

# Trabajo en equipo en proyectos de desarrollo de software: estrategia docente e infraestructura software

Patricio Letelier, M<sup>a</sup> Carmen Penadés y Juan Sánchez

Departamento de Sistemas Informáticos y Computación

Universidad Politécnica de Valencia

{letelier, mpenades, jsanchez}@dsic.upv.es

## Resumen

En términos de contratación y promoción, cada vez se valora más la capacidad de un individuo para participar de manera efectiva en un equipo. El trabajo en equipo es un escenario laboral habitual en proyectos de desarrollo de software. En este artículo presentamos una estrategia docente que hemos venido aplicando y refinando, tanto en escuela como en facultad, para ofrecer a nuestros alumnos una experiencia formadora de trabajo en equipo dentro del contexto de metodologías para el desarrollo de software. Además, describimos la implantación de una infraestructura software de apoyo al trabajo cooperativo, la cual hemos incorporado este último año académico.

## 1. Introducción

En la actualidad la mayor parte del software no puede desarrollarse de manera individual y exige la participación de equipos de desarrollo, que en algunos casos ni siquiera comparten el mismo lugar o país de trabajo. Así, para un ingeniero informático es importante contar con los conocimientos y habilidades para participar eficazmente en un equipo de trabajo en el contexto de un proyecto de desarrollo de software. Sin embargo, en muchas universidades los estudiantes de informática terminan la carrera sin haber desarrollado un proyecto software de tamaño significativo y sin haber trabajado en un equipo de desarrollo.

En la UPV, desde hace 5 años hemos venido depurando una estrategia para integrar de manera

efectiva en asignaturas de último curso (en escuela y en facultad) la experiencia práctica del trabajo en equipo en el marco de metodologías para el desarrollo de software. Las asignaturas involucradas son: Laboratorio de Desarrollo de Sistemas de Información (LSI-ETSIA), optativa de 3er año de la Escuela Técnica Superior de Informática Aplicada y Laboratorio de Sistemas de Información (LSI-FI), optativa de 5º año de la Facultad de Informática. Paulatinamente hemos ido introduciendo en estas asignaturas mejoras e innovaciones (métodos activos [8], evaluación continua [7], metodologías de desarrollo [5,6], trabajo en equipos, etc.), todas ellas avaladas por la entusiasta participación y positiva valoración de los alumnos.

El objetivo de este artículo es describir el estado actual de nuestra propuesta docente en dichas asignaturas, y especialmente, presentar los resultados de la reciente puesta en marcha de una infraestructura software para apoyar el trabajo en equipo, la cual permite a los miembros de cada equipo compartir información relativa al proyecto a través de la Web.

Lo que resta del artículo está organizado como se indica a continuación. La sección 2 presenta el método docente y de evaluación en las asignaturas involucradas, así como la planificación de las actividades. En la sección 3 se describe la infraestructura software que estamos utilizando en dichas asignaturas para apoyar el trabajo en equipo. En la sección 4 se comentan algunos trabajos relacionados. Finalmente, la sección 5 presenta las conclusiones y trabajo futuro.

## 2. Trabajo en equipo en asignaturas LSI

LSI-FI y LSI-ETSIA son asignaturas de 6 créditos. El horario de la asignatura se organiza en una sesión semanal de 3 horas en laboratorio y una sesión de 2 horas, cada 2 semanas y en aula. Normalmente se dispone de 12 sesiones de laboratorio y 6 sesiones de aula. La matrícula en estas asignaturas oscila entre 40 y 50 alumnos.

