

Formación integral en la intensificación de Ingeniería del Software en el grado en Ingeniería Informática

Félix O. García, David G. Rosado, M^a Ángeles Moraga, Manuel A. Serrano
Departamento de Tecnologías y Sistemas de la Información
Escuela Superior de Informática de Ciudad Real
Universidad de Castilla-La Mancha
{Felix.Garcia, David.GRosado, MariaAngeles.Moraga, Manuel.Serrano}@uclm.es

Resumen

Tradicionalmente, los estudiantes de Ingeniería del Software adquieren las competencias de cada asignatura mediante trabajos no reales que están acotados en complejidad y alcance. En este artículo se presenta una propuesta de formación con un enfoque transversal, uniendo varias asignaturas para plantear casos de desarrollo de software más complejos y realistas. Como primera aproximación se ha realizado una experiencia que engloba 3 asignaturas de la intensificación, abordando tanto la perspectiva técnica como la gestión en un proyecto común. Se abordan las principales restricciones como la temporalidad de las asignaturas, matriculación de los estudiantes, sistema de evaluación homogéneo, formación de equipos, calendario, revisiones conjuntas con los estudiantes, etc. Además, se presentan unas primeras evidencias de la viabilidad de la propuesta, mostrando los resultados de su aplicación en el curso 2017/2018 así como la opinión de los estudiantes matriculados, demostrando una buena aceptación.

Abstract

Traditionally, Software Engineering students have acquired competencies basing their learning in not real projects due to constrains in effort and time are limited in complexity and extent. In this paper, we propose a transversal approach for Software Engineering Education integrating related subjects and with special emphasis in proposing more complex and realistic assignments. As a starting approach we have carried out a experience of transversal education encompassing three Software Engineering subjects, forcing the students to work in a complex common project from both technical and management perspectives. In this approach, we have faced new challenges and restrictions, such as time ordering of the subjects, students' enrollment, homogeneous evaluation system, team creation and management, common sched-

ule, common reviews, and so on. Also, here we present the first evidences of proposal feasibility, showing the results of the project application during the 2017/2018 course along with the students' opinion, which demonstrated a good acceptance.

Palabras clave

Ingeniería del Software, Formación integral, Coordinación, Aprendizaje basado en proyectos.

1. Introducción

El título de Grado en Ingeniería Informática que se oferta en la Universidad de Castilla-La Mancha¹ se diseñó como un único grado con cuatro intensificaciones y un catálogo de optativas, el cual cumple con el acuerdo del Consejo de Universidades sobre recomendaciones para los títulos oficiales del ámbito de la Ingeniería Técnica Informática². En concreto, existe un módulo de 60 ECTS de formación común para la rama de Ingeniería Informática, más 36 ECTS adicionales para reforzar los 60 anteriores. Se diferencian cuatro módulos de tecnología específica de 48 ECTS, correspondientes a las intensificaciones: computación, ingeniería de computadores, ingeniería del software y tecnologías de la información. Dentro de este mapa del grado, nuestro objetivo es centrarnos en la intensificación de Ingeniería del Software (ISo). Esta intensificación tiene como objetivo principal profundizar en los contenidos básicos que el estudiante ha aprendido en la formación común, en lo relativo a las distintas etapas del ciclo de vida (requisitos, diseño, construcción, pruebas, mantenimiento) y gestión de proyectos informáticos. También se tratan en profundidad la calidad y la seguridad, así como el desarrollo de bases de datos y los sistemas de información empresariales avanzados.

¹ <http://webpub.esi.uclm.es/spa/paginas/formacion-grado-en-ing-informatica>

² Secretaría General de Universidades. Boletín Oficial de Estado, núm. 244, 9 de octubre de 2009, páginas 85345 a 85351.

Tradicionalmente, entre las metodologías utilizadas en la impartición de las asignaturas de la intensificación de ISO, destaca el aprendizaje basado en proyectos [3] y en problemas [9]. Se ha demostrado que una adecuada aplicación de estas metodologías permite conseguir buenos resultados, de modo que los estudiantes adquieren tanto los conocimientos (saber) como las destrezas requeridas en cada una de las materias (saber hacer). No obstante, en nuestra experiencia, nos encontrábamos con dos problemas principales a la hora de aplicarlas de forma individual en cada asignatura. En primer lugar, tanto los proyectos como los problemas que se abordaban tenían una complejidad limitada, respetando el número de horas que el alumno debe dedicar a su resolución y que los conocimientos impartidos en una asignatura es necesario aplicarlos en otra, por lo que vimos necesario realizar una reestructuración en la impartición de las asignaturas.

