



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Gº en Ingeniería en Informática



**Trabajo final de Gº Ing. Informática:**

**Autocorrección de prácticas en Java:  
Plataforma web para el análisis automático  
de prácticas en Java.**



Presentado por Álvaro Vázquez Gómez

en febrero 2016

Tutor Bruno Baruque Zanón





UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Gº en Ingeniería en Informática



D. Bruno Baruque Zanón, profesor del departamento de Ingeniería Civil, área de Lenguajes y Sistemas Informáticos.

Expone:

Que el alumno D. Álvaro Vázquez Gómez, con DNI 71293190K, ha realizado el Trabajo final del GºIng.Informática titulado: Autocorrección de prácticas en Java.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual, Se autoriza su presentación y defensa.

En Burgos a 10 de Febrero de 2016

Bruno Baruque Zanón





## Índice de contenido

I -	Introducción.....	6
II -	Objetivos del proyecto .....	7
1.	Objetivos funcionales .....	7
2.	Objetivos de carácter técnico .....	8
III -	Conceptos Teóricos .....	9
1.	Learning Management System (LMS).....	9
2.	Plataforma Moodle .....	10
3.	Software QA .....	11
IV -	Técnicas y herramientas.....	13
1.	Técnicas de desarrollo .....	13
1.1.	LTI .....	13
1.2.	Modelo-vista-controlador (MVC) .....	13
1.3.	Desarrollo ágil de software .....	15
1.3.1.	SCRUM.....	15
2.	Herramientas de software .....	16
2.1.	Lenguajes .....	16
2.1.1.	HTML5 .....	16
2.1.2.	PHP .....	16
2.1.3.	JavaScript .....	17
2.1.4.	AJAX .....	17
2.1.5.	JQuery .....	18
2.1.6.	JSON .....	18
2.1.7.	Bootstrap .....	18
2.1.8.	XML .....	19
2.2.	Herramientas .....	19
2.2.1.	GitHub.....	19
2.2.2.	XP-DEV .....	19
2.2.3.	Moodle .....	20
2.2.4.	WampServer.....	20
2.2.5.	CakePHP .....	20





2.2.6.	Maven.....	21
2.2.7.	Eclipse for PHP Developers .....	21
2.2.8.	MySQL.....	21
2.2.9.	HeidiSQL .....	22
2.2.10.	Astah.....	22
2.2.11.	PoEdit .....	23
2.2.12.	PHPUnit.....	23
2.2.13.	FileSync.....	23
V -	Trabajos relacionados .....	24
1.	UBUVirtual.....	24
1.1.	Similitudes con nuestra aplicación web.....	24
2.	SonarQube .....	26
2.1.	Similitudes con nuestra aplicación web.....	27
3.	Laboratorio Virtual de Programación para Moodle (VPL) .....	27
3.1.	Similitudes con nuestra aplicación web.....	28
4.	Autocorrección de Prácticas en Java para Moodle v2.0.....	31
4.1.	Similitudes con nuestra aplicación web.....	31
VI -	Aspectos relevantes del desarrollo .....	34
1.	Utilizar LTI y desarrollar una aplicación web.....	34
2.	Seguridad de la aplicación .....	34
3.	Estructura de directorios generada automáticamente en el servidor.....	35
4.	Elección de la forma de validar el consumer key .....	36
5.	Elección del framework CakePHP .....	36
6.	Elección de Maven.....	36
7.	Elección de Bootstrap .....	37
8.	Elección de los plugins para los reportes .....	37
9.	Elección de la forma de añadir los plugins.....	37
10.	Elección de la herramienta para los plagios .....	38
VII -	Conclusiones y líneas de trabajo futuras .....	39
1.	Conclusiones.....	39
2.	Líneas de trabajo futuras .....	40
VIII -	Bibliografía .....	41



## Índice de ilustraciones

ILUSTRACIÓN 1: PARTES INVOLUCRADAS EN UN LMS.....	9
ILUSTRACIÓN 2: INTERACCIÓN EN MOODLE .....	10
ILUSTRACIÓN 3: TAREAS DEL QUALITY ASSURANCE (QA).....	11
ILUSTRACIÓN 4: CICLO DE VIDA DEL MVC.....	14
ILUSTRACIÓN 5: PARÁMETROS DE LA TAREA EN UBUVIRTUAL .....	24
ILUSTRACIÓN 6: PARÁMETROS DE LA TAREA EN NUESTRA APLICACIÓN .....	25
ILUSTRACIÓN 7: FORMULARIO DE SUBIDA DE FICHEROS EN UBUVIRTUAL .....	25
ILUSTRACIÓN 8: FORMULARIO DE SUBIDA DE FICHEROS EN NUESTRA APLICACIÓN .....	26
ILUSTRACIÓN 9: EJES DE LA CALIDAD DE CÓDIGO EN SONARQUBE .....	27
ILUSTRACIÓN 10: PARÁMETROS DE LA TAREA EN VPL.....	28
ILUSTRACIÓN 11: VER PARÁMETROS DE LA TAREA EN NUESTRA APLICACIÓN .....	29
ILUSTRACIÓN 12: PARÁMETROS DE LA TAREA EN NUESTRA APLICACIÓN .....	29
ILUSTRACIÓN 13: ENUNCIADO DE PRÁCTICAS EN NUESTRA APLICACIÓN .....	29
ILUSTRACIÓN 14: FORMULARIO DE SUBIDA DE PRÁCTICAS EN VPL .....	30
ILUSTRACIÓN 15: FORMULARIO DE SUBIDA DE PRÁCTICAS EN NUESTRA APLICACIÓN .....	30
ILUSTRACIÓN 16: FORMULARIO SUBIDA DE PRÁCTICAS APJ PARA MOODLE 2.0 .....	32
ILUSTRACIÓN 17: FORMULARIO SUBIDA DE FICHEROS Y RESULTADOS EN NUESTRA APLICACIÓN.....	32
ILUSTRACIÓN 18: ESTADÍSTICAS EN APJ PARA MOODLE 2.0 .....	33
ILUSTRACIÓN 19: ESTADÍSTICAS EN NUESTRA APLICACIÓN .....	33
ILUSTRACIÓN 20: ESTRUCTURA DE CARPETAS.....	35





## Resumen

Hoy en día, nuevas herramientas tecnológicas están adquiriendo mayor importancia en el proceso de enseñanza-aprendizaje.

La aplicación de Autocorrección de Prácticas en Java ha sido desarrollada con el objetivo de crear una aplicación web que pueda ser usada dentro de un entorno educativo, en la que tanto profesores como alumnos puedan obtener un beneficio al usarla.

Por un lado, los alumnos podrán subir sus prácticas realizadas en lenguaje Java para obtener los resultados tras el proceso de corrección de la misma, sin necesidad de esperar a una re-alimentación por parte del profesor.

Por otro lado, los profesores podrán subir los test que van a aplicarse las prácticas y realizar un seguimiento general del avance de la solución de las mismas por parte de la clase, así como comprobar si existen plagios entre prácticas.

La aplicación se conecta con un LMS gracias al LTI, el cual puede ser incorporado en una gran cantidad de LMS presentes y futuros.

La gran variedad de dispositivos con pantallas de diferentes tamaños presentes en el mercado, hacen que la aplicación de Autocorrección de Prácticas en Java haya sido desarrollada con la idea de ofrecer la máxima compatibilidad, es por ello por lo que se ha empleado el diseño *responsive design*.

## Palabras clave

*“LTI, Moodle, PHP, CakePHP, MVC, Test, Prácticas, Plugins, Moodle, Consumer Key, Secret, Responsive Design”*





## Abstract

Today, new technological tools are becoming more important in the teaching-learning process.

Autocorrect Java Practices application has been developed with the aim of creating a web application that can be used within an educational environment in which teachers and students can make profit by using it.

On the one hand, students can upload their practices developed in Java language to get the results after correction process of the same, without having to wait for a feedback by the teacher.

On the other hand, teachers can upload the test to be applied to practices and conduct a general monitoring of progress in solving them by the class and check for plagiarism between practices.

The application connects to an LMS thanks to the LTI, which can be incorporated into a lot of LMS today and tomorrow.

The variety of devices with different screen sizes on the market, make Autocorrect Java Practices application has been developed with the idea of offering the maximum compatibility, it is for this reason that it has been used responsive design.

## Keywords

*“LTI, Moodle, PHP, CakePHP, MVC, Test, Practices, Plugins, Moodle, Consumer Key, Secret, Responsive Design”*





## I - INTRODUCCIÓN

La idea de desarrollar este trabajo fin de grado surgió con la necesidad de crear una herramienta capaz de permitir al alumno subir sus prácticas para que sean analizadas y posteriormente poder conocer los resultados obtenidos tras este proceso de auto corrección de las mismas a partir de los test subidos por el profesor. Por otro lado, el profesor podrá tener una visión global de los resultados de las prácticas de todos los alumnos, además de otra serie de funcionalidades.

Por lo tanto, la aplicación proporcionará diferentes funcionalidades que dependerán del rol (profesor o alumno) del usuario que haya accedido a la misma.

Una de las metas de la aplicación ha sido que los alumnos tras subir sus prácticas para ser analizadas, pudieran conocer los resultados de este análisis por medio de una serie de reportes (métricas, violaciones de código cometidas, errores cometidos...). Además también se ha considerado como punto fuerte ofrecerles unas gráficas en las que pudieran ir viendo la evolución respecto a esas métricas de sus prácticas subidas.

