

Un Modelo de Evaluación de Prácticas en Laboratorio de Ingeniería del Software

A. Garrido, M.C. Penadés, V. Pelechano

Dpto. Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
Camino de Vera, s/n, 46071 - Valencia
e-mail: {agarridot, mpenades, pele}@dsic.upv.es

Resumen

En este trabajo se presenta un modelo de evaluación de las prácticas realizadas en la asignatura Laboratorio de Ingeniería del Software que se imparte en la titulación de Ingeniería Técnica en Informática de la Escuela Universitaria de Informática de la Universidad Politécnica de Valencia. En estas prácticas, los alumnos deben desarrollar un proyecto software, cubriendo todas las fases del ciclo de vida, desde la especificación de requisitos de la aplicación hasta la documentación y entrega del producto final. Este planteamiento de las prácticas se complementa con la evaluación que se realiza de las mismas. La idea es que los alumnos, además de desarrolladores, desempeñen también el papel de evaluadores del producto y evalúen los proyectos realizados por otros alumnos. De esta forma, en la nota final de las prácticas, el profesor evalúa dos aspectos: el trabajo realizado y la capacidad de evaluar un proyecto software. Además, en la calificación del trabajo realizado, se añade a la evaluación del profesor, la evaluación realizada por otros alumnos. Esta forma de evaluación se ha puesto en práctica en los dos últimos cursos académicos, mostrándose los resultados obtenidos.

1. Introducción

Las necesidades actuales en el mercado laboral de desarrollo de software exigen unos conocimientos tanto teóricos como prácticos en el campo de la ingeniería del software. Ante estas exigencias, es importante incluir en la formación de los futuros profesionales del software métodos y técnicas que

los capaciten como buenos técnicos del software [Error!Argumento de modificador desconocido.]. Esto incluye las nuevas tecnologías, el desarrollo de aplicaciones cliente/servidor que se ejecuten a través de Intranets o de Internet, el conocimiento básico sobre gestión de proyectos, el diseño, modelado e implementación de sistemas utilizando el paradigma de la orientación a objetos, etc. Por lo tanto, resulta evidente la necesidad de dar una mayor importancia a la parte práctica de la ingeniería del software, dando un mayor peso a las prácticas de laboratorio e incluso incorporando asignaturas eminentemente prácticas en el contexto de la ingeniería del software [Error!Argumento de modificador desconocido.]. Como consecuencia directa de esto, en el plan de estudios de la Escuela Universitaria de Informática de la Universidad Politécnica de Valencia se incorporó la asignatura de Laboratorio de Ingeniería del Software (en adelante LIS). En dicha asignatura las prácticas de laboratorio cobran una especial relevancia, consiguiendo que los alumnos desarrollen un proyecto software completo. En este proceso de desarrollo, los alumnos pasan por todas las etapas que constituyen un proyecto software hasta llegar a la entrega del producto final, como son: i) planificación del proyecto, ii) especificación de requisitos, iii) análisis/modelado, iv) diseño, v) implementación/pruebas y vi) documentación (con la obtención del manual de usuario).

La realización de estas prácticas de laboratorio, totalmente aplicadas e intentando simular en todo lo posible (teniendo en cuenta las limitaciones existentes) el desarrollo de un proyecto software, pensamos que es el mejor enfoque que se le puede dar a los créditos prácticos de una asignatura como LIS. Sin

embargo, pensamos que esto también supone tener que introducir un modelo de evaluación distinto al seguido en los enfoques tradicionales. La evaluación no debe basarse tanto en la entrega de memorias, sino en una forma de que los alumnos obtengan una realimentación de provecho y puedan aprender de los errores cometidos por ellos mismos y por sus compañeros: *'hay que evitar la tendencia a escribir memorias, pues las prácticas son un elemento de comunicación e intercambio de información no de papel'* [**Error!Argumento de modificador desconocido.**].

