

Experiencia de aprendizaje de programación basada en proyectos

Laura Igual

Departament de Matemàtica Aplicada i Anàlisi
Universitat de Barcelona
08007 Barcelona
ligual@ub.edu

Xavier Baró

Estudis d'Informàtica, Multimèdia i Telecomunicació
Universitat Oberta de Catalunya
08018 Barcelona
xbaro@uoc.edu

Resumen

En este artículo se presenta una experiencia desarrollada los últimos cuatro años dentro del contexto de una asignatura de programación del primer curso del Grado en Ingeniería Informática. El objetivo principal de esta experiencia es que los estudiantes trabajen en un proyecto realista de programación que mejora su motivación e implicación en el proceso de aprendizaje. Los profesores facilitan una librería de código que permite a los estudiantes explotar sus conocimientos en los aspectos importantes de los paradigmas de programación para obtener resultados más realistas desde el primer día. En particular, se consigue visualizar una imagen o escuchar una canción en lugar de mostrar un simple mensaje por pantalla. Los enunciados de los proyectos están planteados para que aparezcan de forma natural las necesidades de aplicar ciertos conceptos y estrategias de programación. Esto permite introducir de forma práctica los conceptos básicos trabajados en las clases de teoría, como son los mecanismos de programación orientada a objetos, las nociones básicas de control de errores basado en excepciones y la gestión de eventos. De esta forma los estudiantes ven la utilidad de cada concepto asimilándolo mejor. Para facilitar el desarrollo de las prácticas, el proyecto se divide en diferentes fases incrementales con un enunciado asociado donde se facilita toda la información sobre los métodos a implementar así como la ayuda adicional para alcanzar los objetivos. A partir de esta experiencia se observa una gran implicación de los estudiantes en las prácticas y una alta correlación entre las calificaciones de los proyectos y de los exámenes teóricos. En el artículo se detallan los proyectos de programación implementados, su organización en etapas y división en entregas, así como los métodos de seguimiento y evaluación. Por último, se presentan los resultados académicos y de satisfacción del alumnado y se discuten los problemas encontrados y las posibles mejoras.

Abstract

This article presents an experience developed over the last four years within the context of a programming course of the first year of the Degree in Computer Engineering. The main objective of this experience is that students work in a realistic programming that enhances their motivation and involvement in the learning process. Teachers provide a library of code that allows students to exploit their knowledge of the important aspects of programming paradigms to obtain more realistic results from day one. In particular, they can visualize an image or hear a song instead of displaying a simple message on the screen. The statements of the projects are posed to make appear naturally the needs of applying certain concepts and programming strategies. This allows a practical way to introduce basic concepts presented in the lectures, as are the mechanisms of object-oriented programming, the basics of exception-based error handling and event management. In this way, students see the usefulness of each concept assimilating it better. To facilitate the development of the practices, the project is divided into incremental phases associated with a statement which give all the information on the methods to implement and further assistance to achieve the objectives. From this experience it is noted a great student involvement in practices and a high correlation between the qualifications of the project and the theoretical examinations. The article details the programming projects implemented, the organization in staged and division in deliveries, as well as the monitoring and evaluation methods. Finally, we present the academic results and student satisfaction results and discuss the problems encountered and the possible improvements.

Palabras clave

Experiencia de aprendizaje de programación, aprendizaje basada en proyectos, programación orientada a

objetos y a eventos.

1. Introducción

En el curso 2009-2010 se inició en la Universitat de Barcelona la implantación de los nuevos estudios del Grado de Ingeniería Informática dentro del marco del Espacio Europeo de Educación Superior (EEES). Se crearon nuevas asignaturas que antes no existían como tales, y que se tenían que planificar de nuevo siguiendo nuevos planes de estudios que abarcaran los retos planteados en el EEES. Este contexto de cambio se planteó como un momento idóneo para introducir una nueva metodología pedagógica reconocida como innovadora y exitosa, pero con poca presencia en los antiguos estudios de informática de la UB: “El Aprendizaje Basado en Proyectos” (ABP) [1], [2].

Hasta entonces, en los estudios de Ingeniería Técnica en Informática de Sistemas, la mayoría de las asignaturas de programación se estructuraban en clases magistrales de teoría y clases de prácticas donde se resolvían una serie de problemas pequeños o medios para trabajar los conceptos presentados. Dentro de los nuevos planes de estudios, los conceptos de la Programación Orientada a Objetos y Eventos se reestructuraron dentro de la asignatura de “Programación II” del segundo cuatrimestre del primer curso del Grado de Ingeniería Informática.

