

# Extensión de un proyecto Open Source con un proceso de pruebas bien asentado como modelo de aprendizaje para una asignatura de Verificación y Validación

Javier Nogueras-Iso, Miguel Á. Latre, Javier Lacasta, F. Javier Lopez-Pellicer, Rubén Béjar  
Grupo de Sistemas de Información Avanzados (IAAA)

Depto. de Informática. e Ing. de Sistemas / Instituto de Investigación en Ingeniería de Aragón (I3A)  
Universidad de Zaragoza

{jnog, latre, jlacasta, fjlopez, rbejar}@unizar.es

## Resumen

La asignatura de Verificación y Validación es una asignatura obligatoria dentro de la intensificación de Ingeniería del Software en el Grado en Ingeniería Informática. El principal objetivo de esta asignatura es que el alumno aprenda a diseñar, planificar e integrar las pruebas dentro del ciclo de vida de desarrollo de productos *software*, prestando especial atención a la automatización de las pruebas. Dado que los conceptos implicados en la puesta en marcha de un proceso de pruebas no se pueden adquirir únicamente desde un punto de vista teórico, se ha planteado la aplicación de una metodología de aprendizaje basada en proyectos tomando como punto de partida una herramienta Open Source que ya cuenta con un proceso de pruebas bien definido. Sobre este proyecto Open Source desarrollado con tecnología Java y con una amplia comunidad de usuarios, los alumnos deben extender la herramienta para incorporar una nueva funcionalidad de complejidad limitada pero que les permita poner en marcha un plan de pruebas adaptando la metodología TMAP y aplicar técnicas para diseñar y ejecutar pruebas en los distintos niveles de pruebas de desarrollo, sistema y aceptación. Este artículo describe la experiencia de la utilización de este tipo de proyectos durante tres cursos académicos.

## Abstract

Verification and Validation is a mandatory course of the Software Engineering graduate program in Computer Science Engineering studies. The main objective of this course is to train students for the design, planning and integration of test process within the life cycle of software products development, paying special attention to test automation. Since the concepts implied in the set up of a testing process cannot be learnt so-

lely from a theoretical perspective, this work proposes a project-based learning methodology taking as a starting point the development of an Open Source tool, which already has a well-defined testing process. Upon this Open Source project developed with Java technology, and supported by a wide user community, students must launch a test plan adapting TMAP methodology and apply the necessary techniques to design and execute tests for development, system and acceptance test levels. This paper describes the experience of using this kind of projects during three academic courses.

## Palabras clave

Proceso de pruebas, TMAP, *testing*, Verificación y Validación, Aprendizaje basado en proyectos

## 1. Introducción

La asignatura de Verificación y Validación es una asignatura obligatoria englobada en la materia de tecnología específica en “Proceso de desarrollo y ciclo de vida” dentro de la intensificación de Ingeniería del Software en el Grado en Ingeniería Informática de la Escuela de Ingeniería y Arquitectura de la Universidad de Zaragoza. Esta asignatura se imparte dentro del primer cuatrimestre de la intensificación dada la importancia de la verificación y validación dentro del ciclo de vida de desarrollo de un producto *software*. Aunque la mayor parte de los errores de un *software* se detectan durante las pruebas del código fuente compilado, según las estadísticas, más del 50 % de los errores de un producto *software* viene motivado por los defectos cometidos en las fases tempranas del desarrollo como la determinación de requisitos o el análisis [1]. El principal objetivo de esta asignatura es por tanto que el

alumno aprenda a diseñar, planificar e integrar el proceso de pruebas como una actividad que se desarrolla a lo largo de todo el ciclo de vida de un producto *software*. Además, dado que las pruebas pueden suponer hasta el 40 % del tiempo de desarrollo, se presta especial atención a que los alumnos sean capaces de utilizar herramientas que asistan y automaticen el diseño, codificación, ejecución y evaluación de las pruebas del *software*.

Por otro lado, dado que los conceptos implicados en la puesta en marcha de un proceso de pruebas no se pueden adquirir únicamente desde un punto de vista teórico, se ha planteado la aplicación de una metodología de aprendizaje basada en proyectos [6]. Esta metodología se considera especialmente adecuada en el contexto de la educación superior y lleva años aplicándose en la enseñanza de la informática en numerosas asignaturas [7], y especialmente en Ingeniería del Software [2, 3].

