



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Gº en Ingeniería en Informática



Trabajo final de GºIng.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



Índice de contenido

I -	Plan de proyecto software	8
1.	Introducción	8
2.	Planificación temporal del proyecto	8
2.1.	Iteración 1 (25 sept 2015 a 1 oct 2015)	9
2.2.	Iteración 2 (1 oct 2015 a 15 oct 2015)	11
2.3.	Iteración 3 (15 oct 2015 a 29 oct 2015)	12
2.4.	Iteración 4 (29 oct 2015 a 12 nov 2015)	13
2.5.	Iteración 5 (12 nov 2015 a 26 nov 2015)	15
2.6.	Iteración 6 (26 nov 2015 a 10 dic 2015)	17
2.7.	Iteración 7 (10 dic 2015 a 24 dic 2015)	19
2.8.	Iteración 8 (24 dic 2015 a 7 ene 2016)	20
2.9.	Iteración 9 (7 ene 2016 a 21 ene 2016)	22
2.10.	Iteración 10 (21 ene 2016 a 4 feb 2016)	23
3.	Estudio de viabilidad	25
3.1.	Estudio de viabilidad legal	25
3.2.	Estudio de viabilidad económica	26
II -	Especificación de requisitos	31
1.	Introducción	31
2.	Requisitos funcionales	31
2.1.	Propios de Moodle	31
2.2.	Propios de la aplicación web	31
2.3.	Requisitos de restricción	33
3.	Requisitos no funcionales	33
4.	Diagrama de casos de uso	34
4.1.	Plantillas de casos de uso	37
III -	Especificación de diseño	52
1.	Diseño de datos	52
1.1.	Base de datos	52
1.2.	Diagrama de paquetes	59



1.3.	Diagrama de clases	61
2.	Diseño procedimental.....	63
2.1.	Diagramas de secuencias	63
3.	Diseño arquitectónico.....	68
3.1.	Diagrama de despliegue.....	68
IV -	Manual del programador	69
1.	Introducción	69
2.	Estructura de directorios.....	69
3.	Estructura del código fuente.....	69
4.	Instalación de herramientas	76
4.1.	Sistema operativo	76
4.2.	Servidor	76
4.3.	Base de datos	78
4.4.	Java JDK	80
4.5.	Maven.....	81
4.6.	Plugins Reportes.....	81
4.7.	CakePHP	82
4.8.	Moodle	83
4.9.	Plugin LTI	85
4.10.	PHPUnit	85
4.11.	Instalación de un IDE	85
5.	Conexión LTI.....	86
6.	Procesamiento de los test, prácticas y generación de informes	87
7.	Pruebas	88
7.1.	Ejecutar pruebas	88
7.2.	Pruebas unitarias.....	90
7.3.	Pruebas de cobertura de código	92
V -	Manual del usuario.....	93
1.	Manual del profesor	93
1.1.	Registrar profesor en la aplicación	93
1.2.	Panel principal	98



1.3.	Panel Configuración	98
1.4.	Panel Subida	99
1.5.	Panel Estadísticas	100
1.6.	Panel Gráficas	101
1.7.	Panel Plagios	102
2.	Manual del alumno.....	104
2.1.	Aspecto Inicial.....	104
2.2.	Información y Enunciado	105
2.3.	Parámetros de la tarea.....	106
2.4.	Subir práctica.....	107
2.5.	Ver reportes y gráficas.....	108
3.	Cambiar Idioma	108
VI -	Bibliografía	109





Índice de Ilustraciones

ILUSTRACIÓN 1: RESUMEN GLOBAL DE LA PLANIFICACIÓN DEL TFG.....	9
ILUSTRACIÓN 2: SPRINT BACKLOG ITERACIÓN 1	10
ILUSTRACIÓN 3: BURNDOWN CHART ITERACIÓN 1	10
ILUSTRACIÓN 4: SPRINT BACKLOG ITERACIÓN 2	11
ILUSTRACIÓN 5: BURNDOWN CHART ITERACIÓN 2	12
ILUSTRACIÓN 6: SPRINT BACKLOG ITERACIÓN 3	13
ILUSTRACIÓN 7: BURNDOWN CHART ITERACIÓN 3	13
ILUSTRACIÓN 8: SPRINT BACKLOG ITERACIÓN 4	14
ILUSTRACIÓN 9: BURNDOWN CHART ITERACIÓN 4	15
ILUSTRACIÓN 10: SPRINT BACKLOG ITERACIÓN 5	16
ILUSTRACIÓN 11: BURNDOWN CHART ITERACIÓN 5	16
ILUSTRACIÓN 12: SPRINT BACKLOG ITERACIÓN 6	18
ILUSTRACIÓN 13: BURNDOWN CHART ITERACIÓN 6	18
ILUSTRACIÓN 14: SPRINT BACKLOG ITERACIÓN 7	19
ILUSTRACIÓN 15: BURNDOWN CHART ITERACIÓN 7	20
ILUSTRACIÓN 16: SPRINT BACKLOG ITERACIÓN 8	21
ILUSTRACIÓN 17: BURNDOWN CHART ITERACIÓN 8	21
ILUSTRACIÓN 18: SPRINT BACKLOG ITERACIÓN 9	22
ILUSTRACIÓN 19: BURNDOWN CHART ITERACIÓN 9	23
ILUSTRACIÓN 20: SPRINT BACKLOG ITERACIÓN 10.....	24
ILUSTRACIÓN 21: BURNDOWN CHART ITERACIÓN 10	24
ILUSTRACIÓN 22: DCU - COLORES.....	34
ILUSTRACIÓN 23: DCU - GENERAL DEL SISTEMA	35
ILUSTRACIÓN 24: DCU - PROFESOR SELECCIONAR TAREA	35
ILUSTRACIÓN 25: DCU - ALUMNO SELECCIONAR TAREA	36
ILUSTRACIÓN 26: MODELO RELACIONAL	52
ILUSTRACIÓN 27: DIAGRAMA DE PAQUETES	59



ILUSTRACIÓN 28: DIAGRAMA DE CLASES	61
ILUSTRACIÓN 29: DIAGRAMA DE SECUENCIA "CONEXIÓN DEL USUARIO"	63
ILUSTRACIÓN 30: DIAGRAMA DE SECUENCIA "REGISTRO DEL PROFESOR EN LA APLICACIÓN" ...	64
ILUSTRACIÓN 31: DIAGRAMA DE SECUENCIA "PROFESOR CONFIGURA PARÁMETROS TAREA" ...	64
ILUSTRACIÓN 32: DIAGRAMA DE SECUENCIA "PROFESOR SUBE TEST"	65
ILUSTRACIÓN 33: DIAGRAMA DE SECUENCIA "ALUMNO SUBE PRÁCTICA"	66
ILUSTRACIÓN 34: DIAGRAMA DE SECUENCIA "ALUMNO SUBE PRÁCTICA INCORRECTA"	67
ILUSTRACIÓN 35: DIAGRAMA DE DESPLIEGUE.....	68
ILUSTRACIÓN 36: DIRECTORIO RAÍZ DE LA APLICACIÓN.....	70
ILUSTRACIÓN 37: DIRECTORIO "CONFIG" DE LA APLICACIÓN	70
ILUSTRACIÓN 38: DIRECTORIO "FICHEROS_TEST" DE LA APLICACIÓN	71
ILUSTRACIÓN 39: DIRECTORIO "SRC/CONTROLLER" DE LA APLICACIÓN	71
ILUSTRACIÓN 40: DIRECTORIO "SRC/TEMPLATE" DE LA APLICACIÓN	72
ILUSTRACIÓN 41: DIRECTORIO "SRC/MODEL" DE LA APLICACIÓN.....	73
ILUSTRACIÓN 42: DIRECTORIO "SRC/LOCALE" DE LA APLICACIÓN	74
ILUSTRACIÓN 43: DIRECTORIO "VENDOR" DE LA APLICACIÓN.....	75
ILUSTRACIÓN 44: DIRECTORIO "WEBROOT" DE LA APLICACIÓN	76
ILUSTRACIÓN 45: INSTALAR WAMP SERVER PASO 1	77
ILUSTRACIÓN 46: INSTALAR WAMP SERVER PASO 2	77
ILUSTRACIÓN 47: INSTALAR WAMP SERVER PASO 3	78
ILUSTRACIÓN 48: HEIDI SQL CREACIÓN DE TABLAS	79
ILUSTRACIÓN 49: JAVA JDK DESCARGA.....	80
ILUSTRACIÓN 50: MAVEN DESCARGA.....	81
ILUSTRACIÓN 51: CAKEPHP INSTALACIÓN	82
ILUSTRACIÓN 52: MOODLE INSTALACIÓN SELECCIONAR IDIOMA	83
ILUSTRACIÓN 53: MOODLE INSTALACIÓN AJUSTE BASE DE DATOS.....	84
ILUSTRACIÓN 54: MOODLE INSTALACIÓN DATOS ADMIN.....	84





ILUSTRACIÓN 55: SALIDA CMD OBTENIDA TRAS EJECUTAR PRUEBAS UNITARIAS	88
ILUSTRACIÓN 56: SALIDA CMD OBTENIDA TRAS EJECUTAR PRUEBAS DE COBERTURA	89
ILUSTRACIÓN 57: RESULTADO DEL TEST DE COBERTURA.....	89
ILUSTRACIÓN 58: RESULTADO DE TEST DE COBERTURA CARPETA "SRC"	90
ILUSTRACIÓN 59: COBERTURA DE CÓDIGO DE LOS CONTROLADORES	92
ILUSTRACIÓN 60: FORMULARIO DE REGISTRO DEL PROFESOR	93
ILUSTRACIÓN 61: PARÁMETROS LTI OBTENIDOS	94
ILUSTRACIÓN 62: MOODLE ASPECTO CURSO	94
ILUSTRACIÓN 63: MOODLE SELECCIÓN HERRAMIENTA EXTERNA	95
ILUSTRACIÓN 64: MOODLE CONFIGURACIÓN PARÁMETROS DE LA TAREA PASO 1.....	96
ILUSTRACIÓN 65: MOODLE CONFIGURACIÓN PARÁMETROS DE LA TAREA PASO 2.....	97
ILUSTRACIÓN 66: MOODLE TAREA CREADA	97
ILUSTRACIÓN 67: PANEL PRINCIPAL DEL PROFESOR DE LA APLICACIÓN.....	98
ILUSTRACIÓN 68: PANEL DEL PROFESOR DE CONFIGURACIÓN DE PARÁMETROS DE LA TAREA ...	99
ILUSTRACIÓN 69: PANEL DEL PROFESOR DE SUBIDA DE TEST	99
ILUSTRACIÓN 70: PANEL DEL PROFESOR DE ESTADÍSTICAS	100
ILUSTRACIÓN 71: PANEL DEL PROFESOR DE ESTADÍSTICAS CONSULTA DE REPORTES	101
ILUSTRACIÓN 72: PANEL DEL PROFESOR DE GRÁFICAS	101
ILUSTRACIÓN 73: PANEL DEL PROFESOR DE SELECCIÓN DE GRÁFICAS	102
ILUSTRACIÓN 74: PANEL DEL PROFESOR DE COMPROBACIÓN DE PLAGIOS	103
ILUSTRACIÓN 75: PANEL DEL PROFESOR DE SELECCIÓN DE PLAGIOS	103
ILUSTRACIÓN 76: PANEL PRINCIPAL DEL ALUMNO	104
ILUSTRACIÓN 77: PANEL DEL ALUMNO VISUALIZACIÓN DE GRÁFICAS Y REPORTES	105
ILUSTRACIÓN 78: PANEL DEL ALUMNO INFORMACIÓN Y ENUNCIADO	105
ILUSTRACIÓN 79: PANEL DEL ALUMNO PARÁMETROS DE LA TAREA	106
ILUSTRACIÓN 80: PANEL DEL ALUMNO ERROR SUBIDA DE PRÁCTICA	107
ILUSTRACIÓN 81: PANEL DEL ALUMNO VER REPORTES Y GRÁFICAS.....	108
ILUSTRACIÓN 82: CAMBIAR IDIOMA.....	108



Índice de Tablas

TABLA 1: COSTES DE SOFTWARE	28
TABLA 2: PLANTILLA TIPO CASOS DE USO	37
TABLA 3: PLANTILLA RF-M01 "CREAR LA TAREA"	38
TABLA 4: PLANTILLA RF-M02 "CONFIGURAR LA TAREA"	39
TABLA 5: PLANTILLA RF-M03 "SELECCIONAR LA TAREA"	40
TABLA 6: PLANTILLA RF-AW01 "REGISTRARSE EN LA APLICACIÓN WEB"	41
TABLA 7: PLANTILLA RF-AW02 "CONFIGURAR LOS PARÁMETROS DE LA TAREA"	42
TABLA 8: PLANTILLA RF-AW03 "SUBIR TEST"	43
TABLA 9: PLANTILLA RF-AW04 "CONSULTAR ESTADÍSTICAS DE LAS PRÁCTICAS"	44
TABLA 10: PLANTILLA RF-AW05 "DESCARGAR PRÁCTICAS"	45
TABLA 11: PLANTILLA RF-AW06 "COMPROBAR PLAGIOS DE PRÁCTICAS"	46
TABLA 12: PLANTILLA RF-AW07 "VISUALIZAR GRÁFICAS"	47
TABLA 13: PLANTILLA RF-AW08 "SUBIR PRÁCTICA"	48
TABLA 14: PLANTILLA RF-AW09 "CONSULTAR DATOS DE LA TAREA"	49
TABLA 15: PLANTILLA RF-AW10 "REEMPLAZAR LA PRÁCTICA SELECCIONADA"	50
TABLA 16: PLANTILLA RF-AW11 "CONSULTAR REPORTES"	50
TABLA 17: PLANTILLA RF-AW12 "CONSULTAR GRÁFICAS"	51
TABLA 18: TABLA PROFESORES	53
TABLA 19: TABLA ALUMNOS	53
TABLA 20: TABLA TAREAS	54
TABLA 21: TABLA TESTS	55
TABLA 22: TABLA INTENTOS	56
TABLA 23: TABLA VIOLACIONES	57
TABLA 24: TABLA ERRORES	58
TABLA 25: TABLA PRUEBAS UNITARIAS	91





I - PLAN DE PROYECTO SOFTWARE

1. *Introducción*

En este apartado se tratará la planificación temporal del proyecto, que se refiere a una estimación dentro de un marco de tiempo limitado en la que se recogerán todas las tareas necesarias y sus plazos, para obtener el resultado final esperado.

Por lo tanto, la planificación temporal será una referencia con la que se podrá estimar el estado del proyecto en cada momento.

En este apartado se hablará también sobre la viabilidad económica del proyecto, que dará una perspectiva acerca de la rentabilidad o no del desarrollo del proyecto. Además del estudio de viabilidad legal, sobre el cuál se debe estudiar la clasificación de las distintas licencias software y sus características.

2. *Planificación temporal del proyecto*

El trabajo fin de grado se ha desarrollado en base a los principios de las metodologías ágiles, SCRUM concretamente.

El procedimiento seguido a la hora de planificar, es el de distribuir el trabajo en iteraciones de dos semanas, llevando a cabo para ello reuniones con el tutor. Al finalizar cada iteración se decidirá, entre tutor y alumno, las tareas a incluir en la siguiente iteración, intentando maximizar en todo momento el valor del proyecto.

Las tareas seleccionables en cada iteración residen en el *ProductBacklog*, que constituye una lista de tareas priorizadas.

Dentro de las metodologías ágiles, uno de los indicadores de productividad más comunes es la velocidad. Hace referencia a la cantidad de trabajo (historias de usuario) que se realiza en una iteración. Aquí cobran especial importancia los gráficas *BurnDown*.

Los gráficos *BurnDown* son una representación del trabajo que queda por hacer y a su vez, permiten comparar el progreso del proyecto o del *Sprint* con la planificación inicial. Normalmente el *Sprint Backlog* (tareas a realizar en una iteración) se muestra en el eje vertical y los días hábiles de los que se compone el Sprint en el eje horizontal.

En el siguiente gráfico, se resumen todas las iteraciones llevadas a cabo durante el trabajo de fin de grado:

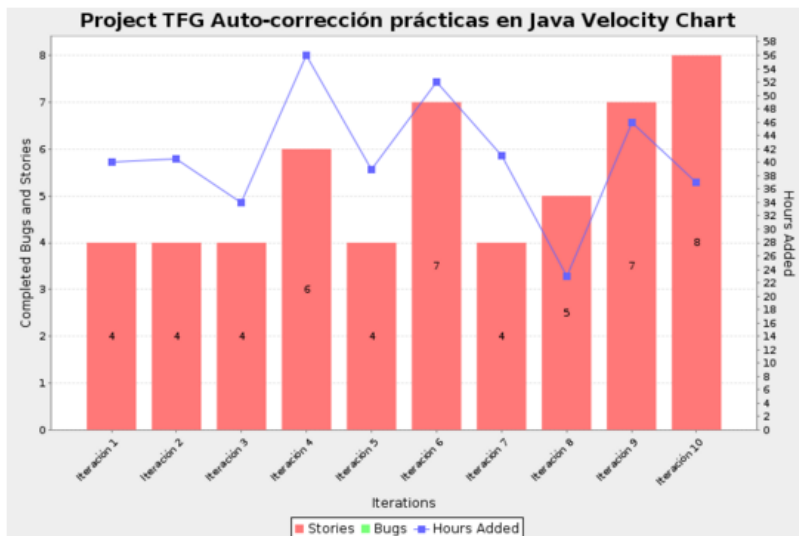


Ilustración 1: Resumen global de la planificación del TFG

Las barras de color rosa reflejan el número de historias que abarca cada iteración. A su vez, estas historias se componen de tareas. La línea azul, muestra las horas invertidas durante la iteración.

En total, el trabajo de fin de grado se ha desarrollado en 10 iteraciones.

A continuación se van a mostrar los *Sprint Backlogs* de cada iteración junto con sus correspondientes *BurndownCharts*.

2.1. Iteración 1 (25 sept 2015 a 1 oct 2015)

Esta fue la primera reunión mantenida con el tutor.

En esta se inició el trabajo fin de grado con tareas más básicas y prácticamente enfocadas a instalaciones y a una investigación inicial:

- Instalación de los programas necesarios para realizar el trabajo fin de grado, así como las configuraciones previas necesarias para la realización del mismo:
 - Instalación del WampServer
 - Instalación de Moodle
 - Instalación del *plugin* LTI
 - Instalación de Webmin





- Instalación de GitHub
- Creación de una Base de datos para la plataforma Moodle
- Creación de una cuenta en XP-DEV
- Primera toma de contacto con Moodle y primer ejemplo usando el *plugin* LTI.
- Investigación acerca del protocolo OAuth.
- Comienzo de la documentación del Anexo del manual del programador.

Type	Id	Description	Hours			Assigned To	Completed
			Total	Added	Remaining		
📁	#1	Instalación de herramientas	18	19	0	avg0043	105 %
📁	#11	Moodle LTI	4	4	0	avg0043	100 %
📁	#13	PHP-OAuth	7	12	0	avg0043	171 %
📁	#17	Anexo "Manual del programador"	7	5	0	avg0043	71 %

Ilustración 2: Sprint Backlog Iteración 1

Iteration starts from Fri 25 Sep, 2015 to Thu 01 Oct, 2015

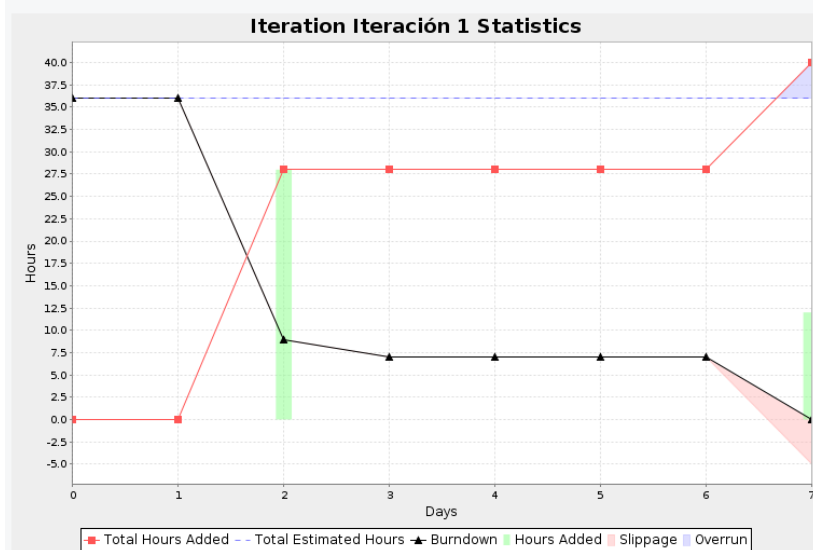


Ilustración 3: Burndown Chart Iteración 1



2.2. Iteración 2 (1 oct 2015 a 15 oct 2015)

En esta iteración:

- Se crean los hosts virtuales modificando el fichero de configuración de apache.
- Se trabaja sobre los Anexos y se modifican los siguientes apartados:
 - Manual del programador.
 - Especificación de requisitos.
 - Plan de proyecto software.
 - Diagramas de casos de uso.
 - Modificación del formato del documento (tamaño y letra de los títulos).
- Se trabaja con el protocolo OAuth. Para ello en primer lugar se busca una librería para poder trabajar con el protocolo, y después se realiza un ejemplo de ejecución para ver su funcionamiento.
- Se sigue investigando sobre el LTI y se realiza un ejemplo que permite enlazar Moodle con nuestra aplicación web creando para ello una actividad de tipo “herramienta externa” desde Moodle. El resultado es que al seleccionar en dicha actividad, se mostrará en la aplicación web la información del usuario que la ha seleccionado.