El valor de la estrategia ABP aplicada a la enseñanza de Programación II se centra en reproducir los pasos que se siguen en procesos de desarrollo de código, aquellos que se realizan cuando se plantean problemas reales. En consecuencia, el alumnado se enfrenta, aprende y desarrolla, un conjunto de habilidades que pueden ser de especial importancia en la formación de los futuros programadores.

En particular, nuestra propuesta de aplicar la metodología ABP en esta asignatura tenía los siguientes objetivos:

1. Mejorar la estrategia para una evaluación continuada del alumnado.
2. Aprender a utilizar los conocimientos adquiridos para resolver un problema de la vida real.
3. Aumentar la implicación y motivación del alumnado.
4. Aprender a trabajar con módulos de programación externos proporcionados por el profesorado.
5. Familiarizarse con el uso de herramientas informáticas de apoyo a la programación.
6. Obtener un producto final de programación perdurable y ampliable a lo largo de la carrera.

La metodología general que propusimos, se basa en plantear el desarrollo de un proyecto de programación pasando por diferentes etapas de la implementación,

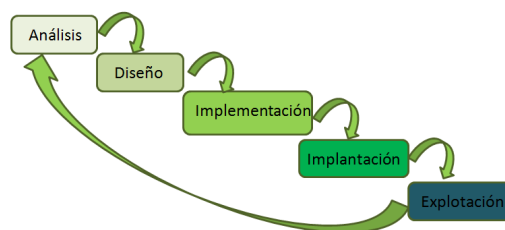


Figura 1: Ciclo de Vida del Software.

las cuales se tienen que alcanzar paulatinamente para llegar al objetivo final.

El resto del artículo se estructura de la siguiente manera: En el apartado 2 se contextualiza la experiencia realizada. En el apartado 3, se describe la metodología utilizada y se detalla el portafolio utilizado. En el apartado 4, se presentan los resultados obtenidos y las dificultades encontradas. Por último, en el apartado 5, se concluye el trabajo y se plantean posibles mejoras.

2. Contexto y experiencia

La asignatura de Programación II forma parte de la materia de programación integrada por un total de 5 asignaturas impartidas desde primero a cuarto curso del Grado de Ingeniería Informática de la UB. En esta materia se cubre todo el ciclo de vida del software o proceso de desarrollo de software en el que las necesidades del usuario son traducidas en requisitos de software, éstos transformados en diseño y el diseño implementado en código, seguido de la implantación y explotación de este código. Este último paso da pie a un nuevo análisis que cierra de nuevo el ciclo. En la Figura 1 se ilustra este ciclo.

La asignatura de Programación II se centra en el paso de implementación. En particular, se concentra en introducir nuevos conceptos de módulo y de abstracción de datos, técnicas de descomposición modular de programas, paradigmas de programación orientada a objetos y orientada a eventos [3]. Se explican los principios de los paradigmas de programación y su justificación, así como las ventajas y los inconvenientes de adoptar estos paradigmas en un proyecto de programación.

La asignatura de 6 créditos ECTS, contiene 60 horas de actividades presenciales organizados en clases teóricas y clases prácticas. Las primeras son clases magistrales para la introducción de conceptos teóricos y las segundas son clases realizadas en laboratorios de ordenadores, en los cuales se ponen en práctica los conocimientos adquiridos en las clases de teoría mediante una metodología docente basada en proyectos.

En esta asignatura, además de los objetivos específicos de aprendizaje, se desarrollan las siguientes competencias:

- Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad.
- Tener capacidad de abstracción: crear y utilizar modelos que reflejen situaciones reales.
- Tener capacidad para el razonamiento crítico y lógico.
- Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.

3. Metodología y Portafolio

El ABP es una estrategia de enseñanza que invierte la organización tradicional de los procesos de aprendizaje. En los procesos tradicionales, en primer lugar y de forma muy diversa, se exponen los objetivos; a continuación, a través de un conjunto heterogéneo de actividades, se presenta la información pertinente y se trabaja el marco conceptual. Posteriormente, pero no siempre, se promueve la aplicación del conocimiento adquirido en el marco de la resolución de un problema. En cambio, en el Aprendizaje Basado en Problemas, primero se presenta el problema, y a partir de la delimitación de aquello que ya se conoce y de aquello que no se sabe, se establecen los objetivos, se identifican las necesidades de aprendizaje que son básicas, se diseña y se sigue un plan de actuación, para que a lo largo del proceso en que se produce el aprendizaje, se resuelva simultáneamente el problema.