Para aplicar esta metodología se ha tomado como punto de partida una herramienta Open Source que ya cuenta con un proceso de pruebas bien definido. Sobre este proyecto Open Source desarrollado con tecnología Java y con una amplia comunidad de usuarios, se plantea a los alumnos que amplíen la herramienta para incorporar una nueva funcionalidad de complejidad limitada, pero que les permita poner en marcha un plan de pruebas adaptando la metodología TMAP Next (Test Management Approach) [5] y aplicar técnicas para diseñar y ejecutar pruebas en los distintos niveles de pruebas de desarrollo, sistema y aceptación.

El objetivo de este artículo es describir la experiencia de la implantación de este tipo de proyectos durante tres cursos académicos para que pueda ser de utilidad en su posible aplicación en otras asignaturas con unos contenidos similares. La sección 2 describe las principales características del proyecto propuesto a los alumnos. Después, la sección 3 detalla cada una de las tareas que componen el proyecto. Finalmente, el artículo termina con una descripción de los resultados obtenidos a través de esta experiencia docente, así como una sección con conclusiones sobre los problemas encontrados y las áreas de mejora.

## 2. Descripción del proyecto exigido a los alumnos

Tal como se ha comentado en la introducción, el trabajo propuesto para la implantación del proceso de pruebas se basa en la utilización de un proyecto Open Source sobre el que se realizan pequeñas ampliaciones de funcionalidad. En concreto, el proyecto Open Source utilizado es CatMDEdit<sup>1</sup>, una herramienta de

edición de metadatos que facilita la documentación de recursos, con especial énfasis en la descripción de recursos de información geográfica (ej: mapas con cartografía básica tanto en formato papel como digital; datos vectoriales conteniendo fenómenos discretos como parcelas catastrales, límites administrativos o redes fluviales; datos ráster conteniendo fenómenos continuos como imágenes aéreas o índices de vegetación; etc.).

Entre otras funcionalidades, la herramienta (véase fig. 1) permite editar metadatos y configurar la edición según distintos estándares de metadatos (ej: ISO19115 o Dublin Core), generar de forma automática metadatos para algunos formatos de información geográfica (ej: Shapefile, ECW, GeoTIFF, GIF, JPG, BMP) o conectarse con herramientas de Sistemas de Información Geográfica (SIG) para la visualización y modificación de los recursos referenciados por los metadatos.

Aparte de que CatMDEdit es una herramienta multi-plataforma (Windows, Linux, ...), desarrollada en Java (los programas de instalación y código fuente se puede descargar desde la plataforma SourceForge) y utilizada por una amplia comunidad de usuarios (más de 50.000 descargas desde 2004), la ventaja principal de este proyecto es que la organización que coordina su desarrollo está certificada en el nivel 2 de capacidad de la norma SPICE (Norma ISO/IEC 15504 “Software Process Improvement Capability Determination”) [4] y ha sistematizado una serie de prácticas relacionadas con las pruebas. SPICE es una norma abierta e internacional para evaluar y mejorar la capacidad y madurez de los procesos. En concreto, los proyectos de desarrollo certificados en nivel 2 (procesos gestionados) por la organización responsable de CatMDEdit deben seguir los siguientes procesos:

- MAN.3: *Project Management* (Gestión de proyecto). Dentro de este proceso hay que distinguir dos aspectos. Por un lado, en la planificación del proyecto se tiene en cuenta siempre una fase de pruebas para cada iteración planificada. Por otro lado, en el seguimiento y control del proyecto se establece la utilización de una herramienta de gestión de defectos para controlar errores.
- ENG.1: *Requirements elicitation* (Gestión de requisitos). Dentro de este proceso, y si el cliente lo solicita, se acuerda un conjunto de pruebas de aceptación y se crea un documento de pruebas de aceptación del producto.
- SUP.8: *Configuration Management* (Gestión de configuraciones). En este proceso se menciona explícitamente: “Todo Elemento de Configuración de *software* debe tener asociada una batería de pruebas que deberá ser superada antes de incorporar dicho Elemento de Configuración a una línea base”. Se cuenta con una plantilla de pruebas funcionales en base a una hoja de cálculo.

