

APRENDER A PROGRAMAR ORDENADORES MEDIANTE UNA METODOLOGÍA BASADA EN PROYECTOS

DEL CANTO RODRIGO, Pablo; GALLEGO FERNÁNDEZ, Isabel; HIDALGO CARRILLO, Ruben; LÓPEZ CHAPARRO, Johann LÓPEZ CANDALDA, José Manuel; MORA SERRANO, Javier; RODRÍGUEZ LUNA, Eva SANTAMARIA BARNADAS, Eduard; VALERO GARCÍA, Miguel

eduard.santamaria@upc.edu

Universitat Politècnica de Catalunya, España, Escola Politècnica Superior de Castelldefels, Departamento de Arquitectura de Computadores

RESUMEN

En esta ponencia se describe la aplicación de la metodología de aprendizaje basado en proyectos en dos asignaturas destinadas al aprendizaje de la programación de ordenadores en las titulaciones de Ingeniería Técnica de Telecomunicación e Ingeniería Técnica Aeronáutica que se imparten en la Escuela Politécnica Superior de Castelldefels (EPSC) de la Universidad Politécnica de Catalunya.

El aprendizaje basado en proyectos promueve el aprendizaje activo, facilita el desarrollo de competencias transversales y motiva a los estudiantes. Esta metodología presenta como principales retos: (1) la definición de un buen proyecto que permita alcanzar los objetivos de aprendizaje, (2) la planificación del curso identificando tareas clave y plazos de entrega que aseguren un progreso adecuado en el desarrollo de los proyectos y (3) la definición de un método de evaluación que permita establecer cuál ha sido el resultado en la consecución de los objetivos. En esta ponencia se presentan los objetivos, proyectos, organización y metodología de evaluación de las asignaturas de Introducción a los Computadores y Laboratorio de Programación de la EPSC como ejemplos de aplicación del aprendizaje basado en proyectos. Creemos que la información aportada puede resultar de utilidad para aquellos profesores interesados en la implantación de esta metodología.

Palabras clave: Aprendizaje basado en proyectos, Aprendizaje cooperativo, Espacio europeo de educación superior.

1. Introducción

La idea básica del aprendizaje basado en proyectos (PBL) [1] es organizar el curso (o parte del curso) entorno a un proyecto que se enuncia al inicio de la actividad y que los alumnos deben tratar de llevar a cabo en grupo, aprendiendo para ello lo que sea necesario. En un escenario PBL, los alumnos, en grupo, deben:

1. Identificar los conocimientos que ya tienen y que les pueden ser útiles para realizar el proyecto con éxito, e identificar los conocimientos que deberían adquirir.
2. Establecer un plan de aprendizaje para adquirir esos nuevos conocimientos.
3. Ejecutar el plan de aprendizaje

Este proceso se repite hasta que se alcanzan todos los objetivos de aprendizaje establecidos por el profesor. El rol del profesor, esencialmente consiste en:

- Crear buenos proyectos (que requieran adquirir los conocimientos y habilidades que se hayan establecido como objetivos del curso)
- Ayudar en el establecimiento del plan de aprendizaje

- Dar retroalimentación a tiempo para guiar a los alumnos en el proceso
- Dar soporte en la gestión del trabajo de los grupos

Un escenario PBL tiene múltiples ventajas potenciales, entre las cuales podemos destacar las siguientes:

- Promueve un aprendizaje activo (frente a una actitud más pasiva típica de las organizaciones más tradicionales, basadas en clases expositivas).
- Facilita el desarrollo de competencias transversales como trabajo en grupo, aprendizaje autónomo, comunicación, etc.
- Motiva a los estudiantes (especialmente si los proyectos son realistas y ambiciosos).

Por todo ello, se considera que PBL es una estrategia que encaja muy bien con los objetivos del Espacio Europeo de Educación Superior (EEES) y en particular, con el sistema de crédito europeo (ECTS), que promueve una programación docente basada en el aprendizaje del alumno, y no tanto en la enseñanza del profesor.

PBL es una estrategia docente muy extendida, especialmente en la enseñanza de las ingenierías (escenario en el que resulta especialmente adecuada). No obstante, PBL no es una estrategia fácil de usar, porque plantea muchos retos para el profesorado, especialmente en primeros cursos.