De esta forma, abordamos aspectos clave en la formación del Ingeniero Software de forma transversal, evitando redundancia de contenidos y uniendo varias asignaturas para plantear casos de desarrollo de software más complejos y realistas. Además, es importante formar a los Ingenieros Software de forma más cercana al tipo de problemas que van a tener que resolver en la industria, mostrándoles las mejores prácticas y desarrollando sus capacidades para resolver problemas, adaptarse, tener mente crítica, etc. De hecho, tal como sugiere Limoncelli en [6], los currículos de Tecnologías de la Información e Informática se deberían estructurar para ser más inmersivos, de modo que se refleje de mejor manera el mundo real. Así, los estudiantes se preparan para el mundo empresarial y “entienden por qué están aprendiendo algo, incluso antes de que lo aprendan”, abordando la problemática en primer lugar.

En [2, 4, 5, 8, 10, 11] se pueden encontrar propuestas de integración de asignaturas en base a la realización de un proyecto común aplicadas a todo el grado o a itinerarios específicos como Ingeniería del Software o Tecnologías de la Información. Todas estas iniciativas demuestran con experiencias reales y resultados de éxito, las mejoras en el aprendizaje y en la motivación de estudiantes y profesores de este tipo de enfoques. Partiendo de esta base, en esta experiencia docente adoptamos una estrategia general similar, vertebrada en torno a un proyecto común, pero adaptándonos a la problemática particular de nuestra intensificación, estableciendo un contexto con una importante carga de gestión ágil, procesos de soporte y tecnologías de desarrollo continuo de software, así como en los mecanismos específicos de coordinación de las distintas asignaturas con evaluación homogénea y con un calendario consensuado de todas las asignaturas. En este contexto afrontamos nuevos retos y restricciones que pueden limitar la aplicación de la

propuesta como la temporalidad de las asignaturas. Finalmente, para proporcionar unas primeras evidencias de la viabilidad de la propuesta, se ha realizado un proyecto piloto integrando tres asignaturas (Procesos de Ingeniería del Software, Gestión de Proyectos Software y Seguridad de Sistemas Software).

El resto del presente artículo se estructura de la siguiente manera: en la sección 2 se presenta la propuesta, propiamente dicha, para impartir las asignaturas de la intensificación de ISO. La sección 3 presenta una pequeña validación preliminar mientras que las limitaciones se exponen en la sección 4. Finalmente, la sección 5 presenta las principales conclusiones y el trabajo futuro.

2. Formación integral: propuesta

Con el objetivo de generar una formación integral en Ingeniería del Software se seleccionaron tres asignaturas para un primer proyecto piloto: Procesos de Ingeniería del Software (PrIS), Gestión de Proyectos Software (GPS) y Seguridad de Sistemas Software (SSS). Las asignaturas fueron seleccionadas teniendo en cuenta limitaciones temporales, de composición de los grupos de alumnos y cohesión temática. El reto principal era establecer un contexto adecuado para plantear un proyecto más realista de Ingeniería del Software considerando la suma de esfuerzo de las asignaturas seleccionadas, debiéndose abordar las dos perspectivas, técnica y de gestión. Por ello, en primer lugar, era necesario incluir las asignaturas de procesos (aborda la perspectiva técnica) y la de gestión de proyectos, y se incluye una asignatura que aborda un proceso crítico como es la seguridad. Como resultado se obtiene la propuesta de integración con los elementos mínimos necesarios (técnico + gestión + proceso de soporte) que facilita un crecimiento futuro, considerando nuevas perspectivas técnicas (requisitos, diseño, desarrollo de bases de datos, sistemas de información empresariales) o de procesos auxiliares (calidad). Las principales actividades realizadas y los criterios de evaluación estaban orientadas a cohesionar la realización del caso práctico global y compartido por las 3 asignaturas.

2.1. Visión general

Cuando se planteó el proyecto piloto, se detectaron algunas posibilidades de mejora, que se resumen a continuación:

- En la asignatura de Procesos de Ingeniería del Software (PrIS) se deben realizar proyectos de desarrollo software para que los estudiantes adquieran las habilidades de desarrollo de software aplicando gestión de la configuración y desarrollo dirigido por pruebas, entre otros aspectos. Pero también se utilizaba Scrum para la gestión de los proyectos, tema que era abordado con mayor

profundidad en la asignatura de GPS, existiendo duplicidad de contenidos.