<sup>1</sup><http://catmdedit.sourceforge.net/>

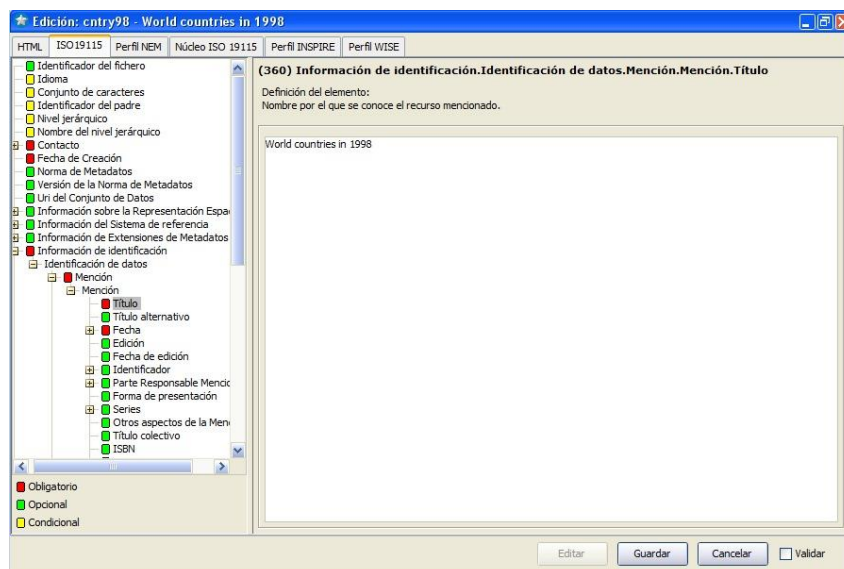


Figura 1: Herramienta Open Source CatMDEdit

En concreto, en el caso de CatMDEdit se dispone desde hace años de una hoja de control de pruebas de sistema. Esta hoja de control es una lista de verificación organizada por funcionalidades generales y específicas. Por cada elemento de esta lista, se describe brevemente el propósito, las acciones básicas de ejecución y el resultado esperado. Además, se dispone de un conjunto de datos y metadatos para realizar las pruebas especificadas. Aunque el objetivo fundamental de las pruebas preparadas es verificar la funcionalidad, también se prueban otras características de la calidad: portabilidad, conectividad, continuidad, facilidad de uso, ...

- MAN.6: *Measurement* (Medidas y análisis). Dentro de este proceso de medidas y análisis, se toman medidas sobre desviaciones de esfuerzos dedicados a las pruebas.

Tomando como punto de partida la política de pruebas adoptada en CatMDEdit y una pequeña ampliación de la herramienta realizada en 2012 para dar soporte a un nuevo estándar de metadatos, los profesores de la asignatura, que también han colaborado en el desarrollo del proyecto, implantaron la metodología TMAP Next para la gestión del proceso de pruebas de esta pequeña ampliación con el objetivo de que pudiese servir como base posteriormente para los proyectos propuestos a los alumnos de la asignatura de Verificación y Validación.

TMAP Next (Test Management Approach) es una metodología para la gestión de pruebas propuesta por la empresa holandesa Sogetti que ha ido y evolucionándose desde 1995. Ampliamente adoptada por empre-

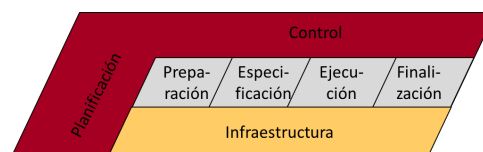


Figura 2: Fases del ciclo de vida de TMAP Next para un nivel de pruebas

sas y organizaciones a nivel mundial y utilizada como inspiración para la norma ISO 29119 publicada posteriormente [8], TMAP propone un proceso de pruebas estructurado que facilita responder a las cuestiones típicas de *qué*, *quién*, *cuándo*, *cómo* y *con qué* respecto a las actividades a realizar en un proceso de pruebas dirigido por el negocio donde se racionaliza el esfuerzo para abordar en primer lugar aquellos riesgos que resulten más prioritarios.

La aplicación de TMAP implica la puesta en marcha de una gestión global del proceso de pruebas, cuya principal herramienta es un plan de pruebas maestro, y procesos de prueba individualizados para los niveles de pruebas de desarrollo, sistema y aceptación. En la figura 2 se pueden observar las siete fases del modelo de ciclo de vida genérico para cada nivel de pruebas (los detalles concretos sobre las tareas a desarrollar en la implantación del proceso de pruebas se describen en la sección 3).

Dada la complejidad y amplitud del trabajo, y aunque se realiza en equipos de tres alumnos, este pondera un 80 % en la nota de la asignatura y todos los alumnos del mismo equipo reciben la misma calificación (salvo excepciones en las que se justifique claramente que algún alumno no ha contribuido adecuadamente al equi-

po). Además, la calificación del trabajo deber ser igual o superior a 5,0 para poder aprobar la asignatura y realizar la media ponderada con la nota del examen de la asignatura (que supone el 20 % de la nota de la asignatura y se realiza de forma individual). Adicionalmente, dentro del trabajo se ha establecido una valoración porcentual sobre cada uno de los conceptos implicados en la implantación de un proceso de pruebas:

- Gestión del proceso de pruebas: 30 %.
- Preparación, especificación y ejecución de las pruebas: 60 %.
- Presentación escrita y oral del trabajo: 10 %.