Un grupo de profesores de la Universidad Politécnica de Cataluña llevamos ya algún tiempo trabajando en la implantación de PBL en nuestras dos asignaturas de enseñanza de la programación, dentro del primer curso de las ingenierías técnicas de telecomunicación y de aeronáutica de la EPSC. Esta experiencia nos permite aportar algunas ideas que pueden ser de utilidad para profesores que se enfrenten a las dificultades mencionadas. En particular, la sección 2 de esta ponencia describe brevemente las asignaturas implicadas en la experiencia, las características más importantes de los proyectos que se están utilizando y la forma en que se organiza el curso usando esos proyectos como motor del aprendizaje, la sección 3 contiene algunas valoraciones sobre la experiencia y la sección 4 presenta algunas conclusiones.

2. Aprendizaje basado en proyectos en nuestras asignaturas de programación

En esta sección presentamos la organización y objetivos de Introducción a los Computadores (IC) y Laboratorio de Programación (LP), su esquema de evaluación, las características de los proyectos realizados, y como éstos se integran en el desarrollo del curso.

2.1. Organización y objetivos de las asignaturas

Los objetivos generales de IC son: la adquisición de unos conocimientos básicos de programación de ordenadores en lenguaje C, el uso adecuado de una herramienta de programación (Visual C++) y, como competencias transversales, el aprendizaje autónomo y el trabajo en grupo. Por otra parte, los objetivos de LP, que es la continuación natural de IC, son: la adquisición de unos conocimientos básicos de programación en un lenguaje orientado a objetos, el uso adecuado del entorno de programación Visual Studio (Visual C++ y Visual Basic), y se insiste en el aprendizaje autónomo y el trabajo en grupo.

Tanto IC como LP tiene 6 créditos. Según el plan de estudios vigente, a cada una de estas asignaturas le corresponden 4 horas de clase semanales que deben ir acompañadas de 4 horas adicionales de trabajo del alumno, a lo largo de 16 semanas (14 semanas de clase y dos semanas de exámenes). Esto da un total de 128 horas de dedicación del alumno (4,8 ECTS). En la organización docente actual para estas asignaturas [2], lo habitual es que el alumno tenga cada semana una sesión de 2 horas de clase y 6 horas de trabajo personal, aunque en tres de las 14 semanas la sesión de clase es doble (4 horas). En esas tres semanas el trabajo adicional es de 4 horas, para completar las 8 de dedicación semanal.

El trabajo fuera de clase consiste esencialmente en leer materiales, realizar ejercicios, realizar prácticas guiadas con el ordenador y trabajar en el desarrollo de los proyectos. A menudo, los alumnos tendrán que reunirse con sus compañeros para realizar algunas de estas actividades en grupo. En las sesiones de clase el profesor recogerá las dudas del trabajo semanal y aprovechará para aclarar aquellos aspectos que hayan generado más dificultades, propondrá ejercicios y realizará el seguimiento de los estudiantes. El resultado del trabajo dentro y fuera del aula va a dar lugar a lo que llamamos entregas. Estas entregas pueden ser hojas de dudas, ejercicios resueltos, auto-evaluaciones, etc., y facilitan la labor de seguimiento por parte del profesor.

El proyecto y otras actividades se realizan en grupo. Los grupos se forman en las primeras sesiones de clase con la intención de que se mantengan estables a lo largo de todo el curso. Los grupos son de tres personas (y si no es posible, de cuatro) y el criterio que consideramos más importante para su formación es la disponibilidad de tiempo libre en común para reunirse fuera de clase.

El esquema de evaluación es el mismo en ambas asignaturas. La nota final se obtiene a partir de los siguientes apartados:

- 20% entregas
- 30% conocimientos mínimos
- 40% proyecto
- 10% actitud y participación

Para aprobar el curso es imprescindible haber realizado el 80% de las entregas. La nota de esta componente se calcula en función del número de entregas realizadas teniendo en cuenta si hubo entregas fuera de plazo. La calidad de las entregas no afecta a la nota, pero las entregas mal hechas pueden ser rechazadas por el profesor y deberán realizarse de nuevo.

En IC se han identificado 8 problemas tipo que consideramos que todo alumno debe saber resolver individualmente para superar el curso con independencia de los resultados del proyecto. Estos problemas cubren distintos aspectos del temario y sirven para evaluar los conocimientos mínimos [3]. Es necesario resolver correctamente al menos 7 de estos problemas para aprobar. Se evalúan mediante las pruebas de mínimos y su peso en la nota final es del 30%. En LP la filosofía es la misma, aunque en este caso sólo hay 6 conocimientos mínimos, de los cuales hay que superar al menos 5. A diferencia de IC, en LP las pruebas se realizan con ordenador.