- La asignatura de Gestión de Proyectos software (GPS) trata entre otros temas fundamentales la formación en gestión ágil y complementa los conocimientos adquiridos por los estudiantes en gestión de proyectos (siguiendo PMBOK [7]) abordando las particularidades de la gestión de proyectos software y áreas clave no tratadas en el plan de estudios como la gestión de riesgos. El caso práctico debía ser la gestión de un proyecto de “juguete”, ya que los equipos de trabajo no podían realizar un desarrollo de software completo, al abordarse exclusivamente la parte de gestión.
- La imposibilidad de proponer proyectos de desarrollo software más realistas es una problemática presente en el resto de las asignaturas de la intensificación, derivando en proponer proyectos de juguete o prácticas demasiado específicas, como las realizadas en la asignatura de SSS.

Ante este escenario, se propone la realización de un caso práctico en equipo, compartido por las tres asignaturas seleccionadas y que permita abordar sus competencias. Para que la propuesta se pueda materializar, es necesario que se cumplan una serie de condiciones fundamentales:

- La matriculación del alumnado en las tres asignaturas.
- Homogeneizar las guías docentes de las asignaturas en lo que respecta a sus actividades y criterios de evaluación, en consonancia con la memoria del título de grado.

En cuanto a la evaluación, la realización del caso práctico se realiza en equipo y supone un 30% de la nota. La parte grupal de la nota se completa con la realización de los trabajos de teoría (20% de la nota), y el desempeño individual de cada estudiante se evalúa mediante la actividad de prueba final y su nota sobre presentaciones o su participación en las clases magistrales (50% restante).

Otro aspecto clave es la temporalidad con la que se imparten los contenidos, puesto que el caso práctico determina el orden en el que los estudiantes deben aprender los contenidos teóricos y prácticos. Por ejemplo, el conocimiento del entorno tecnológico a utilizar (asignatura PrIS: herramientas de gestión de configuración, integración continua, pruebas, etc.) así como los conocimientos principales de gestión (asignatura GPS: Scrum) deben impartirse en primer lugar, antes de poder empezar a trabajar con el proyecto práctico.

A continuación, se presentan los principales elementos que conforman la propuesta.

2.2. Trabajo en equipo

El caso práctico de la asignatura se realiza en equipos de 6 estudiantes (excepcionalmente, 7). Es un tamaño de equipo mayor que los que habitualmente se aplican en otras asignaturas (4-5), con el fin de tener un contexto lo más realista posible y adaptado a la complejidad del caso y a la metodología de gestión (Scrum). Para la formación de los equipos se da libertad a los estudiantes, aunque los profesores son los responsables de dar el visto bueno a la composición final de los equipos.

Un aspecto fundamental es el seguimiento de los equipos de trabajo, realizando las reuniones que impone la metodología Scrum (planificación y revisión de Sprint). Los profesores responsables participan también en las reuniones de retrospectiva, aunque son los miembros de los equipos los que deben mejorar de forma continua su proceso de trabajo. Además, los equipos de trabajo pueden hacer uso del horario de tutorías para resolver aspectos más técnicos o de gestión relacionados con cada asignatura. La filosofía fundamental que se sigue es la misma de Scrum (funcionamiento auto-organizado y equipos multifuncionales). No obstante, al tratarse de su primera experiencia de aplicación de Scrum realista, los profesores proporcionan una mayor guía a los equipos y se dedica un esfuerzo adicional en las reuniones. A partir del tercer Sprint los equipos ya deben demostrar su capacidad de funcionar de forma prácticamente autónoma.

2.3. Caso práctico

El caso práctico constituye el núcleo de la transversalidad al aplicarse habilidades de las tres asignaturas para realizarse satisfactoriamente. El objetivo principal del proyecto es desarrollar una aplicación web que modele una red social interna a la empresa con unas funcionalidades similares a Facebook. La práctica debe crearse utilizando Java y Spring Framework y debe utilizar buenas prácticas de gestión e Ingeniería del Software:

- Gestión del proyecto usando *Scrum*.
- Planificación de tareas con paneles (*Kanban*).
- Gestión de la configuración y control de versiones con *git* y *github*.
- Utilización de patrones de IS.
- Behavior-Driven Development (BDD) y Test-Driven Development (TDD).

El caso práctico se plantea de forma general, para establecer el contexto y a partir de ahí los estudiantes, aplicando la metodología de gestión ágil Scrum, asisten a reuniones periódicas con los propietarios del producto (profesores) que les van añadiendo detalles a dichos requisitos.

Por ejemplo, en la reunión de planificación del primer Sprint, los equipos de trabajo y el propietario del producto (profesor) van elaborando historias de usuario a partir de la descripción general proporcionada, como las historias de usuario específicas para hacer una publicación en la red social. Destacamos que, aunque es responsabilidad del propietario del producto la definición de las historias de usuario, para facilitar un mejor aprendizaje, se deja que los propios equipos de trabajo propongan historias de usuario y el propietario del producto las revisa y acepta o propone aquellas que considera que los estudiantes no han identificado.

