# Plan de proyecto software

## 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 taras necesarias y sus plazas, 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.

## 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.

A continuación se van a mostrar los *Sprint Backlogs* del proyecto junto con sus correspondientes *BurndownCharts.*

**AQUÍ HAY QUE PONER EL GRÁFICO GLOBAL DEL PROYECTO**

### 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.



Ilustración : Sprint Backlog Iteración 1



Ilustración : Burndown Chart Iteración 1

### 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 nuestro servicio web creando para ello una actividad de tipo “herramienta externa” desde Moodle. El resultado es que al pinchar en dicha actividad, se mostrará en el servicio web la información del usuario que la ha pincado.



Ilustración : Sprint Backlog Iteración 2



Ilustración : Burndown Chart Iteración 2

### 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 (consumer key y secret) necesarios a la hora de realizar la conexión entre Moodle y el servicio 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.



Ilustración : Sprint Backlog Iteración 3



Ilustración : Burndown Chart Iteración 3

### 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 todo el código que se tenía realizado para poder aplicar dicho framework.



Ilustración : Sprint Backlog Iteración 4



Ilustración : Burndown Chart Iteración 4

### 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.



Ilustración : Sprint Backlog Iteración 5



Ilustración : Burndown Chart Iteración 5

### 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”.

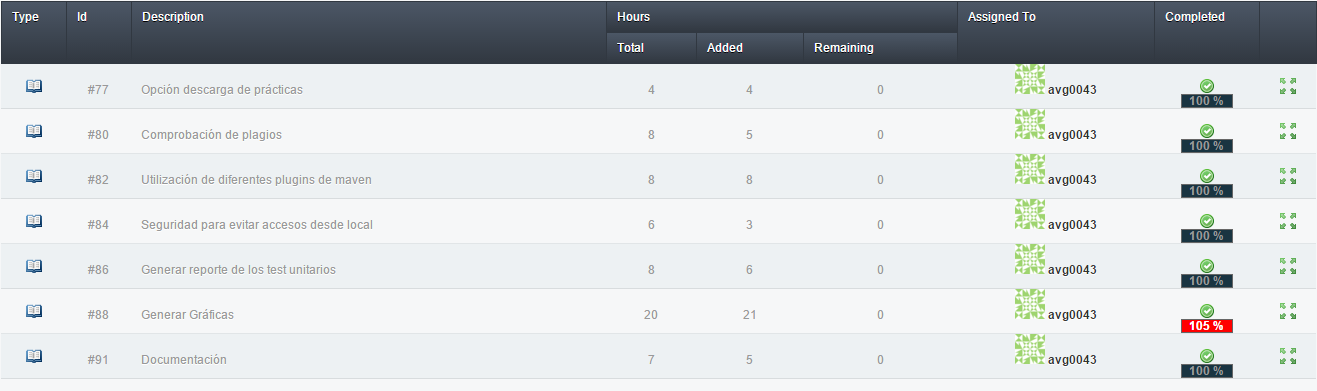


Ilustración : Sprint Backlog Iteración 6



Ilustración : Burndown Chart Iteración 6

### 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.



Ilustración : Sprint Backlog Iteración 7



Ilustración : Burndown Chart Iteración 7

### 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 pueden saltar 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”.