Al proyecto le corresponde el 40% de la nota. Como veremos más adelante, esta nota se desglosa en distintos apartados. Por último, un 10% de la nota depende de la actitud y de la participación del alumno durante el curso.

2.2. Características de los proyectos

Los proyectos llevados a cabo en las últimas ediciones de IC y LP son, respectivamente, la implementación del software de gestión de un vídeo-cajero y de un programa que permita al usuario jugar al conocido pasatiempo de los Sudokus.

Tanto en IC como en LP el proyecto incorpora todos (o casi todos) los elementos del temario del curso. En ambos casos el enunciado especifica un conjunto de funcionalidades mínimas que el proyecto debe soportar. Este conjunto de funcionalidades puede implementarse de distintas formas y pueden añadirse nuevas funcionalidades no previstas inicialmente, a criterio de los miembros del grupo. El proyecto de IC, al tratarse de la primera asignatura de programación a la que se enfrenta el alumno, es de tamaño reducido y de menor complejidad. El proyecto de LP es más extenso, incluye conceptos más avanzados, y está más abierto a propuestas que vayan más allá de los requisitos mínimos.

En todos los casos se intenta que el proyecto represente una carga de trabajo que requiera la participación de todos los miembros del grupo. Esto puede implicar una negociación entre los profesores y cada grupo en particular en cuanto al alcance de las propuestas de funciones adicionales que, por ejemplo, deben ser más ambiciosas y/o numerosas en el caso de los grupos de cuatro alumnos.

Proyecto del Vídeo-Cajero

El proyecto de IC consiste en el desarrollo de un programa que permita la gestión de un vídeo-cajero por parte de un encargado y el alquiler y devolución de películas por parte de los socios. Los usuarios interactúan con el programa mediante un sistema de menús en modo texto.

Por parte del encargado el programa va a permitir gestionar socios (altas/bajas/consultas) y películas (altas/bajas/actualización del número de copias y listados). En lo que a los socios se refiere, el programa va a permitir alquilar películas, devolver películas, recargar el saldo de la cuenta y modificar la clave de acceso. Dentro de la opción de alquiler, se podrán visualizar las novedades, la lista de películas de un determinado género o un listado completo de todos los títulos. Hay un primer prototipo que debe incluir básicamente las opciones del encargado y hay que añadir el resto para la versión final.

Veamos cómo el proyecto abarca los distintos temas que se tratan durante el curso:

- *Tipos de datos:* El almacenamiento de toda la información relativa a socios y películas requiere tanto de la utilización de tipos básicos (enteros, reales y caracteres) como de tipos estructurados (estructuras y vectores).
- *Esquemas algorítmicos:* Las consultas en modo listado de películas y socios requieren de la aplicación del esquema de recorrido. La localización de los datos del usuario que entra en la aplicación, la consulta de los datos de un socio y otras operaciones, como la eliminación de películas, deben realizarse aplicando el esquema de búsqueda.
- *Procedimientos y funciones:* El código debe estar organizado en procedimientos y funciones. En muchos casos las funciones van a devolver códigos de error u otra información y se van a utilizar los distintos tipos de paso de parámetros (por valor y por referencia mediante punteros). El uso de variables globales no está permitido.
- *Ficheros:* Toda la información referente a socios y películas debe almacenarse en ficheros de texto.
- *Organización y documentación del código:* Se pide que el código sea legible, esté bien organizado e incluya comentarios que indiquen la finalidad de cada función y aclaren el funcionamiento de las partes de código de mayor complejidad.
- *Uso del entorno de programación VC++:* Éste es el entorno que se va a utilizar y por tanto los alumnos tendrán que adquirir unos conocimientos básicos sobre su funcionamiento.
- *Aprendizaje autónomo:* Para facilitar el desarrollo de esta competencia se reducen al máximo las clases expositivas. Los alumnos deben estudiar materiales de autoaprendizaje en casa, y explicárselos entre ellos mediante la técnica del Puzzle [4].
- *Trabajo en grupo:* La elaboración del diseño, la planificación, el reparto de trabajo y la integración de las distintas partes son actividades que deben llevarse a cabo en grupo.