Además, para la aplicación de los conocimientos y habilidades de la asignatura de seguridad (SSS) se añaden requisitos específicos como, entre otros: sistema de control de acceso RBAC, acceso de usuario mediante usuario y contraseña (políticas de contraseña), sistema de acceso con Captcha para evitar ataques o peticiones malintencionadas, acceso con certificados o DNI electrónico, web segura (https), claves guardadas/datos sensibles cifrados, política de copias de seguridad, incorporación de medidas de seguridad para evitar ataques, como la inyección SQL.

Finalmente, para una adecuada realización del caso práctico, es necesario aplicar un método de trabajo completo con técnicas de gestión y desarrollo, y un entorno tecnológico que facilite su labor, como se presenta en el siguiente apartado.

2.4. Método de trabajo y entorno tecnológico

A la hora de establecer el método de desarrollo y gestión para llevar a cabo el caso práctico se han considerado los objetivos formativos de las asignaturas que forman parte de la integración, siendo necesario por ello incluir: gestión ágil y visual, equipos auto-organizados, desarrollo dirigido por comportamiento (BDD) y desarrollo dirigido por pruebas (TDD), y gestión de la configuración.

El método de trabajo a seguir por los equipos está basado en un método ágil de gestión que combina Scrum y Kanban, donde se diferencian los siguientes roles:

- Propietario del producto: profesor de PrIS.
- Scrum Master: profesor de GPS.
- Experto en Seguridad (asesor equipo): profesor de SSS.
- Equipo: estudiantes.

A partir de la aplicación de Scrum, el equipo debe ir desarrollando y probando en 4 Sprints los requisitos que establece el propietario del producto aplicando un enfoque BDD, incluyendo TDD. A la hora de realizar el proceso de construcción del software se sigue el

enfoque de integración continua. Además, se requiere disponer de paneles para la visualización de las historias de usuario, tareas y su estado, etc. Para ello, se utilizan unos tableros que hay en el aula (Figura 1). Toda esta información está siempre actualizada y a la vista de todo el equipo, permitiendo a los profesores ir observando el progreso y detectar si existe algún impedimento. Además, se usan tableros Kanban para el seguimiento de las tareas, tableros de estrella de mar para la realización de las retrospectivas y tableros *Niko-Niko* para indicar el estado de ánimo de los estudiantes.

Adicionalmente a los paneles del aula, los equipos deben introducir toda la información anterior usando herramientas de gestión específicas. El método de trabajo propuesto proporciona, por lo tanto, un soporte bastante amplio a paradigmas como DevOps [1], sobre todo fomentando agilidad y alto grado de automatización, aunque al tratarse de un entorno académico no se da soporte a la etapa de operaciones.



Figura 1. Panel Kanban del aula

No obstante, para que este método de trabajo se pueda llevar a cabo adecuada y eficientemente, se debe contar con un alto grado de soporte automatizado y para ello se debe proponer un entorno tecnológico que incluya las herramientas necesarias (Figura 2).

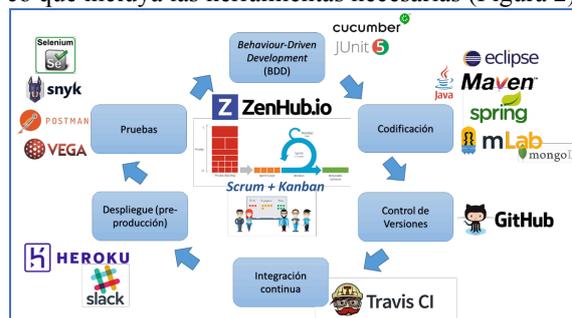


Figura 2: Entorno Tecnológico.

Tal como se muestra en la Figura 2, los aspectos de gestión se cubren mediante el uso de *ZenHub*, que permite realizar la gestión ágil de proyectos y que permite la administración de peticiones (*issues*) propias de la gestión del proyecto y la integración con *GitHub*, que es la herramienta de soporte a la gestión de configuración. Por su parte, se hace uso de *Cu-*

cumber a la hora de estudiar el dominio del problema y especificar el comportamiento asociado a los requisitos. A partir de estas especificaciones se van realizando las pruebas de forma automatizada siguiendo un enfoque *TDD* y usando *JUnit* como librería de soporte. La construcción se realiza en *Java*, en un entorno *Eclipse* y haciendo uso del *framework Spring*, que permite implementar el sistema con el patrón Modelo-Vista-Controlador (MVC). Mediante *Maven* se realiza la construcción (*build*) del proyecto usando *TravisCI* para la Integración Continua, que permite el despliegue automático a la plataforma *Heroku* y automatiza pruebas con herramientas como *Selenium* o *Snyk*.