Type	Id	Description	Hours			Assigned To	Completed
			Total	Added	View Story #10 Remaining		
	#19	Webmin	4	1	0	avg0043	
	#22	Documentación	13	13.5	0	avg0043	
	#24	OAuth	13	12	0	avg0043	
	#34	LTI	10	14	0	avg0043	

Ilustración 4: Sprint Backlog Iteración 2





Iteration starts from Thu 01 Oct, 2015 to Thu 15 Oct, 2015

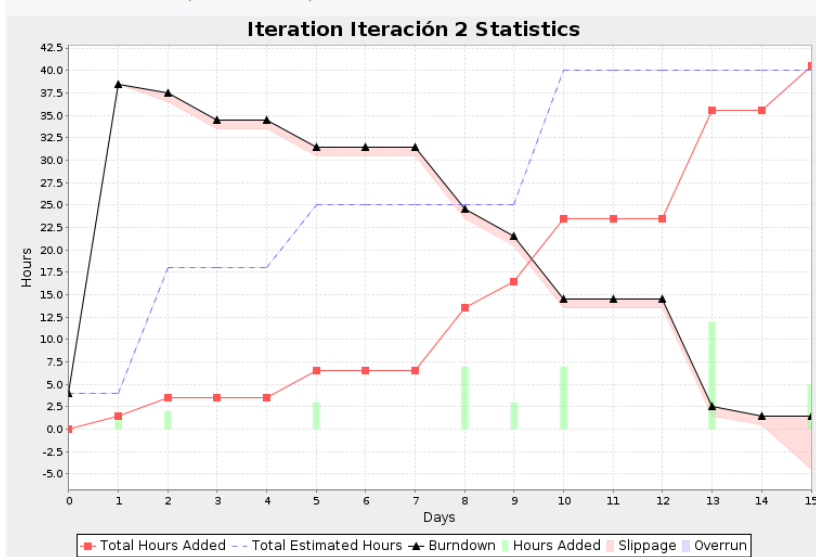


Ilustración 5: Burndown Chart Iteración 2

2.3. Iteración 3 (15 oct 2015 a 29 oct 2015)

Tareas realizadas durante esta iteración:

- Creación de un formulario de subida de ficheros, que va a permitir que posteriormente los alumnos puedan subir sus prácticas y los profesores los test que van a aplicarse a dichas prácticas.
- Creación y uso de una tabla en la base de datos que va a almacenar los parámetros LTI (URL, *consumer_key* y *secret*) necesarios a la hora de realizar la conexión entre Moodle y la aplicación web.
- Investigación sobre Bootstrap e implementación de algún ejemplo para entender mejor su funcionamiento.
- Documentación de los Anexos:
 - Se añade la anterior iteración a la planificación temporal del proyecto.
 - Modificación de los requisitos, diagramas y plantillas de los casos de uso.



Type	Id	Description	Hours			Assigned To	Completed
			Total	Added	Remaining		
📄	#36	Documentación	4	4	0	avg0043	100%
📄	#38	Formulario con subida de fichero	12	12	0	avg0043	100%
📄	#40	Tabla con usuarios y contraseñas para conexión	15	16	0	avg0043	106%
📄	#47	Bootstrap	3	2	0	avg0043	66%

Ilustración 6: Sprint Backlog Iteración 3

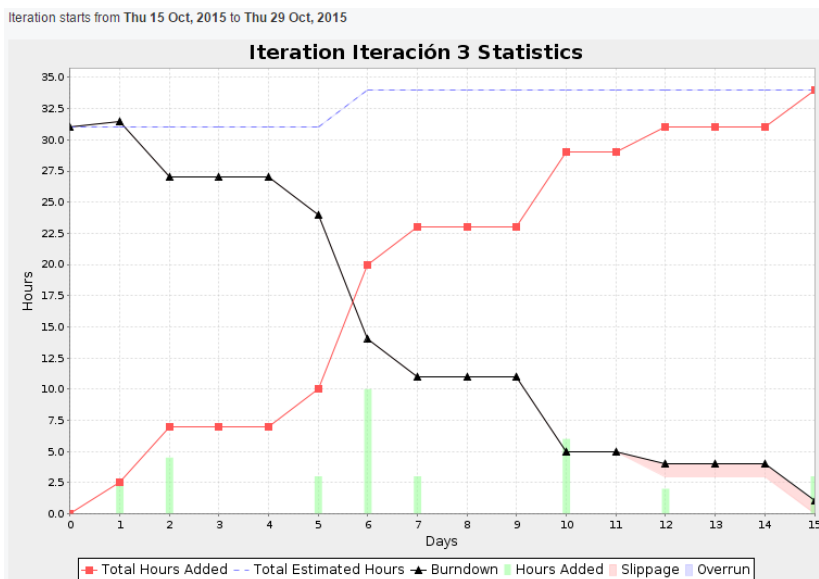


Ilustración 7: Burndown Chart Iteración 3

2.4. Iteración 4 (29 oct 2015 a 12 nov 2015)

Tareas realizadas en esta iteración:

- Creación de un formulario para registrar a los profesores que quieran usar el servicio web. Tras rellenar el formulario los profesores obtendrán los parámetros LTI necesarios para crear la tarea de tipo herramienta externa desde Moodle.
- Se lleva a cabo que cuando un alumno acceda por primera vez al servicio web, este sea registrado en la base de datos con sus datos de Moodle.





- Se crea por primera vez un script SQL que va a permitir la creación y eliminación de las tablas de la base de datos.
- Se investiga sobre Maven y se consigue:
 - La creación automática desde php de un arquetipo maven.
 - El proceso de guardado de test y prácticas subidas por profesores y alumnos respectivamente (no se consigue que puedan subirse en archivos zip).
 - Ejecutar los test subidos por el profesor para corregir las prácticas y comprobar si han sido pasados correctamente o no.
- En cuanto a la documentación de los Anexos:
 - Se añade la pasa iteración en la planificación temporal del proyecto.
 - Se modifican los diagramas de casos de uso.
 - Se añade el apartado de diccionario de datos.
 - Se modifican los requisitos funcionales.
- Se investiga sobre el *framework* CakePHP y se adapta el código que se ha realizado hasta el momento para poder integrarse en dicho *framework*.

Type	ID	Description	Hours			Assigned To	Completed
			Total	Added	Remaining		
	#49	Formulario para registrar al profesor	15	15	0	avg0043	
	#51	Registrar alumno en base de datos	8	8	0	avg0043	
	#53	Script sql base de datos	2	2	0	avg0043	
	#56	Investigación sobre Maven	12	12	0	avg0043	
	#58	Documentación "Anexos"	7	5	0	avg0043	
	#62	Framework CakePHP	18	14	0	avg0043	

Ilustración 8: Sprint Backlog Iteración 4

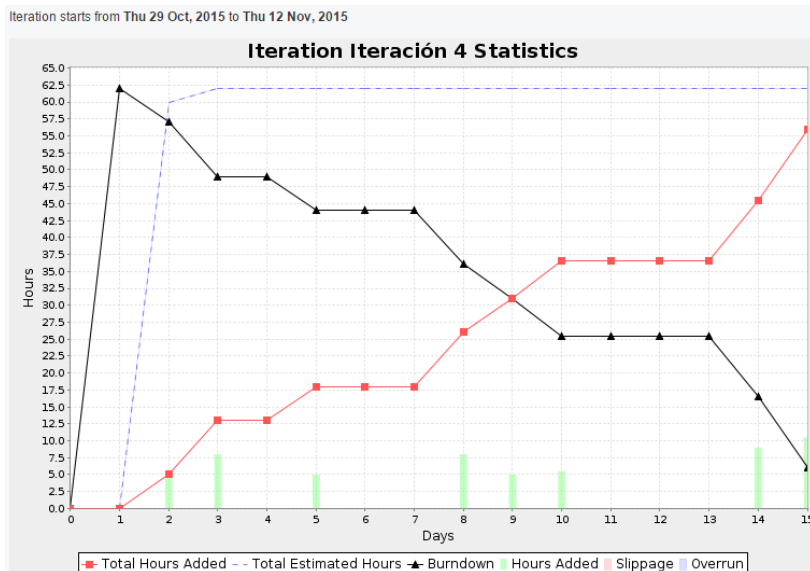


Ilustración 9: Burndown Chart Iteración 4

2.5. Iteración 5 (12 nov 2015 a 26 nov 2015)

Lista de tareas realizadas:

- Se permite la subida de ficheros .zip para que los profesores y alumnos puedan subir sus test y prácticas respectivamente.
- Tras la subida de las prácticas de los alumnos, se comprueba que estas tengan la estructura correcta de carpetas acorde al paquete establecido por el profesor, se comprueba una correcta compilación, y se ejecutan los test verificando la salida generada.
- Se comienza a leer documentación acerca del *plugin* PMD de Maven y se consigue:
 - Generar el fichero xml.
 - Generar el reporte en formato html.
- Se investiga acerca de cómo leer ficheros xml desde php.
- En cuanto a la documentación:





- Anexos: se añade la pasada iteración a la planificación del proyecto y se añaden las herramientas utilizadas en el correspondiente apartado.
- Memoria: se crea un documento para la memoria y se comienza a trabajar sobre el apartado de aspectos relevantes.

Type	Id	Description	Hours			Assigned To	Completed
			Total	Added	Remaining		
task	#65	Permitir subir ficheros dentro de archivos zip	15	12	0	avg0043	100%
task	#67	Realizar comprobaciones tras subir los alumnos las prácticas	8	4	0	avg0043	100%
task	#69	Probar a utilizar PMD con Maven	27	18	0	avg0043	100%
task	#73	Documentación	7	5	0	avg0043	100%

Ilustración 10: Sprint Backlog Iteración 5

Iteration starts from Thu 12 Nov, 2015 to Thu 26 Nov, 2015

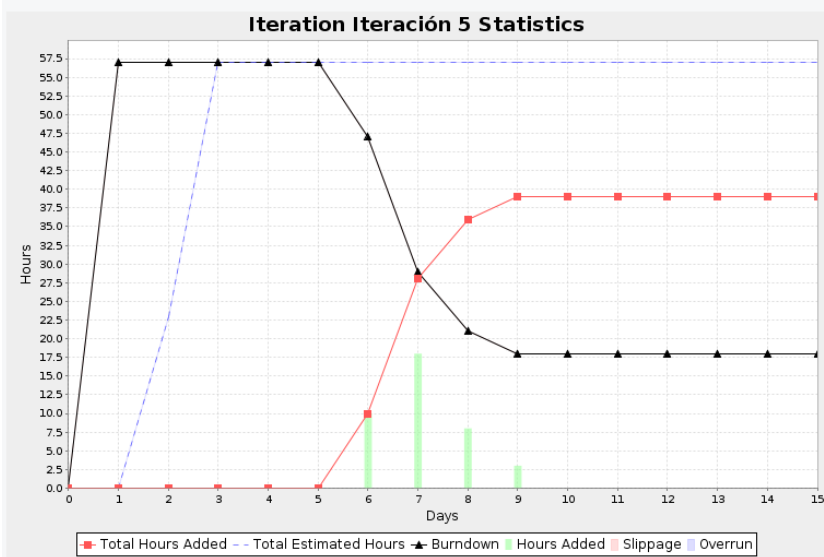


Ilustración 11: Burndown Chart Iteración 5



2.6. Iteración 6 (26 nov 2015 a 10 dic 2015)

Tareas que se plantearon en esta iteración:

- En el panel del profesor se crea una opción para que este pueda descargarse cada una de las prácticas subidas por los alumnos en los diferentes intentos.
- En el panel del profesor se crea una opción para que pueda comprobar si existen plagios entre las prácticas subidas por los alumnos en su último intento.
- Se integran diferentes *plugins* (PMD, JAVANCSS, JDEPEND y FINDBUGS) en Maven para poder generar reportes que puedan ser mostrados al alumno tras su intento de subida de práctica.
- Se trabaja en el aspecto de seguridad para evitar accesos locales a la aplicación. Se consigue lanzar una excepción en tal caso pero no acaba de convencer del todo dicha solución, por lo que queda incompleta.
- Se consiguen generar reportes para los test unitarios, para poder ser visualizados por los alumnos tras sus intentos de subida de prácticas.
- Se generan gráficas con diferentes estadísticas (número de alumnos que pasan los test, número de violaciones de código cometidas...) para mostrárselas a los alumnos y profesores.
- En cuanto a la documentación:
 - Memoria:
 - Añadidos nuevos sub apartados en el apartado “aspectos relevantes”.
 - Anexos:
 - Añadida pasada iteración en la planificación temporal del proyecto.
 - Añadidos nuevos requisitos funcionales.
 - Añadido nuevo sub apartado “conexión lti” en el apartado “manual del programador”.





Type	Id	Description	Hours			Assigned To	Completed
			Total	Added	Remaining		
📄	#77	Opción descarga de prácticas	4	4	0	Zurfi H242 avg0043	100 %
📄	#80	Comprobación de plagios	8	5	0	Zurfi H242 avg0043	100 %
📄	#82	Utilización de diferentes plugins de maven	8	8	0	Zurfi H242 avg0043	100 %
📄	#84	Seguridad para evitar accesos desde local	6	3	0	Zurfi H242 avg0043	100 %
📄	#86	Generar reporte de los test unitarios	8	6	0	Zurfi H242 avg0043	100 %
📄	#88	Generar Gráficas	20	21	0	Zurfi H242 avg0043	105 %
📄	#91	Documentación	7	5	0	Zurfi H242 avg0043	100 %

Ilustración 12: Sprint Backlog Iteración 6

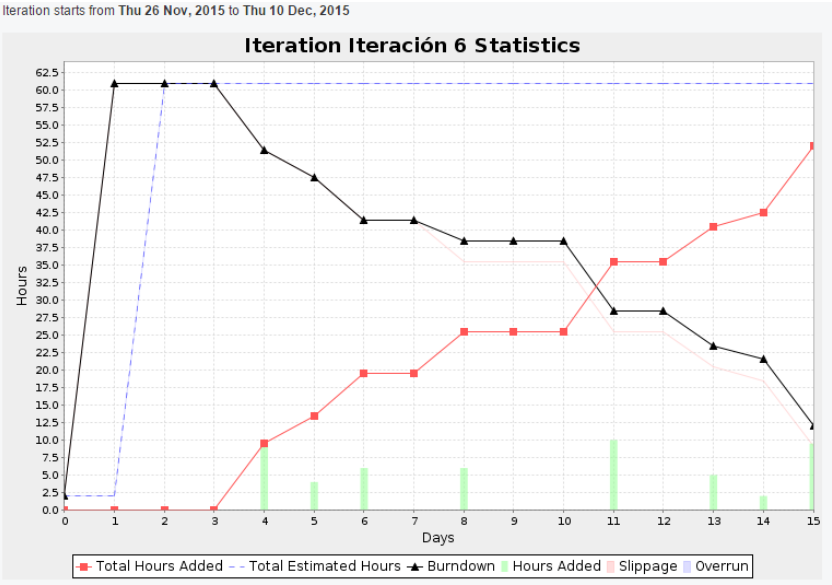


Ilustración 13: Burndown Chart Iteración 6



2.7. Iteración 7 (10 dic 2015 a 24 dic 2015)

Tareas planteadas en esta iteración:

- Se diseña una página de error para que sea mostrada cuando se producen accesos locales a la aplicación, los cuales no son permitidos.
- Se crean generan nuevas gráficas para ser mostradas tanto para el profesor como para el alumno. Además se modifican gráficas existentes.
- Se comienza a trabajar sobre el diseño de la interfaz gráfica de la aplicación.
- Documentación:
 - Memoria:
 - Creación el apartado “herramientas”.
 - Creación del apartado “conceptos teóricos”.
 - Anexos:
 - Se añade la pasada iteración a la planificación temporal del proyecto.
 - En el apartado “manual del programador” se hace referencia a los nombres de clases y métodos cuando es necesario.

Type	Id	Description	Hours			Assigned to	Completed
			Total	Allocated	Remaining		
📄	#97	Excepción, página de error	2	1.5	0	arg043	75%
📄	#99	Gráficas	15	15	0	arg043	100%
📄	#101	Diseño interfaz	20	13	0	arg043	65%
📄	#103	Documentación	11	11.5	0	arg043	104%

Ilustración 14: Sprint Backlog Iteración 7





Iteration starts from Thu 10 Dec, 2015 to Thu 24 Dec, 2015

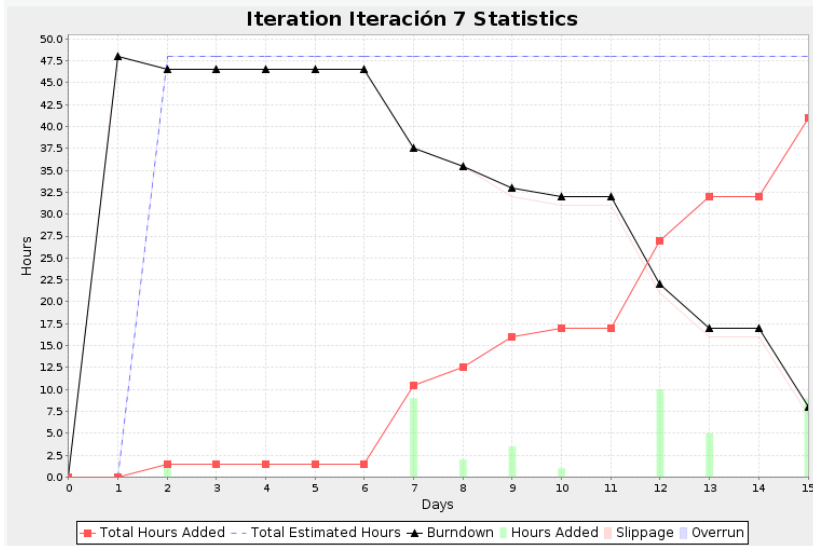


Ilustración 15: Burndown Chart Iteración 7

2.8. Iteración 8 (24 dic 2015 a 7 ene 2016)

Tareas realizadas en esta iteración:

- Se añade la funcionalidad necesaria para poder capturar y guardar las excepciones que se pueden lanzar al ejecutar las prácticas de los alumnos.
- Se crean nuevas gráficas para ser mostradas en el panel del profesor y de los alumnos.
- En la tabla de estadísticas del profesor se crean varios botones con los que el profesor podrá observar los reportes asociados a la práctica del alumno que haya seleccionado.
- Se generan pruebas unitarias utilizando PHPUnit.
- En cuanto a la documentación, en la memoria se crea el apartado “trabajos relacionados” y en los anexos el de “estudio de la viabilidad”.



Type	Id	Description	Hours			Assigned To	Completed
			Total	Added	Remaining		
task	#110	Tratamiento de las excepciones que pueden saltar	8	3	0	avg0043	100 %
task	#112	Añadir nuevas gráficas	8	3	0	avg0043	100 %
task	#114	Dar opción de ver los reportes en la tabla de las estadísticas del profesor	10	5	0	avg0043	100 %
task	#116	Realizar Pruebas	6	7	0	avg0043	116 %
task	#119	Documentación	5	5	0	avg0043	100 %

Ilustración 16: Sprint Backlog Iteración 8

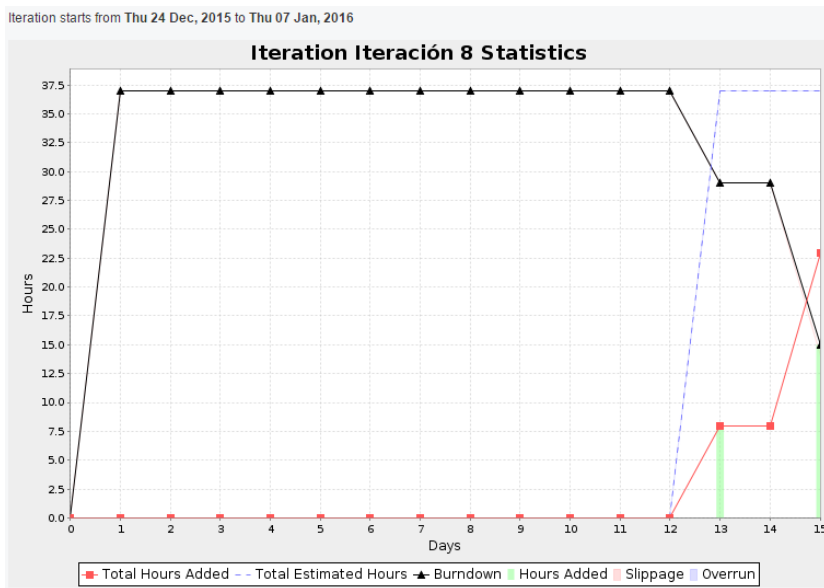


Ilustración 17: Burndown Chart Iteración 8





2.9. Iteración 9 (7 ene 2016 a 21 ene 2016)

Tareas realizadas en esta iteración:

- Eliminadas gráficas innecesarias.
- Realizadas pruebas sobre la aplicación con diferentes ficheros zip a subir en los formularios.
- Internacionalización: se crean los ficheros .po necesarios.
- Se corrige la tabla de estadísticas presente en el panel del profesor.
- Actualizadas pruebas unitarias.
- Se trabaja en la documentación, tanto en la Memoria como en los Anexos.
- Se realizan últimos cambios en la interfaz de la aplicación.