El juego del Sudoku

El proyecto de LP consiste en la implementación de un programa que permita jugar al pasatiempo de los Sudokus. El programa presentará una interfaz de usuario gráfica que se va a implementar mediante Visual Basic. El motor del programa se implementa en C++ y se accede desde el programa en Visual Basic mediante llamadas a funciones de una DLL.

El programa debe cargar sudokus almacenados en ficheros, permitir jugar y guardar a media partida, detectar valores erróneos, determinar si el sudoku se ha resuelto correctamente, disponer de una ayuda

que indique posibles valores para una casilla (descartando los valores que se encuentran en la misma fila/columna/bloque) y sugerir el siguiente valor a introducir de forma automática (detectar casos donde por descarte podemos deducir qué número corresponde a una casilla). Todas estas funcionalidades deben estar presentes en el primer prototipo del proyecto. La versión final añade la opción de deshacer la última decisión tomada y un apartado de rankings y estadísticas.

A continuación pasamos a analizar cómo se reflejan en el proyecto los distintos temas del curso:

- *Programación orientada a objetos*: El tablero y las casillas serán las principales clases del proyecto. La clase pila va a dar soporte a la opción de deshacer.
- *Programación visual*: Para diseñar y programar la interfaz gráfica de la aplicación se utilizará Visual Basic. Aunque no de forma explícita, se introduce a los alumnos en el paradigma de la programación dirigida por eventos.
- *Esquemas algorítmicos*: Los esquemas de ordenación y búsqueda se trabajan en las opciones de rankings y estadísticas.
- *Estructuras de datos avanzadas*: En la implementación del deshacer se utilizará la clase pila.
- *Gestión dinámica de la memoria*: Se puede utilizar en la implementación de la clase pila y también en la creación y eliminación de los objetos que ésta contiene.
- *Organización y documentación del código*: El código debe ser legible, estar bien indentado y documentado. Por primera vez los alumnos van a desarrollar un proyecto formado por varios módulos con sus respectivas cabeceras y ficheros de implementación.
- *Usar adecuadamente el entorno de programación Visual Studio (Visual C++ y Visual Basic)*: Los alumnos van a hacer uso de estas herramientas para desarrollar el proyecto.
- *Aprendizaje autónomo*: En LP se insiste en la técnica del Puzzle para que los grupos aprendan de forma autónoma los conceptos nuevos del curso. La implementación de funcionalidades extra también puede representar el aprendizaje autónomo de conceptos avanzados. Este es el caso de los grupos que se proponen incorporar la funcionalidad de resolución automática de sudokus, que deben aprender de forma autónoma las técnicas de recursividad y *backtracking*.
- *Trabajar en grupo*: De nuevo, la realización del proyecto, con sus etapas de diseño, planificación, reparto de trabajo, etc, inciden en el desarrollo de esta competencia.

Como vemos, puede ocurrir que alguno de los temas no sea tratado, a través del proyecto, con la profundidad deseable. Este es el caso, por ejemplo, de las colas (una estructura de datos avanzada que es objetivo formativo de LP). Las pruebas de mínimos que se realizan al margen del proyecto intentan garantizar que el alumno ha adquirido unos conocimientos suficientes también en estos temas.

La realización de un proyecto con interfaz gráfica confiere mayor atractivo al proyecto y lo aproxima a lo que a menudo vamos a encontrar en el mundo profesional. Se ha optado por la utilización de Visual Basic por su facilidad de uso. Aunque supone también el aprendizaje de un nuevo lenguaje, como contrapartida este esquema permite hacer hincapié en la separación entre interfaz gráfica y motor de ejecución.

Evaluación de los proyectos

En ambas asignaturas el proyecto, que corresponde a un 40% de la nota final de la asignatura, se califica con el mismo esquema:

- 10% primer prototipo
- 20% versión final
- 10% ampliación individual

La calificación del prototipo y la versión final es la misma para todos los componentes del grupo. Esta calificación se realiza en base unos criterios de calidad que se hacen públicos al inicio del proyecto.

