

Mejora de la formación en el diseño y desarrollo de software a partir de la coordinación de distintas asignaturas

Maria Ferré, Carlos García-Barroso, Montse García-Famoso, David Sánchez, Aida Valls

Departament d'Enginyeria Informàtica i Matemàtiques

Escola Tècnica Superior d'Enginyeria

Universitat Rovira i Virgili

Tarragona

{maria.ferre, carlos.garciabarroso, montse.garcia, david.sanchez, aida.valls}@urv.cat

Resumen

Esta contribución explica el diseño e implementación de una metodología de formación puesta en marcha en el Grado de Ingeniería Informática de la Universidad Rovira i Virgili desde hace 4 años. El objetivo es mejorar las competencias prácticas de los alumnos en el diseño y desarrollo de software. Para ello se han coordinado y dado continuidad a las prácticas de tres asignaturas obligatorias de diferentes cursos. Se trata de desarrollar un mismo proyecto de software a lo largo de la carrera bajo la perspectiva de tres roles diferentes: diseñador, desarrollador y director. En segundo curso, los estudiantes ejercen el *rol de desarrollador* en grupos de 4 (programación, integración y prueba) bajo la supervisión por un estudiante de cuarto que ejerce el *rol de director* (planifica y distribuye tareas y coordina el equipo). El director también propone el diseño formal del software, que ya ha trabajado en la asignatura de tercero (*rol de diseñador*: análisis de requisitos y diseño). Los resultados muestran una mejora en las capacidades formativas trabajadas en las asignaturas, que se reflejan en una mejora de la calidad de la programación, mejor organización del trabajo y cumplimiento de plazos y una visión mucho más práctica del rol de director.

Abstract

This contribution explains the design and implementation of a teaching methodology applied during 4 years in the Computer Engineering Degree at University Rovira i Virgili. The goal is to improve the students' practical skills on software design and development. With this purpose, we have coordinated the practical projects of three mandatory subjects of different courses. The work consists on developing the same software project in consecutive courses of the degree taking three different roles: designer, developer and director. In the subject of 2nd course,

the students exercise the *developer role* in teams of 4 people (programming, integration and testing) under the supervision of a student of the 4th course with the *director role* (in charge of planning, distributing tasks and coordinating the team). The director follows the formal software design of the project he has developed in a 3rd course subject (*designer role*: requirements analysis and design). The results show an improvement of the skills and competences related to these 3 subjects, which include better programming quality, better team coordination and fulfilment of deadlines, as well as a much practical view of the director's role.

Palabras clave

Proyecto integrador, dirección de proyectos, diseño y desarrollo de software.

1. Introducción

Uno de los perfiles informáticos más demandados por el mercado laboral es el de diseñador y desarrollador de software [1]. El diseño de software es, de hecho, la actividad profesional natural de un ingeniero informático pues requiere de competencias que le son exclusivas a su perfil [4]. En consecuencia, resulta fundamental que los estudios de Grado en Ingeniería Informática formen adecuadamente a los futuros ingenieros en las competencias relacionadas con el diseño y desarrollo de proyectos de software.

Si algo diferencia a los proyectos de software de otros proyectos de ingeniería, es su complejidad [5]. El software, como producto industrial, es singular (lo que hace de su producción un proceso lento) y complejo (lo que impide probarlo en todas sus posibles condiciones de ejecución). La productividad tiende a ser baja y se requiere de una planificación, estimación de costes y gestión del desarrollo muy cuidadosas.

La formación en el diseño y desarrollo de proyectos de software es, en consecuencia, también compleja. Las múltiples perspectivas que se deben tratar (análisis, diseño, gestión, construcción, prueba, ver Sección 2) y competencias que se deben trabajar y evaluar (detalladas en la Sección 3) hacen que sea imposible abarcarlas todas en una sola asignatura. De hecho, las dependencias entre estas competencias hacen difícil cubrirlas en un mismo curso.

Si enfocamos la formación de manera tradicional, a base de trabajos independientes entre asignaturas, corremos el riesgo de ofrecer una visión parcial y simplista de los proyectos de software. Para atacar este problema, en el Grado de Ingeniería Informática de la Universidad Rovira i Virgili hemos diseñado e implementado una metodología de formación integradora basada en coordinar y dar continuidad a las prácticas de tres asignaturas obligatorias de diferentes cursos con contenidos nucleares al desarrollo de software: Programación (2º), Análisis y Diseño de Aplicaciones (3º) y Proyectos y Sistemas Informáticos (4º). A groso modo, se trata de desarrollar un mismo proyecto de software coordinado a lo largo de la carrera bajo la perspectiva de los tres roles principales involucrados en el diseño y desarrollo de software: diseñador, desarrollador y director. El resultado es una simulación integral de un proyecto de software en el que cada alumno acaba ejerciendo todos los roles (a lo largo de diferentes cursos) e interactuando con alumnos de otros roles (director – desarrollador). Esto da a los alumnos una visión práctica y completa del proceso y de sus interacciones.