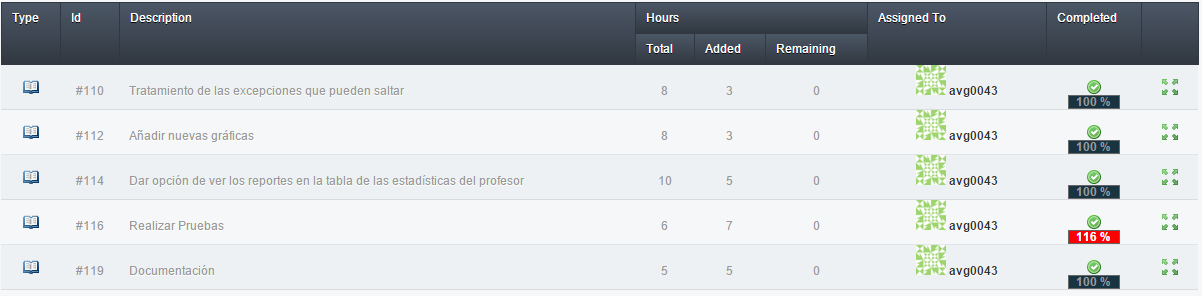


Ilustración : Sprint Backlog Iteración 8

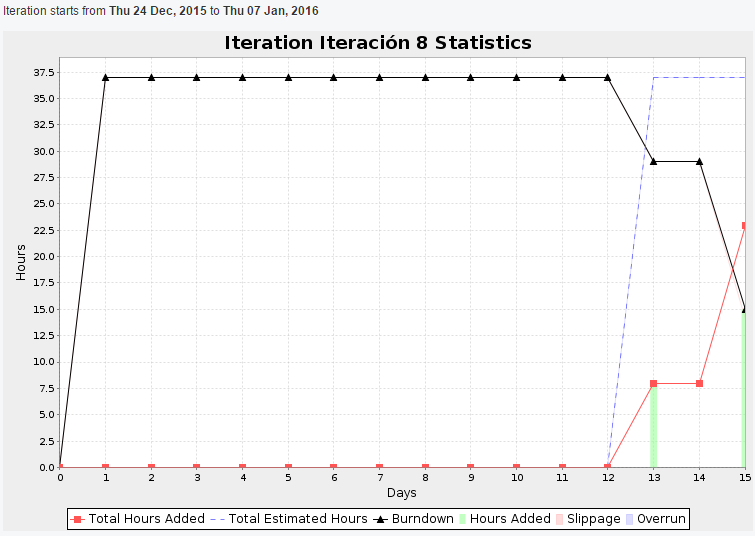


Ilustración : Burndown Chart Iteración 8

## 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.

### 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* que es la licencia utilizada en la mayoría de las aplicaciones.

A continuación se citan las principales características de dicha licencia.

#### 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].

#### 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.

### 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.

#### 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 analizar son los de personal, hardware, software, otros costes y los totales:

* **Costes de personal:**

El proyecto será realizado por un graduado en Ingeniería Informática 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:

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

Por último hay que tener en cuenta los gatos 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:

* **Coste de hardware:**

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

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:

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

* **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** |
| *Microsoft Windows 8.1 (64 bits)* | Individual | 1 | 100,00 € |
| *Microsoft Office 2013* | Individual | 1 | 99,00 € |
| *Moodle* | GNU GPL | 1 | 0,00 € |
| *XP-DEV* | GNU GPL | 1 | 0,00 € |
| *WampServer* | GNU GPL | 1 | 0,00 € |
| *Eclipse PHP Developers* | GNU GPL | 1 | 0,00 € |
| *Maven* | APACHE License | 1 | 0,00 € |
| *GitHub* | GNU GPL | 1 | 0,00 € |
| *HeidiSQL* | GNU GPL | 1 | 0,00 € |
| *Astah Community* | Gratuita | 1 | 0,00 € |
| *FileSync* | GNU GPL | 1 | 0,00 € |
| *Bootstrap* | MIT License | 1 | 0,00 € |
| TOTAL | | | 199, 00 € |

Tabla : 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:

* **Otros costes:**

Además hay que tener en cuenta los siguientes costes:

* **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:

# Especificación de requisitos

## 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.

## Requisitos funcionales

### Propios de Moodle

#### 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, key y secret correspondiente. Cuando esta tarea sea seleccionada, se redirigirá a dicha aplicación web. 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…

#### 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.

### Propios de la aplicación web

#### 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…).

#### 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 les podrá ver, 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 que va teniendo tras realizar cada intento de subida de práctica.

### Requisitos de restricción

* **RR-01:** Únicamente se puede acceder a la aplicación web desde las tareas creadas en Moodle, en ningún momento se podrá tener acceso a la aplicación de otra forma.
* **RR-02:** La aplicación sólo permite trabajar con test y prácticas escritas en Java, en este momento no soporta otros lenguajes de programación.
* **RR-03:** 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.

## 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.

## 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 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 se muestran los casos de uso pertenecientes a Moodle y con color amarillo los de la Aplicación web:



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”**

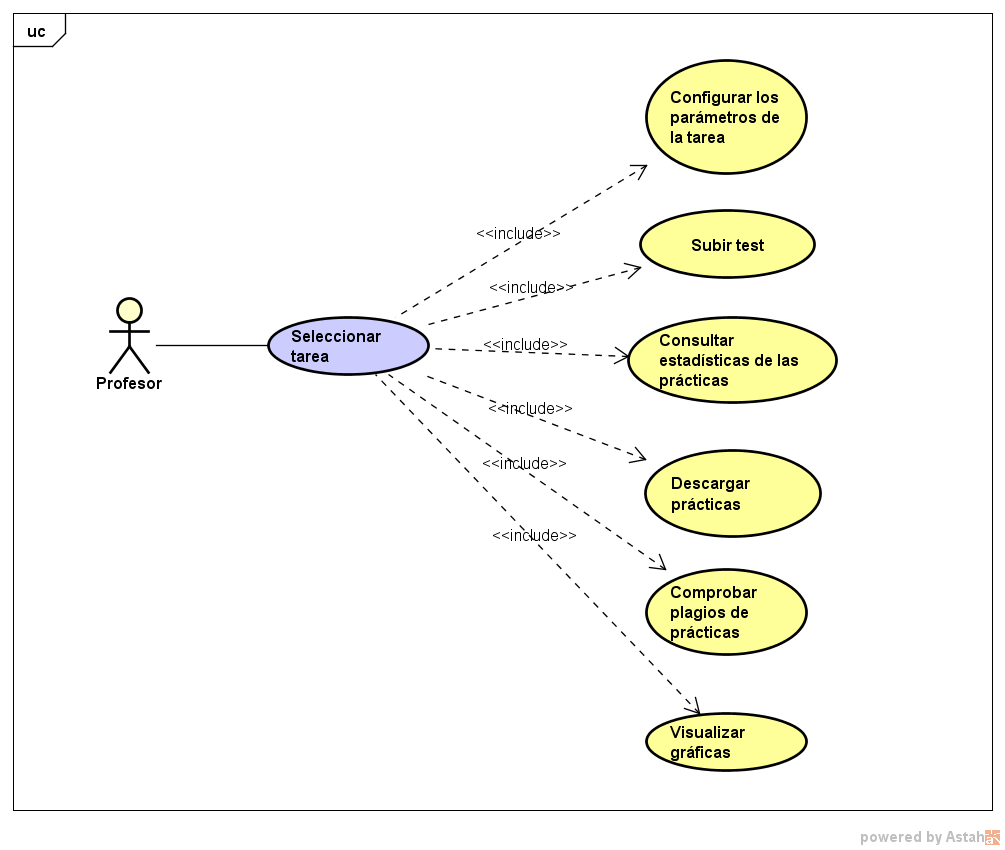
Este diagrama va a mostrar el diagrama de caso de uso más general, es decir, el de todo el sistema. Se va a ver reflejado las diferentes relaciones que hay entre los requisitos enumerados anteriormente.

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

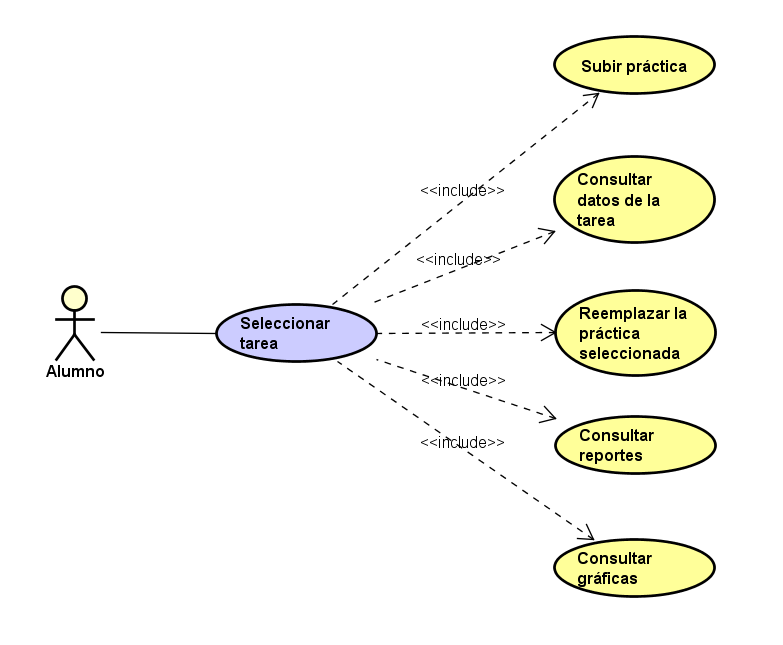


A continuación se van a mostrar los diagramas de casos de uso más específicos:

* **Diagrama casos de uso “seleccionar actividad”**
  1. Rol profesor:



* 1. Rol alumno:



### 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> |
| 2 | Si <condición>, {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 : Plantilla tipo casos de uso

#### Pertenecientes a Moodle

|  |  |  |
| --- | --- | --- |
| **RF-M01** | **Crear la tarea** | |
| **Descripción** | Crear una tarea de tipo “herramienta externa” que al ser clicada, 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, key y secret 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, key, secret) incorrectamente, no podrá acceder a la aplicación web. |
| **Importancia** | Alta. | |
| **Frecuencia** | Baja. | |
| **Urgencia** | Alta. | |