La planificación del trabajo está condicionada por la organización del contenido teórico-práctico de la asignatura. El cuadro 1 muestra el conjunto de hitos del trabajo a lo largo de las 15 semanas del curso y su sincronización con las clases de teoría y problemas en el aula y las sesiones prácticas en el laboratorio. Además, en la columna donde se muestran los contenidos prácticos se menciona también las herramientas de prueba utilizadas tanto para las sesiones en el laboratorio como para el trabajo.

### 3. Tareas del proyecto

#### 3.1. Gestión global del proceso de pruebas

Como se ha comentado anteriormente, para la gestión global del proceso de pruebas los alumnos deben seguir la metodología TMAP. Respecto a los entregables de la gestión global del proceso de pruebas se solicita, al menos, lo siguiente:

- Elaboración de un plan de pruebas maestro. Se toma como base la plantilla del plan de pruebas maestro facilitada por la metodología TMAP *Template Master Test Plan* y el ejemplo de plan de pruebas maestro para la versión 4.6.6 de CatMDEdit visto en clases de teoría. En dicho plan de pruebas no puede faltar la descripción de las actividades realizadas para el análisis de riesgo de productos, la estimación del esfuerzo de las pruebas, y un cronograma con la planificación temporal de los distintos niveles y fases de las pruebas. Además, dentro del plan de pruebas maestro se deben describir las infraestructuras que se planea crear para gestionar los productos de las pruebas, así como cualquier otro tipo de documentación o código fuente relacionado con la aplicación CatMDEdit. Es decir, se describe la organización de la carpeta del proyecto en formato digital con acceso para todo el equipo del proyecto (Google

Project Hosting, GitHub, Dropbox o similar) con las secciones que se consideren relevantes.

- Recopilación de esfuerzos por parte de cada miembro del equipo al nivel de fase, actividad y tarea. Se facilita una hoja de recopilación de esfuerzos con un estándar de codificación de esfuerzos previamente establecido.
- Gestión de defectos utilizando la herramienta propuesta en la segunda sesión de prácticas. El procedimiento de gestión de defectos se documentará también dentro del plan de pruebas maestro.
- Informe de pruebas. Se debe tomar como base la plantilla del informe final de pruebas de la metodología TMAP: *Template Final Test Report*, que se proporciona a los alumnos traducida al español.

#### 3.2. Pruebas de sistema y aceptación

Para la gestión del nivel de pruebas de sistema y el nivel de pruebas de aceptación se sigue también la metodología TMAP. Sin embargo, dadas las limitaciones de tiempo para realizar el trabajo en un cuatrimestre, no se exige elaborar una documentación de planificación y seguimiento del nivel de pruebas de sistema y aceptación de forma separada a la documentación de la gestión global del proceso de pruebas, sino que se pide integrar los detalles de estos dos niveles en los documentos descritos en la sección 3.1.

Respecto a los entregables del nivel de pruebas de sistema y aceptación, se exige a los alumnos, al menos, lo siguiente:

- Elaboración de una nueva versión del plan de pruebas maestro. En esta nueva versión se incorporarán los detalles sobre la planificación del nivel de pruebas de sistema y aceptación tomando en consideración la tabla de técnicas de diseño que se les ha presentado en las clases teóricas.
- Actualización del informe de pruebas presentado en la sección 3.1 para reflejar los avances en los niveles de pruebas de sistema y aceptación.
- Diseño de los casos de prueba. Dependiendo de los tipos de pruebas de sistema o aceptación que se planteen, se realiza un análisis para determinar los casos de prueba más relevantes utilizando las técnicas basadas en la estructura o en la especificación que se consideren más adecuadas. En la plantilla del informe de pruebas se prevé un apéndice para describir: las técnicas aplicadas para derivar los casos de prueba; los casos de prueba que se han obtenido como resultado de la aplicación de las técnicas anteriores; y la correspondencia con los *scripts* que automatizan o describen el procedimiento para ejecutar los casos de prueba.
- En el caso de que las pruebas de sistema y acep-