El objetivo específico de LSI-FI y LSI-ETSIA es: *“capacitar al alumno para participar efectivamente en un proyecto de desarrollo de software siguiendo una metodología”*. Esto supone conseguir que el alumno sea capaz de:

- Aplicar una metodología de desarrollo en un proyecto software
- Trabajar en un equipo de desarrollo, desempeñando un rol
- Realizar la planificación y control del tiempo del proyecto
- Comunicarse con el cliente para capturar y validar los requisitos
- Negociar con el cliente las características del producto incorporadas en los plazos establecidos
- Realizar presentaciones de seguimiento del proyecto mostrando los resultados al cliente

Esencialmente en estas asignaturas se aborda un proyecto de desarrollo en equipo siguiendo una metodología. Los alumnos se organizan en equipos de 6 a 8 integrantes. Cada equipo tiene asignado dos profesores, uno actuando como cliente y el otro como instructor. El cliente atiende todo lo referente a requisitos del producto y a las características incorporadas en los plazos establecidos. El instructor ofrece un apoyo metodológico a lo largo del proyecto. Esta separación de roles de los profesores hace más realista el desarrollo de proyecto y permite que los alumnos ejerciten distintos mecanismos de interacción: equipo-cliente (captura de requisitos y negociación) y equipo-instructor (asesorando en la correcta aplicación de la metodología seguida).

### 2.1. Método docente

Básicamente los tipos de actividades que se realizan a lo largo del curso son:

- Sesiones de trabajo en equipo, realizadas en horario de laboratorio, o bien en horarios acordados por propio equipo, al que asisten todos o parte de los integrantes
- Sesiones de trabajo con el cliente realizadas en horario de laboratorio y en tutorías
- Sesiones de trabajo con el instructor, realizadas en horario de laboratorio y en tutorías.
- Presentaciones de seguimiento ante el cliente, al final de cada fase o iteración.
- Presentaciones técnicas de la arquitectura ante el instructor. Se realizan dos; una revisión de la arquitectura de los datos y otra de la arquitectura del código.

En la primera sesión se realiza la presentación de la asignatura; se comentan los objetivos, el método docente, el método de evaluación, etc. Además, en esta sesión se da comienzo oficial al proyecto de desarrollo, haciendo públicos los equipos. Los equipos se configuran previamente, respetando sólo dos criterios: que los integrantes sean del mismo grupo horario e intentando una distribución equitativa por sexo. Cada equipo tiene establecida una metodología de desarrollo. Cada integrante tiene asignado un rol (Jefe, Analista/Diseñador, Programador, Tester, etc.), de acuerdo con la metodología de desarrollo.

En LSI-FI la mitad de los equipos trabaja con una metodología ágil y la otra con una metodología de corte tradicional. Hemos seleccionado Extreme Programming (XP) [1] como metodología ágil por ser sin lugar a dudas la más popular y de la cual se dispone de mayor cantidad de información. Por otra parte, utilizamos Rational Unified Process (RUP) [4] como metodología tradicional, por contar con una especificación muy detallada. Aunque RUP es una marca comercial, la suscripción al IBM Scholars Program<sup>1</sup> permite disponer de todo el material asociado (y de otros productos) para propósitos académicos. Lo ideal sería que los alumnos pudiesen experimentar con ambos tipos de metodologías y desempeñando todos los roles, pero por las restricciones naturales de una asignatura esto no es factible. Sin embargo, la

<sup>1</sup> [www.developer.ibm.com/university/scholars/](http://www.developer.ibm.com/university/scholars/)

puesta en común que se realiza en las sesiones de seguimiento permite a los alumnos conocer el tra-

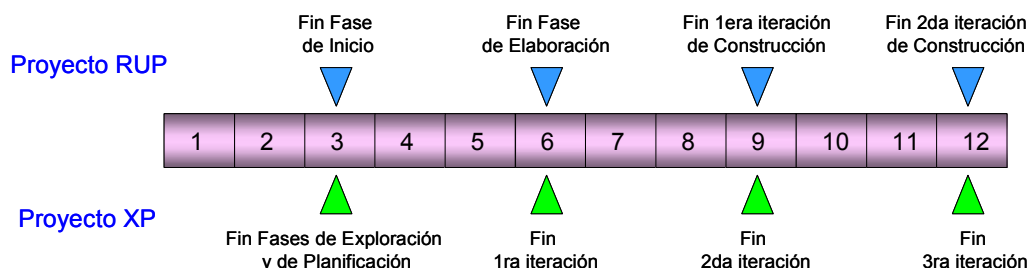


Figura 1: Planificación del proyecto

bajo de otros equipos. En la FI, debido a que los alumnos no tienen una asignatura previa donde se describan dichas metodologías, realizamos dos presentaciones a modo de introducción, proporcionándoles además material complementario. A la semana siguiente de las presentaciones se efectúa un breve examen de conocimientos de ambas metodologías.

