección de Datos de Carácter Personal) y profesionales (por ejemplo, la calidad personal o la deontología).

El eje de desarrollo de la asignatura es el trabajo práctico, canalizado a través de una espiral de cuatro proyectos (ver Cuadro 2). En la cuarta iteración, el

proyecto alcanza una cierta entidad y es responsabilidad de un equipo de cinco/seis personas que trabaja conjuntamente durante seis semanas [5]. La experiencia se ha llevado a cabo homogéneamente en un total de 15 grupos de clase (alrededor de 150 estudiantes por curso) de dos universidades distintas durante los últimos cinco cursos académicos (2015-2020). En este periodo se han realizado con éxito cientos de proyectos, de los cuales más de 100 han sido abordados por equipos de entre cinco y seis estudiantes.

	Semanas	Tamaño equipo	Dedicación/estudiante [margen de horas]
P1	1	-	[3..6]
P2	3	3	[3..7]
P3	3	3	[9..12]
P4	6	5-6	[20..25]
Total	13	-	[35..50]

Cuadro 2: Distribución de proyectos por semanas, tamaño equipos y dedicaciones previstas

3.1. Aspectos éticos, legales y profesionales

Entre las competencias asignadas a las asignaturas de Gestión de Proyectos se incluyen algunas relacionadas con la práctica profesional que, aunque no de forma exclusiva en el plan de estudios, también se incluyen en el alcance a abordar. Por otra parte, en el área de la dirección de proyectos, además de las competencias denominadas *técnicas*, se incluyen las *contextuales* y las de *comportamiento*, entre las que tienen su espacio la ética, la legalidad y los conocimientos de normas y prácticas profesionales.

En el marco del ejercicio profesional, más allá del cumplimiento de la legalidad y las normativas, de por sí obligatorio, es habitual (dentro del asociacionismo profesional) la asunción de códigos éticos o deontológicos como condición para ser miembros de las diversas asociaciones. En algunos casos es preceptiva la colegiación para poder ejercer una determinada profesión, aunque no es así para la Ingeniería Informática. Por esta razón, únicamente aquellas personas que se adscriban a una determinada asociación que lo exija se verán obligadas por el *código de conducta* que tenga establecido.

Durante el desarrollo de la asignatura el estudiante toma contacto con los códigos éticos y deontológicos del PMI, IEEE, ACM y la ALI. Como se ha dicho, en todos los códigos deontológicos se hace un énfasis especial en el cumplimiento de las exigencias legales, lo que refuerza ante el alumnado la nada popular introducción de textos legales como material de trabajo. Este primer contacto con algunas de las leyes que limitan el ejercicio profesional en informática, consideramos interesante enmarcarlo en un horizonte

ético más amplio, sobre el que profesorado y alumnado podemos y debemos reflexionar. El preámbulo del código ético⁹ de la ACM señala en esa dirección: *Las acciones de los profesionales de la Informática cambian el mundo. Para actuar de forma responsable, deben reflexionar sobre los impactos amplios de su trabajo*, proponiendo como primer principio ético que *un profesional de la Informática debería contribuir a la sociedad y al bienestar humano, reconociendo que todas las personas son partes interesadas en la Informática*. Este principio se desarrolla en varios párrafos, el último de los cuales reza: *Además de un entorno social seguro, el bienestar humano requiere de un entorno natural seguro. Por lo tanto, los profesionales de la Informática deberían promover la sostenibilidad del medio ambiente tanto a nivel local como global*.

Aparte de un conocimiento teórico respecto a la existencia de estos recursos y su contexto de uso, se trata de que el alumnado reflexione sobre la aplicación práctica de esos códigos a cada actividad que desarrolla en la asignatura, tanto en su trabajo dentro de los equipos de proyecto, como en su relación con otros interesados (clientes, usuarios, proveedores, etc.). El marco de referencia en el que se insiste en mayor medida, y que forma parte de los contenidos específicamente evaluables, es el que establece el PMI [14]. En esa línea, se ha venido trabajando desde la implantación de las asignaturas en la concreción de los *estándares obligatorios* de comportamiento que establece el PMI respecto a los principios de *responsabilidad, respeto, imparcialidad y honestidad*. En el Cuadro 3 se presentan los estándares que son más referenciados durante el desarrollo de la asignatura.

4. Nuestra propuesta

Como se ha comentado, la asignatura Gestión de Proyectos se articula en torno al desarrollo, progresivo e incremental, de una secuencia de proyectos [5]. En ella se abordan, en cuatro etapas, problemas y soluciones cuya duración, alcance y dificultad de gestión va aumentando conforme el estudiante va incorporando conocimientos teóricos e instrumentales relacionados con las actividades de planificación, seguimiento y control de proyectos.

Los proyectos, además de la caracterización de los productos a desarrollar, tienen asociados objetivos específicos, tanto en lo relativo a las técnicas de gestión, como en lo que corresponde a las competencias de comportamiento, de forma que, sucesivamente, se recurre a las competencias trabajadas anteriormente, mientras se abordan nuevas dificultades y retos.

Principio	Estándar de actuación
Responsabilidad (2.3.1)	Nos informamos acerca de las políticas, normas, regulaciones y leyes que rigen nuestras actividades laborales, profesionales y de voluntariado y las defendemos.
Responsabilidad (2.3.2)	Informamos acerca de las conductas faltas de ética o ilegales a la dirección apropiada y, de ser necesario, a los afectados por dichas conductas
Respeto (3.3.2)	No ejercemos el poder de nuestra experiencia o posición para influir en las decisiones...
Respeto (3.3.3)	No actuamos de manera abusiva con los demás.
Respeto (3.3.4)	Respetamos los derechos de propiedad de los demás.
Imparcialidad (4.3.4)	No discriminamos a los demás por causa de género, raza, edad, religión, discapacidad, nacionalidad u orientación sexual.
Honestidad (5.3.1)	No nos involucramos ni toleramos el comportamiento que está diseñado para engañar a otros, incluyendo entre otros, hacer declaraciones engañosas o falsas...
Honestidad (5.3.2)	No nos comportamos de forma deshonesto con la intención de obtener un provecho personal o a expensas de otro.

Cuadro 3: Subconjunto de estándares de actuación del código ético del PMI [14]

El cambio introducido en el curso 2019-20 ha consistido en que los productos a desarrollar en el proyecto P4 sean sitios web plurilingües, accesibles por Internet, orientados a informar/concienciar/motivar a la acción a estudiantes y profesionales de Ingeniería Informática respecto a los ODS. Dentro de esta temática general, deben cubrir de forma específica, al menos, los siguientes cinco temas: Trabajo decente y crecimiento económico, Educación de calidad, Producción y consumo responsable, Acción por el clima, y Paz, justicia e instituciones sólidas.

La propuesta de cambio ha supuesto sustituir una orientación de los productos, también interesante, que tenía como objetivo la promoción de buenos hábitos relacionados con la seguridad informática para jóvenes, por los ODS. Se ha optado por un tema con menos connotaciones técnicas, pero que encaja y complementa muy bien con el tratamiento de los aspectos éticos. De esta forma, en los cuatro proyectos se interrelacionan aspectos relacionados con la deontología profesional y los ODS:

- P1. Ética en entorno profesional. Se presenta por medio del Código de Conducta del PMI, que aporta un marco de reflexión sobre aspectos re-

⁹ <https://www.acm.org/about-acm/code-of-ethics-in-spanish/>



Figura 1: Ejemplo de sitio web (curso 2019-20)

lacionados con el comportamiento ético en el ejercicio profesional actual en dirección de proyectos [14].

- P2. Además de la referencia a los aspectos éticos trabajados durante P1, se introducen aspectos legales, relacionados con la propiedad intelectual y el uso de licencias. También es necesario respetar la privacidad de las personas y de cualquier interesado en los productos desarrollados, que quedan expuestos en Internet. Los estudiantes deben elegir como tema del proyecto uno de los ODS entre el grupo de cuatro/cinco alternativas propuestas por el profesorado.
- P3. Se trabajan aspectos de Gestión de Calidad, utilizando como materiales los trabajos elaborados por diferentes equipos de P2 o por compañeros de cursos anteriores. Esto permite abarcar de forma conjunta tanto aspectos técnicos (cumplimiento de requisitos o el coste de la calidad, por ejemplo) como una revisión reflexiva de la forma en que han sido abordados por otros grupos el resto de los ODS propuestos y que el grupo no trabajó en P2.
- P4. El alcance del proyecto incluye la discusión sobre el respeto a la privacidad, asociada al cumplimiento de las restricciones establecidas por la Ley Orgánica de Protección de Datos de carácter personal (LOPD). A lo trabajado anteriormente, se suma un aspecto con connotaciones éticas, legales y normativas: el cumplimiento de las especificaciones de accesibilidad web de la iniciativa del W3C (WAI). El desarrollo de la web incluye dos secciones que vuelven a animar la búsqueda de contenidos y la selección proactiva de información útil en relación a los ODS. Se trata de un primer apartado de interacción, con preguntas relacionados con diversos aspectos, y de un segundo apartado de referencias donde se proponen materiales de consulta o profundización para los interesados.

Los materiales generados en los proyectos se distribuyen con licencias *Creative Commons*¹⁰ que permiten la realización de obras derivadas y la distribución abierta. Esto posibilita que aquellos materiales que lleguen a tener un valor se puedan utilizar más allá de las asignaturas implicadas, a la vez que sirven de base para el trabajo de promociones futuras. Mientras tanto, la comparación y la integración de soluciones permite a los estudiantes mejorar sus competencias en Gestión de Proyectos, a la vez que les acerca, por medio de entregables producidos por ellos mismos, a los ODS (ver Figura 1, captura de uno de los productos¹¹ de P4).

5. Resultados

La valoración de los resultados obtenidos en la implantación de la experiencia, en el curso 2019-20, en comparación con los de cursos anteriores (en que la temática nuclear de los proyectos era la *seguridad informática*) está absolutamente condicionada por el impacto que tuvo, en la segunda mitad del cuatrimestre la extensión de la pandemia del coronavirus. El cierre de los centros y el paso al modelo no presencial, que se produjo en torno a la octava semana del cuatrimestre conllevó un cambio completo en las circunstancias personales y académicas de todos los intervinientes. Por esta razón, las comparaciones con la satisfacción y/o el desempeño respecto a cursos anteriores no tendrían mayor valor. Aun así, el desarrollo de la experiencia culminó razonablemente, cumpliéndose con los objetivos asociados al último proyecto (P4), que se llevó a cabo completamente en modo no presencial. De esta forma, del desarrollo de la experiencia se pueden obtener una serie de indicadores cuyo estudio puede orientar el trabajo futuro.

En cuanto a los aspectos cualitativos, el alumnado reaccionó de forma positiva y aceptó con naturalidad

¹⁰ <https://creativecommons.org/>

¹¹ <https://sites.google.com/view/puis>

el viraje que conllevaba el abandono de una temática más técnica hacia una más social. A partir de una somera introducción motivacional por parte del profesorado, el alumnado acogió la propuesta con naturalidad e interés. En todo momento se asumió que era un tema de calado y sobre el que los futuros profesionales de la Ingeniería Informática tenían algo que decir. En muchos productos desarrollados se incluyeron reflexiones personales valiosas, incorporando en algunos casos aportaciones originales y con un alto grado de creatividad. Además, salvo algún caso excepcional, resultó ser su primer contacto con los ODS en el contexto de la docencia universitaria.

Entre los temas propuestos fue el de la *Producción y el consumo responsable* el escogido en mayor medida como objeto inicial de trabajo, seguido de cerca por *Acción por el clima*. El tercer lugar, el alumnado optó por el *Trabajo decente* y, en cuarto y último lugar, y en mucha menor medida, por el objetivo *Paz, justicia e instituciones sólidas*. Hay que tener en cuenta que durante el año 2019 se produjeron numerosas movilizaciones juveniles en relación al movimiento *Fridays For Future* que tenía como cabeza visible a la activista Greta Thunberg y que estaban focalizadas en la emergencia asociada al cambio climático. Esas movilizaciones habían incluido huelgas y manifestaciones en los campus universitarios de las universidades participantes en esta experiencia docente.

Cuando, para orientar las acciones del curso 2020-21, se encuestó al alumnado (120 estudiantes) sobre su percepción respecto a la relación que existía entre los ODS y la Ingeniería Informática, destacaron los ODS relativos a la Educación y al Trabajo decente (una media superior a 4, en una escala de 1-5), cerca de la Producción y el consumo responsable (3,9). Más atrás quedaron Acción por el clima (3,5) y Paz, justicia e instituciones sólidas (3,1).

A la vista de estos primeros resultados, la experiencia se pudo adaptar, en Septiembre del curso 2020-21, a la IUT de Bayona y del País Vasco¹². En este caso, el profesorado responsable de la asignatura optó por proponer el mismo tema (Acción por el clima) a todos los grupos, desarrollándose la experiencia con parecidos resultados a los obtenidos en cursos precedentes cuando la temática era la Seguridad Informática [19]. La percepción general del profesorado (10 profesores de las tres universidades implicadas) resultó muy positiva.

6. Conclusiones

En este trabajo se ha mostrado y analizado una experiencia de inclusión de la Agenda 2030 y los ODS en asignaturas de Gestión de Proyectos del Grado en

Ingeniería Informática de dos universidades trabajando de manera coordinada. Se ha aprovechado la relación existente con las competencias relativas al comportamiento ético y a la deontología profesional para dar un sentido integrado de cara al alumnado.

Las asignaturas tienen algunas características que las hacen especialmente interesantes para incluir actividades y metas relacionados con los ODS. Por un lado, el hecho de ser obligatorias y situarse en tercer curso permite garantizar un alcance común y un alto nivel de madurez académica del alumnado, además de aprovechar que el estudiante ya tiene un conocimiento amplio de las competencias técnicas informáticas proporcionadas por las asignaturas cursadas previamente. La cercanía a un futuro ejercicio profesional, y el hecho de que sea una asignatura donde la dirección de grupos humanos sea competencia nuclear, establece un contexto adecuado para una visión amplia sobre los efectos de las aplicaciones de la tecnología informática, y de la responsabilidad que tenemos quienes directa o indirectamente colaboramos en su desarrollo e implantación.

La propuesta planteada promueve la reflexión, la proactividad y la creatividad del alumnado en relación a los ODS. El hecho de que los diferentes proyectos se aborden siguiendo un modelo común de trabajo en grupo fomenta el intercambio de información y opiniones dentro de los propios grupos, en las clases y entre las universidades implicadas. Creemos que esta propuesta puede inspirar a otros centros y titulaciones para incorporar la Agenda 2030 sin necesidad de esperar a un cambio sistemático previo en los planes de estudio que, de llegar, no tendrá efecto para las promociones que terminen sus estudios antes de 2025.

La generación de recursos reutilizables [3] da un valor adicional al resultado conseguido hasta el momento y permitirá, en un futuro, una mejora colaborativa y una mayor difusión e impacto de las aportaciones de los estudiantes.

Por último, no hay que olvidar que los ODS forman parte de un proyecto social global, donde los profesores somos unos agentes más, que no tenemos el aval de una autoridad académica basada en una formación previa o una acreditación homologada. Compartir inquietudes, retos y trabajo en este ámbito con el alumnado no solo ayuda a los estudiantes a desarrollar sus competencias, también permite al profesorado reflexionar y aprender sobre un tema que nos compromete a todos.

Agradecimientos

Los autores quieren reconocer la valiosa colaboración del resto de docentes del Grupo DIMAROVE y de las asignaturas implicadas. Asimismo, agradecen

¹² La IUT forma parte de la Universidad de Pau y de los Países del Adour

la participación del alumnado de las Universidades de la UPV/EHU y la Rioja.

Este trabajo ha sido parcialmente subvencionado por el Vicerrectorado de Innovación, Responsabilidad social y Acción cultural de la UPV/EHU, a través del proyecto i320-19 (oDsIMAROVE+) dentro del Programa de Innovación Educativa, convocatoria 2020.

Referencias

- [1] ACM Code Task Force (2018). ACM Code of Ethics and Professional Conduct. Disponible en: <https://www.acm.org/about-acm/code-of-ethics-in-spanish> [accedido el 2 de febrero de 2021]
- [2] ALI Asociación de Ingenieros e Ingenieros Técnicos en Informática (2009) Código Deontológico
- [3] Rosa Arruabarrena, Ana Sánchez, José M. Blanco, José A. Vadillo e Imanol Usandizaga (2019) Integration of good practices of active methodologies with the reuse of student-generated content. *International Journal of Educational Technology in Higher Education (ETHE)* Vol. 16 (10) Pag.:1-20 DOI: 10.1186/s41239-019-0140-7
- [4] Asamblea General de las Naciones Unidas (2015) Transforming Our World: The 2030 Agenda for Sustainable Development, A/RES/70/1, disponible en: <https://www.refworld.org/docid/57b6e3e44.html> [accedido el 2 de febrero de 2021]
- [5] José Miguel Blanco, Imanol Usandizaga y Arturo Jaime: Gestión de Proyectos en el Grado en Ingeniería Informática: del PBL a la espiral de proyectos (2014). *ReVisión*, vol. 7, número 3, septiembre de 2014.
- [6] Gro Harlem Brundtland (1987) Report of the world commission on environment and development: our common future (Brundtland report). World Commission on Environment and Development, Oxford: Oxford University Press; New York.
- [7] Joan Climent, Jose Cabré, Fermín Sánchez-Carracedo, David López, Carme Martín, Eva Vidal y Ramón Bragós (2020). De la teoría a la práctica: 1011(2) años después de la integración de la sostenibilidad en el Grado en Ingeniería Informática de la FIB. En actas de las *XXVI JENUI*, Vol. 5, Valencia, pp. 133-140.
- [8] Crue Universidades Españolas (2019) COP25 Manifiesto de Crue Universidades Españolas por un planeta más sostenible <https://www.crue.org/documentos-e-informes/> [accedido el 2 de Febrero de 2021]
- [9] Jordi García, Fermín Sánchez, David López, Eva Vidal, José Cabré, Helena García y Marc Aliet (2014). De la teoría a la práctica: cinco años después de la integración de la competencia genérica de sostenibilidad en el Grado en Ingeniería Informática. En Actas de las *XX JENUI*, Oviedo, pp. 253-260.
- [10] IEEE Board of Directors (2020) IEEE Code of ethics (last revision). Disponible en: <https://www.ieee.org/about/corporate/governance/p7-8.html> [accedido el 2 de febrero de 2021]
- [11] Arturo Jaime, José Miguel Blanco, César Domínguez, Ana Sánchez, Jónathan Heras e Imanol Usandizaga (2016). Spiral and project-based learning with peer assessment in a computer science project management course. *Journal of Science Education and Technology*, 25(3), pp.. 439-449.
- [12] Rafael Miñano Rubio (2017). Integración de competencias de responsabilidad social, sostenibilidad y ética profesional en los grados de ingeniería informática. En actas de las *XXIII JENUI*, Vol.2, Cáceres, pp. 63-70.
- [13] Rafael Miñano, Gonzalo Génova, Sara Román y Eloy Portillo (2018). Reflexión sobre el papel de las asignaturas relativas a aspectos éticos, sociales, legales y profesionales en los grados de ingenierías informáticas. En Actas de las *XXIV JENUI*, Vol. 3, Barcelona, pp. 271-278.
- [14] Project Management Institute (2006). Código de ética y conducta profesional del PMI.
- [15] Project Management Institute (2017). A guide to the Project Management Body of Knowledge, Sexta Edición.
- [16] Fermín Sánchez-Carracedo y David López (2020). Innovation in Engineering Education for Sustainable Development—Introduction to a Special Issue. *Sustainability* 12, no. 19: 8132. <https://doi.org/10.3390/su12198132>
- [17] Fermín Sánchez-Carracedo, David López, Carme Martín, Eva Vidal, Jose Cabré y Joan Climent (2020) The Sustainability Matrix: A Tool for Integrating and Assessing Sustainability in the Bachelor and Master Theses of Engineering Degrees. *Sustainability* 12, no. 14: 5755. <https://doi.org/10.3390/su12145755>
- [18] Fermín Sánchez Carracedo, Ferran Sabate, Karina Gibert (2019). ¿Aprenden sostenibilidad las alumnas de la FIB? En actas de las *XXV JENUI*, Vol.4, Murcia, pp. 63-70.
- [19] Imanol Usandizaga, José Ángel Vadillo, Rosa Arruabarrena, Marta Toribio, Philippe Lopistéguy, Chakib Alami, y Christophe Marquesuzaa (2018). Una experiencia pedagógica en un entorno universitario internacional para formación en Gestión de Proyectos. En Actas de las *XXIV JENUI*, Vol. 3, Barcelona, pp. 319-326.

Validación de la escala de actitud ante la programación

Montse Garcia-Famoso¹

Jorge-Manuel Dueñas²

M.Àngels Moncusí¹

¹Departament d'Enginyeria Informàtica i Matemàtiques;

²Departament de Psicologia

Universitat Rovira i Virgili

43007 Tarragona

montse.garcia@urv.cat

jorgemanuel.duenas@urv.cat

angels.moncusi@urv.cat

Resumen

Existen muchos estudios que evalúan la dificultad que suponen los cursos de programación para los alumnos, en todos los niveles. Nosotros hemos observado que, en nuestra Universidad, el fracaso en la asignatura de Fundamentos de programación suele ser mayor en aquellos alumnos de las ingenierías menos relacionadas con la informática o las telecomunicaciones. Diversos estudios establecen una correlación entre el rendimiento de los alumnos y su percepción personal de competencia en programación. Para poder actuar de manera adecuada, y tratar de prevenir el fracaso, es imprescindible poder hacer un buen diagnóstico previo a los alumnos. Existen diversas encuestas para valorar la actitud de los alumnos ante la programación, sin embargo, la mayoría de estas encuestas, o no se encuentran disponibles o se trata de encuestas en inglés. En cualquier caso, antes de poder utilizar cualquiera de ellas, es necesario validarlas para nuestro contexto cultural. En este trabajo presentamos la adaptación y validación del instrumento *Computer Programming Attitude Scale* (CPAS) para estudiantes universitarios. Tras realizar el análisis factorial exploratorio podemos concluir que presenta unas propiedades psicométricas adecuadas para ser utilizado en la población universitaria española.

Abstract

There are many studies that evaluate the difficulty of computer programming courses for students, at all levels. We have observed that, in our university, dropout in the introductory computer programming subject tends to be greater in those students less related to computing or telecommunications engineering. Several studies establish a correlation between students learning performance and their personal perception of competence in programming. In order to be able to act appropriately and try to prevent failure, it is essential to make a good prior diagnosis of the students. There

are several surveys to assess the attitude of students to programming, however, most of these surveys are either not available or are surveys in English. In any case, before being able to use any of them, it is necessary to validate them for our cultural context. In this paper we present the adaptation and validation of the *Computer Programming Attitude Scale* (CPAS) instrument for university students. After performing the exploratory factor analysis, we can conclude that it presents adequate psychometric properties to be used in the Spanish university population.

Palabras clave

Escala de actitud, CPAS, programación, educación universitaria, validez, fiabilidad, auto-percepción, motivación.

1. Introducción

Los cursos de programación suelen resultar difíciles para los estudiantes debido a que requieren un alto nivel de abstracción así como de capacidad en la resolución de problemas [18]. Son difíciles, sobre todo, para alumnos de primer curso, lo que en ocasiones provoca un alto grado de suspensos y de abandonos. En [11] se presenta una revisión de las distintas problemáticas con las que se encuentran los estudiantes en los cursos de introducción a la programación, así como de las distintas estrategias pedagógicas que se han intentado para solucionarlas.

Algunos estudios se han centrado en intentar determinar aquellos conceptos de programación que pueden resultar más difíciles de comprender por parte de los estudiantes [22, 30, 34]. Otros autores introducen metodologías activas y centradas en el estudiante, [1, 29] como por ejemplo, técnicas de programación ágil, el SCRUM, aprendizaje basado en proyectos o el uso de la clase invertida.

Otros autores [24] proponen usar soluciones tecnológicamente más atractivas como el Logo y el Scratch, que complementan con una plataforma propia, el "Algorithmi", que usan de apoyo al aprendizaje de la algorítmica minimizando así, los errores en los algoritmos y corrigiéndolos de manera automatizada.

En otra serie de estudios, observan que el problema está en la dificultad que tienen los estudiantes para obtener una retroalimentación rápida y efectiva, lo que conlleva a una pérdida en la motivación, [29] propone la evaluación por pares, sin embargo, [7] considera la dificultad de obtener una retroalimentación correcta en este tipo de evaluación y proponen usar en las correcciones "alumnos" simulados. De forma similar, [4] utiliza la plataforma *On-line Judge* para dar una evaluación inmediata de los ejercicios propuestos y así reforzar la confianza de los alumnos en sus habilidades como programadores.

Hay autores que observan que los estudiantes tienden a experimentar miedo y falta de motivación para aprender a programar, y proponen diversas soluciones como por ejemplo, un asistente inteligente que responda a las preguntas sobre programación de los alumnos de forma inmediata [19], o el uso de una plataforma de juego para que les permita aprender mientras se divierten [6]. Por otro lado, observan que hay una tasa de abandono en los cursos de programación muy elevada y proponen mejorar la actitud de los estudiantes hacia el aprendizaje promoviendo el uso de libros electrónicos de auto-aprendizaje personalizados según la experiencia del estudiante [16].

Desde nuestro punto de vista, resulta imprescindible realizar un análisis previo que pueda ofrecernos una visión particular de nuestros alumnos para poder prever las medidas a aplicar en el curso. Por lo tanto, resulta imprescindible disponer de herramientas validadas para realizar dicho análisis [21], tal y como presentamos en la sección 2. El resto del artículo se organiza de la siguiente manera: En la sección 3 realizamos una breve revisión de la utilización de encuestas en el contexto de la programación. En la sección 4 explicamos la metodología llevada a cabo para validar el cuestionario. En la sección 5 mostramos los resultados obtenidos y en la sección 6 presentamos las conclusiones. Finalmente, en la sección 7 se indican futuras líneas de trabajo. En el anexo A incluimos la versión final del instrumento.

2. Motivación

De igual manera que sucede en otras universidades, el curso de introducción a la programación presentan altos índices de fracaso. En nuestro caso concreto, se trata de una asignatura, de primer curso, común a múltiples grados de ingeniería: desde ingeniería eléctrica hasta ingeniería informática. Esto hace que haya una

gran diversidad en cuanto a la formación previa, intereses personales y motivación de los alumnos.

Gomes, Santos y Mendes [12], demuestran que existe una gran correlación entre el rendimiento de los alumnos y la percepción personal de su competencia en programación. En nuestro contexto particular, con una alta heterogeneidad de alumnos, la actitud ante la programación de los alumnos puede ser un elemento decisivo en los resultados de los mismos, por lo que resulta imprescindible realizar un buen diagnóstico previo sobre dicha percepción. Dicho diagnóstico puede ayudarnos a tomar medidas más apropiadas, precisas y concretas para reducir el fracaso.

La escala de actitud ante la programación (*Computer Programming Attitude Scale* - CPAS) [3] se ha demostrado como un instrumento válido y de confianza para evaluar la actitud de los alumnos ante la programación, sin embargo, dicha escala se ha desarrollado en un contexto cultural diferente del español, lo que supone que antes de poder utilizarla hay que adaptarla, y por lo tanto, resulta necesario volver a evaluar su validez y fiabilidad en el nuevo contexto [21]. En este artículo se presenta la adaptación de la CPAS y su validación para poder utilizarla dentro del contexto universitario español.

3. Revisión de la literatura

Hemos llevado a cabo una búsqueda de artículos relacionado con las escalas y cuestionarios utilizados en el campo de la programación. El objetivo principal de la búsqueda no ha sido realizar una revisión exhaustiva de la literatura, si no tener una visión inicial tanto de las escalas propuestas, como de su utilización en el campo de la programación.

La búsqueda se ha realizado únicamente en la base de datos Scopus, ya que incluye las principales bases de datos, entre otras, IEEE, ACM, Elsevier, Springer. Hemos utilizando cualquier combinación de los siguientes términos en el título, en el resumen o en las palabras clave: ("*programming course*?") OR ("*computer programming*") OR ("*computer educatio*") AND ("*self-efficacy*") OR ("*attitude*") AND ("*scale*") AND ("*learning*").

El resultado ha sido de un total de 38 artículos, de los cuales se han descartado 23 entradas, porque no se ajustaban al objetivo de la búsqueda o porque no hemos podido acceder a ellos ya que estaban fuera de nuestro alcance.

Algunos de los artículos revisados diseñan sus propias herramientas de evaluación a partir de encuestas *ad hoc* o entrevistas [8, 13, 32].

Otra parte de los artículos revisados utilizan diversos instrumentos validados para evaluar diversos aspectos relacionados con la programación:

- Evaluación de la actitud o de la percepción de la eficacia respecto a la programación, la programación de móviles o su aprendizaje [2, 5, 9, 10, 14, 17, 27, 35, 36], los ordenadores [26].
- Evaluación de la actitud hacia los entornos de aprendizaje personalizados [28].
- Evaluación de la percepción de los estudiantes respecto a los laboratorios de informática [26].
- Escalas para evaluar la ansiedad respecto a la informática [17, 33].
- Evaluación de los estilos de aprendizaje [27, 35].

Uno de los artículos [15] presenta la validación de la adaptación cultural a Turquía de un cuestionario previo. Adapta la escala de auto-eficacia hacia las tecnologías en línea (*Online Technologies Self-Efficacy Scale - OTSES*), desarrollada originalmente por M. Miltiadou (2003).

Finalmente, Ö. Korkmaz, R. Çakir y M.Y. Özden [20] desarrollan y validan una escala para evaluar el nivel de pensamiento computacional (CTS - *Computational thinking scales*). Se trata de una encuesta de 29 ítems organizados en 5 factores: creatividad, pensamiento algorítmico, cooperatividad, pensamiento crítico y resolución de problemas.

Después de hacer una primera revisión de la literatura, se ha podido constatar que, existen diversos cuestionarios validados en otros países, pero no se ha encontrado ninguno validado para utilizar en el contexto universitario español.

4. Método

4.1. Participantes

Dado que el objetivo principal de la investigación consiste en la adaptación del cuestionario *Computer Programming Attitude Scale* (CPAS) [3], se recogieron datos de los estudiantes de la Universidad Rovira i Virgili. La muestra consta de 370 estudiantes universitarios (73,02 % hombres, 26,89 % mujeres, 0,5 % prefirió no decirlo), de los cuales, el 85,14 % son alumnos de primero (Fundamentos de programación) y el 14,86 % son de segundo (Programación), de diversos grados de ingeniería según la distribución mostrada en la Figura 1. El rango de edad de los sujetos está comprendido entre los 17 y los 48 años, con una media de edad de 19,66 años (D.T. = 3,30).

4.2. Instrumentos

Se utilizó el *Computer Programing Attitude Scale* (CPAS) [3] para estudiantes universitarios que consta de 18 ítem en escala *Likert*, con cinco alternativas de respuesta: (1) Completamente en desacuerdo, (5)

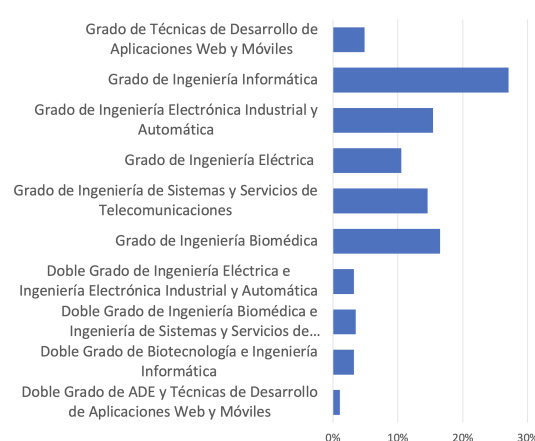


Figura 1: Distribución de estudiantes encuestados según el grado cursado.

Completamente de acuerdo. Seis de los ítems corresponden al factor afecto, seis al factor cognición y los seis restantes al factor comportamiento. El instrumento mide las actitudes hacia la programación en estudiantes universitarios.

Utilizamos un cuestionario *ad hoc* para evaluar la actitud hacia la programación, mediante tres ítems: 1) En general, tengo creencias positivas hacia la programación; 2) En general, tengo sensaciones agradables hacia la programación; 3) Habitualmente me siento cómodo/a realizando actividades relacionadas con la programación. Para responder se utilizó una escala *Likert* con cinco alternativas de respuesta: (1) Completamente en desacuerdo, (5) Completamente de acuerdo. Además, se preguntó el sexo, la edad, la educación previa al ingreso universitario, y el curso actual.

4.3. Procedimientos

Los cuestionarios se administraron colectivamente a través del espacio de las asignaturas en la plataforma de docencia virtual de la Universidad. Se garantizó el derecho al anonimato de los alumnos y la confidencialidad de sus resultados particulares. Se proporcionó el tiempo necesario para contestar los cuestionarios, sin límite de tiempo, con autorización voluntaria por parte de los estudiantes. Se les proporcionaron instrucciones sobre la manera adecuada de responder.

La traducción del instrumento original del inglés al castellano se realizó en cuatro fases. Primero, se realizó la traducción mediante el procedimiento de *back-translation* descrito por [25], un traductor bilingüe que domina el español y el inglés tradujo el CPAS al español. A continuación, un hablante nativo de inglés y traductor profesional, que no estaba familiarizado con la versión original del CPAS, volvió a traducir la versión en español al inglés, posteriormente otro traductor profesional evaluó si los ítems en español eran fieles al

significado de los ítems originales. Segundo, dos investigadores, un experto en evaluación psicológica y otro experto en programación revisaron la idoneidad, claridad, y expresiones culturales de los ítems. Tercero, se realizó una prueba de jueces. Específicamente, se seleccionó a seis investigadores externos, tres expertos en evaluación psicológica y tres en programación. Cuarto, se realizó una prueba piloto con un grupo de 35 estudiantes, a los cuales se les preguntó sobre la claridad de los ítems.

4.4. Análisis de datos

Respecto al análisis de los datos, se llevó a cabo un análisis factorial exploratorio (AFE) para determinar el número más apropiado de factores subyacentes a los datos e identificar cualquier ítem que funcione mal y que haya que eliminar, debido a que nuestros datos mostraron una estructura unifactorial, no fue necesario realizar un análisis factorial confirmatorio. El AFE se realizó utilizando FACTOR 10.9.02 [23]. Para los análisis de escala se utilizó SPSS 27.

5. Resultados

Respecto a la traducción del cuestionario, la evaluación realizada por jueces, concluyó en su idoneidad para el contexto cultural de nuestros alumnos. Así mismo, la prueba piloto realizada con los estudiantes, validó la claridad de los ítems. En cuanto al análisis de los datos, se realizaron pruebas previas con la finalidad de determinar si los datos eran aptos para realizar un análisis factorial. Concretamente, se comprobó la medida de adecuación maestra Kaiser-Meyer-Olkin ($KMO=0.89$), por lo que concluimos que la matriz era adecuada para el análisis factorial. También comprobamos el test de esfericidad de Bartlett que fue estadísticamente significativo ($X^2 = 2069.0$ (153); $p < 0.001$).

En el cuadro 1, se pueden ver los resultados del AFE, los datos mostraron una estructura unifactorial y parsimoniosa, el modelo propuesto se extrae en base al análisis paralelo, que aconseja extraer sólo un factor [31], de la misma manera el análisis MIREAL (0,21) sugiere que es unidimensional. Basándonos en los análisis realizados se opta por extraer un único factor, denominado "actitud hacia la programación". Se eliminó el ítem número dos debido a que tenía una saturación muy baja en el factor. Además, se han tenido que eliminar aquellos ítems que estaban muy correlacionados y/o que compartían mucha varianza residual. Específicamente se eliminaron los ítems: 5, 10, 11, 12, 16 y 17. Esto ya se esperaba porque las raíces de los ítems correspondientes estaban redactadas de manera muy similar o con un contenido similar. Como se puede ver en el Cuadro 1, todas las saturaciones son superiores a

0,3 y todos los índices de ajuste son adecuados, incluidos la raíz de la desviación cuadrática media ($RDCM = 0,06$). Las medidas comparativas de ajuste en relación con el modelo nulo fueron adecuadas. De esta manera, los datos sugieren claramente que la solución propuesta es sostenible.