Sem	Contenidos teóricos	Contenidos prácticos	Hitos del trabajo
1	1. Gestión del proceso de pruebas		Creación de grupos
2	1. Gestión del proceso de pruebas	1. Ejecución de pruebas de sistema y aceptación	
3	1. Gestión del proceso de pruebas		Entrega de análisis y diseño
4	1. Gestión del proceso de pruebas	2. Introducción a las herramientas de gestión de defectos (herramientas: <i>Bugzilla</i> , gestión de <i>Issues</i> en <i>GitHub</i> )	
5	2. Pruebas dinámicas o <i>testing</i>		Entrega del plan de pruebas maestro
6	2. Pruebas dinámicas o <i>testing</i>		Presentación del plan de pruebas maestro
7	2. Pruebas dinámicas o <i>testing</i>	3. Automatización de pruebas dinámicas de código (herramientas: <i>JUnit</i> , <i>Hamcrest</i> y <i>jMock</i> )	
8	2. Pruebas dinámicas o <i>testing</i>		Evaluación de análisis y diseño
9	2. Pruebas dinámicas o <i>testing</i>	4. Automatización de pruebas en Android (herramientas: <i>AndroidJUnitRunner</i> y <i>Android Studio</i> )	
10	3. Evaluación		Entrega de la segunda versión del plan de pruebas maestro y primera versión del informe de pruebas
11	3. Evaluación	Taller de <i>Test Driven Development</i>	
12	3. Evaluación	5. Utilización de depuradores / análisis estático de código (herramientas: <i>PMD</i> y depuradores de <i>Eclipse</i> e <i>IntelliJ</i> )	
13	4. Evaluación y <i>testing</i> de interfaces persona ordenador		
14	4. Evaluación y <i>testing</i> de interfaces persona ordenador	6. Automatización de pruebas para aplicaciones con GUI (herramientas: <i>Sikuli</i> , <i>Selenium</i> , <i>Espresso</i> , <i>UI Automator</i> , <i>Exerciser Monkey</i> )	
15			Presentación y defensa final del trabajo

Cuadro 1: Cronograma con los contenidos de la asignatura e hitos del trabajo

tación se hayan automatizado, se solicita también la entrega de los *scripts* que se hayan preparado.

- Resultados de la ejecución de los casos de prueba. Para informar sobre los resultados de ejecución de las pruebas se permite a los alumnos tomar como base la hoja de control de pruebas de sistema utilizada en CatMDEdit, comentada en este artículo previamente y descrita a los alumnos en la primera sesión de prácticas. Como alternativa se permite informar brevemente sobre la ejecución de las pruebas en el apéndice del informe de pruebas antes mencionado.

### 3.3. Pruebas de desarrollo

Al igual que para los niveles de sistema y aceptación, dadas las limitaciones de tiempo que impone la realización del trabajo en un cuatrimestre, no se elabora una documentación de planificación y seguimiento

del nivel de pruebas de desarrollo de forma separada a la documentación de la gestión global del proceso de pruebas, sino que se integran los detalles de este nivel en los documentos presentados en la sección 3.1.

Respecto a los entregables del nivel de pruebas de desarrollo, se exige, al menos, lo siguiente:

- Elaboración de una nueva versión del plan de pruebas maestro. En esta nueva versión se pide incorporar detalles sobre la planificación del nivel de pruebas de desarrollo tomando en consideración la tabla de técnicas de diseño presentadas en las clases teóricas. También se permite añadir descripciones textuales sobre la aproximación que se va a utilizar (TDD, programación por pares, integración continua, ...).
- Actualización del informe de pruebas presentado en la sección 3.1 para reflejar los avances producidos en el nivel de pruebas de desarrollo.

- Diseño de los casos de prueba. En lo relativo a las pruebas unitarias del código desarrollado, para cada método o clase relevante, se habrá realizado un análisis para determinar los casos de prueba más relevantes utilizando las técnicas basadas en la estructura o en la especificación que se hayan considerado más adecuadas.

En la plantilla del informe de pruebas se prevé un apéndice para describir: las técnicas aplicadas para derivar los casos de prueba; los casos de prueba que se han obtenido como resultado de la aplicación de las técnicas anteriores; y la correspondencia con los *scripts* que automatizan o describen el procedimiento para ejecutar los casos de prueba.