Type	Id	Description	Hours			Assigned To	Completed
			Total	Added	Remaining		
	#122	Eliminar gráficas innecesarias	2	1	0	avg0043	100 %
	#126	Realizar pruebas de subida de ficheros zip	8	12	0	avg0043	150 %
	#128	Internacionalización	8	7	0	avg0043	100 %
	#130	Corregir tabla del panel del profesor	8	5	0	avg0043	100 %
	#132	Pruebas unitarias	6	7	0	avg0043	116 %
	#134	Documentación	10	6	0	avg0043	100 %
	#136	Interfaz	8	8	0	avg0043	100 %

Ilustración 18: Sprint Backlog Iteración 9

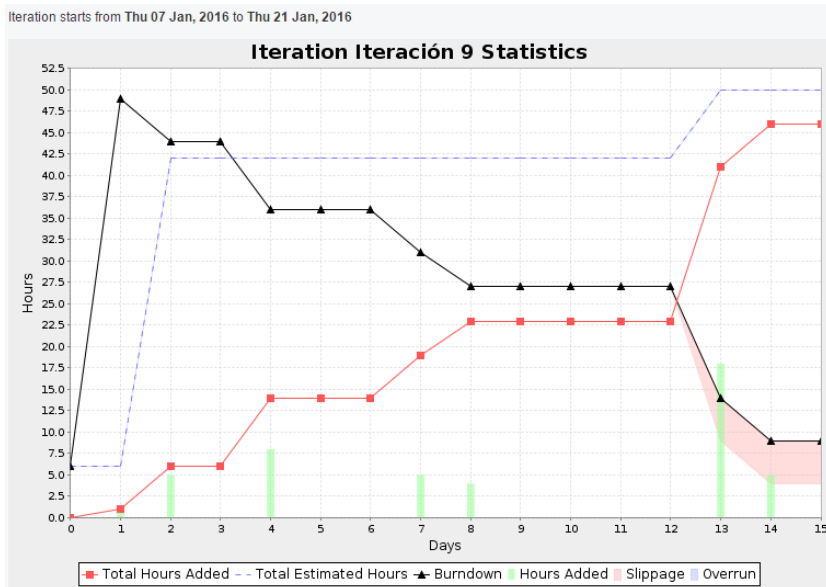


Ilustración 19: Burndown Chart Iteración 9

2.10. Iteración 10 (21 ene 2016 a 4 feb 2016)

Se trata de la última iteración realizada. Tareas desarrolladas:

- Se actualizan las pruebas unitarias.
- Se termina de realizar toda la documentación.
- Documentación del código de la aplicación (clases, métodos...).
- Últimas mejoras en la interfaz.
- Internacionalización: se habilita un botón en la barra de navegación para realizar el cambio de idioma.
- Máquina virtual: se instala la aplicación dentro de una máquina virtual.
- Se realizan pruebas sobre la aplicación, subiendo diferentes prácticas.
- Creación de vídeos con el funcionamiento de la aplicación.





Type	Id	Description	Hours			Assigned To	Completed
			View Story #138	Added	Remaining		
	#138	Pruebas Unitarias		2	1	0	avg0043
	#141	Documentación	View Story #143	15	0	0	avg0043
	#143	Documentar código de la aplicación	View Story #145	2	0	0	avg0043
	#145	Interfaz		5	3	0	avg0043
	#147	Internacionalización		8	3	0	avg0043
	#149	Máquina Virtual		9	7	0	avg0043
	#151	Pruebas de la aplicación		5	3	0	avg0043
	#153	Videos con el funcionamiento de la aplicación		5	3	0	avg0043

Ilustración 20: Sprint Backlog Iteración 10

Iteration starts from Thu 21 Jan, 2016 to Thu 04 Feb, 2016

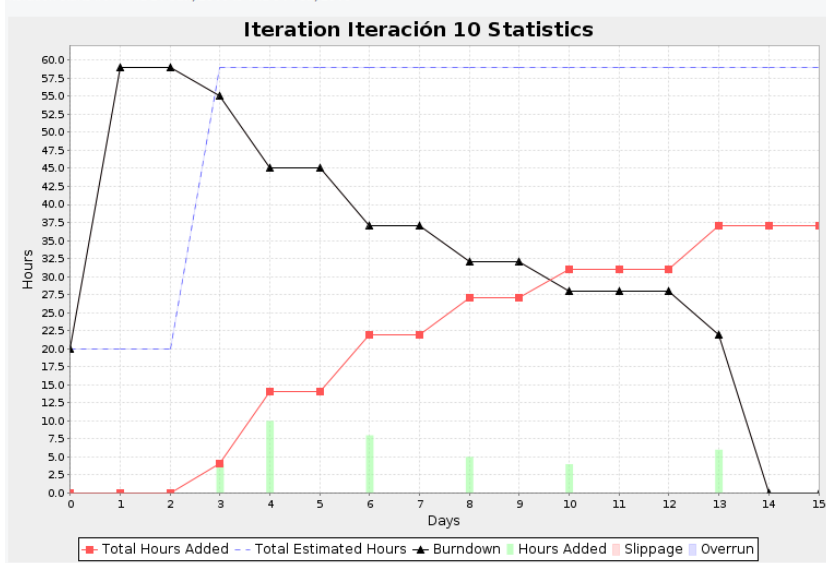


Ilustración 21: Burndown Chart Iteración 10



3. Estudio de viabilidad

El estudio de la viabilidad comprende las acciones llevadas a cabo para determinar los factores económicos y legales, entre otros, requeridos para el desarrollo del proyecto. Este apartado tiene gran importancia ya que de él dependerá determinar si el proyecto es viable o no.

Cuando se afronta un proyecto han de tenerse en cuenta una serie de riesgos que podrían surgir a la hora de evaluar la viabilidad del proyecto. La magnitud del proyecto, los conocimientos necesarios para desarrollarlo, el dinero y tiempo necesarios son algunas de las trabas que podrían surgir a la hora de realizar el proyecto.

3.1. Estudio de viabilidad legal

Se basa en comprobar cualquier posibilidad de infracción, violación o responsabilidad legal en que se podría incurrir al desarrollar el proyecto.

En este apartado se explica el tipo de licencia que tendrá nuestra aplicación. La aplicación se distribuirá bajo la licencia *Apache License v.2*.

A continuación se citan las principales características de dicha licencia.

3.1.1. Apache License versión 2.0

La licencia Apache (*Apache License* o *Apache Software License* para versiones anteriores a 2.0) es una licencia de software libre creada por la *Apache Software Foundation (ASF)*. La licencia *Apache* (con versiones 1.0, 1.1 y 2.0) requiere la conservación del aviso de *copyright* y el *disclaimer*, pero no es una licencia *copyleft*, ya que no requiere la redistribución del código fuente cuando se distribuyen versiones modificadas [1].

3.1.2. Condiciones de la licencia

Esta licencia permite al usuario del software la libertad de usarlo para cualquier propósito, distribuirlo, modificarlo y distribuir versiones modificadas de ese software.

La Licencia *Apache* no exige que las obras derivadas (versiones modificadas) del software se distribuyan usando la misma licencia, ni siquiera que se tengan que distribuir como software libre/*Open Source*. La Licencia *Apache* sólo exige que se mantenga una noticia que informe a los receptores que en la distribución se ha usado código con la Licencia *Apache* [1].

Se deben añadir dos archivos en el directorio principal de los paquetes de software redistribuidos:





- LICENSE - Una copia de la licencia
- NOTICE - Un documento de texto, que incluye los "avisos" obligatorios del software presente en la distribución.

3.2. Estudio de viabilidad económica

Este apartado se va a centrar en conocer y señalar los costes de desarrollo del proyecto, de forma que se pueda evaluar la viabilidad del mismo y así poder realizar una comparación de dichos costes con los beneficios esperados tras la finalización del proyecto.

3.2.1. Análisis de costes

A continuación se van a analizar los posibles costes que pueden producirse durante el transcurso del proyecto. Los costes que se van a analizar son los de personal, hardware, software, otros costes y los totales:

- **Costes de personal:**

El proyecto será realizado por un programador con experiencia considerada similar a un graduado que va a ser el encargado de realizar todo el trabajo, realizando las tareas correspondientes a las fases de análisis, diseño, implementación y pruebas.

La duración del proyecto se estima en 4 meses, además teniendo en cuenta que el desarrollador informático ha trabajado una media de 5 horas diarias y suponiendo que el salario por hora trabajada es de 9€, el coste del personal sería el siguiente:

$$\frac{9 \text{ €}}{\text{hora}} \times \frac{5 \text{ horas}}{\text{día}} = 45 \frac{\text{€}}{\text{día}}$$

$$\frac{45 \text{ €}}{\text{día}} \times \frac{5 \text{ días}}{\text{semana}} = 225 \frac{\text{€}}{\text{semana}}$$

$$\frac{225 \text{ €}}{\text{semana}} \times \frac{4 \text{ semanas}}{\text{mes}} = 900 \frac{\text{€}}{\text{mes}}$$

Por lo tanto dado que el desarrollador ha trabajado durante 4 meses para realizar el proyecto, el coste del desarrollador sería:

$$\frac{900 \text{ €}}{\text{mes}} \times 4 \text{ meses} = 3600 \text{ €}$$

Por último hay que tener en cuenta los gastos de la Seguridad social, que es uno de los principales impuestos que tiene que afrontar la empresa. Según 2015 para las



contingencias comunes el porcentaje es del 23,60 % a cargo de la empresa [2], por lo tanto:

$$\text{Coste de personal} = 3600 \text{ €} + 23,60 \% \text{ de } 3600 \text{ €} = 4449,6 \text{ €}$$

- **Coste de hardware:**

Para desarrollar el proyecto se ha hecho uso del siguiente portátil:

$$\text{Toshiba Satellite L755} - 1\text{J9} = 620 \text{ €}$$

El uso de este ordenador no va a ser exclusivo de este proyecto en particular sino que podrá seguir siendo utilizado, y por lo tanto los costes son amortizables.

Según la ley del Impuesto de Sociedades, la vida media del inmovilizado hardware oscila entre 4 y 8 años y puesto que el hardware adquirido es de calidad media se estimará que el tiempo de vida del hardware es de 5 años, de esta forma la amortización resultante sería:

$$\text{Tiempo amortización: } 5 \text{ años} = 60 \text{ meses}$$

$$\text{Precio del ordenador portátil} = 620 \text{ €}$$

$$\text{Coste de amortización: } \frac{620 \text{ €}}{60 \text{ meses}} = 10,33 \frac{\text{€}}{\text{mes}}$$

Como la duración del proyecto está estimada en 4 meses, el coste hardware del proyecto sería el siguiente:

$$\text{Coste de hardware} = 10,33 \frac{\text{€}}{\text{mes}} \times 4 \text{ meses} = 41,32 \text{ €}$$





- **Coste de software:**

El coste de software para el proyecto viene detallado en la siguiente tabla, en la que va a mostrarse la licencia utilizada y su número junto al coste que conllevan:

Software	Licencia	Nº licencias	Coste
<i>Microsoft Windows 8.1 (64 bits)</i>	Individual	1	100,00 €
<i>Microsoft Office 2013</i>	Individual	1	99,00 €
<i>Moodle</i>	GNU GPL	1	0,00 €
<i>XP-DEV</i>	GNU GPL	1	0,00 €
<i>WampServer</i>	GNU GPL	1	0,00 €
<i>Eclipse PHP Developers</i>	GNU GPL	1	0,00 €
<i>Maven</i>	APACHE License	1	0,00 €
<i>GitHub</i>	GNU GPL	1	0,00 €
<i>HeidiSQL</i>	GNU GPL	1	0,00 €
<i>PoEdit</i>	GNU GPL	1	0,00 €
<i>Astah Community</i>	Gratuita	1	0,00 €
<i>FileSync</i>	GNU GPL	1	0,00 €
<i>Bootstrap</i>	MIT License	1	0,00 €
TOTAL			199, 00 €

Tabla 1: Costes de software

Del mismo modo que ocurría con el hardware, el software también es amortizable. Y según la ley del Impuesto de Sociedades, el tiempo de vida para el software oscila entre 3 y 6 años, considerando un tiempo de vida medio de 4 años el coste de amortización sería el siguiente:

$$\text{Coste de amortización: } \frac{199 \text{ €}}{48 \text{ meses}} = 4,14 \frac{\text{€}}{\text{mes}}$$

$$\text{Coste de software} = 4,14 \frac{\text{€}}{\text{mes}} \times 4 \text{ meses} = 16,56 \text{ €}$$



- **Otros costes:**

Además hay que tener en cuenta los siguientes costes:

$$\text{Material de oficina: } 5 \frac{\text{€}}{\text{mes}} \times 4 \text{ meses} = 20 \text{ €}$$

$$\text{Electricidad: } 20 \frac{\text{€}}{\text{mes}} \times 4 \text{ meses} = 80 \text{ €}$$

$$\text{Internet: } 30 \frac{\text{€}}{\text{mes}} \times 4 \text{ meses} = 120 \text{ €}$$

$$\text{Coste de otros costes} = 20 \text{ €} + 80 \text{ €} + 120 \text{ €} = 220 \text{ €}$$

- **Costes totales:**

Para obtener el coste total se realiza la suma de los costes anteriormente calculados, por lo que el coste total obtenido sería el siguiente:

$$\text{Coste de personal} = 4449,6 \text{ €}$$

$$\text{Coste de hardware} = 41,32 \text{ €}$$

$$\text{Coste de software} = 16,56 \text{ €}$$

$$\text{Otros costes} = 220 \text{ €}$$

$$\text{Coste total} = 4449,6 \text{ €} + 41,32 \text{ €} + 16,56 \text{ €} + 220 \text{ €} = 4727,48 \text{ €}$$

3.2.2. Análisis de costes – beneficio

Como ya se ha mencionado anteriormente, la aplicación es compatible con cualquier plataforma LMS que cumpla el estándar LTI, por lo que esta puede ser utilizada desde un gran número de plataformas.

En este apartado se explica la forma de recuperar la inversión realizada. Para ello el procedimiento que se sigue es el de ofrecer paquetes de servicio a los usuarios finales.

Los paquetes de servicio que se pueden ofertar son los siguientes:

- Personalizar la aplicación, con el estilo usado por el centro educativo que contrate esta opción.
- Solución de problemas o bugs.
- Atención al cliente para cualquier tipo de sugerencia o problema.





En 2008, el número de institutos, academias, universidades y empresas españolas inscritas en Moodle ascendían a 4000 [3]. Además existen multitud de centros en otros países que utilizan Moodle y otras plataformas LMS que cumplen con el estándar LTI [4].

Tomando como dato que 480 centros de este tipo están interesados en la aplicación y que estos paquetes de servicio tienen un precio de 10€, por lo que se obtendrían 4800€, ya permitiría recuperar la inversión realizada que era de 4727,48€.

A partir de este momento, todos los nuevos paquetes ofrecidos garantizarían beneficios.



II - ESPECIFICACIÓN DE REQUISITOS

1. Introducción

En este apartado se van a detallar los diferentes requisitos de nuestra aplicación web.

Estos requisitos van a estar divididos en dos grupos: funcionales y no funcionales.

2. Requisitos funcionales

2.1. Propios de Moodle

2.1.1. Rol profesor

- **RF-M01: Crear la tarea:** El profesor creará en Moodle y dentro del curso correspondiente, una tarea de tipo “herramienta externa” que enlazará con la aplicación web estableciendo la URL, *consumer_key* y *secret* correspondiente. Cuando esta tarea sea seleccionada, se redirigirá a dicha aplicación web desarrollada en este proyecto. Tal enlazamiento es permitido gracias al LTI.
- **RF-M02: Configurar la tarea:** El profesor en todo momento podrá configurar o editar la tarea creada: cambiar su nombre, URL, ocultarla, ponerla visible...

2.1.2. Rol profesor y alumno

- **RF-M03: Seleccionar tarea:** Al hacer clic en Moodle sobre la tarea de tipo “herramienta externa”, el usuario será redirigido en función de su rol (profesor/alumno) a la aplicación web.

2.2. Propios de la aplicación web

2.2.1. Rol profesor

- **RF-AW01: Registrarse en la aplicación web:** El profesor rellenará el formulario correspondiente para registrarse y obtendrá los parámetros LTI necesarios (URL, *consumer_key* y *secret*) para crear en Moodle la tarea de tipo “herramienta externa”.
- **RF-AW02: Configurar los parámetros de la tarea:** El profesor configurará los parámetros correspondientes de la tarea (número máximo de intentos posibles, fecha límite de entrega...) para que posteriormente los alumnos puedan subir sus prácticas.





- **RF-AW03: Subir test:** El profesor podrá subir el/los test que desee dentro de un zip desde el formulario de subida de ficheros. Estos test se aplicarán posteriormente a las prácticas subidas por los alumnos. Además en la subida del test podrá subir un enunciado de la práctica que tengan que realizar los alumnos.
- **RF-AW04: Consultar estadísticas de las prácticas:** El profesor podrá ver en una tabla las estadísticas que se corresponden con cada uno de los intentos de subida de práctica realizado por los alumnos. Estas estadísticas serían: número del intento realizado, si la práctica ha pasado los test o no, fecha de subida de la práctica y opción de ver los reportes asociados.
- **RF-AW05: Descargar prácticas:** El profesor desde la tabla de estadísticas podrá descargar las prácticas en formato zip subidas por los alumnos.
- **RF-AW06: Comprobar plagios de prácticas:** El profesor podrá comprobar si existen plagios entre las prácticas subidas por los alumnos. Esta comprobación se realizará entre el último intento de subida de práctica realizado de todos los alumnos.
- **RF-AW07: Visualizar gráficas:** El profesor tendrá la opción de ver diferentes gráficas para tener una visión global referente a las diferentes prácticas subidas por los alumnos (media de violaciones de código cometidas, prácticas que pasan los test...).

2.2.2. Rol alumno

- **RF-AW08: Subir práctica:** El alumno podrá subir la práctica desde el correspondiente formulario de subida de ficheros para su proceso de corrección. La práctica se subirá dentro de un fichero zip. Además también podrá incluir comentarios a la práctica que va a subir.
- **RF-AW9: Consultar datos de la tarea:** El alumno podrá ver la información relacionada con la tarea: nombre del paquete que debe de tener la práctica, fecha límite de entrega de la práctica, número de intentos máximo de subida de la práctica y número de intentos realizados.
- **RF-AW10: Reemplazar la práctica seleccionada:** Si el alumno ha seleccionado en la entrega de la práctica una práctica que no quería subir, podrá reemplazarla por otra.
- **RF-AW11: Consultar reportes:** El alumno podrá ver los reportes correspondientes a la última práctica subida. Los reportes pertenecientes a los *plugins* JDepend y JavanCSS siempre estarán disponibles, mientras que los de los *plugins* PMD, FindBugs y de Errores (unitarios o excepciones)



únicamente cuando la práctica subida presente violaciones en el código o errores.

- **RF-AW12: Consultar gráficas:** El alumno podrá visualizar gráficas para observar la evolución de su ejercicio comparado con versiones anteriores tras realizar cada intento de subida de práctica.

2.3. Requisitos de restricción

- **RR-01:** La aplicación sólo permite trabajar con test y prácticas escritas en Java, en este momento no soporta otros lenguajes de programación.
- **RR-02:** La aplicación prohíbe que los alumnos accedan al panel del profesor y de la misma manera el profesor tampoco puede acceder al panel de los alumnos.

3. *Requisitos no funcionales*

- **RNF-01: Rapidez:** Después de que el alumno suba la práctica para ser corregida, los resultados obtenidos de dicha corrección deberán ser recibidos en un tiempo lo más corto posible.
- **RNF-02: Usabilidad y diseño responsivo:** Se debe permitir que la aplicación web pueda ser utilizada desde diferentes dispositivos, independientemente del tamaño de la pantalla que tengan.
- **RNF-03: Interfaz:** La interfaz de la aplicación web debe ser clara para que no dé lugar a ningún tipo de confusión.
- **RNF-04: Idiomas:** La aplicación puede visualizarse en varios idiomas, siendo solo necesario incluir un fichero en formato .po para incluir nuevos idiomas.





4. Diagrama de casos de uso

En este apartado se van a incluir los diagramas de casos de uso de la aplicación. Va a poder verse reflejada la interacción entre el usuario (profesor y alumno) sobre la aplicación.

Para facilitar la comprensión y la visualización, se va a utilizar un color diferente para diferenciar los casos de uso que van a pertenecer a las dos partes involucradas en la funcionalidad que ofrece la aplicación:

- Las propias de Moodle.
- Las propias de la Aplicación web.