Respecto al índice de fiabilidad calculado para el instrumento, el alpha de Cronbach indica que la escala tiene una muy buena consistencia interna ($\alpha = .97$). Así mismo, obtuvimos resultados satisfactorios al evaluar la validez del constructo de la versión española del *Computer Programming Attitude Scale*. Específicamente, obtuvimos correlaciones positivas con los tres ítems del cuestionario *ad hoc*. Ítem 1 "En general, tengo creencias positivas hacia la programación" ($r = .21$, $p < .01$). Ítem 2 "En general, tengo sensaciones agradables hacia la programación" ($r = .27$, $p < .01$). Ítem 3 "Habitualmente me siento cómodo/a realizando actividades relacionadas con la programación" ($r = .25$, $p < .01$).

6. Discusión y conclusiones

Los resultados del presente estudio sugieren que la versión española del *Computer Programming Attitude Scale* (SV-CPAS) tiene unas buenas propiedades psicométricas. Por lo tanto, nos provee de una medida en español del constructo "actitud hacia la programación" que se describe como las diferentes actitudes de los estudiantes universitarios hacia la programación, que pueden presentarse en forma de afecto, cognición y comportamiento. El objetivo principal del presente estudio fue la adaptación y validación del cuestionario *Computer Programming Attitude Scale* (CPAS) en población universitaria española, es la primera vez que se presenta esta escala en español. Los resultados del análisis factorial exploratorio del CPAS-versión española muestran una estructura unifactorial, esto difiere de la versión original diseñada para estudiantes universitarios turcos (Cetin y Ozden, 2015), la cual presenta una estructura factorial de tres factores: afecto, cognición y comportamiento hacia la programación. Esta discrepancia se puede deber a razones culturales. Por ejemplo, se eliminó el ítem número dos "programar es una habilidad específica" porque presentó una saturación muy baja en el factor, en el contexto español este ítem no contribuye a analizar las actitudes hacia la programación, debido que una habilidad específica no describe ninguna posición hacia la programación. Por otra parte, fueron eliminados los ítems: 5, 10, 11, 12, 16 y 17 porque compartían mucha varianza residual, los ítems redundantes expresan la misma idea con una redacción mínimamente distinta, esto genera un problema cuando explican más varianza que la que es directamente explicada por el factor común.

Ítem	Factor 1
1. Creo que programar es frustrante.	-0,65
2. Programar es una habilidad específica.	-
3. Creo que programar es innecesario.	-0,48
4. La programación no tiene importancia en la vida diaria.	-0,40
5. Siempre me esfuerzo por escribir un programa mejor.	-
6. Programar es aburrido.	-0,79
7. La programación facilita la vida humana.	-0,31
8. Si encuentro un problema que no puedo resolver a corto plazo mientras programo, no me doy por vencido/a hasta que lo resuelvo.	-0,40
9. Me pongo nervioso/a mientras programo.	-0,55
10. Participo en proyectos de programación si tengo la oportunidad.	-
11. Las actividades relacionadas con la programación me ponen nervioso/a.	-
12. Busco información y recursos para ser un/a buen/a programador/a.	-
13. Me aburro mucho cuando empiezo a programar.	-0,77
14. La programación mejora mis habilidades para resolver problemas.	-0,63
15. Una vez que empiezo a trabajar en un programa, trato de terminarlo antes que nada.	-0,34
16. La sola idea de escribir código me pone nervioso/a.	-
17. Creo que muchos de los problemas que quedan sin resolver en nuestra época se resolverán gracias a los avances en programación.	-
18. Sigo los avances en programación.	-0,41
Fiabilidad	0,93

Cuadro 1: Matriz de carga obtenida en el análisis factorial exploratorio.

7. Trabajo futuro

Disponer de un instrumento validado nos permitirá realizar, al inicio de curso, un análisis detallado de la actitud de los alumnos ante la programación. Dicho estudio se puede analizar, de forma segmentada, teniendo en consideración aspectos como el género, estudios previos, grado que cursa el alumno, etc. A partir de las conclusiones extraídas, se pueden planificar diversas actuaciones para adaptar materiales, metodologías, tecnologías... con el objetivo de incrementar la motivación de estos alumnos hacia las asignaturas de programación.

Referencias

- [1] Isaiah T. Awidi y Mark Paynter. "The impact of a flipped classroom approach on student learning experience". En: *Computers and Education* 128. September 2018 (2019), págs. 269-283.
- [2] Recep Çakir y Sabri Serkan Tan. "Development of educational applications on the social network of facebook and its effects on students' academic achievement". En: *Kuram ve Uygulamada Eğitim Bilimleri* 17.5 (2017), págs. 1525-1546.
- [3] Ibrahim Cetin y M. Yasar Ozden. "Development of computer programming attitude scale for university students". En: *Computer Applications in Engineering Education* (2015).
- [4] Brenda Cheang, A. Kurnia, Andrew Lim y Wee-Chong Oon. "On automated grading of programming assignments in an academic institution". En: *Computers and Education* 41.2 (2003), págs. 121-131.
- [5] Emre Çoban, Özgen Korkmaz, R. Çakir y Feray Ugur Erdogmus. "Attitudes of IT teacher candidates towards computer programming and their self-efficacy and opinions regarding to block-based programming". En: *Education and Information Technologies* (2020).
- [6] António Coelho, Enrique Kato, João Xavier y R. Gonçalves. "Serious game for introductory programming". En: *International Conference on Serious Games Development and Applications*. Springer. 2011, págs. 61-71.
- [7] Alexandre De Andrade Barbosa y Evandro De Barros Costa. "Simulated learners in peers assessment for introductory programming courses". En: vol. 1432. 2015, págs. 55-64.
- [8] Ronald Erdei, John A. Springer y David M. Whittinghill. "An impact comparison of two instructional scaffolding strategies employed in

- our programming laboratories: Employment of a supplemental teaching assistant versus employment of the pair programming methodology". En: vol. 2017-October. 2017, págs. 1-6.
- [9] Lisa Facey-Shaw y Paul Golding. "Effects of peer tutoring and attitude on academic performance of first year introductory programming students". En: vol. 2005. 2005, S1E-1-S1E-6.
- [10] Deniz Mertkan Gezgin, Muge Adnan y Meltem Acar Guvendir. "Mobile learning according to students of Computer Engineering and Computer Education: A comparison of attitudes". En: *Turkish Online Journal of Distance Education* 19.1 (2018), págs. 4-17.
- [11] Anabela de Jesus Gomes, António José Mendes y Maria José Marcelino. "Computer Science Education Research: An Overview and Some Proposals". En: *Innovative Teaching Strategies and New Learning Paradigms in Computer Programming* (2015), págs. 1-29.
- [12] Anabela de Jesus Gomes, Álvaro Nuno Santos y António José Mendes. "A study on students' behaviours and attitudes towards learning to program". En: *Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE* (2012), págs. 132-137.
- [13] Jamie Gorson y Eleanor O'Rourke. "How do students talk about intelligence? An Investigation of Motivation, Self-efficacy, and Mindsets in Computer Science". En: 2019, págs. 21-29.
- [14] Halit Gunbatar Mustafa Serkan abd Karalar. "Gender differences in middle school students' attitudes and self-efficacy perceptions towards MBlock programming". En: *European Journal of Educational Research* 7.4 (2018), págs. 925-933.
- [15] Mehmet Bans Horzum y Öziem Çakir. "The validity and reliability study of the Turkish version of the online technologies self-efficacy scale". En: *Kuram ve Uygulamada Egitim Bilimleri* 9.3 (2009), págs. 1343-1356.
- [16] Yen-Ping Huang y Yueh-Min Huang. "Programming language learning supported by an accredited course strategy". En: 2013, págs. 327-329.
- [17] Akpan Iniobong, Patrick Uko y Theresa Ekanem. "Computer anxiety, computer self-efficacy and attitude towards internet among secondary school students in Akwa Ibom State, Nigeria". En: 2013, págs. 193-198.
- [18] Reyes Juárez-Ramírez, Christian X. Navarro, Verónica Tapia-Ibarra, Ricardo Macías-Olvera y César Guerra-García. "What is Programming? Putting all Together - A Set of Skills Required". En: *2018 6th International Conference in Software Engineering Research and Innovation (CONISOFT)*. 2018, págs. 11-20.
- [19] Mario Konecki, N. Kadoić y Rok Piltaver. "Intelligent assistant for helping students to learn programming". En: 2015, págs. 924-928.
- [20] Özgen Korkmaz, Recep Çakir y M. Yaşar Özden. "A validity and reliability study of the computational thinking scales (CTS)". En: *Computers in Human Behavior* 72 (2017), págs. 558-569.
- [21] Carmen Lacave, Ana Molina, M. Fernández y Miguel Redondo. "Análisis de la fiabilidad y validez de un cuestionario docente". En: *ReVisión* 9.1 (2016), pág. 2.
- [22] Essi Lahtinen, Kirsti Ala-Mutka y Hannu-Matti Järvinen. "A study of the difficulties of novice programmers". En: *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education* (2005), págs. 14-18.
- [23] Urbano Lorenzo-Seva y Pere J. Ferrando. "FACTOR: A computer program to fit the exploratory factor analysis model". eng. En: *Behavior research methods* 38.1 (2006), págs. 88-91.
- [24] Antonio Manso, Celio Goncalo Marques y Paulo Santos. "Algorithmi: Software system to support the learning of programming". En: 2018.
- [25] Jordi Miró. "Translation, validation, and adaptation of an instrument to assess the information-seeking style of coping with stress: the Spanish version of the Miller Behavioral Style Scale". En: *Personality and Individual Differences* 23.5 (1997), págs. 909-912.
- [26] Michael Newby y Darrell Fisher. "A Model of the Relationship between University Computer Laboratory Environment and Student Outcomes". En: *Learning Environments Research* 3.1 (2000), págs. 51-66.
- [27] Hacer Özyurt y Özcan Özyurt. "Analyzing the effects of adapted flipped classroom approach on computer programming success, attitude toward programming, and programming self-efficacy". En: *Computer Applications in Engineering Education* 26.6 (2018), págs. 2036-2046.

- [28] Muhittin Şahin y Tarık Kışla. “An analysis of university students’ attitudes towards personalized learning environments”. En: *Turkish Online Journal of Educational Technology* 15.1 (2016), págs. 1-10.
- [29] Sónia Rolland Sobral. “Two different experiments on teaching how to program with active learning methodologies: A critical analysis”. En: vol. 2020-June. 2020.
- [30] Errol Thompson, Lynn Hunt y Kinshuk Kinsluk. “Exploring learner conceptions of programming”. En: *Conferences in Research and Practice in Information Technology Series* 52 (2006), págs. 205-211.
- [31] Marieke E. Timmerman y Urbano Lorenzo-Seva. “Dimensionality assessment of ordered polytomous items with parallel analysis”. En: *Psychological Methods* 16.2 (2011). cited By 537, págs. 209-220.
- [32] Laura Toma y Jan Vahrenhold. “Self-efficacy, cognitive load, and emotional reactions in collaborative algorithms labs - A case study”. En: cited By 8. 2018, págs. 1-10.
- [33] Murat Tuncer, Yunus Doğan y Ramazan Tanaş. “Investigation of vocational high-school students’ computer anxiety”. En: *Turkish Online Journal of Educational Technology* 12.4 (2013), págs. 90-95.
- [34] R. Paul Wiegand, Anthony Bucci, Amruth N. Kumar, Jennifer L. Albert y Alessio Gaspar. “A data-driven analysis of informatively hard concepts in introductory programming”. En: *SIGCSE 2016 - Proceedings of the 47th ACM Technical Symposium on Computing Science Education* March (2016), págs. 370-375.
- [35] Mustafa Yağcı. “Web-mediated problem-based learning and computer programming: Effects of thinking style on academic achievement and attitude”. En: *Computer Applications in Engineering Education* 26.6 (2018), págs. 2012-2025.
- [36] Erman Yukselturk y Serhat Altıok. “An investigation of the effects of programming with Scratch on the preservice IT teachers’ self-efficacy perceptions and attitudes towards computer programming”. En: *British Journal of Educational Technology* 48.3 (2017), págs. 789-801.

A. Anexo

Escala de actitud ante la programación. A continuación se presentan un conjunto de frases en relación a la programación. Lee cuidadosamente cada una de ellas y marca la respuesta que mejor describe tu opinión, las alternativas de respuesta para cada afirmación son: “Completamente en desacuerdo”, “En desacuerdo”, “Neutral”, “De acuerdo” y “Completamente de acuerdo” (Cuadro 2). No hay respuestas correctas ni incorrectas. Ninguna de las respuestas o resultados de los test se tendrán en cuenta en la evaluación de la asignatura. Su fin será meramente el de la investigación educativa. Todos los datos obtenidos se codificarán de forma que no aparezca ningún nombre, ni ninguna otra información que permita la identificación de los participantes. Sólo las personas relacionadas con el proyecto tendrán acceso a los archivos. Estos procedimientos están sujetos a la Ley Orgánica 15/1999 del 13 de diciembre sobre protección de datos de carácter personal. La participación en este estudio es totalmente voluntaria, por lo que el sujeto puede retirar su permiso en cualquier momento. En este caso, tus datos serían retirados de la base de datos.

	Completamente en desacuerdo	En desacuerdo	Neutro	De acuerdo	Completamente de acuerdo
1. Creo que programar es frustrante.	☐	☐	☐	☐	☐
2. Creo que programar es innecesario.	☐	☐	☐	☐	☐
3. La programación no tiene importancia en la vida diaria.	☐	☐	☐	☐	☐
4. Programar es aburrido.	☐	☐	☐	☐	☐
5. La programación facilita la vida humana.	☐	☐	☐	☐	☐
6. Si encuentro un problema que no puedo resolver a corto plazo mientras programo, no me doy por vencido/a hasta que lo resuelvo.	☐	☐	☐	☐	☐
7. Me pongo nervioso/a mientras programo.	☐	☐	☐	☐	☐
8. Me aburro mucho cuando empiezo a programar.	☐	☐	☐	☐	☐
9. La programación mejora mis habilidades para resolver problemas.	☐	☐	☐	☐	☐
10. Una vez que empiezo a trabajar en un programa, trato de terminarlo antes que nada.	☐	☐	☐	☐	☐
11. Sigo los avances en programación.	☐	☐	☐	☐	☐

Cuadro 2: Escala de actitud ante la programación.

Adaptación de un modelo de Aprendizaje Basado en Proyectos: experiencias en tiempo de confinamiento

Eduard de Torres, Joan Navarro, Xavi Canaleta, Daniel Amo,
Jordi Malé y Xavi Solé

Departamento de Ingeniería – La Salle
Universitat Ramon Llull
Barcelona

{eduard.detorres, jnavarro, xavier.canaleta, daniel.amo, jordi.male, xavier.sole}
@salle.url.edu

Resumen

Durante el segundo semestre del curso 2019-20, la pandemia de la COVID-19 provocó el confinamiento de la sociedad, incluyendo a los estudiantes universitarios. Esta situación obligó a las titulaciones a adaptar rápidamente su modelo metodológico a un entorno virtual. La virtualización de las asignaturas podía implicar un conjunto de obstáculos adicionales para los estudiantes: mayor complejidad para gestionar el trabajar en equipo, dificultades para el seguimiento síncrono de las sesiones o problemas de adaptación al cambio de medio o el riesgo de aumentar de la carga de trabajo, entre otros.

Este trabajo expone la adaptación realizada por la asignatura de Diseño y Programación Orientados a Objetos en la cual el Aprendizaje Basado en Proyectos es su eje vertebrador. Los cambios aplicados fueron tanto tecnológicos, para facilitar la gestión de proyectos en línea y la comunicación síncrona y asíncrona con los estudiantes, como metodológicos, introduciendo estrategias de modelos educativos en línea, como el trabajo asíncrono con reuniones síncronas, el seguimiento con retroalimentación o la evaluación mediante entrevistas.

Los resultados académicos obtenidos por los estudiantes presentan una mejora que satisface a los docentes y las encuestas de satisfacción a los estudiantes reflejan una valoración positiva de los cambios aplicados.

Abstract

During the second semester of the 2019-20 academic year, the COVID-19 pandemic led to the confinement of society, including university students. This situation forced degree courses to quickly adapt their methodological model to a virtual environment. The virtualisation of subjects could entail a series of additional obstacles for students: greater complexity in managing

teamwork, difficulties in synchronous monitoring of sessions, problems in adapting to the change of medium or the risk of increasing the workload, among others.

This paper presents the adaptation made by the subject of Object Oriented Design and Programming in which Project-Based Learning is the backbone. The changes applied were both technological, to facilitate the management of online projects and synchronous and asynchronous communication with students, and methodological, introducing strategies of online educational models, such as asynchronous work with synchronous meetings, monitoring with feedback or evaluation through interviews.

The academic results obtained by the students show an improvement that satisfies the teachers, and the student satisfaction surveys reflect a positive assessment of the changes implemented.

Palabras clave

Virtualización, Aprendizaje Basado en Proyectos, educación asíncrona, herramientas de gestión de proyectos.

1. Introducción

La asignatura de Diseño y Programación Orientados a Objetos de La Salle - URL, es una asignatura anual de 6 ECTS de segundo curso del Grado en Ingeniería Informática. Se imparte a través de tres horas lectivas a la semana. Tiene como descriptores generales [8]:

1. Introducción a la Orientación a Objetos
2. Características de la Orientación a Objetos
3. Análisis Orientado a Objetos
4. Diseño Orientado a Objetos
5. Implementación en Lenguaje Java
6. Patrones de diseño (GRASP, MVC y GOF)

7. Estructuras de datos lineales
8. Bases de datos
9. Diagramas UML

El primer semestre de la asignatura presenta los conceptos relacionados con el paradigma de programación orientado a objetos, así como el lenguaje de programación Java. Durante el segundo semestre, la asignatura aplica una metodología de Aprendizaje Basado en Proyectos (ABP) [6, 16] donde los estudiantes deben trabajar en equipo para desarrollar un proyecto de *software* durante los meses de marzo, abril y mayo. Este proyecto pone en práctica los conocimientos adquiridos en el primer semestre y los conocimientos del segundo semestre que se imparten durante el mes de febrero.

El día 12 de marzo de 2020 se anunció el cierre de todos los centros educativos del estado, incluidos los universitarios [4]. Las indicaciones recibidas fueron en la línea de seguir con la actividad académica en formato virtual. Con un nulo margen de reacción y adaptación las clases lectivas se reanudaron el lunes 16 de marzo en todo el centro universitario [7].

Con el semestre ya en marcha desde inicios de mes de febrero, la adaptación del campus en general y de la asignatura en particular debía que ser ágil y efectivo. Este artículo se centra en explicar las adaptaciones que se realizaron en el contexto descrito y también presenta los resultados obtenidos, tanto académicos como de satisfacción, en este periodo de pandemia.

Dado este contexto, el equipo docente se planteó los siguientes objetivos:

1. Minimizar las interacciones síncronas entre estudiantes y profesores.
2. Introducir herramientas digitales para facilitar la gestión de proyectos y el trabajo en equipo.
3. Adaptar el sistema de evaluación para un entorno en remoto.
4. Incrementar el seguimiento del profesorado para reforzar el aprendizaje y la evaluación en remoto.

Ahora sabemos que el confinamiento y el consecuente cambio de contexto tuvo un gran impacto en el sistema educativo. Distintos estudios [12, 13] apuntan a problemas psicológicos, así como problemas de concentración y agotamiento académico [5, 11].

También hemos visto, durante los últimos meses, experiencias parecidas a esta con resultados similares [9].

2. Modelo inicial

Para poder entender las adaptaciones realizadas, se describe de forma somera la metodología docente de la asignatura durante el segundo semestre del curso.

El segundo semestre utiliza la metodología de Aprendizaje Basado en Proyectos (ABP) [6]. No se usa ABP de forma pura, ya que se aprovecha la realización

de un proyecto de programación para consolidar los conocimientos logrados durante el primer semestre. El equipo docente estuvo formado durante el curso 2019-2020 por tres docentes y dos monitores de proyecto.

2.1. Metodología de trabajo

Los estudiantes deben formar grupos de cinco personas [15] y elegir uno de los cuatro proyectos enunciados. Para cada grupo, se asigna un docente como mentor. Éste ejerce el rol de cliente del proyecto y tiene como función resolver dudas sobre los requerimientos del proyecto.

Las sesiones de clase presenciales están destinadas a que los estudiantes trabajen en el desarrollo del proyecto, coordinen tareas con el grupo, resuelvan dudas de diseño con su mentor o dudas técnicas con todos los miembros del equipo docente. En este sentido, la tarea del equipo docente es la de resolver dudas sobre el enunciado del proyecto o sobre problemas que los estudiantes se encuentran al implementar la solución.

La asignatura aplica conceptos de *SCRUM* [14], para ayudar a los estudiantes en la gestión del desarrollo del proyecto. Al iniciar el proyecto, el mentor facilita al grupo las historias de usuario ya definidas, así como la planificación del primer *sprint* [14]. El proyecto se divide en un total de cuatro *sprints*, donde cada uno tiene una duración de dos semanas lectivas [14].

Para el seguimiento del proyecto se utiliza un póster físico en el que se le añaden *post-its* para representar las historias de usuario. El uso del póster tiene como objetivo facilitar la interacción intra e inter equipo, así como generar discusión entre el equipo y el mentor.

Al finalizar cada *sprint*, se dedica una sesión para que todos los grupos expongan el estado actual de su proyecto. Durante estas presentaciones, los compañeros de otros grupos pueden preguntar sobre las decisiones tomadas y proponer posibles mejoras.

2.2. Sistema de evaluación

La asignatura presenta dos convocatorias por curso académico, la convocatoria ordinaria y la extraordinaria. Al tratarse de una asignatura anual, la nota de la asignatura se calcula a partir de las notas del primer y segundo semestre, ponderadas a partes iguales. La nota del segundo semestre se calcula a partir de la nota de conocimientos teóricos (30 %) y la nota de conocimientos prácticos (70 %). La nota de conocimientos teóricos incluye:

- Examen de conocimientos (Exam): Evalúa de forma individual a cada estudiante. Al final del semestre se realiza un examen escrito. Se debe aprobar para poder superar el semestre. Pondera un

70 % de la nota de conocimientos teóricos, siempre que el estudiante presente los ejercicios de evaluación continua. En caso contrario, pondera el 100 % de la nota de conocimientos teóricos.

- Evaluación continua (EV): Evalúa de forma individual a cada estudiante. Es opcional y tiene un objetivo formativo. Se realiza antes de empezar el proyecto en forma de tres ejercicios, y sirve para ayudar a consolidar conceptos básicos del segundo semestre. Pondera un 30 % de la nota de conocimientos teóricos, siempre que el estudiante presente los ejercicios de evaluación continua.

La nota de conocimientos prácticos incluye:

- Nota del proyecto (Proyecto): Evalúa de forma grupal a todos los estudiantes de un grupo. El equipo docente evalúa la calidad técnica del proyecto utilizando rúbricas de evaluación. Se evalúa que (1) la ejecución del proyecto cumpla con los requerimientos funcionales del enunciado, (2) que el diseño del *software* cumpla con los principios de la programación orientada a objetos y (3) que la documentación generada explique correctamente las decisiones tomadas en el transcurso del desarrollo del proyecto. Supone el 90 % de la nota de conocimientos prácticos.
- Asistencia a clase (Asistencia): Evalúa de forma individual a cada estudiante. Dado que las sesiones presenciales sirven para fomentar y coordinar el trabajo en equipo, es obligatoria la asistencia a clase. Representa un 10 % de la nota de conocimientos prácticos.
- Revisión por pares [3]: Evalúa de forma individual a cada estudiante. Cada estudiante de un grupo de trabajo evalúa en cada *sprint* al resto de compañeros de su grupo. El mentor tendrá en cuenta estas cuatro evaluaciones como retroalimentación, para gestionar problemas de trabajo en equipo y para aplicar factores de corrección a las notas individuales del proyecto de los miembros de un equipo. Estos factores de corrección pueden implicar el aumento o disminución de la nota del proyecto de un estudiante en un +0,5 o -0,5.

3. Adaptación metodológica

Dado que la adaptación de la asignatura se tuvo que plantear en tres días naturales, muchas de las decisiones tomadas se basaron en la experiencia propia del profesorado de la asignatura, la experiencia de los compañeros de la universidad o las sugerencias de los coordinadores académicos.

3.1. Adaptación de la universidad

Para adaptar las clases a la pandemia, la universidad decidió transformar las sesiones presenciales en sesiones virtuales [7]. Este cambio se hizo efectivo a partir del lunes 16 de marzo del 2020. El calendario académico y el horario de clases se mantuvo sin cambios.

Todas las asignaturas de la universidad disponen de un espacio dentro de Moodle, que es el Entorno Virtual de Aprendizaje (EVA) elegido por la universidad, donde poder compartir recursos, actividades o información con todos los estudiantes. La universidad integró en Moodle la herramienta de videoconferencias Blackboard Collaborate Ultra para realizar las sesiones virtuales.

3.2. Cambios metodológicos

Para la adaptación a un modelo de asignatura virtual, se decidió consultar con la metodología *Self Directed Based Learning* (SDBL) [2] que se aplica en los programas en línea de la universidad. SDBL es una variante del *Self Directed Paced Learning* (SDPL) definida en La Salle - URL, que ha validado su éxito [2] en su uso como metodología principal de los programas de postgrado que se imparten en línea.

Uno de los problemas que preocupaban al equipo docente durante la conceptualización del formato virtual, era el cansancio académico [5, 12] que podía producir en los estudiantes la asistencia obligatoria a unas clases virtuales. Dado que las tres horas semanales de sesiones lectivas eran de trabajo personal en equipos de proyecto, se decidió pasar a un modelo de trabajo asíncrono con encuentros síncronos [2]. Con esto en mente se decidió que la asistencia a las clases pasara a ser opcional. Las sesiones lectivas seguían sirviendo para la resolución de dudas tanto técnicas, con el equipo docente, como conceptuales, con el mentor.

Se aplicó un sistema de reuniones semanales de seguimiento de 30 minutos. En ellas, cada equipo se debía reunir con su mentor. Estas reuniones virtuales eran síncronas y su asistencia era obligatoria. Se planificaron dentro del horario lectivo de la asignatura y para cada grupo se calendarizó un día y hora concretos de la semana. Estudiantes y equipo docente se debía conectar a la sesión de Blackboard Collaborate Ultra que se configuraba para el trabajo en equipos. Se creaban salas dedicadas a cada uno de los grupos y el equipo docente que no estaba reunido con tareas de mentoría se encontraba en la sala principal. En el momento de realizar una mentoría, el mentor se conectaba a la sala dedicada al grupo en particular donde realizaba la reunión virtualmente. En caso que los estudiantes necesitaran de soporte para la resolución de dudas, uno de los miembros del grupo debía conectarse a la sala

principal y utilizar la herramienta digital de "levantar la mano". El equipo docente podía en todo momento ver cuantos estudiantes estaban en la sala principal con la "mano levantada." Incluso podía ver el orden en que estos habían pedido ayuda.

El objetivo de las reuniones de seguimiento era el de acompañar a los estudiantes con una atención personalizada para asegurar que el equipo seguía trabajando en el proyecto de forma constante. Las reuniones también tenían como objetivo detectar problemas dentro del grupo con la mayor rapidez posible para poder buscar soluciones a dichos problemas, como por ejemplo: dificultades de un estudiante para seguir el ritmo de trabajo, problemas de comunicación dentro del grupo o diferencias significativas de nivel de conocimiento entre miembros de un equipo. Para cada situación particular se propusieron soluciones a medida.

En este sentido, el mentor dejaba de ser una figura reactiva a quién los estudiantes podían consultar dudas; para convertirse en una figura activa dentro del ciclo de desarrollo del proyecto.

Dada la pérdida de interacción física docente-estudiante, al equipo docente le preocupó que algún estudiante pudiera perderse información relevante sobre la evolución de la asignatura. En cursos anteriores, mucha información de interés sobre el desarrollo de los proyectos se comunicaba de forma oral en clase. Esta información podía ser la respuesta a una pregunta interesante realizada por un estudiante, la resolución a una duda común sobre el enunciado de un proyecto o informaciones similares más allá de los comunicados oficiales de la asignatura.

Se decidió potenciar el uso del EVA y que toda comunicación de interés para todos los estudiantes debía quedar reflejada en los foros de la asignatura para asegurar la correcta impartición asíncrona de la asignatura.

Se decidió adaptar las presentaciones orales que se realizaban al finalizar un *sprint*. Se plantearon dos posibilidades: (1) convertir las presentaciones orales presenciales en grabaciones de vídeo para su visualización asíncrona, o (2) eliminar las presentaciones.

Finalmente se decidió eliminar las presentaciones para reducir la percepción de carga de trabajo de los estudiantes [5].

3.3. Cambios tecnológicos

La virtualización de la asignatura significaba virtualizar también todas las herramientas utilizadas para facilitar el desarrollo de *software* en equipo y la gestión del proyecto.

La planificación de la asignatura contemplaba el uso de un póster DIN-A0 en papel. Este póster estaba colgado en el aula donde se impartían las clases de la asignatura. En él los estudiantes empleaban *post-its* para representar el seguimiento de las historias de usuario en cada uno de los *sprints*.

Dada la virtualización de las sesiones, era necesario migrar esta herramienta a otra alternativa digital en línea. Por ello se decidió utilizar licencias educativas de Jira a disposición de la universidad para que los estudiantes pudieran gestionar el proyecto de forma remota. El docente-mentor del proyecto recibía el rol de *project manager* dentro de la herramienta con permisos de gestión, mientras los estudiantes recibían roles de desarrollador para poder interactuar con las historias de usuario, asignar responsables o modificar el estado de éstas.

Para facilitar la introducción de la herramienta a los estudiantes, los docentes gestionaron la creación de los proyectos en la plataforma Jira, así como de la definición de las historias de usuario en ella. Para formar a los estudiantes en la herramienta, se realizó un seminario síncrono en línea y se generó documentación en forma de manual de uso.

También se introdujo Git como herramienta de control de versiones para facilitar la integración del código dentro del equipo de desarrollo.

3.4. Cambios en el sistema de evaluación

Los cambios en el contexto, así como los cambios metodológicos, planteaban la necesidad en el equipo docente de revisar el sistema de evaluación.

Uno de los retos de la evaluación en un modelo en línea que preocupaba al equipo docente, eran los problemas relacionados con la verificación de la identidad de los estudiantes [1, 10]. Se creyó que un examen en línea presentaba dificultades para poder comprobar la identidad de los estudiantes.

Aunque actualmente existen herramientas de *proctoring* que ayudan a disuadir a los estudiantes de tener actitudes deshonestas con la evaluación de la asignatura [1], en el momento de la adaptación se decidió eliminar el examen final de la asignatura y optar por otros sistemas de evaluación que favorecieran poder asegurar la identidad del estudiante en los distintos escenarios que se podían imaginar en ese momento.

La eliminación del examen final de la asignatura, implicaba que se debía evaluar las competencias adquiridas por los estudiantes de un modo alternativo. Para ello se decidió introducir dos nuevas iniciativas: (1) la evaluación individual del estudiante por parte del mentor durante las reuniones de seguimiento del proyecto; y (2) la aplicación de un examen oral

en formato de entrevista en línea para validar los conocimientos adquiridos.

Una de las adaptaciones metodológicas propuestas consistía en que el mentor del proyecto asumía un rol más activo en el desarrollo del proyecto. Este acercamiento al grupo de estudiantes permitía que el mentor pudiera conocer con más detalle la evolución individual de cada estudiante dentro de su grupo de trabajo. De esta forma, se decidió que, durante las reuniones semanales de seguimiento, el mentor evaluaría el trabajo de los estudiantes en función de los conocimientos demostrados. Cada semana este debía puntuar del 0 al 3 a cada estudiante, donde:

- 0 = "No presentado"
- 1 = "NO ha adquirido los conocimientos"
- 2 = "Ha adquirido conocimientos, pero comete errores"
- 3 = "Ha adquirido los conocimientos"

Este seguimiento permitía evaluar de forma continua a los estudiantes viendo su evolución y desarrollo a lo largo del transcurso del proyecto. De esta forma se podía validar la adquisición de conocimientos por parte de cada estudiante.

Dado que el examen oral en línea propuesto, tenía como objetivo validar la adquisición de conocimientos de los estudiantes; se estimó redundante realizar dicho examen a aquellos estudiantes que ya hubieran demostrado sus conocimientos durante el seguimiento. Se decidió que todo estudiante que obtuviera una media $\geq 2,5$ sobre 3 sería liberado del examen oral.

Se definió un examen oral grupal en línea de 20 minutos de duración. En el examen oral se debía tener la cámara y micrófonos encendidos en todo momento para poder verificar la identidad del estudiante.

Una de las dudas que generaba este sistema en el equipo docente era la fiabilidad de dicha entrevista para evaluar a los estudiantes con una nota sobre 10 puntos, como se hubiera hecho en un examen escrito. El examen escrito de la asignatura que se realizaba según el modelo inicial, tenía una duración de 3 horas en las que los estudiantes debían resolver problemas de diseño de *software* a partir de una especificación de requerimientos. Convertir el examen escrito de 3 horas a un examen oral de 20 minutos no parecía equivalente según el equipo docente en el momento de la adaptación de la asignatura. Por este motivo, se decidió que el examen oral en formato de entrevista tendría únicamente dos posibles notas: superada o no superada.

4. Resultados

Para explicar los resultados de esta experiencia, se han analizado los de resultados académicos, los resul-

Curso	# estudiantes	% aprobados	% NPs	Media
19-20	188	77	9	6.53
18-19	160	54	27	6.36
17-18	132	63	28	6.69
16-17	129	60	34	7.14

Cuadro 1: Resultados académicos del segundo semestre desde el 2017 al 2020.

tados de las encuestas de satisfacción docente pasadas a los estudiantes y se han recogido datos aproximados sobre los aumentos y disminuciones de carga en el equipo docente. En un entorno tan complejo como es el educativo donde, además, ha habido un cambio prácticamente de paradigma con la pandemia, siempre corremos el riesgo de una simplificación en el análisis y que múltiples variables no contempladas puedan influir en los cambios observados. Aun así pensamos que el foco está puesto donde tocan y que, sin negar la influencia de factores externos a los analizados, creemos que los resultados muestran una correlación aceptable con los factores analizados.

4.1. Resultados académicos

En el cuadro 1 se muestra una comparativa de los resultados académicos del segundo semestre durante los últimos cuatro cursos. Como se puede observar entre los cursos 2016-17 y 2018-19, la asignatura se ha mantenido estable en cuanto a los porcentajes de aprobados, abandonos (NP) y nota media. Durante el curso 2019-2020, cuando se aplicaron las adaptaciones expuestas en este trabajo, destacamos (1) la leve mejora del porcentaje de aprobados en el curso 2019-2020, y (2) la disminución del abandono durante el mismo curso.

El equipo docente cree que la mejora del porcentaje de aprobados se puede explicar por el aumento del seguimiento realizado. En cursos anteriores, el docente-mentor de cada equipo presentaba un rol de *product owner* [14] con quién los estudiantes podían resolver sus dudas. Esta interacción implica que el estudiante debe tomar la iniciativa en cuanto a preguntar a su mentor. Un estudiante podría no ser consciente de sus errores y por ello no consultar con su mentor. Estos errores se acumulan y se detectan finalmente en la corrección final del proyecto. Por contra, cuando el mentor realiza el seguimiento activo del proyecto, éste puede detectar errores durante la evaluación formativa que se realiza. Estas revisiones ayudan a la formación del estudiante, quién ya no propaga los errores hasta la evaluación sumativa final.

Curso	% apro- bados examen ordina- ria	% apro- bados examen extra- ordina- ria	% apro- bados pro- yecto ordina- ria	% apro- bados pro- yecto extra- ordina- ria
19-20	-	-	47	30
18-19	48	9	26	34
17-18	48	11	32	44
16-17	59	6	20	42

Cuadro 2: Resultados académicos ampliados del segundo semestre desde el 2017 al 2019.