Un pilar importante de la aplicación ha sido poder proporcionarle al profesor la máxima funcionalidad posible para poder tener un control de la información de todos los alumnos, además de funcionalidades básicas como son los formularios con los que podrá subir sus test y configurar los parámetros de la tarea.

Derivado de lo anterior, era muy importante ofrecerle al profesor una visualización de las estadísticas referentes a las prácticas subidas por los alumnos. También desde el primer momento se pensó en proporcionarle una opción con la que pudiera comprobar si se han cometido plagios entre las prácticas de los alumnos.

Otro de los pilares sobre los que se ha desarrollado la aplicación, ha sido que pudiera tener un acoplamiento o enlazamiento con la gran mayoría de plataformas denominadas *Learning Management Systems* (como Moodle, Blackboard, etc), de forma que los usuarios que accedan a dicha plataforma puedan entrar en la aplicación a partir de las tareas externas creadas desde esas plataformas, aprovechando toda la información y control de usuarios ofrecidas por éstas. Para que esto haya sido posible se ha empleado el estándar LTI que ha permitido realizar la conexión entre el LMS y la aplicación, además de dar la ventaja de permitir obtener datos procedentes de la plataforma origen (nombre de usuario, rol, correo...) para posteriormente poder ser tratados dentro de la aplicación presentada.

En líneas generales, el trabajo fin de grado se ha realizado con el objetivo de que tanto el profesor como los alumnos, obtuviesen un beneficio mutuo derivado de su uso.

A lo largo del presente documento se explicarán en diferentes apartados los procesos seguidos para desarrollar el trabajo fin de grado.



## II - OBJETIVOS DEL PROYECTO

### 1. *Objetivos funcionales*

Los principales objetivos del trabajo fin de grado son:

1. Generar una aplicación web que permita:

#### A los profesores:

- Poder registrarse en la aplicación web para obtener los parámetros LTI necesarios para poder crear tareas, que enlacen con la aplicación web, desde el LMS de su elección (preferentemente Moodle).
- Crear tareas de tipo “herramienta externa” (que se corresponderían con las prácticas del curso) desde Moodle que le permitan generar un espacio para la subida de ejercicios prácticos en la aplicación.
- Subir los test que van a aplicarse a las prácticas que suban los alumnos. También se le permite subir un enunciado para la práctica.
- Configurar los parámetros (nombre del paquete, número máximo de intentos que tienen los alumnos para subir sus prácticas...) de la tarea.
- Consultar las estadísticas referentes a cada una de las prácticas subidas por los alumnos.
- Consultar gráficas para tener una visión global de todas las prácticas subidas por los alumnos.
- Comprobar si existen plagios entre las prácticas subidas por los alumnos.

#### A los alumnos:

- Acceder a la aplicación desde las tareas del LMS creadas por el profesor.
- Consultar los parámetros de la tarea: número máximo de intentos posible, fecha límite de entrega, número de intentos realizados...
- Consultar el enunciado de la práctica subido por el profesor.
- Subir sus prácticas y que sean analizadas, obteniendo los resultados correspondientes tras este proceso de “corrección”: si la práctica ha compilado o no, si la práctica pertenece al paquete correcto, si ha pasado los test o no...





- Tras subir una práctica poder consultar los reportes generados.
  - Poder visualizar las gráficas relacionadas con las prácticas subidas.
2. Crear una base de datos que contenga toda la información necesaria para hacer funcionar a la aplicación web.
  3. Hacer que la aplicación web esté integrada con las plataformas LMS (en especial en la plataforma Moodle, por ser la empleada en la Universidad de Burgos).

Otros objetivos serían que la aplicación web presente un diseño sencillo e intuitivo y que permita la visualización en diferentes dispositivos.

Y por último destacar que tras la realización del trabajo fin de grado, se adquieran conocimientos en el campo del desarrollo de aplicaciones web.

## **2. Objetivos de carácter técnico**

En cuanto a los objetivos de carácter técnico se encontrarían los siguientes:

- Para realizar el diseño de la aplicación web se utilizará el *framework* Bootstrap.
- Se utilizará la plataforma Moodle como plataforma de LMS para poder acceder a la aplicación web.
- Para enlazar Moodle con la aplicación web se utilizará el *plugin* LTI de Moodle, el cual va a permitir conectarse a una aplicación externa, así como poder obtener datos de Moodle (nombre del usuario, correo...) para poder ser usados en la aplicación web.
- Desarrollar la plataforma empleando herramientas de tipo *open-source* (por ser una plataforma educativa) y a ser posible con PHP por ser más compatible con Moodle.



### III - CONCEPTOS TEÓRICOS

#### 1. Learning Management System (LMS)

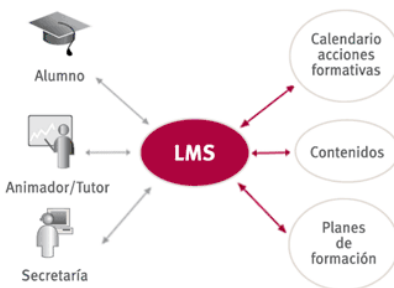
El LMS, Sistema de Gestión de Aprendizaje, es una aplicación instalada en un servidor que se va a emplear para gestionar las actividades relacionadas con la educación [1].

Principalmente las funcionalidades básicas que proporcionan los LMS son las siguientes:

- **Gestión de usuarios:** Relacionada con el registro de profesores y alumnos, dando la posibilidad de modificar los datos personales de cada uno de ellos.
- **Gestión de cursos y grupos:** Permite la creación y gestión de cursos y grupos de trabajo. Dentro de los cursos se van a encontrar los materiales educativos que se van a proporcionar posteriormente a los alumnos.
- **Herramientas de comunicación:** Dentro del sistema se da la posibilidad de incluir diferentes herramientas de comunicación, como por ejemplo foros, chats, etc.
- **Herramientas de evaluación:** La presencia de alumnos obliga a tener algún tipo de metodología que permita evaluar las tareas realizadas por los alumnos. Algunas metodologías podrían ser la realización de algún tipo de examen, subida de trabajos y prácticas para evaluar y obtener la nota correspondiente.

El objetivo de este tipo de herramientas es permitir el aprendizaje en cualquier parte y en cualquier momento. La mayoría de estas herramientas son herramientas web. Por lo tanto para la los alumnos, la plataforma LMS es un sitio web al que se conectan para acceder a los contenidos docentes.

Existen multitud de sistemas LMS disponibles para la comunidad educativa, tanto comerciales (WebCT, BlackBoard, entre otros) como de libre distribución (Moodle, Dokeos, Claroline, entre otros) [2].



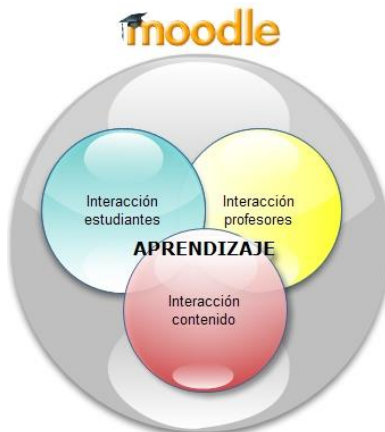
*Ilustración 1: Partes involucradas en un LMS*





## 2. **Plataforma Moodle**

Moodle, acrónimo de *Modular Object Dynamic Learning Environment* (Entorno de Aprendizaje Modular Orientado a Objetos), es un Sistema de Gestión de Cursos de código abierto, conocido también como Sistema de Gestión de Aprendizaje (*Learning Management System*, LMS). Es la plataforma más utilizada de todos los LMS disponibles, muy popular entre los educadores de todo el mundo como una herramienta para crear sitios web dinámicas en línea para sus estudiantes [3].



*Ilustración 2: Interacción en Moodle*

Para utilizarla el usuario únicamente necesitará un navegador web en su ordenador y una conexión a internet para poder interactuar con la herramienta. Permite utilizarse en múltiples sistemas operativos (Windows, Linux, Mac OS, etc).

Dentro de las características generales de Moodle encontramos [2]:

- Permite ser ejecutado en múltiples sistemas operativos que soporten la tecnología PHP.
- Está diseñado de manera modular, lo que permite una gran flexibilidad tanto para añadir como para eliminar funcionalidades encontradas en diferentes niveles.
- Tiene una interfaz simple, ligera, eficiente y compatible con multitud de navegadores web.
- Puede ser utilizado para impartir múltiples cursos, a los cuales van a poder matricularse los diferentes alumnos. El profesor que ha creado el curso podrá dar acceso a invitados e incluso a otros profesores.



Algunos de los módulos que se ofrecen en Moodle son: módulo de tareas, de consulta, de foro, diario, cuestionario, recurso, encuesta, wiki y taller.

Existen diferentes roles de usuarios en Moodle: Administrador (puede realizar cualquier modificación), Creador de cursos, Profesor, Profesor sin permisos de edición, Estudiante e Invitado.

### 3. *Software QA*

El software QA, *Quality Assurance*, o aseguramiento de la calidad de una aplicación o de un sistema siempre es esencial, en especial en equipos medios-grandes de desarrollo. A pesar de que su traducción da razones para ello, QA no es sólo *testing*, sino que también es análisis, diseño, seguimiento, control, planificación... Y muchas otras tareas que pueden estar involucradas y que son tan importantes dentro de cualquier proyecto.