Tabla : 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 pulsa en “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 : Plantilla RF-M02 "Configurar la tarea"

|  |  |  |
| --- | --- | --- |
| **RF-M03** | **Seleccionar tarea** | |
| **Descripción** | El usuario, bien sea el profesor o el alumno, clica 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 clica 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 : Plantilla RF-M03 "Seleccionar la tarea"

#### 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 : 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 | Pincha en el botón 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 : 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. | |
| **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 | Pincha en el botón 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 a rellenado el enunciado. |
| **Importancia** | Alta. | |
| **Frecuencia** | Media. | |
| **Urgencia** | Alta. | |

Tabla : 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 : 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 : 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 : 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 | Pincha en el botón 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 : 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 | Pincha en el botón 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 : 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 : 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 : 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 : 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 : Plantilla RF-AW12 "Consultar gráficas"

# 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.

## Diseño de datos

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

### Base de datos

#### Modelo relacional

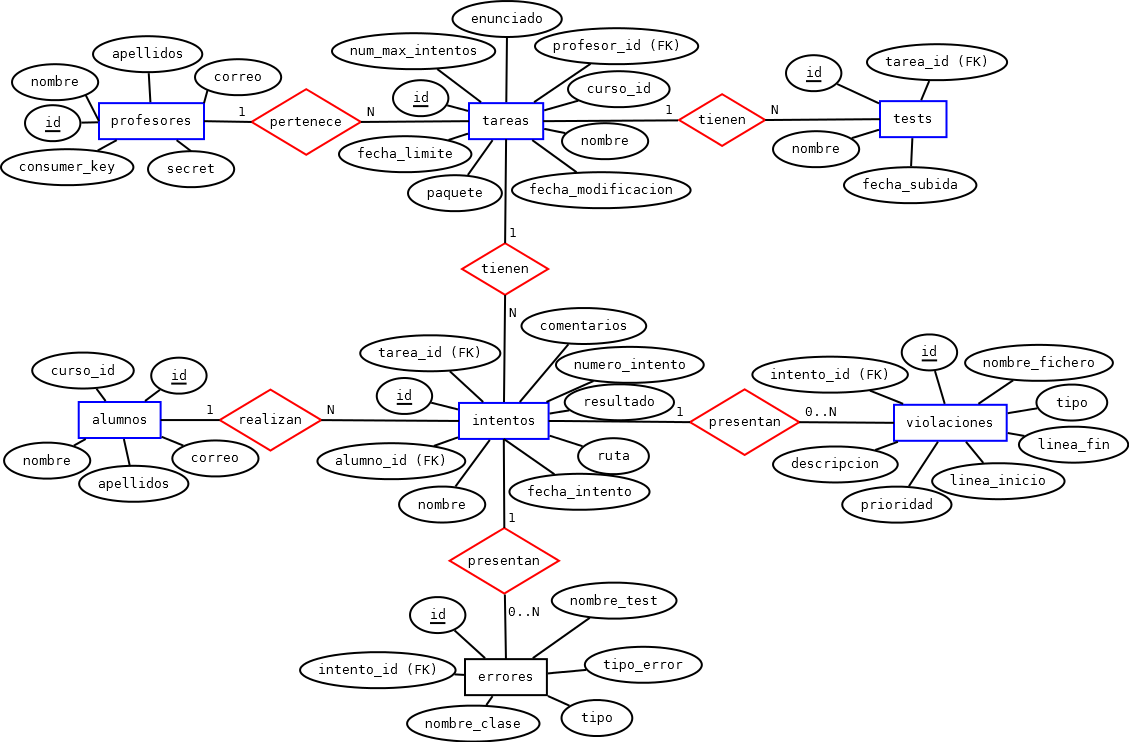


Ilustración : Modelo relacional

#### Diccionario de datos

En este apartado va a mostrarse la información referente a las tablas de la base de datos utilizadas en la aplicación.

* **profesores**: Contiene la 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 : Tabla profesores

*\* Primary key (PK):* id.

* **alumnos**: Contiene la información referente a los alumnos que han accedido al servicio 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 : 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 : 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 : 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 : 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. |
| cescripción | 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 : 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 : Tabla errores

*\* Primary key (PK):* id.

*\* Foreign key (FK):*

* intento\_id -> FK hacia el campo “id” de la tabla “intentos”.