La ampliación individual consiste en un ejercicio, realizado en clase y ante el ordenador, dónde se pide al alumno que realice algún pequeño cambio al proyecto. El cambio resultará trivial para el

alumno que domine la globalidad del proyecto, pero resultará difícil para el que se ha limitado a realizar su parte, sin implicarse en el trabajo de los compañeros. Un ejemplo de ejercicio de ampliación individual que cumple este propósito puede ser: *“Añade a los usuarios del video-cajero información sobre su edad, e incorpora una opción nueva que permita obtener los datos del usuario de mayor edad, para darle un regalo”*. La asignación de esta componente de la calificación del proyecto (que es individual) se realiza según el criterio siguiente:

- 0 si no se consigue realizar la ampliación
- 5 si la ampliación funciona correctamente
- 10 si TODOS los miembros del grupo realizan la ampliación correctamente

De esta forma se persigue que los miembros del grupo se sientan vinculados y tengan motivos para ayudarse mutuamente.

2.3. Integración de los proyectos en el desarrollo de las asignaturas

Las tablas 1 y 2 muestran, de forma esquemática, las actividades de las semanas clave para la realización del proyecto en IC y LP respectivamente. También nos permiten hacernos una idea del grado de planificación de las actividades de estas asignaturas.

En IC se asume que los alumnos llegan sin conocimientos previos de programación y la primera mitad del curso (semanas de la 1 a la 7) se dedica al aprendizaje de los aspectos más básicos. La realización del proyecto se concentra en la segunda mitad del curso (semanas de la 9 a la 15, que se muestran en la tabla 1). Los temas que los alumnos van a aprender como consecuencia de la realización del proyecto son esencialmente tres: funciones, estructuras y ficheros. El aprendizaje inicial de estos tres temas se realiza, durante las semanas 9, 10 y 11, mediante la técnica del puzzle [4], que consiste en repartir los tres temas a estudiar entre los miembros del grupo de modo que cada miembro se convierte en “experto” de su tema. Tras una puesta en común en clase con expertos de otros grupos (alumnos que han trabajado el mismo tema) los grupos se reúnen fuera de clase, durante la semana 10 para explicarse unos a otros cada uno de los tres temas. Para consolidar los nuevos conocimientos los alumnos realizan ejercicios de integración (donde hace falta combinar los distintos temas) en grupo y también a nivel individual. Estos ejercicios consisten en implementar algunas de las funcionalidades del proyecto y juntos constituyen el primer prototipo. Este prototipo consiste básicamente en un programa que carga los datos del video-cajero de fichero de texto a estructuras y realiza algunas operaciones simples de alta, baja y modificación de la información. Esta estrategia permite abordar el inicio del desarrollo del proyecto de forma no traumática. Un objetivo importante del primer prototipo es poder verificar que se está trabajando en la dirección correcta y permitir las rectificaciones necesarias para enfocar el desarrollo de la versión final con garantías de éxito. Por ello, en el plazo de una semana, el grupo recibe una calificación del profesor basada en los criterios de calidad establecidos, con abundantes comentarios para cada criterio de calidad, que resultarán útiles al grupo para encarrilar la versión final del proyecto.

La evaluación entre grupos de la semana 15 sirve para aprender de los aciertos y errores tanto propios como ajenos. Para realizar esta evaluación cada grupo prueba los proyectos de otros dos grupos (versión final), los evalúa utilizando los mismos criterios de calidad que finalmente utilizará el profesor y cumplimenta dos hojas con el resultado de su evaluación, una para el grupo corregido y otra para el profesor. La nota propuesta es de carácter orientativo y es el profesor quien decide la nota de la versión final de los proyectos.

En LP el proyecto está presente desde el inicio del curso. Aunque en ese momento todavía no se dispone de las herramientas suficientes para iniciar su desarrollo sí que es posible empezar a buscar información sobre Sudokus y pensar qué funcionalidades podría ofrecer el programa. Los alumnos hacen una propuesta preliminar de funcionalidades la semana 2, que se revisa la en la semana 5, y que, junto con las previstas en el enunciado, constituyen el punto de partida para el diseño del primer

prototipo. Su desarrollo se inicia la semana 7. La semana 10 empieza el trabajo para la versión final con su correspondiente documento de diseño y planificación. Para evitar que se acumule todo el trabajo para el final se pide a los alumnos que se comprometan a realizar entregas intermedias mostrando al profesor el estado del proyecto.

En LP también se utiliza la técnica del puzzle para el aprendizaje de los tres temas iniciales del curso: Visual Basic, Programación Orientada a Objetos y estructuras de datos avanzadas: pilas y colas.

El proyecto 0 de la semana 1 es un proyecto similar al de IC, pero de menor tamaño, que sirve para refrescar conocimientos previos y poner a prueba el funcionamiento de los grupos.