En cuanto a la curva de aprendizaje del estudiante a la hora de asimilar las distintas tecnologías del entorno tecnológico, se considera que, al menos, todo estudiante ha trabajado con *Java*, *Eclipse*, *JUnit* y *GitHub*. No obstante, en la asignatura PrIS se explican dichas tecnologías en las primeras semanas del curso. El modo de trabajo consiste en asignar un trabajo teórico a equipos de 2 ó 3 personas para analizar y presentar a los compañeros de clase una de las tecnologías del entorno.

Además del entorno general, en las asignaturas de GPS y SSS se utilizan herramientas específicas de soporte a gestión de proyectos y seguridad como, por ejemplo: COCOMO II USC (para estimaciones software), ASiRis 2.0 (herramienta desarrollada internamente para gestión de riesgos), *OpenSSL* y el conjunto de herramientas proporcionado por el entorno *KaliLinux* (para SSS).

2.5. Actividades y calendario

Para la puesta en marcha de la propuesta explicada anteriormente, en primer lugar, se realiza una presentación general de la formación integral, para explicar la transversalidad y el caso práctico, así como de la arquitectura tecnológica necesaria. Se presenta el calendario general y se explican los criterios de evaluación.

A continuación, cada asignatura hace su presentación específica, explicando sus contenidos, competencias y resultados de aprendizaje, así como su forma de evaluación.

El aspecto central del calendario es el hecho de que la organización temporal de las asignaturas está marcada por el seguimiento de Scrum y por ello el calendario se organiza en torno a dicha planificación. Se definen 4 Sprints, los tres primeros centrados en el desarrollo del caso práctico y el cuarto centrado en una iteración de mantenimiento de software. La aplicación de Scrum implica la celebración de las siguientes reuniones:

- *Reunión de planificación del Sprint* (40 min-1h). Cada equipo se reúne con el propietario del producto y Scrum Master para definir/refinar historias de usuario y planificar el trabajo a realizar en dicho Sprint, seleccionando las historias de usuario acordadas con el propietario del producto y definiendo las tareas a hacer para satisfacer las historias, que van colocando en sus tableros Kanban. En el calendario se destinan cuarenta minutos para esta reunión, aunque en el primer Sprint la duración es de una hora, dado que se debe realizar un mayor esfuerzo en la definición de las historias. Los equipos de trabajo deben negociar con el propietario del producto a la hora de definir las historias y el Scrum Master se asegura de que los estudiantes están aplicando convenientemente Scrum actuando, si es necesario, para asegurar una aplicación correcta. En la primera reunión de planificación se parte de una pila de producto vacía y a partir de la descripción general del caso práctico se negocian las historias de usuario concretas a incorporar en el backlog. El resto de reuniones se centran en añadir nuevas historias (si surgen) o en refinar la descripción de las historias de usuario de la pila.
- *Reunión de revisión del Sprint* (20 min). En esta reunión el equipo de trabajo realiza la demo del incremento al propietario del producto, con la asistencia del Scrum Master. El propietario debe aceptar o rechazar los resultados mostrados, aceptando o rechazando historias de usuario. En caso de rechazo, dicha historia de usuario debe volver a la pila del producto y el trabajo pendiente para finalizarla se aborda en la siguiente reunión de planificación. Esta reunión se realiza por primera vez tras finalizar el primer Sprint, precediendo la de planificación del segundo Sprint.
- *Reunión de retrospectiva* (10 min). Al finalizar cada Sprint y una vez realizada la revisión, el equipo se reúne (con presencia del Scrum Master) para analizar lo que ha ido bien y mal en el Sprint para seguir en una línea positiva de trabajo, mejorando su proceso de desarrollo y resolviendo conflictos que haya surgido. Para esta reunión se hace uso del calendario de la asignatura GPS, de modo que en la sesión posterior a la reunión de revisión se dedican los primeros diez minutos para esta reunión y se comparten las impresiones entre equipos para una mejor realimentación entre toda la clase.