2. Laboratorio de Ingeniería del Software en Ingeniería Técnica en Informática

La asignatura LIS está enmarcada en el tercer curso (sexto cuatrimestre) de la titulación de Ingeniería Técnica en Informática por la Escuela Universitaria de Informática de la Universidad Politécnica de Valencia, contando con 3 créditos teóricos y 3 prácticos. Se trata de una asignatura troncal para los alumnos que cursan la especialidad de gestión (ITIG) y optativa para los que cursan la especialidad de sistemas (ITIS) impartida por profesores del Departamento de Sistemas Informáticos y Computación (DSIC).

Esta asignatura pertenece al grupo de asignaturas relacionadas con la Ingeniería del Software impartidas en la Universidad Politécnica de Valencia, tanto en la titulación de Ingeniería Técnica en Informática como en la de Ingeniería (superior) en Informática, tal y como se presenta en [**Error!Argumento de modificador desconocido.**]. LIS pretende ser una continuación, más práctica y aplicada de la asignatura Ingeniería del Software ([**Error!Argumento de modificador desconocido.**], [**Error!Argumento de modificador desconocido.**]) que intenta dar soluciones y educar a los futuros técnicos del software, con el objetivo de proporcionarles los métodos y técnicas suficientes para desenvolverse en el mercado laboral. Debido a que la asignatura se imparte en el último cuatrimestre de la titulación y la mayoría de alumnos no pretende continuar con sus estudios en un segundo ciclo, LIS está planteada con un enfoque eminentemente práctico, enseñando a los alumnos a trabajar de

forma ingenieril y disciplinada en el desarrollo de aplicaciones software.

Puesto que en [**Error!Argumento de modificador desconocido.**] se expone con mayor detalle el contenido teórico de la asignatura LIS, indicando sus prerequisites, contenidos y fundamentos, en este artículo se hará especial hincapié en la parte de prácticas de laboratorio.

3. El Papel de las Prácticas de Laboratorio en la Asignatura

Tradicionalmente, donde los alumnos mejor asimilan los conceptos presentados en las clases teóricas es en las prácticas de laboratorio. Por lo tanto, en las prácticas de la asignatura se pretende poner de manifiesto la importancia de seguir y cumplir un método correcto en el desarrollo completo de un proyecto software. Así, los objetivos que se pretenden poner en práctica son, entre otros, los siguientes:

- elaborar una correcta planificación de las etapas del proyecto software a desarrollar. Aunque el objetivo no es que los alumnos aprendan las técnicas de gestión de proyectos, sí se pretende ilustrar la forma en que se pueden mejorar los calendarios de ejecución del proyecto: estudio del camino crítico, gestión de los recursos disponibles, satisfacción de las restricciones temporales (plazos de entrega), etc. Esto se plantea en las prácticas como una serie de hitos que los alumnos deben alcanzar y entregar en unas fechas señaladas desde el comienzo de la asignatura.
- determinar y especificar los requisitos software de acuerdo a la guía IEEE-830 para la especificación de requisitos software. No se trata de que estudien a fondo técnicas de captura de requisitos (tarea más adecuada para un ingeniero superior de software que para un ingeniero técnico en un equipo de desarrollo [**Error!Argumento de modificador desconocido.**], [**Error!Argumento de modificador desconocido.**]), pero sí que sean capaces de identificar los requisitos mínimos que debería cumplir la aplicación a desarrollar. En las prácticas, se les proporciona un enunciado

con el caso de estudio que deben desarrollar. Sobre este caso de estudio, los alumnos deben realizar la especificación de requisitos software, consultando cualquier duda al profesor que desempeña el rol del cliente.