Gracias a que IC y LP comparten metodología los alumnos llegan a LP ya entrenados en la mecánica de trabajo. Esto permite que en esta asignatura el desarrollo del curso esté mucho menos pautado y que los alumnos asuman más responsabilidades. Así por ejemplo, el plan de trabajo para las semanas de la 11 en adelante (que no se muestran en la tabla 2) consiste esencialmente en realizar las tareas previstas de acuerdo con la planificación propia de cada grupo.

Por último, vale la pena mencionar el hecho de que existen dos competencias transversales específicas que se desarrollan de forma coordinada entre ambas asignaturas, aprovechando el hecho de que son consecutivas en el plan de estudios. En particular, la habilidad para planificar proyectos se empieza a trabajar en IC, ya que en la semana 12 se pide a los grupos que realicen una planificación de la versión final del proyecto. Para ello se usa una herramienta de libre distribución que les permite construir diagramas de Gantt y diagramas de recursos [5]. Este tema se retoma en LP, donde se les pide que, usando la misma herramienta, realicen, en la semana 5, la planificación de la primera versión del proyecto, y en la semana 10 la planificación de la versión final. Estos tres ejercicios, con los comentarios apropiados de los profesores, mejoran significativamente la capacidad de planificar el trabajo.

Tabla 1: Sesiones destacadas de IC

Sem	Tareas en clase (2 horas)	Trabajo personal o en grupo (6 horas)
9	Repaso de los resultados de la primera prueba de mínimos (1h) Presentación del proyecto (30') Actividad <i>¿Qué puede salirnos mal?</i> , con resultado parcial de grupo: Reglas de funcionamiento del grupo (20') Instrucciones para el puzzle: estructuras, ficheros y funciones (10')	Estudio individual puzzle (3h) Práctica guiada sobre herramienta de planificación (1h 30')
10	Reunión de expertos (1h) Ejercicio para expertos (1h)	Preparar explicación para el grupo base (2h) Reunión con el grupo base (2h) Ejercicios de compañeros (2h)
11	<i>SESIÓN DOBLE</i> Clase expositiva sobre los materiales del puzzle (1h) Práctica guiada sobre paso de parámetros por referencia (1h) Ejercicio de integración de grupo (2h)	Repaso de los materiales del puzzle de los que no se es experto (2h) Acabar el ejercicio de integración de grupo (2h)
12	Presentación del ejercicio de integración de grupo (1h) Ejercicio de integración individual (1h)	Acabar el ejercicio de integración individual (2h) Puesta a punto del primer prototipo (3h) Diseño y planificación de la versión final (2h)
13	Entrega del primer prototipo (1h) Discusión sobre los diseños de la versión final (1h)	Preparación de la prueba de mínimos (2h) Desarrollo de la versión final (4h)
14	Segundo control de mínimos (2h)	Desarrollo de la versión final (6h)
15	<i>SESIÓN DOBLE</i> Últimos retoques de la versión final (1h) Evaluación entre grupos (1h) Ejercicio individual de ampliación (1h 30') Cuestionarios de final de curso (30')	Última oportunidad para superar los mínimos (durante la semana de exámenes finales) (2h)

Tabla 2: Sesiones destacadas de LP

Sem	Tareas en clase (2 horas)	Trabajo personal o en grupo (6 horas)
1	Presentación del curso (20') Presentación del proyecto del curso (15') Presentación del proyecto 0 (15') Organización de los grupos y primeros pasos proyecto 0 (1h)	Tareas proyecto 0 (4h) Tareas proyecto 1 (2h)
2	Puesta a punto proyecto 0 (1h) Puesta en común material Sudokus (30') Presentación temas del puzzle: Visual Basic, programación orientada a objetos y estructuras de datos avanzadas (30')	Trabajo individual puzzle (4h) Documento de funcionalidades proyecto del curso (1h 30')
5	Progresar en el desarrollo de los ejercicios de integración (1h) Discusión sobre funcionalidades del prototipo 1 (1h)	Ejercicio de integración (2h) Documento de diseño y planificación del prototipo 1 (2 h) Estudiar otros materiales del puzzle (2h)
7	<i>SESIÓN DOBLE</i> Prueba de mínimos (2h) Discusión sobre los documentos de diseño y primeros pasos prototipo 1 (2h)	Desarrollo prototipo 1 (4h)
10	Puesta a punto y presentación prototipo 1 (1h) Presentación nuevos temas: gestión dinámica de memoria y algoritmos de ordenación (1h)	Trabajo individual nuevos temas (4h) Documento de diseño y planificación de la versión final (2h)
11	Explicación y dudas nuevos temas (1h) Discusión sobre los documentos de diseño (1h)	Desarrollo versión final (6h)
15	<i>SESIÓN DOBLE</i> Puesta a punto versión final (1h) Evaluación entre grupos (1h) Ejercicio de ampliación (1h 30') Cuestionarios de final de curso (30')	Última oportunidad de superar los conocimientos mínimos (durante la semana de exámenes finales) (4h).