Sistema de evaluación	Peso	PrIS	GPS	SSS
Prueba final	40%	Se realizan 10 tests de forma continua a lo largo de todo el curso, abordando cada tecnología. El peso de cada test es de 4 puntos	Se realizan dos pruebas de modo que de los 40 puntos acumulables la primera cuenta 15 y la segunda 25	
Elaboración de trabajos teóricos	20%	Se realizan trabajos en equipos reducidos (2-3) sobre cada una de las tecnologías de desarrollo del entorno tecnológico	Se realiza un trabajo teórico que cubre las partes no evaluadas en el caso práctico, incluyendo: Planificación estratégica, estimación y riesgos	Los estudiantes realizan un trabajo sobre alguna de las herramientas de KaliLinux
Realización de prácticas en laboratorio	30%	Evaluación del caso práctico (parte de desarrollo/ procesos)	Evaluación del caso práctico (parte de gestión)	Evaluación del caso práctico (parte de seguridad)
Participación en clase / Presentación oral de temas	10%	Exposición trabajos teóricos y participación en clase	Participación en reuniones del caso práctico	Exposición trabajos teóricos y participación en clase

Cuadro 1: Evaluación de las asignaturas PrIS, GPS y SSS.

Además de las reuniones anteriores, los equipos de trabajo deben realizar la reunión diaria. En este caso, salvo en la primera, no asiste el Scrum Master, dado que una vez que el equipo sabe realizarla convenientemente, deben funcionar como un equipo auto-organizado.

A partir de esta planificación, se elabora el calendario específico de cada asignatura, teniendo en cuenta que los estudiantes adquieran los conocimientos que se requieren para avanzar en el caso práctico. Por ejemplo, en GPS se dedican las sesiones magistrales iniciales y las prácticas de laboratorio a aprender Scrum y la herramienta de gestión (Zenhub).

Del mismo modo en PrIS se aborda el aprendizaje de las herramientas del entorno tecnológico y en SSS el aprendizaje de técnicas de web segura y criptografía. Las tres primeras semanas del calendario se centran en proporcionar dicha formación básica para que los equipos lleguen preparados a la primera reunión de planificación (semana 4).

En el primer Sprint, se destina una parte del esfuerzo previsto por el equipo a la instalación y configuración de las herramientas del entorno tecnológico, en paralelo al desarrollo de las clases magistrales de PrIS, centradas en las exposiciones de los trabajos que permiten profundizar en el aprendizaje de las herramientas.

En lo que respecta a la realización de pruebas escritas, las tres asignaturas van realizando varias pruebas siguiendo un sistema de evaluación continua. Los estudiantes que lo realizan de este modo obtienen su nota de prueba final como la media de las pruebas realizadas. En otro caso, el estudiante debe asistir a la prueba final en la convocatoria ordinaria y/o extraordinaria.

2.6. Evaluación

La evaluación aplica un sistema de puntuación en el que el estudiante puede acumular hasta 100 puntos (nota 10), siendo necesario para aprobar:

- Obtener 50 puntos (de los 100 posibles).
- Obtener una nota mínima de 4 en las partes obligatorias (prueba final, caso práctico).

En relación con la evaluación de las asignaturas, de acuerdo con la guía docente y siguiendo las directrices de la memoria de grado para la materia, se definen una serie de actividades comunes evaluables que consisten en la prueba final (40%), elaboración de trabajos teóricos (20%), realización de prácticas de laboratorio (30%) y participación/presentación oral de temas (10%). En el Cuadro 1 se muestra cómo se realiza la evaluación de cada apartado, de acuerdo con la evaluación específica que se realiza en cada asignatura.

Tal como se muestra en el Cuadro 1, la forma de conseguir los puntos relativos a la prueba final se planifica conforme a las particularidades de cada asignatura, de modo que se realizan dos o más pruebas parciales. En todo caso, el estudiante puede optar por realizar la prueba final si lo desea o en caso de no haber obtenido los mínimos necesarios en las pruebas escritas. Por su parte, el apartado de realización de prácticas de laboratorio consiste en evaluar el caso práctico común realizado, pero en cada asignatura se evalúa una perspectiva distinta (gestión, desarrollo, seguridad). Finalmente, en las asignaturas PrIS y SSS la nota de participación se obtiene a partir de las presentaciones en equipo, mientras que en GPS se obtiene evaluando la participación de los estudiantes en las reuniones Scrum.

3. Validación preliminar

La propuesta docente ha sido aplicada en el curso 2017/18, con 25 alumnos matriculados, agrupados en 4 equipos de trabajo. Los estudiantes rellenaron una encuesta anónima en la que se les preguntaba por su experiencia en cada una de las asignaturas y su experiencia global. Se optó por anonimizar la encuesta

dado que el objetivo era que los estudiantes contestasen con la mayor objetividad y libertad posible. Se obtuvo la respuesta de 10 estudiantes. En el Cuadro 2 se muestran los estadísticos descriptivos de la satisfacción de los estudiantes. Finalmente, se indica la satisfacción con la experiencia integrada y con la evaluación. Tal como se puede observar en el Cuadro 2, las asignaturas tuvieron una muy buena aceptación tanto a nivel de experiencia individual como de experiencia colectiva. Los estudiantes prefieren del mismo modo la experiencia integral respecto a impartir las asignaturas de forma individual.