Durante cuatro cursos seguidos (2009-2010, 2010-2011, 2011-2012, 2012-2013) se ha utilizado la estrategia ABP en las prácticas de la asignatura de Programación II. Se ha planteado un proyecto (problema real) a los estudiantes que se ha dividido en diferentes módulos, los cuales se tienen que ir desarrollando para llegar al objetivo final. Este objetivo final es la realización de un producto útil para el alumno que, además representa una aplicación perdurable y ampliable por él. De este modo se motiva al alumno con un problema atractivo desde el punto de vista de programación y de aplicación. Hasta ahora se han planteado dos proyectos de prácticas diferentes: un reproductor de música (2009-2010 y 2010-2011) y un visor de imágenes (2011-2012 y 2012-2013). Para la realización del proyecto de prácticas se ha utilizado el lenguaje de programación Java y el entorno de desarrollo integrado, NetBeans. Además, para asegurar un buen funcionamiento del ABP, se ha llevado a cabo un control exhaustivo y sistemático de las entregas periódicas, con un acotamiento y

seguimiento de las mismas. Esta gestión se ha realizado mediante la plataforma Moodle.

A continuación se presentan los detalles del proyecto de desarrollo de un visor de imágenes, que se ha basado en una biblioteca de imágenes formada por distintos ficheros y un conjunto de álbumes para organizar estas imágenes.

3.1. Definición de las entregas

El proyecto se presenta en un primer enunciado general. Cada uno de las entregas del proyecto viene guiado por un enunciado con las especificaciones necesarias.

1. Objetivos
2. Descripción de la entrega
3. Material para la entrega
4. Ayuda para la entrega
5. Forma de la entrega y contenido
6. Código con una estructura de paquetes determinada
7. Memoria con una serie de puntos marcados.

Estos enunciados se van publicando en el Campus Virtual y las entregas se hacen también a través del Campus Virtual los días antes de las sesiones de corrección. Cada enunciado viene acompañado por el material proporcionado para la entrega. Como parte del material, se proporciona a los estudiantes una librería con las funcionalidades necesarias para visualizar las imágenes y generar los eventos temporales. Tal como se indica en los enunciados, cada una de las entregas consiste en un código y una memoria que hay que preparar. El código corresponde a una versión cada vez más completa de un mismo visor de imágenes. Por esto, se nombra el proyecto igual que la entrega: Visor1, Visor2, Visor3 y Visor4.

Para ayudar a distribuir el trabajo, se divide el proyecto en cuatro entregas graduales, de forma que se debe finalizar una entrega para poder realizar la siguiente. El proyecto se diseña siguiendo el modelo MVC (Modelo-Vista-Controlador), y en cada entrega se avanza en las tres vertientes:

Entrega 1: Diseñar e implementar una aplicación con un menú de texto que nos permita gestionar ficheros dentro de una lista.

Entrega 2: Diseñar e implementar una aplicación con un menú de texto que nos permita organizar las imágenes dentro de la biblioteca de imágenes y los álbumes, así como visualizarlas.

Entrega 3: Implementar la visualización de una de las listas de imágenes, con las opciones de iniciar, parar y pausar la visualización, mediante el control de eventos.

Entrega 4: Implementar una interfaz gráfica que ten-

ga las mismas opciones que en el caso de la Entrega 3. Se trata de evidenciar las ventajas de un buen diseño, cambiando la vista de la aplicación sin necesidad de alterar el controlador y el modelo.

En la entrega 1 se repasan conceptos vistos en la asignatura previa de programación. Esta fase permite al estudiante situarse en el proyecto y comenzar una primera versión básica del visor. Aquí se comienza a incluir la utilización de estructuras de datos implementadas en la librería de Java. En la entrega 2 se implementan nuevas clases y aparece de forma natural la primera herencia con la que los alumnos trabajan (fichero y fichero de imagen). Además, se amplía la utilización de la librería de útiles para la funcionalidad de visualización de las imágenes. En la entrega 3 se amplía el número de funcionalidades posibles del visor. Esto complica la implementación que debe estar bien organizada de forma modular, tal como se indica durante el curso, para que sea fácilmente ampliable. Por último, en la entrega 4, no se amplía ninguna funcionalidad del visor, pero se cambia totalmente el aspecto del mismo, pasando de un menú de texto a una interfaz gráfica.