### Diagrama de paquetes

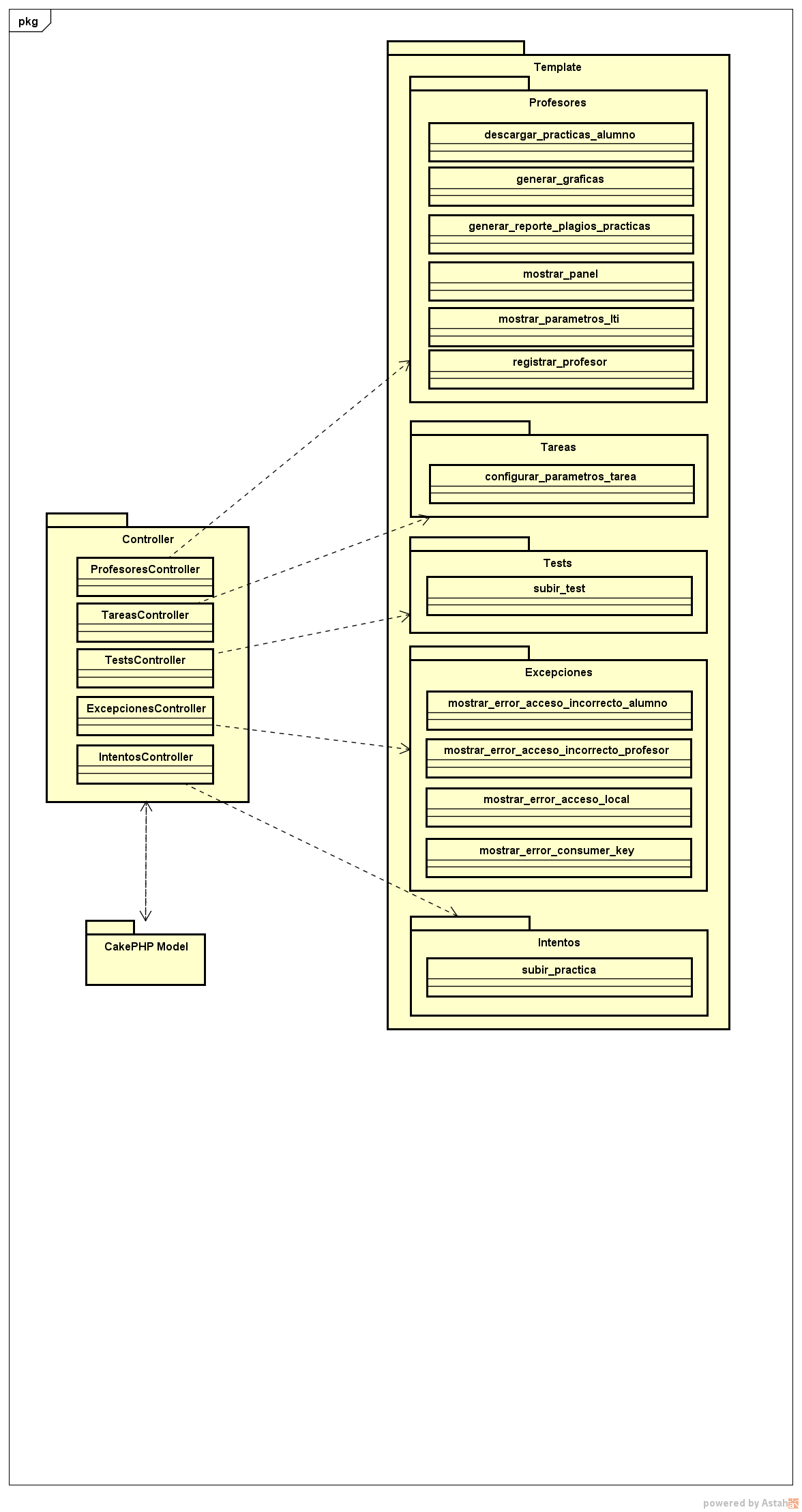


Ilustración : Diagrama de paquetes

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

* **Controller**: Dentro de la cual van a estar todos los controladores.
* **Template**: Contiene las vistas. 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 [3].

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 [4].

Respecto al modelo, no ha sido necesario crear entidades debido a que CakePHP nos lo facilita y no obliga a tener creada 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 [5].