- aplicar las metodologías de análisis y diseño orientado a objetos, concretamente OMT [**Error!Argumento de modificador desconocido.**]. De esta forma los alumnos llevan a cabo un proceso de diseño de acuerdo al paradigma orientado a objetos con el objetivo de advertir cuáles son los beneficios que supone este tipo de modelado. Por ejemplo, el modelo de objetos realizado permite una traducción casi inmediata a un modelo relacional de bases de datos con el objetivo de satisfacer la persistencia de los objetos a mantener en el caso de estudio propuesto.
- conocer y utilizar uno de los entornos de desarrollo rápido de aplicaciones (RAD) visuales orientados a objetos. El objetivo aquí es introducir este tipo de herramientas como un medio que les facilite el desarrollo de las aplicaciones, pero no como un fin propio, por lo que se introducen los conceptos comunes a todo este tipo de entornos. Gracias a este tipo de herramientas, los alumnos pueden desarrollar rápidamente un prototipo horizontal y entregarlo al profesor, quien de nuevo en el rol del cliente, se encarga de validarlo de acuerdo a los requisitos iniciales presentes en el enunciado del caso de estudio. Puesto que el desarrollo final lo realizan en el entorno *Borland Delphi* [**Error!Argumento de modificador desconocido.**], también se proporcionan algunos conceptos específicos de este entorno de desarrollo. La elección de *Borland Delphi* se basa en que su lenguaje de programación *Object Pascal* permite poner en práctica los conceptos básicos de la orientación a objetos: definición de clases, herencia, polimorfismo, etc. Además, el lenguaje de programación *Pascal* ya es conocido por los alumnos porque la mayoría de las prácticas realizadas en las asignaturas cursadas en semestres anteriores dentro de la titulación se desarrollan en este lenguaje.
- implementar, en un lenguaje de programación orientado a objetos la aplicación propuesta en

el caso de estudio, documentando correctamente el código implementado.

- generación de un manual de usuario que permita al usuario final de la aplicación conocer el funcionamiento básico de la aplicación.

Contenido de las Prácticas de Laboratorio

La planificación de las prácticas realizadas durante el curso es la siguiente:

Práctica 1. Introducción a la programación en el IDE de *Borland Delphi*. Esta primera práctica tiene como objetivo que los alumnos se familiaricen con el entorno integrado de desarrollo de aplicaciones visuales *Delphi*, desarrollando una sencilla aplicación.

Práctica 2. Desarrollo de un sencillo editor de textos MDI (Interfaz de Múltiples Documentos), que permite a los alumnos afianzar sus conocimientos en el manejo de ventanas, menús, barras de herramientas, cuadros de diálogo estándar de *Windows*, gestión de eventos y asignación de métodos de respuesta, etc.

Práctica 3. Desarrollo de sencillos ejemplos que manejan bases de datos relacionales. Mediante estos ejemplos los alumnos aprenden a realizar aplicaciones que trabajan con bases de datos, y así adquieren los conocimientos necesarios para poder abordar el desarrollo de una aplicación de mayor envergadura

Es importante señalar que estas tres primeras prácticas de la asignatura pretenden introducir a los alumnos en la filosofía de programación guiada por eventos (manejando, incluso, bases de datos relacionales) y en el manejo del entorno de desarrollo en el que van a desarrollar el proyecto software final.

La *Práctica 4* de la asignatura constituye el proyecto final a desarrollar, siendo el objetivo principal de las prácticas de LIS. Concretamente, esta práctica final consiste en el desarrollo de una aplicación que permita aplicar la mayoría de las técnicas necesarias para el desarrollo de aplicaciones de gestión utilizando bases de datos relacionales en entornos de desarrollo visuales (en nuestro caso *Delphi*). Los proyectos a desarrollar consisten en típicas aplicaciones de gestión que los alumnos conocen y que, por tanto, no precisan de una extensa etapa para la especificación de

requisitos. Por ejemplo, en el curso académico pasado 99-00 se les propuso la realización de una aplicación informática que fuera capaz de gestionar todos los datos relacionados con la matrícula de los alumnos de un centro de enseñanza. Se da el caso de que en la Universidad Politécnica de Valencia se viene utilizando desde hace varios años un sistema de automatrícula para que los alumnos realicen el proceso de matrícula ellos mismos y de forma totalmente automática. Así, los alumnos no se enfrentan a un problema totalmente desconocido, sino que conocen los requerimientos básicos, así como los escenarios de ejecución de una aplicación muy similar, pudiendo incluir cualquier nueva aportación que consideren oportuna.

El desarrollo de la práctica final se realiza en grupos de dos personas. Con ello se pretende fomentar el trabajo en equipo, pero intentando minimizar al máximo los inconvenientes de los grupos numerosos en los que surgen grandes diferencias, tanto en lo referente a la motivación e interés de los alumnos por su trabajo en la asignatura como a sus conocimientos básicos.