*Ilustración 3: Tareas de Quality Assurance (QA)*

El proceso de asegurarse de que un sistema funcione tal y como ha sido desarrollado, y de la forma que tiene que funcionar, no es un proceso sencillo sino que requiere de mucha planificación y disciplina. Y además, convencer a otros de que el sistema va a funcionar correctamente requiere de un esfuerzo aún mayor. Por lo tanto QA se debe de realizar en todas las etapas del proyecto y no sólo al final del proyecto, con las prisas que eso conlleva.

El plan QA comienza el proceso de desarrollo desde el nacimiento del proyecto hasta la implementación del software. En las primeras etapas se verifican que los objetivos estén bien definidos, mientras que en las siguientes etapas se vigila el cumplimiento de los estándares fijados. Finalmente, revisa que el software en funcionamiento respete los requerimientos pedidos y que la entrega al cliente sea realizada en las condiciones adecuadas [4].





El QA se basa en un conjunto de pruebas de calidad entre las que se incluyen:

- **Testeo unitario:** cada uno de los módulos son probados por separado para verificar que funcionan correctamente.
- **Prueba de stress:** es una prueba de resistencia de la aplicación, que consiste en enviar una cantidad de peticiones excesiva con el objetivo de que colapse.
- **Test de integración:** los módulos que han sido probados en el testeo unitario se acoplan y se comprueba que funcionan correctamente en conjunto.
- **Test funcional:** se prueba que el software ofrezca las funciones solicitadas.
- **Test de aceptación:** el usuario se encarga de verificar que el producto satisfaga sus expectativas.

Estas pruebas no son beneficiosas solo para el usuario final sino que también lo son para el equipo de desarrollo del producto, ya que al haber establecido un control permanente sobre el proceso evitará en gran medida tener que corregir errores en etapas más avanzadas del proyecto [5].

En resumen, QA trata sobre el aseguramiento de la calidad y la credibilidad del producto: es decir, el producto funcionará según lo especificado y los usuarios deben creer que funcionará correctamente.

En este proyecto se adoptan algunas de las técnicas empleadas en el mundo profesional pero muy simplificadas y adaptadas a la docencia en programación de ordenadores. Esto ayuda a los alumnos a empezar a relacionarse con este concepto y tareas de forma sencilla.





## IV - TÉCNICAS Y HERRAMIENTAS

### 1. Técnicas de desarrollo

#### 1.1. LTI

*Learning Tools Interoperability (LTI)* se trata de un estándar desarrollado por *IMS Global Learning Consortium*. El objetivo principal del LTI es establecer un marco en el que integrar aplicaciones educativas con portales que gestionan entidades académicas.

Permite interrelacionar aplicaciones diferentes, pudiendo transmitir información de una a otra, siempre que implementen dicho estándar. Esta interrelación se realiza entre una plataforma LMS (en nuestro caso Moodle) y una aplicación web, de manera que la primera pueda enviar aquella información necesaria para que la segunda la reciba y pueda utilizar.

Para que lo anterior sea posible además se necesitan un *consumer\_key*, *secret* y *url* que son proporcionados por la aplicación web, para posteriormente poder crear desde la plataforma LMS una tarea que enlace con la aplicación web. Es importante que esos tres parámetros se establezcan correctamente, ya que sino no será posible enlazar con la aplicación.

El estándar LTI se trata de un concepto fundamental a la hora de realizar el trabajo fin de grado, ya que es el que va a permitir que nuestra aplicación establezca una conexión con Moodle, y de esta forma pueda recoger todos los datos necesarios de Moodle (rol del usuario, id's, nombres, etc.).

#### 1.2. Modelo-vista-controlador (MVC)

Es un patrón de diseño de software que separa los datos y la lógica de negocio de la interfaz de usuario. Convierte la aplicación en un paquete más fácil de mantener y además provoca una mejora de rapidez de desarrollo.

Separa las tareas de la aplicación en tres capas: modelos, vistas y controladores. Esto hará que la aplicación sea más sencilla de entender, permitiendo que las nuevas características se añadan fácilmente. Hacer cambios en una parte de la aplicación no va a afectar a las demás partes. Explicación de cada una de las capas [6]:

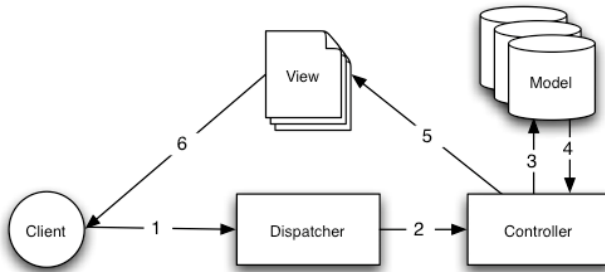
- **Modelo:** Representa la parte de la aplicación que implementa la lógica de negocio. Es la capa responsable de gestionar los datos: procesamiento, recuperación, validación, asociación...
- **Vista:** La vista hace una representación de los datos del modelo, es decir, se trata de la interfaz de usuario.





- **Controlador:** Se encarga de gestionar las peticiones recibidas por parte de los usuarios, y es la capa encargada de responder la ayuda solicitada apoyándose tanto en el modelo como en la vista.

El ciclo de vida de una petición típica en un modelo MVC sería el siguiente:



*Ilustración 4: Ciclo de vida del MVC*

Y su explicación:

1. Todo comienza cuando el usuario solicita una página o un recurso de la aplicación. Esta petición será procesada por el despachador, el cual va a ser el encargado de seleccionar el controlador correcto para manejarlo
2. Una vez le llega al controlador la solicitud, este se va a encargar de comunicarse con el Modelo para cualquier proceso que requiera de obtención y guardado de datos.
3. Una vez el controlador se ha comunicado con el modelo, el controlador procederá a comunicarse con la vista apropiada para generar una presentación resultante de los datos proporcionados por el modelo.
4. Finalmente, cuando esta presentación es generada se la envía de inmediato al usuario.



## 1.3. Desarrollo ágil de software

Se refiere a los métodos de ingeniería del software que están basados en el desarrollo iterativo e incremental, en el cual los requisitos y soluciones van evolucionando mediante la colaboración de equipos auto-organizados y multidisciplinares.

El ciclo de vida de cada iteración incluye: planificación, análisis de requisitos, diseño, codificación, pruebas y documentación. Los métodos ágiles se centran en las comunicaciones cara a cara en vez de la documentación.

Existen varios métodos ágiles de desarrollo software, algunos son: SCRUM, Kanban o Programación Extrema (XP) [7].

Se ha utilizado metodología ágil en vez de la metodología tradicional. Esta última se caracteriza por su rigidez ante los cambios, grupos de gran tamaño, mínimos roles, poco *feedback* o retroalimentación y seguimiento estricto del plan inicial de desarrollo [8].

### 1.3.1. SCRUM

SCRUM es una metodología de desarrollo ágil de software, se podría decir que es el proceso más destacado de este tipo de metodología. SCRUM, al ser metodología ágil, es un proceso sobre el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo.

Principalmente este proceso se caracteriza por: realizar constantes entregas parciales del producto final, presentar requisitos cambiantes o poco definidos, la innovación, competitividad, flexibilidad y la productividad. Por lo tanto SCRUM está especialmente indicado para proyectos en entornos complejos, pero también se utiliza en situaciones en las que no se está entregando al cliente lo que necesita [9].

Dentro de SCRUM existen varios roles [10]:

- **Scrum Master:** Es la persona encargada de que el proceso SCRUM se lleve a cabo como es debido, es decir, es la persona que lidera al equipo de desarrollo para que cumpla las reglas y procesos de la metodología.
- **Product Owner:** Representa al cliente, traslada la visión del proyecto al equipo haciéndole que trabaje correctamente para conseguir satisfacer las necesidades del cliente.
- **Equipo de desarrollo:** Grupo de profesionales que desarrollan el proyecto de manera conjunta, que disponen de los conocimientos técnicos necesarios.

SCRUM ha sido la metodología de desarrollo ágil utilizada. Por ello cada dos semanas se realizaba una reunión o iteración entre tutor y alumno.





En cada reunión se realizaban las siguientes tareas y en ese mismo orden:

1. Se evaluaban las tareas establecidas en la anterior reunión y se comprobaban que los resultados eran los esperados. En caso de que no lo fueran, se establecían mejoras a realizar.
2. Una vez revisado el punto anterior, se procedía a establecer las nuevas tareas a realizar para la próxima iteración.

## **2. Herramientas de software**

### **2.1. Lenguajes**

#### **2.1.1. HTML5**

El lenguaje HTML, siglas de *HyperText Markup Language*, se trata de un lenguaje de programación que se utiliza para el diseño de páginas webs de Internet. Se encarga de desarrollar una descripción sobre los contenidos que aparecen como textos y sobre su estructura, complementando dicho texto con diversos objetos: imágenes, vídeos, animaciones...

El lenguaje HTML es un lenguaje simple y general el cual se crea a partir de etiquetas o *tags* que aparecen especificadas a través de corchetes o paréntesis angulares < y > [11].

HTML5 es la última versión de HTML, y representa dos conceptos nuevos [12]:

- Contiene nuevos elementos, atributos y comportamientos.
- Contiene un conjunto más amplio de tecnologías que permite a los sitios web y a las aplicaciones ser más diversas y de gran alcance.

Una gran ventaja de HTML5 es que es posible adaptarlo a las pantallas de cualquier tipo de dispositivos (portátiles, tablets, móviles, etc).

Ha sido utilizado para realizar todo el diseño de la aplicación web.

Página web de la herramienta: <http://www.w3.org/TR/html5/>

#### **2.1.2. PHP**

El lenguaje PHP, siglas de *HyperText Preprocessor*, es un lenguaje de programación de uso general de código del lado del servidor, que se encuentra embebido en páginas HTML.