- Se solicita la entrega, junto con el código del proyecto, de los *scripts* que se hayan preparado: programas aislados, *scripts* codificados para herramientas tipo JUnit o herramientas similares.
- Resultados de la ejecución de los casos de prueba. Para informar sobre los resultados de ejecución de las pruebas se sugiere utilizar las plantillas de TMAP (*Template Test Script* o *Template Test Scenario*) o informar brevemente sobre la ejecución de las pruebas en el apéndice del informe de pruebas antes mencionado.
- Dentro del apéndice del informe de pruebas antes mencionado se informa también del porcentaje de cobertura del nuevo código desarrollado que se haya alcanzado a través de las pruebas realizadas. Dicha cobertura se mide con herramientas como EclEmma, que se ve en las clases de teoría y problemas.
- En lo relativo al análisis estático de código, se exige que el código presentado esté libre de advertencias realizadas por el inspector del entorno de desarrollo. Para ello, los alumnos deben realizar una inspección del código del proyecto completo, de acuerdo a las indicaciones de la quinta sesión de prácticas.

## 4. Resultados obtenidos

La metodología de aprendizaje basado en proyectos se ha aplicado durante tres cursos académicos para poner en marcha procesos de prueba en pequeñas ampliaciones de la herramienta Open Source CatMDEdit.

Durante el primer curso, se contó con tres grupos de alumnos y a cada grupo de le encargó una ampliación distinta de CatMDEdit, pero de una complejidad similar:

- Actualización de la conexión con gvSIG<sup>2</sup> para la visualización de distintos tipos de datos como Shapefile, ECW, GeoTIFF, GIF, JPG o BMP.

<sup>2</sup><http://www.gvsig.org>

- Importación de metadatos recuperados de un servicio web de catálogo conforme con el estándar internacional CSW (*Catalog Services for the Web*)<sup>3</sup>.
- Exportación de metadatos de un repositorio a un servicio web de catálogo conforme el estándar CSW mencionado anteriormente.

Durante el segundo curso, se contó con cinco grupos de alumnos a los cuales también se les planteó cinco ampliaciones distintas de CatMDEdit:

- Edición de metadatos conformes con el estándar MARC21, utilizado habitualmente por archiveros y bibliotecarios.
- Edición de metadatos de conjuntos de datos de acuerdo al perfil DCAT. La Secretaría de Estado de Administraciones Públicas aprobó en febrero de 2013 una *Norma Técnica de Interoperabilidad de Reutilización de Recursos de Información*<sup>4</sup>. En particular, dentro de la norma técnica se propone un modelo de metadatos basado en RDF para la descripción de catálogos, que agrupan la descripción de un conjunto de recursos. Este modelo RDF está basado en el vocabulario internacionalmente reconocido como DCAT (*Data Catalog Vocabulary*)<sup>5</sup>.
- Gestión y descripción de repositorios de acuerdo a la descripción de catálogos del perfil DCAT, cuyo modelo RDF puede utilizarse como una alternativa para caracterizar los repositorios internos en los que CatMDEdit organiza los metadatos.
- Importación de los metadatos de un catálogo descritos de acuerdo al formato RDF propuesto por DCAT.
- Exportación de varios registros de metadatos como un grupo de registros de metadatos de acuerdo al formato RDF propuesto por DCAT para la descripción de catálogos.

Finalmente, en el tercer curso analizado bajo este tipo de proyecto se contó con cuatro grupos de alumnos, pero en este caso se decidió encargar el mismo tipo de trabajo a todos ellos: la construcción de una versión simplificada de la aplicación CatMDEdit que pudiese utilizarse en dispositivos móviles y ofreciendo un conjunto muy reducido de funcionalidades respecto a la versión escritorio. Además, para reducir la complejidad de la herramienta solo se solicitó la edición de metadatos con un estándar de metadatos muy simple denominado Dublin Core.

<sup>3</sup>[http://portal.opengeospatial.org/files/?artifact\\_id=20555](http://portal.opengeospatial.org/files/?artifact_id=20555)

<sup>4</sup><http://www.boe.es/boe/dias/2013/03/04/pdfs/BOE-A-2013-2380.pdf>

<sup>5</sup><http://www.w3.org/TR/2013/WD-vocab-dcat-20130312/>

Curso	Alumnos	MH	Sb	No	Ap	Su	NP
1.º	9	1	5	3			
2.º	16			6	8		2
3.º	14		2	11			1

Cuadro 2: Calificaciones de los alumnos por curso

Curso	N	J	S	Sb	No	Ap	NP	R
1.º	3	3	0	1	2			3
2.º	5	3	2		2	3		4
3.º	5	4	0		4		1	N/A

N: Número de equipos.

J: Número de trabajos entregados en primera convocatoria.

S: Número de trabajos entregados en segunda convocatoria.