4. Modelo de Evaluación

Las prácticas de la asignatura son de carácter obligatorio, por lo que se controla la asistencia. De un total de, aproximadamente, 12 sesiones de 2 horas sólo se permite un máximo de 3 faltas sin justificar. De esta forma, el profesor puede llevar un seguimiento continuo del trabajo desarrollado por el alumno. Además, con el objetivo de facilitar la evaluación de las prácticas de la asignatura, se exigen las siguientes entregas en unas fechas determinadas:

- especificación de requisitos de la aplicación propuesta en el caso de estudio,
- análisis en OMT con las clases de la aplicación,
- prototipo horizontal de la aplicación, con el diseño de la interfaz gráfica de usuario,
- implementación de la aplicación junto con el manual de usuario.

En la evaluación se tienen en cuenta tanto los criterios del profesor como los criterios de los alumnos. De esta forma la evaluación de la

aplicación desarrollada (y en consecuencia de las prácticas de la asignatura) se hace de acuerdo a la siguiente fórmula:

$$\text{Nota evaluación} = (A*0.40 + B*0.30 + C*0.30)$$

Puesto que la aplicación a desarrollar pretende ser una aplicación de gestión con objetivos similares a los de una aplicación real, el apartado A consiste en una evaluación anónima (mediante un cuestionario) de la aplicación por otro grupo de alumnos en el rol de usuarios y *probadores* de la aplicación. Se trata, en este caso, que los propios alumnos sean capaces de evaluar y valorar el trabajo que han realizado el resto de alumnos, aprendiendo de los posibles errores cometidos y/o de las buenas soluciones propuestas por sus propios compañeros. El apartado B consiste en la evaluación por parte del profesor de la evaluación que cada grupo realiza de otra aplicación, es decir, el profesor evalúa a los alumnos en el rol de usuario y probadores. En este apartado se valora negativamente una mala evaluación de una aplicación, tanto por exceso como por defecto. Es decir, a un grupo se le valorará negativamente si califica incorrecta e injustamente una aplicación sin ningún motivo justificado (se exige que todas las valoraciones realizadas estén suficientemente razonadas). Por último, el apartado C consiste en la evaluación por parte del profesor del trabajo personal realizado basándose en la documentación entregada a lo largo del curso. Tal y como puede observarse en la fórmula anterior, la calificación de mayor peso recae sobre el apartado A. Puesto que la aplicación pretende ser un producto por y para un cliente, parece justo que sea el cliente final quien la califique (y a falta de clientes, los propios alumnos realizan dicho papel). Adicionalmente, esta fórmula presenta un factor de corrección adicional definido por el profesor que permite modificar la calificación final de acuerdo al conocimiento que el profesor posee del trabajo realizado por los alumnos. Este factor de corrección se introduce para evitar que una mala corrección de la práctica por parte de los alumnos afecte negativamente a la nota de la práctica.

Esta forma de realización y evaluación de prácticas se ha venido realizando desde el curso 1998-99. Esta experiencia nos ha servido para plantearnos si el enfoque utilizado es adecuado o no, así como posibles mejoras a introducir en el

mismo. A este respecto destacar que, el alto grado de satisfacción de los alumnos con el proyecto software realizado, una vez finalizado el mismo, refuerza nuestra opinión de que el enfoque dado a las prácticas es correcto y provechoso para el alumno, pues lo acerca bastante a la realidad con la que se va a enfrentar en el mercado laboral.

Por otra parte, los resultados de las diferentes evaluaciones realizadas a los alumnos a lo largo de los dos cursos académicos anteriores (98-99 y 99-00) nos ha posibilitado llevar a cabo un sencillo estudio estadístico para valorar si los coeficientes aplicados a cada una de las partes evaluadas en la practica (A, B y C) son correctos, o por el contrario es conveniente realizar alguna corrección sobre los mismos.

Un resumen del cuestionario con los aspectos y las preguntas que los alumnos deben contemplar en la evaluación de la práctica se presenta en la Tabla **¡Error!Argumento de modificador desconocido.** El valor, en porcentaje sobre la nota final, de cada pregunta aparece en la segunda columna.