Comparado con metodologías similares, las novedades de nuestro enfoque son dos: por un lado, cada estudiante acaba desempeñando todos los roles del proyecto, a diferencia de otros trabajos en que los roles quedan repartidos entre diferentes estudiantes [3]; y, por otro lado, cada rol se desempeña en un momento diferente de la formación (distintos cursos), de forma que éstos se pueden ubicar en el momento oportuno en función de las dependencias entre las competencias necesarias para ejercerlos (a diferencia de experiencias como [6, 9, 10] donde el proyecto común se introduce en asignaturas paralelas). Además, a diferencia de otros proyectos integradores desarrollados a lo largo de la carrera, pero de forma independiente en cada asignatura [2], nuestra propuesta simula las interacciones entre roles que ocurrirían en un entorno real (director-desarrollador; desarrollador-desarrollador).

Los resultados de esta metodología, en comparación con una aproximación tradicional basada en prácticas independientes, muestran una mejora en las capacidades formativas trabajadas en las asignaturas, incluyendo una mejora de la calidad de la programación, mejor organización del trabajo y cumplimiento de plazos y una visión mucho más práctica del rol de

director. A nivel de plan de estudios, también ha permitido mejorar la coordinación vertical entre las asignaturas involucradas, evitando solapamientos y lagunas en los contenidos.

El presente trabajo continúa la experiencia para fomentar el trabajo en equipo presentada en [7]. En ese trabajo se explicó la experiencia de crear equipos de programadores dirigidos por estudiantes de 4º y se estudió la mejora respecto al trabajo en pareja y sin director. Sin embargo, solo participaban dos asignaturas y no se cubría el ciclo de vida de un proyecto.

El resto del artículo se ha organizado de la siguiente manera. La Sección 2 describe el ciclo de vida de los proyectos que trabajamos en el Grado y describe los roles principales. La Sección 3 detalla las asignaturas que forman parte de esta experiencia integradora, comentando su contenido y competencias. La Sección 4 describe cómo se ha implementado el proyecto integral entre las diferentes asignaturas y qué trabaja el alumno en cada rol. La Sección 5 detalla los criterios de evaluación y las herramientas usadas para obtener indicadores. En la Sección 6 discutimos los resultados e impacto obtenidos. Finalmente, la Sección 7 presenta las conclusiones.

2. Ciclo de vida del software

Nuestra metodología se basa en las fases del ciclo de vida del software definidas en el modelo *Unified Process* [8] y representadas en la Fig. 1.

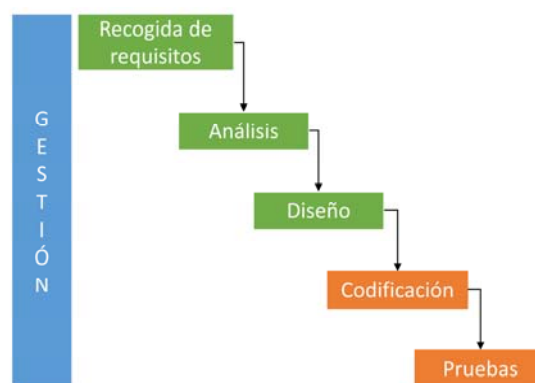


Figura 1: Fases principales del ciclo de vida del software. En azul: fases del director; en verde: fases del diseñador; en naranja: fases del desarrollador.

El proyecto comienza con la recogida de requisitos, en la que el alumno recopilará y documentará los requisitos funcionales y no funcionales de la aplicación a partir de guiones proporcionados por el profesor. Los requisitos funcionales serán sometidos a análisis, tras lo cual se obtendrá una especificación formal (diagramas UML) de estos. Esta especificación, junto con los requisitos no funcionales se utilizarán para diseñar la aplicación considerando la tecnología escogida. El resultado será una especifica-

ción detallada de las clases y módulos de la aplicación, así como del diseño de la interfaz gráfica y de la persistencia de los datos. Esto será la entrada para la fase de codificación, en la que se implementará el diseño propuesto. Finalmente, el software obtenido se someterá a un conjunto de pruebas diseñadas para validar su funcionalidad.

En cuanto a los roles implicados, tal como se muestra en la Fig. 1, las tres primeras fases del ciclo de vida culminan en el diseño de la aplicación y las realiza el *diseñador*. Durante este proceso se emplean técnicas de ingeniería del software y herramientas de diseño (UML), pero no se realiza ninguna codificación. Las dos últimas fases se refieren a la construcción de la aplicación en sí y las realiza un *equipo de desarrolladores* a partir de la especificación del diseño. Se centran exclusivamente en la codificación y pruebas definidas por el diseño, así que sólo requieren de conocimientos técnicos en programación. En paralelo a todas las fases, el *director* realiza un proceso de gestión, consistente en la identificación y asignación de tareas, planificación temporal y monitorización del desarrollo. Esto requiere de conocimientos en planificación y gestión de proyectos y personal.