R: Número de equipos que crearon una rama con su trabajo en SourceForge.

Cuadro 3: Calificaciones de los trabajos por curso

Respecto a las calificaciones obtenidas por los alumnos (puede verse un resumen de los tres cursos en los cuadros 2 y 3), en el primer curso, contando con 9 alumnos distribuidos en 3 grupos, 2 grupos obtuvieron notable en el trabajo y 1 sobresaliente. Ese primer curso, además, se optó por incentivar la creación de una rama nueva del CatMDEdit en el repositorio SourceForge con una subida de 0,5 puntos para cada integrante de cada grupo que la crearan. Finalmente, todos los equipos terminaron creando la rama en SourceForge correspondiente a su trabajo.

Durante el segundo curso, con un número superior de 16 alumnos, se observó un descenso en la calificación de los trabajos ya que 3 grupos obtuvieron un aprobado en el trabajo y los otros 2, un notable. Probablemente, una posible causa fue la elevada carga de trabajo que tienen los alumnos en el conjunto de las asignaturas del sexto semestre de la titulación, que en este curso se hizo evidente, junto con una mala planificación global de la gestión del tiempo por parte de los alumnos: 2 alumnos abandonaron completamente la asignatura, 2 equipos tuvieron que retrasar la entrega a la segunda convocatoria, y solo 4 de los 5 equipos crearon una nueva rama en SourceForge. Por otro lado, también concluimos que el tamaño del *software* y la complejidad de su estructura interna había sido un problema importante para que los alumnos se acercasen al verdadero objetivo de su trabajo, la puesta en marcha de un proceso de pruebas.

Por ello, en el tercer año, como se ha comentado anteriormente, se decidió plantear un mismo tipo de aplicación, con una funcionalidad muy limitada y con una base tecnológica que los alumnos podían reutilizar de la asignatura de Ingeniería del Software, cursada por

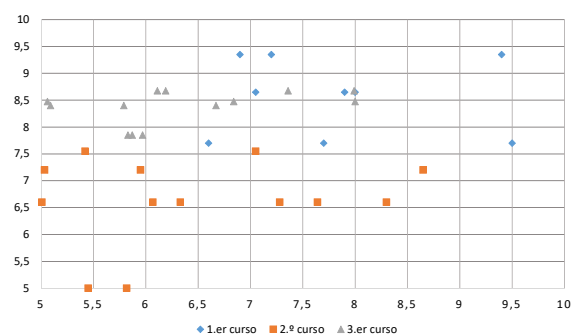


Figura 3: Calificación del trabajo (ordenadas) frente a la calificación del examen teórico (abscisas)

la gran mayoría de ellos en el semestre anterior. Como resultado, en este tercer curso con 14 alumnos (aunque uno abandonó) se volvió a remontar el nivel de las calificaciones ya que los cuatro grupos obtuvieron un notable alto.

Además, teniendo en cuenta la evaluación global de los alumnos en la asignatura donde también se tiene en cuenta el 20 % relativo a la puntuación del examen, debe resaltarse que los 38 alumnos que realizaron el trabajo sacaron adelante la asignatura dentro de su curso académico y con una nota media de 7,7. Entre los 3 cursos académicos solo 2 alumnos del segundo curso y 1 alumno del tercer curso decidieron no entregar el trabajo ni realizar el examen teórico. El gráfico de dispersión de la figura 3 permite ver la relación entre las calificaciones de los alumnos en el examen escrito (abscisas) y en el trabajo (ordenadas) y la evolución durante los tres cursos académicos: notas elevadas en ambas pruebas durante el primer curso, notas más bajas también en ambas pruebas durante el segundo y, en el tercero, el incremento general en las notas del trabajo pese a que la variedad en las notas del examen persisten. En cualquier caso, el gráfico nos permite observar que las notas del trabajo son en su mayoría superiores a las del examen teórico (la mayoría de los puntos están por encima de la recta correspondiente a la función identidad), lo cual nos permite deducir que la consecución del trabajo permite asentar los conceptos teóricos evaluados con el examen.

Respecto a la satisfacción de los alumnos recogida en las evaluaciones de la enseñanza de esta asignatura en el segundo y tercer curso (no se tienen datos del primero), esta fue bastante alta, obteniendo una puntuación cercana a 4 en una escala de 1 a 5. Aunque en los comentarios de respuesta abierta los alumnos remarcan que el tema donde se expone la metodología del proceso de pruebas les parece un poco más denso y menos útil a corto plazo, la impresión global de la asignatura es que les ayuda a afianzar los conceptos adquiridos en otras materias relacionadas con la Ingeniería del Soft-

ware y a conocer desde un contexto más realista cómo se realiza el desarrollo de proyectos.