En LSI-ETSIA, sin embargo, todos los equipos trabajan con la misma metodología, XP. Los alumnos de la ETSIA tienen la posibilidad de cursar una optativa previa llamada “El Proceso del Software” (PSW), en la cual se introducen de forma teórica las metodologías de desarrollo y los estándares de proceso software. Sin embargo, de acuerdo con nuestra experiencia consideramos que resulta más adecuado y efectivo trabajar con una metodología ágil en 3er curso con los Ingenieros Técnicos en Informática. Pensamos que una metodología ágil está más cercana al perfil profesional y entorno de trabajo al que se enfrentarán. Además, salvo algunas prácticas puntuales tales como pruebas y *refactoring*, la sencillez de XP favorece el hecho que los alumnos asimilen rápidamente la metodología. A los alumnos que no han cursado previamente PSW se les proporciona un documento resumen de XP que les permite nivelar sus conocimientos al respecto.

## 2.2. Planificación del proyecto

La Figura 1 muestra la planificación global del proyecto a lo largo de las 12 semanas disponibles (sin incluir días festivos ni vacaciones). La sesión de presentación e inicio del proyecto corresponde a la semana 0, no incluida en la Figura 1. Los

principales hitos del proyecto tanto para equipos RUP como para equipos XP son distribuidos en el calendario de manera que cada hito coincide con la presentación de seguimiento respectiva. Además, alrededor de las semanas 7-8 y 10-11 se realizan las dos presentaciones para revisión de la arquitectura. Esta planificación, ajustada al calendario académico, es inamovible para así lograr una adecuada duración y número de iteraciones. Aunque es difícil, se intenta hacer coincidir las presentaciones de seguimiento con las sesiones de aula, establecidas oficialmente cada dos semanas. Sin embargo, la no coincidencia con un par de sesiones oficiales de aula no ha presentado mayores inconvenientes, principalmente porque no supone mayor dedicación horaria de parte del alumno y tampoco deberían producirse solapes con otras asignaturas.

Tanto en los proyectos RUP como XP el producto no se finaliza por completo durante la asignatura. En el caso de RUP, faltarían quizás algunas iteraciones de construcción y la posterior Fase de Transición. En el caso de XP serían necesarias iteraciones adicionales para tener un producto totalmente operativo. Sin embargo, de acuerdo con los objetivos de la asignatura esto no reviste mayor importancia, puesto que es más relevante que los equipos asimilen la rutina de iteraciones, presentaciones de seguimiento, control de su productividad y negociación de compromisos con el cliente.

A continuación se describe brevemente lo que implica cada fase/iteración. Los detalles metodológicos quedan fuera del alcance de este artículo.

En las tres primeras semanas de trabajo, los equipos XP realizan las Fases de Exploración y de

Planificación. El resultado final es la obtención del conjunto de historias de usuario y la planificación de la entrega. En estas sesiones hay una gran interacción entre los miembros del equipo y el cliente, pues se trata de establecer requisitos y acordar una planificación. Para los equipos RUP estas primeras semanas corresponden a la Fase de Inicio. En esta fase los equipos RUP suelen requerir mucho más apoyo del instructor que los equipos XP, principalmente en lo que respecta a comprender los artefactos que se utilizarán y la forma incremental e iterativa en la cual se generan. Por este motivo, damos ya definida una configuración de RUP para proyecto pequeño, exigiendo la utilización de sólo algunos artefactos. Entre ellos: Plan de Desarrollo, Glosario, Visión, Modelo de Casos de Uso, Modelo de Análisis/Diseño, Modelo de Datos, Modelo de Implementación, Modelo de Despliegue, Casos de Prueba y Código. Así, el hito para los equipos RUP en la Fase de Inicio se concentra en el avance de los artefactos Plan de Desarrollo y Visión.