### Diagrama de clases

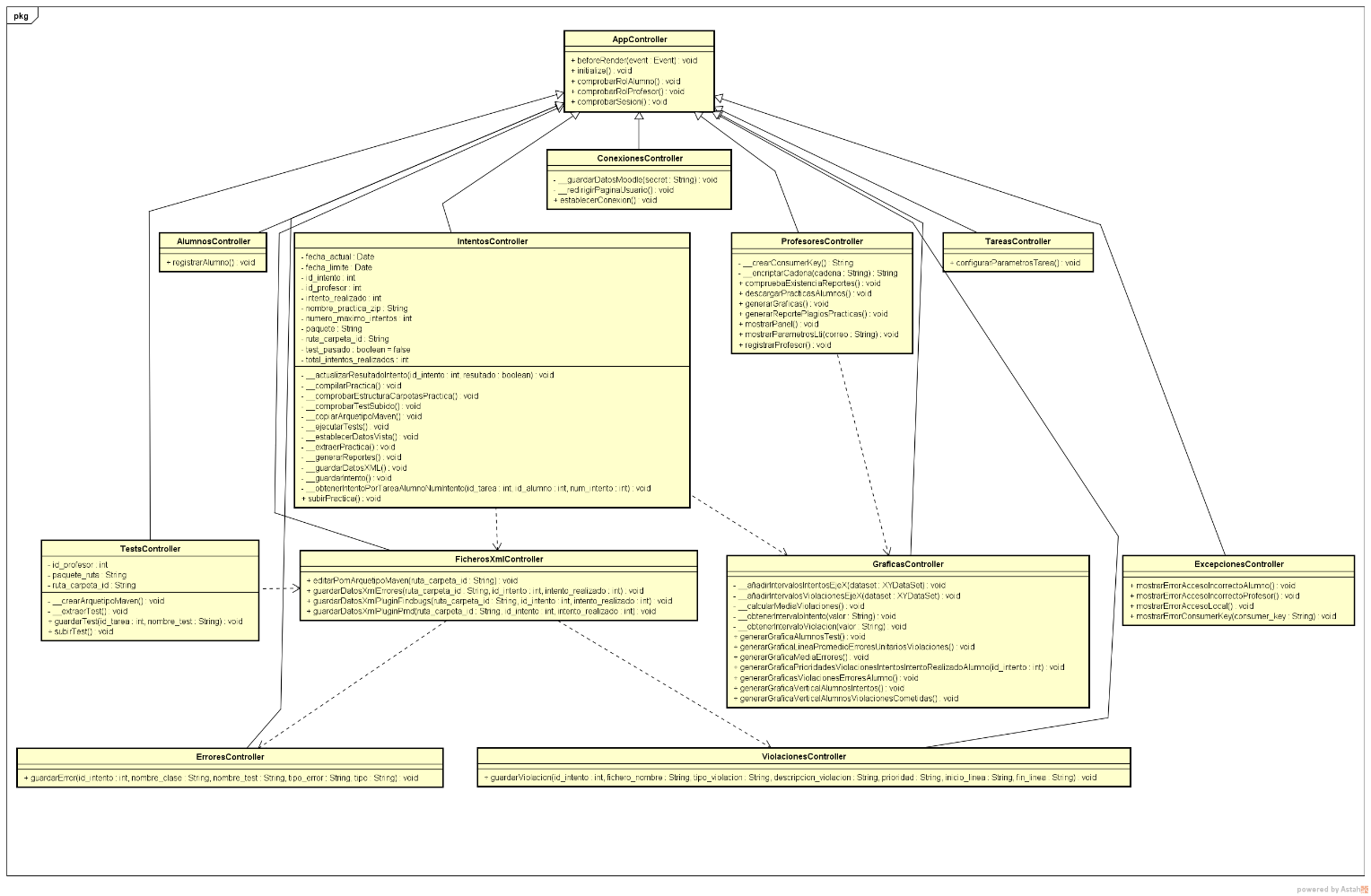


Ilustración : Diagrama de clases

**Explicación del diagrama de clases**

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

En primer lugar, todos los controladores heredan de “AppController”. La aplicación siempre choca contra “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:
  1. Si es su primer acceso, será redirigido a “TareasController” para que rellene el formulario obligatorio de configuración de parámetros de la tarea.
  2. 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:
  1. Si es su primer acceso, será redirigido a “AlumnosController” donde se procederá a su registro en la aplicación.
  2. 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.