Otra explicación para la mejora del porcentaje de aprobados podría ser la modificación del sistema de evaluación. Al eliminar el examen escrito, la evaluación del semestre recae aún más en el proyecto realizado. En el cuadro 2 se pueden observar los resultados ampliados con los porcentajes de aprobados en las dos convocatorias de examen y de proyecto.

Se puede observar en el cuadro 2 que los estudiantes superan el examen de conocimientos mayoritariamente en la convocatoria ordinaria, mientras que los proyectos, se superan mayoritariamente en la convocatoria extraordinaria. En el cuadro 3 se presentan los resultados de estudiantes no-presentados (NP) en cada una de las evaluaciones. Todos los porcentajes están calculados a partir del total de estudiantes matriculados en la asignatura en cada curso.

Dada esta información, la lectura del equipo docente es que (1) la mayoría de los estudiantes entrega el proyecto en la convocatoria extraordinaria, (2) la mayoría de los estudiantes superan el examen en la convocatoria ordinaria, (3) pocos estudiantes se presentan a la convocatoria extraordinaria del examen y, en consecuencia, (4) los estudiantes que ya no se presentan al

Curso	% NP examen ordina- ria	% NP examen extra- ordina- ria	% NP pro- yecto ordina- ria	% NP pro- yecto extra- ordina- ria
19-20	-	-	48	9
18-19	24	18	63	23
17-18	33	25	61	23
16-17	19	16	71	34

Cuadro 3: Resultados de abandono ampliados del segundo semestre desde el 2017 al 2019.

examen o que lo suspenden, ya no presentan el proyecto en la convocatoria extraordinaria y aparecen como abandonos (NPs).

En el cuadro 3 se observa que, en el curso 2019-20, aunque casi la mitad de los estudiantes no presentan el proyecto en la convocatoria ordinaria, la gran mayoría de ellos terminan presentando el proyecto en convocatoria extraordinaria. El equipo docente atribuye estos datos a que (1) no hay un examen que los estudiantes puedan suspender, y consecuentemente suspender el semestre con indiferencia del resultado del proyecto, y (2) el seguimiento realizado por los mentores durante el desarrollo del proyecto favorece que los estudiantes terminen entregando dicho proyecto.

4.2. Resultados de las encuestas de satisfacción docente

Semestralmente la universidad realiza una encuesta de satisfacción docente en todas las asignaturas. En esta encuesta los estudiantes pueden responder de forma anónima a 6 preguntas. Todas las preguntas son iguales para cualquier asignatura de la universidad. Cada pregunta se evalúa de 1 a 5. Durante el segundo semestre del curso 2019-20 se añadió una pregunta adicional para que los estudiantes pudieran valorar la adaptación de la asignatura al formato virtual. Las preguntas de la encuesta son las siguientes:

1. ¿Cómo valoras globalmente la asignatura?
2. ¿Cómo valoras la organización y planificación?
3. ¿Cómo valoras los materiales usados?
4. ¿Cómo valoras la innovación y el grado de actualización?
5. ¿Cómo valoras la información proporcionada sobre la actividad docente?
6. ¿Cómo valoras la adecuación de la carga de trabajo?
7. ¿Cómo valoras la adaptación al formato virtual de este semestre?

En el cuadro 4 se muestran los resultados de las encuestas de satisfacción docente que realiza la univer-

Pregunta	2017-18	2018-19	2019-20
(1)	4.2	4.0	3.9
(2)	3.9	3.5	3.7
(3)	3.8	3.3	3.4
(4)	4.6	4.3	4.3
(5)	3.9	3.5	3.7
(6)	3.2	3.3	3.4
(7)	-	-	3.9

Cuadro 4: Resultados de las encuestas de satisfacción docente desde el 2018 al 2020.

sidad de forma semestral a todas las asignaturas. En éstas se aprecia que las valoraciones de los estudiantes sobre la asignatura no han variado significativamente respecto a cursos anteriores.

Destacar la nueva pregunta (7) que se ha añadido en las encuestas del segundo semestre del curso 2019-2020 donde se pregunta explícitamente a los estudiantes sobre su satisfacción con las adaptaciones realizadas en la asignatura. Se aprecia en estos resultados que los estudiantes han valorado de forma ligeramente positiva la adaptación al formato virtual.

4.3. Cambios en la carga docente

Tras finalizar el semestre, el equipo docente se reunió para valorar las adaptaciones realizadas. En este proceso de revisión, se obtuvieron datos aproximados sobre los aumentos y disminuciones de la carga de trabajo en distintos momentos del semestre. Se detectaron tres espacios temporales, (1) la preparación de los proyectos, (2) la ejecución de éstos y (3) su evaluación.

Actividad	Horas
Configuración de Jira	$+0.5 * (N/5/3)$
Resolución dudas (8 semanas)	+16
Preparación examen	-3
Realización examen	-3
Corrección examen	$-0.25 * (N/3)$
Preparación entrevista	+2
Realización entrevista	$+0.07 * (N*2/3)$
Total (N = 188)	+10.8

Cuadro 5: Incrementos y decrementos aproximados de la carga docente. N = número de estudiantes.

En el cuadro 5 se pueden observar las cifras aproximadas respecto al aumento y disminución de la carga de un docente. Mencionar que algunos elementos están divididos entre 3 para calcular la repartición entre los 3 docentes de la asignatura.

Como parte del proceso de preparación de los proyectos, el único cambio significativo fue el uso de Jira como herramienta digital de gestión de proyectos.

Durante la ejecución de los proyectos, aunque el día a día de las sesiones no varió significativamente, el equipo docente detectó un aumento en las horas dedicadas a la resolución de dudas fuera de las sesiones lectivas. Se ha estimado este aumento en 2 horas semanales por mentor (durante 8 semanas lectivas). El equipo docente considera que este aumento se puede haber dado por dos factores: (1) durante las sesiones sincrónicas, los mentores utilizaron gran parte del tiempo a la evaluación y gestión de los equipos, reduciendo la capacidad de atender a dudas técnicas y (2) la creencia de

que algunos estudiantes no aprovecharon las sesiones lectivas para resolver dudas, ya que la asistencia no era obligatoria. Hay que tener en cuenta que los horarios de algunas sesiones son a las ocho de la mañana.

Durante la fase de evaluación de los proyectos se añadió un examen oral grupal de 20 minutos pero se eliminó el examen escrito de 3 horas. La experiencia nos permite aproximar que la corrección de un examen se realiza en 15 minutos. En cada entrevista grupal, 2 docentes entrevistaron a 5 estudiantes en 20 minutos.

5. Conclusiones

Tras presentar el contexto inicial de la asignatura, describir los cambios realizados y analizar los resultados obtenidos, el equipo docente se muestra satisfecho con la adaptación realizada en la asignatura.

Los cambios hacia un modelo de trabajo asíncrono con reuniones sincrónicas ha dado buenos resultados. Una de las consecuencias del confinamiento de 2020 fue el cansancio académico y el consecuente aumento del abandono [5, 11]. Los resultados presentados muestran una disminución de este abandono, por lo que el equipo docente está satisfecho con las decisiones tomadas, e incluso ve los cambios aplicables a un modelo presencial.

El seguimiento semanal de los grupos de trabajo nos ha permitido mejorar ligeramente los resultados académicos de los estudiantes. En cursos anteriores, los estudiantes realizaban el proyecto con el soporte de un mentor, pero el mentor no estaba implicado directamente en el desarrollo del proyecto. En el momento en que el mentor se ha reunido periódicamente con cada grupo, se ha podido guiar mejor para mejorar la tasa de éxito final.

La satisfacción de los estudiantes según se han visto en las encuestas no varía mucho respecto a años anteriores. La nueva pregunta añadida a raíz del confinamiento, nos muestra que los estudiantes se muestran positivos ante los cambios aplicados a la asignatura.

Aunque el equipo docente percibió un aumento significativo de su carga docente; tras analizar los datos calculados de forma orientativa; este coste no se ve inasumible. Gran parte del aumento de la carga se concentró durante el desarrollo del proyecto, mientras que el periodo de evaluación final fue mucho menos costoso.

El éxito resultante de la aplicación de estas adaptaciones al diseño de la asignatura nos motiva a seguir utilizando este modelo para explorar las ventajas que puede suponer en una educación presencial o incluso híbrida. Los cambios aplicados a la asignatura se van a mantener durante el curso 2020-2021, para seguir analizando los resultados obtenidos y poder seguir mejorando el diseño de ésta.

Referencias

- [1] J. Adkins, Cindy Kenkel y Chilo Lim. «Derrents to online academic dishonesty». En: *The Journal of Learning in Higher Education* 1 (ene. de 2005), págs. 17-22.
- [2] Maria Alsina, Xavi Canaleta, Albert Cubeles y Ricardo Torres. «Self Directed Based Learning, una metodología de aprendizaje para programas no presenciales». En: ene. de 2019, págs. 325-330. DOI: 10.26754/CINAIC.2019.0070.
- [3] Carlos Blanco y Pablo Sánchez. «Aplicando evaluación por pares: análisis y comparativa de distintas técnicas». En: *Actas de las JENUI* (jul. de 2012). URL: <http://hdl.handle.net/2099/15087>.
- [4] Jefatura del Estado. *Real Decreto-ley 11/2020, de 31 de marzo, por el que se adoptan medidas urgentes complementarias en el ámbito social y económico para hacer frente al COVID-19*. Agencia Estatal Boletín Oficial del Estado, 2020, págs. 27885-27972. URL: <https://www.boe.es/eli/es/rdl/2020/03/31/11/dof/spa/pdf>.
- [5] Antonio Fernández-Castillo. «State-Anxiety and Academic Burnout Regarding University Access Selective Examinations in Spain During and After the COVID-19 Lockdown». En: *Frontiers in Psychology* 12 (ene. de 2021), pág. 621863. DOI: 10.3389/fpsyg.2021.621863.
- [6] María José García García, Juan José Escribano Otero y M^a Cruz Gaya López. «Experiencia de aplicación de ABP al Grado de Ingeniería Informática». En: Universidad de Oviedo. Escuela de Ingeniería Informática, jul. de 2014. ISBN: 978-84-697-0774-6. URL: <http://hdl.handle.net/2099/15460>.
- [7] La Salle - URL. *¡COMIENZA LA VIRTUALIZACIÓN DE LAS CLASES DE LA SALLE-URL!* Mar. de 2020. URL: <https://www.salleurl.edu/es/comienza-la-virtualizacion-de-las-clases-de-la-salle-url> (visitado 10-02-2021).
- [8] La Salle - URL. *Diseño y programación orientados a objetos*. 2020. URL: <https://www.salleurl.edu/es/diseño-y-programación-orientados-a-objetos-1> (visitado 11-02-2021).
- [9] Faraón Llorens Largo, Carlos Villagrà Arnedo, Francisco Gallego Durán y Rafael Molina Carmona. «CoVID-proof: cómo el aprendizaje basado en proyectos ha soportado el confinamiento». En: (2021). URL: <http://hdl.handle.net/10272/19338>.
- [10] Patricia Mcgee. «Supporting Academic Honesty in Online Courses». En: *Journal of Educators Online* 10 (ene. de 2013). DOI: 10.9743/JEO.2013.1.6.
- [11] Jorge Moreno-Fernandez, Julio Ochoa, Inmaculada Lopez-Aliaga, Maria Alferez, Manuel Gómez-Guzmán, Sagrario Lopez-Ortega y Javier Díaz-Castro. «Lockdown, Emotional Intelligence, Academic Engagement and Burnout in Pharmacy Students during the Quarantine». En: *Pharmacy* 8 (oct. de 2020), pág. 194. DOI: 10.3390/pharmacy8040194.
- [12] Paula Odriozola-González, Álvaro Planchuelo-Gómez, María Irurtia y Rodrigo Luis-García. «Psychological effects of the COVID-19 outbreak and lockdown among students and workers of a Spanish university». En: *Psychiatry Research* 290 (mayo de 2020), pág. 113108. DOI: 10.1016/j.psychres.2020.113108.
- [13] Naiara Ozamiz-Etxebarria, Nahia Idoiaga, Maria Santamaria y Maitane Picaza. «Psychological Symptoms During the Two Stages of Lockdown in Response to the COVID-19 Outbreak: An Investigation in a Sample of Citizens in Northern Spain». En: *Frontiers in Psychology* 11 (sep. de 2020), pág. 2116. DOI: 10.3389/fpsyg.2020.02116.
- [14] Dan Rawsthorne y Doug Shimp. *HANDBOOK SINGLE-TEAM SCRUM (STS)*. 11.^a ed. 2019. ISBN: 9781719047265.
- [15] Xavier Solé-Beteta, Joan Navarro, David Vernet, Agustin Zaballos, Ricardo Torres Kompen, David Fonseca y Alan Briones. «Automatic tutoring system to support cross-disciplinary training in Big Data». En: *The Journal of Supercomputing* 77 (feb. de 2021). DOI: 10.1007/s11227-020-03330-x.
- [16] Miguel Valero García y Javier García Zubia. «Cómo empezar fácil con PBL». En: *Jornadas de Enseñanza Universitaria de la Informática* (2011). Ed. por AENUI: Asociación de Enseñantes Universitarios de Informática, págs. 109-116. URL: <http://hdl.handle.net/11441/61478>.

Aprendizaje basado en proyectos en el ámbito de la Ingeniería de la Salud: desarrollo de sistemas para la rehabilitación y la ayuda a la discapacidad

L. Duran-Lopez, D. Gutierrez-Galan, E. Cerezuela-Escudero, A. Rios-Navarro
J. P. Dominguez-Morales

Dpto. Arquitectura y Tecnología de Computadores
Universidad de Sevilla
41012 Sevilla
jpdominguez@atc.us.es

Resumen

El reciente Grado en Ingeniería de la Salud se originó con el planteamiento de cubrir la carencia de conocimiento sobre la rama sanitaria de un Ingeniero Informático. Es en dicho grado donde, de forma optativa, el alumno cursa la asignatura Sistemas de Rehabilitación y Ayuda a la Discapacidad, que tiene un papel fundamental para el diseño de sistemas enfocados tanto a la rehabilitación como a proveer facilidades a pacientes con ciertas dificultades. En este artículo se describe la metodología de las prácticas de laboratorio para la docencia en dicha asignatura, la cual se basa en el concepto de aprendizaje basado en proyectos. Las distintas sesiones de laboratorio se reparten en dos bloques, diferenciando entre sesiones de introducción a ciertos contenidos y dispositivos hardware, y un gran bloque donde los alumnos desarrollan un proyecto de rehabilitación o ayuda a la discapacidad con los conocimientos adquiridos, que posteriormente documentan en formato artículo. Esta dinámica ha conseguido una gran acogida y ha incrementado la motivación de los alumnos, llevando a muchos a desarrollar un Trabajo Fin de Grado relacionado con la asignatura. La dinámica propuesta tiene un papel fundamental en futuros Ingenieros de la Salud tanto a nivel de aprendizaje como a nivel laboral.

Abstract

The recently created Degree in Health Engineering was originated from the idea of covering the need of knowledge related to Computer Engineering within the healthcare industry. In this degree, last-year students can take an optional subject titled Rehabilitation and Disability aid systems, which has a fundamental role for the design of systems focused on both rehabilitation and providing support and aiding patients with certain

difficulties. This paper describes the methodology of this subject's lab classes, which are based on the concept of project-based learning methodology. The different laboratory sessions are divided into two blocks, differentiating between introductory sessions to fundamental contents and hardware devices, and a block where students develop a group project related to rehabilitation or disability aid with the concepts and knowledge acquired in the introductory session plus other literature research conducted by them, which is later documented using an IEEE manuscript template. This teaching approach has been very well received and has increased the motivation of the students, leading many of them to continue working in the same topic for their Final Degree Project. The proposed approach plays a key role in future Health Engineers both in terms of education and as a future field of employment.

Palabras clave

Aprendizaje Basado en Proyectos, Ingeniería de la Salud, dinámica en sesiones prácticas, trabajo en grupo.

1. Introducción y motivación

La asignatura en la que se enmarca este trabajo, Sistemas de Rehabilitación y Ayuda a la Discapacidad (SRAD), se imparte en el cuarto curso dentro del plan de estudios del Grado en Ingeniería de la Salud. Este grado tuvo su origen en 2011, apoyado y propulsado por el Campus de Excelencia Internacional Andalucía Tech, formado por la Universidad de Málaga y la Universidad de Sevilla. El Grado en Ingeniería de la Salud en el que se sitúa el plan de estudios mencionado tiene como objetivo cubrir un perfil profesional básico que permita a los egresados acometer actividades orientadas a diseñar, desarrollar, seleccionar y evaluar siste-

mas, instalaciones y procesos en el ámbito de la Salud. Este grado está dividido en dos ciclos, que tienen lugar los dos primeros y los dos últimos años, respectivamente. El primer ciclo se trata de un ciclo común a todos los alumnos del grado, y reúne aquellas asignaturas de conocimiento transversal y que sientan las bases de lo que en el segundo ciclo será una formación mucho más específica. En particular, algunas asignaturas clave están relacionadas con contenidos relacionados con álgebra, cálculo, estadística, física y programación, las cuales son transversales a toda Ingeniería, aunque también se imparten otras más propias del grado en cuestión como aquellas relacionadas con biología, anatomía y fisiología. Sin embargo, es en el segundo ciclo donde un Ingeniero de la Salud escoge entre tres menciones, a elegir entre Ingeniería Biomédica, Informática Clínica y Bioinformática, donde se especializa en un determinado campo. En el caso de la mención en Ingeniería Biomédica, su perfil se encuentra más focalizado al de un Ingeniero de Hardware dentro del ámbito sanitario, focalizándose en diseño, desarrollo, y mantenimiento de sistemas de instrumentación, maquinaria y control de instalaciones en hospitales y todo tipo de centros sanitarios. Es en esta mención donde se encuentra la asignatura en la que nos centramos en este trabajo, la cual fomenta y promueve la formación en diseño para todos y en tecnologías de apoyo a la rehabilitación [7, 2, 3]. Dicha asignatura, la cual es optativa, tiene un papel fundamental para el diseño de sistemas enfocados tanto a la rehabilitación como a proveer facilidades a pacientes con ciertas dificultades, además de proveerle conocimientos con un fuerte impacto social y que pueden ayudar a definir el futuro laboral del estudiante. Debido a que la carga en horas de clases prácticas en esta asignatura es el doble que las asignadas a la teoría (3 créditos ECTS frente a 1.5 de teoría), los docentes dan a ésta un carácter eminentemente práctico, donde los alumnos trabajan y participan activamente en el laboratorio y desarrollan un proyecto real en equipo siguiendo la metodología de aprendizaje basado en proyectos (ABP) [4, 5, 8, 1], la cual fue introducida en el curso 2019/2020. Esta metodología ha demostrado ser eficaz a la hora de que los alumnos adquieran una serie de competencias transversales relacionadas con el trabajo y la comunicación en grupo, además de la capacidad de resolver problemas y situaciones que se planteen de forma colaborativa. En este artículo se describe la metodología de las prácticas de laboratorio para la docencia en dicha asignatura, y se aportan resultados tanto a nivel cuantitativo como cualitativo de la experiencia de los profesores durante los dos últimos cursos.

2. Estructura de los contenidos

Los contenidos prácticos de la asignatura complementan y afianzan los conceptos aprendidos en las clases magistrales, especialmente en las primeras sesiones de laboratorio. Dentro de los márgenes que permiten los planes de estudio y el programa docente, las clases prácticas ahondan y van un paso más allá de los conceptos estudiados en la teoría. En concreto, los contenidos de las sesiones de prácticas de la asignatura están estructurados de la siguiente forma:

- La primera sesión se centra en la accesibilidad web, donde se imparten conceptos básicos sobre diseño para todos dentro de este ámbito, los cuales se ven reforzados por los contenidos teóricos de la asignatura. En esta sesión los alumnos conocen distintas herramientas que reportan la calidad de la accesibilidad de un sitio web, dando indicios de qué elementos y decisiones son los más adecuados y porqué. Entre estas herramientas, los alumnos hacen uso de WAVE (Web Accessibility Evaluation Tool)¹ y Google Lighthouse².
- La segunda sesión práctica se centra en una introducción a diferentes interfaces de entrada de las que un usuario puede hacer uso para controlar el ratón y el teclado del ordenador de forma accesible. Destacamos el uso de una webcam junto con el software Camera Mouse³, el teclado en pantalla tanto en el modo por defecto como en modo scan, un micrófono con el que poder dar órdenes al asistente personal del sistema operativo, y un dispositivo de eye tracking [6] como Tobii 4C⁴.
- La tercera sesión acerca a los estudiantes al desarrollo de un sistema final que puede ser orientado tanto a una aplicación directa a rehabilitación como a accesibilidad y control del ordenador por gestos. Para ello, se hace uso del dispositivo Leap Motion⁵. Se les aporta un proyecto base en Visual Studio para que, utilizando C#, la librería del dispositivo y una serie de directrices sean capaces de desarrollar un sistema que permita manejar y hacer click del ratón gracias al movimiento de la mano en el aire.
- Las sesiones que abarcan desde la cuarta hasta la penúltima (9 sesiones en total) están orientadas al proyecto de la asignatura. Es aquí donde los profesores aplican la metodología ABP, de forma que los alumnos sean capaces de desarrollar y documentar un proyecto o sistema relacionado con los contenidos vistos en la asignatura.

¹<https://wave.webaim.org>

²<https://developers.google.com/web/tools/lighthouse>

³<http://www.cameramouse.org>

⁴<https://www.tobii.com>

⁵<https://www.ultraleap.com>

La última sesión práctica está destinada a la presentación de dichos proyectos, donde los alumnos deben demostrar el trabajo realizado y defenderlo ante un tribunal formado por los profesores de la asignatura. A continuación se explicará en detalle la metodología docente de las sesiones prácticas, donde se hará especial hincapié en las sesiones destinadas al proyecto.

3. Metodología docente

Las distintas sesiones prácticas de la asignatura pueden dividirse en dos bloques principales: un primer bloque que correspondería a las tres primeras sesiones, y un segundo bloque, que correspondería al proyecto final de la asignatura. En el caso del primero, la metodología aplicada para el desarrollo de las prácticas consiste en proporcionar a los estudiantes un boletín que contiene una introducción teórica y una serie de hitos a conseguir. En esas sesiones, el profesor introduce los contenidos y guía a los estudiantes por las diferentes herramientas o dispositivos que necesitan. Sumando esto a los hitos conseguidos durante las dos horas de cada práctica, las cuales son meramente presenciales, cada alumno desarrolla una memoria entregable fuera del aula y que, entre todas, supone un 20 % de la evaluación continua de la asignatura.

El segundo bloque anteriormente mencionado es donde se aplica directamente la metodología ABP. En la primera sesión se explica el workflow que se seguirá durante las sesiones, además de dar a conocer el inventario de material del cual se dispone y que los alumnos pueden solicitar para la realización del proyecto. Este puede ser realizado por una persona, aunque siempre se aconseja que se realice en grupos de no más de 3-4 personas. Los alumnos documentarán el desarrollo del proyecto en una memoria con formato paper de la editorial IEEE, además de presentarlo oralmente ante clase, un tribunal lo evaluará, otorgando una nota tanto a la memoria como a la presentación, que dan lugar a un 45 % de la nota de la asignatura en la evaluación continua. Junto a esto, se les pide, una vez tengan la idea definida, realizar un elevator pitch para dar a conocer su idea al resto de estudiantes y así poder establecer relaciones entre grupos que usen tecnologías similares.

El workflow establecido divide de forma orientativa a las sesiones prácticas, a las que los alumnos deben acudir, asociadas al proyecto en los siguientes puntos:

- Elección de tecnología y revisión del estado del arte: 4 horas. Los alumnos deciden el proyecto a realizar y buscan información sobre el estado del arte al respecto.
- Diseño y desarrollo del proyecto/prototipo: 8 horas. Los alumnos desarrollan y documentan el proyecto.

- Diseño de test y pruebas: 2 horas. Los alumnos realizan y documentan una serie de pruebas al sistema para demostrar su correcto funcionamiento.
- Preparación de la presentación: 2 horas. Los alumnos preparan una presentación que resuma el proyecto desarrollado en 10 minutos.
- Presentación oral del trabajo. La última sesión se dedica a que cada grupo exponga su proyecto, donde un tribunal formado por los profesores de la asignatura lo evalúan y hacen preguntas.

Cada semana, los alumnos asisten a clase para trabajar en grupo, siendo supervisados por el profesor, al que pueden consultar los problemas o dudas que surjan durante el desarrollo. Fuera del aula de prácticas, se espera un trabajo por parte del alumno de 45 horas totales, que incluyen tanto el tiempo dedicado a las memorias individuales correspondientes al primer bloque práctico, como al desarrollo y documentación del proyecto grupal. El 35 % de la evaluación alternativa corresponde a un examen de los contenidos teóricos de la asignatura.

4. Resultados obtenidos

Con la metodología descrita en anteriores secciones se ha logrado una mejoría en la calidad de la docencia y de los resultados de los alumnos, que pueden verse reflejados tanto de forma cuantitativa como cualitativa en diversos factores.

En primer lugar, las notas finales de los alumnos en el curso 2020/2021 han mejorado frente a las del curso anterior. Mientras que en el curso 2019/2020 la nota media era de un 7.70, este curso la nota media pasa a ser de sobresaliente (9.13), teniendo en cuenta que se han seguido los mismos mecanismos de evaluación y seguimiento, aunque adaptados a la situación de semipresencialidad debida a la COVID-19. La Figura 1 refleja el histograma con las notas medias finales tanto para el curso actual como para el anterior. Son muchos los alumnos que se han matriculado este curso gracias a las recomendaciones de antiguos alumnos de la asignatura. Esto provoca una motivación de base por parte de los alumnos que, al conocer la dinámica de la asignatura a través de estas recomendaciones, les hace idear proyectos mucho más interesantes y de mayor calibre como trabajo final. Estos datos cualitativos pudieron ser recogidos gracias a una encuesta de satisfacción anónima realizada al final del curso donde los alumnos puntuaban del 1 al 5 (siendo 5 la nota más alta y 1 la más baja) los siguientes campos: interés por la asignatura, concordancia con el proceso de evaluación, satisfacción con la metodología docente, relevancia de la asignatura dentro de la especialidad, y posibilidad de recomendar la asignatura. Esta encuesta dio como

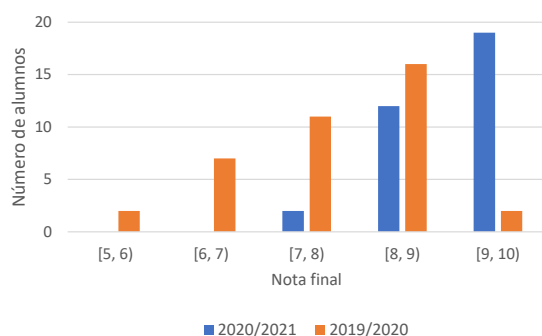


Figura 1: Histograma de la evolución de las notas finales de la asignatura SRAD entre los cursos 2019/2020 y 2020/2021.

resultado los siguientes resultados medios: 4.38, 4.72, 4.81, 4.02, y 4.54, respectivamente.

A pesar de la situación actual, son muchos los alumnos que han decidido hacer proyectos muy ambiciosos y que incluso podrían haberse tratado de forma individual como Trabajos Fin de Grado (TFGs). Esta motivación se ha visto reflejada en un incremento de las tutorías solicitadas por los alumnos y en no conformarse con el proyecto mínimo viable para obtener la nota suficiente con la que aprobar. Esto, de nuevo, se ve claramente en las notas mostradas en la Figura 1.

Mientras que el año pasado fueron 3 los alumnos que mostraron inquietud en ampliar el trabajo de la asignatura de cara a realizar su TFG, este año ha aumentado a 6 alumnos, lo que supone un 18 % de la clase frente al 7.9 % correspondiente en el año anterior.

5. Conclusiones

En este trabajo se recoge y presenta la metodología de aprendizaje basado en proyectos aplicada a la docencia en las sesiones prácticas de la asignatura SRAD, la cual se imparte dentro de la mención en Ingeniería Biomédica del Grado en Ingeniería de la Salud. En estas sesiones los alumnos llevan a cabo un proyecto en grupo centrado en los contenidos de la asignatura y en el que desarrollan un sistema orientado a la rehabilitación de un paciente o a la ayuda a la accesibilidad de un grupo de usuarios con cierta discapacidad. Los proyectos son realizados de forma grupal por parte de los alumnos, apoyándose en la tutorización de los profesores, lo cual refuerza habilidades y competencias transversales como el trabajo en grupo, la comunicación entre los miembros y el autoaprendizaje. Los informes cualitativos llevados a cabo reflejan una muy alta motivación e implicación por parte de los alumnos, que ven en esta asignatura un espacio para aprender nuevas tecnologías que puedan serles de utilidad en su futuro laboral dentro del ámbito sanitario. El enfoque de in-

vestigación que se les pide a los estudiantes les acerca, por otro lado, a dicho mundo, abriéndoles las puertas al recientemente creado Máster en Ingeniería Biomédica y Salud Digital. En cuanto a la evaluación cuantitativa, la alta implicación de los alumnos se ve reflejada en notas que en ningún caso bajan del notable y donde el número de suspensos en evaluación alternativa ha sido de cero alumnos. Los trabajos realizados en la asignatura han dado lugar a numerosos Trabajos Fin de Grado, cuya calidad se ve incrementada al haber sido tutorizados desde el primer cuatrimestre, donde se imparte la asignatura que se trata en este artículo.

Los conocimientos adquiridos en la asignatura abren a los alumnos muchas puertas en el ámbito laboral, especialmente aquellos centrados en sistemas de rehabilitación. Son muchas las empresas que están desarrollando sistemas robóticos y prótesis impresas en 3D para la rehabilitación de miembros superiores e inferiores. La metodología de trabajo llevada a cabo en la asignatura, centrada en conseguir una prueba de concepto de un producto final, introduce de lleno al estudiante en la filosofía de trabajo en equipo de una empresa, donde las distintas partes colaboran y unifican el trabajo de cara a dar una solución conjunta.

Referencias

- [1] Elena Cerezuela-Escudero et al. Aprendizaje basado en proyectos: implementación de interfaces gráficas para microcontroladores ARM. *JENUi*, 2014.
- [2] John Clarkson and other. Inclusive design: Design for the whole population. *Springer Science & Business Media*, 2013.
- [3] Albert Cook and Janice Miller Polgar. *Assistive technologies-e-book: principles and practice*. Elsevier Health Sciences, 2014.
- [4] Diana Dolmans et al. Problem-based learning: Future challenges for educational practice and research. *Medical education*, 39(7):732–741, 2005.
- [5] Barbara Duch, Susan Groh, and Deborah Allen. *The power of problem-based learning: a practical "how to" for teaching undergraduate courses in any discipline*. Stylus Publishing, LLC., 2001.
- [6] Päivi Majaranta. *Gaze Interaction and Applications of Eye Tracking: Advances in Assistive Technologies: Advances in Assistive Technologies*. IGI Global, 2011.
- [7] Marcia Scherer and Rob Glueckauf. Assessing the benefits of assistive technologies for activities and participation. *Rehabilitation Psychology*, 50(2):132, 2005.
- [8] Gwen Solomon. Project-based learning: A primer. *Technology and learning-dayton-*, 23(6):20–20, 2003.

Ayuda al aprendizaje de la comprobación de tipos mediante una API de desarrollo

Eva López Puente

Departamento de Informática y Estadística
Universidad Rey Juan Carlos
Móstoles
e.lopezpu@alumnos.urjc.es

Jaime Urquiza Fuentes

Departamento de Informática y Estadística
Universidad Rey Juan Carlos
Móstoles
jaime.urquiza@urjc.es

Resumen

La enseñanza de compiladores debe afrontar diferentes dificultades, entre ellas el nivel de abstracción de los conceptos que se manejan y el esfuerzo de desarrollo por parte de los estudiantes a la hora de trabajar en los proyectos propuestos en este tipo de asignaturas. Este trabajo presenta una API dedicada al desarrollo de sistemas de tipos. No evita que los estudiantes afronten la complejidad del desarrollo de un sistema de tipos, pero facilita la tarea ofreciendo una infraestructura básica para la manipulación de expresiones de tipo contemplando diversos paradigmas de programación. Aunque este trabajo sigue en progreso, se ha realizado una evaluación preliminar con un enfoque analítico que ha dado resultados prometedores.

Abstract

The teaching of compilers must face different difficulties, among them the level of abstraction of the concepts that are handled and the development effort on the part of the students when working on the projects proposed in this type of subjects. This paper presents an API dedicated to the development of type systems. It does not prevent students from facing the complexity of developing a type system, but it does facilitate the task by offering a basic infrastructure for manipulating type expressions by contemplating various programming paradigms. Although this work is still in progress, a preliminary evaluation with an analytical approach has been carried out and has given promising results.

Palabras clave

Enseñanza de compiladores, Prácticas, Sistemas de tipo, API de desarrollo.

1. Introducción

Compiladores es una asignatura compleja debido fundamentalmente al grado de abstracción de los conceptos que maneja [3]. Este trabajo se enmarca en una asignatura de máster de 6 ECTS (4 horas semanales durante un cuatrimestre) donde la primera mitad se dedica a compiladores y la segunda mitad a intérpretes y máquinas virtuales. El temario dedicado a compiladores supone conocimientos previos de procesadores de lenguajes hasta los fundamentos de traducción dirigida por la sintaxis (comúnmente impartidos en los grados de informática) y trata los temas de tabla de símbolos, comprobación de tipos, generación de código intermedio, optimización de código y generación de código final. Asociado a los temas de tabla de símbolos y comprobación de tipos existe una práctica sobre sistemas de tipos cuyo desarrollo lleva toda la mitad del cuatrimestre.

La complejidad de esta asignatura ha motivado el desarrollo de un buen número de herramientas para la docencia. Los ejemplos más significativos son las herramientas generadoras de parsers, que a pesar de tener una clara utilidad en el mundo profesional también lo son en la enseñanza. Así, se pueden encontrar generadores de analizadores sintácticos ascendentes como CUP¹, descendentes como Coco² o generadores de analizadores léxicos como Flex³. Pero también hay otro tipo de herramientas, algunas más centradas en los fundamentos teóricos como JFLAP [5], que permite experimentar de forma visual e interactiva conceptos de autómatas y lenguajes formales o algo más prácticos como VAST[2], que está centrada la visualización de árboles sintácticos y su proceso de construcción.

Aun así, no se ha podido encontrar nada desarrollado más relacionado con el objeto de este trabajo, centrado en facilitar el aprendizaje de la parte de comprobación de tipos mediante un API específica que propor-

¹<http://www2.cs.tum.edu/projects/cup>, 2021

²<http://ssw.jku.at/coco>, 2021

³<https://github.com/westes/flex>, 2021

cione a los estudiantes una infraestructura que ahorre tiempo de trabajo sin que dejen de encarar todos los retos que supone aprender a manejar y construir sistemas de tipos para diversos paradigmas.

2. Sobre los sistemas de tipos

El sistema de tipos [1] de un lenguaje comprueba el correcto uso, desde el punto de vista semántico, de las construcciones del lenguaje. Así, indica cómo se clasifican los valores, las expresiones y cómo se pueden manipular e interactuar con ellos. Un tipo de dato es el conjunto de términos del lenguaje que posee unas características comunes que les permite interactuar entre sí o ser objeto de transformaciones solo aplicables a ellos mismos. Así, podemos diferenciar los tipos de datos en cada uno de los siguientes cuatro paradigmas: imperativo, declarativo funcional, declarativo lógico y orientado a objetos.

El paradigma imperativo hace uso de variables con valores modificables (lógicos, números, caracteres, ...), tipos valor o tipos de referencia, y tiene procedimientos y funciones que nos permite dividir el programa en bloques sencillos consiguiendo que sea más legible y comprensible por todos. Un lenguaje ejemplo sería *Pascal*.