A continuación se muestra una ilustración que va a reflejar lo explicado anteriormente, por un lado y con color gris azulado se muestran los casos de uso pertenecientes a Moodle y con color amarillo los de la Aplicación web:



Ilustración 22: DCU - colores

A continuación se va a mostrar los diagramas de casos (DCU) de uso de los diferentes requisitos definidos anteriormente.

- **Diagrama casos de uso “general del sistema”**

Este diagrama va a mostrar el diagrama de caso de uso más general, es decir, el de todo el sistema. Se va a ver reflejado las diferentes relaciones que hay entre los requisitos enumerados anteriormente.

Va a tener la presencia de los dos roles posibles: profesor y alumno.

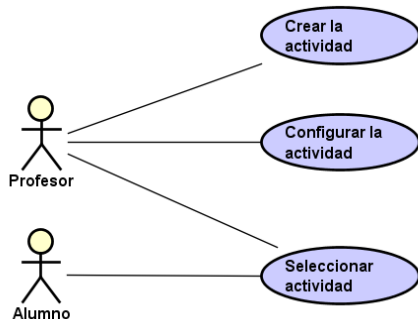


Ilustración 23: DCU - general del sistema

A continuación se van a mostrar los diagramas de casos de uso más específicos:

- **Diagrama casos de uso “seleccionar actividad”**

- a. Rol profesor:

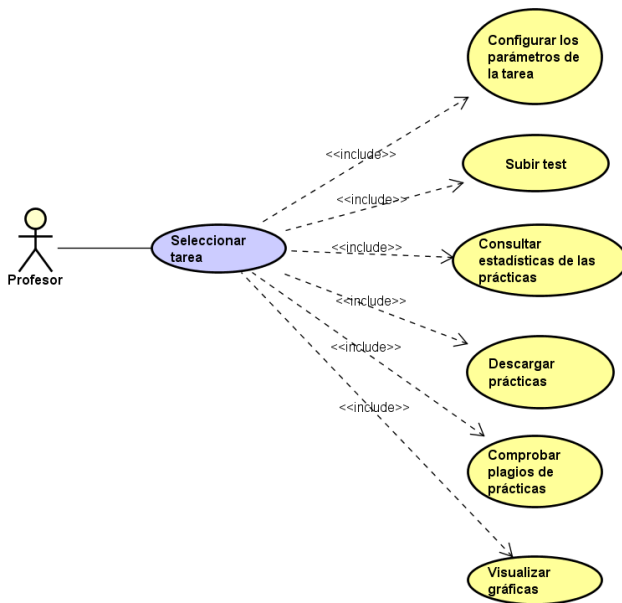
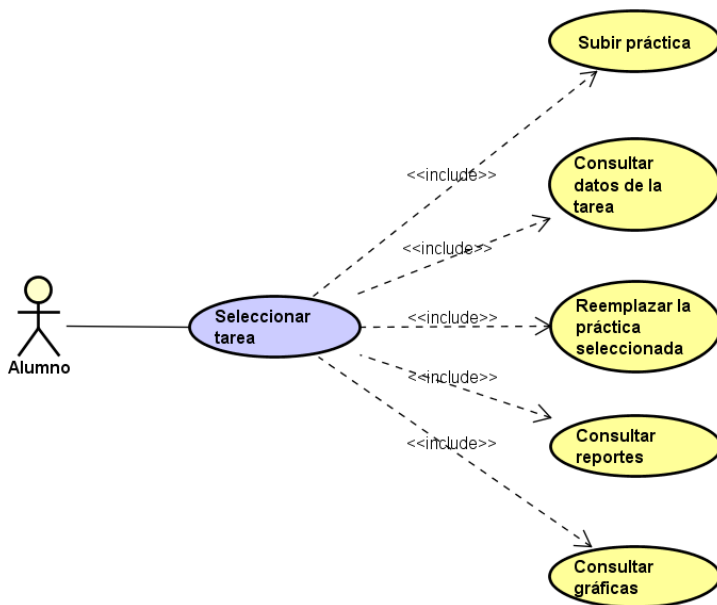


Ilustración 24: DCU - profesor seleccionar tarea



b. Rol alumno:*Ilustración 25: DCU - alumno seleccionar tarea*



4.1. Plantillas de casos de uso

La plantilla tipo que se va utilizar en este apartado tiene la siguiente forma:

REF <id del requisito>	<nombre del requisito>	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso { concreto cuando <evento de activación> , abstracto durante la realización de los casos de uso <lista de casos de uso> }	
Precondición	<precondición del caso de uso>	
Secuencia Normal	Paso	Acción
	1	{El <actor> , El sistema} <acción realizada por el actor o sistema>, se realiza el caso de uso < caso de uso RF-x>

Post-condición	<post-condición del caso de uso>	
EXCEPCIONES	Paso	Acción
	1	Si <condición de excepción>,{el <actor> , el sistema} }<acción realizada por el actor o sistema>>, se realiza el caso de uso < caso de uso RF-x>, a continuación este caso de uso {continua, aborta}

Importancia	<baja, media, alta>	
Frecuencia	<baja, media, alta>	
Urgencia	<baja, media, alta>	

Tabla 2: Plantilla tipo casos de uso





4.1.1. Pertenecientes a Moodle

RF-M01	Crear la tarea	
Descripción	Crear una tarea de tipo “herramienta externa” que al ser seleccionada, redirija a la aplicación web.	
Precondición	Acceder a Moodle con los datos de acceso del profesor y que el curso en el que va a añadirse la actividad esté ya creado.	
Secuencia Normal	Paso	Acción
	1	El profesor accede a Moodle con sus datos de acceso, y entra al curso en el que desea crear la tarea.
	2	Crea una tarea de tipo “herramienta externa” estableciendo la URL, <i>consumer_key</i> y <i>secret</i> de la aplicación web con el que se va a enlazar y guarda los cambios.
Post-condición	La nueva tarea aparece en el curso para poder ser accedida.	
EXCEPCIONES	Paso	Acción
	1	Si el profesor introduce datos de acceso incorrectos, no podrá acceder a Moodle.
	2	Si el curso no está creado, no se podrá crear la tarea.
	3	Si introduce alguno de los parámetros (URL, <i>consumer_key</i> , <i>secret</i>) incorrectamente, no podrá acceder a la aplicación web.
Importancia	Alta.	
Frecuencia	Baja.	
Urgencia	Alta.	

Tabla 3: Plantilla RF-M01 “Crear la tarea”



RF-M02	Configurar la tarea	
Descripción	La tarea creada podrá ser configurada o editada por el profesor.	
Precondición	Tener creada la tarea.	
Secuencia Normal	Paso	Acción
	1	El profesor accede a Moodle con sus datos de acceso y solicita “editar” sobre la tarea.
Post-condición	La tarea queda configurada.	
EXCEPCIONES	Paso	Acción
	1	Si el profesor introduce datos de acceso incorrectos, no podrá acceder a Moodle.
Importancia	Media.	
Frecuencia	Baja.	
Urgencia	Media.	

Tabla 4: Plantilla RF-M02 "Configurar la tarea"





RF-M03		Seleccionar tarea	
Descripción		El usuario, bien sea el profesor o el alumno, selecciona sobre la tarea de Moodle y es redirigido a la aplicación web.	
Precondición		La tarea debe de estar creada y configurada correctamente.	
Secuencia Normal	Paso	Acción	
	1	El profesor o alumno accede a Moodle con sus credenciales y selecciona sobre la tarea.	
Post-condición		El usuario accede correctamente a la aplicación web en función de su rol.	
EXCEPCIONES	Paso	Acción	
	1	Los parámetros de la tarea no están configurados correctamente.	
Importancia		Alta.	
Frecuencia		Alta.	
Urgencia		Alta.	

Tabla 5: Plantilla RF-M03 "Seleccionar la tarea"



4.1.2. Pertenecientes a la Aplicación web

- **Rol profesor:**

RF-AW01	Registrarse en la aplicación web	
Descripción	El profesor rellenará el formulario correspondiente para registrarse en la aplicación web y así poder obtener los parámetros LTI necesarios para crear una tarea desde Moodle.	
Precondición	Que el profesor no se haya registrado previamente.	
Secuencia Normal	Paso	Acción
	1	El profesor rellena el formulario de registro con sus datos.
Post-condición	El profesor queda registrado y obtiene los parámetros LTI.	
EXCEPCIONES	Paso	Acción
	1	El profesor no rellena correctamente los campos del formulario (campos vacíos, datos incorrectos).
Importancia	Alta.	
Frecuencia	Media.	
Urgencia	Alta.	

Tabla 6: Plantilla RF-AW01 "Registrarse en la aplicación web"





RF-AW02	Configurar los parámetros de la tarea	
Descripción	El profesor configura los parámetros de la tarea desde el correspondiente formulario.	
Precondición	Que el profesor haya seleccionado desde su panel la opción de configuración de parámetros.	
Secuencia Normal	Paso	Acción
	1	Rellena el formulario y establece los parámetros de la tarea.
	2	Selecciona la opción del formulario para guardar los cambios.
Post-condición	Los parámetros de la tarea quedan configurados correctamente.	
EXCEPCIONES	Paso	Acción
	1	El profesor no rellena correctamente los campos del formulario.
Importancia	Alta.	
Frecuencia	Baja.	
Urgencia	Media.	

Tabla 7: Plantilla RF-AW02 "Configurar los parámetros de la tarea"



RF-AW03	Subir test	
Descripción	El profesor podrá subir los test que desee desde el correspondiente formulario de subida de ficheros. Además también podrá subir un enunciado de la práctica en forma textual.	
Precondición	Que el profesor haya seleccionado desde su panel la opción de subida de test.	
Secuencia Normal	Paso	Acción
	1	El profesor selecciona el fichero zip con los test que desea subir y/o un escribe un enunciado de la práctica.
	2	Selecciona la opción del formulario para subir el test.
Post-condición	Los test quedan subidos.	
EXCEPCIONES	Paso	Acción
	1	El profesor no ha subido los test dentro de un fichero zip.
	2	El profesor no ha seleccionado ningún test a subir ni ha rellenado el enunciado.
Importancia	Alta.	
Frecuencia	Media.	
Urgencia	Alta.	

Tabla 8: Plantilla RF-AW03 "Subir test"





RF-AW04	Consultar estadísticas de las prácticas	
Descripción	El profesor puede visualizar en una tabla todas las estadísticas referentes a los diferentes intentos de subida de práctica realizados por los alumnos.	
Precondición	Que el profesor haya seleccionado desde su panel la opción de consultar estadísticas.	
Secuencia Normal	Paso	Acción
	1	El profesor visualiza en una tabla las estadísticas de todas las prácticas de los alumnos.
Post-condición	Las estadísticas son mostradas en la tabla.	
EXCEPCIONES	Paso	Acción
	1	Los alumnos no han realizado ninguna subida de sus prácticas y por lo tanto no hay estadísticas a visualizar.
Importancia	Media.	
Frecuencia	Media.	
Urgencia	Media.	

Tabla 9: Plantilla RF-AW04 "Consultar estadísticas de las prácticas"



RF-AW05	Descargar prácticas	
Descripción	El profesor podrá descargar todas las prácticas en formato zip que han subido los alumnos.	
Precondición	Que el profesor haya seleccionado desde su panel la opción de consultar estadísticas.	
Secuencia Normal	Paso	Acción
	1	El profesor selecciona la práctica a descargar.
Post-condición	La práctica es descargada correctamente.	
EXCEPCIONES	Paso	Acción
	1	Los alumnos no han realizado ninguna subida de sus prácticas y por lo tanto el profesor no puede descargar ninguna práctica.
Importancia	Media.	
Frecuencia	Media.	
Urgencia	Media.	

Tabla 10: Plantilla RF-AW05 "Descargar prácticas"





RF-AW06	Comprobar plagios de prácticas	
Descripción	El profesor comprobará la existencia de plagios entre la última práctica subida por los alumnos.	
Precondición	El profesor ha seleccionado desde su panel la opción de comprobación de plagios.	
Secuencia Normal	Paso	Acción
	1	El profesor selecciona la opción de comprobación de plagios de las prácticas.
Post-condición	El reporte de plagios se crea correctamente.	
EXCEPCIONES	Paso	Acción
	1	Ningún alumno ha subido práctica.
	2	Únicamente un alumno ha subido prácticas.
Importancia	Media.	
Frecuencia	Baja.	
Urgencia	Media.	

Tabla 11: Plantilla RF-AW06 "Comprobar plagios de prácticas"



RF-AW07	Visualizar gráficas	
Descripción	El profesor puede visualizar diferentes gráficas para obtener una visión global referente a las prácticas subidas por los alumnos.	
Precondición	El profesor ha seleccionado desde su panel la opción de visualizar gráficas.	
Secuencia Normal	Paso	Acción
	1	El profesor selecciona las gráficas que desea visualizar.
	2	Selecciona la opción para que las gráficas sean generadas.
Post-condición	Las gráficas son generadas y mostradas.	
EXCEPCIONES	Paso	Acción
	1	Los alumnos no han realizado ninguna subida de sus prácticas y por lo tanto el profesor no puede descargar ninguna práctica.
Importancia	Media.	
Frecuencia	Media.	
Urgencia	Media.	

Tabla 12: Plantilla RF-AW07 "Visualizar gráficas"





- **Rol Alumno:**

RF-AW08	Subir práctica	
Descripción	El alumno podrá subir su práctica en un zip desde el correspondiente formulario de subida de ficheros.	
Precondición		
Secuencia Normal	Paso	Acción
	1	El alumno selecciona la práctica zip a subir.
	2	Selecciona la opción del formulario para subir la práctica.
Post-condición	Los resultados quedan guardados.	
EXCEPCIONES	Paso	Acción
	1	El alumno no ha seleccionado una práctica con extensión zip.
	2	La fecha límite de entrega de las prácticas ha finalizado.
	3	El alumno ha realizado ya el número máximo de intentos posibles de subida de prácticas.
	4	El alumno sube una práctica que contiene una incorrecta estructura de carpetas, que no se corresponde con el paquete correcto.
	5	El alumno sube una práctica que presenta errores de compilación.
Importancia	Alta.	
Frecuencia	Alta.	
Urgencia	Alta.	

Tabla 13: Plantilla RF-AW08 "Subir práctica"



RF-AW09	Consultar datos de la tarea	
Descripción	El alumno podrá consultar los datos/parámetros asociados a la tarea.	
Precondición	Que el profesor haya configurado los parámetros de la tarea.	
Secuencia Normal	Paso	Acción
	1	El alumno visualiza los datos de la tarea.
Post-condición	Los datos de la tarea son visualizados.	
EXCEPCIONES		
Importancia	Alta.	
Frecuencia	Alta.	
Urgencia	Alta.	

Tabla 14: Plantilla RF-AW09 "Consultar datos de la tarea"

RF-AW10	Reemplazar la práctica seleccionada	
Descripción	El alumno podrá reemplazar la práctica que había seleccionado para subir.	
Precondición	Haber seleccionado una práctica previamente.	
Secuencia Normal	Paso	Acción
	1	El alumno vuelve a seleccionar la práctica que desee subir.
Post-condición	La práctica seleccionada anteriormente queda reemplazada por la nueva.	
EXCEPCIONES	Paso	Acción
	1	Que el alumno no haya seleccionado una práctica previamente.
Importancia	Baja.	
Frecuencia	Baja.	
Urgencia	Baja.	





Tabla 15: Plantilla RF-AW10 "Reemplazar la práctica seleccionada"

RF-AW11	Consultar reportes	
Descripción	El alumno podrá ver los reportes generados pertenecientes a la última práctica subida.	
Precondición	Que el alumno haya subido por lo menos una práctica.	
Secuencia Normal	Paso	Acción
	1	El alumno selecciona el botón del reporte que desee visualizar.
Post-condición	El reporte se abre en una nueva pestaña y puede visualizarse.	
EXCEPCIONES	Paso	Acción
	1	El alumno nunca ha subido una práctica.
Importancia	Alta.	
Frecuencia	Media.	
Urgencia	Alta.	

Tabla 16: Plantilla RF-AW11 "Consultar reportes"



RF-AW12	Consultar gráficas	
Descripción	El alumno podrá ver las gráficas para observar la evolución que va teniendo en sus intentos de subida de práctica.	
Precondición	Que el alumno haya subido por lo menos una práctica.	
Secuencia Normal	Paso	Acción
	1	El alumno visualiza las gráficas.
Post-condición	Las gráficas están generadas y pueden visualizarse.	
EXCEPCIONES	Paso	Acción
	1	El alumno nunca ha subido una práctica.
Importancia	Alta.	
Frecuencia	Media.	
Urgencia	Alta.	

Tabla 17: Plantilla RF-AW12 "Consultar gráficas"





III - ESPECIFICACIÓN DE DISEÑO

A lo largo de este apartado, se describirá el diseño de la aplicación. Se dividirá en diferentes apartados en los que se analizarán aspectos relacionados con la base de datos, diagramas y diseños.

1. Diseño de datos

En este apartado van a incluirse los modelos relacionales y diagramas de clase del trabajo fin de grado.

1.1. Base de datos

1.1.1. Modelo relacional

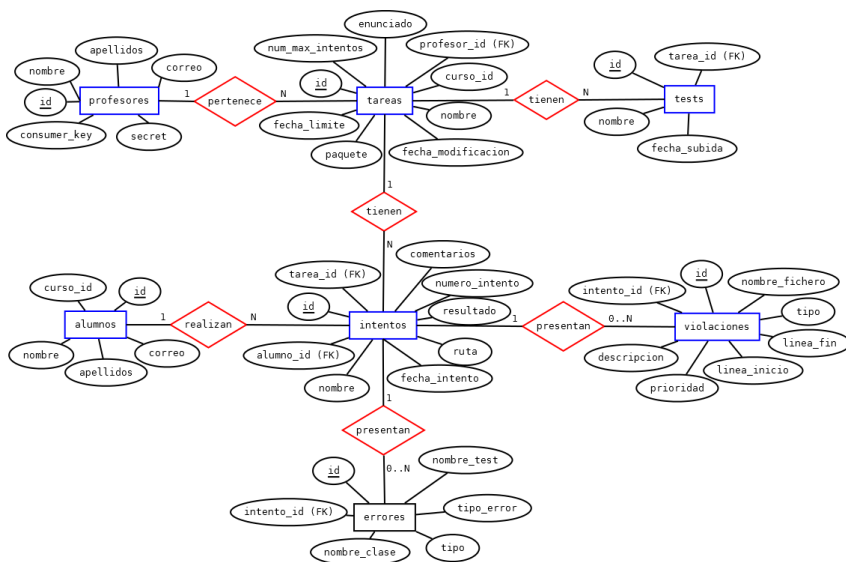


Ilustración 26: Modelo relacional



1.1.2. Diccionario de datos

Información referente a las tablas de la base de datos utilizadas en la aplicación.

- **profesores:** Información referente a los datos de cada uno de los profesores.

Atributo	Tipo	Descripción
id	INT (AUTO_INCREMENT)	Código que identifica al profesor de forma única.
nombre	VARCHAR	Nombre del profesor.
apellidos	VARCHAR	Apellidos del profesor.
consumer_key	VARCHAR	Parámetro LTI que va a introducirse al crear la tarea de tipo externa en Moodle.
secret	VARCHAR	Parámetro LTI que va a introducirse al crear la tarea de tipo externa en Moodle.
correo	VARCHAR	Correo electrónico del profesor, se corresponde con el de Moodle.

Tabla 18: Tabla profesores

* *Primary key (PK):* id.

- **alumnos:** Contiene la información referente a los alumnos que han accedido a la aplicación web.

Atributo	Tipo	Descripción
id	INT	Código que identifica al alumno de forma única, se corresponde con el de Moodle.
curso_id	INT	Código que identifica al curso en el que se encuentra creada la tarea.
nombre	VARCHAR	Nombre del alumno, se corresponde con el de Moodle.
apellidos	VARCHAR	Apellidos del alumno, se corresponde con el de Moodle.
correo	VARCHAR	Correo electrónico del alumno, se corresponde con el de Moodle.

Tabla 19: Tabla alumnos





* *Primary key (PK)*: id.

- **tareas**: Contiene la información referente a las tareas que van a ser creadas desde Moodle de tipo “herramienta externa”. A dichas tareas son a las que los profesores y alumnos van a acceder desde Moodle:

Atributo	Tipo	Descripción
id	INT	Código que identifica a la tarea de forma única, se corresponde con en el de Moodle.
curso_id	INT	Código que identifica al curso en el que se encuentra creada la tarea.
profesor_id	INT	Código que identifica al profesor asociado a la tarea.
nombre	VARCHAR	Nombre de la tarea, se corresponde con el de Moodle.
num_max_intentos	INT	Número de intentos máximo establecido por el profesor, que tienen los alumnos para subir sus prácticas.
paquete	VARCHAR	Nombre del paquete al que pertenece la tarea. Tanto los test que suban los profesores como las prácticas que suban los alumnos, deben de pertenecer a este paquete.
enunciado	TEXT	Enunciado de la tarea.
fecha_limite	DATETIME	Fecha límite que tienen los alumnos para entregar sus prácticas referentes a la tarea.
fecha_modificacion	DATETIME	Fecha de modificación de la tarea.

Tabla 20: Tabla tareas

* *Primary key (PK)*: id.

* *Foreign key (FK)*:

- profesor_id -> FK hacia el campo “id” de la tabla “profesores”.