3. Las asignaturas implicadas

El proyecto integra las prácticas de tres asignaturas: PR (Programación), ADA (Análisis y Diseño de Aplicaciones) y PSI (Proyectos y Sistemas Informáticos). Todas son asignaturas obligatorias del Grado de Ingeniería Informática y pertenecen a la rama de formación en desarrollo de software.

En el Cuadro 1 se dan algunos datos de las asignaturas involucradas en el proyecto integrador que presentamos: el curso y cuatrimestre, el número de créditos de las asignaturas, el número de alumnos promedio de cada asignatura y el peso que tiene este proyecto en la evaluación en cada una de ellas.

Asig.	Curso Cuatri.	ECTS	Peso Proyecto	Alumnos
PR	2º, C1	6	20%	100
ADA	3º, C2	6	50%	55
PSI	4º, C1	6	15%	30

Cuadro 1: Información de las asignaturas involucradas en el proyecto.

El despliegue del proyecto en tres cursos nos permite trabajar diferentes competencias en distintos momentos de la carrera y a varios niveles. En el Cuadro 2 mostramos las competencias comunes en las asignaturas que se trabajan específicamente mediante la metodología propuesta. Estas competencias serían difícilmente tratables a través de trabajos individuales e independientes. Por ejemplo, la competencia CM3 se adquieren cuando el equipo de alum-

nos de PR trabaja bajo la supervisión del director de PSI. En los dos cursos es importante la negociación, comunicación y los hábitos de trabajo, pero el liderazgo se adquiere en PSI. En cambio, las competencias A2, CM1 y CM2 se trabajan en ADA a un primer nivel más teórico y en PSI de forma más práctica.

Id	Competencias	Asig.
A2	Conocimientos para la realización de mediciones, cálculos, valoraciones, tasaciones, peritaciones, estudios, informes, planificación de tareas y otros trabajos análogos de informática.	ADA PSI
CM1	Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente.	ADA PSI
CM2	Capacidad para planificar, concebir, desplegar y dirigir proyectos, servicios y sistemas informáticos en todos los ámbitos, liderando su puesta en marcha y su mejora continua y valorando su impacto económico y social.	ADA PSI
CM3	Capacidad para comprender la importancia de la negociación, los hábitos de trabajo efectivos, el liderazgo y las habilidades de comunicación en todos los entornos de desarrollo de software.	PR PSI

Cuadro 2: Competencias de PR.

4. El proyecto en detalle

Debido a la complejidad del desarrollo de software profesional, éste se suele llevar a cabo en equipos de trabajo. Además, tal como se ha introducido en la Sección 2, todo proyecto medianamente complejo pasa por diferentes fases en las que varios profesionales ejercen diferentes roles. Para que cada alumno pueda tener una experiencia similar a la de un entorno real y adquirir destrezas y competencias asociadas a todos los roles de forma integrada, hemos distribuido las tareas y roles de la siguiente forma (véase Fig. 2):

- En ADA, cada alumno ejerce el *rol de diseñador* y analiza y diseña la aplicación propuesta.
- En PSI, los mismos alumnos ejercen el *rol de director* y utilizan la especificación del diseño que han realizado en ADA para dirigir el trabajo de desarrollo realizado por equipos de PR.
- En PR, equipos de alumnos ejercen el *rol de desarrolladores* y probadores bajo la supervisión de un alumno de PSI.

El objetivo es dar a los alumnos una visión práctica y completa de todo el proceso y de sus interacciones.

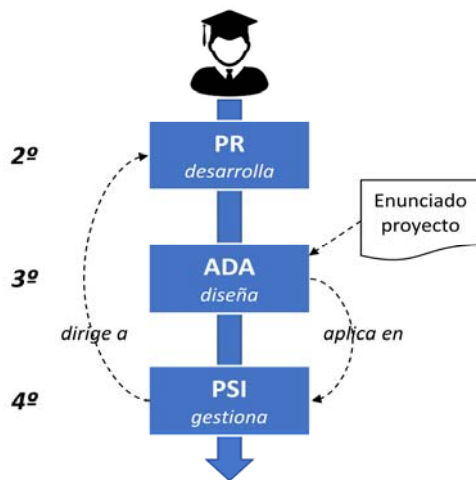


Figura 2: Esquema general del proyecto integrador.

Más en detalle, el proyecto consiste en realizar una aplicación informática de gestión de información con interfaz gráfica y persistencia de datos. Algunos ejemplos de proyectos ya realizados son: una aplicación para gestionar el catálogo de una agencia de viajes que permita a los clientes realizar y gestionar reservas, una aplicación para gestionar peticiones de comida online, permitiendo configurar y preparar menús, o una aplicación para gestionar diferentes tipos de usuarios de una biblioteca, permitiéndoles realizar consultas y/o préstamos de libros.