Las entregas son individuales debido a que así se ve necesario y se establece en todas las asignaturas de primero del grado. Con la finalidad de evaluar la progresión de los estudiantes, además de las entregas se realizan pruebas de validación. Estas pruebas permiten simplemente asegurar que los estudiantes han realizado realmente el trabajo y que éste no ha sido copiado. Esto constituye un peligro común, ya que todos los alumnos deben realizar el mismo proyecto.

Con la finalidad de evitar subjetividades en las correcciones, se definen unos criterios de evaluación específicos para cada entrega de prácticas y se corrigen todas las entregas siguiendo éstos criterios. Además es muy importante proporcionar un continuo retorno a los estudiantes. Para ello, se establece un plazo máximo de una semana para la corrección de cada entrega y se realizan comentarios sobre cada práctica mediante el Campus Virtual y personalmente, durante las sesiones de prácticas, si es necesario.

En el documento de la normativa de prácticas, que se explica el primer día de clase, se presentan los métodos y criterios de evaluación y se definen también unos requisitos mínimos para que la practica sea considerada "Apta" y por tanto pase a ser evaluada y comentada. Estos requisitos son los siguientes:

- La entrega se realiza dentro del plazo indicado y en el formato especificado.
- El código entregado compila correctamente.
- El programa entregado obtiene todos los resultados que se piden en el enunciado.
- La documentación contiene todos los puntos pedidos en el enunciado.

De esta manera se reduce el trabajo del profesor que no tiene que corregir prácticas no acabadas.

Uno de los peligros de la metodología utilizada es el abandono de algún estudiante que no cumple con las fechas y acaba quedándose atrás. Por este motivo, desde el primer día, se dan unas pautas bien claras a los alumnos de cómo se tiene que trabajar dentro de la metodología utilizada. Se explica que es necesario ir al día, presentar cada entrega y seguir los comentarios y correcciones de los profesores para poder finalizar bien el proyecto. Se ha observado que la gran mayoría de los alumnos que realizan todas las entregas aprueban las prácticas. Para evitar que un estudiante que no pueda llegar a una entrega quede fuera del proceso de aprendizaje, los estudiantes disponen de una segunda oportunidad de entregar sus proyectos con una penalización.

3.2. Método de evaluación

La evaluación de la asignatura se hace a partir de la realización de dos exámenes parciales de contenido teórico, para calcular la nota de teoría, y las entregas del proyecto de prácticas y la realización de pruebas de validación individuales durante el curso, para calcular la nota de prácticas. Así, la nota final se calcula ponderando 50 % de la nota de prácticas y 50 % de la nota de teoría. De esta manera se le da la importancia merecida al esfuerzo que supone el proyecto realizado en las prácticas. Además, se ha observado que la gran mayoría de los estudiantes que aprueban las prácticas aprueban la teoría.

Para poder hacer el cálculo de la nota final se impone como condición indispensable que el estudiante obtenga una nota de prácticas y de teoría mayor o igual que 4.

Para aprobar las prácticas se impone como requisito indispensable tener un mínimo de 3 de las 4 entregas como "Aptas", tener notas de las dos pruebas de validación mayor o igual que 4 y tener una nota final mayor o igual que 4. Para recuperar una entrega que haya sido considerado "No Apta" o con nota inferior a 5, hay que volver a entregar todo el material pedido y una entrega recuperada puede obtener una calificación máxima de 7.

4. Resultados

Los resultados de esta experiencia son satisfactorios desde el punto de vista académico y de satisfacción de los estudiantes.

En el Cuadro 1, se presentan los resultados académicos de los cursos 2009-2010, 2010-2011 y 2011-2012. Se incluye el porcentaje de estudiantes que obtuvieron

	MH	S	N	A	SP	NP	Total
2011-2012	2.2 %	1.1 %	23.1 %	22.0 %	18.7 %	33.0 %	93
2010-2011	2.1 %	3.1 %	16.7 %	25.0 %	30.2 %	22.9 %	96
2009-2010	0.0 %	1.1 %	16.8 %	29.5 %	14.7 %	37.9 %	95

Cuadro 1: Resultados académicos: Porcentaje de Matriculas de Honor (MH), Sobresalientes (S), Notables (N), Aprobados (A), Suspensos (SP), No presentados (NP) y el total de estudiantes matriculados.