El paradigma funcional tiene tipos que pueden cambiar de valor (valores lógicos, números, cadenas...), tipos algebraicos donde el usuario necesita una estructura no definida y la crea a partir de constructores de datos, así como listas y árboles. Un lenguaje ejemplo sería *Haskell*.

En el paradigma lógico se define un conjunto finito de fórmulas lógicas que reflejan el conocimiento del que se dispone para resolver un problema. Así, constaría de variables, constantes, listas, hechos y reglas para hallar la solución a la pregunta. Un lenguaje ejemplo sería *Prolog*.

El paradigma orientado a objetos tiene tipos primitivos, que no necesitan invocación para ser creados y tipos objetos, proporcionados por la bibliotecas. También se tienen arrays y tipos abstractos de datos, estos últimos creados por el usuario. Un lenguaje ejemplo sería *Java*.

3. Descripción de la API

El objetivo de este trabajo es desarrollar una API Java –lenguaje utilizado en la asignatura– que ayude a los estudiantes en el desarrollo de sistemas de tipo. Dicha API se basa en una jerarquía de clases (véase la figura 1) que representa los diferentes tipos que podrían ser necesarios manejar. Comienza con una clase común de

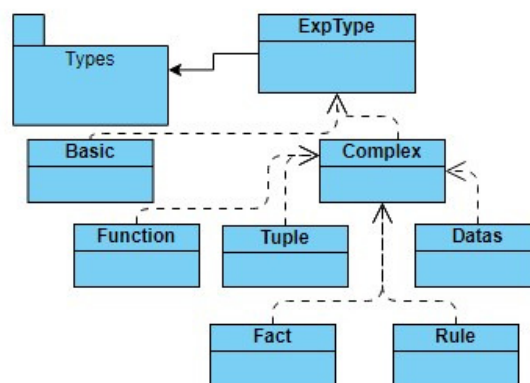


Figura 1: Diagrama de clases del paquete *Types* dedicado a los tipos.

tipos (*ExpType*) en la que se podrá almacenar el nombre y el resultado booleano de la comparación entre dos tipos. A través de esta clase se realizará la división en tipos básicos (*Basic*) y complejos (*Complex*).

Los tipos básicos se crearán fácilmente utilizando su nombre *ExpType createBasic (String name)* y se comparará con otro tipo a través del nombre y el contenido almacenado, devolviendo *true* o *false*, según sean comparables o no, *boolean compareType (ExpType e1, ExpType e2)*.

En los tipos complejos, el contenido almacenado de una expresión de tipo se coloca en un array, puesto que serán funciones (clase *Function*), pares de elementos o tuplas (clase *Tuple*) y estructuras de datos (clase *Datas*), y no se pueden almacenar como tipos básicos, asimismo, tendrá el método de comparación para conocer si dos expresiones son comparables o no. En este caso, el método sí es el mismo que en el apartado para crear tipos básicos.

Una función está formada por los elementos que recibe como argumentos y el tipo devuelto, por tanto, necesitamos una estructura que nos permita guardar dos expresiones de tipo, en este caso, hemos optado por un array de dos elementos, siendo el primero los argumentos y el segundo aquello que devuelve la función, *ExpType createFunction (ExpType e1, ExpType e2)*.

Para el caso de las tuplas, se almacenarán, de nuevo en un array de dos elementos, las expresiones de tipo. Si fuera necesario almacenar más de dos expresiones, éstas se concatenan con la anterior, es decir, en el primer hueco del array irían las dos 'antiguas' mientras que en el segundo aparecería la nueva, *ExpType createTuple (ExpType e1, ExpType e2)*.

Por último, las estructuras de datos, donde no se conoce de antemano qué es lo que el usuario va a almacenar, por tanto, en vez de almacenar las expresiones en un array de tamaño definido, esta vez será una lista dinámica de expresiones de tipo y se añadirán a través del método *add* de las listas, *Datas createDatas (String*

name), se proporciona un nombre a la estructura y dentro tendrá la lista de expresiones.

Con todo lo aportado anteriormente se cubren los paradigmas imperativo, declarativo funcional y orientado a objetos. Sin embargo, la programación lógica es un poco especial y se han creado nuevas clases que permitan manipular sus elementos fundamentales: los hechos y las reglas. Para crear un hecho, primero se crean los objetos *ExpType* *createObject* (*String name*) que forman parte de ese hecho y después se les asignará una relación en el hecho, *ExpType createFact* (*String name*, *ExpType e1*). En el caso de las reglas, nuestro array será de dos elementos, donde el primero corresponde al hecho de la parte izquierda de una regla, la cabeza, mientras que el segundo serán todos los hechos que indican que esa regla es cierta, la parte derecha de la misma o cuerpo, *ExpType createRule* (*ExpType e1*, *ExpType e2*).

4. Ejemplos de uso

En este apartado vamos a dar unos ejemplos de cómo se utilizaría esta API y las salidas que nos proporcionaría. En todos los paradigmas, lo primero que hay que hacer es crear un objeto de aquel que se va a utilizar, *XParadigm pX = new XParadigm()*;, donde 'X' se sustituye por el nombre del paradigma. Una vez tenemos el paradigma, se procede a crear tipos básicos o complejos, según se necesiten. Además la API proporciona una salida textual a modo de log para depuración informando sobre las acciones realizadas. En la figura 2 se puede ver un fragmento de la especificación de un parser utilizando la API para realizar labores de comprobación de tipos. Usaremos esta figura para los ejemplos descritos a continuación.

4.1. Crear un tipo básico

Para crear un tipo básico basta con darle un nombre y asignarlo a una expresión de tipo para que después se pueda usar, por ejemplo, si queremos crear un tipo entero o un tipo en coma flotante, sería *ExpType t0 = pI.createBasic('int');*;, '*ExpType t1 = pI.createBasic('float');*;', respectivamente, donde 'pI', indica el paradigma que se está utilizando, en este caso, el paradigma imperativo. La salida textual mostraría *BasicType : int*, *BasicType : float*. Su uso se puede ver en las líneas 1-3 de la figura 2.

4.2. Crear un tipo complejo

Un tipo complejo, como se ha dicho anteriormente, sería una función. Supongamos que tenemos una función que recibe un valor entero y nos devuelve un valor en coma flotante, primero tendríamos que crear

```

1 value returns [ExpType t] :
2   NUM_INT_CONST { $t = pI.createBasic('int'); }
3   | NUM_REAL_CONST { $t = pI.createBasic('float'); };
4
5 dclfun returns [ExpType t] :
6   type IDENT '(' formalparams ')' ';' {
7     $t = pI.createFunction($type.t, $formalparams.t);
8     // ST es el objeto de la tabla de símbolos
9     ST.insertType($IDENT.text, $t);
10  };
11
12 dclstruct returns [ExpType t] :
13   'struct' IDENT {
14     Datas tipo = pI.createDatas($IDENT.text);
15   }
16   '{' dclvarlist[tipo] ';' {
17     $t = tipo;
18   };
19
20 dclvarlist[ExpType t] :
21   type IDENT ';' {
22     (Datas)$t.addElement($type.t);
23   } dclvarlist[$t]
24   |
25   ;
26
27 sent :
28   IDENT '=' exp {
29     ExpType t1 = ST.getType($IDENT.text);
30     if(!pI.compareTypes(t1, $exp.t)){
31       // Tratar error de tipos ...
32     }
33   };

```

Figura 2: Ejemplo especificación de parser con llamadas a la API para comprobación de tipos

los tipos básicos para el entero (*t0*) y la coma flotante (*t1*) y después crear la función, la cual se crearía como *ExpType t2 = pI.createFunction(t0, t1)*;. Su salida se mostraría como *ComplexType : Function : (int) ->(float)*. La utilización de estos tipos se puede ver en las líneas 5-10 de la figura 2 donde se declara una función.

4.3. Crear una estructura

Al igual que ocurre con las funciones o las tuplas, primero es necesario crear los objetos básicos que forman la estructura, para después poder añadirlos. Así, si queremos una estructura formada por tres colores (rojo, verde y azul), primero crearemos los básicos asociados. A continuación, la estructura que los almacena, *Datas color = pF.createDatas('dataColor');*;, y después, añadirle a esa estructura o lista los tipos básicos, *color.addElement(rojo)*, así con cada uno de ellos. Una vez está creado, se mostraría al usuario como *ComplexType : dataColor (Rojo, Verde, Azul)*. El uso de estas estructuras se puede ver en las líneas 12-25 de la figura 2 donde se reconoce una sentencia '*struct*'.

4.4. Comparación de tipos

En la comparación de tipos, el programa muestra aquello que se está comparando y el resultado obtenido de la misma. Igualmente, es necesario acceder al método de comparación a través del programa seleccionado. Si comparamos un valor entero (*t0*) con un valor en coma flotante (*t1*), *pI.compareTypes(t0, t1)*, el

programa nos devuelve *Comparación int con float: false*. Se puede ver su uso en las líneas 27-33 de la figura 2 donde se comprueban los tipos de una sentencia típica de asignación.

5. Evaluación preliminar

Debido a las restricciones actuales impuestas por la pandemia, no ha sido posible evaluar esta API en contexto educativo controlado, por ello hemos utilizado un enfoque más analítico. El diseño de una API abarca decisiones que van desde su arquitectura y funcionalidad hasta el nombre específico de clases, funciones o métodos. Myers y Stylos [4] proponen una manera sencilla de evaluar el diseño APIs adaptando las diez reglas heurísticas de Nielsen. Nuestra API cumple siete de estas diez pautas. Descartamos la *visibilidad e información del sistema* ya que no podemos aportar en qué estado de ejecución o comprobación se encuentra; el *control y libertad del usuario*, dado que en principio, el usuario no tiene acceso al código; y por último la *recuperación de errores* (que no está relacionado con los errores detectados por el sistema de tipos en desarrollo).

Según nuestro análisis, nuestra API cumple el resto de reglas. Los nombres de los métodos y su organización en clases coinciden con las *expectativas de los usuarios*, indicando nombres genéricos y conocidos. Todas las partes de la API son *coherentes* a través de los nombres identificativos y semejanza en el uso. Se *ayuda al usuario* haciendo que la API funcione correctamente, por ejemplo, con valores predeterminados o errores de escritura. Es indispensable que los *nombres sean claros y comprensibles*, no demasiado largos y que no exista un mismo método repetido varias veces con el único cambio en el número de argumentos, así se puede utilizar la ventana emergente de autocompletar en los entornos de programación y se necesita escribir poco para que los reconozca. Las tareas se realizan de manera eficiente consultando al mínimo el manual de ayuda, debido a que el usuario apenas tiene funciones que *recordar*. Por último, es posible utilizar la API a través de un *manual* que se proporciona junto con el ejecutable de la misma.

6. Conclusiones y trabajos futuros

Este trabajo presenta una API cuyo objetivo es ayudar a los estudiantes a la hora de desarrollar las prácticas de la asignatura de Compiladores relacionadas con la fase de comprobación de tipos. La API pretender abarcar diferentes paradimas, ofreciendo una estructura que permite a los estudiantes manejar fácilmente las

expresiones de tipos sin evitar la complejidad que supone el diseño de parsers que implementan sistemas de tipos. Hemos analizado la API según las heurísticas de Myers y Stylos [4] obteniendo un buen resultado.

No olvidamos que este es un trabajo en curso y se pueden realizar mejoras, por ejemplo comparando la complejidad de las soluciones con más parámetros aparte del número de líneas de código. También se pretende aumentar el juego de prueba de prácticas para realizar las anteriores comparativas con una población mayor. Finalmente, si el contexto actual lo permitiera se planea realizar un experimento controlado con estudiantes para hacer un estudio más detallado de la aportación de la API al proceso de aprendizaje.

Para terminar, la API podría ampliar su funcionalidad, tanto a nivel detallado permitiendo que los programadores cambien la salida de log de depuración según sus necesidades como a nivel conceptual cubriendo otros aspectos como la generación y representación de código intermedio.

Agradecimientos

Este trabajo ha sido financiado por el Ministerio de Economía y Competitividad (ref. TIN2015-66731-C2-1-R) y la Comunidad de Madrid, con el proyecto e-Madrid-CM (P2018/TCS-4307), también cofinanciado por los fondos estructurales (FSE y FEDER).

Referencias

- [1] Alfred V. Aho, Monica S. Lam, Ravi Sethi y Jeffrey D. Ullman. *Compilers: Principles, Techniques, and Tools* (2nd Edition) Addison-Wesley Longman Publishing Co., Inc. 2006.
- [2] Francisco J. Almeida-Martínez, Jaime Urquiza-Fuentes y J. Ángel Velázquez-Iturbide. *Visualization of Syntax Trees for Language Processing Courses* J. of Universal Computer Science 15(7):1546–1561, apr 2009.
- [3] Michael Hewner. *Undergraduate Conceptions of the Field of Computer Science*. In ICER 2013, pages 107–114, New York, NY, USA, 2013. ACM.
- [4] Brad A. Myers y Jeffrey Stylos. *Improving API Usability*. Association for Computing Machinery: 62–69, Communications of the ACM, 2016.
- [5] Susan H. Rodger, Eric Wiebe, Kyung Min Lee, Chris Morgan, Kareem Omar y Jonathan Su. *Increasing Engagement in Automata Theory with JFLAP*. ACM Press: 403–407. Proceedings of the 40th ACM Technical Symposium on Computer Science Education 2009.

De la simulación a la realidad: metodología para el aprendizaje de sistemas ciber-físicos

Aitor Arrieta, Gaizka Bellido, Jorge Da Silva, Urtzi Markiegi
Departamento de Electrónica e Informática
Mondragon Unibertsitatea
20500 Arrasate-Mondragon
{aarrieta, gbellido, umarkiegi}@mondragon.edu
jorge.dasilva@alumni.mondragon.edu

Jennifer Gago
Dpto. Técnico Académico
MathWorks
Dirección MathWorks
jgagomu@MathWorks.com

Resumen

El estudio de los sistemas ciber-físicos en el marco de la asignatura Informática Industrial no resulta atractivo para el alumnado del grado de ingeniería informática. Este aspecto queda reflejado en las encuestas de satisfacción que el alumnado rellena cada curso. Otras asignaturas como programación web, inteligencia artificial o seguridad son más atractivas a priori para los estudiantes. Con objeto de mejorar la motivación e implicación del alumnado, los cursos 19-20 y 20-21 se ha incorporado la herramienta de simulación MATLAB® y Simulink® para desarrollar parte de las competencias. Los resultados positivos obtenidos indican que es el camino a seguir. Además se prevé que complementar la herramienta de simulación con maquetas (basadas en hardware Arduino) que permitan experimentar la parte física, resultará en una formación más completa y atractiva para el alumnado. El artículo presenta dos contribuciones relevantes para la comunidad. Primero, se presenta la metodología seguida para establecer un marco de colaboración entre universidad y empresas líderes del sector de sistemas ciber-físicos. Y segundo, se presenta la metodología llevada a cabo para incorporar las nuevas tecnologías, alinearlas con los objetivos de aprendizaje y adaptar los criterios de evaluación, de cara a mejorar la experiencia de aprendizaje de los estudiantes.

Abstract

The study of cyber-physical systems within the framework of the Industrial Computer Science subject is not particularly appealing to students of the computer engineering degree. This aspect is reflected in the satisfaction surveys that students fill out each year. Other subjects such as web programming, artificial intelligence or security are more attractive a priori for students. With the aim of improving student motivation

and involvement, the simulation tool MATLAB and Simulink has been incorporated in academic years 19-20 and 20-21 to develop part of the skills. The positive results obtained indicate that this is the way forward. Furthermore, it is expected that complementing the simulation tool with mockups (based on Arduino hardware) that allow students to experience the physical part, will result in a more complete and attractive learning experience for students. The article presents two relevant contributions for the community. Firstly, it presents the methodology followed to establish a collaboration framework between the university and leading companies in the cyber-physical systems sector. Secondly, it presents the methodology used to incorporate new technologies, align them with learning objectives and adapt evaluation criteria, in order to improve the learning experience of students.

Palabras clave

Aprendizaje basado en proyectos, Sistemas ciber-físicos, Informática industrial.

1. Introducción

La metodología de aprendizaje basado en proyectos ha sido demostrada como una metodología efectiva a la hora de enseñar y motivar a los alumnos [1, 2, 5, 7]. Los sistemas ciber-físicos integran tecnologías digitales con procesos físicos. El diseño y control de sistemas ciber-físicos es instruido en diferentes cursos a través de diferentes asignaturas, no solo del grado en ingeniería informática, sino también en otros grados y másteres (p.ej., grado en ingeniería electrónica). La naturaleza multidisciplinar de los sistemas ciber-físicos, donde la electrónica, mecánica y software son integrados conjuntamente, hace difícil que las asignaturas donde se trabajan tales competencias sean atractivas para el alumnado.

Este artículo tiene como objetivo reportar la experiencia de un proyecto que se está llevando a cabo de cara a cambiar la manera en la que se enseña el diseño y control de los sistemas ciber-físicos. En lugar de hacer uso únicamente de herramientas de simulación, se hará uso de ambas, las herramientas de simulación y maquetas reales basados en los Arduino Engineering Kits (AEK).

La Universidad de Mondragon y MathWorks, la empresa líder en modelado y simulación de sistemas ciber-físicos, han lanzado un proyecto colaborativo de cara a mejorar el contenido académico para la enseñanza del diseño y control de sistemas ciber-físicos. Para ello se analizarán los tres proyectos AEK (motocicleta auto-balanceada, robot controlado por web-cam y robot de dibujo), obteniendo para cada uno de los proyectos su resultado de aprendizaje. Cada uno de estos resultados de aprendizaje se relacionará posteriormente con los resultados de aprendizaje de las asignaturas seleccionadas tanto de grado como de máster. Esto permitirá identificar qué proyecto AEK puede utilizarse en cada una de las asignaturas.

En el aspecto práctico, se desarrollarán librerías específicas para cada proyecto AEK que permitan un mayor uso de los kits en diferentes asignaturas, haciendo posible el uso de un proyecto AEK específico incluso sin tener conocimiento de todos los resultados de aprendizaje. Por ejemplo, para la asignatura de informática industrial que se imparte en el segundo grado de ingeniería informática, un resultado de la asignatura está relacionado con el control basado modelos utilizando Simulink y Stateflow y otro con el control basado en bloques. El proyecto AEK de control por cámara web hace uso de ambos resultados. Sin embargo, para desarrollar el proyecto también son necesarios conocimientos de visión artificial, lo que no permitiría utilizar este proyecto AEK en la asignatura ya que los alumnos no tienen conocimientos de visión artificial en segundo curso de grado. Este problema será resuelto por las librerías desarrolladas (por ejemplo, en este caso una librería de algoritmos de visión) para ser utilizadas por los estudiantes cuyos conocimientos son limitados en un resultado específico del proyecto AEK. Esto permitirá trabajar un resultado de aprendizaje específico mediante el uso de un proyecto AEK que implemente este resultado, incluso sin tener conocimiento de otros resultados de aprendizaje del proyecto AEK.

La contribución de este artículo se centra en compartir con la comunidad dos metodologías: (1) una metodología para establecer el marco colaborativo con proveedores de primer nivel del ámbito educativo-profesional de los sistemas ciber-físicos y (2) una metodología para acondicionar y completar los recursos proporcionados por los proveedores al currículo universitario.

2. Trabajos Relacionados

En los sistemas ciber-físicos se da una integración de los procesos de computación con los procesos físicos cuyo comportamiento está determinado tanto por la parte computacional como por la parte física. Por lo tanto, el diseño y desarrollo de los sistemas ciber-físicos requiere de aprendizaje multidisciplinar como ha sido recogido en múltiples trabajos [3, 6, 8, 9]. Stankovic et al. [8] identifican en su trabajo las áreas fundamentales a desarrollar al diseñar un currículo para el aprendizaje de los sistemas ciber-físicos. Estas áreas incluyen el desarrollo de conocimiento en computación, matemáticas discretas y continuas, sensores y modelado de sistemas dinámicos y de control. A diferencia del trabajo de Stankovic et al., en la metodología propuesta se plantean áreas más limitadas, pero con un nivel de desarrollo más práctico y detallado. En su trabajo, Mourtzis et al. [6] describen como Educación 4.0, la educación necesaria para dar respuesta a la cuarta revolución industrial en el ámbito de la fabricación. Para ello proponen un marco de educación para integrar los sistemas ciber-físicos con las tecnologías de la cuarta revolución industrial. La propuesta de este trabajo, sin embargo, se centra exclusivamente en las tecnologías relacionadas con los sistemas ciber-físicos. En su trabajo, Törngren et al., [9] recopilan los retos y oportunidades que los sistemas ciber-físicos proporcionan en el sistema educativo. Destacan como aspecto clave en el diseño de los planes de estudio buscar un equilibrio entre la profundidad y extensión de los conceptos trabajados, así como entre la teoría y la práctica. Esta propuesta profundiza en este concepto realizando una propuesta específica que permite adaptar la profundidad y extensión y la teoría y prácticas realizadas en función del contexto en el que se aplique. Recientemente, el estudio llevado a cabo por Marwedel et al. [3] ha analizado las diferentes propuestas que la educación superior ha realizado para dar respuesta a las necesidades de formación en sistemas ciber-físicos. En su trabajo, concluyen que es posible realizar una formación básica en los sistemas ciber-físicos con un coste moderado para los centros, si bien una formación más en profundidad requeriría de cursos más especializados y en consecuencia más costosos para los centros. Este trabajo persigue optimizar los recursos proporcionados por MathWorks y Arduino para adaptarlos al currículo, haciendo posible que se pueda profundizar en la materia de aprendizaje.

3. MATLAB Simulink y AEK

MATLAB Simulink es una herramienta para el diseño, modelado y desarrollo de sistemas de control basado en modelos, la cual es la principal herramienta

utilizada por desarrolladores de sistemas ciber-físicos [4]. Esto es debido a que, además de ser una herramienta muy potente de simulación, ofrece la posibilidad de generar código automáticamente. Este código es, además, compatible con diferentes estándares de seguridad, como por ejemplo AUTOSAR (estándar de automoción). Por ello, la Universidad de Mondragón cuenta con una licencia de campus que ofrece la posibilidad de utilizar MATLAB Simulink a todos los alumnos de la facultad de ingeniería.

Por otro lado, Arduino es una empresa de desarrollo de software y hardware libre que diseña y proporciona componentes hardware. Una parte de su negocio está dedicado a la educación, donde proporcionan soluciones para diferentes rangos de edades. Entre las ofertas de este negocio, se encuentran diferentes kits orientados a la capacitación de los estudiantes universitarios (p.ej., Kit de IOT). Entre estos kits se encuentra el Arduino Engineering Kit (AEK), el cual proporciona tres proyectos hardware con todos los componentes requeridos para llevar a cabo cada proyecto. Este kit tiene como objetivo proporcionar una comprensión de los conceptos básicos de sistemas ciber-físicos a través de un entorno de aprendizaje colaborativo.

Se ha seleccionado el AEK como kit para la inclusión en el currículo de las diferentes asignaturas debido a su alta interconexión con MATLAB y Simulink, ya que cubre los conceptos clave de programación en este entorno (p.ej., sistemas de control, diseño basado en modelos, procesamiento de imágenes, etc.). La posibilidad de que los alumnos pudieran ver un resultado práctico del uso de la herramienta MATLAB Simulink conllevó a que nos decantásemos por este kit de cara a evolucionar una enseñanza basada en la simulación de los sistemas al uso de hardware real. El kit fue galardonado en los BETT 2020 awards debido a que “es un juego de herramientas innovador y bien diseñado que permite a los estudiantes de ingeniería utilizar el software estándar de la industria para completar proyectos interesantes, atractivos y relevantes”.

4. Metodología

Esta sección presenta las dos metodologías definidas para llevar a cabo el proyecto. Estas metodologías preferiblemente recomendamos que se apliquen secuencialmente, de forma que la primera garantice el contexto de desarrollo de la segunda.

4.1. Metodología Colaborativa

Para llevar a cabo el proyecto se ha establecido una metodología colaborativa entre la universidad y MathWorks, en el que ambos agentes son beneficiarios. Además, MathWorks financia la beca de un becario y

su supervisión. En esta sección se explica dicha metodología y cómo se puede establecer en el contexto de otras universidades.

Una vez aceptado el proyecto, se lanza el proyecto, estableciendo para ello los diferentes actores que formarán parte en ella. Los actores de la universidad son (1) el becario, (2) el supervisor y (3) el coordinador de título. Por otra parte, el actor por parte de MathWorks será el responsable de la universidad. El becario tiene como objetivo principal la puesta en marcha de la parte técnica, en este caso, los tres proyectos de los AEKs. También apoyará en tareas de documentación. El supervisor tiene como objetivo ayudar en la parte técnica al becario, estableciendo para ello reuniones regulares (2-3 por semana). Además, es responsable de la generación del material docente que va a ser puesto a disposición de la comunidad académica. Por otro lado, se encargará de gestionar las relaciones con MathWorks mediante reuniones bisemanales y reportar los resultados al coordinador de título. El coordinador de título se asegura que todo el trabajo cumple con los criterios establecidos por la universidad, y coordina la relación con profesores de otras asignaturas (tanto del grado en informática como de otros grados y másteres) de cara a que el material pueda ser lo más reutilizado posible. Por último, el responsable de la universidad por parte de MathWorks será el encargado de supervisar todo el proceso, asegurarse que el contenido generado cumple con los criterios de calidad establecidos por MathWorks y da soporte técnico a la universidad cuando estos requieren de ella.

Los siguientes pasos son simples. El becario se encarga de la parte técnico-tecnológica, asesorado por el supervisor para tareas técnicas de las que carece de dominio técnico. Cada dos o tres días, el becario y el supervisor se reúnen para realizar el seguimiento técnico y definir los siguientes pasos a dar. Se han montado dos unidades de cada proyecto AEK, para que el supervisor y el becario puedan trabajar en paralelo. Por su parte, el supervisor se encarga de la identificación de las competencias técnicas a trabajar en los tres kits. En total se han identificado doce competencias (por ejemplo, computación matemática, diseño de software y hardware basado en modelos, procesamiento de imágenes, etc.). Por cada una de estas competencias, el supervisor será el encargado principal de desarrollar el material educativo, tanto teórico como práctico. Dicho material será repasado por (1) el becario, (2) otro profesor del área experto en la temática, (3) el coordinador de título y (4) el responsable de MathWorks, con el objetivo de mejorar su calidad. Cada dos semanas, el supervisor se reúne con el responsable de MathWorks para reportarle los resultados, consultar dudas y establecer objetivos comunes. Mensualmente, el supervisor se reúne con el coordinador de título para indicarle el progreso

del proyecto.

4.2. Adaptación de los kits

Los materiales proporcionados por MathWorks requieren de una adaptación para que puedan ser empleados en las actividades académicas del currículo. Para identificar las posibilidades que ofrecen los materiales y adaptarlos a las necesidades del currículo se ha empleado una metodología que consta de cinco pasos. En un primer paso (1) se han identificado las competencias que el fabricante describe en los materiales didácticos proporcionados. Estas competencias están orientadas a las capacidades tecnológicas que pueden desarrollarse con el AEK. Además, cada competencia puede desarrollarse con uno o varios de los proyectos. Es por ello, que en el segundo paso (2), se han relacionado las competencias descritas por el fabricante, con cada uno de los proyectos. En el tercer paso (3), se han identificado las competencias del currículo del grado de Ingeniería Informática que pueden desarrollarse con el AEK. En el cuarto paso (4) se han asociado las competencias del kit con las competencias del currículo académico. Y finalmente, en el quinto paso (5) se ha realizado un diseño detallado de las actividades a desarrollar para adaptar los contenidos de aprendizaje del AEK al currículo. Para cada una de las competencias asociadas en el paso cuarto, se han definido (5.1) los prerrequisitos o conocimiento previos necesarios para desarrollar la competencia, (5.2) los objetivos de aprendizaje de la misma, (5.3) los contenidos a desarrollar y (5.4) el sistema de evaluación.

Los materiales desarrollados con esta metodología permitirán al profesorado configurar las competencias a trabajar en función factores como los conocimientos previos, objetivos y profundidad del aprendizaje deseado.¹ De esta forma, una asignatura que desee trabajar una competencia de conceptos básicos podrá configurar los materiales para desarrollar únicamente aquellos aspectos deseados y se le proporcionarán el resto de módulos ya completados. Sin embargo, otra asignatura que busque profundizar en competencias de forma más avanzada, podrá seleccionar los materiales con menos módulos resueltos, dejando que sean los alumnos quienes los desarrollen.

5. Conclusiones y Líneas Futuras

Este artículo ha presentado dos metodologías de cara a introducir el uso de hardware real en la enseñanza de asignaturas de diseño y control de sistemas ciber-físicos. Para ello, la Universidad de Mondragon y MathWorks han lanzado una estrecha colaboración

que se llevará a cabo durante el presente curso. Una vez terminado el proyecto, los resultados de este serán accesibles para la comunidad académica. Internamente, los resultados del proyecto se aplicarán dentro de la asignatura de Informática Industrial dentro del grado en ingeniería informática, así como en otras asignaturas relacionadas dentro de otros grados y másteres.

Como línea futura, durante el curso 2021-2022, se evaluarán empíricamente los resultados obtenidos del proyecto mediante su aplicación en la asignatura de informática industrial del grado en ingeniería informática de la Universidad de Mondragon. Se medirá para ello cómo de atractivo resulta la experiencia con una encuesta.

Referencias

- [1] Phyllis C Blumenfeld y col. «Motivating project-based learning: Sustaining the doing, supporting the learning». En: *Educational psychologist* 26.3-4 (1991), págs. 369-398.
- [2] Finn Kjærdsdam y Stig Enemark. *The Aalborg experiment project innovation in university education*. Aalborg Universitetsforlag, 1994.
- [3] Peter Marwedel y col. «Survey on education for cyber-physical systems». En: *IEEE Design & Test* 37.6 (2020), págs. 56-70.
- [4] Reza Matinnejad y col. «Test generation and test prioritization for simulink models with dynamic behavior». En: *IEEE Transactions on Software Engineering* 45.9 (2018), págs. 919-944.
- [5] Rafael Molina-Carmona y col. «Convencido del Aprendizaje Basado en Proyectos, ¿por dónde empiezo?». En: *Actas de las JENUi* (2020), págs. 117-124.
- [6] Dimitris Mourtzis y col. «Cyber-physical systems and education 4.0—the teaching factory 4.0 concept». En: *Procedia manufacturing* 23 (2018), págs. 129-134.
- [7] Eduardo Rodríguez-Sandoval, Édgar Mauricio Vargas-Solano y Janeth Luna-Cortés. «Evaluación de la estrategia de aprendizaje basado en proyectos». En: *Educación y educadores* 13.1 (2010), págs. 13-25.
- [8] John A Stankovic, James W Sturges y Jon Eisenberg. «A 21st century cyber-physical systems education». En: *Computer* 50.12 (2017), págs. 82-85.
- [9] Martin Törngren y col. «Strategies and considerations in shaping cyber-physical systems education». En: *ACM SIGBED Review* 14.1 (2017), págs. 53-60.

¹El material puede encontrarse en <https://github.com/MU-MATHWORKS>

Evaluación online sostenible mediante vídeos interactivos. Una experiencia en el ámbito universitario.

Miguel Garcia-Pineda¹, Esther De Ves¹, Xaro Benavent¹, Miguel Arevalillo-Herráez¹, Sandra Roger¹,
Mario Montagud¹, Máximo Cobos¹, Juan Gutierrez-Aguado¹, Jose M. Claver¹ y Mercedes Marqués-Andrés²

¹Departament d'Informàtica. Universitat de València. València, Spain.

²Departamento de Ing. y Ccia. de los Comput. Universitat Jaume I de Castelló. Castelló, Spain.

{migarp, esther.deves, xaro.benavent, miguel.arevalillo, sandra.roger}@uv.es

{mario.montagud, maximo.cobos, juan.gutierrez, jclaver}@uv.es, mmrques@uji.es

Resumen

A lo largo de los últimos años, especialmente en 2020 y 2021, donde el COVID-19 nos ha obligado a pasar a una metodología docente semipresencial o en línea, hemos podido comprobar cómo los recursos multimedia han jugado un factor clave para transmitir conocimiento y ayudar en la formación. Si a estos recursos multimedia se añade además un componente de interactividad, podemos mejorar la atención al alumnado, favorecer el aprendizaje activo, fomentar los mecanismos de regulación del aprendizaje autónomo y trabajar la metacognición del estudiante. Este trabajo presenta una experiencia en la que se aborda la participación activa de los estudiantes universitarios en el propio proceso de evaluación, como medio para desarrollar su capacidad de aprendizaje a lo largo de la vida. Para ello, se ha diseñado y desarrollado una estrategia de evaluación en línea orientada al aprendizaje. Dicha estrategia se fundamenta en el uso de objetos multimedia interactivos y asegura su sostenibilidad a lo largo del tiempo a través de la reutilización de los materiales.

Abstract

Over the last few years, especially in 2020 and 2021, where COVID-19 has forced us to move to a blended or online teaching methodology, we have been able to see how multimedia resources have played a key factor in transmitting knowledge and assisting in training processes. If we add to these multimedia resources the characteristic of interactivity, we can improve the students' attention, favor active learning, promote the mechanisms of regulation of autonomous learning and work on the students' metacognition. This work presents an experience where the active participation of university students in the evaluation process itself is addressed, as a means to develop their capacity for lifelong learning. For this purpose, an online assess-

ment strategy based on learning has been designed and developed. It is based on the use of interactive multimedia objects and ensures sustainability over time through the reuse of materials.

Palabras clave

Evaluación sostenible, online, vídeos interactivos, informática, multimedia

1. Motivación

Antes de la pandemia del COVID-19, más del 85 % de la juventud utilizaba los elementos multimedia como principal fuente para adquirir conocimiento y formarse [1]. Durante la crisis sanitaria y ante la situación de incertidumbre que atravesamos actualmente, el uso de elementos digitales es indispensable en la docencia. Es por ello que el profesorado debemos diseñar y desarrollar nuevos objetos multimedia en línea para mejorar las metodologías docentes. La característica de *interactividad* en el objeto multimedia es la clave para tener éxito en el modelo de enseñanza-aprendizaje, y el hecho de introducir elementos interactivos (preguntas, aclaraciones, enlaces, etc.) en un vídeo permite optimizar el proceso de aprendizaje y hacerlo mucho más efectivo que el simple visionado "tradicional" de un vídeo [2].

Mediante experiencias previas [3], hemos podido observar el alto grado de efectividad y aceptación del uso de recursos multimedia interactivos en el aula. En dicho trabajo, se puso de manifiesto la actitud positiva del estudiantado en cuanto a la utilización de este tipo de recursos, recibiendo una valoración muy positiva con respecto a la contribución de las actividades interactivas en la mejora del proceso de aprendizaje.

A través de este trabajo, se presenta una experiencia donde se aborda la participación activa del alumnado

universitario en el propio proceso de evaluación, siendo ésta esencial para desarrollar su capacidad de aprendizaje a lo largo de la vida. Para ello, se ha diseñado y desarrollado una estrategia de evaluación en línea basada en el aprendizaje, fundamentada en el uso de objetos multimedia interactivos y que asegura la sostenibilidad a lo largo del tiempo a través de la reutilización de los materiales.

En cuanto a su estructura, el artículo comienza analizando la importancia de la característica de interactividad de los objetos multimedia. Se continúa explicando y contextualizando las experiencias realizadas. Después, se muestra la evaluación de la actividad llevada a cabo por el alumnado a través de una encuesta. Finalmente, se presentan las conclusiones derivadas de la investigación realizada.