Posteriormente, los equipos XP desarrollan tres iteraciones de construcción, según los contenidos inicialmente pactados con el cliente y las negociaciones por cambios en los requisitos. La planificación se va ajustando según se tiene un conocimiento más exacto de la productividad del equipo (medida en XP como la Velocidad de Proyecto).

Los equipos RUP durante la Fase de Elaboración tienen como objetivo avanzar suficientemente el Modelo de Casos de Uso y en menor medida los otros artefactos que determinan la arquitectura del sistema. Con esto deberían ser capaces de planificar los casos de uso que serán implementados en las dos iteraciones de construcción que se efectuarán a continuación.

Tanto en el caso de XP como de RUP, las presentaciones de seguimiento en las iteraciones de construcción incluyen los siguientes aspectos: revisión del plan indicando posibles incidencias, demostración de la versión actual del producto, resultados de las pruebas de aceptación, métricas de tamaño del producto y productividad del equipo.

### 2.3. Método de evaluación

Se utiliza un esquema de evaluación continua basado en los hitos del proyecto. Cada uno de los

cuatro hitos y sus correspondientes presentaciones de seguimiento tienen una calificación doble. Además, se otorga otra calificación doble a cada una de las revisiones técnicas. También se evalúa con una calificación simple los siguientes conceptos: el examen de conocimientos RUP y XP (sólo en LSI-FI), la actualización de los Registros de Tiempos (donde se registra la dedicación de cada integrante en el proyecto según lo propuesto en Personal Software Process [3]) y la actualización de los artefactos en el sitio Web del equipo. Con todo esto, cada alumno tiene alrededor de 10 calificaciones, las cuales son simplemente promediadas para obtener la nota final.

La fecha de la presentación de la última iteración se hace coincidir con la fecha oficial del examen de la asignatura. Es decir, en lugar de examen final simplemente se realiza una presentación de seguimiento del proyecto, similar a las anteriores.

Un elemento importante a mencionar es el procedimiento seguido para establecer la calificación de cada alumno en los hitos del proyecto y las revisiones técnicas. Inmediatamente después de las presentaciones correspondientes, el cliente y el instructor acuerdan una nota para cada equipo. Dicha nota se multiplica por el número de integrantes del equipo y esta cantidad se le hace llegar por email al jefe del equipo, quien debe acordar su distribución con el resto de integrantes y devolver la puntuación decidida para cada uno de los miembros. Este mecanismo ha resultado efectivo para establecer necesarias diferencias en cuanto a la dedicación y compromiso con el proyecto. Por otra parte esto añade una cuota de realismo en cuanto a la valoración y crítica del trabajo de cada integrante.

### 3. Infraestructura software de apoyo

La coordinación y el compartir información habían sido dos aspectos problemáticos detectados en años anteriores. Los alumnos disponen de un espacio privado asociado a sus cuentas de usuario, pero este no es fácil de acceder desde fuera del campus. Por otra parte cada asignatura dispone de un sitio web denominado microweb donde se les puede dejar disponible material a los alumnos, es decir, es sólo comunicación en un sentido, desde el profesor al alumno. Claramente estos recursos