El proyecto se inicia en ADA, asignatura en la que los alumnos reciben un enunciado consistente en una descripción textual de alto nivel de las funcionalidades del proyecto (guiones) y un conjunto de requisitos no funcionales. A lo largo de varias entregas incrementales durante el 2º cuatrimestre de tercero, los alumnos realizan las fases de recogida de requisitos, análisis y diseño de la aplicación (Fig. 1 color verde). Este proceso se realiza por parejas para fomentar el intercambio de ideas y consensuar puntos de vista. Como resultado, obtiene una especificación formal en UML que describe el diseño de las clases (métodos y atributos), el comportamiento de los casos de uso, el diseño y comportamiento de la interfaz gráfica y el diseño de la persistencia de los datos. A cada entrega, el profesor indica los errores e insta a los alumnos a corregirlos con el fin de no propagarlos a fases posteriores y disponer de una especificación correcta de cara a la dirección que realizarán en cuarto.

Estos mismos estudiantes continúan el trabajo el curso siguiente en PSI, gestionando el proyecto y dirigiendo a un equipo de 4 programadores de PR, que se encargarán de implementar ese diseño (o una parte de éste). Aquí los directores explican el diseño a los alumnos de PR (considerando los conocimientos básicos de modelado de estos), planifican y reparten las tareas (de forma que cada alumno tenga atribucio-

nes y responsabilidades claras y únicas en el proyecto) y coordinan y supervisan al equipo durante la codificación y las pruebas (Fig. 1 color azul). El nivel de desarrollo se adapta a los conocimientos de los alumnos de PR, por ejemplo, sustituyendo la base de datos por ficheros de texto.

Por su parte, los alumnos de PR deben implementar correctamente (de forma eficiente y siguiendo el paradigma orientado a objetos) el diseño propuesto por el director. Aquí no sólo son responsables de la parte del desarrollo que se le ha asignado, sino que también se valora su integración en el equipo y su contribución al éxito del proyecto. Además, deben realizar un juego de pruebas (supervisado por el director) para testear el programa y validar su funcionamiento (Fig.1. color naranja). Esta práctica compartida dura unas 7 semanas y se realiza al final del cuatrimestre (de mediados de octubre a mitad de diciembre), que es cuando los estudiantes de PR ya tienen los conocimientos técnicos necesarios para realizar la codificación. Los estudiantes de PR forman equipos de 4 personas a su elección. Una vez hechos, se les asigna a 1 o 2 directores de PSI a cada equipo y también a un profesor de PR de referencia (éste último encargado de supervisar la interacción y resolver dudas puramente técnicas a los alumnos de PR). También se les facilitan herramientas para trabajar de forma coordinada a través de la plataforma Moodle, concretamente:

- Cada equipo (programadores + director(es)) tiene un *foro privado* que también es visible para el profesor de PR de referencia. En él, el director redacta un acta de cada reunión con su equipo y todos pueden formular y contestar preguntas.
- Cada equipo dispone de un repositorio *GIT* en un servidor de la universidad. Esto les permite realizar la codificación de forma colaborativa y sincronizada. También permite al profesor de referencia una supervisión más directa e individualizada del desarrollo, incluyendo qué ha codificado cada alumno.
- Todos los alumnos disponen de un *modelo de documentación* a entregar junto con la aplicación desarrollada. Aunque se redacta juntamente con el director, está enfocada principalmente a la evaluación en PR.
- Los directores disponen del *calendario de clases* y *contenidos de PR* de forma que puedan planificar el desarrollo en consistencia con los conocimientos de los alumnos de PR.

5. Evaluación de los estudiantes

En este apartado se detallan los criterios de evaluación empleados en cada una de las asignaturas.

Aunque el trabajo en PR se realiza en equipo, su valoración es individual y se tiene en cuenta:

- El código implementado por cada alumno.
- Los juegos de pruebas que validan ese código.
- La documentación realizada por el alumno.
- La entrevista individual del trabajo realizado.
- La participación y colaboración en el equipo.

Esto implica que cada alumno puede tener una nota distinta y permite valorar el trabajo incluso si algún miembro del equipo abandona antes de acabar.

El trabajo en ADA se realiza por parejas, aunque con un reparto claro de tareas (cada alumno se encarga de un subconjunto de casos de uso, aun realizando un diseño del modelo de información común). Por tanto, las notas son individuales. Se realizan dos entregas: la primera abarca los requisitos y el análisis y la segunda el diseño. En cada una reciben comentarios y se les da la posibilidad de corregir errores y mejorar nota. El objetivo es que todos acaben con un diseño válido para ser aplicado en PSI. Los criterios de evaluación son:

- Corrección y detalle del análisis y del diseño.
- Uso apropiado de los diagramas UML y de la metodología de ingeniería del software.
- Respuestas a las preguntas durante las entrevistas realizadas a la entrega de los trabajos.