2. Evaluación con objetos multimedia interactivos

La evaluación orientada al aprendizaje se fundamenta en tres pilares básicos: a) tareas/actividades auténticas a realizar por el alumnado (que sean útiles y orientadas a la profesión a desarrollar); b) retroalimentación al estudiantado; y c) participación de todos los agentes en el proceso de evaluación a través de autoevaluación, heteroevaluación y evaluación por pares [4]. En el contexto de la docencia online o semipresencial, cuanto más cerca se encuentre el profesorado de seguir un enfoque orientado al aprendizaje, más fácil será la planificación y puesta en marcha del proceso de evaluación. Junto a la evaluación orientada al aprendizaje intervienen otros conceptos como son la evaluación sostenible y la evaluación para el empoderamiento. La evaluación sostenible se basa en la aplicación de estrategias colaborativas, tales como la auto-evaluación y la co-evaluación, donde los miembros de los equipos llevan a cabo las tareas de evaluación. Además, ofrece a los estudiantes la confianza necesaria para progresar en su aprendizaje a lo largo de su vida, sin aumentar la carga de trabajo del profesorado. En la evaluación para el empoderamiento, el papel principal lo posee el estudiante, y por tanto el agente de la evaluación ya no reside exclusivamente en la figura del profesor, sino que el alumnado toma un papel activo en dicho proceso. Esto implica que el propio estudiantado debe aprender a evaluar y mejorar por sí mismo sus actuaciones. Como consecuencia de lo anterior, el proceso de evaluación conlleva una gran complejidad, siendo ésta aún mayor si hablamos de la evaluación en línea, ya que se requiere que los procesos de evaluación se distribuyan de forma continuada a lo largo del aprendizaje, en lugar de que solo ocurran en momentos específicos o, peor aún, si solo ocurren al final del proceso

enseñanza-aprendizaje.

En dichas estrategias de evaluación el alumnado tendrá un papel activo, lo que implica aprender a evaluar y mejorar por sí mismos sus actuaciones. Para ello, el diseño de la evaluación tomará como punto de partida los resultados de aprendizaje, seguido del objeto de la evaluación, que, en este caso, será el nivel de aprendizaje adquirido y, más concretamente, la evaluación del grado de adquisición y desarrollo de las competencias. En tercer lugar, se tendrán en cuenta los productos o evidencias, las tareas de evaluación, así como los criterios, técnicas e instrumentos utilizados.

Estos sistemas de evaluación en línea basados en el aprendizaje serán desarrollados mediante vídeos y/o presentaciones de corta duración que irán avanzando y planteando cuestiones o aclaraciones acerca del vídeo visualizado. Este tipo de objetos multimedia interactivos permitirán almacenar las contestaciones de las preguntas realizadas, lo cual podría ser utilizado tanto por el alumnado como por el profesorado como retroalimentación para conocer cómo se van asimilando conocimientos y así contribuir en la mejora continua del aprendizaje [5].

3. Actividad de evaluación interactiva

A continuación, vamos a presentar la actividad de evaluación interactiva realizada en la asignatura de Información Multimedia (IM-UV) en el grado de Ing. Multimedia y en la asignatura de Fundamentos de Redes de Computadores (FRC-UV) en el grado de Ing. Informática, ambos impartidos en la ETSE-UV. Para realizar estos objetos multimedia interactivos se ha utilizado la plataforma de código abierto H5P¹, que permite añadir interacciones superpuestas sobre los vídeos y/o presentaciones tales como imágenes, elaboración de texto, enlaces y preguntas mientras el estudiante visualiza el vídeo. Esta herramienta se encuentra integrada en la plataforma Moodle², permitiendo crear diferentes tipos de recursos interactivos en el propio curso.

Para realizar la actividad propuesta, el alumnado tiene que formar grupos de n personas y elegir un vídeo elaborado por el profesorado sobre el cual plantear cuestiones, ejercicios y otros elementos interactivos dando como resultado un vídeo interactivo. Los vídeos han sido divididos por temas o bloques temáticos, y tratan conceptos vistos en clase. Para automatizar dicho proceso el profesorado puede utilizar la herramienta “auto-selección de grupo” que posee Moodle. A través de esta herramienta se plantean tantos vídeos como grupos se quieren formar según el número

¹<https://h5p.org>

²<https://moodle.org/?lang=es>

de estudiantes y el alumnado se asigna a cada uno de esos grupos/vídeos. En la actividad cada grupo tiene que realizar dos tareas diferentes:

- **Propuesta y creación de preguntas:** a partir del vídeo asignado, los miembros del grupo, después de visualizar el vídeo, deberán pensar un mínimo de 5 preguntas del contenido e incrustarlas en el punto que se crea adecuado dentro de la línea temporal del vídeo.
- **Evaluación de preguntas de otros grupos:** cada grupo, mediante una rúbrica³ proporcionada por el profesorado, deberá evaluar las preguntas propuestas por sus compañeros de otros grupos. Cada grupo evaluará las preguntas realizadas por otros dos grupos (evaluación por pares).

Se trata de una actividad optativa pero evaluable. La nota de la actividad vendrá dada por una media ponderada de las dos tareas anteriores. Es indispensable realizar las dos tareas para que la actividad se considere realizada. La valoración del trabajo será 25 % nota del profesor y 75 % nota de los compañeros (valoración de las preguntas y valoración de la evaluación). La nota obtenida representará un porcentaje (entre el 5 y el 10 %) de la nota final de la asignatura. La entrega de la actividad se realizará a través del moodle mediante un "TALLER". En este taller se deberá entregar el vídeo interactivo en formato .h5p que puede ser descargado desde la cuenta creada en la plataforma H5P.

Estos vídeos cortos con preguntas estarán disponibles en el aulavirtual (ver Figura 1), una vez terminados, como material adicional para poder repasar contenidos de cara a la prueba objetiva de la asignatura. Para fomentar su visualización por parte del alumnado como material de repaso de los contenidos de la asignatura, en el examen habrá un ejercicio tipo test que incluirá algunas cuestiones de las propuestas por los propios estudiantes en los vídeos interactivos creados.

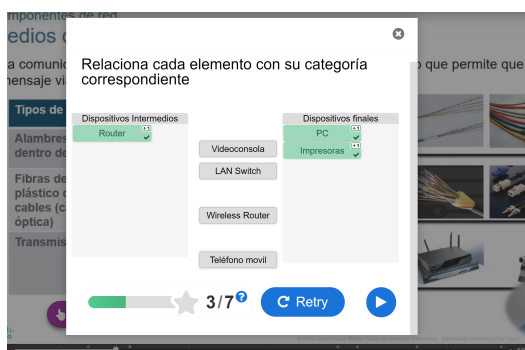


Figura 1: Fotograma de una pregunta incrustada en un vídeo de la asignatura FRC.

³<https://docs.google.com/document/d/1qH0CyUnzSkoYcOWnNs87gMu2EeI52YevDxBAOxMBKK/edit?usp=sharing>

4. Evaluación y resultados

Para evaluar las actividades propuestas en la sección anterior hemos utilizado la encuesta que se muestra en el Cuadro 1. Esta evaluación ha sido contestada por 60 alumnos/as de las dos asignaturas donde se ha llevado a cabo la implantación de la actividad de evaluación con objetos multimedia. A continuación, analizamos las respuestas de manera global con el fin de observar el grado de adecuación y aceptación de estas actividades por parte del alumnado.

Según las respuestas obtenidas de la pregunta 1 (ver Figura 2a), un 64 % del alumnado tenía ya un conocimiento previo del tema asignado, y al realizar la parte de la actividad de inclusión de elementos interactivos ha afianzado el contenido en ese tema o parte del mismo. Además, sumando las respuestas a), b) y c) podemos ver que el 93 % del alumnado considera que dicha actividad le ha servido para reforzar sus conocimientos y aprender aspectos nuevos. En la Figura 2b se observan las respuestas relacionadas con la pregunta 2 de la encuesta. En esta pregunta se trata la fase de evaluación de la actividad, y de manera implícita la metacognición del alumnado. En este caso, un 30 % opina que ha aprendido aspectos nuevos sobre el tema tratado en el vídeo, y un 44 % (sumando respuestas b) y c) de la pregunta 2) considera que la fase de evaluación de la actividad le ha venido bien para afianzar conocimientos sobre el tema visualizado. En la Figura 2c se observa el tiempo requerido por el estudiante para preparar la actividad de pensar y escribir las preguntas de un vídeo. En ella se puede observar que en torno al 71 % del alumnado necesitó entre 1 y 2 horas para desarrollar la actividad de evaluación interactiva, y solo un 3 % requirió de más de 4 horas para llevarla a cabo. Por tanto, podemos indicar que se trata de una actividad grupal, cuyo tiempo de realización no influye significativamente sobre el desarrollo normal de la asignatura ni sobre el progreso de otras materias que el alumno pudiera estar cursando simultáneamente. Finalmente, la cuarta pregunta (ver Figura 2d) aborda la valoración global de la actividad. Un 91 % lo consideró como una experiencia positiva y que le ha sido de ayuda para mejorar el proceso enseñanza-aprendizaje de la asignatura.

5. Conclusiones

En este artículo, hemos presentado una experiencia donde se aborda la participación activa del alumnado universitario en el propio proceso de evaluación, como medio para desarrollar su capacidad de aprendizaje. Para ello, se ha diseñado y desarrollado una estrategia de evaluación en línea fundamentada en el uso de objetos multimedia interactivos, y cuyo objetivo ha

OBJETIVO DE LA ACTIVIDAD
1. La actividad en la que has tenido que crear preguntas para evaluar el vídeo asignado, ¿te ha servido para reforzar los conocimientos que ya tenías sobre ese tema? a) No me ha servido para nada, ha sido una pérdida de tiempo. b) No tenía ni idea del tema, y al pensar y escribir las preguntas he afianzado el contenido. c) Tenía ya un conocimiento previo del tema asignado, y al hacer las preguntas he afianzado el contenido. d) He aprendido aspectos nuevos del tema que desconocía.
2. La actividad de evaluación por pares, en la que habéis evaluado dos vídeos realizados por vuestros compañeros de otras temáticas distintas, ¿te ha servido para fomentar la auto-reflexión (meta-cognición) en el aprendizaje de la asignatura? a) No me ha servido para nada, ha sido una pérdida de tiempo. b) No tenía ni idea del tema del vídeo, y al visualizar los vídeos y contestar las preguntas he afianzado el contenido. c) Tenía ya un conocimiento previo del tema del vídeo, y al hacer las preguntas he afianzado el contenido. d) He aprendido aspectos nuevos del tema tratado en el vídeo que desconocía.
TIEMPO DE DEDICACIÓN
3. ¿Cuántas horas aproximadamente has dedicado para preparar la actividad de pensar y escribir las preguntas? a) 1 hora b) 2 horas c) 3 horas d) 4 horas e) Más de 4 horas
SATISFACCIÓN
4. ¿Qué valoración haces de la actividad con vídeos interactivos propuesta? a) Positiva b) Negativa

Cuadro 1: Encuesta realizada a los alumnos para evaluar la actividad propuesta.

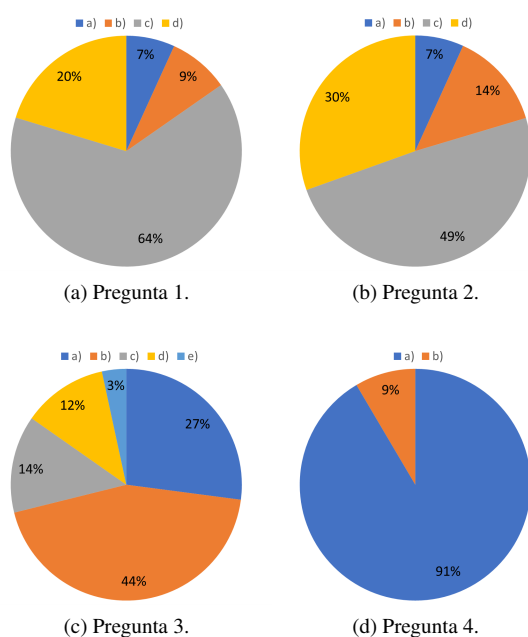


Figura 2: Valoración de la actividad propuesta por parte del alumnado.

sido fomentar el aprendizaje mediante la explicación. En general, hemos observado que la actividad ha sido valorada positivamente por el estudiantado y ha servido como ayuda para aprender y afianzar contenidos de las asignaturas implicadas según un alto porcentaje de alumnado. No obstante, y si bien estos primeros resultados nos animan a continuar con este tipo de desarrollos, aún debemos de mejorar este tipo de actividades, ya que un 9 % del alumnado encuestado considera que este tipo de actividades no les aporta ninguna mejora

en el proceso enseñanza-aprendizaje.

6. Agradecimientos

Este proyecto (SFPIE-PID20-1352382) ha sido financiado por el SFPIE de la Universitat de València.

Referencias

- [1] Carolina Almeida y Pedro Almeida. Online educational videos: The teenagers' preferences. En *Iberoamerican Conference on Applications and Usability of Interactive TV*, pp. 65–76. Springer, 2016.
- [2] Cynthia J Brame. Effective educational videos: Principles and guidelines for maximizing student learning from video content. *CBE—Life Sciences Education*, 15(4), 2016.
- [3] Miguel Garcia-Pineda, Esther De Ves, Sandra Roger, Máximo Cobos, José M. Claver, Xaro Benavent, Miguel Arevalillo-Herráez y Juan Gutierrez-Aguado. Vídeos interactivos para mejorar el proceso enseñanza-aprendizaje en la generación YouTube. En *Actas de las Jenui*, vol. 5, pp. 353–356, 2020.
- [4] David Carless. Learning-oriented assessment: conceptual bases and practical implications. *Innovations in education and teaching international*, vol. 44 (1), pp. 57–66. Routledge, 2007.
- [5] Susana Olmos Miguelañez. Evaluación formativa y sumativa de estudiantes universitarios: aplicación de las tecnologías a la evaluación educativa. Tesis doctoral, Universidad de Salamanca. Ed. Universidad de Salamanca, 2008.

Experiencia colectiva de aplicación de la clase invertida en el grado de informática

Iñigo Aldalur, Miren Illarramendi, Mikel Iturbe, Urtzi Markiegi, Ibai Roman

Departamento de Electrónica e Informática

Mondragon Unibertsitatea

20500 Arrasate-Mondragon

{ialdalur, millarramendi, miturbe, umarkiegi, iroman}@mondragon.edu

Resumen

Con el cambio de los hábitos de consumo de información, la educación superior se encuentra frente al desafío de transformar la experiencia docente de cara a maximizar la implicación y motivación del alumnado. Dicha transformación, basada en contenidos multimedia y en el control continuo de adquisición de conocimientos por parte del alumnado, se ha acelerado por la digitalización forzada provocada por la pandemia Covid19. De los diferentes paradigmas, la clase invertida toma fuerza como una de las alternativas transformadoras más estudiadas y relevantes. Este trabajo presenta los resultados preliminares de un caso de estudio de una implantación del aula invertida a escala -en doce asignaturas de diferentes cursos- en un grado de ingeniería informática durante el curso 2020/21. Se expone la principal motivación detrás de la transformación, la capacitación de recursos y formación del profesorado, la metodología docente aplicada (centrada en el contenido multimedia y el control de los conceptos adquiridos) y la evaluación de los resultados obtenidos. Para la evaluación, se han realizado encuestas específicamente diseñadas al profesorado y alumnado involucrado.

Abstract

As information consumption habits change, higher education is facing the challenge of transforming the teaching experience in order to maximize student involvement and motivation. This transformation, based on multimedia content and the continuous control of knowledge acquisition by students, has been accelerated by the forced digitalisation caused by the Covid19 pandemic. Among the different paradigms, the inverted class is one of the most studied and relevant transformation alternatives. This work presents the preliminary results of a case study of an implementation at scale of the inverted classroom -in twelve subjects from different academic courses- in a computer engi-

neering degree during the 2020/21 academic year. The main motivation behind the transformation is presented, how the resources and training of the teaching staff have been enabled, the teaching methodology applied (focused on multimedia content and control of the concepts acquired) and the evaluation of the obtained results. For the evaluation, surveys specifically designed for the involved teaching staff and students have been carried out.

Palabras clave

Clase invertida, metodologías activas, grado informática.

1. Introducción

En las últimas décadas el ámbito universitario está afrontando cambios transcendentales derivados principalmente de las innovaciones tecnológicas y pedagógicas. Las nuevas tecnologías proporcionan mecanismos para multiplicar las oportunidades de comunicación y colaboración durante el proceso de aprendizaje, extendiendo el contexto tradicional del aula a un espacio digital. Sin embargo, estos avances tecnológicos se viven de forma dispar entre las comunidades del profesorado y alumnado. Por un lado, el alumnado que se incorpora a la universidad demanda un mayor empleo de la tecnología en los procesos de aprendizaje. Esta demanda, caracteriza a una generación cuyo patrón de consumo de información está marcado por el uso continuado de la tecnología (p.ej., son capaces de utilizar durante horas sus teléfonos inteligentes, tabletas u ordenadores). Esta capacidad para estar continuamente conectados a través de los dispositivos digitales, sin embargo, rivaliza con la capacidad para concentrarse y aprender. Por otro lado, la comunidad docente, consciente de las oportunidades y los retos, está involucrada en una progresiva transformación para incorporar la tecnología en la educación de forma racional.

Los avances en la neurociencia apuntan a la motivación e implicación del alumnado como un factor determinante en su proceso de aprendizaje. En las últimas décadas, la comunidad docente ha realizado propuestas pedagógicas innovadoras que buscan la participación del alumnado mediante las metodologías activas como marco para aumentar esta motivación e implicación. Una de las metodologías más relevantes es la clase invertida. Este modelo pedagógico consiste en que el alumnado estudie y prepare las lecciones fuera de clase, accediendo en casa a los contenidos de las asignaturas mediante el uso de las tecnologías. Posteriormente, con el profesorado como guía, será en el aula donde desarrollen los contenidos prácticos, interactúen y realicen actividades más participativas (analizar ideas, debates, trabajos en grupo, etc).

Con la irrupción de la pandemia del Covid19, los docentes se han visto obligados a una educación remota de emergencia durante el segundo semestre del curso pasado. Pero no todo ha terminado ahí, las clases universitarias han tomado el camino de la semipresencialidad, compaginando la impartición de clases de manera presencial y remota. Además, la pandemia ha intensificado el uso de las tecnologías así como la adopción de metodologías pedagógicas innovadoras por parte de los centros educativos superiores. En este contexto, en nuestra universidad hemos decidido implantar el modelo pedagógico de clase invertida en el grado de ingeniería informática.

En el resto del documento se presenta la metodología de trabajo llevada a cabo, así como los resultados preliminares de la implantación junto con las principales conclusiones.

2. Caso de estudio

El punto de partida del proyecto se da durante el despliegue del plan estratégico de la universidad en el grado de ingeniería informática realizado en mayo del 2020. Entre los objetivos establecidos, se define una propuesta de proyecto de implantación del modelo pedagógico de clase invertida como mecanismo para (1) aumentar la implicación y motivación del alumnado en clase y (2) capacitación del profesorado en la digitalización de contenidos. El proyecto se presenta en la coordinación académica y recibe el visto bueno y recursos para su implantación en el curso 2020-2021. El proyecto supone la implantación a escala del modelo pedagógico de la clase invertida en dos asignaturas por curso y semestre, esto es, un total de 12 asignaturas de los tres primeros cursos del grado, con la involucración de 20 profesores y 253 alumnos. Antes de llevar a cabo la experiencia, recopilamos información de otros trabajos para ver cómo podríamos adaptarlo a nuestro propio caso. El Cuadro 1 muestra un resumen de algu-

nas de las experiencias evaluadas.

Aunque la universidad llevaba varios años proporcionando píldoras formativas al equipo docente en metodologías activas (incluidas formaciones en el modelo pedagógico de clase invertida), se diseña una formación específica para el proyecto que se desarrolla en el mes de julio del 2020. Durante la formación el equipo docente del proyecto fue formado en los conceptos básicos del modelo pedagógico de clase invertida. Y la formación se completó con tres talleres: un primer taller para establecer las normas de la metodología que todas las asignaturas debían seguir, un segundo taller de diseño de la implantación de la metodología en cada una de las asignaturas implicadas y un último taller práctico para iniciarse en el manejo autónomo del estudio de grabación. Señalar que previamente al periodo de formación se realizó el diseño y gestiones para instalar el estudio de grabación.

Todas las asignaturas han adoptado 4 normas establecidas en la metodología común. (1) Fuera de las horas lectivas, el alumnado visualiza un(os) vídeo(s) con los contenidos de la asignatura y (2) en la siguiente sesión de clase presencial realizan un breve cuestionario relacionado con el vídeo que han visto. De esta forma, se apremia al alumnado a que vengan a clase con los conceptos básicos ya revisados y con la posibilidad de aclarar las dudas que puedan tener del material audiovisual. Una vez cerrado el ciclo de la parte más teórica y conceptual, (3) todas las asignaturas también se han apoyado en ejercicios y prácticas. Esta actividad se ha llevado a cabo en clase (pudiendo terminar fuera de clase) y así se ha aprovechado el tiempo presencial para realmente ir resolviendo dudas en la aplicación de los conceptos teóricos. El objetivo perseguido ha sido aprovechar al máximo el tiempo de las clases presenciales para reforzar el proceso de aprendizaje. Después de cada tema o módulo de aprendizaje, (4) el profesorado ha proporcionado *feedback* al alumnado de los cuestionarios realizados. Sin embargo, la evaluación aplicada en cada asignatura no se ha visto afectada respecto a cursos anteriores con la incorporación de la metodología. En todo el grado se aplica la evaluación continua que cada asignatura implementa con diferentes tipos de pruebas: cuestionarios, prácticas (individuales o en grupo) y exámenes individuales sobre la materia.

3. Valoración de la experiencia

Los resultados presentados en este trabajo corresponden a la implantación de la metodología en el primer semestre del curso 2020-2021. En este periodo se han involucrado 6 asignaturas, 10 profesores y profesoras y 253 alumnos y alumnas. Al no disponer datos de la evaluación en el momento de presentar este traba-

Ref.	Nivel	Actividad en clase	Actividad fuera de clase	Número estudiantes
[5]	Programación I en Grado de diseño	Explicación de las soluciones publicadas. El resto de la clase se dedica a la resolución de ejercicios de programación de mayor complejidad	Lectura de material del libro y resolución de un pequeño número (2 o 3) de ejercicios básicos	19
[7]	Segundo curso del grado de Ingeniería del Software.	Para saber si los conceptos vistos en los vídeos habían quedado claros, usan la plataforma de gamificación Kahoot para realizar pruebas interactivas con los alumnos al comienzo de la clase. De acuerdo a las respuestas obtenidas, el profesor decidía qué conceptos debía aclarar. Después ejercicios.	Videos que los alumnos debían visualizar antes de asistir a clase	434 (dos cursos) 225 primero sin FL 209 segundo con FL
[6]	Ingeniería del Software del Grado de Ingeniería Telemática	Para comprobar que los estudiantes habían adquirido los conocimientos esperados, al inicio de la clase se realizaba un test. Para la realización de dichos test se usaba la plataforma Socrative. Una vez finalizado el test, se pasaba a resolver los ejercicios correspondientes que, esta vez, debían entregar a través de Aula Virtual.	Cada día se les indicaba a los estudiantes qué temas y qué vídeos debían revisar para la siguiente sesión teórico-práctica	Entre 30 y 60
[1]	Programación de dos titulaciones de grado: Matemáticas e Ingeniería Electrónica y de telecomunicaciones	Se hacía un test previo a la clase, la clase, y un taller de ejercicios evaluados por pares al finalizar cada tema.	NO SE ESPECIFICA	50 por curso

Cuadro 1: Trabajos previos del modelo pedagógico de clase invertida analizados.

jo no se puede valorar el rendimiento académico. Estos datos serán presentados más adelante. Para la evaluación del caso de estudio, se ha analizado la acogida que ha tenido la experiencia. Para ello, se han realizado dos encuestas diferenciadas al alumnado y profesorado.

Siguiendo el trabajo realizado por [2], hemos encuestado al alumnado para conocer su opinión sobre esta nueva iniciativa en diez diferentes aspectos (véase el Cuadro 2). Se han modificado ligeramente las preguntas utilizadas en [2], y las respuestas han sido recogidas basándose en la escala Likert [4] como en el trabajo original. En la Figura 1 se muestra un resumen de las respuestas (253 cuestionarios en total) por parte del alumnado.

Preguntas	
P1	Por lo general, veía los videos antes de la clase
P2	Los videos fueron fáciles de entender/seguir
P3	El material de los videos fue diseñado correctamente, bien estructurados y definidos claramente
P4	Los videos me han ayudado a aprender
P5	Las actividades desarrolladas en clase después del cuestionario me han ayudado a aprender
P6	La posibilidad de volver a ver y rebobinar los videos me ha ayudado a aprender
P7	Un formato de vídeo breve que presenta los principales temas de estudio me han ayudado a aprender, más que los vídeos muy detallados y extensos
P8	La realización de cuestionarios con múltiples opciones después de ver los videos ha permitido profundizar en los contenidos más complejos antes de la clase y, por lo tanto, ha ayudado a comprenderlos mejor
P9	El haber visto los videos y revisado los materiales proporcionados antes de las sesiones de clase me ha ayudado a completar las actividades de clase con más confianza ya que no me encontraba perdido
P10	El haber visto los videos y revisado los materiales proporcionados antes de las sesiones de clase me ha ayudado a completar las actividades de clase con más facilidad porque las actividades me resultaban familiares

Cuadro 2: Encuesta dirigida al alumnado.

En general, el alumnado se ha mostrado de acuerdo con las afirmaciones que se realizaban en la encuesta

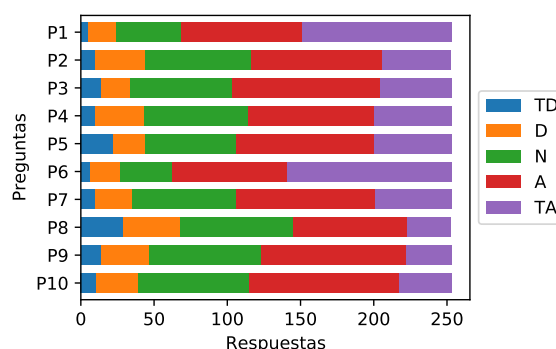


Figura 1: Resultados de la encuesta al alumnado (TD: Totalmente en desacuerdo, D: En desacuerdo, N: Ni de acuerdo ni en desacuerdo, A: De acuerdo, TA: Totalmente de acuerdo).

ta y la acogida de la experiencia ha sido positiva. Cabe destacar las afirmaciones P1 y P6, donde las respuestas han sido más favorables. Se afirma que los vídeos son visionados con anterioridad a las clases, y que un aspecto a valorar de los medios audiovisuales es la posibilidad de revisionado. Por otro lado, la realización de cuestionarios no ha recibido el mismo consenso (P8).

Para valorar la acogida de la experiencia por parte del profesorado, se ha seguido una metodología similar al del alumnado, centrándose en la realización de un cuestionario. Las preguntas realizadas se muestran en el Cuadro 3 y han sido definidas siguiendo el trabajo realizado por [3]. Las respuestas del profesorado (10 en total) se han resumido en la Figura 2. En general, la acogida a la experiencia ha sido buena y agradable (P1 y P2), si bien se identifican dificultades para seguir adelante con ella (P8, P9 y P10). Los medios proporcionados por la universidad se han identificado como adecuados para poder afrontar la experiencia (P6 y P7).

Preguntas	
P1	Disfruto tratando de utilizar la enseñanza invertida
P2	Disfruto con el método de flipped teaching que es completamente nuevo para mí
P3	Rara vez pienso en el reconocimiento que puedo obtener por hacer flipped teaching de los alumnos
P4	Me preocupa el mecanismo de reconocimiento que existe por parte de la comunidad de flipped teaching
P5	Podría completar la tarea de enseñanza invertida si tuviera mucho tiempo para ejecutar la enseñanza invertida
P6	La universidad proporciona instalaciones y recursos para la enseñanza invertida
P7	La universidad proporciona recursos tecnológicos y de software para la enseñanza invertida
P8	Tengo la intención de seguir utilizando la enseñanza invertida
P9	Tengo la intención de seguir utilizando la enseñanza invertida en lugar de utilizar únicamente la enseñanza tradicional
P10	Si pudiera, me gustaría seguir utilizando la enseñanza invertida

Cuadro 3: Preguntas dirigidas al profesorado [3].

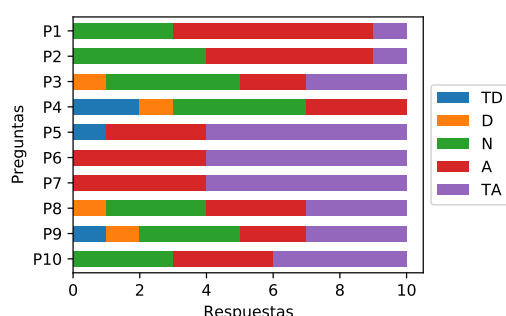


Figura 2: Resultados de la encuesta al profesorado (TD: Totalmente en desacuerdo, D: En desacuerdo, N: Ni de acuerdo ni en desacuerdo, A: De acuerdo, TA: Totalmente de acuerdo).

4. Conclusiones

En este trabajo presentamos los resultados preliminares de una experiencia de clase invertida realizada en el grado de ingeniería informática. La motivación para esta experiencia se basa por un lado en el cambio de los hábitos de consumo de información de las nuevas generaciones así como por la oportunidad identificada por el profesorado para evolucionar metodológicamente a raíz lo vivido durante la enseñanza de emergencia del curso 2019-2020 derivada por la pandemia.

Es una experiencia a escala que ha involucrado en el primer semestre del curso 2020-2021 a seis asignaturas de los tres primeros cursos del grado de informática, esto es, a 10 profesores y profesoras y 253 alumnos y alumnas. Durante el transcurso de este primer semestre se han creado 164 vídeos y diseñado 38 cuestionarios.

Para la evaluación, se han llevado a cabo sendas encuestas al alumnado y profesorado. De la encuesta del alumnado podemos concluir que en línea con otros trabajos similares, la motivación del alumnado se ve incrementada con el uso de este modelo de clase inver-

tida. Por parte del profesorado, se constata que el esfuerzo realizado por los docentes es elevado aunque se considera necesario. En base a estos resultados iniciales animamos a la comunidad docente del ámbito de la ingeniería informática a que adopten la metodología.

Durante el segundo semestre, se llevará a cabo el resto de la iniciativa con otras seis asignaturas implicadas, se recopilarán los datos de encuestas correspondientes al segundo semestre y podrá realizarse una valoración del impacto de la actividad en el rendimiento académico del alumnado.

Agradecimientos

Este trabajo se ha desarrollado gracias al apoyo del Gobierno Vasco a los planes de Mondragon Unibertsitatea en los ejes de desarrollo del PSU 2019-2022.

Referencias

- [1] Xaro Benavent y col. «Clase invertida en asignaturas de programación usando la plataforma de e-learning Moodle». En: *Actas de las JENUI 5* (2020), págs. 329-332.
- [2] Jin Su Jeong, David González-Gómez y Florentina Cañada-Cañada. «Students' Perceptions and Emotions Toward Learning in a Flipped General Science Classroom». en. En: *J Sci Educ Technol* 25.5 (oct. de 2016), págs. 747-758. ISSN: 1573-1839.
- [3] Hui-Min Lai, Yu-Lin Hsiao y Pi-Jung Hsieh. «The role of motivation, ability, and opportunity in university teachers' continuance use intention for flipped teaching». en. En: *Computers & Education* 124 (sep. de 2018), págs. 37-50. ISSN: 0360-1315.
- [4] R. Likert. «A technique for the measurement of attitudes». En: *Archives of Psychology* 22 140 (1932), págs. 55-55.
- [5] Ángeles López y Mar Marcos. «Una experiencia de clase invertida en la enseñanza de la programación». En: *Actas de las JENUI 3* (2018), págs. 47-54.
- [6] Silvia Rueda Pascual y col. «De la Clase Tradicional a la Clase Invertida: Aplicación Práctica en Ingeniería del Software». En: *Actas de las JENUI 3* (2018), págs. 119-126.
- [7] Javier Troya y col. «Invirtiendo las clases de laboratorio en Ingeniería Informática: Un enfoque ágil». En: *Actas de las JENUI 4* (2019), págs. 15-22.

FreeWheeling Travelers: viajando en SQL

Jorge Silvestre, Miguel A. Martínez-Prieto

Departamento de Informática

Universidad de Valladolid

jsilvestre@infor.uva.es, migumar2@infor.uva.es

Resumen

Freewheeling Travelers (FWT) es una actividad gamificada que permite a los estudiantes practicar sus conocimientos de SQL de forma participativa y amena. Para ello, FWT convierte a los estudiantes en turistas y les ofrece la posibilidad de vivir situaciones cotidianas de un viajero, que abordarán utilizando SQL. En este artículo describimos la prueba piloto de FWT, en la que se constata un efecto positivo en la motivación y en el rendimiento académico del alumnado de una asignatura de Sistemas de Bases de Datos.

Abstract

Freewheeling Travelers is a gamified learning activity that aims to help students to practice SQL in an interactive and entertaining manner. FWT turns students into tourists and offers them everyday situations of a traveller, which they will experience through SQL. This paper describes the first experience of using FWT, reporting improved academic performance and motivation of students in a Database Systems subject.

Palabras clave

SQL, gamificación, aprendizaje activo.

1. Introducción

Uno de los principales retos en la educación es lograr que los estudiantes sientan interés por el conocimiento que deben adquirir y que se impliquen en las actividades orientadas a proporcionárselo. La dificultad de la materia, su nivel de abstracción o la falta de relación con lo que el estudiante conoce, o espera aprender, pueden llevarle a afrontar las actividades formativas como un obstáculo a superar y no como un medio hacia su aprendizaje. Por ello, es necesario plantear dinámicas, como la *gamificación*, que motiven al alumnado a afrontar su proceso de aprendizaje de una forma

más consciente y constructiva. La *gamificación* introduce elementos de juego en actividades que no son juegos [2], ayudando a construir entornos de aprendizaje cercanos a los alumnos, contribuyendo a incrementar su receptividad para aprender y motivando su participación en el proceso de aprendizaje [3].

En este artículo describimos una propuesta de gamificación del aprendizaje de SQL, en el ámbito de la asignatura Sistemas de Bases de Datos (SBD), en el Grado de Ingeniería Informática de Servicios y Aplicaciones (Universidad de Valladolid): *FreeWheeling Travelers* (FWT). Nuestro objetivo es que los alumnos aprendan SQL de forma gamificada, mientras “hacen turismo” interactuando con una base de datos relacional. Los resultados obtenidos muestran una mejora notable en la adquisición de competencias relacionadas con SQL, además de una valoración positiva por parte de los alumnos que han participado en esta prueba piloto, realizada durante el curso 2020-2021.

La sección 2 introduce el concepto de gamificación del aprendizaje. Las secciones 3 y 4 plantean, respectivamente, el diseño de FWT y la forma en la que se ha integrado en la asignatura. La sección 5 analiza el rendimiento académico, la motivación y la experiencia del alumnado en la prueba piloto de FWT. Finalmente, la sección 6 plantea una breve conclusión sobre esta experiencia y describe nuestras líneas de trabajo futuro.

2. Gamificación en SQL

La gamificación favorece la motivación de los alumnos y puede mejorar la calidad de su aprendizaje [1]. Los elementos de juego influyen de diferentes maneras sobre estas dimensiones [2, 4], por lo que su elección es determinante para alcanzar los objetivos planteados al gamificar una actividad de aprendizaje [2].

Herramientas como MonstER Park [5] o SQL Island (sql-island.informatik.uni-kl.de) ofrecen propuestas de gamificación destinadas al aprendizaje de conceptos básicos de bases de datos, como su diseño conceptual o su consulta con SQL, sobre una plataforma web. En [6] se estudia el efecto de incorporar elementos de juego a una herramienta de

aprendizaje tradicional, SQL-Tutor, con resultados positivos. La gamificación también puede aplicarse a la consolidación del aprendizaje, en lugar de a su adquisición. SchemaVerse (schemaverse.com) o HackerRank (hackerrank.com) son ejemplos de herramientas que asumen que los jugadores tienen un conocimiento previo y le facilitan ponerlo en práctica a través de la resolución de diferentes supuestos, definidos en el marco de una narrativa.