- **tests:** Contiene la información de cada uno de los test que han subido los profesores sobre las tareas.

Atributo	Tipo	Descripción
id	INT	Código que identifica al test de forma única.
tarea_id	INT	Código que identifica a la tarea asociada.
nombre	VARCHAR	Nombre del test.
fecha_subida	DATETIME	Fecha en la que ha sido subido el test.

Tabla 21: Tabla tests

* *Primary key (PK):* id.

* *Foreign key (FK):*

- tarea_id -> FK hacia el campo “id” de la tabla “tareas”.





- **intentos:** Contiene la información de cada uno de los intentos de subida de prácticas realizadas por los alumnos sobre las tareas.

Atributo	Tipo	Descripción
id	INT	Código que identifica el intento de forma única.
tarea_id	INT	Código que identifica a la tarea asociada.
alumno_id	INT	Código que identifica al alumno asociado.
nombre	VARCHAR	Nombre del zip que contiene la práctica.
numero_intento	INT	Número del intento de subida de la práctica.
resultado	TINYINT(1)	Indica si el intento realizado ha pasado los test o no.
comentarios	TEXT	Comentarios referentes a la práctica subida.
fecha_intento	DATETIME	Fecha en la que se ha realizado el intento de subida de la práctica.

Tabla 22: Tabla intentos

* *Primary key (PK):* id.

* *Foreign key (FK):*

- tarea_id -> FK hacia el campo "id" de la tabla "tareas".
- alumno_id -> FK hacia el campo "id" de la tabla "alumnos".



- **violaciones:** Contiene la información de las violaciones de código que se han cometido en el intento de subida de la práctica.

Atributo	Tipo	Descripción
id	INT	Código que identifica la violación de forma única.
intento_id	INT	Código que identifica al intento de subida de práctica asociada.
nombre_fichero	VARCHAR	Nombre del fichero .java en el que se ha cometido la violación de código.
tipo	VARCHAR	Tipo de violación de código.
descripcion	VARCHAR	Descripción de la violación de código.
prioridad	INT	Indica la prioridad de la violación de código.
línea_inicio	INT	Línea de inicio del fichero .java en el que se ha cometido la violación de código.
línea_fin	INT	Línea de fin del fichero .java en el que se ha cometido la violación de código.

Tabla 23: Tabla violaciones

* *Primary key (PK)*: id.

* *Foreign key (FK)*:

- intento_id -> FK hacia el campo “id” de la tabla “intentos”.





- **errores:** Contiene la información de los errores que se han cometido en el intento de subida de práctica. Estos errores pueden ser errores unitarios o excepciones.

Atributo	Tipo	Descripción
id	INT	Código que identifica al error de forma única.
intento_id	INT	Código que identifica al intento de subida de práctica asociada.
nombre_clase	VARCHAR	Nombre del fichero .java en el que se ha cometido el error.
nombre_test	VARCHAR	Nombre del test en el que se ha cometido el error.
tipo_error	VARCHAR	Indica qué tipo de error se ha cometido, es decir: unitario (failure) o por excepción (error).
tipo	VARCHAR	Descripción el error cometido.

Tabla 24: Tabla errores

* *Primary key (PK):* id.

* *Foreign key (FK):*

- intento_id -> FK hacia el campo “id” de la tabla “intentos”.



1.2. Diagrama de paquetes

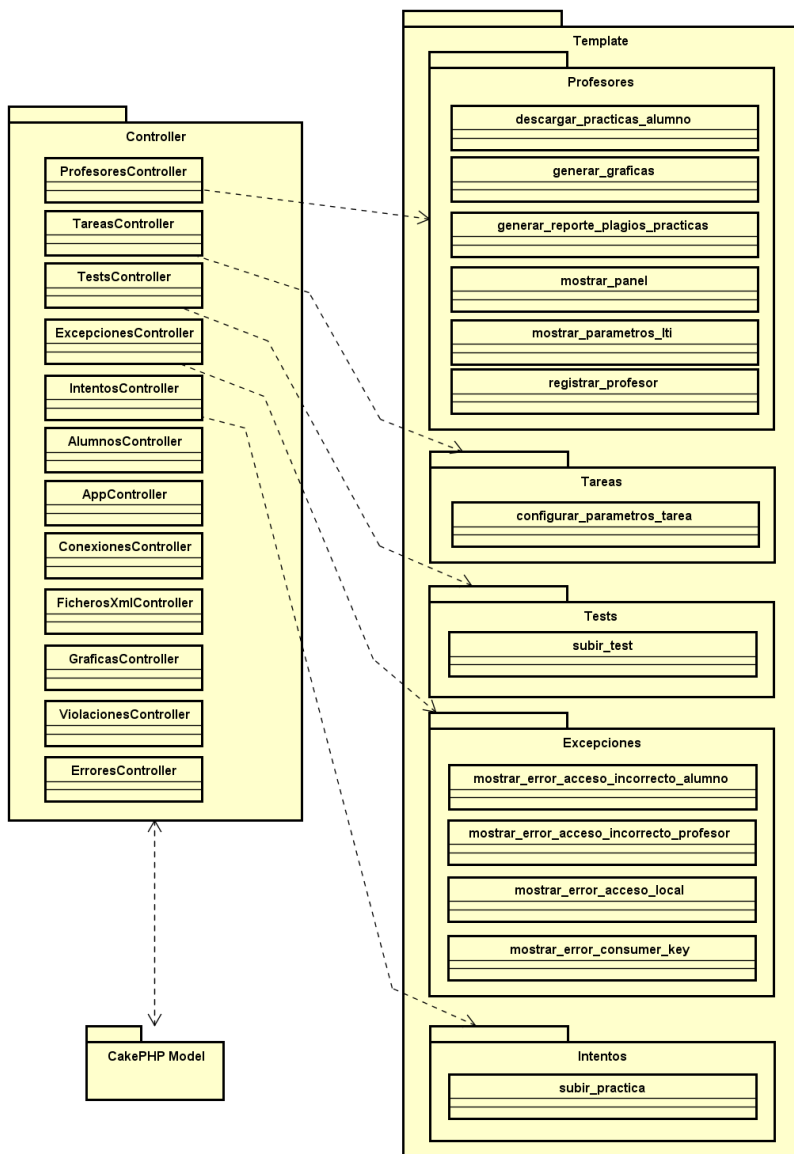


Ilustración 27: Diagrama de paquetes





Explicación del diagrama de paquetes

CakePHP proporciona por defecto una estructura de carpetas que va a permitir implementar el patrón de Modelo Vista Controlador. En esta estructura de carpetas se encuentran las siguientes carpetas:

- **Controller:** Dentro de la cual van a estar todos los controladores. Se trata de clases PHP que implementan la mayor parte de la funcionalidad de la aplicación.
- **Template:** Contiene las vistas, cuyo funcionamiento viene establecido por el *framework* CakePHP. Si un controlador tiene vistas, primero hay que crear una subcarpeta dentro de esta carpeta, cuyo nombre debe de ser el del controlador. Por ejemplo si “ProfesoresController” tiene vistas, hay que crear una subcarpeta “Profesores” dentro de la carpeta “Template”, y dentro de la subcarpeta “Profesores” se crearían las vistas de extensión .ctp [5].

Además, si la vista creada recibe por ejemplo el nombre de “generar_graficas.ctp”, dentro del controlador “ProfesoresController” deberá de existir un método público o *action* con nombre “generarGraficas” que esté ligado a esa vista [6].

Respecto al modelo, no ha sido necesario crear entidades debido a que CakePHP nos lo facilita y permite evitar el tener que crear una clase para cada entidad. Por ello gracias al método “newEntity()” de CakePHP la creación de entidades para la tabla que queramos es un proceso sencillo y rápido de hacer [7].



1.3. Diagrama de clases

Para mejor visualización, ver foto del mismo en la carpeta “Documentación” del DVD.

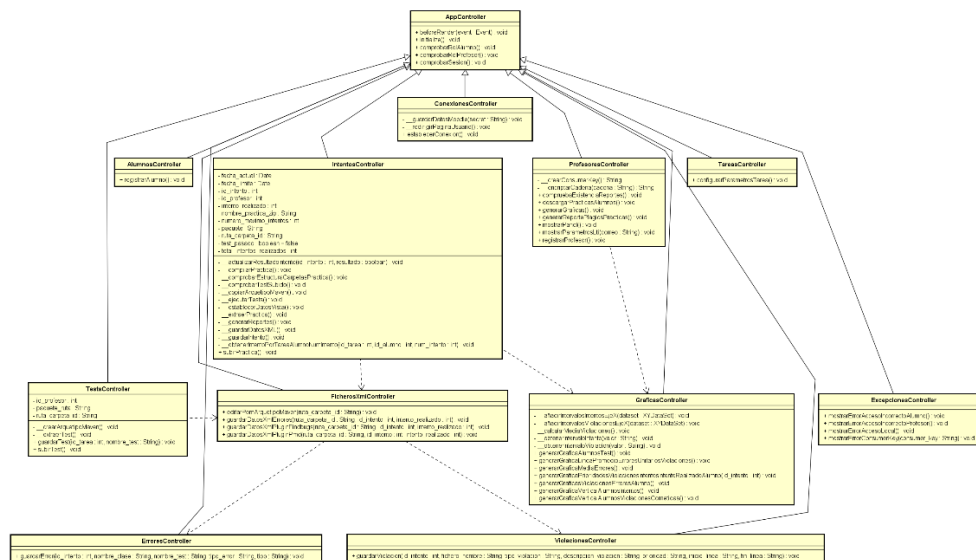


Ilustración 28: Diagrama de clases



Explicación del diagrama de clases

La explicación del diagrama de clases y la función que tiene cada uno de los controladores dentro de aplicación sería la siguiente:

En primer lugar, todos los controladores heredan de “AppController”. Las peticiones web hacia la aplicación siempre se reciben por la clase “ConexionesController”, que es el controlador que en función del rol del usuario que haya accedido a la aplicación va a re direccionar (para ello se utiliza una función de CakePHP que permite hacerlo sin necesidad de instanciar el controlador al que se va a re direccionar) a uno de los 4 controladores que están por debajo:

- Si el que accede a la aplicación es un **profesor** entonces:
 - a. Si es su primer acceso, será redirigido a “TareasController” para que rellene el formulario obligatorio de configuración de parámetros de la tarea.
 - b. En los siguientes accesos, será redirigido al panel principal del profesor que se encuentra en “ProfesoresController”.
- Si el que accede a la aplicación es un **alumno** entonces:
 - a. Si es su primer acceso, será redirigido a “AlumnosController” donde se procederá a su registro en la aplicación.
 - b. En los siguientes accesos será redirigido al panel principal del alumno, que es el formulario de subida de prácticas, perteneciente a “IntentosController”.

Por lo tanto por parte del profesor “ProfesoresController” sería el controlador principal, y por parte del alumno sería “IntentosController”. A partir de ahora estos controladores serán los que usen los controladores que se encuentran por debajo cuando sea necesario. Por ejemplo: tanto “IntentosController” como “ProfesoresController” van a hacer uso del controlador “GraficasController” para generar las gráficas necesarias.

2. Diseño procedimental

2.1. Diagramas de secuencias

- Diagrama de secuencia: “Conexión del usuario (profesor/alumno)”

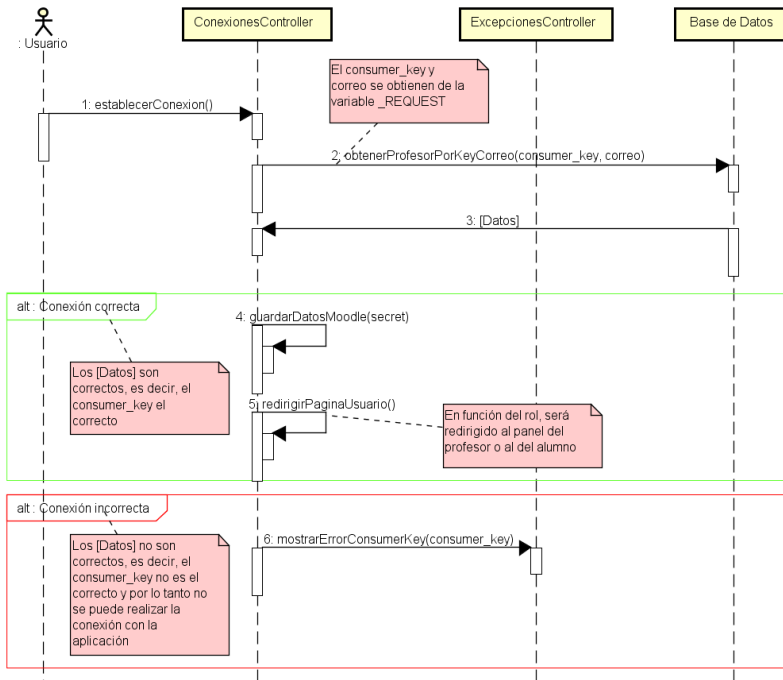


Ilustración 29: Diagrama de secuencia "Conexión del usuario"





- Diagrama de secuencia: “Registro del profesor en la aplicación”

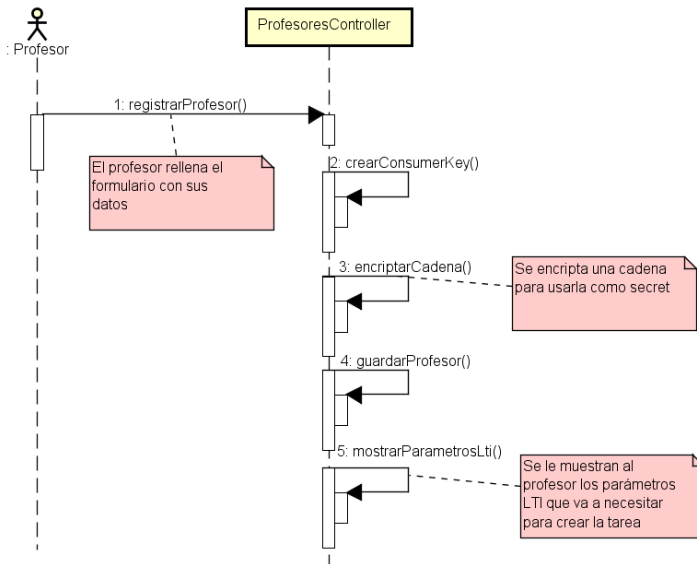


Ilustración 30: Diagrama de secuencia "Registro del profesor en la aplicación"

- Diagrama de secuencia: “Profesor configura parámetros de la tarea”

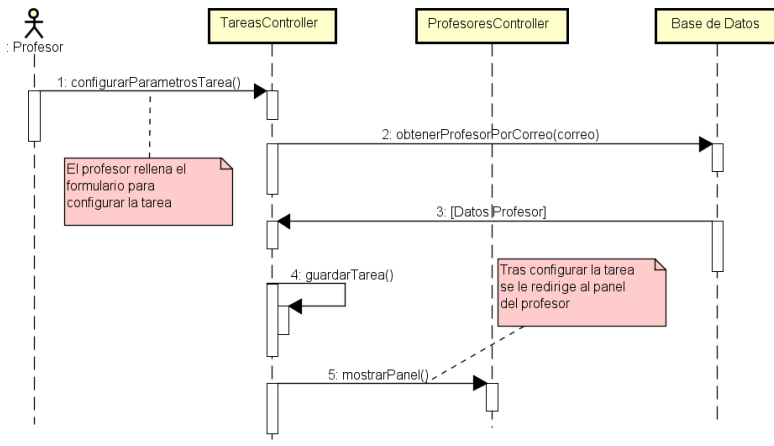


Ilustración 31: Diagrama de secuencia "Profesor configura parámetros tarea"



- Diagrama de secuencia: “Profesor sube test”

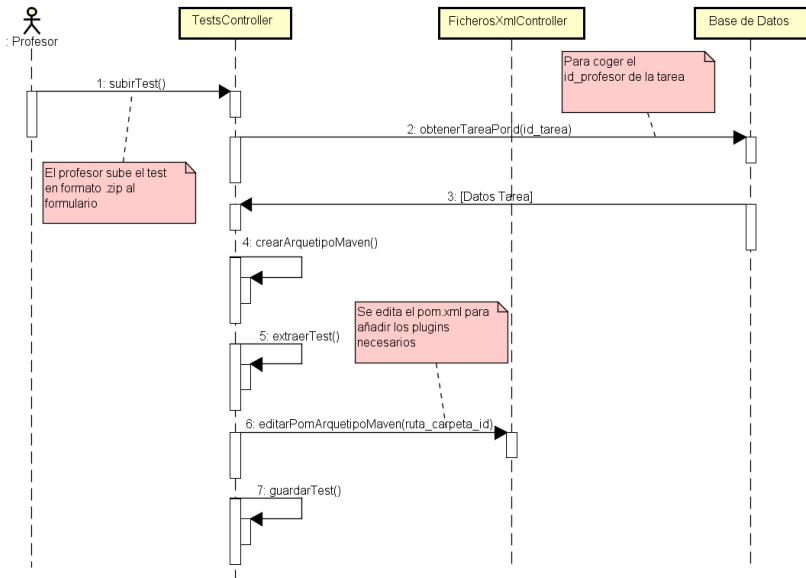


Ilustración 32: Diagrama de secuencia "Profesor sube test"





- Diagrama de secuencia: “Alumno sube práctica”

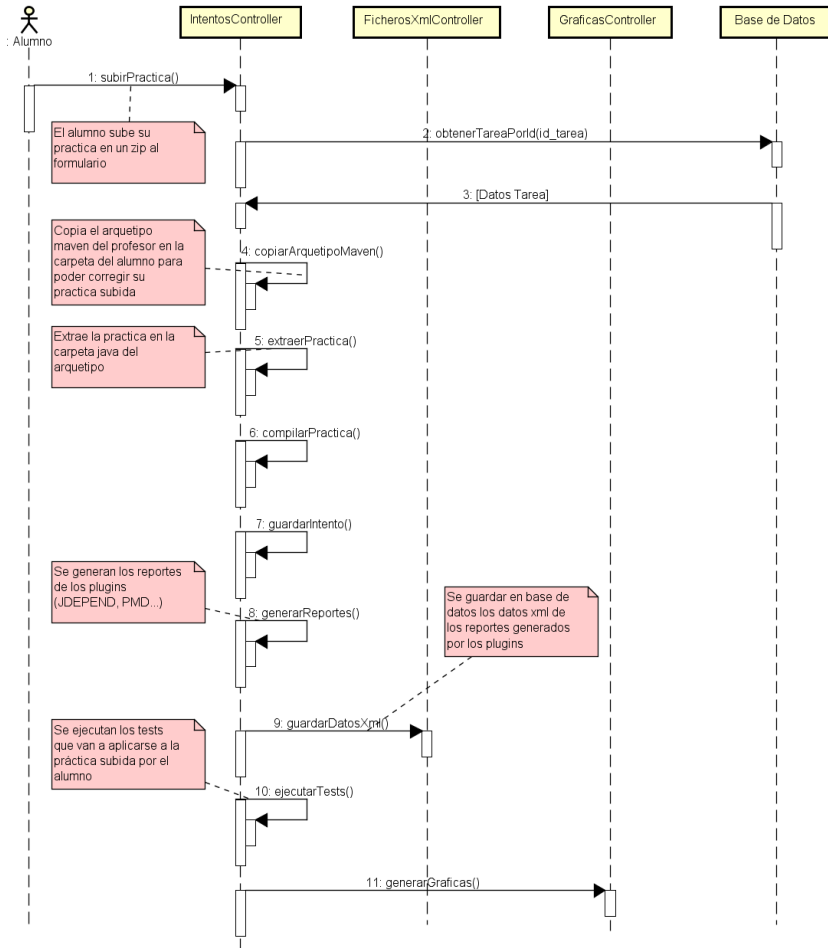


Ilustración 33: Diagrama de secuencia "Alumno sube práctica"



- Diagrama de secuencia: “Alumno sube práctica incorrecta”

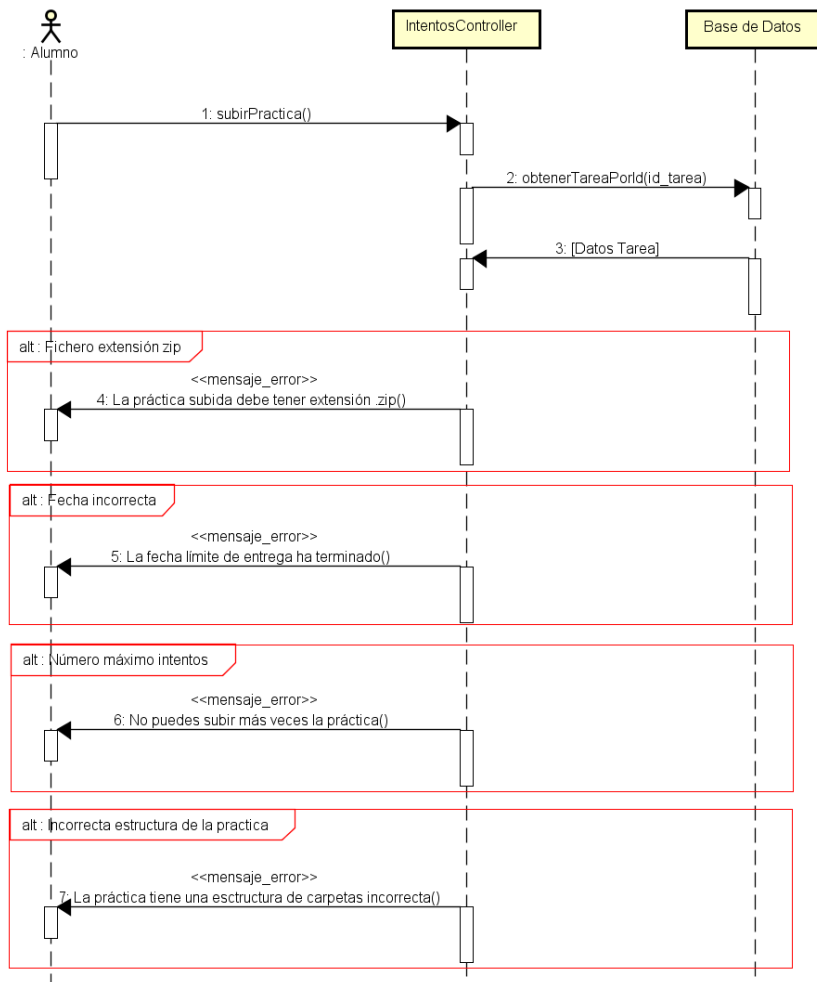


Ilustración 34: Diagrama de secuencia "Alumno sube práctica incorrecta"





3. Diseño arquitectónico

3.1. Diagrama de despliegue

En el diagrama de despliegue podemos observar cómo interactúan las diferentes partes que forman el sistema:

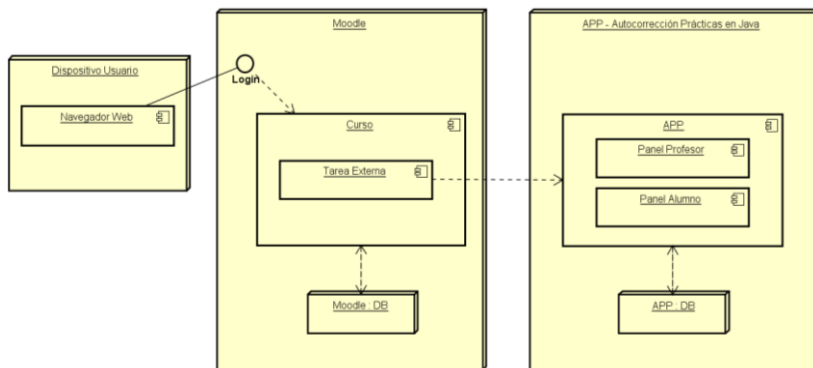


Ilustración 35: Diagrama de despliegue



IV - MANUAL DEL PROGRAMADOR

1. *Introducción*

En este apartado se va a tratar todo lo referente a los aspectos técnicos, necesarios para un programador, relacionados con la programación del trabajo fin de grado.