Por otra parte, los alumnos realizan varias tareas que les ayudan a mejorar su capacidad de trabajo en equipo. Por ejemplo, en ambas asignaturas, al poco de iniciarse el proyecto, cada grupo escribe sus reglas de funcionamiento interno, para prevenir los problemas típicos del trabajo en grupo. Hay también, de vez en cuando, tareas de reflexión sobre el funcionamiento de grupo, que ayudan a los alumnos a identificar correcciones en la forma de trabajar. Finalmente, la última semana tanto en IC como en LP, los alumnos rellenan un cuestionario que les ayuda a hacer un balance final de su experiencia con el trabajo en grupo.

3. Valoraciones

Tanto en IC como en LP, a lo largo del curso los alumnos contestan 4 cuestionarios de respuesta abierta donde deben señalar qué es lo mejor y qué es lo peor del curso hasta ese momento. Los resultados de todas ellas son bastante coincidentes. Entre los comentarios negativos destacan los siguientes:

Necesito más clases teóricas. Los alumnos no acaban de acostumbrarse a la idea de que pueden enfrentarse a los problemas sin haber aprendido toda la teoría necesaria. Tienden a achacar las dificultades a la falta de horas de teoría en las que el profesor les explique con antelación todo lo que deben saber.

Demasiado tiempo para las actividades. Los alumnos se quejan de la excesiva carga de trabajo que suponen las asignaturas. Lo cierto es que los datos recogidos mediante cuestionarios de tiempo de dedicación muestran una dedicación semanal ligeramente inferior a las ocho horas previstas.

Los comentarios positivos más frecuentes son los siguientes:

El trabajo en grupo me gusta mucho. Esto siempre es así. Si bien puede haber casos aislados de alumnos que prefieren trabajar solos, cualquier encuesta de opinión pondrá de manifiesto que a los alumnos les gusta trabajar en grupo.

Es una asignatura muy práctica. Los alumnos agradecen el enfoque práctico de la asignatura. Aprender aplicando los conocimientos les resulta una experiencia gratificante.

Se agradece la planificación detallada. Consideran que disponer de una planificación semanal detallada resulta muy útil para el seguimiento del curso.

He aprendido mucho. Al final del curso los alumnos se sorprenden de los logros alcanzados. Además son muy conscientes de haberlo conseguido gracias a su esfuerzo. Este es un aspecto que se ve especialmente potenciado por la organización basada en proyectos.

Estos resultados se complementan con los obtenidos mediante una encuesta final donde los alumnos valoran de 1 (en desacuerdo) a 5 (totalmente de acuerdo) distintos aspectos del curso. La Tabla 3 muestra los resultados de esta encuesta. Los resultados ponen de manifiesto, por encima de todo, la sensación de haber aprendido mucho. Por lo demás, el curso fue muy bien valorado por los alumnos en todos los aspectos.

Tabla 3: Resultados de la encuesta final

En este curso he aprendido cosas que considero valiosas para mi formación	4,2
La labor del profesor me ha facilitado el proceso de aprendizaje	3,32
El material del curso está bien preparado y es adecuado	3,44
En todo momento he tenido claro lo que tenía que hacer (tanto en clase como fuera de clase)	3,43
Siempre me he sentido bien informado sobre mi progreso (o falta de progreso) en el curso	3,72
El trabajo en grupo me ha resultado de gran ayuda	3,66
La forma de evaluación me ha parecido adecuada	3,53
Este curso me ha ayudado a mejorar la gestión que hago de mi propio tiempo	3,3

La tabla Tabla 4 muestra el rendimiento obtenido en las dos asignaturas, en sus últimas imparticiones. Se observa en IC un rendimiento aceptable (más del 55% de aprobados) aunque con una tasa de NP elevada. Se trata de alumnos que muy pronto abandonan el trabajo del curso porque no pueden dedicar el tiempo previsto, o porque les falta hábito de trabajo continuado (recordemos que IC es una asignatura que cursan en su primer cuatrimestre en la Universidad). Los resultados mejoran mucho en LP, ya que los alumnos están más habituados a la metodología de trabajo, y no tienen excesivas dificultades para seguir el plan de trabajo.