Algunas de las características de este lenguaje son las siguientes [13]:

- Es considerado un lenguaje fácil de aprender.
- Tiene capacidad para conectarse con la mayoría de los motores de base de datos.



- Lenguaje libre y abierto.
- Multitud de documentación, foros y ejemplos.
- No requiere definición de tipos de variables, permite técnicas de programación orientada a objetos y manejo de excepciones.

En la aplicación PHP se ha utilizado como lenguaje de trabajo principal en el servidor. Permite el acceso a la base de datos, el procesamiento y gestión de datos y la generación de páginas HTML que se envían al cliente.

Página web de la herramienta: <http://php.net/>

### **2.1.3. JavaScript**

Se trata de un lenguaje orientado a objetos, que se basa en prototipos, imperativo, débilmente tipado y dinámico.

Este lenguaje se utiliza principalmente en el lado del cliente implementado como parte de un navegador web, permitiendo así mejoras en la interfaz y páginas web dinámicas. Aunque sí que existe una forma de JavaScript del lado del servidor [14].

En el trabajo fin de grado se ha utilizado para la programación del lado del cliente.

### **2.1.4. AJAX**

AJAX, acrónimo de *Asynchronous JavaScript And Xml*, se trata de una técnica para el desarrollo web que va a permitir crear aplicaciones interactivas. Estas aplicaciones van a ejecutarse en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma, va a ser posible realizar cambios sobre las páginas sin que estas sean recargadas, de tal manera que se obtendrán mejoras de interactividad, velocidad y usabilidad de las aplicaciones [15].

Realmente AJAX no es una tecnología en sí mismo, sino que en realidad se trata de varias tecnologías independientes que se unen. Las tecnologías que forman AJAX son [16]:

- XHTML y CSS: para crear una presentación basada en estándares.
- DOM: para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON: para el intercambio y la manipulación de la información.
- XMLHttpRequest: para el intercambio asíncrono de información.
- JavaScript: para unir todas las demás tecnologías.

Por lo tanto ha sido utilizado para poder realizar cambios sobre páginas sin necesidad de recargarlas. En la aplicación se ha utilizado para hacer una llamada a un método de un controlador que va a encargarse de comprobar qué reportes existen para una práctica.





### 2.1.5. JQuery

Es una biblioteca de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

JQuery es software libre y de código abierto, y ofrece una funcionalidades basadas en JavaScript que de otra manera requeriría mucho más código, lo que significa que con las funciones de esta biblioteca se van a logran grandes resultados en menos tiempo y espacio.

Su característica principal es que permite cambiar el contenido de una página web sin necesidad de recargarla, mediante la manipulación del árbol DOM y las peticiones AJAX [17].

Se ha utilizado para poder manipular con mayor facilidad el árbol DOM.

Página web de la herramienta: <https://jquery.com/>

### 2.1.6. JSON

JSON, acrónimo de *JavaScript Object Notation*, es un formato ligero de intercambio de datos. Básicamente JSON describe los datos con una sintaxis dedicada que se usa para identificar y gestionar los datos. Una de las grandes ventajas que tiene el uso de JSON es que puede ser leído por cualquier lenguaje de programación, por lo que puede ser usado para el intercambio de información entre distintas tecnologías [18].

En la aplicación se ha utilizado para realizar el intercambio de *arrays*.

Página web de la herramienta: <http://www.json.org/>

### 2.1.7. Bootstrap

Bootstrap es un *framework* que permitir diseñar la interfaz de las aplicaciones web. Se basa en HTML, CSS y JavaScript, y contiene gran variedad de plantillas aplicadas a todo tipo de elementos de diseño: formularios, cuadros, botones, menús de navegación...

Su gran ventaja es que ofrece la posibilidad de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. A esta técnica de diseño y desarrollo se la conoce como *responsive design* o diseño adaptativo [19].

Se podría decir que sus principales características de Bootstrap son las siguientes:

- Permite diseño adaptativo.
- Diseños simples, limpios e intuitivos de crear.
- Multitud de tutoriales y comunidad activa.
- Compatible con la mayoría de los navegadores web.



Este *framework* ha permitido realizar todo el diseño de la interfaz de la aplicación.

Página web de la herramienta: <http://getbootstrap.com/>

### 2.1.8. XML

XML, que proviene de *eXtensible Markup Language* (lenguaje de marcas extensible), es un lenguaje basado en etiquetas y desarrollado por *World Wide Web Consortium* (W3C). Permite la organización y el etiquetado de diferentes documentos, es decir, no es un lenguaje en sí mismo sino un sistema que va a permitir definir lenguajes de acuerdo a las necesidades.

Presenta multitud de ventajas para los desarrolladores ya que permite relacionar aplicaciones escritas en diferentes lenguajes y pertenecientes a diferentes plataformas. Se caracteriza por su gran potencial, fácil uso e innegable utilidad [20].

Con una explicación breve, ha sido utilizado en la herramienta presentada para las siguientes tareas:

- Escritura en el fichero *pom.xml* perteneciente al arquetipo Maven creado.
- Lectura de los diferentes reportes xml generados para el almacenamiento de los datos en las correspondientes tablas de la base de datos.

Página web de la herramienta: <http://www.w3.org/XML/>

## 2.2. Herramientas

### 2.2.1. GitHub

Herramienta utilizada para alojar proyectos y llevar a cabo el control de versiones. El código es almacenado de forma pública pero también es posible hacerlo de forma privada, creando una cuenta de pago [21].

Ha sido elegido por ser muy intuitivo y fáciles de usar, además de haber sido utilizado en varias asignaturas impartidas en la carrera.

Página web de la herramienta: <https://github.com/>

### 2.2.2. XP-DEV

Herramienta utilizada para gestionar o manejar proyectos ágiles. Dispone de diferentes planes o versiones de pago pero la que se ha utilizado es la gratuita.

Dicha herramienta ha sido utilizada para realizar y gestionar la planificación ágil concretada entre el tutor y el alumno. Tras las reuniones acordadas con el tutor (cada 2 semanas) se añadía la iteración correspondiente a realizar, con sus tareas y horas estimadas.

A medida que se iban realizando las tareas se añadían las horas empleadas para ello y en caso de conseguir acabar la tarea, se marcaba como completa o finalizada.





Página web de la herramienta: <https://xp-dev.com/>

### 2.2.3. Moodle

Se trata de una plataforma o aplicación web de ámbito educativo, diseñada para proporcionar a los educadores y estudiantes un entorno común con el que puedan interactuar. Permite la creación de cursos en los que tanto profesores como alumnos podrán matricularse [3].

Una de sus principales características es el LTI, el cual nos ha permitido que nuestra aplicación sea conectada con la plataforma *Moodle*.

Dentro de los cursos existe la posibilidad de crear tareas o actividades de tipo herramienta externa o *external tool*, gracias a la cual introduciendo los valores *URL*, *LTI\_consumer* y *LTI\_secret* correctos podrá realizarse la conexión con nuestra aplicación web.

Por lo anteriormente citado y por ser la plataforma utilizada en *UBUVirtual* de la Universidad de Burgos, Moodle ha sido utilizado para poder comprobar el correcto funcionamiento del software generado durante este proceso.

Página web de la herramienta: <https://moodle.org/?lang=es>

### 2.2.4. WampServer

*WampServer* es un entorno de desarrollo web para el sistema operativo Windows que ofrece a los desarrolladores la posibilidad de crear aplicaciones web de manera local. Para ello utiliza Apache, PHP y una base de datos MySQL.

Esta herramienta ha sido utilizada para crear el servidor. Se caracteriza porque puede ser usado de forma libre, sin tener que usar ningún tipo de licencia [22].

Página web de la herramienta: <http://www.wampserver.com/en/>

### 2.2.5. CakePHP

Es un *framework* para el desarrollo de aplicaciones web escrito en PHP, creado sobre los conceptos de *Ruby on Rails*.

CakePHP es un marco de trabajo que facilita el desarrollo de aplicaciones web, utilizando el patrón de diseño MVC (Modelo Vista Controlador). Es de código abierto y se distribuye bajo licencia MIT. Algunas características son las siguientes [23]:

- Compatible con PHP4 y PHP5.
- Sistema de plantillas rápido y flexible.
- Ayudas para AJAX, JavaScript, HTML, *forms* y más.
- Validación integrada.





- Componentes de seguridad y sesión.

En el actual trabajo fin de grado ha sido utilizado para diseñar y programar toda la aplicación, y por lo tanto ha sido fundamental.

Página web de la herramienta: <http://cakephp.org/>

### 2.2.6. Maven

Es una herramienta de software para la gestión y construcción de proyectos Java. Es similar en funcionalidad a *Apache Ant*, pero tiene un modelo de configuración más simple, basado en un formato XML. Utiliza un fichero POM (Project Object Model) para describir el proyecto de software a construir, sus dependencias de otros módulos y componentes externos y el orden de construcción de los elementos [24].

Las partes del ciclo de vida principal del proyecto Maven son:

1. *Compile*: Genera los ficheros .class compilando los fuentes .java.
2. *Test*: Ejecuta los test automáticos de JUnit.
3. *Package*: Genera el fichero .jar con los .class compilados.
4. *Install*: Copia el fichero .jar a un directorio de nuestro ordenador.
5. *Deploy*: Copia el fichero .jar a un servidor remoto.