Dentro del mismo se explicará la estructura de directorios, estructura del código fuente, instalación de herramientas y las pruebas realizadas sobre la aplicación.

2. *Estructura de directorios*

En el DVD entregado, se encuentra la siguiente estructura de carpetas:

- **Código fuente:** contiene el código fuente de la aplicación.
- **Documentación:** contiene la memoria, anexos y una foto del diagrama de clases.
- **Pruebas:** contiene los archivos .bat encargados de ejecutar las pruebas unitarias y de cobertura de código.
- **Vídeos:** contienen los vídeos de la aplicación en funcionamiento, diferenciando los roles de profesor y alumno en dos subcarpetas.
- **Base de datos:** contiene el script sql de creación de las tablas de la base de datos.
- **Máquina virtual:** contiene un zip con la imagen de la máquina virtual.

3. *Estructura del código fuente*

En esta sección se va a explicar brevemente los directorios que componen el código fuente que se ha utilizado para crear la aplicación.

En primer lugar decir que CakePHP genera una estructura de carpetas por defecto al configurar un nuevo proyecto, y por lo tanto hay muchas carpetas que no han sido empleadas por el desarrollador directamente. El directorio raíz de la aplicación sería el siguiente:



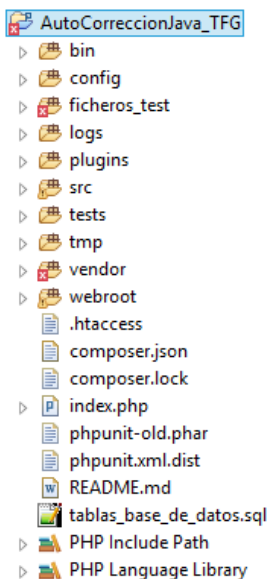


Ilustración 36: Directorio raíz de la aplicación

Las carpetas que no han sido empleadas por el desarrollador directamente son: bin, logs, plugins y tmp.

- Directorio AutoCorreccionJava_TFG/**config**:

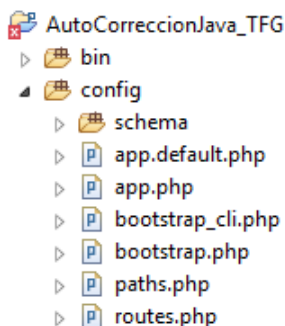


Ilustración 37: Directorio “config” de la aplicación

Dentro de este directorio el único fichero modificado ha sido el de “app.php”, el cual ha sido necesario configurar para poder establecer conexión con la base de datos utilizada en la aplicación, además de la base de datos adicional que se ha utilizado para las pruebas unitarias.



- Directorio AutoCorreccionJava_TFG/**ficheros_test**:

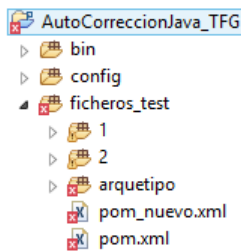


Ilustración 38: Directorio "ficheros_test" de la aplicación

Dentro de esta carpeta van a estar una serie de ficheros que van a usarse para poder realizar algunas pruebas unitarias, ya que hay controladores que tienen métodos que realizan procesos de lectura y escritura sobre ficheros xml.

- Directorio AutoCorreccionJava_TFG/**src/Controller**:

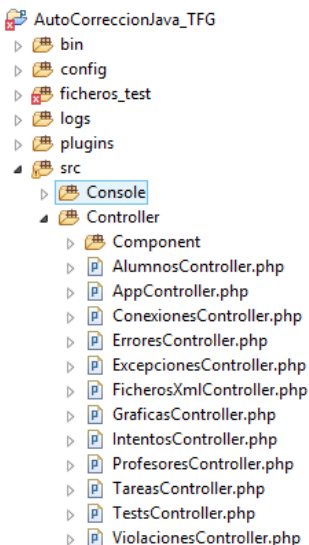


Ilustración 39: Directorio "src/Controller" de la aplicación

Contienen los diferentes controladores utilizados para desarrollar la aplicación. Estos controladores van a interactuar tanto con la vista como con el modelo.





- Directorio AutoCorreccionJava_TFG/src/Template:

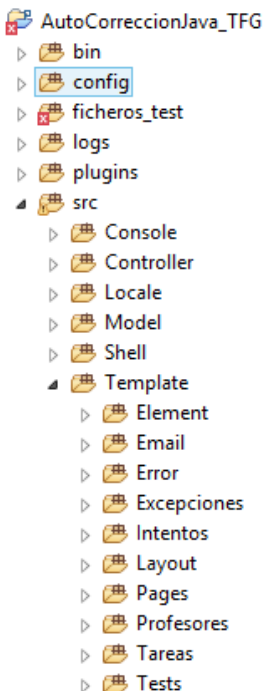


Ilustración 40: Directorio "src/Template" de la aplicación

Contiene las subcarpetas con las vistas que se corresponden con cada uno de los controladores. Es decir, dentro de la subcarpeta "Excepciones" van a estar las vistas de extensión .ctp que se corresponden con el controlador "ExcepcionesController".

Los ficheros de extensión .ctp básicamente son ficheros HTML en los que se reservan partes para ser rellenadas por medio de datos proporcionados por los ficheros PHP.

De las subcarpetas presentes, las que no han sido empleadas directamente por el desarrollador y que vienen por defecto al usar CakePHP son: Element, Email, Error y Pages.



- Directorio AutoCorreccionJava_TFG/**src/Model**:

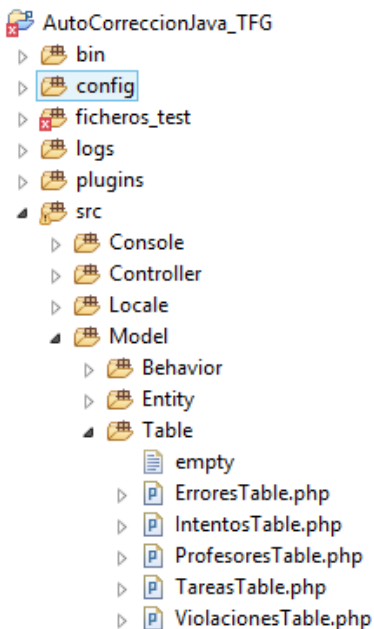


Ilustración 41: Directorio "src/Model" de la aplicación

Como se ha explicado en el apartado (Ver Diagrama de paquetes), CakePHP permite evitar el tener que crear unas clases entidad con las que poder interactuar con la base de datos.

Por ello dentro de esta carpeta la única subcarpeta utilizada es la de "Table", la cual contiene diferentes ficheros php los cuáles van a encargarse de realizar la validación de los datos introducidos en los campos de los formularios y de establecer relaciones entre las tablas de la base de datos para posteriormente poder realizar consultas JOIN desde los controladores.





- Directorio AutoCorreccionJava_TFG/src/Locale:

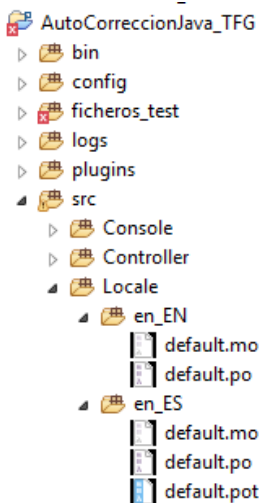


Ilustración 42: Directorio "src/Locale" de la aplicación

Contiene los ficheros .po que van a permitir realizar la Internacionalización de la aplicación.



- Directorio AutoCorreccionJava_TFG/**vendor**:

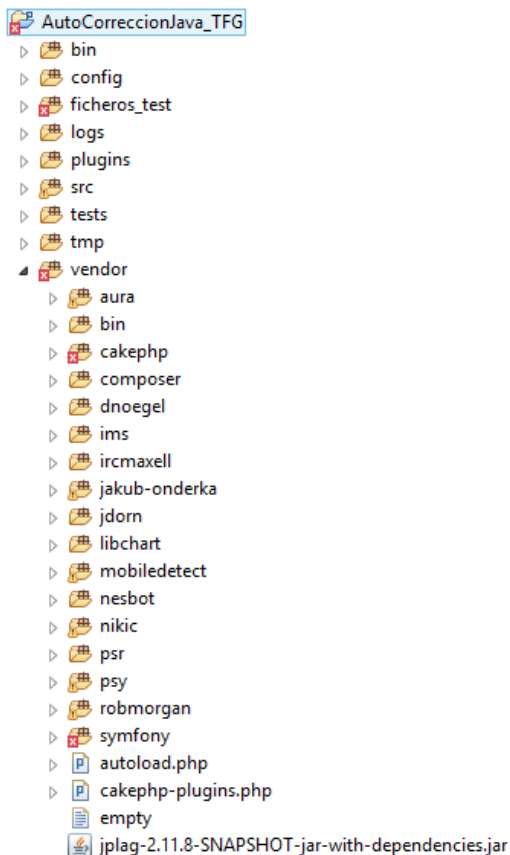


Ilustración 43: Directorio "vendor" de la aplicación

Dentro de esta carpeta van a estar las librerías externas, de las cuales las únicas que han sido añadidas son: “ims” [8], encargada de establecer la conexión correcta entre Moodle y la aplicación, además de permitir la obtención de los datos de Moodle, “libchart” [9], librería utilizada para la generación de gráficas y “jplag-2.11.8-SNAPSHOT-jar-with-dependencies.jar” [10], utilizado para poder generar el reporte de plagios de las prácticas.





- Directorio AutoCorreccionJava_TFG/**webroot**:

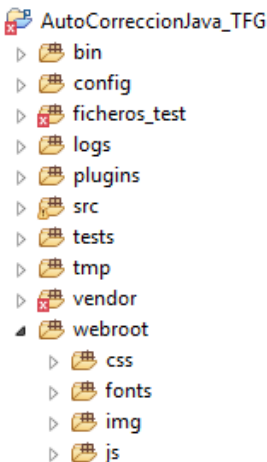


Ilustración 44: Directorio "webroot" de la aplicación

Contiene las carpetas con los ficheros css, javascript y las diferentes imágenes utilizadas en la aplicación.

El lenguaje javascript se ha utilizado para diferentes funcionalidades dentro de la aplicación: para las imágenes de información y enunciado de la práctica, para la barra de progreso que se muestra tras subir un fichero en los formularios, etc.

4. Instalación de herramientas

4.1. Sistema operativo

La aplicación ha sido desarrollada desde un portátil con sistema operativo: Windows 8.1 (64 bits).

4.2. Servidor

Se ha utilizado el programa WampServer en la versión 2.5 y arquitectura de 64 bits. La instalación contiene:

- Apache (versión 2.4.9)
- MySQL (versión 5.16.17)
- PHP (versión 5.5.12)
- PHPMyAdmin (versión 4.1.14)

A continuación se va a mostrar de forma detallada el proceso de instalación:



1. Acceder a la web oficial del WampServer y descargar la versión de 64 bits:

<http://www.wampserver.com/en/#download-wrapper>

2. Ejecutamos el fichero descargado y se nos mostrará la siguiente pantalla:

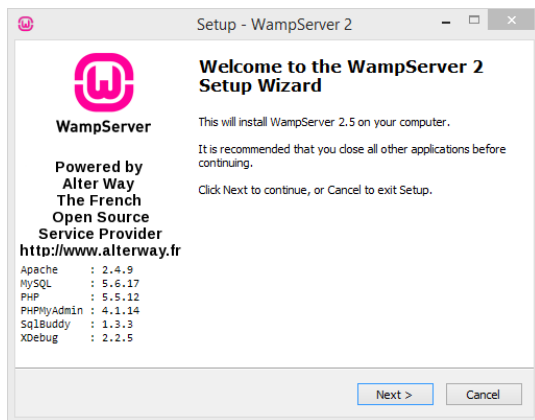


Ilustración 45: Instalar WampServer paso 1

3. Presionamos “Next”, y continuación se nos mostrará la siguiente pantalla:

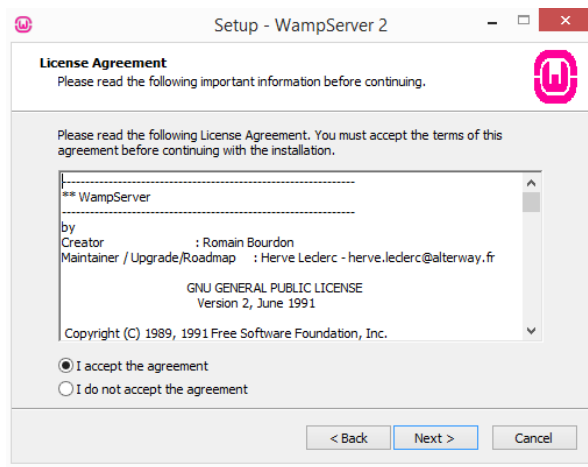


Ilustración 46: Instalar WampServer paso 2





4. Aceptamos el acuerdo y presionamos “Next”. Nos aparecerá la siguiente pantalla:

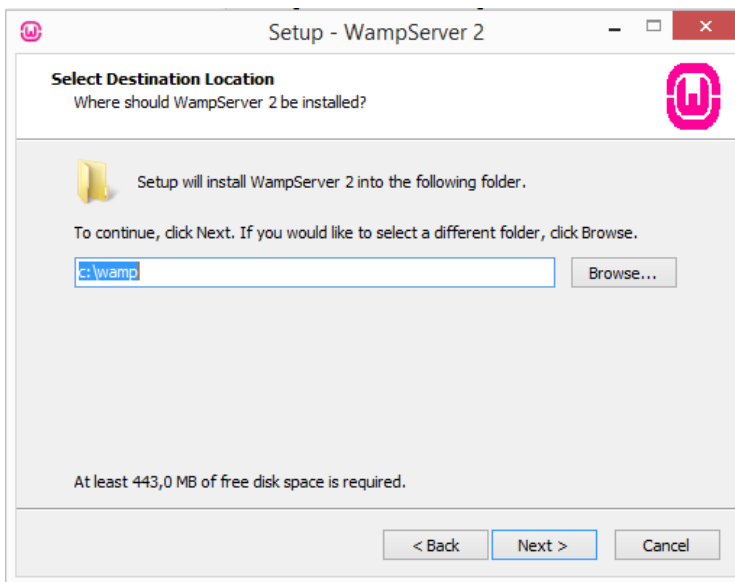


Ilustración 47: Instalar WampServer paso 3

5. Seleccionamos la carpeta en la que se instalará WampServer y pulsamos “Next” hasta que nos aparezca la opción de “Install”, en cuyo caso habrá finalizado el proceso de instalación y de esta forma ya tendríamos el Servidor y la base de datos MySQL instalada.

4.3. Base de datos

4.3.1. Creación de las tablas

Para realizar el proyecto va a ser necesario crear la base de datos de nombre “autocorrecc_tfg”.

Gracias al script .sql, presente en la carpeta “Base de datos” del DVD, que se tiene creado con las diferentes tablas de la base de datos, el proceso de la creación de las mismas va a ser sencillo.

Para ello los pasos a seguir serían los siguientes:



1. Abrimos el programa de gestión de base de datos que queramos (en nuestro caso se ha utilizado HeidiSQL), nos conectamos a nuestra base de datos “autocorrecc_tfg” y la seleccionamos.
2. A continuación copiamos todo el contenido de nuestro .sql y lo copiamos en la pestaña “Consulta” presente en el HeidiSQL. El aspecto sería parecido al siguiente:

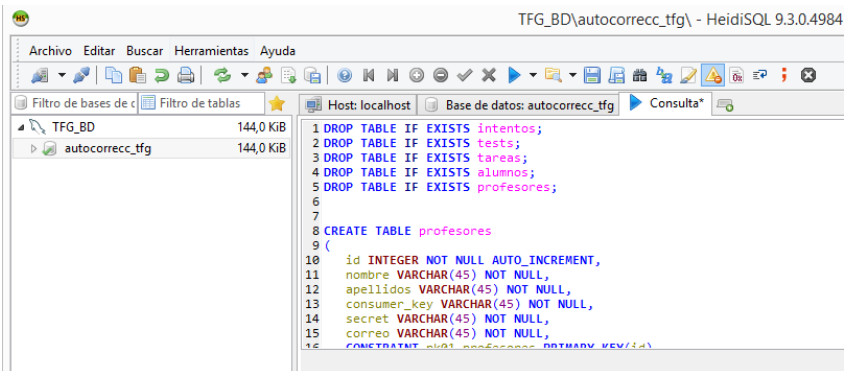


Ilustración 48: HeidiSQL sql creación de tablas

3. Por último presionando la tecla “F9” o pinchando en el botón azul que tiene forma de “play”, se crearán las tablas sobre nuestra base de datos.





4.4. Java JDK

Pasos para descargarse el JDK de Java:

1. Acceder al siguiente enlace:

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html?ssSourceSiteId=otnes>



Java SE Development Kit 8 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

See also:

- Java Developer Newsletter: From your Oracle account, select **Subscriptions**, expand **Technology**, and subscribe to **Java**.
- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

JDK 8u71 Checksum
JDK 8u72 Checksum

Java SE Development Kit 8u71		
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.		
<input type="radio"/> Accept License Agreement <input checked="" type="radio"/> Decline License Agreement		
Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.71 MB	jdk-8u71-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.65 MB	jdk-8u71-linux-arm64-vfp-hflt.tar.gz
Linux x86	154.75 MB	jdk-8u71-linux-i586.rpm
Linux x86	174.91 MB	jdk-8u71-linux-i586.tar.gz
Linux x64	152.74 MB	jdk-8u71-linux-x64.rpm
Linux x64	172.9 MB	jdk-8u71-linux-x64.tar.gz
Mac OS X	227.24 MB	jdk-8u71-macosx-x64.dmg
Solaris SPARC 64-bit	139.78 MB	jdk-8u71-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.05 MB	jdk-8u71-solaris-sparcv9.tar.gz
Solaris x64	139.98 MB	jdk-8u71-solaris-x64.tar.Z
Solaris x64	96.19 MB	jdk-8u71-solaris-x64.tar.gz
Windows x86	181.21 MB	jdk-8u71-windows-i586.exe
Windows x64	186.55 MB	jdk-8u71-windows-x64.exe

Ilustración 49: Java JDK descarga

2. Aceptamos los acuerdos de licencia y seleccionamos la opción Windows x64.
3. Procedemos a instalar el ejecutable descargado.
4. Y por último, establecemos una nueva variable de sistema de nombre "JAVA_HOME" que apunte al directorio jdk de Java.