En la práctica de PSI hay 2 entregas y se valora:

- La planificación previa del trabajo y su revisión posterior a lo largo del desarrollo.
- Organización del equipo y seguimiento.
- Anticipación de incidencias y actuaciones cuando éstas ocurren.

Al finalizar el proyecto se ha encuestado a los alumnos de PR i de PSI para conocer su opinión, detectar posibles problemas de funcionamiento y poder actuar sobre ellos en cursos venideros. La encuesta se ha realizado en la plataforma de campus virtual Moodle utilizando la actividad “Encuesta”. Se ha realizado de forma anónima y por tanto voluntaria para los estudiantes. Al ser voluntaria no todos los alumnos la van a responder, pero permite que sea anónima y que las respuestas sean más sinceras.

5.1. Encuestas a alumnos de PR

Se ha realizado una encuesta a los estudiantes de PR con 12 preguntas sobre la experiencia del proyecto realizado. La encuesta ha sido diseñada para que sea rápida de responder y solo 2 de las preguntas son de respuesta abierta. En éstas se pide al alumno reflexionar sobre su aprendizaje y experiencia obtenida más allá de los conceptos técnicos aprendidos, así como posibles mejoras de cara a una nueva edición.

Nueve preguntas son de respuesta única a elegir en la mayoría de los casos entre cuatro opciones; *mucho*, *bastante*, *poco* o *nada*. Estas preguntas se refieren a los plazos, las herramientas utilizadas (GIT para control de versiones del código, fórum para la comunicación entre el equipo), la actitud en el equipo (compromiso, cumplimiento plazos) o la satisfacción con los compañeros de equipo y con el trabajo del director, entre otras. Una última cuestión de respuesta múltiple les pregunta por los aspectos que consideran que han realizado mejor gracias a tener un director que les ofrece una guía. Entre las opciones a escoger se encuentran la facilidad de coordinación en el equipo, llevar el trabajo al día o la implementación.

5.2. Encuestas alumnos de PSI

De forma similar se ha realizado una encuesta para los alumnos de PSI (y, por ende, también de ADA). La encuesta no solo recoge la opinión del trabajo realizado durante el proceso de dirección a los alumnos de PR, sino también la percepción de la experiencia de realizar las prácticas de las tres asignaturas de forma integrada. El formato de las preguntas es similar a las de PR.

Sobre la dirección del proyecto se pide valorar su experiencia como directores, el cumplimiento de los estudiantes de segundo, tanto en tiempo como en resultados, o la comunicación en el equipo. También se les pregunta si han podido realizar correctamente el trabajo de dirección, validar el trabajo realizado por los alumnos de segundo, realizar las actas de seguimiento, y se pide que reflexionen sobre las dificultades que se han encontrado.

Unas preguntas que consideramos muy importantes son las que se refieren al nivel de satisfacción respecto a la experiencia de integrar las prácticas de las tres asignaturas. En este caso se pide al alumno que valore la ventaja de utilizar el enunciado que diseñó en ADA como punto de partida para el proyecto que debe dirigir como estudiante de PSI a los alumnos de PR. También se le pide su opinión sobre si la integración de las prácticas le permite tener una visión más clara del ciclo de vida del desarrollo de software.

6. Resultados e impacto obtenidos

En este apartado se detallan los resultados académicos obtenidos en PR, comparándolos con los de cursos previos en los que no se aplicó la metodología propuesta. También se incluyen los resultados medios de las encuestas realizadas a los alumnos de PR y PSI. Finalmente se analiza el impacto de la propuesta.

6.1. Resultados académicos de PR

Los datos a comparar son los resultados de la práctica de PR realizada en equipo y dirigida por estudiantes de PSI. Se han considerado los resultados obtenidos en dos cursos previos a la implantación de la experiencia y los de los cursos en que se ha trabajado de esta forma. Para tener una visión más global se ha calculado la media de los resultados de los distintos cursos y se indica el valor en % para facilitar la interpretación. Se ha utilizado la media en lugar de los resultados por cursos para ver la tendencia de los resultados y eliminar en lo posible las particularidades propias de un curso. En la Fig. 3 se representa la información y se observa que disminuye el porcentaje de alumnos que no superaron (o entregaron) la práctica. El hecho de que un alumno de cuarto curso dirija y gestione al equipo favorece que los alumnos de segundo empiecen a trabajar antes y de forma más constante, y previene que algunos alumnos “tiren la toalla”. Se detecta que las notas han mejorado, hay más alumnos que superan la asignatura y con mejores notas. Estos resultados se obtienen gracias a partir de un diseño y reparto de tareas claro, una buena planificación y al seguimiento continuo del trabajo.