3. FreeWheeling Travelers

En cursos anteriores se observó que los estudiantes presentaban dificultades para alcanzar los objetivos de aprendizaje relacionados con SQL. Estos objetivos se introducían a través de pequeños supuestos prácticos, que apenas unos pocos alumnos resolvían, al ser una actividad de trabajo individual y no entregable. En consecuencia, la primera experiencia práctica de los alumnos con SQL se llevaba a cabo en la actividad final de aprendizaje basado en proyectos, de mayor complejidad y gran influencia en la calificación final (40 %), lo que impedía realizar un aprendizaje adecuado.

FreeWheeling Travelers surge como una alternativa a las actividades tradicionales de resolución de supuestos prácticos y las lleva a un entorno gamificado con dos objetivos principales: (i) reducir la abstracción de los contenidos de SQL y dotarlos de mayor interés, ejemplificando su utilidad práctica en un contexto realista; y (ii) mejorar la motivación y participación de los alumnos para afrontar una parte de la asignatura que, tradicionalmente, les resultaba más difícil. Para ello, FWT introduce los siguientes elementos.

Narrativa. Se plantea una historia atractiva para los estudiantes, que se desarrolla en un contexto familiar para ellos: cada estudiante inicia un viaje turístico por la región de Castilla y León, debido a las restricciones en los desplazamientos a causa de la COVID-19. El estudiante va pasando por varios municipios, en los que afronta necesidades habituales del turista (buscar alojamiento, hacer reservas en restaurantes, reseñar un punto de interés) utilizando *Freewheeling Travelers*, una aplicación de viajes ficticia. Esta aplicación está en una fase *alpha* de desarrollo, en la que únicamente está operativa su base de datos, por lo que el acceso a todos sus servicios debe hacerse directamente en SQL. El viaje progresa de acuerdo a las decisiones del estudiante, que puede elegir qué municipios visitar o qué acciones realizar en su estancia.

Niveles. La actividad se divide en viajes y etapas, abiertas por tiempo limitado, para adecuar su progresión al avance de la asignatura y ajustar la dificultad de las cuestiones planteadas. No se impusieron requisitos al progreso en los niveles para evitar la aparición de

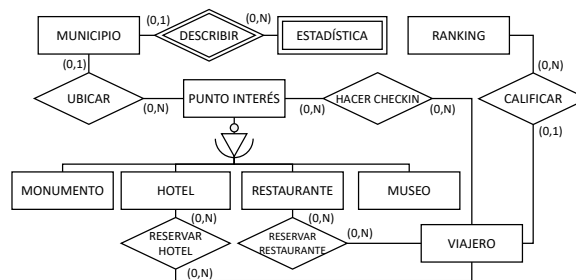


Figura 1: Diseño conceptual de FWT.

bloqueos y la consiguiente frustración del alumnado.

Retroalimentación. Los alumnos reciben una calificación al final de cada viaje de acuerdo con su desempeño, con el objetivo de que sean conscientes de sus avances y puedan identificar sus carencias.

Clasificación. Los resultados de cada ejercicio se publican en la propia base de datos, siendo accesibles para todos los participantes.

3.1. Base de Datos

La base de datos de FWT contiene información sobre conceptos pertenecientes al ámbito turístico. Como se observa en la Figura 1, el *punto de interés* (POI) es el concepto central de la base de datos, en el que se generalizan las características comunes de *monumentos*, *hoteles*, *restaurantes* y *museos*. Cada POI está relacionado con el *municipio* en el que se ubica, del que se recoge información *estadística* sobre su historial de visitantes en los últimos años. La base de datos se pobló utilizando datos del Portal de Datos Abiertos de la Junta de Castilla y León (datosabiertos.jcyl.es). También se incluyó los concepto *reserva*, en *hoteles* y *restaurantes*, y *check-in*, para registrar las actividades de los participantes en sus desplazamientos. Finalmente, la base de datos incluye las entidades VIAJERO y RANKING, que describen a los alumnos participantes y sus puntuaciones.

La base de datos se desplegó en un servidor centralizado, al que se accedía usando *phpMyAdmin*, con el fin de propiciar la interacción del alumnado con la información generada por otros compañeros, potenciando el aspecto social de FWT como elemento motivador.

4. Desarrollo de la actividad

FWT se planteó como una actividad complementaria a las sesiones magistrales impartidas en la asignatura. Así, el inicio de cada viaje estaba precedido por una primera clase de “explicación teórica” de contenidos sobre SQL, y una segunda en la que el profesor ejemplificaba su uso mediante la resolución de pequeños

supuestos prácticos. No obstante, FWT se utilizó puntualmente para introducir algunas funciones o cláusulas de SQL (como LIMIT y OFFSET) no tratadas previamente. En estos casos, la cuestión añadía una “Pista” explicativa y un enlace a información adicional.

4.1. Estructura y organización temporal

La impartición de los contenidos sobre SQL se abordó en las últimas 3,5 semanas de la asignatura, por lo que planteamos 3 viajes, alineados con los contenidos desarrollados cada semana. La actividad finalizó el día previo a la realización del examen de SQL. Los alumnos disponían de 5 días para realizar cada viaje.

El *Viaje 1* planteaba una única etapa con cuestiones destinadas a modificar la base de datos: registrar un viajero; crear o cancelar reservas; y realizar *check-ins*.

El *Viaje 2* constaba de dos etapas: la primera planteaba cuestiones para resolver con consultas simples (SELECT...WHERE) y la segunda añadía el uso de agrupaciones y subconsultas. Además, se incluyeron cuestiones que requerían modificar la base de datos.

El *Viaje 3* constaba de dos etapas: la primera se centraba en el uso de JOIN y UNION, mientras que la segunda planteaba cuestiones más complejas, que requerían usar todos los conceptos aprendidos, para que los alumnos pudiesen autoevaluarse antes del examen.

4.2. Implementación

La implementación actual de FWT se ha realizado usando Moodle. Cada una de las etapas del juego es una *Lección* independiente, compuesta por varias ramificaciones y diferentes páginas. De esta forma, cada etapa presenta diferentes itinerarios de viaje con los supuestos relativos a los destinos elegidos.

Las páginas desarrollan la narrativa de la etapa o contienen cuestiones sobre ella. En estas últimas, el alumno debe enviar la consulta SQL que satisfaga la necesidad de información planteada, y, a continuación, responder una pregunta (de selección múltiple, respuesta corta o emparejamiento de opciones) relacionada con el resultado de la consulta. Esta pregunta de confirmación permite al alumno validar inmediatamente el resultado de su consulta. Cabe destacar que los alumnos solo disponían de un intento para responder a cada pregunta, aunque podían evaluar previamente su respuesta en la base de datos de FWT, sin limitaciones.

Los errores más comunes se abordaron en una clase final de revisión, en la que el profesor explicó las respuestas correctas y facilitó el debate sobre los errores detectados. Además, se realizaron tutorías individuales con aquellos participantes que lo así lo requirieron. Tras cada etapa, el profesor actualizaba la tabla RANKING de acuerdo con los resultados obtenidos. La nota

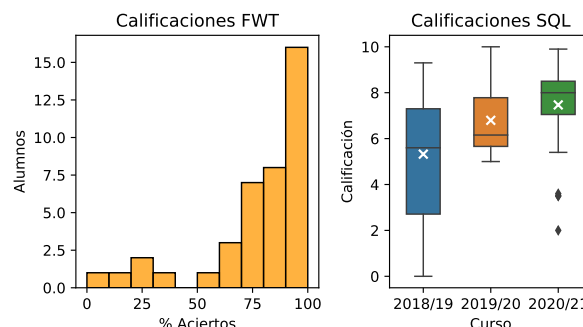


Figura 2: Resultados académicos.

de la actividad se tuvo en cuenta en la calificación final de la asignatura, con un peso de 0,5 puntos.

5. Resultados

La prueba piloto de FWT se ha realizado en el primer cuatrimestre del curso 2020-2021, sobre un grupo de 43 alumnos. De ellos, solo 3 no participaron en la actividad (por abandono de la asignatura). Esta tasa de participación del 93 % es el primer resultado positivo, ya que en los cursos previos apenas la mitad de los alumnos matriculados completaban en tiempo y forma los supuestos propuestos para practicar SQL.

A continuación se analizan los resultados académicos derivados de la incorporación del juego al proceso de enseñanza-aprendizaje de SBD y la valoración que hacen los alumnos de su impacto en la asignatura.

5.1. Resultados académicos

Como puede observarse en el histograma de la Figura 2 (izq.), la mayoría de los alumnos obtuvo más del 50 % de los puntos disputados en el juego, y 16 alumnos obtuvieron el 90 % o más. Esta tendencia positiva se mantuvo en las pruebas de SQL, que fueron superadas por el 83 % de los alumnos con una nota media final de 7,5 puntos, lo que supone una mejora de 0,7 y 2 puntos respecto a los dos cursos anteriores (Figura 2 (der.)). Por otro lado, destaca la disminución de la tasa de abandono (alumnos no presentados): de 18 % en el curso 2019-2020, a 7 % en el 2020-2021. Es decir, más estudiantes se vieron preparados para afrontar la prueba, en vez de desistir de presentarse, lo que sugiere un mayor trabajo en los contenidos en el curso 2020-2021.

5.2. Valoración de los alumnos

Al finalizar la asignatura, planteamos una encuesta voluntaria que incluía 17 preguntas sobre FWT, 14 de ellas de tipo *Likert* con 5 elementos, y 3 de texto libre. La encuesta fue completada por 25 de los 40 alumnos.

La valoración general del juego fue de 4,04 puntos, aunque esta nota se incrementó hasta los 4,4 puntos cuando los alumnos compararon la efectividad de este tipo de actividad gamificada frente a los mecanismos tradicionales de resolución de supuestos prácticos “sin ningún hilo argumental”. Asimismo, valoraron con 4,04 puntos la utilidad de FWT para consolidar los conceptos de SQL impartidos en las clases magistrales y con 4,12 puntos su contribución para superar los criterios de aceptación establecidos en el examen. Otros aspectos de la actividad también fueron bien valorados: la posibilidad de elegir entre diferentes itinerarios de viaje (4,28 sobre 5), la interacción con las respuestas del otros participantes (4,28 sobre 5) o la existencia de un ranking público como elemento motivador (22 se manifestaron a favor, y 3 fueron “indiferentes”).

La planificación temporal, la carga de trabajo y la retroalimentación fueron los aspectos peor valorados del juego. En el primer caso, los alumnos valoraron con 3,2 puntos el calendario de la actividad y manifestaron que el tiempo que transcurría entre cada viaje era escaso. Respecto a la carga de trabajo, 15 alumnos la consideraron adecuada y 10 excesiva. Cabe destacar que FWT se realizó en las últimas semanas de la asignatura, coincidiendo con el final del cuatrimestre, donde la carga de trabajo suele aumentar en términos globales. Finalmente, la retroalimentación ha sido uno de los aspectos peor valorados, en comparación con el *feedback* que reciben en otras actividades de la asignatura. Hasta 17 alumnos han solicitado “disponer de los supuestos y las soluciones al final de cada viaje”, y otros 8 han pedido que “se muestre el resultado de la consulta o la sentencia correcta, una vez enviada la respuesta”.

El análisis de las respuestas textuales confirma el impacto positivo de FWT en los alumnos, y valoran el interés y la utilidad del juego para aprender SQL, y su efecto motivador. Sirva como ejemplo la siguiente respuesta: “esta actividad me ha resultado muy interesante, he aprendido mucho SQL y he entendido de forma práctica por qué es necesaria una base de datos. Estaba tan bien redactado que sentía como si estuviera viajando de verdad; conseguía abstraerme por completo de la realidad. Además, me he reído mucho leyendo las tuplas introducidas por mis compañeros”.

6. Conclusiones y trabajo futuro

En este artículo hemos descrito nuestra experiencia con FWT, una actividad gamificada que ayuda a los alumnos a practicar sus conocimientos de SQL de forma distendida. Tras analizar los resultados y las opiniones recabadas, hemos comprobado la utilidad de esta iniciativas para incrementar la motivación del alumnado a la hora de abordar el aprendizaje de contenidos que, tradicionalmente, resultan más desafiantes.

Esta prueba piloto nos ha ayudado a detectar oportunidades de mejora, como establecer un mecanismo de retroalimentación más efectivo e inmediato, que ayude a los alumnos a conocer el alcance de su aprendizaje e identificar sus carencias, resultando en un conocimiento más completo y satisfactorio. También es necesario introducir un aspecto fundamental en entornos gamificados y de aprendizaje: permitir el error. En esta experiencia, los alumnos solo disponían de un intento, lo que, de acuerdo a las opiniones manifestadas en la encuesta, generaba cierta ansiedad. Para abordar ambos objetivos, optaremos por reemplazar Moodle por una plataforma más apropiada.

Agradecimientos

Este trabajo ha sido financiado por el *Proyecto de Innovación Docente* No. 90, del plan PID-2020/2021 de la Universidad de Valladolid (UVA). El primer autor disfruta de un contrato predoctoral de la Universidad de Valladolid, cofinanciado por el Banco de Santander.

Referencias

- [1] Miguel Ehécatl Morales-Trujillo y Gabriel Alberto García-Mireles. «Gamification and SQL: An Empirical Study on Student Performance in a Database Course». En: *ACM Transactions on Computing Education* 21.1 (2020), págs. 1-29.
- [2] Fiona Fui-Hoon Nah, Qing Zeng, Venkata R. Tejaprolu, Abhishek P. Ayyappa y Brenda Eschenbrenner. «Gamification of Education: A Review of Literature». En: *International Conference on HCI in Business*. 2014, págs. 401-409.
- [3] Alma María Pisabarro y Carlos Enrique Vivarcho. «Gamificación en el aula: gincana de programación». En: *ReVision* 11 (2018), págs. 85-93.
- [4] Mohd H.A. Rahman, Ismail Y. Panessai, Anida Z.M. Noor y Nor S.M. Salleh. «Gamification Elements and Their Impacts On Teaching and Learning». En: *International Journal of Multimedia & Its Applications* 10 (2018), págs. 1-10.
- [5] Johannes Schildgen. «MonstER Park: The Entity-Relationship Diagram Learning Game». En: *ER Forum, Demo and Posters*. Vol. 2716. CEUR Workshop Proceedings. 2020, págs. 150-157.
- [6] Faiza Tahir, Antonija Mitrovic y Valerie Sotardi. «Investigating the Effects of Gamifying SQL-Tutor». En: *International Conference on Computers in Education*. Vol. 1. Nov. de 2020, págs. 416-425.

Iniciativa basada en Kahoot para motivar a los alumnos de Arquitectura de Computadores

Gloria Ortega, Ester M. Garzón, Leocadio G. Casado, Pilar M. Ortigosa,
Vicente González, Juan Francisco Sanjuan, Juana L. Redondo, Francisco J. Orts,
Juan José Moreno, Luis O. López, Nicolás C. Cruz, Miriam Ruiz

Departamento de Informática
Universidad de Almería

{gloriaortega, gmartin, leo, ortigosa, vruiz, jsanjuan, jlredondo, francisco.orts,
juanjomoreno, lol766, ncalvocruz, mrferrandez}@ual.es

Resumen

Debido al auge de la formación universitaria en remoto, es común que las clases magistrales teóricas deriven en un monólogo del profesor con baja participación del alumnado. Existe una escasa utilización, de las herramientas disponibles para participar en la clase: mensajería instantánea, micrófono o funciones de “levantar la mano”. Esta situación se agrava aún más cuando el ratio de alumnos es alto y, por tanto, es más complicada la comunicación con todos ellos, así como saber si están asimilando los conceptos.

En este trabajo se describe la experiencia docente en la asignatura de Arquitectura de Computadores de incorporar una herramienta de aprendizaje móvil electrónico (M-learning), concretamente, Kahoot. Esta herramienta permite que el profesor plantee actividades participativas en el aula para reforzar el aprendizaje y aumentar la participación de los alumnos. Se ha realizado un estudio para determinar si el uso de Kahoot ha estimulado el aprendizaje de la asignatura de Arquitectura de Computadores y si ha mejorado la nota global final del alumnado.

Abstract

Due to the rise of remote university training, it is common for theoretical lectures to result in a monologue by the professor with low student participation. There is little use of the tools available to participate in the class: instant messaging, microphone or 'raise your hand' functions. This situation is even worse when the ratio of students is high and, therefore, it is more complicated to communicate with all of them, as well as to know if they are assimilating the concepts.

This paper describes the teaching experience in the Computer Architecture course of incorporating a mobile e-learning tool (M-learning), specifically, Kahoot.

This tool allows the teacher to propose participatory activities in the classroom to reinforce learning and increase student participation. A study has been carried out to determine whether the use of Kahoot has stimulated the learning of the Computer Architecture subject and whether it has improved the students' final overall grade.

Palabras clave

Educación en remoto, motivación alumnado, M-learning, Kahoot.

1. Introducción

La mayoría de los profesores reconocen que es un reto mantener la motivación, el compromiso y la concentración de los estudiantes a lo largo de una clase. La falta de motivación puede provocar una reducción de los resultados del aprendizaje y un ambiente negativo en el aula [5]. En el mundo universitario, este hecho se ha agravado aún más con la virtualización o semi-virtualización de las clases magistrales, ya que una gran cantidad de profesores se ha encontrado impartiendo clases a grupos de docencia bastante numerosos, pero con una escasa o nula participación del alumnado. En general, dicha lección magistral se convierte en un monólogo del profesor donde los alumnos están conectados a la sesión virtual, pero no utilizan las herramientas disponibles para interactuar: ni la mensajería instantánea ni el micrófono, ya sea para intervenir, preguntar dudas o contestar a las distintas cuestiones o ejercicios que el profesor pregunta durante la clase.

Algunas investigaciones han demostrado que los estudiantes que participan activamente en el proceso de aprendizaje aprenderán más que los que actúan de forma pasiva [1]. Además, existen numerosas pruebas de

que la participación de los estudiantes en las clases magistrales mejora la comprensión y los resultados académicos [8]. Ante este contexto, el uso del aprendizaje móvil electrónico o Mobile Learning (M-Learning), que es un tipo de aprendizaje basado en el uso de dispositivos inteligentes, puede suponer un incremento en la atención y el interés del alumnado [6]. El M-learning puede combinarse con el aprendizaje basado en juegos (gamificación), ya que según [3], si los videojuegos están bien diseñados puede motivar a los jugadores de tal forma que aprendan sin ser conscientes de ello.

Trabajos previos basados en M-learning han creado su propia plataforma web docente para interactuar con los alumnos [2]. Otros trabajos se han centrado en una aplicación gratuita llamada Kahoot que se engloba dentro del M-learning y de la gamificación para motivar a los alumnos por una determinada asignatura [4]. Kahoot permite crear encuestas, cuestionarios y discusiones, obteniendo feedback de los alumnos en tiempo real. Dicha herramienta permite que el profesor cree concursos en el aula para aprender o reforzar el aprendizaje y donde los alumnos son los concursantes. Sin embargo, hasta lo que los autores conocen, no hay trabajos previos que analicen el resultado del uso de Kahoot para incrementar el interés y la participación de los alumnos en la asignatura de Arquitectura de Computadores.

En este artículo se muestra la metodología implantada basada en sesiones Kahoot durante las sesiones magistrales virtuales así como los resultados obtenidos en una asignatura del Grado en Informática de la Universidad de Almería. En concreto, los objetivos del uso de Kahoot durante las clases han sido: el incremento de la participación de los alumnos durante las clases virtuales, la estimulación del aprendizaje de la asignatura de Arquitectura de Computadores y el aumento de la motivación de los alumnos a través de la competitividad. El artículo se ha organizado como se describe a continuación. En la Sección 2 se describe la metodología que se ha seguido en las clases. La Sección 3 muestra los resultados obtenidos durante el uso de Kahoot en el curso 2020/2021 con respecto al curso anterior. Finalmente, la Sección 4 muestra las conclusiones del artículo y posibles trabajos futuros.

2. Metodología

El uso de la tecnología digital en el proceso de aprendizaje depende en gran medida de la capacidad de los profesores de introducirla sin poner en peligro la riqueza del entorno del aula, es decir, la atención que necesitan los alumnos para seguir el flujo de la argumentación y garantizar la calidad de la indagación [7]. En nuestro caso, las sesiones con Kahoot se han incluido en las clases magistrales teóricas remotas, después

de explicar algunos conceptos relevantes o al terminar un determinado tema.

2.1. Asignatura de Arquitectura de Computadores como caso de estudio

La asignatura que hemos considerado ha sido Arquitectura de Computadores, que es una asignatura obligatoria de 6 créditos, del primer cuatrimestre de 2º del Grado en Informática en la Universidad de Almería. El objetivo principal que se plantea en esta asignatura es el estudio de las técnicas que incluyen los núcleos de procesamiento actuales para optimizar su rendimiento. Para desarrollar este objetivo el temario se compone de cuatro temas: (1) repertorio de instrucciones. Caso de estudio ARMv4, (2) rendimiento de los computadores, (3) técnicas de mejora del rendimiento de los procesadores y (4) jerarquía de memoria. Para cada uno de los temas, el número de sesiones Kahoot necesitadas han sido: 5, 2, 3 y 2. Otro dato relevante ha sido el número de preguntas totales de cada tema, que ha sido: 53, 9, 23 y 12.

2.2. Implantación de Kahoot en las clases virtuales de grupos de docencia

La implantación de la herramienta Kahoot para incrementar la participación de los alumnos en las sesiones de teoría virtuales ha consistido en el siguiente proceso.

En primer lugar, se han realizado reuniones entre el profesorado de la asignatura para definir qué preguntas de cada uno de los cuatro temas de teoría de la asignatura son susceptibles de ser realizadas utilizando la herramienta Kahoot. Además, en algunas ocasiones la pregunta requiere la reutilización o diseño de un dibujo complementario a la pregunta, bien debido al límite en el número de caracteres del enunciado de la pregunta o bien porque es necesario para su comprensión.

Posteriormente, los profesores de la asignatura se han tenido que registrar en la web <https://kahoot.com/> y crear cada uno de los cuestionarios utilizando las preguntas que se definieron previamente. En la web <https://kahoot.com/schools/ways-to-play/> se observan las distintas posibilidades de juego que ofrece Kahoot. Concretamente, en estas clases se ha considerado la modalidad “host live”, donde el profesor compartiría la pantalla y los alumnos desde sus dispositivos contestarían a las cuestiones. Se ha utilizado el tipo de pregunta “Quiz”, donde el profesor ha incluido una pregunta, un dibujo (si era necesario) y cuatro posibles soluciones, de las cuales, solo una de ellas es la correcta. A la vez que se definen las preguntas y respuestas, también se ha indicado el tiempo que tendrán los alumnos para contestar a cada una

de las cuestiones. En nuestro caso el tiempo ha sido de entre 25 segundos y 45 segundos, en función de la complejidad de la pregunta.

Durante la impartición de los grupos de docencia, cuando el profesor ha terminado de explicar una parte del temario con suficiente entidad se ha realizado una sesión de Kahoot. En ese momento, el profesor comparte su pantalla de "inicio de un juego Kahoot", donde se muestra a los alumnos un código PIN que les permitirá acceder al juego desde el enlace web <https://kahoot.it/>. Los alumnos accederán desde el navegador o a través de la App. Una vez se hayan conectado los alumnos, el profesor inicia el Kahoot. Los alumnos observarán la pregunta a través de la pantalla compartida del profesor y la contestarán utilizando sus dispositivos móviles u ordenadores. Al finalizar cada pregunta, el profesor señala la opción correcta. El profesor comprueba la tasa de aciertos de cada pregunta y, si el número de personas que se han equivocado es alto, vuelve a explicar los contenidos de teoría asociados a dicha pregunta antes de pasar a la siguiente. Dicha explicación del profesor anima a los alumnos a intervenir en la clase magistral, ya sea con el micrófono o con la herramienta de chat. Además, el profesor busca la interacción con los alumnos solicitando que algún alumno que haya acertado realice una breve explicación de la solución.

3. Análisis de resultados

Antes de comenzar a describir los resultados obtenidos, merece la pena destacar cuáles han sido los criterios de evaluación de los cursos 2019/2020 y 2020/2021 de la asignatura de Arquitectura de Computadores:

- Se plantean dos pruebas parciales. El primer parcial consta del bloque 1 (temas 1 y 2) y el segundo del bloque 2 (temas 3 y 4). Los alumnos aprueban un bloque siempre y cuando la nota sea mayor o igual a 5 sobre 10. Los alumnos que hayan obtenido una puntuación mínima de 5 puntos en ambos bloques no tendrán que presentarse al examen final. En caso contrario, se presentarán al bloque pertinente y, de nuevo, se deberá sacar una puntuación mayor o igual a 5 sobre 10 para poder aprobar la asignatura. Estos exámenes suponen el 50 % de la nota.
- La realización de las prácticas también es obligatoria, necesiéndose una puntuación mínima de 5 sobre 10 para cada una de las prácticas para poder aprobar la asignatura. Estas prácticas suponen el 25 % de la nota final. Antes del examen final, los profesores puntuarán las prácticas y darán a los alumnos un plazo de corrección y nueva en-

Alumnos	2020/2021		2019/2020	
	Total	%	Total	%
Matriculados	146	-	122	-
Presentados al 1º parcial	109	74,7	85	69,7
Aptos del 1º parcial	42	28,8	31	25,4
Presentados al 2º parcial	80	54,8	69	56,6
Aptos del 2º parcial	33	22,6	47	38,5
Total aptos	65	44,5	64	52,5

Cuadro 1: Número y porcentaje de alumnos presentados a cada uno de los parciales y que han aprobado cada uno de los parciales o el examen final de Arquitectura de Computadores.

trega de las prácticas que no habían alcanzado la puntuación mínima.

- La realización de ejercicios y problemas, que supone un 15 % de la nota final y es de carácter optativo.
- El número de aciertos en las partidas de Kahoot, que supone un 10 % de la nota final y es de carácter optativo. En el curso 2019/2020 no estaba este criterio y el apartado de realización de ejercicios y problemas supuso un 25 % de la nota final.

En la Tabla 1 se muestra para los cursos 2019/2020 y 2020/2021 un desglose del número y porcentaje, de alumnos que se han matriculado en la asignatura, que se han presentado a los parciales (y cuántos han aprobado cada uno de ellos), y, finalmente, el número de alumnos que han superado la asignatura tras la opción del examen final.

Para ambos cursos nos vamos a centrar en una serie de indicadores para realizar un análisis de los resultados obtenidos al introducir Kahoot en las lecciones magistrales virtuales. Dichos indicadores son los siguientes:

1. El número de alumnos que se presentan a cada uno de los exámenes parciales. Como se puede observar, el porcentaje de alumnos que se ha presentado al primer parcial durante el curso 2020/2021 ha sido superior al del curso 2019/2020. Los profesores de la asignatura creen que en gran medida el uso de Kahoot ha favorecido la motivación del alumno por la asignatura, así como su interés y, en primera instancia, prácticamente ningún alumno abandonó la asignatura, sino que decidieron asistir al primer parcial. Dicho porcentaje ha sido marcado en negrita para su mejor identificación. Lo mismo no ha ocurrido con el segundo parcial, donde ligeramente bajó el número de presentados y es por tanto inferior que el del curso 2019/2020. Esto pudo deberse al abandono de algunos alumnos al conocer las notas del primer parcial.
2. El número total de aprobados en la asignatura.

Calificación	2020/2021		2019/2020	
	Total	%	Total	%
Matrículas de Honor	3	2,1	3	2,5
Sobresalientes	6	4,1	3	2,5
Notables	36	24,7	29	23,8
Aprobados	19	13,0	29	23,8
Suspensos	49	33,6	24	19,7
No presentados	33	22,6	34	27,9

Cuadro 2: Desglose de las notas obtenidas por los alumnos en los cursos 2019/2020 y 2020/2021 para la asignatura de Arquitectura de Computadores.

En este caso, el hecho de que se haya utilizado una metodología basada en M-learning no ha incrementado el número de aprobados finales de la asignatura. Esto puede deberse a la complejidad añadida de cursar la asignatura a distancia.

El análisis de los resultados comentados anteriormente, nos incentivó a observar las notas obtenidas durante ambos cursos para ver si habían diferencias por el hecho de considerar la metodología basada en M-learning. Concretamente, la Tabla 2 muestra el desglose de notas (matrículas de honor, sobresalientes, notables, aprobados, suspensos y no presentados), tanto en número como en porcentaje. Como se ha remarcado en negrita en dicha tabla, el porcentaje de alumnos con notas altas (sobresalientes y notables) del curso 2020/2021 (28,8 %) es superior con respecto al del curso 2019/2020 (26,3 %) de alumnos. Este es un aspecto muy significativo ya que, aparentemente, los alumnos han sido capaces de estudiar la asignatura asimilando mejor los conceptos clave y obteniendo mejores resultados en su clasificación. Otro aspecto reseñable es que el porcentaje de alumnos no presentados ha sido inferior durante el curso 2020/2021, lo que recalca la idea que describimos anteriormente de que los alumnos han estado más motivados por la asignatura.

Finalmente, una vez cerradas las actas de la asignatura, se realizó una encuesta para obtener realimentación por parte de los alumnos del uso de Kahoot en las clases online. Se realizaron dos preguntas con posibles respuestas de “sí” o “no”. La pregunta (1) fue: “El uso de Kahoot ¿ha mejorado tu interés por esta asignatura?” y la pregunta (2) fue: “El uso de Kahoot ¿ha incrementado tu participación en la clase de docencia online?” Dicha encuesta fue realizada por 40 alumnos y los resultados fueron de 34 resultados positivos para la pregunta (1), frente a 6 negativos y de 29 resultados positivos para la pregunta (2), frente a 11 negativos.

4. Conclusiones y trabajos futuros

En este artículo se ha realizado un análisis de la metodología utilizada para introducir la herramienta de

M-learning llamada Kahoot en la asignatura de Arquitectura de Computadores del grado de Ingeniería Informática, así como una descripción de los resultados obtenidos. Tal y como se ha observado en la encuesta final, los alumnos han mostrado una mayor participación desde que se está utilizando la herramienta Kahoot en los grupos de docencia. De hecho, la impresión de los profesores es que los minutos de la clase en los que están más participativos son los que coinciden con los test de Kahoot. Gracias a esta herramienta, los profesores de la asignatura son conscientes de que algunos conceptos no han sido aprendidos por los alumnos correctamente y, por tanto, necesitan ser reforzados. Además, el análisis de los resultados ha demostrado que el porcentaje de alumnos presentados al primer parcial de la asignatura ha sido mayor que el curso anterior. Como trabajo futuro, se prevé el uso de la herramienta Wooclap que ha sido recientemente integrada en el Aula Virtual del Blackboard que utiliza la asignatura.

Referencias

- [1] Jennifer Butler. Use of teaching methods within the lecture format. *Medical Teacher*, 14 (1) (1992), pp. 11-25
- [2] Antonio Cañas, Javier Fernández, Eva M. Ortigosa, Mancia Anguita. Diseño e integración en una plataforma docente de una herramienta de respuesta de audiencia para mejorar la atención, la evaluación y el aprendizaje de los estudiantes. *Actas de la JENUI*, 5 (2020), pp. 141-148
- [3] James P. Gee. What video games have to teach us about learning and literacy. *Comput. Entertain.*, 1 (1) (2003)
- [4] Julio López-Gómez, Ángeles Carrasco y Francisco Mata. Evaluando la participación con aprovechamiento en clase: de lo cualitativo a lo cuantitativo. *Actas de la JENUI*, 3 (2018), pp. 165-172
- [5] Ou Lydia Liu, Brent Bridgeman y Rachel M. Adler. Measuring learning outcomes in higher education: Motivation matters. *Educational Researcher*, 41 (9) (2012), pp. 352-362
- [6] Mario R. Mejía. M-Learning: Uso, características, ventajas y desventajas. *Revista Internacional Tecnológica Educativa-Docentes 2.0*, volumen 8(1) (2020), pp. 50-53
- [7] Luís F. Mendes, Cláudia M. de Oliveira y Carlos M. das Neves Santos. A critical review of mobile learning integration in formal educational contexts. *Int J of Educ Tech in Higher Ed*, 15(10) (2018)
- [8] Michael Prince. Does active learning work? A review of the research. *Journal of engineering education*, 93 (3) (2004), pp. 223-231

Práctica de desarrollo de una red de sensores basada en la programación de microcontroladores para sistemas en tiempo real

Elena Cerezuela-Escudero, Lourdes Duran-López, Daniel Gutiérrez-Galán, Juan Pedro Domínguez-Morales, Antonio Ríos-Navarro, Ángel Jiménez-Fernández

Departamento de Arquitectura y Tecnología de Computadores
Universidad de Sevilla

Sevilla

ecerezuela@us.es, lduran@atc.us.es, dgutierrez@atc.us.es, jpdominguez@atc.us.es,
arios@us.es, ajimenez@atc.us.es

Resumen

Este artículo presenta una práctica de laboratorio impartida mediante una metodología de aprendizaje basado en proyectos cuyo objetivo principal es adquirir la capacidad de programación de microcontroladores para el diseño e implantación de una red de comunicación inalámbrica de bajo consumo en el ámbito de las comunicaciones industriales. Este proyecto se desarrolla en las sesiones prácticas de la asignatura Comunicaciones Industriales del Máster Universitario en Sistemas Inteligentes en Energía y Transporte cuyas competencias contemplan la capacidad para planificar la instalación de redes de comunicación en entornos industriales, así como la aplicación de los conocimientos adquiridos en la resolución de problemas multidisciplinares. El proyecto de laboratorio expuesto en este trabajo abarca ambas competencias, ya que propone establecer una red de control jerárquica en las comunicaciones industriales mediante la programación de microcontroladores, abordando el desarrollo de sistemas empotrados en tiempo real. Los dispositivos empleados para establecer la red de sensores son módulos XBee conectados a un puerto UART de un microcontrolador ARM Cortex-M0 para establecer comunicación serie asíncrona entre el módulo XBee y el microcontrolador. La arquitectura de la red está basada en la topología estrella. El módulo XBee central está conectado a través de un puerto COM Virtual a un PC. En el PC se ejecuta una interfaz de usuario desarrollada en lenguaje C# usando el entorno Visual Studio basada en la interfaz gráfica Windows Forms.

Abstract

This article presents a laboratory practice taught by project-based learning methodology whose main objective is to acquire the ability to program micro-

controllers for the design and implementation of a low-power wireless communication network in the field of industrial communications. This project is developed during the practical sessions of the Industrial Communications subject of the *Máster Universitario en Sistemas Inteligentes en Energía y Transporte*, whose skills include the ability to plan the communication network installation in industrial environments, as well as the application of the knowledge acquired in the resolution of multidisciplinary problems. The laboratory project exposed in this work covers both skills because it proposes to establish a hierarchical control network in industrial communications through the programming microcontrollers, addressing the development of embedded real time systems. The devices used to establish the sensor network are XBee modules connected to a UART of an ARM Cortex-M0 microcontroller to establish asynchronous serial communication between the XBee module and the microcontroller. The network's architecture is based on a star-shaped topology. The central XBee module is connected through a Virtual COM port to a PC. A user interface is executed on the PC; it is developed in C# language using the Visual Studio environment based on a graphical interface Windows Forms.

Palabras clave

Microcontrolador, sensores, actuadores, sistema en tiempo real, red inalámbrica, XBee.

1. Introducción

La asignatura Comunicaciones Industriales es una de las asignaturas del Máster Universitario en Sistemas Inteligentes en Energía y Transporte. Este plan de estudios es una iniciativa enmarcada dentro del Campus de Excelencia Internacional Andalucía

TECH para responder a la necesidad de ingenieros de alta cualificación en áreas relacionadas con la aplicación de las TICs a sectores como la Energía, el Medio Ambiente o el Transporte.

El propósito principal de este máster es formar titulados de distintas ramas de la Ingeniería para que desempeñen labores profesionales e investigadoras relacionadas con la aplicación de la Electrónica, la Informática y las Comunicaciones en el ámbito de las ciudades inteligentes (Smart Cities), redes eléctricas inteligentes (Smart Grids), energías renovables, transporte inteligente y automatización avanzada de edificios.