Preguntas	Porcentaje
1. Revisión de la especificación de requisitos IEEE-830 (ambigüedad, precisión, correctitud, consistencia, verificabilidad...)	10%
2. Evaluación del modelo de objetos OMT (clases, relaciones...)	5%
3. Diseño de la interfaz gráfica de usuario (existen modificaciones respecto al prototipo entregado, es intuitiva de manejar, posee valores añadidos...)	20%
4. Implementación de la aplicación (facilidad de instalación y uso, evaluación del funcionamiento y del proceso, robustez...)	35%
5. Consistencia de la aplicación con los requisitos iniciales indicados en la IEEE-830 entregada	10%
6. Valoración de las ampliaciones	10%

realizadas (indicando grado de dificultad y adecuación)	
7. Evaluación del manual de usuario	5%
8. Impresión general de la aplicación	5%

Tabla **¡Error!Argumento de modificador desconocido.** Resumen con las preguntas a responder

5. Análisis de Resultados

El estudio de la aplicación de este método de evaluación de las prácticas se ha realizado sobre una población aproximada de unos 150 alumnos repartidos en dos cursos académicos.

En la Tabla 2 se puede observar la calificación media obtenida en el apartado B (donde se mide la calidad de la evaluación realizada por los alumnos), así como la desviación típica. En estos datos se observa como los alumnos resultan más capaces de realizar una mejor valoración de los apartados 3, 4 y 8 (y además con la menor dispersión de los datos) Estos aspectos se corresponden con la interfaz de usuario, la implementación y la impresión general. Los alumnos evalúan mejor la interfaz de usuario y la implementación. Esto no significa que sean las disciplinas que más dominan, sino que son las disciplinas a las cuales han dedicado un mayor esfuerzo, por lo que tienen un perfecto conocimiento de todos los problemas existentes durante el desarrollo y pueden evaluarlas con mayor precisión. El apartado en el que los alumnos se encuentran con mayores problemas a la hora de evaluar es el de la especificación de requisitos. Esto resulta lógico puesto que la especificación de requisitos se les presenta en el último curso de la titulación, por lo que no se sienten suficientemente capacitados para evaluarla correctamente. El modelado OMT y el manual de usuario siguen siendo todavía elementos de evaluación muy subjetiva, por lo que existe una elevada dispersión en los datos.

Preguntas apartado B	1	2	3	4	5	6	7	8
Calificación media (y desviación típica)	5,72 (2,79)	5,86 (2,56)	6,66 (1,65)	6,77 (1,78)	6,47 (2,03)	5,78 (2,08)	5,85 (2,44)	6,89 (2,05)

Tabla **¡Error!Argumento de modificador desconocido.** Calificación media para las preguntas

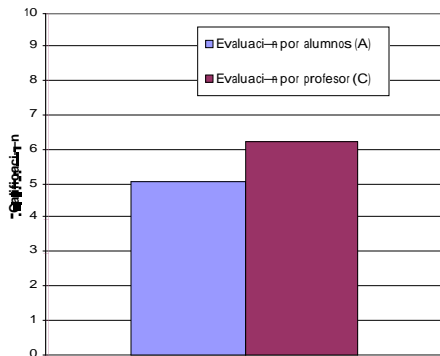


Figura **¡Error!Argumento de modificador desconocido.** Comparación entre la evaluación por los alumnos y por el profesor