4.5. Maven

Para descargar e instalar Maven los pasos a seguir son los siguientes:

1. Accedemos a su web oficial desde la siguiente URL:

<https://maven.apache.org/>

2. Accedemos a la sección “Download”, después al apartado “Files” y seleccionamos el link del fichero binario de extensión .zip:

Files

Maven is distributed in several formats for your convenience. Simply pick a ready-made binary distribution archive and follow the [installation instructions](#). Use a source archive if you intend to build Maven yourself.

In order to guard against corrupted downloads/installations, it is highly recommended to [verify the signature](#) of the release bundles against the public [KEYS](#) used by the Apache Maven developers.

	Link	Checksum	Signature
Binary tar gz archive	apache-maven-3.3.3-bin.tar.gz	apache-maven-3.3.3-bin.tar.gz.md5	apache-maven-3.3.3-bin.tar.gz.asc
Binary zip archive	apache-maven-3.3.3-bin.zip	apache-maven-3.3.3-bin.zip.md5	apache-maven-3.3.3-bin.zip.asc
Source tar gz archive	apache-maven-3.3.3-src.tar.gz	apache-maven-3.3.3-src.tar.gz.md5	apache-maven-3.3.3-src.tar.gz.asc
Source zip archive	apache-maven-3.3.3-src.zip	apache-maven-3.3.3-src.zip.md5	apache-maven-3.3.3-src.zip.asc

Ilustración 50: Maven descarga

3. Por último extraemos el zip en C:\ y establecemos en las variables de entorno del sistema (Path) la ruta a la carpeta \bin del Maven.

4.6. Plugins Reportes

Para generar los reportes que van a mostrarse al alumno tras la subida de sus prácticas, se han empleado diferentes *plugins* que van a integrarse dentro del arquetipo Maven.

Estos *plugins* son los siguientes:

- **JAVANCSS:** Ofrece información sobre métricas (complejidad ciclomática, número de líneas...) [11].
- **JDEPEND:** Ofrece información sobre métricas a nivel de paquetes (abstractness, inestability, afferent couplings...) [12].
- **PMD:** Ofrece información acerca de violaciones de código cometidas (variables sin usar, métodos privados sin usar...) [13].
- **FINDBUGS:** Al igual que el PMD ofrece información acerca de violaciones de código, pero detecta violaciones que el PMD no detecta (por ejemplo, bucles infinitos) [14].





Y se integrarán dentro del fichero “pom.xml” presente en el arquetipo Maven.

4.7. CakePHP

El uso del framework CakePHP va a permitirnos implementar el patrón MVC (Modelo-Vista-Controlador) de una forma más sencilla.

El proceso de instalación es el siguiente:

1. Accedemos a su web oficial desde la siguiente URL:

<http://cakephp.org/>

2. Seleccionamos la opción “Download” y se nos redirigirá a la página de la guía de instalación. A continuación simplemente tendremos que seguir los pasos que se nos van marcando: en primer lugar instalaremos el “composer” y posteriormente ya podremos crear la aplicación cake php utilizando el composer. El aspecto de esta guía de instalación es el siguiente:

Ilustración 51: CakePHP Instalación



4.8. Moodle

La aplicación se comunica con cualquier LMS que cumpla el estándar LTI. Para probar la aplicación se utilizó Moodle (versión 2.9.2), y a continuación se va a explicar el proceso de instalación del mismo:

1. Accedemos a la web oficial de Moodle y descargamos el archivo zip correspondiente a la versión 2.9.2:

<https://download.moodle.org/releases/latest/>

2. Copiamos el fichero descomprimido en la siguiente carpeta:
“C:\wamp\www”
3. Abrimos el navegador e introducimos la siguiente URL para comenzar el proceso de instalación de Moodle:
“localhost/moodle”
4. Se nos mostrará la siguiente pantalla:

Seleccionar idioma

Por favor, seleccione un idioma para el proceso de instalación. Este idioma se usará también como idioma por defecto del sitio, si bien puede cambiarse más adelante.

Idioma

Siguiente »



Ilustración 52: Moodle instalación seleccionar idioma

5. Seleccionaremos “Siguiente” dejando todos los valores que vienen por defecto hasta que aparece la siguiente pantalla, en la cual rellenaremos los datos de la siguiente forma:





Ajustes de base de datos

MySQL mejorado (native/mysqli)

Ahora tiene que configurar la base de datos donde se almacenarán la mayoría de los datos de Moodle. La base de datos solo podrá crearse si el usuario de la base de datos tiene los permisos necesarios. El nombre de usuario y la contraseña ya deben existir. El prefijo de la tabla es opcional.

Servidor de la base de datos

Nombre de la base de datos

Usuario de la base de datos

Contraseña de la base de datos

Prefijo de tablas

Puerto de la base de datos

« Anterior **Siguiente »**

Ilustración 53: Moodle instalación ajuste base de datos

6. Pinchamos “Siguiente”, después “Continuar” y el proceso de instalación de Moodle comenzará.
7. Una vez instalado, se nos mostrará la siguiente pantalla donde rellenaremos los campos de la siguiente forma:

Nombre de usuario*

Escoger un método de identificación: ☒ Cuentas manuales

La contraseña debería tener al menos 8 carácter(es), al menos 1 dígito(s), al menos 1 minúscula(s), al menos 1 mayúscula(s), al menos 1 carácter(es) no alfanuméricos

Nueva contraseña* ☐ Desenmascarar

Forzar cambio de contraseña ☐

Nombre*

Apellidos*

Dirección de correo*

Ilustración 54: Moodle instalación datos admin



8. Continuamos el proceso de instalación y por último nos piden los nombre para la página principal, los cuales serán rellenados de la siguiente forma:
 - Nombre completo del sitio: TFG Auto-corrección prácticas en Java
 - Nombre corto del sitio: TFG
9. La instalación de Moodle estaría ya realizada.

4.9. Plugin LTI

La instalación de dicho *plugin* va a ser necesaria para permitir una comunicación bidireccional entre Moodle y el servidor donde van a ejecutarse las pruebas.

El proceso a llevar a cabo para instalar este *plugin* LTI es el siguiente:

1. Accedemos a la web oficial de Moodle y desde el apartado plugins buscamos y descargamos “LTI Provider” con la versión 2.7.1.

https://moodle.org/plugins/pluginversions.php?plugin=local_ltiprovider

2. Descomprimos el zip descargado y le colocamos dentro del siguiente directorio:

“C:\wamp\www\moodle\local”

3. Por último accedemos a Moodle y entramos en el apartado de “Administración > Administración del sitio > Notificaciones”, y pinchamos en “Compruebe actualizaciones disponibles”. En unos minutos nos aparecerá que hay una actualización del “LTI Provider” disponible la cual va a permitir finalizar con la instalación del plugin.

4.10. PHPUnit

No ha sido necesario instalar PHPUnit, sino que simplemente se ha descargado el fichero .phar correspondiente a PHPUnit [15].

4.11. Instalación de un IDE

La edición del código de la aplicación no requiere de ningún IDE en concreto, a pesar de ello se ha usado Eclipse for PHP Developers [16].





5. Conexión LTI

Uno de los aspectos más destacados e importantes de la aplicación es que debe poder conectarse con la plataforma Moodle (u otra que respete el estándar LTI), de manera que se ofrezca un servicio externo a Moodle pero que se integre en él como si fuera cualquier otra tarea.

Esta conexión entre Moodle y la aplicación va a poder realizarse gracias al *plugin* LTI que ofrece la plataforma de Moodle, el cual previamente ha tenido que ser instalado. De esta forma, la aplicación va a poder recibir peticiones desde Moodle y además obtener los diferentes datos de Moodle (para esto se usa una librería en el código) que van a ser necesarios para hacer funcionar la aplicación: id de la tarea, id del curso, id del usuario, nombre del usuario...

Para que el profesor pueda crear una tarea de tipo “herramienta externa” en Moodle, va a necesitar en primer lugar unos parámetros LTI que van a tener que ser establecidos en el momento de creación de dicha tarea. Estos parámetros LTI van a ser los siguientes:

- URL.
- *Consumer key*.
- *Secret*.

Estos parámetros los obtendrá el profesor en el momento que este se registre en la aplicación rellenando el correspondiente formulario. Una vez el profesor rellena el formulario con sus datos, se le redirigirá a una nueva página en la que se le van a ofrecer los anteriores parámetros LTI.

Es importante que el profesor introduzca correctamente estos parámetros a la hora de crear la tarea desde Moodle ya que si no, ni él ni los alumnos podrán acceder a la aplicación.

Los parámetros LTI que le han sido entregados al profesor, primeramente van a ser guardados en la correspondiente tabla de la base de datos. Por ello cuando un profesor/alumno intente acceder a la aplicación, siempre en primer lugar se va a verificar que los parámetros LTI establecidos en la tarea de Moodle se correspondan con los correctos que se encuentran en la base de datos, ya que si no son los correctos no podrá realizarse dicho acceso a la aplicación.

Este proceso de verificación de los parámetros LTI se va a realizar con ayuda de una librería [8], desde el controlador *ConexionesController* y el proceso es el siguiente:

1. La comprobación de que el *consumer key* sea el correcto se realiza obteniendo el *consumer key* establecido en la tarea y comparándolo con el que se obtiene de la base de datos. Si no es el correcto se muestra un mensaje



de error, y si es el correcto se pasaría al siguiente paso. En este paso no se hace uso de la librería.

2. Para verificar que el *secret* establecido es el correcto, en primer lugar se obtiene el *secret* que tiene asociado el *consumer key* anterior desde la base de datos y este *secret* se le pasa como parámetro a una clase, perteneciente a la librería utilizada, la cual es instanciada.
3. Al instanciar dicha clase pasándola como parámetro el *secret*, se realiza un proceso interno propio de la librería de comprobaciones para ver si el *secret* pasado es el correcto. Si no lo es lanzaría una excepción, y si lo es quedaría establecida la conexión entre Moodle y la aplicación. Además, con la clase instanciada ya se podrían obtener los datos de Moodle necesarios (id del usuario, nombre, id de la tarea...).

6. *Procesamiento de los test, prácticas y generación de informes*

Otra parte clave de la aplicación es la forma en la que la aplicación recibe los test subidos por el profesor, las prácticas subidas por los alumnos y cómo se le pide a Maven que recoja todo, ejecute los test y genere los informes.

El arquetipo Maven es generado dinámicamente en el momento que el profesor sube el primer test. Si se quiere comprobar el contenido que tiene el fichero de configuración POM de Maven usado en la aplicación, se puede observar en el fichero de nombre “pom_ejemplo.xml” de la carpeta “ficheros_test” perteneciente a la carpeta “Código fuente” de la aplicación entregada en el DVD.

Teniendo en cuenta la estructura de carpetas que se genera automáticamente en el servidor (Ver *Memoria – Aspectos relevantes del desarrollo: Estructura de directorios generada automáticamente en el servidor*), los test subidos por el profesor se guardan automáticamente en la correspondiente subcarpeta del arquetipo Maven creado al generar la tarea de subida correspondiente.

Por otro lado, cuando un alumno sube una práctica, ésta se guardará también en la subcarpeta del arquetipo Maven, y a su vez se guardará en una nueva carpeta creada aparte, que representará el número del intento de subida de práctica realizado.

Una vez el arquetipo Maven contiene el test subido por el profesor y la práctica del alumno, se realizan los siguientes pasos:

1. Se comprueba que la práctica subida por el alumno está dentro de un fichero zip.





2. Se comprueba que la práctica que ha subido el alumno pertenece al paquete correcto.
3. Se compila la práctica (comando Maven: *mvn compile*) y se comprueba que la compilación es correcta.
4. Si los pasos anteriores se cumplen, se ejecuta el comando de Maven que genera los reportes de los *plugins* (comando Maven: *mvn jxr:jxr site*).
5. Se comprueba que el reporte .xml generado por el *plugin* FindBugs no contiene bucles infinitos. Si es así, no se ejecutan los test, en caso contrario sí (comando Maven: *mvn test*).

7. Pruebas

Las pruebas que se han llevado a cabo han sido de dos tipos: pruebas unitarias y de cobertura de código.

A continuación se explica el proceso que hay que realizar para poder ejecutar dichas pruebas y se mostrarán los resultados obtenidos.

7.1. Ejecutar pruebas

7.1.1. Ejecutar pruebas unitarias

El proceso a seguir es el siguiente:

1. Accedemos a la carpeta del DVD, llamada “Pruebas”.
2. Abrimos el *cmd.exe*.
3. Arrastramos a la consola el fichero llamado “ejecutarTest_unitarios.bat” y pulsamos *enter*.
4. Se ejecutarán los test y se obtendrá un mensaje como el siguiente:

```
C:\Users\Álvaro>C:\Users\Álvaro\Desktop\ejecutarTest_unitario.bat
C:\Users\Álvaro>cd C:\wamp\www\AutoCorreccionJava_TFG
C:\wamp\www\AutoCorreccionJava_TFG>php phunit-old.phar tests/TestCase/Controller
PHPUnit 4.8.21 by Sebastian Bergmann and contributors.

.....
Time: 7.87 seconds, Memory: 34.75Mb
OK (30 tests, 164 assertions)
```

Ilustración 55: Salida cmd obtenida tras ejecutar pruebas unitarias



7.1.2. Ejecutar pruebas de cobertura de código

El proceso a seguir es el siguiente:

1. Accedemos a la carpeta del DVD, llamada “Pruebas”.
2. Abrimos el `cmd.exe`.
3. Arrastramos a la consola el fichero llamado “ejecutarTest_cobertura.bat” y pulsamos *enter*.
4. Se ejecutarán los test y se obtendrá un mensaje como el siguiente:

```
C:\Windows\System32>C:\Users\Álvaro\Desktop\ejecutarTest_cobertura.bat
PHPUnit 4.8.21 by Sebastian Bergmann and contributors.
Warning:      No whitelist configured for code coverage

.....

Time: 34.21 seconds, Memory: 38.00Mb

OK (30 tests, 164 assertions)

Generating code coverage report in HTML format ... done

-----TEST COBERTURA-----
VER RESULTADOS: Copiar esta ruta en el navegador de archivos.
C:\wamp\www\AutoCorreccionJava_TFG\coverage\index.html
```

Ilustración 56: Salida cmd obtenida tras ejecutar pruebas de cobertura

5. Como se puede observar en la anterior ilustración, las pruebas de cobertura de código generan un fichero HTML en el que podrán verse los resultados. Para ello, se introduce esa ruta en el explorador de archivos de Windows y pulsamos *enter*:

`C:/wamp/www/AutoCorreccionJava_TFG/coverage/index.html`

6. Se abrirá una web con los resultados:



Ilustración 57: Resultado del test de cobertura





La carpeta en la que nos tenemos que fijar es la de “src”, ya que la de “config” son pruebas que se ejecutan indirectamente. Por ello si accedemos a la carpeta “src” obtenemos la siguiente vista:

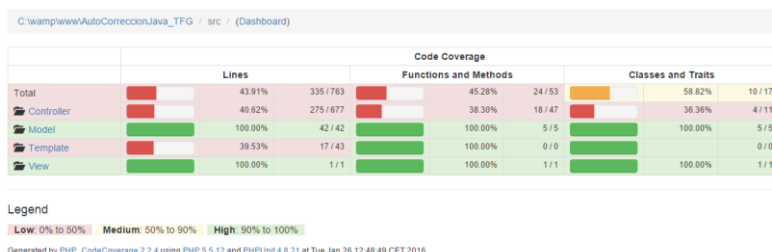


Ilustración 58: Resultado de test de cobertura carpeta "src"

7.2. Pruebas unitarias

Una prueba unitaria es una forma de comprobar el correcto funcionamiento de un módulo de código. Esto permite asegurar que cada uno de los módulos funcione correctamente por separado [17].

Visualizando la anterior ilustración (Ver Ilustración 58: Resultado de test de cobertura carpeta "src"), en nuestro caso se han realizado pruebas para el controlador (carpeta “Controller”). Probando los controladores indirectamente se realizan pruebas sobre las vistas (carpeta “Template”) y sobre el modelo (carpeta “Model”). Importante resaltar que dentro de esa carpeta Model no se tienen entidades con las que representar el modelo, sino que se tienen ficheros dentro de la carpeta “Model/Table” que son los que han permitido realizar las validaciones de los formularios de las vistas y en los que se establecen relaciones entre tablas para poder realizar los JOIN.

Idealmente la cobertura de las pruebas unitarias debería ser del 100%, aunque esto raramente se consigue. En nuestro caso el problema que se ha presentado es que había controladores que tenían métodos privados, además de otros que tenían métodos de generación de gráficos, comprobación de plagios... Y por lo tanto no ha sido posible realizar pruebas unitarias para ellos.

Los ficheros que realizan las pruebas unitarias deben de estar dentro de la carpeta *tests/TestCase*, generada por defecto por CakePHP.

La estructura de cada uno de los test va a ser prácticamente similar en todos los casos, ya que en primer lugar se tienen los métodos *setUp()*, para la inicialización de variables y obtención de las tablas de la base de datos, y *tearDown()*, para una vez ejecutado cada uno de los test se realice el borrado correspondiente de la tabla de la base de datos.



En algunos test también se tienen métodos privados `__crear..()`, que van a ser los encargados de guardar en la tabla correspondiente de la base de datos la nueva fila a crear para utilizarla como ejemplo y poder realizar las pruebas unitarias.

7.2.1. Probando el controlador

Como se ha mencionado anteriormente, no ha sido posible probar todas las funciones de los controladores ya que hay muchos que tienen métodos privados y otros métodos que no permitan realizar estas pruebas.

A continuación se va a mostrar una tabla en la que va a poder visualizarse los ficheros test creados junto a los controladores que afectan, y el número de métodos total y los testeados:

Nombre del Test	Controlador testeado	Nº métodos total	Nº métodos testeados
AlumnosControllerTest	AlumnosController	1	1
AppControllerTest	AppController	5	3
ConexionesControllerTest	ConexionesController	3	3
ErroresControllerTest	ErroresController	1	1
ExcepcionesControllerTest	ExcepcionesController	4	4
FicherosXmlControllerTest	FicherosXmlController	4	4
IntentosControllerTest	IntentosController	13	1
ProfesoresControllerTest	ProfesoresController	9	8
TareasControllerTest	TareasController	1	1
TestsControllerTest	TestsController	4	1
ViolacionesControllerTest	ViolacionesController	1	1

Tabla 25: Tabla pruebas unitarias





7.3. Pruebas de cobertura de código

La cobertura de código es una medida (porcentual) en las pruebas de software que mide el grado en el que el código fuente de un programa ha sido comprobado. Sirve para determinar la calidad del test que se llevar a cabo y para determinar las partes críticas del código que no han sido comprobadas y las partes que ya lo fueron [18].

Sobre las pruebas unitarias anteriores, se han generado los test de cobertura. La herramienta utilizada para ello sigue siendo PHPUnit, ya que permite generar un HTML con los resultados.

A continuación se mostrará la cobertura de código obtenida en los controladores:

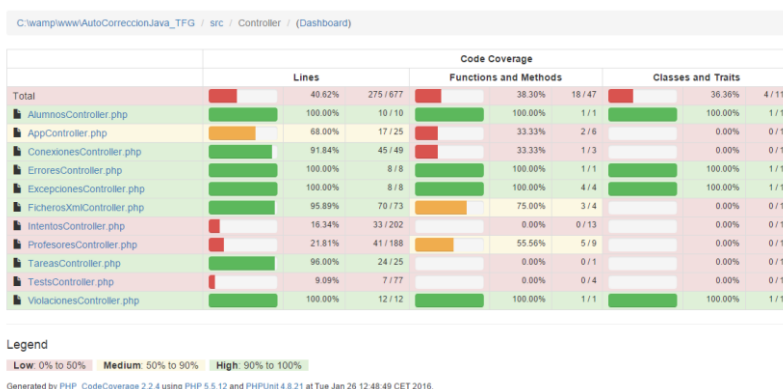


Ilustración 59: Cobertura de código de los controladores

Los controladores de los que peor resultado se ha obtenido son los siguientes:

- IntentosController: Excepto un método, el resto son métodos privados los cuales no ha sido posible probar.
- ProfesoresController: Contiene métodos que no permiten realizar pruebas sobre ellos: generarGráficas, comprobarReportePlagiosPrácticas...
- TestsController: Contiene varios métodos privados, además de un método público que recibe el test subido en el formulario, que ha impedido realizar las correspondientes pruebas.



V - MANUAL DEL USUARIO

Este manual ayudará a usar la aplicación a los posibles futuros usuarios. Dentro de ella se explicará por un lado el manual necesario cuando el usuario se trata de un profesor y por otro cuando se trata de un alumno.

1. *Manual del profesor*

1.1. Registrar profesor en la aplicación

1.1.1. Proceso de registro

Este es el primer paso que deberá de realizar el profesor para poder utilizar la aplicación. Es imprescindible que este paso sea realizado ya que sino ni profesor ni alumnos podrán acceder a la aplicación.