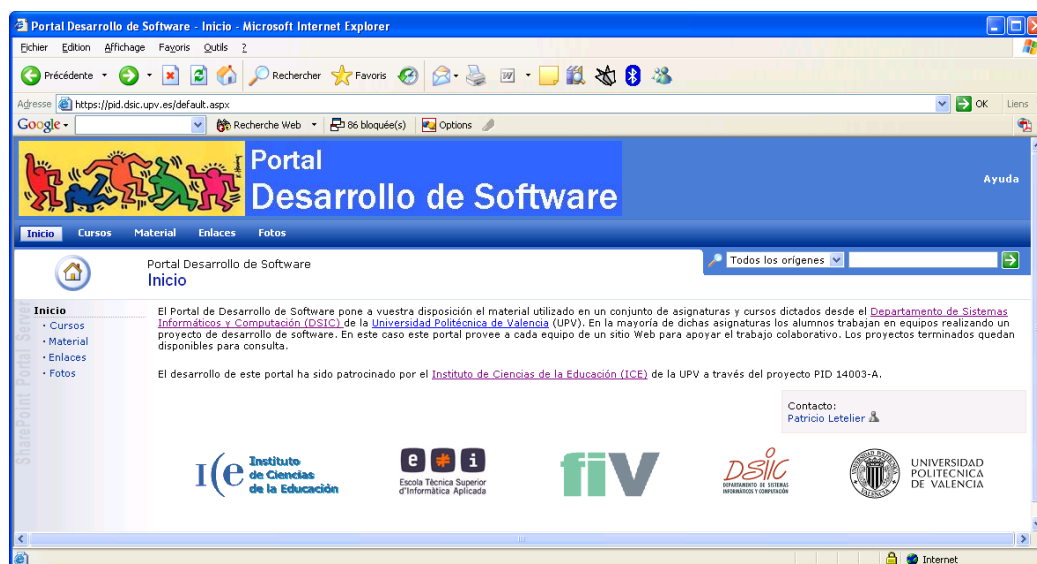


Figura 2: Página de inicio del Portal

resultaban insuficientes para conseguir una interacción adecuada entre los integrantes de un equipo, su cliente y su instructor. Ante esto, algunos equipos utilizaban discos virtuales y foros para solventar estas deficiencias.

El año 2004 planteamos como Proyecto de Innovación Docente (PID) la implantación de una infraestructura software de apoyo al trabajo cooperativo y nos fue concedido un becario y una máquina para actuar como servidor. Estudiamos algunas herramientas para gestión de contenidos y trabajo cooperativo, seleccionando finalmente Microsoft SharePoint. Su elección se basó principalmente en el hecho que la tenemos disponible a través de la suscripción de nuestro departamento a la Academic Alliance de Microsoft. Por otra parte, SharePoint es muy sencillo de instalar, configurar y utilizar, ofreciendo una serie de integraciones interesantes con otros productos Microsoft. Además, las extensiones que hemos implementado se realizan fácilmente como *webparts*, módulos programados en C#.

Los objetivos planteados para esta infraestructura software de apoyo se resumen a continuación:

- Proveer a cada equipo de un sitio web para compartir información entre ellos y con su cliente e instructor

- Proporcionar en línea plantillas para la generación de artefactos del proyecto
- Facilitar la comunicación en el equipo, haciendo disponible la información de contacto, correo a todo el equipo y tablón de anuncios.
- Ofrecer espacio para descarga de material de las asignaturas
- Ofrecer enlaces de interés para las asignaturas
- Ofrecer facilidades de administración para la configuración de equipos y sitios asociados
- Proveer mecanismos de control de acceso para la información privada de cada equipo y su posterior publicación una vez acabado el curso
- Proveer facilidades para seguimiento y evaluación de los equipos

Hasta el momento disponemos de una primera versión de esta infraestructura, que ha sido utilizada con éxito durante este curso académico en las dos asignaturas involucradas.

La Figura 2 muestra la página de inicio del portal (<https://pid.dsic.upv.es>). En el apartado cursos se encuentra la información de los proyectos en curso y terminados en cada asignatura así como el calendario de actividades

Figura 3: Sitio Web de equipo

de la asignatura. Cada integrante tiene acceso sólo al sitio web de su equipo. Una vez terminado los proyectos, éstos quedan disponibles a cualquier visitante del portal en modo sólo consulta. Además, hay un apartado de material para las asignaturas y de enlaces de interés. También existe un espacio de fotografías para almacenar fotos de los equipos y del desarrollo de las actividades.

La Figura 3 muestra la página web del sitio de un equipo XP. En ella podemos observar los siguientes elementos:

- Información de contacto de los integrantes del equipo
- Envío de email a todos los integrantes del equipo
- Tablón de anuncios
- Información de contacto del cliente y del instructor
- Registros de Tiempo de cada uno de los integrantes del equipo
- Enlaces de interés para los miembros del equipo
- Pestañas para los distintos grupos de artefactos generados, de acuerdo con la metodología que utiliza el equipo. Para cada artefacto se proporciona una plantilla directamente accesible al crear un nuevo artefacto.