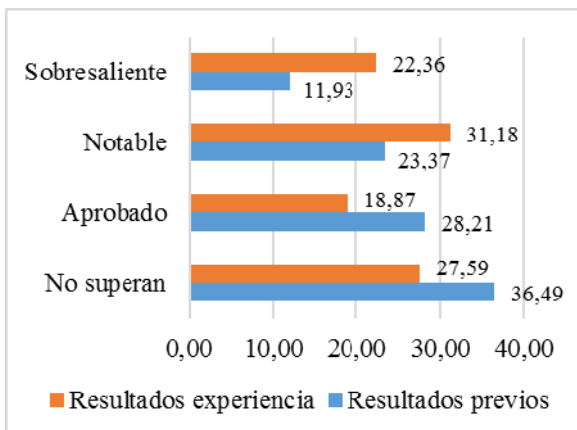


Figura 3: Representación de la evolución de notas.

6.2. Resultados de las encuestas de PR

A la hora de valorar la metodología, no sólo son importantes los resultados académicos, sino también la percepción de los estudiantes que participan en la experiencia. Con este propósito se han realizado las encuestas comentadas en la Sección 5 y a continuación recopilamos y comentamos los resultados.

Se han recogido 52 encuestas. Analizando individualmente cada pregunta, el porcentaje de valoraciones positivas (*mucho, bastante*) siempre es superior a las negativas (*poco, nada*), excepto al valorar la utilidad de las actas de las reuniones y el uso de un repositorio de GIT para compartir el código. En el caso de GIT, se han detectado algunos problemas en

su uso y se mejorará la documentación de ayuda de cara al próximo curso. La Fig. 4 muestra los resultados de las 5 preguntas más relacionadas a la experiencia relativa a este proyecto integrador.

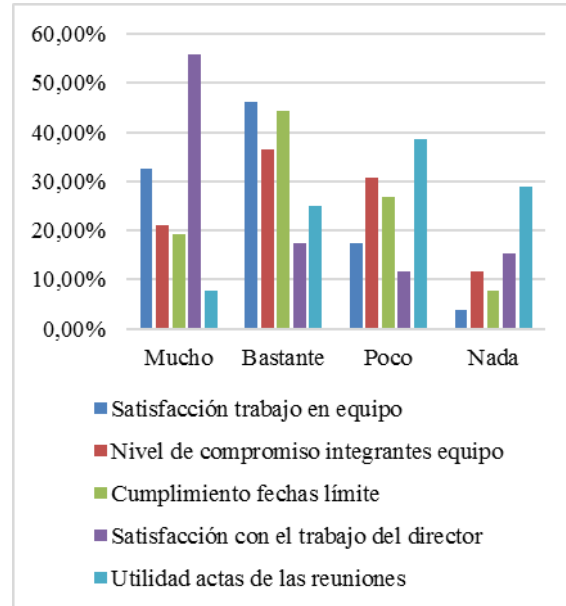


Figura 4: Resultados de la encuesta en lo referente a la satisfacción del trabajo en el equipo de PR.

Destacamos el alto grado de satisfacción y también la buena opinión sobre el director. No obstante, se pone de manifiesto que casi la mitad de los alumnos consideran que sus compañeros no estaban comprometidos de forma suficiente con el trabajo a realizar.



Figura 5: Opinión del modelo de trabajo en equipo.

También se ha preguntado sobre los aspectos que creen haber mejorado gracias a la práctica en equipo dirigida. Las respuestas más frecuentes se refieren a la coordinación del equipo, al cumplimiento de las fechas límite y al evitar confusiones en el desarrollo de la práctica. Finalmente se ha preguntado si se repetiría el modelo de trabajo y se han ofrecido alternativas similares. Los resultados se visualizan

en la Fig. 5, donde podemos ver que la mayoría repetirían el mismo modelo de práctica.

En cuanto a las preguntas de respuesta abierta, algunas de las reflexiones positivas van desde el aprendizaje de la responsabilidad compartida, a la sensación de aprender a planificar para encarar un proyecto y a la motivación que surge del sentir que su trabajo influye en el del conjunto del equipo y el éxito del proyecto. El número de reflexiones positivas supera ampliamente al de negativas.

Las críticas negativas se centran en equipos donde la comunicación no ha funcionado y se han generado problemas entre los alumnos de cuarto y de segundo. Específicamente, una minoría ha considerado que las indicaciones del director les han perjudicado.

Un análisis más detallado permite relacionar las respuestas de las distintas preguntas de forma que se puede analizar las respuestas de los alumnos en algunas preguntas cuando han respondido con opiniones negativas (o positivas) en otras. En general, los alumnos que consideran que el director no ha realizado su trabajo, también indican que no les ha gustado tener un director. También están relacionados los resultados de la pregunta relativa al cumplimiento de las fechas límite con los de la pregunta sobre si los miembros del equipo estaban comprometidos por igual. Con esto se detecta que las valoraciones negativas no se deben a un único factor, sino que son debidas a la suma de problemas intrínsecos al equipo.