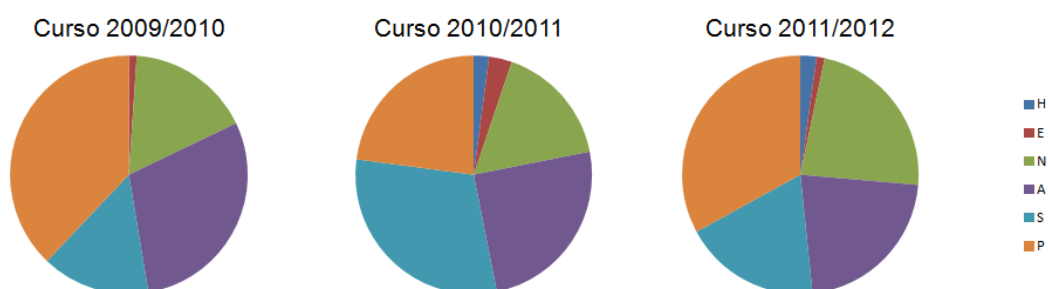
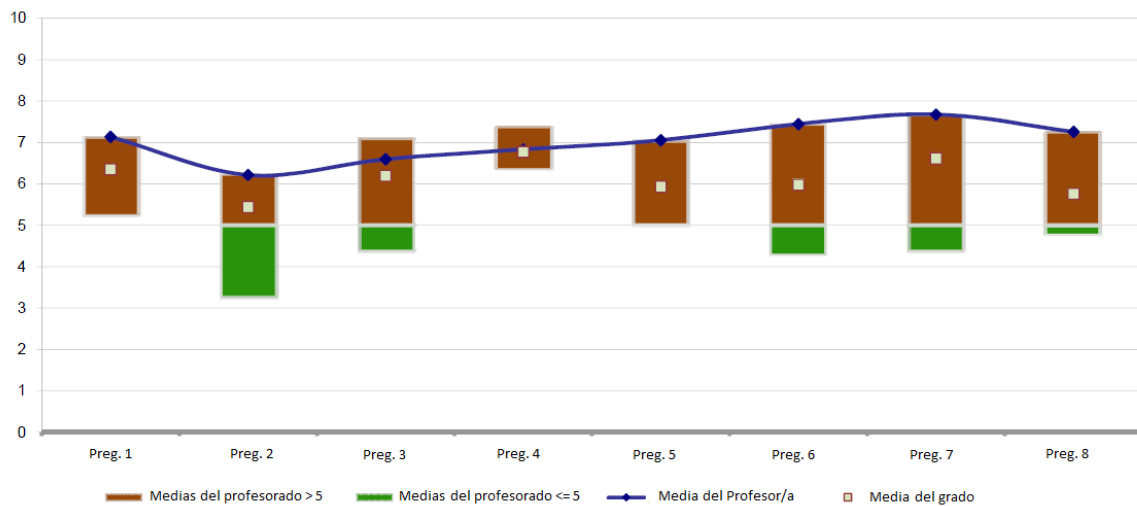


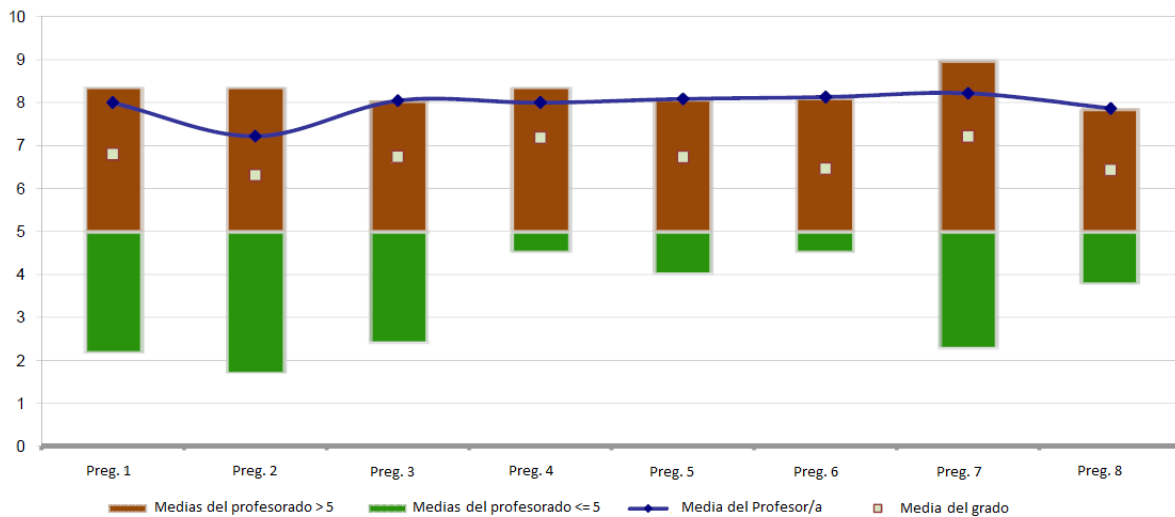
Figura 2: Gráfico de los resultados académicos de la asignatura.