La asignatura Comunicaciones Industriales forma parte del módulo de formación común y su competencia específica es dotar al alumno de la capacidad para seleccionar, diseñar, proyectar e implantar infraestructuras de comunicaciones seguras y eficientes en aplicaciones industriales, y en especial en sistemas con inteligencia distribuida.

La incorporación de las comunicaciones al proceso de producción industrial proporciona mejora del rendimiento y flexibilidad porque permite la supervisión del proceso, toma de datos instantánea, control a distancia, detección y corrección de fallos. Por la complejidad y exigencias del proceso productivo, se proponen modelos jerárquicos para la implantación de sistemas automáticos, siendo el modelo *Computer Integrated Manufacturing* (CIM) de los más difundidos [3]. Este modelo propone la división en niveles funcionales, de forma que cada nivel se caracteriza por llevar a cabo labores específicas, requiriendo cada uno un tipo de información y procesamiento diferente. El flujo de información se produce entre los elementos de un mismo nivel y con los niveles adyacentes. El nivel 1, denominado Sensor y Actuador, está formado por los dispositivos de campo que interactúan con el proceso, como son los sensores, cuya función es medir datos relevantes para el proceso, los elementos de control, y los actuadores, que se encargan de ejecutar las órdenes indicadas por los elementos de control para modificar el proceso. En este nivel, el ancho de banda no es un requisito crítico, porque los mensajes a transmitir son de longitud reducida (del orden de bits o bytes), pero sí es un requisito crítico cumplir con restricciones temporales. La aplicación de microcontroladores como elementos de control en este nivel de la jerarquía resuelve la conexión y control de los sensores/actuadores y aporta herramientas para las restricciones temporales, todo ello a bajo coste y bajo consumo. Para reducir el coste y el consumo energético, alargando la vida útil de las baterías, se propone el uso de la tecnología XBee para la comunicación de los elementos de control (<https://www.digi.com/xbee>). Esta tecnología sigue el estándar Zigbee [2], cuyo ancho de banda

supera el necesario al establecido para el nivel Sensor y Actuador del modelo CIM.

En este artículo se describe el proyecto final realizados en varias sesiones prácticas de la asignatura Comunicaciones Industriales, que consiste en configurar e implementar una red de comunicación inalámbrica basada en microcontroladores y tecnología XBee.

2. Metodología de aprendizaje

Con los objetivos de fomentar el aprendizaje y motivar al alumnado, se trabajan los contenidos de forma práctica aplicando la metodología pedagógica de Aprendizaje Basado en Proyecto (ABP) [4]. Con esta metodología se adquieren tanto los conocimientos técnicos como las siguientes competencias transversales: capacidad de resolución de problemas, trabajo en equipo, habilidad de comunicación oral y escrita, adaptación a situaciones nuevas, planificación y organización [1].

El proyecto consiste en el desarrollo de una red inalámbrica en topología en estrella basada en microcontroladores y tecnología XBee. Cada microcontrolador tiene que gestionar un sensor y un actuador, excepto el nodo central que se encarga del envío/recepción de datos al resto de nodos. Para controlar y monitorizar el sistema se realiza una interfaz gráfica para PC.

Previamente, se han realizado otras sesiones prácticas con la misma metodología de aprendizaje pero con proyectos menos complejos, para adquirir los conocimientos básicos sobre programación de microcontroladores para la lectura de sensores analógicos, el control de actuadores y la comunicación por el puerto serie asíncrono. Para la programación de este tipo de sistemas, que requiere un control temporal de la repuesta, se han explicado los diferentes métodos para un implantar un sistema en tiempo real; procesamiento secuencial, procesamiento mediante interrupciones y sistemas operativos en tiempo real.

En el apartado 4 se explica en detalle la dinámica seguida para la realización de este proyecto.

3. Descripción del material

A continuación, se describen los elementos hardware requeridos y las aplicaciones software necesarias para la configuración y el desarrollo del sistema.

3.1. Placa de desarrollo FRDM-KL25Z

Placa de desarrollo fabricada actualmente por la compañía NXP, basada en un microcontrolador Kinetis ARM Cortex-M0+. Características hardware:

- 48MHz, 16KB RAM, 128KB FLASH
- USB (Host/Device)
- SPI (Serial Peripheral Interface) (2)

- I2C (Inter-Integrated Circuit) (2)
- UART (Universal Asynchronous Receiver-Transmitter) (3)
- ADC (Analog-to-Digital Converter) (16 bit)
- DAC (Digital-to-Analog Converter) (12bit)
- GPIO (General-Purpose Input/Output) (66)

Esta placa supera los requisitos hardware del sistema a desarrollar, pero se ha decidido seleccionar una placa con un microcontroladores ARM para que practiquen con la tecnología actual en este ámbito.

Esta placa está provista de un adaptador OpenSDA cuya funcionalidad simplifica la grabación del firmware en la memoria FLASH y permite la comunicación con el PC por el puerto COM virtual.

3.2. Módems XBee

XBee es una familia de módulos de radio que proveen conectividad inalámbrica con otro dispositivo de la misma serie. El fabricante, Digi International Inc., ofrece módulos que implementan diferentes protocolos de comunicación inalámbrica como Wi-Fi y Zigbee [2]. Son apropiados para aplicaciones de bajo consumo por su capacidad de pasar de inactivo a activo en menos de 30 milisegundos. Están provistos de interfaz UART para la comunicación con microcontroladores.

En este proyecto se usan los módulos XBee-PRO S2C, que se entregan junto a una placa adaptadora para facilitar la conexión al microcontrolador. Para la configuración inicial de los módems desde el PC, se usa una placa intermedia con un puerto USB que permite la comunicación serie con el módem.

3.3. Sensor y actuador

Los nodos finales constan de un sensor de temperatura y un servomotor de modelismo (servo).

- Sensor de temperatura analógico MCP9700A: sensor que contiene Linear Active Thermistor Integrated Circuit que convierte la temperatura a voltaje. De forma que la temperatura se obtiene por la relación lineal con el voltaje generado, conociendo el coeficiente de temperatura (Volts/° Celsius) y el voltaje a 0° Celsius.
- Micro Servo 9g: servo de bajo coste, con una rotación máxima de 180 grados y alimentación requerida de 4.8 voltios.

3.4. Aplicaciones software

Los módems XBee tienen disponible dos modos de funcionamiento: el modo API y el modo transparente. El modo de funcionamiento transparente implementa un puerto serie asíncrono para enviar la información sin formato. Se ha seleccionado este modo para aprovechar la potencia de la comunicación serie asíncrona en redes de con los requisitos: bajo ancho de banda, respuesta en tiempo real y simplicidad de

implementación. Además, los módulos XBee pueden actuar como coordinador, enrutador o dispositivo final. Como la topología de la red propuesta es en estrella, todos los módems serán dispositivos finales excepto el nodo central que es coordinador. El software usado en el PC para realizar la configuración inicial de los módems XBee es XCTU, producto gratuito de Digi multiplataforma (Windows, MacOS y Linux).

Para la programación de los microcontroladores se va a usar el entorno de desarrollo online mBed (<https://os.mbed.com/>). Es una plataforma de desarrollo para microcontroladores en lenguaje C/C++. Tiene disponible una biblioteca para sistemas operativos en tiempo real y cientos de librería de periféricos y módulos reutilizables. Se ha optado por este entorno por su simplicidad de uso, disminuyendo el tiempo inicial de aprendizaje de una tarea ardua como es la programación de microcontroladores, sobre todo para este tipo de alumnos de perfil ingenieril pero en otras ramas diferentes a la informática.

Para el desarrollo de la interfaz gráfica se usa el lenguaje C# usando el entorno Visual Studio basada en la interfaz gráfica Windows Forms.

4. Desarrollo del proyecto

A continuación, se describe la dinámica de las dos sesiones prácticas, cada sesión de dos horas, en las que se trata este proyecto.

Una semana antes de la sesión práctica, los alumnos tienen disponible un documento en el que se describe el caso práctico a implementar: desarrollo de una red inalámbrica con topología en estrella mediante módems XBee y microcontroladores. Cada microcontrolador está conectados a un sensor de temperatura y a un servo. El sistema está monitorizado y controlado por una interfaz gráfica para PC.

Para el desarrollo del firmware de cada microcontrolador se hace entrega de un documento con los siguientes apartados:

- Descripción de los módems XBee entregados y el proceso de configuración del nodo central y los dispositivos finales.
- Esquema general de comunicaciones del sistema mostrado en Figura 1.
- Funcionamiento del sistema a través de mensajes con órdenes desde el nodo central a los dispositivos finales y mensajes de datos desde los dispositivos finales al nodo central.
- Diseño del sistema de comunicación mediante mensajes de 4 caracteres en el que el primer carácter identifica el comando, el segundo y tercer carácter se usan para enviar datos y el cuarto carácter para indicar el final de la trama. Se establece un carácter específico para indicar el final del mensaje.



Figura 1: Esquema de comunicación a implementar (tantos dispositivos finales como alumnos).

El profesor comienza la sesión con una explicación del algoritmo para la recepción cíclica de los caracteres recibidos por el puerto serie asíncrono, tanto para un procesamiento secuencial como para un procesamiento mediante interrupciones generadas por la disponibilidad de un carácter en el puerto serie. El resto de conocimientos sobre programación para afrontar el firmware se ha trabajado en las anteriores sesiones prácticas: lectura de sensores analógicos, control de servo y envío por el puerto serie asíncrono.

El nodo central está configurado y conectado al PC del profesor mediante una placa conversor a USB. En el PC del profesor se ejecuta una interfaz gráfica que lee la información del puerto serie (mensajes que han llegado al nodo central) y envía órdenes a cada uno de los módems XBee conectados para ser procesadas por el microcontrolador.

Una vez que los alumnos han completado el firmware, se les entrega un documento con los contenidos necesarios para desarrollar la interfaz gráfica en Visual Studio y lenguaje C# usando Windows Forms. El desarrollo de la interfaz gráfica lo afrontan en la segunda sesión práctica.

El desarrollo del firmware se realiza individualmente. En cambio, debido a la complejidad de enfrentarse a un entorno de programación y lenguaje que no se ha tratado previamente, se organizan en grupos de 4 o 5 alumnos para realizar la interfaz gráfica. El objeto de evaluación en esta actividad es un documento, realizado de forma individual, que recoja la solución completa para el caso práctico.

5. Resultados

Desde el curso académico 16/17, se dedica el 30% de las sesiones de la asignatura a la programación de microcontroladores para desarrollar sistemas en tiempo real que requieren comunicación. El proyecto final expuesto en este artículo se ha realizado en los cursos 18/19 y 19/20. Se recogió la opinión de los alumnos mediante encuestas anónimas digitales en la última sesión práctica. El cuestionario está formado por las siguientes afirmaciones que el alumno puede evaluar de 1 a 5 según el grado de satisfacción:

- Estoy satisfecho con la parte práctica.
- Considero que he adquirido los conocimientos para diseñar, proyectar e implantar una red inalámbrica basada en microcontroladores.

- Estoy satisfecho con la dinámica de las sesiones prácticas.
- Considero que la programación de microcontroladores es de interés para mi formación.
- Tenía conocimientos previos sobre programación de microcontroladores.

En el curso 18/19 11 de los 12 alumnos matriculados completaron el cuestionario y en el curso 19/20 17 de los 19 alumnos. Los resultados para los 28 cuestionarios recogidos en los dos cursos, sin tener en cuenta la última cuestión, es de 4,2. Se observa como incluso para alumnos que indicaron que no tenían conocimiento previo en programación de microcontroladores, la valoración es mayor a 4.

6. Conclusiones

En este documento hemos presentado la metodología seguida para el desarrollo de un caso práctico en las sesiones prácticas de la asignatura Comunicaciones Industriales. Con el enfoque que hemos elegido pretendemos acercar la programación con microcontroladores a alumnos de perfil ingenieril (excepto ingenieros informáticos) mediante una aproximación amena y atractiva. Además, mediante el aprendizaje basado en proyecto pretendemos formar a los alumnos en las competencias transversales: capacidad de resolución de problemas, trabajo en equipo, habilidad de comunicación oral y escrita, adaptación a situaciones nuevas, planificación y organización.

Para posteriores cursos queremos hacer una evaluación de la metodología realizada de forma más exhaustiva para medir las habilidades adquiridas en esta parte de la asignatura.

Referencias

- [1] Labra, J.E., Fernández, D., Calvo, J., Cernuda, A. Una Experiencia de aprendizaje basado en proyectos utilizando herramientas colaborativas de desarrollo de software libre. XII Jornadas de Enseñanza Universitaria de Informática (JENUi), Bilbao, Junio, 2006
- [2] Ramya, C. Muthu, M. Shanmugaraj, y R. Prabhakaran. Study on ZigBee technology. In 2011 3rd International Conference on Electronics Computer Technology, vol. 6, pp. 297-301. IEEE, 2011.
- [3] Scheer, August-Wilhelm. CIM Computer Integrated Manufacturing: Towards the Factory of the Future. Springer Science & Business Media, 2012.
- [4] Solomon, Gwen. "Project-based learning: A primer." Technology and learning-dayton- 23, no. 6 (2003): 20-20.

Semipresencialidad en tiempos de COVID-19: adaptación de la docencia en el ámbito de Fundamentos de Informática

L. Duran-Lopez, D. Gutierrez-Galan, E. Cerezuela-Escudero, A. Rios-Navarro

J. P. Dominguez-Morales

Dpto. Arquitectura y Tecnología de Computadores

Universidad de Sevilla

41012 Sevilla

lduran@atc.us.es

Resumen

El mundo ha cambiado desde la aparición del nuevo virus de la familia de los coronavirus (COVID-19), el cual fue declarado pandemia en tan sólo unos pocos meses. El COVID-19 ha tenido un alto impacto en todos los ámbitos conocidos, incluida la docencia. Las medidas de seguridad impartidas por las autoridades sanitarias implicaron una necesaria adaptación de la docencia en general, y más en concreto la universitaria. Este artículo se centra en cómo ha afectado la pandemia a la asignatura de Informática en el ámbito del Grado de Ingeniería Química Industrial. En dicha asignatura se introducen a los alumnos los conceptos básicos de la informática, incluyendo sistemas operativos, bases de datos y redes de computadores, además de un bloque de sesiones de laboratorio dedicado a la programación en lenguaje C. La situación actual ha dado lugar a la semipresencialidad en la docencia de la asignatura durante el curso 2020/2021, repercutiendo tanto a las metodologías de enseñanza como a los contenidos teórico-prácticos. En el presente trabajo se describen los cambios realizados de cara a adaptar el contenido de la asignatura a la semipresencialidad, reportando resultados tanto a nivel de evaluación como a nivel de seguimiento y motivación del alumnado.

Abstract

The world has changed due to the emergence of the new virus from the coronavirus family (COVID-19), which was declared as a pandemic a few months after its appearance. COVID-19 has had a high impact in every single field, including education and teaching. The safety measures and prevention protocol against COVID-19 has involved an adaptation process in the education, particularly in the university. This article is focused on how the pandemic has affected the Computer Science subject in the field of the Industrial Chemical Engineering Degree at the Univer-

sity of Seville. In this subject, students are introduced to the basic concepts of computer science, including operating systems, databases and computer networks, as well as a set of laboratory sessions dedicated to programming in C language. The current situation has led to a blended learning in the subject during the 2020/2021 academic year, affecting both the methodology and theoretical-practical contents. The present paper describes the changes performed as a consequence to adapt the subject content to blendedness, reporting results in terms of evaluation marks and also at students' motivation level.

Palabras clave

COVID-19, semipresencialidad, Fundamentos de Informática, dinámica de aprendizaje.

1. Introducción y motivación

La asignatura de Informática se imparte dentro del plan de estudios del Grado de Ingeniería Química. El principal objetivo de esta asignatura es enseñar a los alumnos los conceptos relacionados con fundamentos de programación de ordenadores, así como los conocimientos básicos en cuanto a representación de la información en un computador, sistemas operativos, memoria, redes de computadores, bases de datos y programas informáticos con aplicación en ingeniería. Para ello, la materia consta de una parte teórica y una práctica. La parte teórica se divide en tres bloques independientes para tratar cada uno de los campos mencionados anteriormente. Mientras que, en la parte práctica, se realizan sesiones incrementales para afianzar los conocimientos sobre programación básica en lenguaje C mediante la resolución de ejercicios y problemas usando el entorno de desarrollo libre Zinjai¹.

¹<http://zinjai.sourceforge.net>

De forma general, la metodología que se había seguido hasta ahora en la asignatura se desarrollaba de forma presencial tanto para las sesiones de teoría como para prácticas. En cuanto a las clases magistrales, además de impartir el temario relacionado, cada semana se introducía el contenido teórico de la práctica correspondiente a esa semana. También se dedicaban sesiones completas para la realización de ejercicios de cara a reforzar los conocimientos teórico-prácticos. En cuanto a la parte práctica, se impartían un total de ocho prácticas, de las cuales parte de ellas se realizaban en una única sesión, mientras que aquellas que tuvieran un contenido más extenso o complejo se les dedicaba una sesión doble para que los alumnos reforzaran los conocimientos.

La pandemia que ha tenido lugar desde marzo del año 2019 ha supuesto un cambio radical en todos los aspectos. Uno de los campos más afectados por el virus es la enseñanza, especialmente la universitaria [6, 5]. Los protocolos de prevención impartidos por las autoridades sanitarias para impedir que el virus se expanda han implicado una necesaria adaptación de la docencia universitaria [1, 3]. La presencialidad de la enseñanza ha sido uno de los principales aspectos afectados, el cual ha tenido un alto impacto para el alumnado [2, 4].

En este artículo se describen los cambios tanto en los contenidos teórico-prácticos como en la metodología utilizada para la asignatura de Informática en el ámbito de Ingeniería Química Industrial como adaptación a la nueva situación. Se presenta un reporte cuantitativo comparando los resultados de la evaluación alternativa del curso actual frente a los dos cursos anteriores, en los que la metodología de evaluación no se ha visto modificada.

2. Metodología docente

La asignatura, la cual se imparte durante el primer cuatrimestre de primer curso de Ingeniería Química Industrial, consta de un total de 30 horas asociadas a la docencia teórica y otras 30 horas de clases prácticas, distribuidas en 15 sesiones de dos horas cada una.

En cuanto a la docencia relacionada con los contenidos teóricos, la metodología aplicada se vio afectada por la semipresencialidad. Para poder mantener una distancia mínima de seguridad entre alumnos, el aforo máximo en las aulas se redujo a más de la mitad y, por ello, era inviable que todos los alumnos asistieran de forma presencial simultáneamente. Debido a esto, cada grupo de teoría tuvo que dividirse en dos subgrupos, haciendo que dichos subgrupos se fueran alternando en semanas consecutivas y no coincidieran al mismo tiempo en el aula. Esto dio lugar a que las clases magistrales pasaran a impartirse mediante una semipresencialidad síncrona. De esta forma, al mismo

tiempo que un subgrupo asistía a clase de forma presencial, el docente que impartía dicha clase la retransmitía en directo para el subgrupo contrario gracias al uso de cámaras instaladas en el aula y un micrófono para mejorar la calidad de audio. Así, el feedback con el alumnado no se perdía de forma tan directa como en otras alternativas donde se ha optado por proveer vídeos pregrabados donde el alumno no puede tener interacción ninguna. Esta fue la metodología planteada y seguida durante las primeras semanas de curso. No obstante, esta tuvo que ser adaptada de nuevo rápidamente debido a las nuevas restricciones por parte de la Junta de Andalucía ante la segunda ola de contagios causada por la COVID-19, de forma que las clases teóricas tuvieron que impartirse completamente en modalidad online. De esta forma, los subgrupos de teoría se combinaron de nuevo, y cada grupo asistía a su clase correspondiente de forma telemática.

En cuanto a las prácticas, al igual que ocurrió con las clases de teoría, cada grupo se volvió a dividir en dos subgrupos (A y B), que asistían a clase en semanas consecutivas alternándose para no coincidir en el aula, y así cumplir con la normativa establecida por el centro. De esta forma, los alumnos que asistían a clase recibían una introducción al contenido teórico de la práctica, además de ser guiados de forma personal durante las dos horas de la sesión, mientras que el otro subgrupo continuaba desde su domicilio realizando ejercicios y avanzando con los contenidos ya vistos la semana anterior en clase. En este caso, la docencia no pudo ser asíncrona debido a la falta de cámaras y medios para retransmitir la clase en directo. Dada la situación, se acordó en el programa docente establecer las prácticas como no obligatorias de forma que cada alumno podía decidir por su parte si asistir o no a la sesión sin que esto perjudicase su nota. El cambio en cuanto a las restricciones por parte de la Junta de Andalucía no afectó en absoluto a este esquema de docencia para las sesiones prácticas, ya que el centro donde se impartían obligaba a que los alumnos de primer curso disfrutaran de sesiones prácticas de forma presencial.

La evaluación alternativa se desarrolló de la misma forma que en cursos anteriores, y se compuso de tres partes divididas en los siguientes puntos:

- Una primera prueba que evaluaba los contenidos teóricos del primero de los tres bloques del temario, y de las tres primeras sesiones prácticas. Se aprovechó la presencialidad obligatoria de las clases prácticas para realizar dicha prueba de manera presencial, aunque también se facilitó una alternativa online para aquellos alumnos que lo necesitaran. Esta prueba supuso un 30 % de la nota total de la evaluación alternativa.
- Dos ejercicios entregables, uno de teoría y otro de prácticas. Debido a las restricciones, el corres-

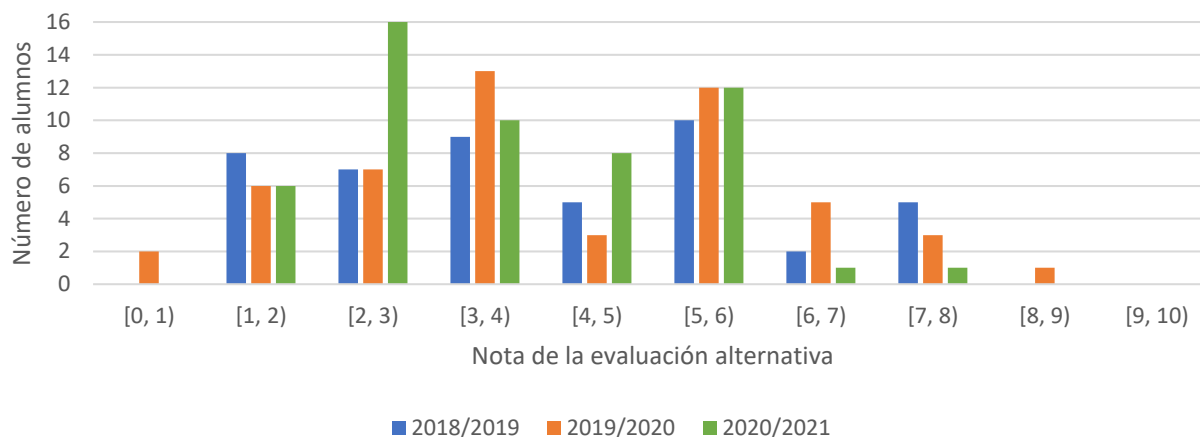


Figura 1: Histograma comparativo de las notas de la evaluación alternativa de la asignatura en los cursos 2020/2021 (verde), 2019/2020 (naranja) y 2018/2019 (azul). Las notas se representan desde 0 hasta 10 por intervalos de uno en uno.

pondiente a la teoría tuvo que realizarse de forma online, mientras que el correspondiente a la parte práctica se realizó presencialmente. La media de ambos ejercicios supuso el 10 % de la nota de evaluación alternativa.

- Una segunda prueba presencial realizada la última clase del curso, que evaluaba todo el contenido de la asignatura tanto teórico como práctico, y que representaba el 60 % restante de la nota. Debido a la carga en cuanto a materia de esta prueba, se estableció una nota mínima de 4 puntos para poder ser evaluado en la evaluación alternativa.

3. Resultados

En esta sección se presentan los resultados obtenidos por los alumnos en el presente curso, junto con una comparación cuantitativa y cualitativa frente a cursos anteriores, donde puede verse el impacto de la pandemia en la docencia de esta asignatura.

Por un lado, los resultados en la evaluación alternativa del curso 2020/2021 siguiendo la metodología presentada en la sección anterior no han sido tan satisfactorios como el profesorado de la asignatura esperaba. En la Figura 1 se muestra un histograma donde se comparan los resultados obtenidos por los alumnos en la evaluación alternativa para los últimos tres cursos.

El número de alumnos presentados en la evaluación alternativa en la asignatura de Informática en los tres últimos cursos ha mantenido una tendencia ascendente, correspondiendo a 46, 52, y 54 alumnos presentados en los cursos 2018/2019, 2019/2020 y 2020/2021, respectivamente. En cuanto a las notas medias obtenidas para cada curso, se observan unos valores de 3.91, 4.10 y 3.67. Puede verse una ligera caída en la nota media en el último curso, correspondiente al 2020/2021.

Este descenso en la nota media de la evaluación alternativa puede estar causada por varios factores. En cuanto a teoría, las clases online impuestas por la Junta de Andalucía como consecuencia de la segunda ola de la pandemia, puede ser uno de estos factores. Durante las distintas sesiones de teoría se ha observado una menor participación del alumnado comparada con años previos, incluso en aquellas sesiones destinadas a la resolución de ejercicios y problemas. Debido a que las clases quedaban grabadas en la plataforma de retransmisión, son muchos los estudiantes que han optado por no seguir la docencia teórica de forma sincrónica y, por contrario, visualizar la grabación a posteriori. Esto ha dado lugar a una participación más reducida y, por consiguiente, la resolución de muchas menos dudas en clase. Junto a esto, las solicitudes de tutorías por parte del alumnado han sido muy inferiores a las de cursos anteriores, centrándose la gran mayoría en la semana antes de la última prueba de evaluación continua.

En cuanto a la docencia en las sesiones prácticas de la asignatura, el hecho de no establecer las clases como obligatorias es otra de las posibles causas que ha podido tener un impacto negativo sobre la nota media de la evaluación alternativa. La no obligatoriedad en la asistencia ha dado lugar a numerosas sesiones con menos de 3 alumnos por clase para un contenido que, teniendo en cuenta el porcentaje que representa dentro de la carga de la asignatura y de la evaluación, se hace indispensable trabajar junto al profesor. Debido a las medidas comentadas, la asistencia del alumnado a las sesiones prácticas se redujo hasta un 50 % con respecto a cursos anteriores, especialmente en aquellos alumnos que no cursaban la asignatura en primera convocatoria.

Si bien la no presencialidad ha podido favorecer o, al menos, no perjudicar a alumnos de más avanzado

curso que ya conocen la dinámica de aprendizaje en la Universidad, los alumnos de primer curso se han visto claramente afectados por las restricciones necesarias debido a la pandemia actual en lo que respecta a la asignatura a la que se refiere este estudio. Por otro lado, el cambio en la metodología docente y en la presencialidad de las sesiones a mitad de curso ha podido causar desconcierto tanto en alumnos como profesores, que han tenido que adaptarse de forma rápida y sin antecedentes previos.

4. Conclusiones

En este artículo se describen los cambios realizados tanto en contenido como en la metodología docente para la asignatura de Informática en el ámbito de Ingeniería Química Industrial como adaptación ante la pandemia provocada por el virus COVID-19.

En cuanto a la parte teórica, las clases se han impartido de manera online como medida de seguridad impuesto por la Junta de Andalucía ante la segunda ola. Con respecto a las prácticas, se ha seguido una metodología basada en la semipresencialidad, de forma que cada grupo fue dividido en dos subgrupos. Cada subgrupo asistía semanas alternas para recibir la sesión práctica que le correspondía, mientras que el otro subgrupo se dedicaba a avanzar por su cuenta. La evaluación alternativa consistió en diferentes pruebas donde se evaluaron tanto los conocimientos teóricos como prácticos.

Los resultados obtenidos en la evaluación alternativa durante el curso 2020/2021 por parte del alumnado muestran el impacto de la pandemia en el seguimiento de la asignatura, el cual se ha visto mermado desde primera instancia del curso. La implicación por parte de los estudiantes no ha sido la esperada, y muchos han preferido recurrir a las grabaciones de las clases teóricas que a la propia clase impartida por el profesor, dando lugar a una menor participación por parte de estos y a un descenso en la motivación de cara a proponer y resolver problemas relacionados con la asignatura. Con respecto a las clases prácticas, la asistencia se ha visto reducida considerablemente al establecerlas como no obligatorias en cuanto a presencialidad (en cursos anteriores sí lo eran), factor que también ha afectado al seguimiento de dichos contenidos, siendo en este caso más crítico que en la teoría, ya que son incrementales

a lo largo de las distintas sesiones. La nota media de la evaluación alternativa se ha visto afectada por consiguiente de forma negativa comparada con otros cursos anteriores, siendo de 3.67 en el curso 2020/2021, frente al 4.10 y 3.91 de los cursos 2019/2020 y 2018/2019, respectivamente.

La falta de antecedentes ha afectado claramente a la docencia universitaria. Si bien tanto alumnos como profesores nos estamos adaptando a estas nuevas medidas, queda claro en base a los resultados cuantitativos y cualitativos obtenidos que la presencialidad es relevante para todos los estudiantes, y en especial en alumnos de nuevo ingreso sin experiencia previa en el ámbito universitario.

Referencias

- [1] Manuel Area-Moreira, Anabel Bethencourt-Aguilar, and Sebastián Martín-Gómez. De la enseñanza semipresencial a la enseñanza online en tiempos de covid19. visiones del alumnado. *Campus Virtuales*, 9(2):35–50, 2020.
- [2] Tom Crick, Cathryn Knight, Richard Watermeyer, and Janet Goodall. The impact of covid-19 and “emergency remote teaching” on the uk computer science education community. In *United Kingdom & Ireland Computing Education Research conference.*, pages 31–37, 2020.
- [3] María Isabel García-Planas, Judit Taberna Torres, et al. Transición de la docencia presencial a la no presencial en la upc durante la pandemia del covid-19. *IJERI: International Journal of Educational Research and Innovation*, (15):177–187, 2021.
- [4] Petar Jandrić, David Hayes, Ian Truelove, Paul Levinson, Peter Mayo, Thomas Ryberg, Lilia D Monzó, Quaylan Allen, Paul Alexander Stewart, Paul R Carr, et al. Teaching in the age of covid-19. *Postdigital Science and Education*, 2(3):1069–1230, 2020.
- [5] Giorgio Marinoni, Hilligje Van’t Land, and Trine Jensen. The impact of covid-19 on higher education around the world. *IAU Global Survey Report*, 2020.
- [6] Pradeep Sahu. Closure of universities due to coronavirus disease 2019 (covid-19): impact on education and mental health of students and academic staff. *Cureus*, 12(4), 2020.

Trabajos fin de grado para la adquisición de competencias de la titulación mediante un servicio a personas con diversidad funcional y cognitiva

Paula M. Castro, Óscar Fresnedo, Adriana Dapena, Javier Pereira
Universidade da Coruña, CITIC, 15071 A Coruña
paula.castro@udc.es, oscar.fresnedo@udc.es,
adriana.dapena@udc.es, javier.pereira@udc.es

Resumen

Aprendizaje y servicio es un modelo pedagógico orientado a la adquisición de competencias transversales y específicas de los títulos a través de la colaboración con entidades sin ánimo de lucro. Este artículo muestra una experiencia, iniciada en el curso académico 2019-2020, para el desarrollo de Trabajos Fin de Grado (TFG) en el Grado en Ingeniería Informática. Los tres objetivos básicos de esta experiencia son: 1) garantizar la adquisición de las competencias específicas y transversales; 2) formar profesionales con responsabilidad social y 3) prestar un servicio adaptado a las necesidades específicas de entidades de nuestro entorno dedicadas a personas con diversidad funcional y/o cognitiva. La experiencia que se ha desarrollado a lo largo de este curso académico abarca dos proyectos realizados en colaboración con asociaciones que trabajan con personas con parálisis cerebral y con niños con trastorno de espectro autista.

Abstract

Service-Learning is a pedagogical method focused on the acquisition of transversal and specific competences with the collaboration of non-profit entities. This paper shows an experience of service-learning, initiated in the academic year 2019-2020, for the development of Bachelor's Thesis (BT) in the Degree in Computer Engineering. The three basic objectives of this experience are: 1) to ensure the acquisition of specific and transversal competences; 2) to train professionals with social responsibility and 3) to provide a service adapted to the specific needs of entities in our environment dedicated to people with functional and/or cognitive diversity. The experience that has been developed throughout this academic year includes two projects carried out in collaboration with two entities working with people with cerebral palsy and autism spectrum

disorder.

Palabras clave

Aprendizaje y Servicio; diversidad cognitiva; diversidad funcional; Trabajos Fin de Grado.

1. Motivación

En la metodología de Aprendizaje y Servicio (ApS), los estudiantes realizan actividades relacionadas con los contenidos de las asignaturas en colaboración con entidades sin ánimo de lucro, normalmente destinadas a la atención de colectivos desfavorecidos [4]. Esta última característica constituye la principal diferencia entre una actividad de ApS y una actividad curricular tradicional, lo que permite además que los estudiantes puedan adquirir competencias transversales y valores sociales.

Aunque la utilización de ApS en titulaciones universitarias lleva más de una década de recorrido, su aplicación en titulaciones relacionadas con la Ingeniería Informática sigue siendo algo muy puntual. De hecho, en nuestra universidad, la experiencia que presentamos en este artículo es la primera realizada en este ámbito. Esto concuerda con la opinión de otros autores como Sanderson que señala en [6] que “la informática no es muy visible en la comunidad de aprendizaje y servicio”. En la misma línea, Adams indica en [1] que “si bien el aprendizaje y servicio se está volviendo más común en los planes de estudios universitarios, todavía está notablemente ausente en muchos programas de informática”. Hanks indica en [2] el interés de realizar este tipo de proyectos porque en ellos “se trabaja en proyectos reales en lugar de proyectos ideados por el profesor. Esto conduce a una mayor motivación y participación de los estudiantes en los proyectos”. Por su parte, Robinson y Hall explican en [5] que “el conteni-

do se aplica en gran medida y requiere la creación de una solución (mínimamente viable) para la finalización del curso”.

En el curso académico 2019-2020, iniciamos el desarrollo de Trabajos Fin de Grado (TFG) en el marco del ApS con la colaboración de dos entidades dedicadas a personas con parálisis cerebral y Trastorno de Espectro Autista (TEA). Gracias a esto, se defendieron dos TFG en el Grado en Ingeniería Informática en septiembre de 2020 y actualmente se están realizando varios TFG de temáticas similares. Hemos dividido el desarrollo en cuatro etapas. En una primera etapa (sección 2), se inicia el contacto entre todos los agentes implicados en las actividades: estudiantes, entidad sin ánimo de lucro y profesor. En la segunda etapa (sección 3), el producto se desarrolla utilizando Agile. La tercera etapa (sección 4) es la evaluación de la experiencia, para la cual es necesario la elaboración de diferentes encuestas considerando aspectos relacionados con el ApS y la calidad del software. Finalmente, se realiza una reflexión de todo lo aprendido (sección 5).

2. Identificación de necesidades

La colaboración entre la entidad sin ánimo de lucro y la universidad es un componente indispensable en cualquier proyecto de ApS. Fue necesario firmar un convenio con las entidades a través de la Oficina de Colaboración y Voluntariado (OCV) de nuestra universidad, donde se recogen, entre otros, los siguientes puntos:

- Los proyectos de ApS son actividades curriculares que se desarrollan en el contexto de las titulaciones universitarias. Estos proyectos pueden, por tanto, desarrollarse en cualquier asignatura, incluyendo TFG y prácticas externas.
- Estos proyectos no implican ningún tipo de relación laboral entre el estudiante y la entidad, ya que la actividad desarrollada es estrictamente académica. Esta actividad debe estar en consonancia con el aprendizaje del estudiante y las competencias de la asignatura y no puede conllevar en ningún caso la sustitución de servicios profesionales.