	Mediana	Min	Max
A. PrIS	9,5	6	10
B. GPS	9,5	8	10
C. SSS	9	7	10
D. General			
Experiencia integral	10	7	10
Satisfecho evaluación	9,5	8	10

Cuadro 2.- Resultados de la encuesta

En cuanto al esfuerzo de cada asignatura, consideran que se trata de un valor razonable. Además, existe unanimidad en su opinión sobre el caso práctico, cuya complejidad consideran adecuada y realista. En cuanto a que la experiencia integral suponga mayor esfuerzo que cursar las asignaturas de modo individual, se obtuvo mayor dispersión de los resultados, pero con una mediana de 4,5 (sobre una escala de menor a mayor esfuerzo de 1 a 10), lo que sugiere que los estudiantes opinan que la experiencia integral no ha supuesto una carga adicional de esfuerzo.

En cuanto a sus recomendaciones de mejora, en líneas generales los estudiantes están satisfechos con los contenidos cursados, y en caso de sugerir profundizar en algún tema indican que se podría haber profundizado más en ciertas tecnologías de desarrollo orientadas a servicios (1 estudiante), en Spring (2 estudiantes) o Kali-Linux (3 estudiantes).

Finalmente, cabe destacar que las calificaciones obtenidas por los alumnos fueron ligeramente mejores que las del curso anterior y se redujo la tasa de abandono.

4. Factores de éxito y limitaciones

El hecho de llevar a cabo este tipo de propuestas de integración y coordinación de asignaturas supone un riesgo mayor que la realización individual de las mismas, sobre todo teniendo en cuenta las limitaciones a nivel de profesorado, horarios y espacios que pueden surgir. Ello sin olvidar las restricciones adicionales que se añaden a los estudiantes a la hora de realizar la matriculación en las asignaturas de la Intensificación en Ingeniería del Software. A continuación, se analizan estas limitaciones y se proponen algunas soluciones que podrían ayudar a resolverlas o mitigarlas:

- *Matriculación de los alumnos.* La participación en la experiencia integral implica una matriculación en bloque en las asignaturas correspondientes. En la experiencia de aplicación de las tres asignaturas se trató de solventar esta limitación avisando con antelación la forma en la que se iban a impartir y se permitió el cambio de matrícula una vez realizada la presentación de la asignatura. Ello facilitó que todo alumno que no había cursado ninguna de las tres asignaturas se matriculara de las 3 a la vez. Para aquellos alumnos que hubieran superado una o dos asignaturas en la modalidad anterior, se les ofreció realizar prácticas individuales o participar en el caso práctico. Aquellos alumnos con dos asignaturas aprobadas optaron por hacer prácticas individuales, y para los que sólo tenían una asignatura aprobada de las tres, seleccionaron realizar el caso práctico aun sabiendo que les suponía algo más de esfuerzo al incluirse en dicho caso los contenidos de las tres asignaturas.
- *Horarios de las asignaturas.* Un aspecto esencial que marca la dinámica del trabajo en el caso práctico es la realización de distintas reuniones siguiendo la filosofía de Scrum. Ello implica disponer de un horario amplio que permita que cada equipo de trabajo disponga al menos de una hora para realizar las reuniones correspondientes (planificación y revisión sprint). Para que ello sea posible en los horarios del curso, las asignaturas deben ocupar huecos horarios consecutivos en los mismos 3 días de la semana. Por tanto, se requiere que el centro planifique los horarios de esta forma, además de la coordinación de los horarios de los profesores responsables. En nuestra experiencia ello no supuso ningún problema, teniendo en cuenta la organización flexible que el centro realiza de los horarios para cada intensificación y el acuerdo de los profesores para que sus horarios permitiesen dar las asignaturas de forma consecutiva.
- *Aulas de teoría y laboratorio.* Para una exitosa aplicación de la integración se debe tener cierta flexibilidad en el horario que corresponde a las clases en aula de teoría y de laboratorio porque no obligatoriamente una asignatura debería planificar dos sesiones de teoría y una de laboratorio por semana, lo cual restaría flexibilidad. En realidad, dicha restricción en la implantación en el curso 2017/18 no existió, al llevar los alumnos sus portátiles al aula de teoría, lo que permitía realizar clases tanto teóricas, como de resolución de ejercicios o prácticas.
- *Esfuerzo del profesorado.* Además de la necesidad de coordinación de sus horarios personales, la implantación de la formación integrada re-

quiere una mayor coordinación entre los profesores de las asignaturas, lo cual incrementa su carga de esfuerzo. En nuestra experiencia, la carga ha sido mayor, pero hay que considerar que en la primera implantación es esperable dicha carga y en futuras repeticiones dicho esfuerzo debe tender a unas cifras razonables.