Puntuación pregunta por curso.	Media (Des.) 2009-2010	Media (Des.) 2010-2011	Media (Des.) 2011-2012
1. En general, estoy satisfecho/a con la actividad docente llevada a cabo por el profesor/a de la asignatura	7,14 (2,06)	8,00 (1,95)	8,00 (2,74)
2. La manera de desarrollar la actividad docente consigue motivar el alumnado	6,22 (2,92)	7,22 (2,52)	8,40 (2,30)
3. Transmite con claridad los contenidos de la asignatura	6,59 (2,40)	8,05 (1,89)	8,60 (2,19)
4. La actividad docente ha permitido cumplir el programa de la asignatura	6,83 (2,16)	8,00 (2,13)	8,60 (2,19)
5. Las actividades de evaluación propuestas por el profesor/a han sido adecuadas a la asignatura	7,05 (2,48)	8,09 (1,76)	7,40 (2,30)
6. El material de estudio y de consulta propuesto (bibliografía, documentos, recursos didácticos, etc.) ha sido útil para el aprendizaje de la asignatura	7,44 (2,29)	8,13 (2,16)	8,80 (2,17)
7. Mantiene un buen clima de comunicación y relación con el alumnado	7,68 (2,33)	8,22 (2,19)	8,80 (2,17)
8. (En caso de que hayáis hecho) Las actividades propuestas por el profesor/a en el Campus Virtual han sido útiles en el proceso de aprendizaje de la asignatura	7,26 (2,37)	7,86 (2,29)	8,00 (2,65)
Nota media de todas las preguntas (porcentaje de respuesta)	7,03 (39,58 %)	7,95 (23,71 %)	8,33 (5,43 %)

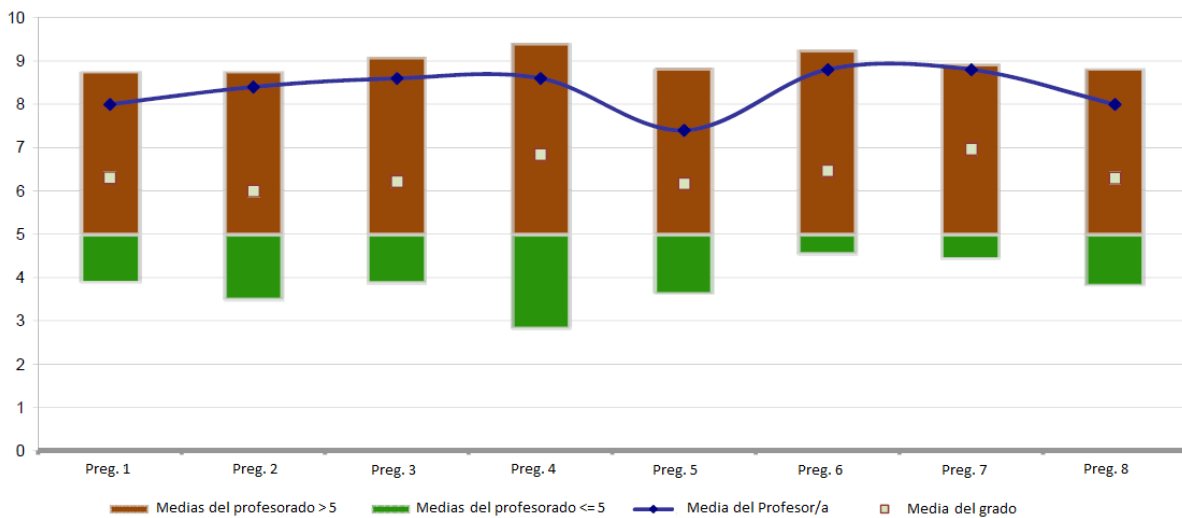
Cuadro 2: Detalles de las respuestas: Puntuación media (Desviación) de cada pregunta, puntuación media de todas las preguntas (porcentaje de respuesta).