Esta herramienta es imprescindible para que funcione toda la aplicación. Ha sido utilizada para crear el arquetipo Maven en el que van a almacenarse los test subidos por el profesor y las prácticas de los alumnos. Además dentro del fichero de configuración *pom.xml* se incluyen todos los *plugins* necesarios para generar los reportes.

Página web de la herramienta: <https://maven.apache.org/>

### 2.2.7. Eclipse for PHP Developers

*Eclipse for PHP Developers* ha sido el IDE utilizado para realizar todo el código de la aplicación. Presenta una interfaz simple y fácil de manejar, además de multitud de opciones que han facilitado el desarrollo de la aplicación.

Página web de la herramienta:

<http://www.eclipse.org/downloads/packages/eclipse-php-developers/marsr>

### 2.2.8. MySQL

Es un sistema de gestión de bases de datos relacionales, multihilo y multiusuario, desarrollado como software libre, y ofrecido bajo la *GNU GPL* para cualquier uso compatible con esta licencia.





Algunas de sus características son las siguientes [25]:

- Amplio subconjunto del lenguaje SQL.
- Está disponible en gran cantidad de plataformas y sistemas.
- Posibilidad de transacciones y claves foráneas.

MySQL ha sido empleado para gestionar la base de datos principalmente por el hecho de que ya se encuentra integrado con WampServer y porque se integra perfectamente con el lenguaje PHP utilizado.

Página web de la herramienta: <https://www.mysql.com/>

### **2.2.9. HeidiSQL**

Es una herramienta diseñada para la gestión de las bases de datos MySQL locales y/o remotas. Ofrece una gran cantidad de funcionalidades que hacen que la gestión de tus bases de datos se convierta en una tarea asequible.

Es *opensource* y presenta una interfaz gráfica intuitiva y sencilla. Algunas de las funciones que ofrece *HeidiSQL* son las siguientes [26]:

- Consulta de las bases de datos y tablas.
- Exportación e importación de las tablas.
- Edición sencilla de tablas de datos.

Página web de la herramienta: <http://www.heidisql.com/>

### **2.2.10. Astah**

Herramienta gratuita utilizada para el modelado de diagramas UML. Fue creada por la compañía japonesa *Change Vision*.

Presenta diferentes versiones pero en nuestro caso se utilizó la versión gratuita *Astah Community* la cual nos ofrece funcionalidades como las siguientes [27]:

- Generación de diagramas de clase.
- Generación de diagramas de casos de uso.
- Generación de diagramas de actividad.
- Generación de diagramas de secuencia.

Página web de la herramienta: <http://astah.net/>



### 2.2.11. PoEdit

Ha sido utilizado para realizar la internacionalización de la aplicación, permitiendo generar los archivos .po con la traducción realizada.

La edición de estos ficheros resulta sencilla utilizando PoEdit. Simplemente hay que abrir el fichero .po que queramos, nos aparecerán todos los campos y la traducción que queramos realizar [28].

Es un editor libre, abierto y multiplataforma de catálogos *Gettext*.

Ha sido utilizado por los siguientes motivos:

- Es un programa sencillo de utilizar con una interfaz simple.
- Multiplataforma.
- Multitud de tutoriales.

Página web de la herramienta: <https://poedit.net/>

### 2.2.12. PHPUnit

Es un entorno para realizar pruebas unitarias en el lenguaje PHP. Permite crear y ejecutar test unitarios de manera sencilla y para ello usa *assertions* que van a ser los encargados de verificar que el comportamiento de una unidad de código es el esperado.

Los beneficios de realizar pruebas unitarias es que te permite aislar cada parte del programar y demostrar que estas partes de forma individual funcionan correctamente [29].

Los motivos por los que se ha utilizado:

- Viene integrado con **CakePHP**.
- En la web de CakePHP hay un apartado dedicado a PHPUnit en el que se explica el proceso que hay que seguir para usarlo, lo cual se convierte en algo sencillo y rápido.

Página web de la herramienta: <https://phpunit.de/>

### 2.2.13. FileSync

Herramienta que te permite sincronizar archivos y carpetas entre dos directorios diferentes.

Ha sido utilizada para mantener actualizado el código fuente que se encuentra alojado en el subdirectorio GitHub del directorio Documentos. Dado que la aplicación se encontraba alojada dentro del directorio “C:\wamp\www”, se ha utilizado *FileSync* para mantener el código actualizado de la aplicación alojada en “Documentos\GitHub”.

Página web de la herramienta: <http://www.freefilesync.org/>





## V - TRABAJOS RELACIONADOS

Para la realización de la aplicación se han tenido en cuenta proyectos similares, que a pesar de que no presentaban la funcionalidad que se buscaba, sí que es verdad que nos han ayudado en ciertos aspectos.

A continuación se van a mostrar las herramientas que nos han servido para mejorar o aclarar ideas en ciertos puntos de la aplicación.

### 1. *UBUVirtual*

UBUVirtual es un entorno dedicado a la enseñanza en el que tanto profesores como alumnos pueden acceder y compartir mismos cursos, tareas...

Desde esta plataforma los alumnos podrán realizar entregas de prácticas en las correspondientes tareas, y obtener las calificaciones correspondientes. Por otra parte, el profesor podrá llevar un control de las calificaciones obtenidas por cada uno de los alumnos.

#### 1.1. Similitudes con nuestra aplicación web

Esta herramienta presenta algunas similitudes con nuestra aplicación:

- En el formulario de subida de prácticas del alumno, era importante representar de alguna forma los parámetros de la tarea. A continuación se mostrará cómo hace la representación de estos parámetros UBUVirtual y cómo lo hace nuestra aplicación:

##### *UBUVirtual*

###### Estado de la entrega

Número del intento	Este es el intento 1 ( 1 intentos permitidos ).
Estado de la entrega	No entregado
Estado de la calificación	Sin calificar
Fecha de entrega	viernes, 15 de enero de 2016, 14:00
Tiempo restante	15 días 23 horas
Última modificación	martes, 1 de septiembre de 2015, 18:19
Comentarios de la entrega	► <a href="#">Comentarios (0)</a>

[Agregar entrega](#)

*Ilustración 5: Parámetros de la tarea en UBUVirtual*



### Nuestra aplicación web

#### Parámetros de la tarea

Nombre del paquete	es.ubu
Número máximo de intentos posibles	15
Fecha límite de entrega	2016-01-21
Número de intentos realizados	7
Número de intentos restantes	8

Cerrar

*Ilustración 6: Parámetros de la tarea en nuestra aplicación*

- Otra similitud entre ambos serían los formularios que ambos presentan para subir ficheros:

### UBUVirtual

Archivos enviados

Tamaño máximo para nuevos archivos: 100MB, número máximo de archivos adjuntos: 2

Archivos



Puede arrastrar y soltar archivos aquí para añadirlos

Guardar cambios

Cancelar

*Ilustración 7: Formulario de subida de ficheros en UBUVirtual*





### Nuestra aplicación web

Enunciado de la práctica

Fichero a subir:

Ningún archivo seleccionado

*Ilustración 8: Formulario de subida de ficheros en nuestra aplicación*

- En ambas se le permite al profesor la subida del enunciado de la práctica que deben realizar los alumnos. Además, este enunciado podrá visualizarse posteriormente en el panel de los alumnos.

## 2. SonarQube

Es una plataforma para evaluar el código fuente enfocada a permitir a los equipos de desarrollo profesionales realizar un seguimiento del QA. Es software libre y usa diversas herramientas de análisis estático de código fuente como Checkstyle, PMD o FindBugs para obtener métricas que pueden ayudar a mejorar la calidad del código de un programa.

Está preparada para poder incluir otras herramientas futuras por medio de una estructura de *plugins*.

En cuanto a sus funciones se encuentran las siguientes:

- Informa sobre código duplicado, pruebas unitarias, cobertura de código, complejidad ciclomática, posibles errores...
- Acepta muchos lenguajes de programación.

Cubre los 7 ejes de la calidad de código:

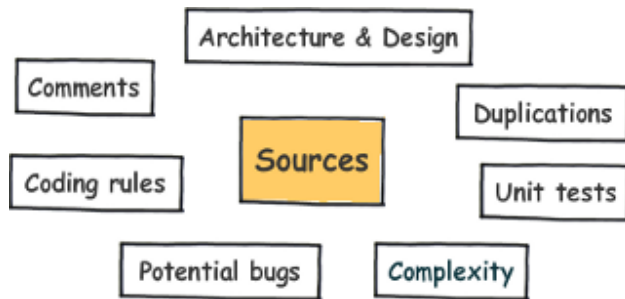


Ilustración 9: Ejes de la calidad de código en SonarQube

## 2.1. Similitudes con nuestra aplicación web

En cuanto a las similitudes que tiene con nuestra aplicación web, estarían las siguientes:

- Ambas utilizan los *plugins* PMD y FindBugs para analizar el código e informar de las posibles violaciones de código encontradas. En nuestra aplicación estos *plugins* están integrados dentro del *pom.xml* del arquetipo Maven, y la visualización de los reportes generados por estos *plugins* se puede visualizar tanto en el panel del alumno como en el del profesor. Dentro de la aplicación se habilitan los botones que se corresponden con estos reportes.
- Ambos dan información acerca de métricas como puede ser la complejidad ciclomática. En nuestra aplicación se utilizan los *plugins* JavaNCSS, ofrece métricas a nivel de clases (número de métodos, número de comentarios...) y JDepend, que ofrece métricas a nivel de paquete (*abstractness*, *instability*...). Dentro de la aplicación estas métricas pueden visualizarse pinchando en los botones habilitados para ello.
- Por último, tanto SonarQube como nuestra aplicación web permiten comprobar y visualizar la posible existencia de código duplicado. En nuestra aplicación web esta comprobación de plagios se realiza mediante el *plugin* JPlag, y permitiendo al profesor elegir entre qué alumnos quiere realizar dicha comprobación.