6.3. Resultados encuesta alumnos PSI

Se han obtenido 17 respuestas por parte de los alumnos de PSI. Analizando individualmente cada pregunta, el porcentaje de valoraciones positivas siempre es superior al de negativas en todas las preguntas excepto en la referente a la utilidad de realizar un acta de las reuniones de coordinación. En la Fig. 6 se muestran los resultados de las tres preguntas relacionadas con la satisfacción del proceso de dirección.

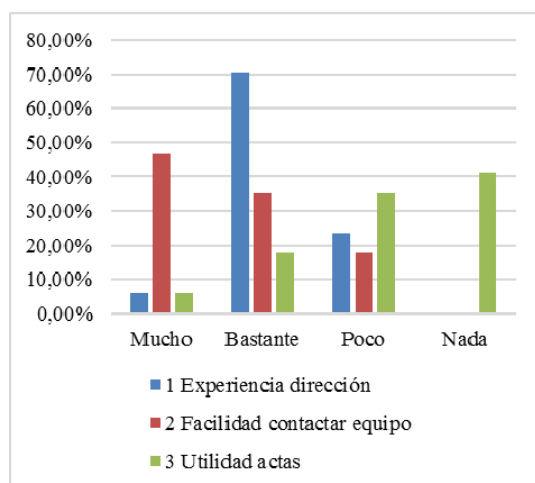


Figura 6: Resultados de la encuesta en lo referente a la satisfacción en la dirección del equipo.

En cuanto a las respuestas a preguntas abiertas, ningún alumno justifica la no utilidad de las actas. En cualquier caso, vistos los resultados se revisarán las instrucciones para detallar y simplificar su uso.

Los mayores problemas que se indican en las respuestas abiertas son referentes a la no implicación de algunos alumnos del equipo. En la Fig. 7 se muestran las respuestas sobre cuáles han sido las dificultades que se ha encontrado a la hora de dirigir. Era una pregunta de respuesta múltiple. Se detecta que los mayores problemas están en la supervisión de la documentación. En general, los directores han indicado que los alumnos la prepararon demasiado tarde.

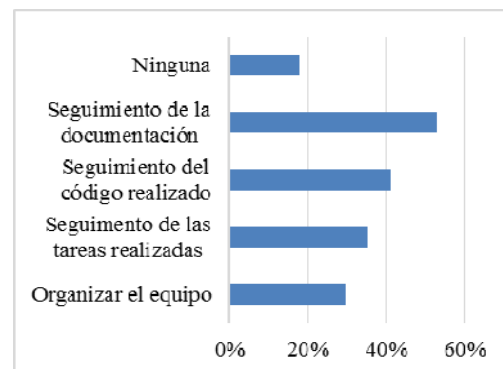


Figura 7: Dificultades en la dirección.

Los resultados a las preguntas para valorar la satisfacción de la experiencia de integrar las prácticas se muestran en la Fig. 8.

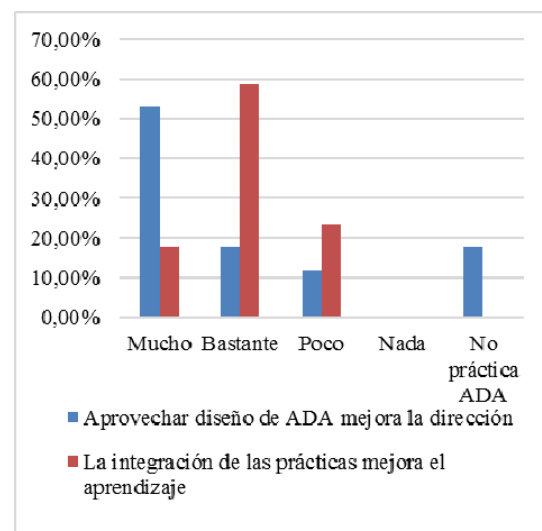


Figura 8: Satisfacción en la integración de prácticas.

Una pregunta se refería a si el hecho de aprovechar el diseño de la práctica realizado en ADA ha ayudado en la dirección del proyecto. Otra preguntaba si se consideraba que la integración de las prácticas de las tres asignaturas mejoraba el entendimiento del ciclo de vida del software. Vemos que en las dos preguntas

los alumnos están ampliamente satisfechos y valoran positivamente la metodología. Hay un subconjunto de alumnos que han cursado ADA en un curso académico distinto al previsto en el itinerario de estudios y por tanto, no conocen el tema del proyecto. Es lo que refleja la columna “No práctica ADA” y que corresponde a sólo 3 alumnos.