(a) Curso 2009-2010



(b) Curso 2010-2011



(c) Curso 2011-2012

Figura 3: Gráfico de medias: situación del profesorado en el conjunto del Grado para las 8 preguntas de la encuesta.

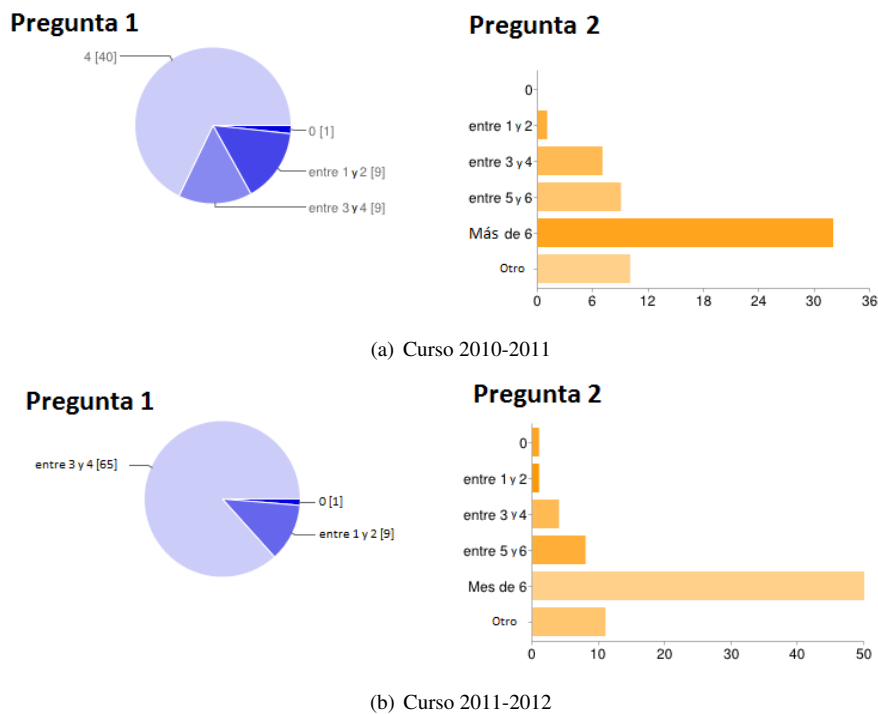


Figura 4: Resultados de las encuestas de dedicación. Preguntas: 1) ¿Cuántas horas semanales has asistido en promedio a las actividades presenciales de la asignatura (las últimas dos semanas)? y 2) ¿Cuántas horas semanales en promedio has dedicado al trabajo autónomo (fuera de las aulas) de la asignatura (las últimas dos semanas)?

Matriculas de Honor (MH), Sobresalientes (S), Notables (N), Aprobados (A), Suspensos (SP), No presentados (NP) y el total de alumnos matriculados. En la Figura 2 se puede ver un gráfico de estos mismos resultados.

Como se puede observar el número de estudiantes no presentados es elevado, pero no dista de los porcentajes de otras asignaturas de primer curso del grado. Este fenómeno puede ser debido a múltiples factores: 1) alumnos sin experiencia se matriculan de demasiadas asignaturas para ellos y tienen demasiada carga docente, 2) la asignatura de Programación II es la continuación de Programación I que tiene un número similar de no presentados, 3) cuando ciertos alumnos no adquieren los conocimientos de Programación I puede ser difícil seguir la asignatura. Además, somos conscientes que la metodología docente utilizada puede añadir al menos una dificultad. Esta consiste en que las entregas periódicas no son independientes entre si y, por ello, es necesario finalizar correctamente cada una de las entregas para realizar las siguientes.

Por último, cabe destacar que se ha observado que el 82,14 % de los estudiantes que realizan las prácticas aprueban la asignatura.