## 3. Laboratorio Virtual de Programación para Moodle (VPL)

VPL es una herramienta software de código abierto que permite la gestión de prácticas de programación en Moodle. Sus características más destacadas son:

- Posibilidad de editar el código fuente y ejecutar las prácticas de forma interactiva desde el navegador.
- Ejecutar pruebas que revisen las prácticas y analizar la similitud entre prácticas para el control de plagio.





- Está integrada en una plataforma de enseñanza electrónica, Moodle.
- Admite múltiples lenguajes de programación.
- Controla el plazo de entrega de las prácticas, control de acceso de usuarios basado en roles y establecimiento del número máximo de prácticas a entregar.
- El alumno puede recuperar en formato zip las prácticas entregadas.
- Permite editar los ficheros a entregar o los ya entregados desde el propio navegador utilizando el componente editor.
- En cuanto a la ejecución de la práctica, el sistema está diseñado para posibilitar la ejecución de programas desde el navegador. Esto se concreta en tres posibles acciones: ejecutar, depurar y evaluar. Estas tres acciones se pueden realizar desde la página de edición.

### 3.1. Similitudes con nuestra aplicación web

A continuación se muestran algunas de las similitudes que existen entre ambas aplicaciones:

- En el formulario de subida de prácticas habilitado para los alumnos, se puede observar la fecha límite de entrega y un enunciado de la práctica que deben de realizar:

#### VPL

Description	Submission	Edition
<h2>Hello World</h2> <p><b>Available from</b> Tuesday, 20 July 2010, 02:05 AM</p> <p><b>Due date</b> Saturday, 31 July 2010, 02:00 AM</p> <p><b>Requested files</b> (<a href="#">hello.c</a>)</p> <div>Write a C program that prints "Hello World!"</div>		

*Ilustración 10: Parámetros de la tarea en VPL*





### Nuestra aplicación web

- Parámetros de la tarea:

Se pueden observar pinchando el botón “Ver” presente al lado de la etiqueta parámetros de la tarea:

Parámetros de la tarea



*Ilustración 11: Ver parámetros de la tarea en nuestra aplicación*

Tras pincharlo aparece el siguiente panel:

Parámetros de la tarea ✕

Nombre del paquete	es.ubu
Número máximo de intentos posibles	15
Fecha límite de entrega	2016-01-21
Número de intentos realizados	7
Número de intentos restantes	8

Cerrar

*Ilustración 12: Parámetros de la tarea en nuestra aplicación*

- Enunciado de la práctica:

SUBIDA DE PRÁCTICAS ? 📄

ENUNCIADO DE LA PRÁCTICA

La práctica deberá de pertenecer al paquete 'es.ubu'. Dentro de este habrá 2 subpaquetes 'model' y 'controller'. Dentro del primero se deberá de crear un fichero 'Fichero.java' que contenga los métodos get() y set() necesarios. Dentro del segundo subpaquete se deberá de crear un fichero 'ontroller.java' que contenga los métodos 'obtenerDiezFicheros()' y 'obtenerCienFicheros()', los cuales devolverán una lista de Ficheros.

Parámetros de la tarea

Comentarios

Fichero a subir:

Ningún archivo seleccionado

*Ilustración 13: Enunciado de prácticas en nuestra aplicación*





- El formulario de subida de prácticas es muy similar ya que en ambos casos se permite la escritura de comentarios relacionados con la práctica, así como la opción para elegir la práctica a subir:

### VPL

*Ilustración 14: Formulario de subida de prácticas en VPL*

### Nuestra aplicación web

*Ilustración 15: Formulario de subida de prácticas en nuestra aplicación*

- En ambas aplicaciones se permite analizar la similitud entre prácticas para el control de plagio.
- Están integradas en una plataforma de enseñanza electrónica como es Moodle.



## **4. Autocorrección de Prácticas en Java para Moodle v2.0**

Se trata de un trabajo fin de carrera de la Ingeniería Técnica en Informática de Gestión. Proporciona al alumno una herramienta automática que proporcione una evaluación preliminar rápida y completa sobre una práctica de programación y facilita al profesor el seguimiento y la evaluación de las prácticas, ofreciéndole de forma automática diferentes medidas de la calidad del software entregado por los alumnos.

Permite tanto al alumno como al profesor, poder visualizar los resultados de todas las pruebas aplicadas a una determinada práctica realizada por un alumno, esta funcionalidad se podrá llevar a cabo gracias a un histórico de archivos donde el usuario elegirá cuál de los ficheros es el que desea visualizar.

La diferencia principal que presenta con el actual trabajo fin de grado es que el primero se desarrolló como un módulo dentro de Moodle, mientras que el segundo es una aplicación independiente completa y que además puede conectarse con otros LMS diferentes a Moodle.

### **4.1. Similitudes con nuestra aplicación web**

A continuación se detallan algunas de las similitudes existentes entre ambas aplicaciones:

- En cuanto al formulario de subida de ficheros y a la posterior visualización de los resultados tras el proceso de corrección de la práctica subida por el alumno:

#### **Autocorrección de prácticas en Java para Moodle 2.0**

Los resultados del proceso de corrección se muestran dentro del propio formulario.





Pepe García García  
Sunday, 12 de June de 2011, 02:24

Calificación: No hay calificación

Final grade: -

Trabaja en: 1 (0 p)

12/06/2011 01:30 - INTENTO 01 - Resultado de las pruebas ejecutadas:  
PRUEBAS UNITARIAS: 12 de 14 test pasados correctamente por MARIA VARGAS LLOSA  
PRUEBAS ESTÁTICAS: El alumno MARIA VARGAS LLOSA ha ejecutado las pruebas estáticas satisfactoriamente.  
Quedan 18 intentos. (1)

12/06/2011 02:24 - INTENTO 02 - Resultado de las pruebas ejecutadas:  
Pah:

☒ Send notification emails

Response files:

Prácticas compiladas y ejecutadas: ☒ TestResult-01.zip x ☒ TestResult-02.zip x (2)

Prácticas no compiladas: ☒ NoCompile-20110612-004330.zip x ☒ NoCompile-20110612-015223.zip x

MARIA VARGAS LLOSA  
Sunday, 12 de June de 2011, 03:59

☒ TestResult-02.zip x  (4)

Este alumno tiene 17 intentos totales restantes.  
0 intentos adicionales fueron concedidos.

Añadir intentos adicionales:  (3)

Terminado

Ilustración 16: Formulario subida de prácticas APJ para Moodle 2.0

### Nuestra aplicación web

En este caso se muestran los resultados mediante los reportes y las gráficas.

SUBIDA DE PRÁCTICAS

Parámetros de la tarea

Comentarios

Fichero a subir:

Ningún archivo seleccionado

Reportes del último intento realizado

Gráficas disponibles

[Violaciones](#) [Errores](#) [Violaciones - Errores](#) [Violaciones - Prioridades](#)

Ilustración 17: Formulario subida de ficheros y resultados en nuestra aplicación



- El histórico de los resultados de las prácticas puede visualizarse en ambos casos desde la tabla correspondiente:

**Autocorrección de prácticas en Java para Moodle 2.0**

Histórico de Resultados					
Intento	Realizado Por	Fichero de Resultados	Informe de Resultados		
1	Luis Enrique Medina García	TestResult-01.zip	Descargar informe	Ver Resultados	Ver Estadísticas
2	Luis Enrique Medina García	TestResult-02.zip	Descargar informe	Ver Resultados	Ver Estadísticas
3	Maitane Saez Alvarado	TestResult-03.zip	Descargar informe	Ver Resultados	Ver Estadísticas
4	Roberto Enzo Marconi	TestResult-04.zip	Descargar informe	Ver Resultados	Ver Estadísticas
Descargar informe general					

Ilustración 18: Estadísticas en APJ para Moodle 2.0

**Nuestra aplicación web**

Estadísticas de las prácticas de los alumnos

Mostrar

10

Buscar

intentos

▲	Nombre	Intento	Test pasado	Comentarios	Fecha subida	Práctica
●	Álvaro Vázquez	1	no	✖	1/18/16, 11:10 PM	<a href="#">practica1.zip</a>
●	Álvaro Vázquez	2	no	💬	1/18/16, 11:18 PM	<a href="#">practica1.zip</a>
●	Álvaro Vázquez	3	no	💬	1/18/16, 11:22 PM	<a href="#">practica1.zip</a>
●	Álvaro Vázquez	4	no	✖	1/18/16, 11:25 PM	<a href="#">practica1.zip</a>
●	Álvaro Vázquez	5	no	✖	1/18/16, 11:27 PM	<a href="#">practica1.zip</a>
●	Álvaro Vázquez	6	no	✖	1/18/16, 11:29 PM	<a href="#">practica1.zip</a>
●	Álvaro Vázquez	7	no	✖	1/21/16, 4:14 PM	<a href="#">practica1.zip</a>
●	Luis Izquierdo	1	no	✖	1/18/16, 11:35 PM	<a href="#">practica1.zip</a>
●	Luis Izquierdo	2	no	✖	1/18/16, 11:37 PM	<a href="#">practica1.zip</a>
●	Luis Izquierdo	3	si	✖	1/18/16, 11:38 PM	<a href="#">practica1.zip</a>

Mostrando 1 a 10 de 18 intentos

Primera

Anterior

1

2

Siguiente

Última

Reporte JavacCS

Reporte JDepend

Reporte PMD

Reporte FindBugs

Reporte Errores

Ilustración 19: Estadísticas en nuestra aplicación





## **VI - ASPECTOS RELEVANTES DEL DESARROLLO**

Cuando se afronta el desarrollo de un proyecto y dado que este va a realizarse durante varios meses, es importante analizar todas las dudas y problemas que pudieran surgir, de modo que se encuentren soluciones que se adapten en la mayor medida posible a las necesidades del proyecto.