Desde el punto de vista de los profesores, también se debe resaltar que la preparación de trabajos individualizados sobre la herramienta CatMDEdit en los dos primeros cursos exigió una importante carga de trabajo adicional, que no fue apreciada especialmente por los alumnos. Aunque se pretendía fomentar la originalidad de cada trabajo y que cada equipo contribuyese a la evolución del proyecto Open Source, la puesta en contexto retrasó un poco que los alumnos se centrasen en los objetivos reales de la asignatura.

## 5. Conclusiones

Este artículo ha presentado la metodología de aprendizaje basado en proyectos aplicada a una asignatura de Verificación y Validación de la intensificación en Ingeniería del Software de los estudios del grado de Ingeniería Informática.

Con esta experiencia se ha pretendido que los alumnos abordasen el problema de la puesta en marcha de un proceso de pruebas en un entorno que se asimila bastante al que se van a encontrar en su futuro profesional. Para no desbordar a los alumnos con el trabajo de desarrollo de ampliación de funcionalidad propuesto sobre el proyecto Open Source CatMDEdit elegido y que no es el foco de los objetivos de la asignatura, los profesores han intentado limitar al máximo posible su complejidad y facilitar todo el apoyo posible para su desarrollo. Además, cabe destacar que se ha elegido este proyecto por contar ya con una política de pruebas bien asentada, que además se ha readaptado a la metodología TMAP, la cual, por su carácter estructurado, es idónea para introducir a los alumnos en la gestión de un proceso de pruebas.

Respecto a los resultados obtenidos por los alumnos, puede extraerse que el método aplicado ha funcionado bastante bien. No solo la inmensa mayoría de los alumnos han realizado tanto el trabajo como el examen teórico, sino que además su nota global ha sido bastante alta.

No obstante, sí que hemos notado a lo largo de los tres cursos que, aunque las ampliaciones de funcionalidad solicitadas sobre CatMDEdit son muy limitadas, pueden ser un obstáculo para que los alumnos se centren más rápidamente en el objetivo principal del trabajo y la asignatura: gestionar un proceso de pruebas, y llevar a cabo las tareas propias del diseño, codificación y evaluación de las pruebas. Por ello, en años posteriores, el trabajo planteado utiliza como objeto de pruebas directamente un proyecto cuyo desarrollo ha sido realizado en una asignatura previa en el plan de estudios (Ingeniería del Software), con lo que los alumnos tie-

nen más facilidad para centrarse únicamente durante el periodo de impartición de Verificación y Validación en poner en marcha el proceso de pruebas.

## Agradecimientos

Este trabajo ha sido financiado parcialmente por el Gobierno de España a través del proyecto TIN2017-88002-R.

## Referencias

- [1] R. Bender. *Requirements Based Testing Process Overview*. Benders RBT Inc., 2009.
- [2] Pablo Fernandez, José María García, José A. Parejo, and Antonio Ruíz-Cortés. Extendiendo ABP con feedback ágil en la enseñanza de Ingeniería del Software. In *Actas de las XXIII JENUI*, pages 281–288, Cáceres, Julio 2017.
- [3] Alfredo Goñi, Jesús Ibáñez, Jon Iturrioz, and José Ángel Vadillo. Aprendizaje Basado en Proyectos usando metodologías ágiles para una asignatura básica de Ingeniería del Software. In *Actas de las XX JENUI*, pages 133–140, Oviedo, Julio 2014.
- [4] International Organization for Standardization/International Electrotechnical Commission. ISO/IEC 15504 Information technology - Process assessment. “Software Process Improvement and Capability Determination (SPICE)”, 2004.
- [5] T. Koomen, L. van der Aalst, B. Broekman, and M. Vroon. *TMap Next for result-driven testing*. Uitgeverij Tutein Nolthenius, Den Bosch, The Netherlands, 2006.
- [6] T. Markham. *Project based learning handbook*. Buck Institute for Education, 2003.
- [7] R. Puchera and M. Lehner. Project based learning in computer science - a review of more than 500 projects. In *2nd Int. Conf. on Education and Educational Psychology*, volume 29, pages 1561–1566, 2011.
- [8] SoftwareTestingStandard.org. ISO/IEC/IEEE 29119 Software Testing. The International Software Testing Standard. <http://www.softwaretestingstandard.org/>, 2013.