5. Conclusiones y trabajo futuro

En este artículo se ha descrito una propuesta de formación integral de asignaturas de la intensificación de Ingeniería del Software de modo que se facilite una mayor “inmersión” de los estudiantes en el tipo de trabajo que deberán desempeñar como Ingenieros Software en su futuro empresarial. Los resultados preliminares obtenidos son prometedores y a pesar de las dificultades organizativas que requiere este tipo de propuestas han sido bien acogidas.

Por ello como trabajo futuro se debe seguir avanzando en la propuesta, de modo que no sólo se considere la integración de una parte de las asignaturas, sino que abarque un número mayor de asignaturas de la intensificación. De hecho, los estudiantes participantes en la propuesta de integración y que también estaban matriculados en la asignatura de “Calidad de Sistemas Software”, han aplicado las herramientas de calidad a su caso práctico, aun no estando la asignatura de calidad incluida en el plan formativo integral.

Tal como también reflejan los resultados obtenidos en las encuestas, los estudiantes son receptivos a incorporar más asignaturas al plan, pero el diseño debe realizarse de forma cuidadosa teniendo en cuenta las restricciones adicionales que podrían surgir.

Referencias

- [1] Len Bass, Ingo Weber, and Liming Zhu. *DevOps: A Software Architect's Perspective*. Addison-Wesley Professional, 2015.
- [2] Óscar Belmonte, Mercedes Segarra, Reyes Grangel and Sergio Aguado. Desde la Iniciativa Empresarial hacia el éxito pasando por Metodologías Ágiles e Ingeniería del Software. *XXII Jornadas de la Enseñanza Universitaria de la Informática (JENUI)*, pp. 169–176, Almería, julio 2016.
- [3] Alejandro Bia and Santiago Matalonga. TBL aplicado a la Ingeniería de Software. *XXIII Jornadas de la Enseñanza Universitaria de la Informática, (JENUI 2017)*, pp. 273-280. Cáceres, 2017.
- [4] Óscar Cánovas, Gabriel López and Gregorio Martínez. Lego: un marco para el aprendizaje en el itinerario de Tecnologías de la Información. *XX Jornadas de la Enseñanza Universitaria de la Informática (JENUI)*, pp. 299-306. Cáceres, julio 2017.
- [5] María José García, Juan José Escribano and María Cruz Gaya. Experiencia de aplicación de ABP al Grado de Ingeniería Informática. *XX Jornadas de la Enseñanza Universitaria de la Informática (JENUI)*, pp. 125–132, Oviedo, julio 2014.
- [6] Thomas Limoncelli. Four ways to make CS and IT more immersive. *Commun. ACM*. 60(10), 50-52, 2017.
- [7] Project Management Institute. *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*. 6th Edition. 2017.
- [8] Pablo Sánchez, Carlos Blanco, Alejandro Pérez, Julio Medina, Patricia López, Alfonso de la Vega, Diego García and Miguel Sierra. Experiencia y Lecciones Aprendidas durante el Desarrollo de un Proyecto Software Común a Diversas Asignaturas. *XX Jornadas de la Enseñanza Universitaria de la Informática (JENUI)*, pp. 291-298. Cáceres, julio 2017.
- [9] Rafael Sebastian, Ricardo Olanda and Juan Manuel Orduña. Introducción de metodologías de aprendizaje basado en problemas en el marco de las TIC. *XIX Jornadas de Enseñanza Universitaria de la Informática (JENUI)*, pp. 153-160. Castellón, 2013.
- [10] Miguel Valero and Javier García. Cómo empezar fácil con PBL. *XVII Jornadas de Enseñanza Universitaria de la Informática (JENUI)*, pp. 109–116, Sevilla, julio 2011.
- [11] Carlos J. Villagrà-Arnedo, Francisco J. Gallego, Rafael Molina and Faraón Llorens. AB-Pgame+: siete asignaturas, un proyecto. *XX Jornadas de la Enseñanza Universitaria de la Informática (JENUI)*, pp. 285–292, Oviedo, Julio 2014.