## Diseño procedimental

### Diagramas de secuencias

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

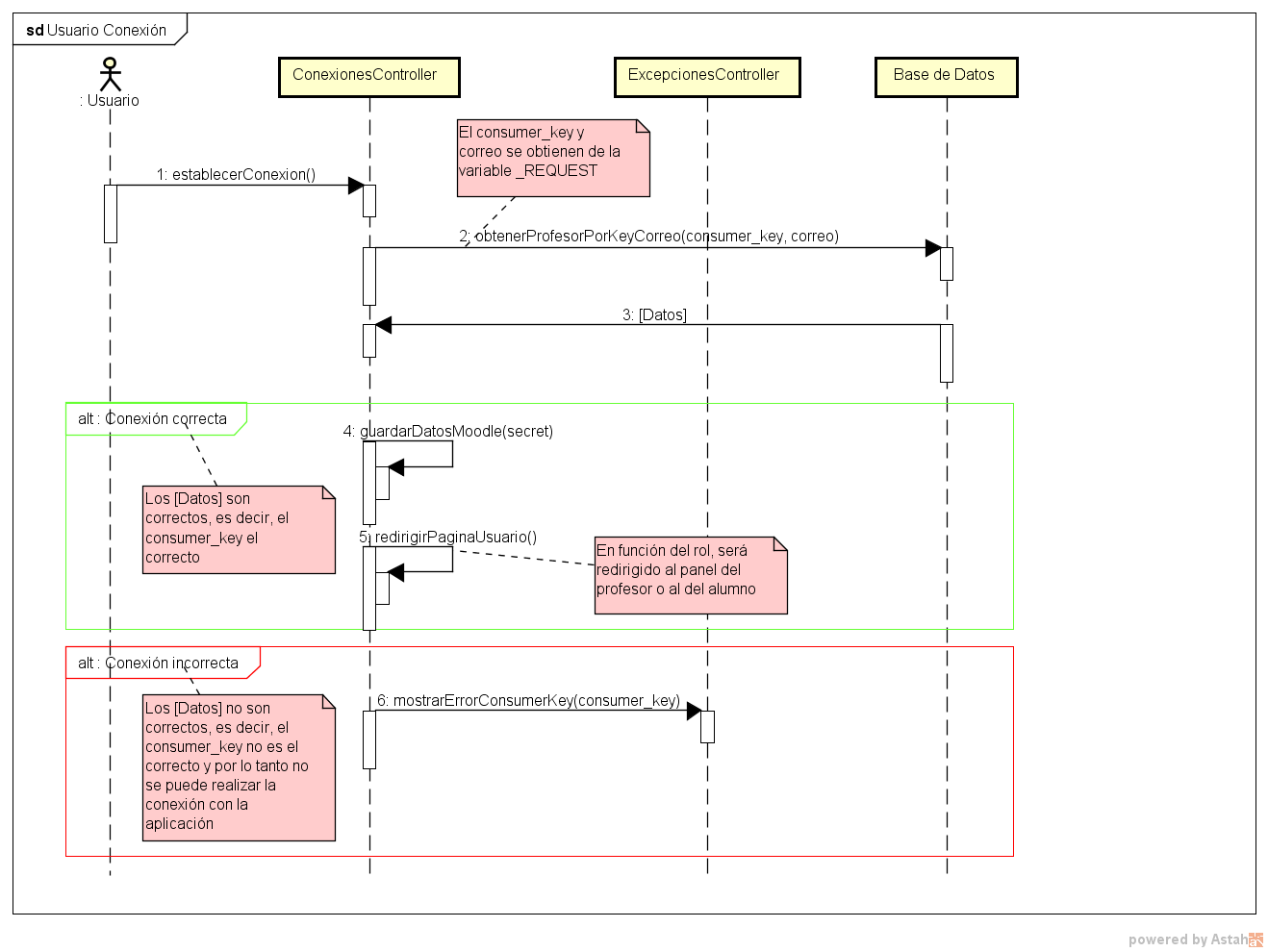


Ilustración : Diagrama de secuencia "Conexión del usuario"