Cada integrante de equipo puede modificar su información personal en el sitio de su equipo así como toda la información compartida. En la pestaña “Seguimiento” los equipos deben dejar

disponible todo el material expuesto en cada presentación de seguimiento o técnica.

La configuración del portal mediante las facilidades de SharePoint es muy sencilla. La mayor parte de la funcionalidad descrita anteriormente se puede implantar arrastrando elementos predefinidos en las áreas correspondientes del portal. Sin embargo, hay un esfuerzo considerable tras la definición manual de los sitios de equipos, los usuarios de cada sitio, el establecimiento de los permisos y todos los posteriores cambios que suelen presentarse en los equipos (particularmente altas y bajas de integrantes). Es por esto que se desarrollaron *webparts* específicas para la gestión del portal, ofreciendo facilidades para cargar los grupos de alumnos desde los listados de matriculados, gestión de equipos con la creación automática de los correspondientes sitios y permisos. La creación de los sitios de equipos se realiza a partir de plantillas predefinidas de sitios según la metodología del equipo. También se ha automatizado la finalización del proyecto, en la cual los proyectos pasan a un estado terminado quedando disponibles para consulta de los visitantes y quitando los permisos de modificación a los integrantes del equipo.

#### 4. Trabajos relacionados

A continuación comentaremos algunas propuestas docentes (publicadas) cercanas a la nuestra en cuanto a su interés por conseguir una formación de equipos de desarrollo y el trabajo de coordinación de los mismos.

En [9] se describe una experiencia en la organización de un curso de análisis y diseño orientado a objetos dirigido a estudiantes que ya han cursado alguna materia de ingeniería del software y que tienen cierta práctica en lenguajes de programación como Java. El caso de estudio o proyecto aborda el desarrollo de un sistema software que se ha dividido en seis subsistemas. Cada equipo de desarrollo debe abordar la construcción de un subsistema y la integración con el resto. Para cada subsistema existen 3 versiones (3 equipos de desarrollo distintos de 5 personas) que deben negociar la venta de su producto al resto de los grupos, y también la compra de los otros subsistemas. En la experiencia se aborda el trabajo dentro de un

equipo, la coordinación con el resto de los equipos y se resalta la importancia de utilizar estándares de documentación y comunicación.

En la Universidad de Kiel, D. Frosch ([2]) propone un curso de ingeniería del software para estudiantes de sistemas de información y de administración de empresas, que se centra en los aspectos prácticos de UML para la construcción de modelos de requisitos. El objetivo del curso es simular un entorno real de trabajo en el cual los estudiantes puedan demostrar sus conocimientos prácticos de UML en el desarrollo de un caso para un cliente, en este caso una empresa real de desarrollo de software. Los estudiantes se organizan en equipos que deben encargarse de gestionar el proyecto, estimar costes y mantener una comunicación fluida con el cliente. Con respecto a la evaluación de resultados se tiene en cuenta la utilización de UML, la cooperación real con el cliente y la cooperación y comunicación entre los componentes de cada grupo.

Por último K. Todd Stevens [10] describe una propuesta de enseñanza de ingeniería del software que prima los aspectos prácticos (desarrollo de casos de estudio) sobre los contenidos teóricos. El núcleo central lo constituye el desarrollo en equipo de un proyecto o caso de estudio. Lo más destacable en nuestra opinión de la experiencia radica en los comentarios de los estudiantes referentes a la utilidad del trabajo en equipo, la interacción individual, la importancia de una buena planificación y la necesidad de una buena gestión o dirección del equipo.