6.4. Impacto

Vistos los resultados, tanto académicos como de opinión, consideramos que el proyecto integrador ha tenido un impacto muy positivo en la formación de los estudiantes de ingeniería informática. Las encuestas indican que la experiencia adquirida a lo largo de las diferentes asignaturas y roles ha favorecido el aprendizaje de valores personales muy importantes específicos a la profesión, y también transversales. A menos es la impresión que indican los estudiantes con sus respuestas. Coinciden que han mejorado en:

- Ser consciente del trabajo que se desempeña en distintos niveles del proceso.
- La importancia de un buen diseño, una buena planificación y un seguimiento y gestión de incidencias adecuado.
- Aprender a trabajar en equipo con personas de diferentes caracteres y niveles, ya sea como director o como programador.
- La importancia de comprometerse con el equipo y cumplir las tareas en el plazo estipulado.

La experiencia también ha tenido un impacto positivo a nivel académico, que se puede observar en el éxito y en las calificaciones obtenidas en PR.

7. Conclusiones

Tras la experiencia de implementar el proyecto integrador durante 4 cursos, y a la vista de los resultados presentados en este artículo, los profesores consideramos el modelo propuesto como muy positivo.

Como recomendaciones generales recopiladas a lo largo de los años de implantación, queremos destacar la necesidad de proporcionar instrucciones claras sobre los roles de cada alumno en cada asignatura. También, durante la fase de dirección-desarrollo es importante que tengan un profesor de PR de referencia para que ayude a resolver las incidencias relativas al comportamiento de los alumnos en el equipo.

Finalmente, mencionar que hemos observado beneficios colaterales derivados de la experiencia. En particular, los alumnos de segundo afirman que conocer a estudiantes de otros cursos ha sido positivo, y les ha ayudado a ampliar su visión sobre los estudios y la relación entre las asignaturas. Por otro lado, los hemos detectado que el proyecto ayuda a integrar en la clase a alumnos a lo que les cuesta relacionarse.

Referencias

- [1] Adecco. Empleos tecnológicos en el mercado laboral español. Abril de 2016. Disponible en: <https://adecco.es/wp-content/uploads/notas-de-prensa/764.pdf>
- [2] Rafael Asenjo, Sonia González, Francisco Corbera, Ángeles Navarro, Andrés Rodríguez, Julio Villalba y Elgius Hendrix. La plataforma Raspberry Pi como base para la coordinación vertical. *Enseñanza y Aprendizaje de Ingeniería de Computadores*, 7, 5-20, 2017.
- [3] Isidro Calvo, José Manuel López-Guede y Ekaitz Zulueta. Aplicando la metodología Project Based Learning en la docencia de Ingeniería Técnica en Informática de Gestión. *Revista de formación e Innovación Educativa Universtia-ria*, 3(4), 166-181, 2010.
- [4] Ricardo Colomo-Palacios, Edmundo Tovar-Caro, Ángel García-Crespo y Juan Miguel Gómez-Berbis. Identifying Technical Competences of IT Professionals: The Case of Software Engineers. *International Journal of Human Capital and Information Technology Professionals* 1(1), 1-13, 2010.
- [5] Brian Fitzgerald. Software crisis 2.0. *IEEE Computer* 45(4), 89-97, 2012.
- [6] Félix O. García, David G. Rosado, M^a Ángeles Moraga, Manuel A. Serrano. Formación integral en la intensificación de Ingeniería del Software en el grado en Ingeniería Informática. En Actas de las XXIV Jornadas de Enseñanza Universitaria de Informática, Jenui 2018, pp. 197-204, Barcelona, julio 2019.
- [7] Montse García-Famoso, Maria Ferré, Aida Valls y Carlos García-Barroso. El trabajo de equipo en acción. En Actas de las XX Jornadas de Enseñanza Universitaria de Informática, Jenui 2014, pp. 245 – 252, Oviedo, julio 2014.
- [8] Ivar Jacobson, Grady Booch y James Rumbaugh. The unified software development process. Addison-Wesley Longman Publishing Co., Boston, USA, 1999.
- [9] Jenny Ruiz, Leydis Lamoth Borrero, María Rita Concepción García y Félix Rodríguez Expósito. El proyecto integrador como experiencia didáctica en la formación del ingeniero informático: Universidad de Holguín, Cuba (UHOLM). *Escenarios*, 10(1), 106-115, 2012.
- [10] Pablo Sánchez, Carlos Blanco, Alejandro Pérez, Julio Medina, Patricia López, Alfonso de la Vega, Diego García y Miguel Sierra. Experiencia y Lecciones Aprendidas durante el Desarrollo de un Proyecto Software Común a Diversas Asignaturas. En Actas de las XXIII Jornadas de Enseñanza Universitaria de Informática, Jenui 2017, pp. 291-298, Cáceres, julio 2017.