En el curso 2019-2020, realizamos la primera oferta de TFG con metodología ApS en la convocatoria oficial del Grado en Ingeniería Informática. Uno de los TFG surgió para atender la demanda de una entidad de proporcionar cobertura a ciertas necesidades específicas del colectivo de personas con parálisis cerebral. La otra propuesta surgió por iniciativa de un estudiante a partir de su experiencia previa como voluntario en campamentos con niños que tenían algún tipo de autismo. Tras esa oferta, los dos TFG fueron asignados a sendos estudiantes, por lo que, a continuación, fue ne-

cesario conocer su situación personal, laboral y/o académica (conocimientos previos, expectativas, disponibilidad horaria, asignaturas pendientes, fecha prevista de defensa del TFG, etc.). Una vez conocidos todos estos datos, se realizó una planificación inicial de las actividades de ApS con una duración estimada total de tres meses. En el curso 2020-2021 hemos ofertado otros tres TFG, que fueron solicitados por varios estudiantes, lo que nos demuestra la alta demanda de este tipo de proyectos.

3. Desarrollo del proyecto

Se ha elegido Scrum como metodología de desarrollo de los TFG porque su empleo contribuye positivamente en esta modalidad de trabajos, especialmente en la identificación de perfiles y en el establecimiento de pautas para el seguimiento de los proyectos.

En el primer TFG del curso 2019-2020, se planteó la realización de una aplicación para terapias basadas en juegos controlados con la placa Micro:bit. A través de esta aplicación, los terapeutas pueden definir terapias basadas en juegos adaptadas a las necesidades de cada usuario y seguir su evolución temporal. Las limitaciones motoras de estos usuarios motivó la utilización de la placa Micro:bit con Bluetooth como interfaz entre los juegos y los usuarios. El estudiante tuvo que familiarizarse con la programación de esta placa y, además, puso en práctica conocimientos relacionados con bases de datos, desarrollos de interfaces e integración de sistemas. Por otro lado, el segundo TFG consistió en una agenda digital para niños con TEA que pudiesen utilizar en un móvil o tablet. La aplicación utiliza pictogramas que representan las tareas que el niño debe realizar cada día y que pueden ser configuradas por los terapeutas o las propias familias mediante una sencilla interfaz. Permite además realizar un seguimiento de las tareas realizadas por el niño y una adaptación al progreso del menor. El estudiante realizó su desarrollo en Android y utilizó de forma exhaustiva patrones para el diseño de interfaces adaptadas a colectivos con diversidad cognitiva, como un tamaño de fuentes y figuras adaptado, colores configurables, instrucciones accesibles o señalizaciones temporales para la realización de las tareas no intrusivas.

En el curso 2020-2021 existe continuidad de estos trabajos con dos nuevos TFG: “Aplicación móvil para la comunicación a través de pictogramas con niños con TEA” y “Aplicación móvil para la detección de dificultades en el aprendizaje de las matemáticas”. Se encuentran en fase inicial o intermedia de desarrollo, por lo que todavía no se ha mostrado un producto funcional a las entidades.

EVALUACIÓN DE ApS – ENTIDAD

1 Totalmente en desacuerdo, 2 en desacuerdo, 3 neutro, 4 de acuerdo, 5 totalmente de acuerdo

	1	2	3	4	5
1 El proyecto surge para cubrir una necesidad de la asociación.					
2 La comunicación entre la asociación y los miembros de la universidad ha sido frecuente y regular para mantener a todos bien informados sobre las actividades y progresos.					
3 Hemos colaborado para establecer una visión compartida y plantear metas comunes para trabajar sobre las necesidades de la comunidad.					
4 Hemos establecido de forma colaborativa planes de acción para lograr los objetivos especificados.					
5 Hemos tenido la oportunidad de compartir conocimientos y comprensión de los recursos y necesidades de la comunidad.					
6 Consideramos que el proyecto será de utilidad para los usuarios de la asociación.					

Figura 1: Encuesta a las entidades.

4. Evaluación del TFG y del ApS

Puesto que se trata de una actividad curricular, los estudiantes deben recibir una calificación. En nuestro caso, esa evaluación es realizada por un tribunal designado por el centro en el que no participan los directores de los TFG. A esta evaluación académica hay que añadir una valoración del desarrollo de la actividad de ApS. Para ello, hemos elaborado varias encuestas para las entidades y para los estudiantes que pueden verse, respectivamente, en las figuras 1 y 2. Además, hemos diseñado otra encuesta para usuarios y familias orientada a conocer su impresión sobre el producto desarrollado y su aplicación a la mejora durante su desarrollo. Por último, debemos mencionar que se ha utilizado una carta de pictogramas para que, aquellos usuarios con dificultades en la visualización o comprensión de dichas encuestas, puedan también transmitir sus valoraciones de la actividad.

Los dos TFG del curso 2019-2020 fueron evaluados en septiembre de 2020, recibiendo calificaciones de 9.5 y de 10 (sobresaliente y matrícula de honor, respectivamente). En cuanto a la encuesta realizada a las entidades (figura 1), en el caso del primer TFG, la respuesta del personal de la entidad a todas las preguntas incluidas en esa encuesta fue la de "totalmente de acuerdo". En el segundo TFG, tres terapeutas participaron en las encuestas y los resultados diferían ligeramente entre sí: una de ellas apuntó una valoración "totalmente de acuerdo" para todos los ítems requeridos, mostrando así su satisfacción con el desarrollo del trabajo; la segunda indicó la máxima calificación para todos los ítems, excluyendo los puntos 4 y 6, que recibieron solo una valoración "de acuerdo", y que están relacionados con el grado de consecución de las metas especificadas y con el uso futuro en la vida diaria de los usuarios de la entidad y, por último, la tercera indicó

EVALUACIÓN DE ApS – ESTUDIANTE

1 Totalmente en desacuerdo, 2 en desacuerdo, 3 neutro, 4 de acuerdo, 5 totalmente de acuerdo

	1	2	3	4	5
1 El proyecto tiene objetivos de aprendizaje claramente definidos.					
2 Está explícita e intencionalmente relacionado con los objetivos y contenidos de la asignatura.					
3 Me ayudará a aprender cómo transferir conocimientos y habilidades del contexto académico a la vida profesional.					
4 Me ayudará a identificar y analizar diferentes puntos de vista para mejorar la comprensión de problemas sociales y educativos.					
5 Contribuirá a desarrollar mis habilidades de resolución de conflictos y toma de decisiones en grupo.					
6 Me ayudará a comprender y valorar los historiales y contextos de quienes reciben el servicio y de mí mismo.					
7 Me estimula a reconocer y superar estereotipos.					
8 Durante la realización del proyecto, me he comprometido en la generación de ideas, en la planificación, desarrollo y evaluación del proceso de ApS.					
9 Me he involucrado en procesos de toma de decisiones.					
10 Me he involucrado en la creación de un entorno que promueva la confianza y la expresión de ideas.					
11 Me he involucrado en la evaluación de la calidad y efectividad de la actividad.					

Figura 2: Encuesta a los estudiantes.

de nuevo "totalmente de acuerdo" en todas las preguntas, pero la calificación "de acuerdo" en la número 5, relacionada con el intercambio de experiencias y conocimientos sobre las necesidades de los usuarios. Estos resultados muestran que la entidad está muy interesada en tener una mayor colaboración con los desarrolladores, lo que es completamente comprensible porque los usuarios están acostumbrados a trabajar con una planificación tradicional en papel y tienen sus dudas al cambiar de formato tan radicalmente a una plataforma móvil.

En cuanto a la evaluación realizada por los estudiantes (figura 2), el estudiante del primer TFG valoró con "totalmente de acuerdo" todos los ítems de la encuesta relacionados con competencias curriculares, pero en el caso de las competencias y valores sociales, la evaluación es "totalmente de acuerdo" siempre excepto en la pregunta 5, que valoró con un "neutro". Finalmente, con respecto al proceso, el estudiante valoró con "totalmente de acuerdo" el ítem 8 y con "de acuerdo" el resto. Este estudiante también añadió el siguiente comentario a sus valoraciones: "trabajar con colectivos desfavorecidos es siempre un trabajo agradable, principalmente por la gratitud que suelen mostrar hacia las personas que quieren ayudar, como ha sido esta experiencia. La alegría y gratitud mostrada por los usuarios del programa durante las pruebas, por sí sola, hace que valga la pena haberse tomado el tiempo para desarrollar algo como este proyecto".

El estudiante del segundo TFG otorgó las valoraciones máximas en todos los ítems, excepto en las preguntas 5 y el 7, que reciben un “de acuerdo” con los siguientes comentarios: “No veo relevante que mi proyecto esté directamente conectado a la resolución de conflictos” y “A nivel personal, en general, no soy una persona que tenga estereotipos; sin embargo, al hablar con mi entorno, me ayudó mucho a reconocer los estereotipos que tienen las personas que yo solía pasar por alto”, respectivamente referidos a las preguntas 5 y 7. Es importante señalar que este estudiante ya estaba inicialmente comprometido con este grupo, ya que había participado como voluntario temporal en campañas de diferentes organizaciones que trabajan con niños con estas dificultades.

5. Reflexión

Los TFG desarrollados, y los que están actualmente en curso, siguen una metodología de ApS y se centran en la realización de aplicaciones para ayudar a personas con parálisis cerebral y con autismo. Los resultados obtenidos en el curso 2019-2020 han sido muy positivos. Las entidades se han involucrado en todo el proceso desde el principio, lo que permitió realizar aplicaciones útiles que atienden a sus necesidades, como se refleja en las encuestas. En cuanto a la valoración recibida de los estudiantes, la respuesta también fue muy positiva, manifestando un alto grado de satisfacción, especialmente en lo que respecta a la adquisición de competencias técnicas. Los dos estudiantes se han enfrentado por primera vez al desarrollo de un proyecto real dentro de un equipo multidisciplinar. Además, uno ellos ha tenido su primer contacto con una organización destinada a personas con diversidad funcional. Aunque todavía es pronto para obtener conclusiones más genéricas sobre los TFG con ApS, el alto grado de aceptación de los TFG del curso 2020-2021 y el excelente trabajo que están realizando los estudiantes nos hace pensar que los resultados serán altamente satisfactorios. También estamos muy satisfechos de la repercusión que han tenido los proyectos en distintos ámbitos. El TFG basado en Micro:bit fue finalista de los premios de la Maker Faire Galicia 2020 y fue presentado en el congreso XoveTIC 2020 [3]. Se espera que este proyecto sea integrado este año en las terapias de la entidad. El TFG para la creación de una agenda digital recibió el primer premio de cooperación al desarrollo de entre todos los TFG presentados de distintos centros de nuestra universidad y el primer premio al mejor TFG aplicado de nuestra facultad. Se ha realizado un registro de software donde se ha indicado su uso gratuito por entidades sin ánimo de lucro.

La realización de TFG con ApS requiere un esfuerzo adicional y la adquisición de ciertos compromisos:

1) requiere tiempo extra y la búsqueda de una entidad adecuada; 2) no es fácil conciliar los horarios de todos los agentes implicados; 3) las actividades deben concluirse en un tiempo limitado (normalmente un cuatrimestre) y 4) el hecho de involucrar a entidades puede generar expectativas en grupos vulnerables que deben ser tratadas con sumo cuidado. La realización de los dos TFG del curso 2019-2020 y de los actuales, se ha visto afectada por las limitaciones derivadas de la pandemia producida por la COVID-19, obligando a reuniones en remoto y pruebas no presenciales en algunos casos. Sin embargo, valorando sus ventajas e inconvenientes, consideramos que los beneficios obtenidos a partir de esta experiencia de ApS compensan el esfuerzo invertido por parte del profesorado, el estudiantado y el personal de las entidades.

Agradecimientos

Este trabajo fue financiado parcialmente por Xunta de Galicia (ED431C 2020/15, ED431G2019/01), la Agencia Estatal de Investigación de España (RED2018-102668-T y PID2019-104958RB-C42) y fondos ERDF de la EU (FEDER Galicia 2014-2020 & AEI/FEDER Programs, UE). Agradecemos la inestimable colaboración de las entidades ASPACE Coruña y DISMACOR.

Referencias

- [1] Adams, J. B., Runkles, E. May we have class outside? implementing service learning in a CS1 curriculum. *Journal of Computing Sciences in Colleges*, 19(5): 25-34, 2004.
- [2] Hanks, B. Becoming agile using service learning in the software engineering course. *Agile 2007*, 121-127, 2007.
- [3] Lopez-Fernandez, A.; Carneiro-Medin, R.; Pousada, T.; Groba-González, B.; Dapena, A. Development of Recreational Content with Micro:Bit for Intervention with People with Cerebral Palsy. En *Proceedings 2020*, 54, 58. <https://doi.org/10.3390/proceedings2020054058>
- [4] Nejme, B. A. Service-learning in the computer and information sciences: Practical applications in engineering education. John Wiley & Sons, 2012.
- [5] Robinson, S., y Hall, M. Combining agile software development and service-learning: A case study in experiential IS education. En *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, 491-496, 2018.
- [6] Sanderson, P. Where's (the) computer science in service-learning? *Journal of Computing Sciences in Colleges*, 19(1): 83-89, 2003.

vylnoviolencia.uv.es: trabajo fin de máster en TIC aplicado a una experiencia ApS en Humanidades Digitales

Raúl Peña-Ortiz
Departament d'Informàtica
Universitat de València
raul.penya@uv.es

Danilo A. Salaz-Pulla
Escola Tècnica Superior d'Enginyeria
Universitat de València
dasapu@alumni.uv.es

Josefa Badía Herrera
Departamento de Filología Española
Universitat de València
josefa.badia@uv.es

Resumen

En el camino hacia una sociedad tolerante e igualitaria, en la que se respeten los derechos y libertades fundamentales y de igualdad entre hombres y mujeres, resulta esencial una implicación efectiva de la comunidad universitaria, en su conjunto, en la reflexión crítica y en el desarrollo de acciones concretas. Alineado con este reto, «*Voces y letras contra la violencia: imaginarios literarios y creación en videoarte*» es un proyecto de innovación docente que apuesta por la aplicación de la metodología aprendizaje-servicio al ámbito de las Humanidades Digitales para afianzar el compromiso ético del alumnado con el respeto, la no violencia, la igualdad y la solidaridad como principios fundamentales en la convivencia. Este artículo describe los resultados obtenidos por el trabajo fin de máster bajo el cual se ha desarrollado la plataforma digital distribuida que relaciona videoarte y literatura en el proyecto.

Abstract

With the aim of achieving a tolerant and egalitarian society, which respects basic human rights and especially the right of equality between men and women, is essential an effective involvement of the university community, as a whole, in critical reflection and by developing specific actions. Aligned with this challenge, a teaching innovation project, named «*Voces y letras contra la violencia: imaginarios literarios y creación en videoarte*», is committed to the application of service-learning methodology in Digital Humanities to strengthen the students commitment to respect, non-violence, equality and solidarity as fundamental principles in coexistence. This paper describes the results of a master's thesis under which a distributed digital platform has been developed to link video art and literature related to the project.

Palabras clave

ApS, computación en la nube, humanidades digitales, literatura, no violencia, TFM, TIC, videoarte.

1. Introducción

En el marco de la sociedad de la información y del conocimiento, ante el riesgo de que la educación superior pueda convertirse en un adiestramiento profesional en detrimento de una preparación más amplia, se hace necesario aplicar metodologías que garanticen un aprendizaje integral que atienda a la dimensión ontológica (saber ser), pragmática (saber hacer), pedagógica (aprender a aprender) y democrática (aprender a convivir) [6].

En este contexto, se ha planteado la aplicación de la metodología aprendizaje-servicio (ApS) al ámbito de las Humanidades Digitales (HHDD) a través de «*Voces y letras contra la violencia: imaginarios literarios y creación en videoarte*»¹, proyecto de innovación docente (PID) sobre patrimonio literario y videoarte que apuesta por afianzar el compromiso ético de los estudiantes con el respeto, la no violencia, la igualdad y la solidaridad como principios fundamentales para la convivencia. La plataforma digital distribuida² que relaciona videoarte y literatura ha sido desarrollada como parte de un trabajo fin de máster (TFM) en el ámbito de las tecnologías de la información y la comunicación (TIC) dentro del PID. Este artículo describe los resultados obtenidos por el citado TFM.

El resto del artículo se estructura como sigue. La sección 2 describe el proceso metodológico adoptado en el PID y el TFM, mientras que la sección 3 muestra los resultados obtenidos. Finalmente, la sección 4 presenta conclusiones y trabajo futuro.

2. Proceso metodológico

ApS es definido por Puig et al. [8] como "una propuesta educativa que combina procesos de aprendizaje y de servicio a la comunidad en un solo proyecto bien articulado en el que los participantes se forman al trabajar sobre necesidades reales del entorno con el objeti-

¹UV-SFPIES_PID19-1098163

²<http://vylnoviolencia.uv.es>

vo de mejorarlo”. Varias décadas antes, Robert Sigmon [9] propuso los tres principios fundamentales del ApS para garantizar que aquellos que sirven también son aprendices y se les debe permitir un control significativo sobre su aprendizaje (tercer principio), de manera que quienes reciben servicios (miembros de la comunidad) deben controlar los servicios brindados (primer principio) y deben estar en mejores condiciones de prestar servicio como resultado del ApS (segundo principio). En consideración a los mismos, Furco [3] distinguió entre tres perspectivas educativas en el servicio de aprendizaje atendiendo al beneficiario, al enfoque adoptado y a su propósito explícito, tal y como muestra la Tabla 1. Únicamente el ApS garantiza los tres principios anteriores al estar centrado en el aprendizaje del estudiante, que desarrolla o perfecciona tanto competencias técnicas de sus estudios como competencias profesionales y cívicas, mientras lleva a cabo una actividad integrable en su currículo y con un impacto positivo en los miembros de la comunidad.

A lo largo de los años, han sido muchas las propuestas, experiencias y reflexiones en la implantación de ApS en los estudios universitarios [1]. Por otro lado, son muchas las experiencias de voluntariado y cooperación que han sido desarrolladas como proyectos ApS por estudiantes de ingenierías relacionadas con las TIC [10]. Sin embargo, muchos de estos proyectos han sido considerados erróneamente experiencias ApS porque, aunque ofrecen un servicio a la comunidad, están más enfocados en conseguir solo un desarrollo profesional y académico, pero no cívico, del estudiante [2].

«*Voces y letras contra la violencia: imaginarios literarios y creación en videoarte*» sitúa su punto de partida en la importancia de generar dinámicas de apropiación patrimonial que fomenten el compromiso ético y favorezcan la reflexión sobre las distintas formas de violencia. Los estudiantes universitarios prestan su servicio en institutos de enseñanza secundaria. Durante el proceso formativo, se impulsa el trabajo cooperativo y la autonomía del alumnado en la selección de los textos literarios, en los intercambios de lecturas y en la negociación de significados para definir los imaginarios literarios que serán utilizados. Sobre la base de la selección textual, se generan muestras de videoarte, destinadas a servir de apoyo en las dinámicas de interacción en las aulas.

Para conseguir una ampliación de la capacidad de transferencia del servicio, se ha apostado por el desa-

rollo de una plataforma digital distribuida que relaciona videoarte y literatura. La plataforma, desplegada en la nube privada del Servicio de Informática de la Universitat de València (UV), ha sido desarrollada por un estudiante del Máster oficial en Tecnologías Web, Computación en la Nube y Aplicaciones Móviles (TW-CAM)³ como parte de su TFM.

3. Resultados

En el primer cuatrimestre del curso 2019-2020, la experiencia de servicio se llevó a cabo en 13 IES de la Comunitat Valenciana durante la última semana de noviembre, haciéndola coincidir con la celebración del Día Internacional de la Eliminación de la Violencia contra la Mujer. La acción de servicio durante el segundo cuatrimestre se había programado inicialmente entre la última semana de abril y la primera semana de mayo en 21 centros educativos de secundaria, con motivo de la conmemoración del Día Internacional Contra el Acoso Escolar. Sin embargo, tuvo que ser suspendida por el confinamiento.

De acuerdo a datos de la UNESCO [7], en abril de 2020 se cerraron escuelas e instituciones de educación superior en 185 países, lo que afectó al 89,4 % del total de matrículas. De igual manera las iniciativas de proyectos ApS se vieron afectadas y se tuvieron que reinventar ante la situación de emergencia sanitaria en todo el mundo [4].

Nuestro PID no fue una excepción. La situación coyuntural modificó la prestación del servicio y trajo como consecuencia más destacable una mayor centralidad de los objetos digitales creados como soportes de la acción. En este sentido, las muestras de videoarte creadas por los estudiantes de humanidades fueron enviadas a los profesores responsables de la acción de servicio en los IES y, en algunos casos, se expusieron a través de herramientas de videollamadas a modo de ApS “digital” en tiempos de emergencia sanitaria.

Al asumir dichas muestras un papel fundamental como elementos desencadenantes de la acción de servicio, la plataforma distribuida que había sido pensada inicialmente como una herramienta para documentar el histórico de las experiencias ApS llevadas a cabo, pasó a asumir un rol de facilitador del servicio en un contexto de docencia no presencial.

³<http://www.uv.es/twcam>

	Voluntariado	ApS	Prácticas externas
Beneficiario	Miembros de la comunidad	Miembros de la comunidad y estudiantes	Estudiantes
Enfoque	Servicio	Servicio y aprendizaje	Aprendizaje
Propósito	Justicia social y desarrollo cívico	Desarrollo cívico y académico	Desarrollo profesional y académico

Tabla 1: Perspectivas educativas en el servicio de aprendizaje

En el marco del proyecto, estaba previsto que la plataforma digital distribuida ofreciera dos secciones principales: imaginarios y vídeos, tal y como muestra la Figura 1. En la primera se presenta una colección de textos literarios, seleccionados por los estudiantes, a través de los que se pueden analizar las representaciones simbólicas de la violencia. Mientras que la segunda recopila una colección de muestras de videoarte creadas por los alumnos, a través de los que se activa la reflexión sobre las representaciones de la violencia en el arte y la cultura.

En el nuevo contexto, se procedió a rediseñar algunas de sus funcionalidades para garantizar su eficacia. La plataforma ofrece tratamiento lematizado, categorización y etiquetado del conjunto de objetos, de modo que permite efectuar búsquedas conceptuales que sirven para correlacionar imaginarios simbólicos en torno al concepto de violencia. Con ello, se consigue "expandir" la función de las muestras de videoarte como objetos desencadenantes de la acción de servicio. El procesamiento de datos en la plataforma y la interacción con los recursos favorece la reflexión crítica sobre las formas simbólicas de la violencia en los imaginarios literarios y artísticos tanto en nuestro pasado como en el presente. Además, la creación colaborativa de los diversos objetos necesarios para que se desarrolle de manera efectiva la acción de servicio (presencial o virtual) no solo ha convertido a los estudiantes en protagonistas del desarrollo de las distintas tareas, sino que ha favorecido también un modelo cooperativo de trabajo interdisciplinar en el proceso de aprendizaje.

El estudiante de máster que desarrolló la plataforma distribuida tuvo éxito en la adquisición de las competencias técnicas, profesionales y éticas recogidas en la guía docente de la asignatura de TFM del Máster TW-CAM [5] gracias, en parte, a la realización de un proyecto que ha beneficiado a la comunidad de las aulas (universitarias y de secundaria) y ha permitido potenciar los valores éticos y cívicos de todos y todas (incluido el propio estudiante del TFM).

De entre todas estas competencias y habilidades adquiridas por el estudiante de TFM podemos destacar tres. En primer lugar, la aplicación de los conocimientos adquiridos en desarrollo de aplicaciones web y despliegue de sistemas distribuidos para el procesamiento de vídeo en la nube en la resolución de un nuevo problema real en el ámbito multidisciplinar de las HHDD. La Figura 1 muestra varias capturas de la aplicación de acceso público que relaciona imaginarios y videoarte, mientras que la Figura 2a es una captura de la zona de administración privada de los mismos y la Figura 2b representa gráficamente la arquitectura distribuida en nube para ambas aplicaciones y el procesamiento del videoarte. En segundo lugar, la capacidad autónoma para la elaboración, planificación, dirección, coordina-

ción, gestión técnica y económica e implantación de un proyecto TIC reflexionando sobre las responsabilidades sociales y éticas vinculadas al mismo. En tercer lugar, fomentar en contextos académicos y profesionales, el avance tecnológico, social o cultural dentro de una sociedad basada en el conocimiento y en el respeto a: a) los derechos fundamentales y de igualdad de oportunidades entre hombres y mujeres, b) los principios de igualdad de oportunidades y accesibilidad universal de las personas con discapacidad y c) los valores propios de una cultura de paz y de valores democráticos.

Finalmente, resaltamos que el TFM no puede ser considerado como una experiencia ApS en todos sus aspectos. Si bien es cierto que los beneficiarios de la plataforma son miembros de la comunidad universitaria y el enfoque del PID es propio del ApS, en los objetivos del TFM se ha priorizado lo académico ante lo cívico. Esta reflexión servirá para planificar futuros proyectos ApS en el ámbito TIC.

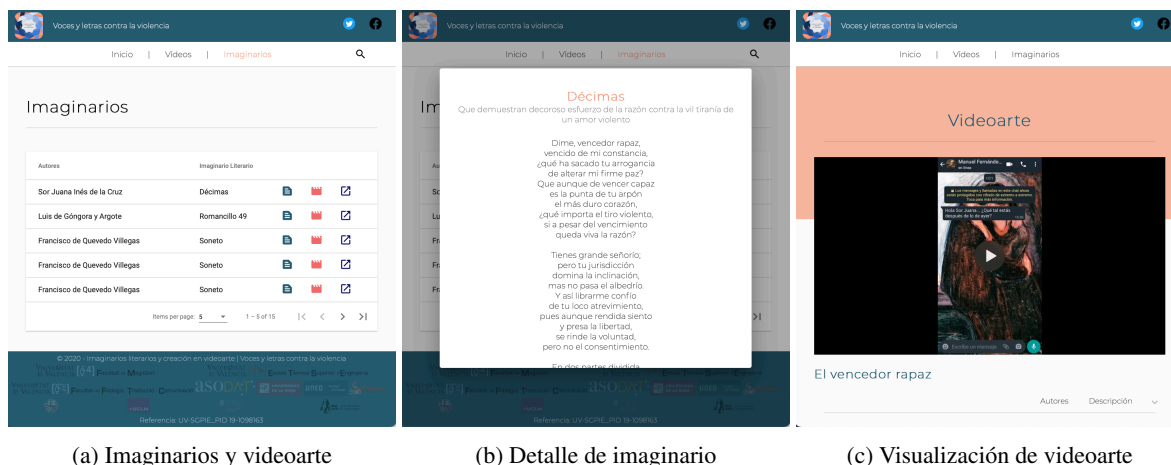
4. Conclusiones y trabajo futuro

La Universidad está llamada a ser referente en la construcción identitaria y cultural, en la transmisión esencial de valores y en la promoción de cambios sociales.

El ApS es una metodología que facilita esta acción social desde las aulas, a fin de conseguir formar no solo profesionales cualificados, sino personas comprometidas, reflexivas y críticas.

El presente artículo ha presentado los primeros resultados y reflexiones de un proyecto de innovación docente sobre literatura y videoarte que apuesta por la aplicación de la metodología ApS al ámbito de las Humanidades Digitales para afianzar el compromiso ético de los estudiantes con el respeto, la no violencia, la igualdad y la solidaridad como principios fundamentales en la convivencia. Concretamente, el artículo se ha centrado en presentar la experiencia del trabajo final del Máster de Tecnologías Web, Computación en la Nube y Aplicaciones Móviles que ha desarrollado la plataforma digital distribuida que relaciona videoarte y literatura.

El proyecto de innovación docente pretendía activar la reflexión crítica del conjunto de estudiantes sobre las distintas formas de violencia a partir de la lectura de textos literarios y de su adaptación al formato audiovisual. El procesamiento de los datos que ofrece la plataforma, así como las nuevas funcionalidades en el acceso, recuperación y correlación de la información favorecen la activación del pensamiento crítico entre los destinatarios del servicio que se presta. Por ello consideramos que el TFM ofrece mejoras significativas en la acción de servicio en el actual contexto digital derivado de la situación de pandemia.



(a) Imaginarios y videoarte

(b) Detalle de imaginario

(c) Visualización de videoarte

Figura 1: Contenidos públicos en *vlnoviencia.uv.es*

(a) Usuarios y contenidos

(b) Infraestructuras

Figura 2: Zona privada de administración de *vlnoviencia.uv.es*

Referencias

- [1] Robert G. Bringle y Julie A. Hatcher: *Institutionalization of service learning in higher education*. The Journal of Higher Education, 2000.
- [2] Randy W. Connolly: *Is There Service in Computing Service Learning?* En *43rd ACM Technical Symposium on Computer Science Education*, 2012.
- [3] Andrew Furco: *Is Service-Learning Really Better than Community Service?* En *Service-Learning: The Essence of the Pedagogy*. 2002.
- [4] Lauren Grenier, Elizabeth Robinson y Debra A. Harkins: *Service-learning in the COVID19 era: Learning in the midst of crisis*. Pedagogy and the Human Sciences, 7(1), 2020.
- [5] Juan Gutiérrez-Aguado: *44836 - TFM en Tecnologías Web, Computación en la Nube y Aplicaciones Móviles*. Guía Docente, Universitat de València, 2020-2021.
- [6] Oswaldo Leyva Cordero, Francisco Ganga Contreras, José Tejada Fernández y Abraham A. Hernández Paz: *La formación por competencias en la educación superior: alcances y limitaciones desde referentes de México, España y Chile*. 2015.
- [7] Giorgio Marinoni, Hilligje Van't Land y Trine Jensen: *The impact of Covid-19 on higher education around the world*. IAU Global Report, 2020.
- [8] Josep Maria Puig, Roser Battle, Carmen Bosch y Josep Palos: *Aprendizaje servicio: Educar para la ciudadanía*. 2007.
- [9] Robert Sigmon: *Service-Learning: Three Principles*. Synergist, 1979.
- [10] Fermín Sánchez-Carracedo, Emiliana Marqués, Xavier Ortega, Andreu Feliu y Eva Vendrell: *El voluntariado TIC como forma de aprendizaje-servicio*. En *XXIII Jornadas sobre la Enseñanza Universitaria de la Informática*, 2017.

Factors Considered in the Assessment of Computer Science Engineering Capstone Projects and their Influence on Discrepancies between Assessors

César Domínguez, Arturo Jaime, Francisco J. García-Izquierdo, Juan J. Olarte
Departamento de Matemáticas y Computación
Universidad de La Rioja
26006 Logroño
{cesar.dominguez, arturo.jaime, francisco.garcia, jjolarte}@unirioja.es

Palabras clave: Capstone Project, assessment, computer science.

Lugar de publicación: ACM Transactions on Computing Education, Vol. 20, No. 2, pp. 12-45, 2020.

Índice de impacto: JCR 1,721, Cuartil Q2, – Posición:20/42 – Área: Education / Scientific Disciplines.

DOI: <https://doi.org/10.1145/3381836>.

Abstract

A capstone project is an extensive learning experience traditionally developed during a student's final academic year. Assessing such a complex assignment involves several challenges and is usually based upon the evaluations of at least two different people: the capstone project advisor, and one or more other assessors. Quantitative studies comparing only different assessors' grades and qualitative studies investigating the origin of possible discrepancies have been conducted. In both cases, contradictory conclusions were reached. The objective of this study is to analyze the factors that are given consideration by assessors of engineering capstone projects and the influence of these factors on the discrepancies between different assessors' opinions of the same project. This study quantitatively examined 162 computer science engineering capstone projects developed by one student and supervised by one advisor. Each project was assessed by the project advisor and a committee. For each project, the advisor and the committee were asked to complete an additional questionnaire on product characteristics, student competences, and project supervision. Competences demonstrated by the student were found to be the most relevant element when a capstone project was evaluated by the advisor and the committee; product characteristics were found to be second in influence. Furthermore, advisors grant minor significance to the advisor-involvement component. Discrepancies between grades seem to be associated with those aspects to which one assessor has access, while the other does not, such as student skills demonstrated during project development or their performance in the oral defense. Both the advisor's and the committee's perspectives are important in the assessment of this complex task and they complement one another.

Girls4STEM: Gender Diversity in STEM for a Sustainable Future

Xaro Benavent¹, Esther de Ves¹, Anabel Forte², Carmen Botella-Mascarell¹, Emilia López-Iñesta³, Silvia Rueda¹, Sandra Roger¹, Joaquin Perez⁴, Cristina Portalés¹, Esther Dura¹, Daniel Garcia-Costa¹ and Paula Marzal⁵

¹Computer Science Department, Universitat de València, Av. de la Universitat s/n, 46100 Burjassot, Spain

²Department of Statistics and Operational Research, Universitat de València, Dr. Moliner 50, 46100 Burjassot, Spain

³Department of Didactics of Mathematics, Universitat de València, Av. Tarongers 4, 46022 València, Spain

⁴Department of Electronic Engineering, Universitat de València, Av. de la Universitat s/n, 46100 Burjassot, Spain

⁵Department of Chemical Engineering, Universitat de València, Av. de la Universitat s/n, 46100 Burjassot, Spain

Palabras clave: STEM; ICT; Gender diversity; Self-efficacy in STEM

Lugar de publicación: Sustainability, Vol. 12, No.(15), 6051, 2020

DOI: <https://doi.org/10.3390/su12156051>

Índice de impacto: JCR 2.576, Cuartil Q2, Posición: 120/265 – Área: Environmental Sciences

Abstract

Science, Technology, Engineering, and Mathematics (STEM) are key disciplines towards tackling the challenges related to the Sustainable Development Goals. However, evidence shows that women are enrolling in these disciplines in a smaller percentage than men, especially in Engineering related fields. As stated by the United Nations Women section, increasing the number of women studying and working in STEM fields is fundamental towards achieving better solutions to the global challenges, since the potential for innovation is larger. In this paper, we present the Girls4STEM project, which started in 2019 at the Escola Tècnica Superior d'Enginyeria de la Universitat de València, Spain. This project works towards breaking the stereotypes linked to STEM fields, addressing both boys and girls aged from 6 to 18, but especially trying to open the range of career options for young girls through interaction with female STEM experts. The goal is to spark girls' interest in STEM disciplines from childhood, so that they become more self-confident in these areas. To achieve this goal, the project is built over three main actions: the Girls4STEM Family Talks, where students, families, and teachers participate; the Girls4STEM Professional Talks, where the target is a general audience; and the Initial Training Seminars for teachers. Short-term results are here presented, showing that aspects related to self-perception and perception from others (family, teachers) play a significant role. Moreover, these results also indicate that there may not be a general understanding of which disciplines are included in STEM.

Insights from Learning Analytics for Hands-On Cloud Computing Labs in AWS

Germán Moltó, Diana M. Naranjo, J. Damian Segrelles
Instituto de Instrumentación para Imagen Molecular (I3M).
Centro mixto CSIC - Universitat Politècnica de València
Camino de Vera s/n, 46022 Valencia, España
{gmolto, dquilis}@dsic.upv.es, dnaranjo@i3m.upv.es

Palabras clave: learning analytics, cloud computing

Lugar de publicación: Applied Sciences, Vol. 10, No. 34, 9148

Índice de impacto: JCR 2,474, Cuartil Q2, -Posición: 32/91 - Área: Engineering, Multidisciplinary

DOI: <https://doi.org/10.3390/app10249148>

Abstract

Cloud computing instruction requires hands-on experience with a myriad of distributed computing services from a public cloud provider. Tracking the progress of the students, especially for online courses, requires one to automatically gather evidence and produce learning analytics in order to further determine the behavior and performance of students. With this aim, this paper describes the experience from an online course in cloud computing with Amazon Web Services on the creation of an open-source data processing tool to systematically obtain learning analytics related to the hands-on activities carried out throughout the course. These data, combined with the data obtained from the learning management system, have allowed the better characterization of the behavior of students in the course. Insights from a population of more than 420 online students through three academic years have been assessed, the dataset has been released for increased reproducibility. The results corroborate that course length has an impact on online students dropout. In addition, a gender analysis pointed out that there are no statistically significant differences in the final marks between genders, but women show an increased degree of commitment with the activities planned in the course.