En este apartado se van a comentar los problemas y dudas que han surgido durante el desarrollo así como las soluciones tomadas, explicando en cada caso los motivos que nos han llevado a la decisión final.

### **1. Utilizar LTI y desarrollar una aplicación web**

La elección de utilizar LTI para desarrollar nuestra aplicación y poder establecer una conexión con Moodle es un punto importante:

#### **Compatibilidad total:**

Utilizando LTI podemos asegurar que nuestra aplicación web funcione no sólo con Moodle sino con cualquier otro LMS que implemente LTI.

#### **Coste de acoplamiento reducido:**

Cuando Moodle establece la conexión con nuestra aplicación, esta obtendrá y guardará los parámetros que le proporciona Moodle (nombre de usuario, id's...).

Si se decide utilizar otro LMS en vez de Moodle y este no proporciona todos los parámetros que nuestra aplicación necesita, simplemente habría que adaptar el código del método de la aplicación encargado de recoger estos parámetros, por lo que no supondría un gran esfuerzo.

### **2. Seguridad de la aplicación**

La seguridad de una aplicación web siempre es un punto muy importante a controlar y a la hora de desarrollar una aplicación, y por ello se ha tenido muy en cuenta.

El acceso a nuestra aplicación web, tanto para profesores como para alumnos, únicamente va a ser posible desde las tareas de tipo “herramienta externa” creadas en Moodle. Es importante que los parámetros (URL, consumer\_key, secret) de esta tarea sean configurados correctamente por el profesor para que el acceso a la aplicación web sea posible.

Si por un casual se intenta acceder a la aplicación desde cualquier otro lugar, se mostrará un mensaje de error al intentar entrar en la aplicación. Además si un alumno ha accedido a la aplicación e intenta acceder al panel del profesor, este acceso será denegado mostrando el mensaje de error correspondiente. Y al igual para el profesor si intentase entrar en el panel del alumno.



### 3. *Estructura de directorios generada automáticamente en el servidor*

Aunque a primera vista no pueda observarse, se ha necesitado crear una importante estructura de directorios en el sistema operativo que alberga la parte servidor de la aplicación. Esta estructura permite la recogida de todos los datos que permiten el funcionamiento de toda la aplicación.

Es un trabajo que implica que PHP debe actuar sobre la estructura de ficheros de la máquina y ha resultado complejo. Se debe de hacer de manera precisa para que luego Maven pueda actuar sin problemas.

La creación de esta estructura de carpetas se produce en el momento que el profesor sube el primer test, y esta estructura tras el momento de su creación tiene un aspecto similar al siguiente:

```
id_curso > id_tarea > Profesor > id_profesor > arquetipo_maven
```

```
Alumno > id_alumno > número_intento_realizado
```

*Ilustración 20: Estructura de carpetas*

Esta estructura permite que cuando el profesor suba nuevos test, estos sean guardados dentro del arquetipo *maven* comprendido en las carpetas correspondientes al id del curso, tarea y profesor.

Además en el momento que un alumno sube una práctica, se crea su carpeta “id\_alumno” correspondiente y también se crea su carpeta “número\_intento\_realizado” correspondiente al número de intento de subida de práctica realizado. Dentro de esta última carpeta, se guardan tanto el zip de la práctica subida como los reportes generados (en formatos xml y html) por los diferentes *plugins*.

Si se quiere conocer en más detalle el proceso que hace Maven con esta estructura, consultar apartado correspondiente de los anexos (Ver *Anexos – Manual del programador: Procesamiento de los test, prácticas y generación de informes*).





#### **4. Elección de la forma de validar el consumer key**

El objetivo principal de las primeras semanas fue intentar conseguir una conexión entre Moodle y la aplicación web. Para ello se empleó una librería LTI [30], la cual se descargó y se realizó una primera investigación de la misma.

Dicha librería en principio, permite llevar a cabo una comparación de los *consumer\_key* y *secret*, que se establecen en la tarea creada en Moodle, con los valores *consumer\_key* y *secret* correctos que se encuentran almacenados en la base de datos. Si dichos valores coinciden, entonces la conexión entre Moodle y la aplicación web puede realizarse correctamente, en caso contrario no.

Sin embargo, durante las pruebas iniciales de la utilización de esta librería han surgido problemas al realizar la comparación del *consumer\_key*, principalmente debido a que la librería utilizaba métodos de acceso a la base de datos que ya estaban obsoletos y otros fragmentos de código que no se conseguían entender correctamente.

Por ello, este proceso se realizará de otra forma (Ver Anexos – *Manual del programador: Conexión LTI*).

#### **5. Elección del framework CakePHP**

Desarrollar una aplicación web a partir de un patrón que facilite el desarrollo y que permita una mejor organización de la misma siempre es un punto a tener en cuenta.

El patrón MVC (Modelo Vista Controlador) es uno de los más utilizados para desarrollar aplicaciones web, por lo que siempre estuvo desde el primer momento en consideración. De esta manera se pensó en el *framework* de CakePHP el cual implementa dicho patrón y genera una estructura de carpetas por defecto acorde al patrón, para que los controladores, vistas... se creen en la carpeta adecuada.

El desconocimiento del lenguaje PHP y del *framework* CakePHP siempre fueron un problema en los inicios del desarrollo del trabajo fin de grado, pero con el tiempo se consiguió adquirir el conocimiento necesario para poder desarrollar la aplicación con CakePHP.

#### **6. Elección de Maven**

Dado que la funcionalidad más destacada de la aplicación consiste principalmente en un proceso de análisis de código de las prácticas subidas por los alumnos, a las cuales iban a aplicarse los test subidos por el profesor, era necesaria una herramienta que permitiera gestionar este tipo de tareas. Por ello como posibles opciones había dos: Ant y Maven.





La decisión de elegir Maven frente a Ant fue principalmente porque se tenían más conocimiento sobre Maven, gracias a que en la asignatura “Sistemas distribuidos” impartida en la carrera, se realizaron varias aplicaciones que utilizaban dicha herramienta de software.

Otro factor que ayudó a tomar tal decisión fue que Maven es más actual que Ant.

## **7. Elección de Bootstrap**

Desarrollar la interfaz gráfica de una aplicación sin usar ninguna librería o *framework* siempre puede convertirse en un proceso complicado.

Por ello se seleccionó el *framework* de Bootstrap, el cual ha sido utilizado en todo momento para desarrollar la interfaz de la aplicación.

Gracias a Bootstrap, desarrollar la interfaz de una aplicación se convierte en un proceso más sencillo gracias a los múltiples componentes y plantillas que integra por defecto, así como los ejemplos de aplicaciones presentes en su web oficial.

## **8. Elección de los plugins para los reportes**

Una duda a resolver, fue la de qué reportes se le iban a mostrar a los alumnos cada vez que realizaban un intento de subida de práctica.

Para obtener un abanico lo más amplio de análisis estático sobre código posible, los *plugins* que se seleccionaron para generar estos reportes fueron: JAVANCSS, JDEPEND, PMD y FINDBUGS (Ver *Anexos – Manual del programador: Instalación de herramientas: Plugins Reportes*).

## **9. Elección de la forma de añadir los plugins**

Dado que los *plugins* hay que añadirles dentro del fichero “pom.xml” del arquetipo Maven, había que elegir la forma con la que se iba a realizar este proceso.

Por ello la decisión tomada fue la de crear un método “editarPomArquetipoMaven()” dentro del controlador “FicherosXmlController”, de forma que cuando el profesor suba su test por primera vez y se cree el arquetipo Maven, se llame a dicho método el cual se va a encargar de:

1. Abrir el fichero pom.xml, para recorrerlo y leer sus datos, con la función correspondiente de php.
2. Crear las etiquetas correspondientes a cada uno de los *plugins*.
3. Añadir dichas etiquetas al pom.xml y guardarlo.





Además del método anteriormente mencionado para realizar la edición del fichero “pom.xml”, dentro del controlador “FicherosXmlController” se van a tener diferentes métodos “guardar...()” encargados de leer los reportes xml generados por los plugins para guardar sus datos en la correspondiente tabla de la base de datos.

Por lo tanto si en un futuro se quisiera añadir un nuevo *plugin*, en primer lugar sería necesario editar el método “editarPomArquetipoMaven()” para añadir el nuevo *plugin* y después sería necesaria la creación de un nuevo método “guardar...()” encargado de leer los datos del fichero xml generado para guardarles en la base de datos.

Por último también sería necesario añadir un botón para el nuevo plugin en el formulario de subida de prácticas del alumno y en la tabla de estadísticas del profesor.

## **10. Elección de la herramienta para los plagios**

Dado que se le quería añadir una opción al profesor para que pudiera comprobar si había plagios entre las prácticas subidas por los alumnos, era necesaria una herramienta para ello.