Uno de los principales problemas con los que nos hemos encontrado en el modelo de evaluación propuesto ha sido la elevada exigencia con la que los alumnos evalúan las aplicaciones desarrolladas por sus propios compañeros. Para ello, la Figura **¡Error!Argumento de modificador desconocido.** compara de forma global, las evaluaciones que de cada práctica realiza, por una parte el profesor, y por otra parte, otro grupo de alumnos. Nos ha llamado la atención que la calificación media de los alumnos en el rol de clientes es prácticamente 1.25 puntos inferior a la calificación de los profesores (cuando cabía esperar todo lo contrario). La sensación que obtuvimos durante el proceso de evaluación fue que los alumnos exigían a las prácticas de los compañeros características que no habían llegado a desarrollar correctamente en su propia práctica. Esto pone de manifiesto la existencia de un alto nivel de competitividad entre ellos y sobre todo destacar que son capaces de exigir a los demás compañeros más de lo que ellos mismos pueden ofrecer, siendo incapaces de aplicar sus criterios de calidad a sus propias prácticas. Con esto se resalta la importancia que tiene sobre la calidad de las prácticas los procesos de revisión llevados a cabo por parte de otros desarrolladores. Esta competitividad que podría incluso resultar favorable para incrementar el interés y motivación por las prácticas de la asignatura (y desarrollar una buena aplicación) ha producido, sin embargo, algunos casos en los que la valoración no resultaba justa. En algunos grupos la calificación por parte de los alumnos no se correspondía en

absoluto con la calificación por parte del profesor. Esto nos ha obligado a modificar en el curso académico actual el peso de cada uno de los apartados A, B y C, dando un mayor peso al criterio del profesor. De esta forma, la calificación queda, finalmente de la siguiente forma:

$$\text{Nota evaluación} = (A*0.25 + B*0.25 + C*0.50)$$

6. Conclusiones

En este artículo se ha presentado un modelo de evaluación de prácticas para la asignatura de Laboratorio de Ingeniería del Software. Como conclusiones se puede indicar que el modelo propuesto se adapta a las necesidades de evaluación de asignaturas con un importante carácter práctico; ahora bien, siempre y cuando el mayor peso de la calificación final recaiga en el criterio del profesor para poder corregir las posibles desviaciones. Además, es fundamental remarcar a los alumnos el interés que deben mostrar en la evaluación con el fin de poder aprender de los trabajos que realizan el resto de sus compañeros.

Como trabajo futuro se pretende obtener una mayor realimentación por parte de los alumnos, para valorar su opinión tras cursar la asignatura, en cuestiones tales como: proporción del trabajo realizado respecto a la carga lectiva, conveniencia del trabajo en grupo, adecuación de las preguntas del cuestionario y satisfacción general del alumno.

Referencias

- ¡Error!Argumento de modificador desconocido.**] J.A. Calvo-Manzano, J. Carrillo, G. Cuevas, T. San-Felú y E. Tovar. *Ingeniería Software I*. Actas JENUI-2000, 2000.
- ¡Error!Argumento de modificador desconocido.**] J.H. Canós, M.C. Penadés, V. Pelechano y J. Sánchez. *La Ingeniería del Software en los planes de estudio de la Escuela Universitaria de Informática de la UPV*. Actas de las II Jornadas de Ingeniería del Software (JIS'97), 1997.
- ¡Error!Argumento de modificador desconocido.**] J.H. Canós, M.C. Penadés, V.

- Pelechano, J. Sánchez, I. Ramos y A. González. *La Ingeniería del Software en los planes de estudio: ¿dos perspectivas de una disciplina o más de lo mismo?* Actas JENUI-98, 1998.
- [**Error!Argumento de modificador desconocido.**] D.E. Douglas y B.C. Hardgrave. *Object-Oriented curricula in academic programs*. Communications of the ACM, 2000.
- [**Error!Argumento de modificador desconocido.**] I. Marteens. *La cara oculta de Delphi 4*. Danysoft Internacional, 1998.
- [**Error!Argumento de modificador desconocido.**] V. Pelechano y M.C. Penadés. *Laboratorio de Ingeniería del Software. La ingeniería del software orientada a objetos y cliente/servidor llevada a la práctica*. Actas JENUI-98, 1998.
- [**Error!Argumento de modificador desconocido.**] V. Pelechano, M.C. Penadés, A. Garrido y J.A. Carsí. *Laboratorio de Ingeniería del Software. Ejercicios prácticos*. Servicio de publicaciones de la UPV, 2000.
- [**Error!Argumento de modificador desconocido.**] R.S. Pressman. *Ingeniería del Software: un enfoque práctico*. McGraw-Hill, 1994.
- [**Error!Argumento de modificador desconocido.**] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy y W. Lorensen. *Modelado y diseño orientado a objetos*. Prentice-Hall, 1994.