Tabla 4: Rendimiento académico

Calificación	IC		LP	
	Alumnos	Porcentaje	Alumnos	Porcentaje
Excelente-MH	20	8,5	27	14,9
Notable	72	30,9	99	54,6
Aprobado	38	16,3	31	17,1
Suspendido	32	13,7	20	12,6
N.P.	71	30,4	4	1,6
TOTAL	233	100	181	100

4. Conclusiones y consideraciones finales

En esta ponencia se ha descrito la aplicación de estrategias PBL en dos asignaturas de programación de ordenadores. Somos conscientes que nuestra metodología no responde a una aplicación pura del aprendizaje basado en proyectos (una aplicación más pura daría a los alumnos más libertad para determinar sus objetivos de aprendizaje), pero sí retiene sus principales beneficios, en particular:

- Promueve un aprendizaje activo.
- Facilita el desarrollo de competencias transversales.
- Motiva a los estudiantes.

La aplicación de esta metodología resulta novedosa para la mayoría de los alumnos y asimilar el funcionamiento de las asignaturas requiere un cierto esfuerzo. Por ello es clave que se aplique con continuidad, como hacemos en IC y LP. En muchos aspectos IC es un entrenamiento para lo que vendrá después. Los resultados indican que el rendimiento mejora mucho en LP respecto a IC. También para el profesor esta transformación resulta un reto importante y puede ser conveniente plantearse una transición progresiva, empezando con un proyecto corto y asequible, que puede volverse más ambicioso en iteraciones sucesivas.

Hay que ser conscientes y aceptar que el aprendizaje de los alumnos no es homogéneo a lo largo de todos los temas ni tampoco entre los distintos alumnos a nivel individual. En nuestro caso, el mecanismo para intentar garantizar que todos los alumnos tienen unos conocimientos suficientes de todos los temas son las pruebas de conocimientos mínimos.

Finalmente, observamos con preocupación el elevado número de abandonos en IC. Si bien tradicionalmente se trata de una asignatura con muchos abandonos uno de los propósitos de la metodología es mejorar este aspecto. Debemos analizar qué medidas nos pueden permitir paliar esta situación. Salvo este aspecto, valoramos muy positivamente los resultados de la implantación de la metodología. La mayoría de los alumnos que siguen el plan previsto consiguen los objetivos de aprendizaje, reconocen que han aprendido mucho y hacen una valoración global positiva.

5. Referencias

- [1] MARKHAM, T. *Project Based Learning, a guide to Standard-focused project based learning for middle and high school teachers*. Buck Institute for Education, 2003.
- [2] ANGUAS, J., DÍAZ, L., GALLEGO, I., LAVADO, C., REYES, A., RODRÍGUEZ, E., SANJEEVAN, K., SANTAMARIA, E., VALERO, M. *Una experiencia de adaptación al EEES de dos asignaturas de programación de ordenadores*. 4º Congreso Internacional de Docencia Universitaria e Innovación (CIDUI 2006). Barcelona, 5 a 7 de Julio 2006.
- [3] DEL CANTO, P., GALLEGO, I., HIDALGO, R., LÓPEZ, J., LÓPEZ, J.M., MORA, J., RODRÍGUEZ, E., SANTAMARIA, E., VALERO, M., *Cómo congeniar los exámenes y los proyectos en asignaturas PBL*. XIII Jornadas de la Enseñanza Universitaria de la Informática (JENUI 2007). Teruel, 16 a 18 de Julio 2007. [Aceptado, pendiente de publicación]
- [4] ANGUAS, J., DÍAZ, L., GALLEGO, I., LAVADO, C., REYES, A., RODRÍGUEZ, E., SANJEEVAN, K., SANTAMARIA, E., VALERO, M. *La técnica del Puzzle al servicio del aprendizaje de la programación de ordenadores*. XII Jornadas de la Enseñanza Universitaria de la Informática (JENUI 2006). Bilbao, 12 a 14 de Julio 2006.
- [5] Gantt Project, <http://ganttproject.biz/>