Por ello la opción que se comenzó a utilizar al principio fue la del *plugin* CPD de Maven. Sin embargo dado que los reportes que generaba no eran muy claros y además tampoco te mostraba indicadores de los porcentajes de plagios, se optó por buscar otra herramienta.

Por ello se investigó acerca de MOSS y JPLAG, que finalmente se acabó utilizando JPLAG ya que era fácil de implementar y además los reportes eran claros y sencillos.



## VII - CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURAS

### 1. Conclusiones

En primer lugar, considero que se han cumplido la mayoría de objetivos que fueron establecidos inicialmente, al igual que los que se han ido proponiendo a lo largo del desarrollo del trabajo fin de grado.

La aplicación web proporciona la funcionalidad necesaria para que los alumnos puedan subir sus prácticas y obtener los resultados resultantes tras el proceso de análisis de la misma, así como los profesores establecer los test que deseen y poder tener un seguimiento de cada una de las prácticas subidas por los alumnos.

El nulo conocimiento del lenguaje PHP que se tenía en los inicios y los escasos conocimientos que se tenían de lenguajes como JavaScript y otros utilizados, se han convertido en conocimientos más avanzados que han sido adquiridos gracias a la realización del presente trabajo fin de grado. Al igual que con los *frameworks* de CakePHP y Bootstrap, donde la situación era similar.

La utilización del patrón MVC (Modelo Vista Controlador) en la presente aplicación ha sido fundamental para poder adquirir un mayor conocimiento del patrón de forma práctica, además de haber permitido que el código de la aplicación quedara más organizado y la realización de la misma haya sido más sencilla y entendible.

La utilización del *framework* CakePHP ha permitido familiarizarse con una nueva sintaxis simplificada para las funciones utilizadas para realizar la manipulación de datos de la base de datos, ya que por ejemplo para realizar un SELECT se utilizaba el FIND.

Se ha aprendido a realizar una conexión entre dos servicios/aplicaciones web como pueden ser Moodle (y LMS similares) y la aplicación web construida, gracias al estándar LTI el cual se desconocía. Sin duda este ha supuesto uno de los mayores esfuerzos a la hora de desarrollar la aplicación.

También he mejorado mis conocimientos en metodologías ágiles y en las herramientas que se han ido utilizando día a día mientras se desarrollaba el trabajo fin de grado.

Por último, realizar la documentación de la memoria y anexos me ha ayudado a aprender a realizar documentos más técnicos y con una organización más clara y acorde a los apartados necesarios, así como la realización de diagramas y la gestión de citas bibliográficas.





## 2. *Líneas de trabajo futuras*

Como consecuencia de la falta de tiempo, se han quedado algunas funcionalidades sin poder implementar que podrían ser interesantes para añadir en un futuro. Serían las siguientes:

- **Recepción automática de notificaciones:** Que consistiera en que cuando un profesor realiza una acción esto se le sea notificado a los alumnos con una notificación o mensaje automático que al acceder a la aplicación pudieran verlo, y a la inversa también. Es decir, que si por ejemplo el profesor ha subido un nuevo test o ha actualizado el enunciado de la tarea, tras este proceso se genere el envío de una notificación automáticamente que los alumnos al acceder a la aplicación podrán observar. De la misma manera, si un alumno sube una nueva práctica que se genere la notificación para que al profesor le aparezca al entrar en la aplicación.
- **Descarga de prácticas y reportes:** Como funcionalidad para el profesor sería crear una opción para que pudiera descargarse todas las prácticas subidas por los alumnos en un único zip y también todos los reportes. Y al alumno darle la opción de poder descargarse todos los reportes relacionados con sus propias entregas en un único zip.
- **Nuevos lenguajes de programación:** Para que la aplicación pueda corregir prácticas que no solo estén escritas en el lenguaje java, sino que de la posibilidad de corregir prácticas realizadas en otros lenguajes de programación.
- **Habilitar editor de código:** Habilitar un editor de código en el panel del alumno para que este pudiera realizar o editar la práctica desde la propia aplicación.
- **Multiplataforma:** Programar la aplicación de forma que pueda ser utilizada desde dispositivos que no solo usen Windows.
- **Mejorar tiempo de análisis de las prácticas:** Mejorar el tiempo de análisis del proceso de corrección de las prácticas subidas por los alumnos. Por ejemplo, permitiéndole elegir los reportes que quiere generar.
- **Hacer más flexible la presentación de resultados:** Relacionado con lo anterior, que se puedan seleccionar con un grado más fino las estadísticas que se quieren ver/recoger o permitir mostrar a los alumnos por parte del profesor.
- **Devolución de calificaciones a la plataforma LMS:** Que aplicación devuelva a Moodle o la plataforma LMS utilizada algún tipo de información en forma de calificación a cada entrega.



## VIII - BIBLIOGRAFÍA

- [1] «Sistema de gestión de aprendizaje», *Wikipedia, la enciclopedia libre*. 21-nov-2015.
- [2] «LMS-Moodle». [En línea]. Disponible en:  
<http://ares.cnice.mec.es/informes/16/contenido/47.htm>.
- [3] «Moodle», *Wikipedia, la enciclopedia libre*. 04-dic-2015.
- [4] S. com-agosto 11th, 2014 at 17:02 Sin duda hay una diferencia gigante entre Asegurar la Calidad de un producto, y C. T. vendría a ser la parte reactiva del proceso, y y Q. la parte proactiva G. artículo! Saludos!, «Quality Assurance no es sólo Testing», *Softqanetwork.com*, 18-abr-2007. [En línea]. Disponible en:  
<http://www.softqanetwork.com/quality-assurance-no-es-solo-testing>. [Accedido: 19-ene-2016].
- [5] «Quality Assurance». [En línea]. Disponible en: <http://www.4rsoluciones.com/que-es-un-plan-de-qa/>.
- [6] «Modelo Vista Controlador», *CookBook*. [En línea]. Disponible en:  
<http://book.cakephp.org/2.0/es/cakephp-overview/understanding-model-view-controller.html>.
- [7] «Desarrollo ágil de software», *Wikipedia, la enciclopedia libre*. 10-dic-2015.
- [8] R. Dario, «Mantenimiento de computadores: Metodologías de Desarrollo Ágiles Vs. Metodologías Tradicionales», *Mantenimiento de computadores*, 20-mar-2014. .
- [9] «Qué es SCRUM», *Proyectos Ágiles*. .
- [10] «Proceso y Roles de Scrum». [En línea]. Disponible en: <https://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum/proceso-roles-de-scrum.html>. [Accedido: 18-dic-2015].
- [11] «Definición de html — Definicion.de», *Definición.de*. [En línea]. Disponible en:  
<http://definicion.de/html/>. [Accedido: 12-dic-2015].
- [12] «HTML5», *Mozilla Developer Network*. [En línea]. Disponible en:  
<https://developer.mozilla.org/es/docs/HTML/HTML5>. [Accedido: 12-dic-2015].
- [13] «PHP», *Wikipedia, la enciclopedia libre*. 07-dic-2015.
- [14] «JavaScript», *Wikipedia, la enciclopedia libre*. 09-ene-2016.
- [15] «AJAX», *Wikipedia, la enciclopedia libre*. 02-dic-2015.
- [16] «Capítulo 1. Introducción a AJAX (Introducción a AJAX)». [En línea]. Disponible en:  
[http://librosweb.es/libro/ajax/capitulo\\_1.html](http://librosweb.es/libro/ajax/capitulo_1.html). [Accedido: 13-ene-2016].
- [17] «jQuery», *Wikipedia, la enciclopedia libre*. 21-dic-2015.
- [18] «JSON I - ¿Qué es y para qué sirve JSON?», *Geeky Theory*. [En línea]. Disponible en:  
<https://geekytheory.com/json-i-que-es-y-para-que-sirve-json/>. [Accedido: 13-ene-2016].
- [19] «¿Qué es Bootstrap y cómo funciona en el diseño web?», *Chucherías*. .
- [20] «Definición de XML — Definicion.de», *Definición.de*. [En línea]. Disponible en:  
<http://definicion.de/xml/>. [Accedido: 15-dic-2015].
- [21] «GitHub», *Wikipedia, la enciclopedia libre*. 15-nov-2015.
- [22] «PHP: WampServer Definicion, Instalación y configuración». .
- [23] «CakePHP», *Wikipedia, la enciclopedia libre*. 23-ago-2015.
- [24] «Maven», *Wikipedia, la enciclopedia libre*. 28-ene-2016.





- [25] «MySQL», *Wikipedia, la enciclopedia libre*. 26-nov-2015.
- [26] «HeidiSQL, la gestión de tus bases de datos a golpe de clic». [En línea]. Disponible en: <http://empresayeconomia.republica.com/aplicaciones-para-empresas/heidisql-la-gestion-de-tus-bases-de-datos-a-golpe-de-clic.html>. [Accedido: 17-dic-2015].
- [27] «Astah\*», *Wikipedia, the free encyclopedia*. 02-sep-2015.
- [28] P. Santamaría, «PoEdit, edición de archivos .po y .mo desde OS X», *Applesfera*, 26-nov-2012. [En línea]. Disponible en: <http://www.applesfera.com/aplicaciones-os-x-1/poedit-edicion-de-archivos-po-y-mo-desde-os-x>. [Accedido: 22-ene-2016].
- [29] «PHPUnit», *Wikipedia, la enciclopedia libre*. 08-dic-2015.
- [30] Andy Smith, *Librería IMS-BLTI*. .



Impreso en Burgos el jueves, 4 de febrero de 2016