Para proceder al registro del profesor los pasos que ha de seguir son los siguientes:

1. Accedemos a la página del registro en la aplicación, que en caso de encontrarse en la propia máquina podrá encontrarse en la siguiente URL:

http://localhost/AutoCorreccionJava_TFG/Profesores/registrar

2. Aparecerá un formulario como el siguiente y rellenaremos los campos del formulario tal y como se nos indican:

Registro del profesor en la Aplicación ⓘ

Nombre *

Apellidos *

Correo electrónico *

Tiene que ser el de Moodle

Contraseña *

Confirmar contraseña *

Asegúrese que sea la misma

Registrar Resetear campos

Ilustración 60: Formulario de registro del profesor





* **IMPORTANTE:** El correo electrónico debe de ser el que se usa en Moodle.

- Una vez rellenado el formulario, seleccionamos “Registrar” y automáticamente seremos redirigidos a otra página en la que se nos mostrarán los parámetros LTI (*URL*, *consumer_key* y *secret*) necesarios para que el profesor posteriormente pueda crear una tarea de tipo “herramienta externa” desde Moodle (proceso explicado en el siguiente apartado). El aspecto de esta página es el siguiente:



Ilustración 61: Parámetros LTI obtenidos

1.1.2. Configurar Moodle para acceder a la aplicación

Una vez el profesor ha sido registrado y ha obtenido los parámetros LTI, a continuación se va a explicar el proceso que debe de seguir para crear desde Moodle una tarea que enlace con la aplicación y que por lo tanto permita el acceso a ella.

Estas tareas se crean dentro de los cursos, por lo tanto en primer lugar hay que tener creado un curso. El proceso de creación de un curso en Moodle no va a explicarse ya que no formaría parte de la aplicación, a pesar de ello se proporciona el siguiente enlace en el que se explica cómo crear un curso en Moodle [19].

Partiendo de que el curso ya está creado y que el profesor ha accedido a Moodle, la apariencia que veríamos al entrar al curso sería la siguiente:



Ilustración 62: Moodle aspecto curso



Y ahora se explicará de forma detallada cómo crear la tarea dentro del curso:

1. Procedemos a añadir la actividad de tipo “herramienta externa” la cual va a ser la que va a enlazar con nuestra aplicación. Para ello seleccionamos el botón “Añadir una actividad o un recurso” que se encuentra en el tema “Tema 1” y marcamos la opción de “Herramienta externa” y en “Agregar”.

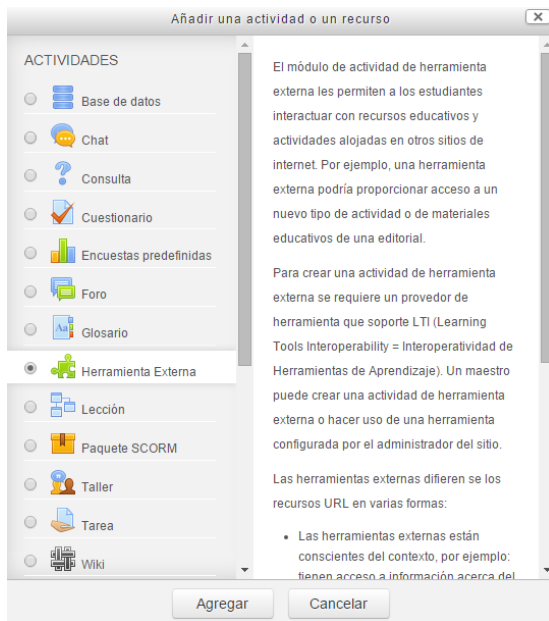


Ilustración 63: Moodle selección herramienta externa

2. A continuación nos aparecerá un formulario con una serie de campos que deberemos de rellenar de la forma que va a explicarse a continuación:

El apartado “General” contendrá los siguientes campos:

- **Nombre de la actividad:** establecemos el nombre que queramos para nuestra actividad.
- **Tipo de herramienta externa:** es la opción más importante a rellenar. Para ello pincharemos en el símbolo “+” que aparece, y así podremos proceder a su configuración. Se nos abrirá una nueva ventana con un nuevo formulario a rellenar:

Apartado “Ajustes de la herramienta”: contiene los siguientes campos:





- Tool name: asignamos el nombre que queramos.
- Tool base URL: será la dirección que permite acceder a nuestra aplicación web, es la conexión. Se deberá de establecer la URL proporcionada al profesor en el momento del registro. **IMPORTANTE** que sea la misma ya que si no será imposible el acceso a la aplicación.
- Clave de cliente (consumer key): Se establecerá la que se le fue proporcionada al profesor en el momento del registro. **IMPORTANTE** que sea la misma ya que si no se denegará el acceso a la aplicación.
- Secret: Se establecerá el que se le fue proporcionada al profesor en el momento del registro. **IMPORTANTE** que sea el mismo ya que si no se denegará el acceso a la aplicación.
- Parámetros personalizados: no rellenamos nada.
- Contenedor de inicio por defecto: opción que viene por defecto.

El aspecto se tendrá que parecer al siguiente:

Tool name* ?

Tool base URL* ?

Clave de cliente ?

Shared secret ?

☐ Desenmascarar

Parámetros personalizados ?

Contenedor de inicio por defecto ?

Ilustración 64: Moodle configuración parámetros de la tarea paso 1



Apartado “Privacy”: Aparecerán 3 desplegables los cuales serán rellenados seleccionando la opción “Siempre”, para que así nuestra aplicación web pueda tener acceso a dicha información.

Compartir el nombre del usuario con la herramienta Siempre ▼
?

Compartir el e-mail del usuario con la herramienta Siempre ▼
?

Aceptar calificaciones desde la herramienta Siempre ▼
?

? ☐ Forzar SSL

Ilustración 65: Moodle configuración parámetros de la tarea paso 2

- Tras estos cambios, acabamos clicando el botón de abajo “Guardar cambios”.
 - **URL de inicio:** lo dejamos vacío.
 - **Iniciar el contenedor:** dejamos la opción que viene por defecto.
3. Finalmente pulsamos el botón “Guardar cambios y regresar al curso” que aparece en la parte inferior y ya tendremos agregada la actividad en nuestro curso, la cual cuando seleccionemos nos enlazará con nuestra aplicación web. El aspecto se asemejará al siguiente:

Tema 1



Autocorrección java

Ilustración 66: Moodle tarea creada





1.2. Panel principal

El aspecto que va a tener la aplicación cuando el profesor accede a ella es el siguiente:



Ilustración 67: Panel principal del profesor de la aplicación

En los siguientes apartados se van a explicar cada una de las tareas que puede realizar el profesor desde su panel:

1.3. Panel Configuración

En primer lugar el profesor deberá seleccionar la opción “Configuración” presente en el panel principal.

Tras esto, se le abrirá una nueva página donde el profesor podrá configurar los parámetros correspondientes a la tarea. Estos parámetros afectarán tanto a las prácticas que suban los alumnos como a los propios test del profesor, como es el caso del “paquete”, y a diferencia de los otros campos que únicamente afectarán a las prácticas de los alumnos.

El aspecto de este panel de configuración del profesor es el siguiente:



Configuración de los parámetros de la tarea ⓘ

Número máximo de intentos *

Paquete *

Fecha límite *

Ilustración 68: Panel del profesor de configuración de parámetros de la tarea

Para ayudar a realizar una configuración correcta, se puede consultar primero el botón “info” presente en el título en el cuál se explicarán puntos a tener en cuenta.

1.4. Panel Subida

En primer lugar el profesor deberá seleccionar la opción “Subida” presente en el panel principal.

Desde este panel el profesor podrá subir sus test y enunciados de las prácticas para que posteriormente los alumnos, cuando accedan a la aplicación puedan subir sus prácticas para que sean corregidas y además poder observar el enunciado que deben de seguir.

El aspecto de este panel es el siguiente:

Subida de Test ⓘ

Enunciado de la práctica

Fichero a subir:

Ningún archivo seleccionado

Ilustración 69: Panel del profesor de subida de test





Al igual que ocurría en el panel de configuración de la tarea, se importante consultar la opción “info” presente al lado del título.

1.5. Panel Estadísticas

En primer lugar el profesor deberá seleccionar la opción “Estadísticas” presente en el panel principal.

Desde este panel el profesor va a poder consultar las estadísticas y reportes de cada una de las prácticas subidas por los alumnos. La tabla permite realizar filtrados, así como ordenaciones de los datos de la misma. Además también se permite la descarga de las prácticas de los alumnos en formato .zip.

1.5.1. Aspecto inicial

El aspecto de este panel es el siguiente:

Estadísticas de las prácticas de los alumnos						
Mostrar						Buscar:
10						
intentos	Nombre	Intento	Test pasado	Comentarios	Fecha subida	Práctica
⊖	Álvaro Vázquez	1	no	✖	1/18/16, 11:10 PM	practica1.zip
⊖	Álvaro Vázquez	2	no	💬	1/18/16, 11:18 PM	practica1.zip
⊖	Álvaro Vázquez	3	no	💬	1/18/16, 11:22 PM	practica1.zip
⊖	Álvaro Vázquez	4	no	✖	1/18/16, 11:25 PM	practica1.zip
⊖	Álvaro Vázquez	5	no	✖	1/18/16, 11:27 PM	practica1.zip
⊖	Álvaro Vázquez	6	no	✖	1/18/16, 11:29 PM	practica1.zip
⊖	Álvaro Vázquez	7	no	✖	1/21/16, 4:14 PM	practica1.zip
⊖	Luis Izquierdo	1	no	✖	1/18/16, 11:35 PM	practica1.zip
⊖	Luis Izquierdo	2	no	✖	1/18/16, 11:37 PM	practica1.zip
⊖	Luis Izquierdo	3	sí	✖	1/18/16, 11:38 PM	practica1.zip
Mostrando 1 a 10 de 18 intentos						
		Primera	Anterior	1	2	Siguiente Última
Reporte JavanCSS Reporte JDepend Reporte PMD Reporte FindBugs Reporte Errores						

Ilustración 70: Panel del profesor de estadísticas

1.5.2. Consultar Reportes

Como se puede observar, los botones referentes a los reportes habilitados en la parte inferior se encuentran deshabilitados de partida.



Estos botones se activarán o desactivarán en el momento que el profesor seleccione una de las prácticas de la tabla, y según el resultado obtenido de la práctica, se activarán o desactivarán dichos botones. A continuación se mostrará el aspecto de los botones de los reportes tras seleccionar una práctica:



Ilustración 71: Panel del profesor de estadísticas consulta de reportes

Como se trata de una práctica que sí que ha pasado todos los test, el botón correspondiente al reporte de errores aparece desactivado.

1.6. Panel Gráficas

En primer lugar el profesor deberá seleccionar la opción “Gráficas” presente en el panel principal.

Desde este panel el profesor podrá visualizar diferentes gráficas en la que se podrán visualizar algunos datos globales acerca de las prácticas subidas por los alumnos.

Además de poder generar estas gráficas globales, el profesor también podrá seleccionar las gráficas de un alumno en concreto.

1.6.1. Aspecto inicial

El aspecto de este panel sería el siguiente:

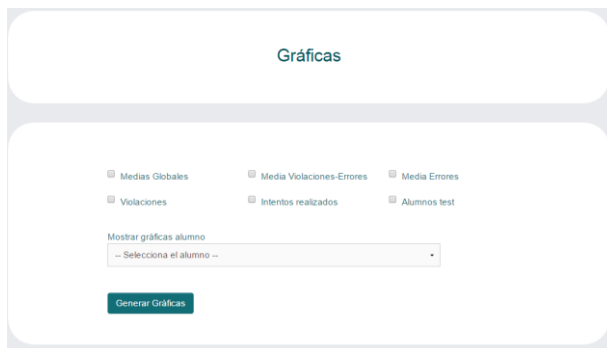


Ilustración 72: Panel del profesor de gráficas





1.6.2. Selección de gráficas

Y si por ejemplo se seleccionan varias gráficas a mostrar, el aspecto sería el siguiente:

☐ Medias Globales☐ Media Violaciones-Errores☐ Media Errores

☒ Violaciones☒ Intentos realizados☒ Alumnos test

Mostrar gráficas alumno

-- Selecciona el alumno --

Generar Gráficas

Gráficas seleccionadas

ViolacionesIntentos realizadosAlumnos test

Porcentaje de alumnos que pasan y no pasan los test

Categoría	Porcentaje
Alumnos que no pasan los test	89%
Alumnos que pasan los test	11%

Ilustración 73: Panel del profesor de selección de gráficas

1.7. Panel Plagios

En primer lugar, el profesor deberá seleccionar la opción “Plagios” desde el panel principal.

Desde este panel el profesor podrá realizar una comprobación de plagios entre la última práctica subida por todos los alumnos, o entre los alumnos que desee.

1.7.1. Aspecto inicial

El aspecto de este panel es el siguiente:



Comprobación de Plagios de prácticas

☐ TODOS CON TODOS

☐ Álvaro Vázquez

☐ Luis Izquierdo

☐ Roberto Flores López

☐ Gustavo García Sánchez

☐ Julián Fernández Gómez

Comprobar Plagios

Ilustración 74: Panel del profesor de comprobación de plagios

1.7.2. Comprobar plagios

Si por ejemplo el profesor quiere realizar una comprobación de plagios entre dos alumnos, el aspecto sería el siguiente:

☐ TODOS CON TODOS

☒ Álvaro Vázquez

☒ Luis Izquierdo

☐ Roberto Flores López

☐ Gustavo García Sánchez

☐ Julián Fernández Gómez

Comprobar Plagios

Reporte de plagios

La comprobación de plagios se ha realizado entre 2 prácticas.

Ver Reporte

Ilustración 75: Panel del profesor de selección de plagios





Desde el botón habilitado “Ver Reporte” podrá ver los posibles plagios entre las prácticas de los alumnos seleccionados.

2. Manual del alumno

En este apartado se muestra el manual del alumno para poder utilizar la aplicación. Se presupone que el alumno ha accedido a la aplicación correctamente desde la tarea correspondiente de Moodle.

2.1. Aspecto Inicial

La página que se le mostrará al alumno al acceder a la aplicación variará dependiendo de si este todavía no ha subido ninguna práctica o si ya lo ha hecho.

2.1.1. Ninguna práctica subida

Si el alumno ha subido ninguna práctica, el aspecto de su panel es el siguiente:

Cambiar Idioma TFG - Autocorrección prácticas en Java

SUBIDA DE PRÁCTICAS

Parámetros de la tarea [Ver](#)

Comentarios

Fichero a subir:

[Seleccionar archivo](#) Ningún archivo seleccionado

[Subir](#) [Resetear campos](#)

Ilustración 76: Panel principal del alumno



2.1.2. Práctica subida

Si el alumno sí que ha subido prácticas, entonces en la parte inferior podrá visualizar los reportes y las gráficas correspondientes a la última práctica subida:

Panel de SUBIDA DE PRÁCTICAS. Incluye un botón 'Ver' para los parámetros de la tarea, un campo de comentarios, un botón 'Ver' para los parámetros de la tarea, un campo de fichero a subir, un botón 'Seleccionar archivo', un botón 'Subir', un botón 'Reseteo campos', una sección de reportes del último intento realizado con botones JAVANCSS, JDEPEND, PMD, FINDBUGS, y una sección de gráficas disponibles con botones Violaciones, Errores, Violaciones - Errores, Violaciones - Prioridades.

Ilustración 77: Panel del alumno visualización de gráficas y reportes

2.2. Información y Enunciado

El alumno podrá consultar la información referente a qué requisitos debe de cumplir la práctica que va a entregar, y también podrá consultar el enunciado de la misma.

Para consultar ambas, se deberá de colocar el cursor encima de las dos imágenes presentes en el título del panel:

Panel de SUBIDA DE PRÁCTICAS. Incluye un botón 'Ver' para los parámetros de la tarea, un campo de fichero a subir, un botón 'Seleccionar archivo', un botón 'Subir', un botón 'Reseteo campos', una sección de reportes del último intento realizado con botones JAVANCSS, JDEPEND, PMD, FINDBUGS, y una sección de gráficas disponibles con botones Violaciones, Errores, Violaciones - Errores, Violaciones - Prioridades.

Ilustración 78: Panel del alumno información y enunciado





2.3. Parámetros de la tarea

El alumno podrá en todo momento consultar los parámetros de la tarea, además de poder ver el número de intentos que ha realizado.

Para ello deberá de pinchar en el botón “Ver” situado al lado de la etiqueta “Parámetros de la tarea” presente en el formulario. Tras pincharle, se abrirá una pequeña ventana que contendrá el siguiente aspecto:

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

Fichero a subir:

Seleccionar archivo Ningún archivo seleccionado

Ilustración 79: Panel del alumno parámetros de la tarea



2.4. Subir práctica

El alumno podrá seleccionar la práctica que desea subir para su corrección desde la opción “Fichero a subir” presente en el formulario.

Después de seleccionar la opción “Subir” del formulario, al alumno se le podrán mostrar diferentes mensajes (que aparecerán en la parte superior derecha del panel) en función de las características de la práctica subida:

- La práctica subida ha superado la fecha límite de entrega.
- Se ha realizado ya el número máximo de intentos posibles de subida de prácticas.
- La práctica subida no está dentro de un fichero .zip.
- La práctica subida no contiene la estructura de carpetas correcta, acorde al paquete establecido por el profesor.
- La práctica subida tiene errores de compilación.
- La práctica subida tiene bucles infinitos, y por lo tanto no se ejecutan los test.
- La práctica subida es correcta, y por lo tanto se realizará el proceso de corrección de la misma.

Un ejemplo de un mensaje que puede aparecer sería el siguiente, el cual se corresponde con una subida de práctica que no está dentro de un zip:

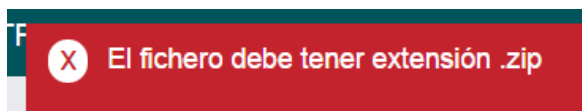


Ilustración 80: Panel del alumno error subida de práctica





2.5. Ver reportes y gráficas

El alumno podrá ver los reportes y las gráficas correspondientes a la última práctica subida.

Para ello simplemente deberá de pinchar en los botones de reportes, o en caso contrario si quiere visualizar gráficas en la opción correspondiente:



Ilustración 81: Panel del alumno ver reportes y gráficas

3. Cambiar Idioma

Por último, y aplicado tanto al panel del profesor como al de los alumnos, ambos tipos de usuarios podrán cambiar el idioma de la aplicación de español a inglés y al revés.

Para ello simplemente tendrán que pinchar en la opción “Cambiar Idioma” presente en la barra de navegación, y automáticamente el idioma de la misma será cambiado:

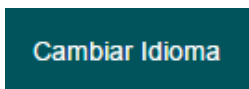


Ilustración 82: Cambiar idioma



VI - BIBLIOGRAFÍA

- [1] «Apache License», *Wikipedia, la enciclopedia libre*. 14-ene-2016.
- [2] «Bases y Tipos de Cotización 2015», *Crear-Empresas*. [En línea]. Disponible en: <http://www.crear-empresas.com/bases-tipos-cotizacion>. [Accedido: 05-ene-2016].
- [3] E. E. País, «Moodle llena la geografía educativa española de campus virtuales», *EL PAÍS*, 04-dic-2008. [En línea]. Disponible en: http://elpais.com/diario/2008/12/04/ciberpais/1228361065_850215.html. [Accedido: 03-feb-2016].
- [4] «IMS Members and Affiliates | IMS Global Learning Consortium». [En línea]. Disponible en: <http://www.imsglobal.org/membersandaffiliates.html>. [Accedido: 03-feb-2016].
- [5] «Vistas - CakePHP cookbook». [En línea]. Disponible en: <http://book.cakephp.org/3.0/en/views.html>.
- [6] «Controladores - CakePHP cookbook». [En línea]. Disponible en: <http://book.cakephp.org/3.0/en/controllers.html>.
- [7] «Modelo - CakePHP cookbook». [En línea]. Disponible en: <http://book.cakephp.org/3.0/en/orm/saving-data.html>.
- [8] Andy Smith, *Librería IMS-BLTI*. .
- [9] «Libchart». [En línea]. Disponible en: <http://naku.dohcrew.com/libchart/pages/introduction/>.
- [10] «JPlag». [En línea]. Disponible en: <https://jplag.ipd.kit.edu/>.
- [11] «Plugin JavaNCSS». [En línea]. Disponible en: <http://www.mojohaus.org/javancss-maven-plugin/>.
- [12] «Plugin JDepend». [En línea]. Disponible en: <http://www.mojohaus.org/jdepend-maven-plugin/>.
- [13] «Plugin PMD». [En línea]. Disponible en: <https://maven.apache.org/plugins/maven-pmd-plugin/>.
- [14] «Plugin FindBugs». [En línea]. Disponible en: <http://gleclaire.github.io/findbugs-maven-plugin/>.
- [15] «PHPUnit - descarga fichero .phar». [En línea]. Disponible en: <https://phpunit.de/#download>.
- [16] «Eclipse for PHP Developers». [En línea]. Disponible en: <http://www.eclipse.org/downloads/packages/eclipse-php-developers/heliosr>.
- [17] «Prueba unitaria», *Wikipedia, la enciclopedia libre*. 30-nov-2015.
- [18] «Cobertura de código», *Wikipedia, la enciclopedia libre*. 01-sep-2015.
- [19] «Moodle - Creación de un curso». [En línea]. Disponible en: https://docs.moodle.org/all/es/A%C3%B1adiendo_un_nuevo_curso.





Impreso en Burgos el jueves, 4 de febrero de 2016