#### 5. Conclusiones

En este artículo hemos presentado nuestra estrategia de formación en trabajo en equipo en el contexto de metodologías de desarrollo de software. El esquema planteado ha sido fruto de una serie de refinamientos y desafíos que emprendimos hace cinco años. Estamos muy satisfechos por los resultados obtenidos, confirmados por la positiva valoración de los alumnos a través de las encuestas y al alto grado de participación y compromiso demostrado en las actividades involucradas. Desde el punto de vista docente estamos convencidos que los objetivos planteados por las asignaturas LSI se cumplen y el alumno consigue experimentar una situación de trabajo en equipo que resulta ejemplar respecto a

las competencias que tendrá que exhibir en su vida profesional como ingeniero de software.

Los Registros de Tiempo de los alumnos nos han permitido confirmar que el tiempo promedio de dedicación del alumno está alrededor de 110 horas, incluyendo las horas de laboratorio y de aula. Esto representaría unos 4 ECTS desde el punto de vista del alumno. Por otra parte, desde la perspectiva del profesor, es indudable que tras el esquema obtenido hay mucho esfuerzo en planificación y preparación de material, todo ello con junto a más horas que las habituales de contacto con los alumnos, especialmente en tutorías. Sin embargo, pensamos que dicho esfuerzo ha valido la pena y en la situación actual, similar a lo que experimentan los alumnos al alcanzar una rutina sostenible de desarrollo en iteraciones, nosotros los profesores también hemos alcanzado un nivel sostenible de repetición de esta estrategia.

El éxito que hemos conseguido con la infraestructura software de apoyo al trabajo cooperativo nos anima a seguir añadiéndole funcionalidad. En la actualidad estamos desarrollando *webparts* para automatizar el procesamiento de los Registros de Tiempos y para permitir gestionar las evaluaciones del proyecto en los sitios de cada equipo.

## Referencias

- [1] Beck K. *Extreme Programming Explained: Embrace Change*. Addison-Wesley, 1999.
- [2] Frosch D. *Using UML in Software Requirements Analysis Experiences from Practical Student Project Work*. Informing Science + Information Technology Education Joint Conference, InSITE 2003. <http://proceedings.informingscience.org/IS2003Proceedings/docs/032Frosch.pdf>
- [3] Humphrey, W.S. *Introducción al proceso software personal*. Addison-Wesley, 2001.
- [4] Kruchten, P., *The Rational Unified Process: An Introduction*. Addison –Wesley, 2004.
- [5] Letelier P., Canós J.H. *Working with Extreme Programming in a Software Development Laboratory*. Proceedings Fifteenth International Conference on Software Engineering and Knowledge Engineering, SEKE 2003, San Francisco, Julio 2003, pp. 612-615.
- [6] Letelier P., Canós J.H. *An Experiment Comparing RUP and XP*. Springer-Verlag, Lecture Notes in Computer Science Vol. 2675, Proceedings of IV International Conference on Extreme Programming, XP 2003, Génova, Italia, Mayo 2003, pp. 41-46.
- [7] Letelier P. *Una Experiencia en Evaluación Continua Multicriterio Aplicada en un Laboratorio de Desarrollo de Software*. Actas de las VIII Jornadas de Enseñanza Universitaria de la Informática (JENUI 2002), Cáceres del 10 al 12 de Julio 2002, pp. 295-302.
- [8] Letelier P. *Métodos Activos para Mejorar la Enseñanza-Aprendizaje en un Laboratorio de Desarrollo de Software*. Actas del 2do Congreso Internacional de Docencia Universitaria e Innovación, Tarragona, 1, 2 y 3 de Julio 2002. <http://cidui.upc.es/upc/-ice/bbdd/congreso.nsf/web/>, pp. 135.
- [9] Pollice G. *Software development theory and practice in the college curriculum*. The Rational Edge. January 2004. [http://www.therationaledge.com/content/jan\\_04/k\\_theorypractice\\_gp.jsp](http://www.therationaledge.com/content/jan_04/k_theorypractice_gp.jsp).
- [10] Stevens T. *Experiences Teaching Software Engineering for the First Time*. 6<sup>th</sup> Annual Conference on Innovation and Technology in Computer Science Education, ITICSE 2001. <http://www.radford.edu/~kstevens2/iticse2001.pdf>.