Además de los resultados académicos, es interesante analizar los resultados de satisfacción del alumnado.

Para ello, se utilizan las encuestas de opinión anónimas realizadas por parte de la Universidad cada curso. En particular, cada curso se ha realizado una encuesta de opinión que contiene ocho preguntas sobre la asignatura. Las preguntas se puntúan en una escala de valoración de 0 a 10 para indicar de "totalmente en desacuerdo" a "totalmente de acuerdo". Los resultados de los cursos 2009-2010, 2010-2011 y 2011-2012 se incluyen en el Cuadro 2. La primera columna especifica la pregunta realizada y las tres columnas restantes contienen la puntuación media y desviación de cada pregunta para los tres cursos. La última fila indica la puntuación media de cada curso, así como el porcentaje de respuesta realizadas ese curso.

En la Figura 3 se pueden ver los gráficos de las medias, para comparar la situación de la asignatura/profesor en el conjunto de asignaturas/profesorado del grado durante los tres cursos 2009-2010, 2010-2011 y 2011-2012. Se muestran las medias del profesorado con puntuaciones >5 y ≤ 5 (frangas marrones y verdes, respectivamente), la media del profesor de la asignatura (línea azul) y la media del grado (cuadrado beis).

A partir de estas encuestas y de conversaciones con los profesores y miembros del equipo de estudios, se puede afirmar que el proyecto de prácticas es atractivo

para los estudiantes y que los contenidos y el modo de trabajarlos resulta interesante para ellos.

Se realizaron también encuestas de dedicación, para saber la carga docente que suponía esta asignatura y si estaba dentro de las estimaciones que se realizan en el grado. Se formularon dos preguntas: 1) ¿Cuántas horas semanales has asistido, en promedio, a las actividades presenciales de la asignatura (las últimas dos semanas)? y 2) ¿Cuántas horas semanales, en promedio, has dedicado al trabajo autónomo (fuera de las aulas) de la asignatura (las últimas dos semanas)? Los resultados se pueden ver en la Figura 4. Estos resultados de las encuestas muestran que la carga de trabajo autónomo para la realización del proyecto de prácticas puede ser excesiva. Sin embargo, creemos que la valoración positiva de las encuestas de opinión compensa este sobreesfuerzo realizado.

5. Conclusiones y trabajo futuro

En este artículo se ha presentado la experiencia de aprendizaje de programación basada en proyectos para la asignatura de Programación II del grado de Ingeniería Informática. En esta experiencia, se profundiza en el desarrollo de las competencias de capacidad para resolver un proyecto con iniciativa, toma de decisiones, autonomía y creatividad. Las tareas a desarrollar dentro del proyecto de programación se delimitan y se proporciona material suficiente para realizarlo en los periodos establecidos. Mediante esta metodología se han conseguido unos resultados académicos satisfactorios y una valoración positiva por parte de los estudiantes. A pesar de esto, se ha observado que la carga docente es mayor que para otras asignaturas. Este es un tema que se debe estudiar y tratar de adaptar en el futuro.

En los cursos siguientes se adoptarán medidas para captar y evitar el abandono de la asignatura, especialmente para captar a aquellos estudiantes que debido a haber tenido problemas en la asignatura predecesora, no llegan a asistir a las sesiones prácticas. Además, se hará énfasis en el seguimiento y soporte de los estudiantes que presentan las entregas.

6. Agradecimientos

Este trabajo se inició con el proyecto de innovación docente *Análisis y implantación de la metodología de aprendizaje basado en problemas en la asignatura de programación II del grado de ingeniería informática* concedido y financiado por la Universitat de Barcelona en 2009.

Referencias

- [1] Pablo Sánchez and Carlos Blanco. Implantación de una metodología de aprendizaje basada en proyectos para una asignatura de Ingeniería del Software. In *Actas de las XVIII Jornadas de Enseñanza Universitaria de Informàtica (JENUI)*, páginas 41 – 48, Ciudad Real (España), Julio 2012.
- [2] Miguel Valero-García and Javier García Zubia. Cómo empezar fácil con PBL. Julio 2011. In *Actas de las XVII Jornadas de Enseñanza Universitaria de Informàtica (JENUI)*, páginas 109 – 116, Sevilla (España), Julio 2011.
- [3] Ian Sommerville. *Software Engineering*. Addison Wesley, March 2010.