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

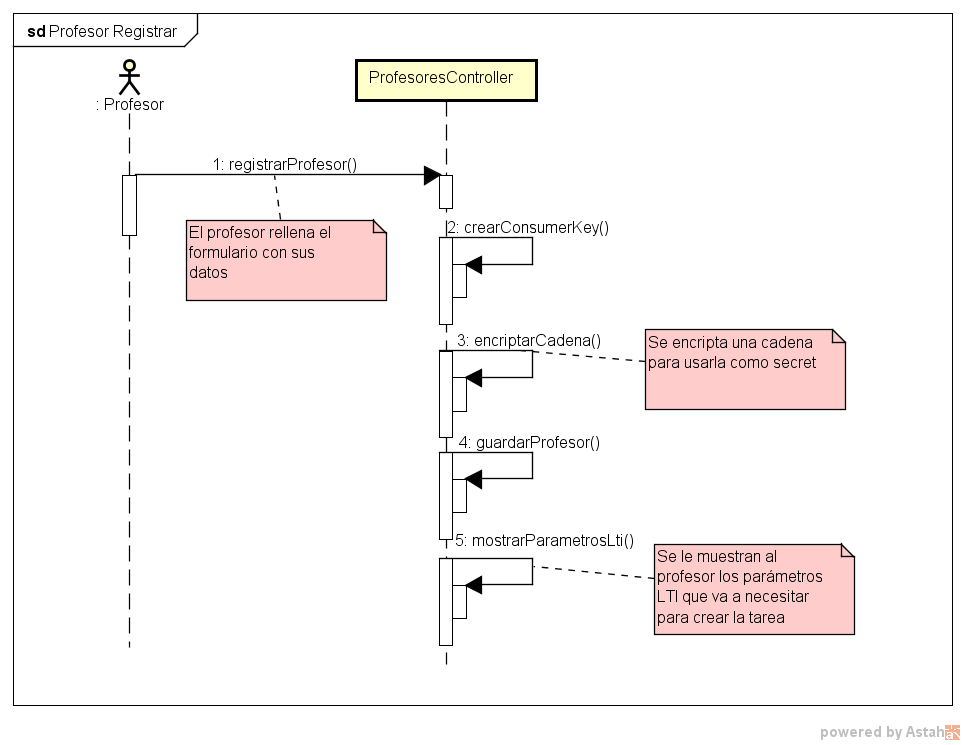


Ilustración : Diagrama de secuencia "Registro del profesor en la aplicaicón"

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

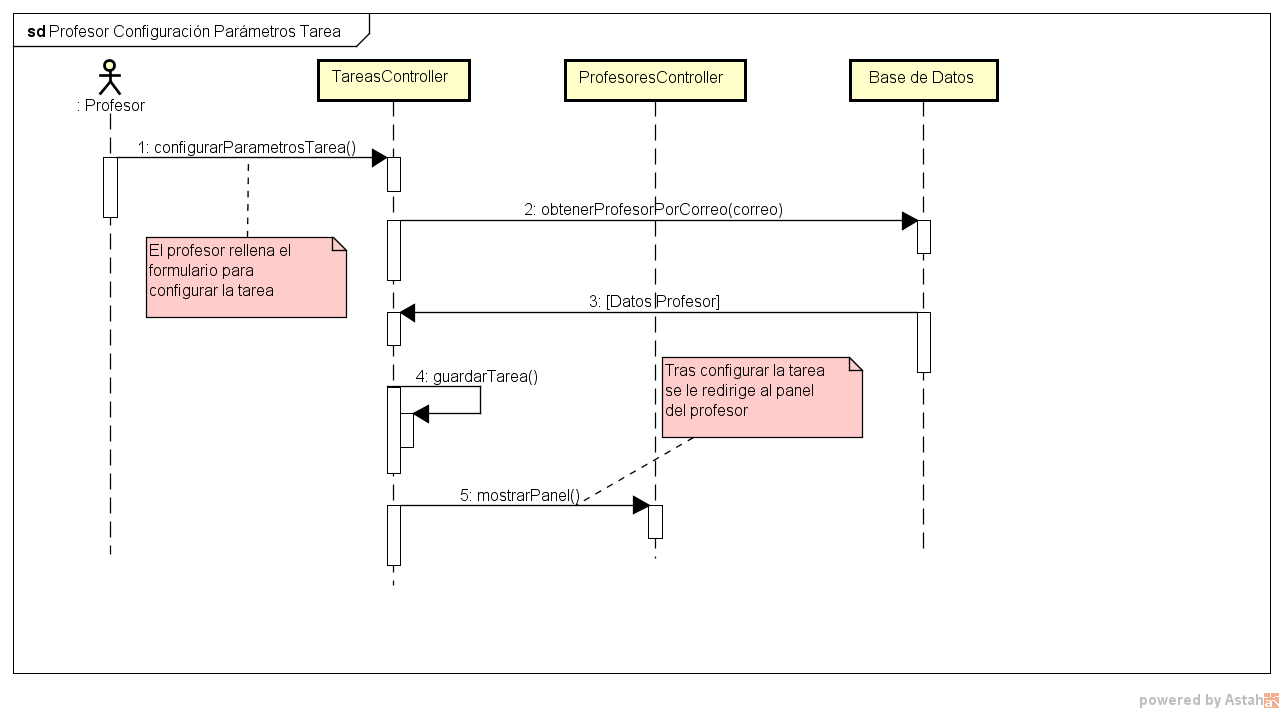


Ilustración : Diagrama de secuencia "Profesor configura parámetros de la tarea"

* Diagrama de secuencia: “Profesor sube test”

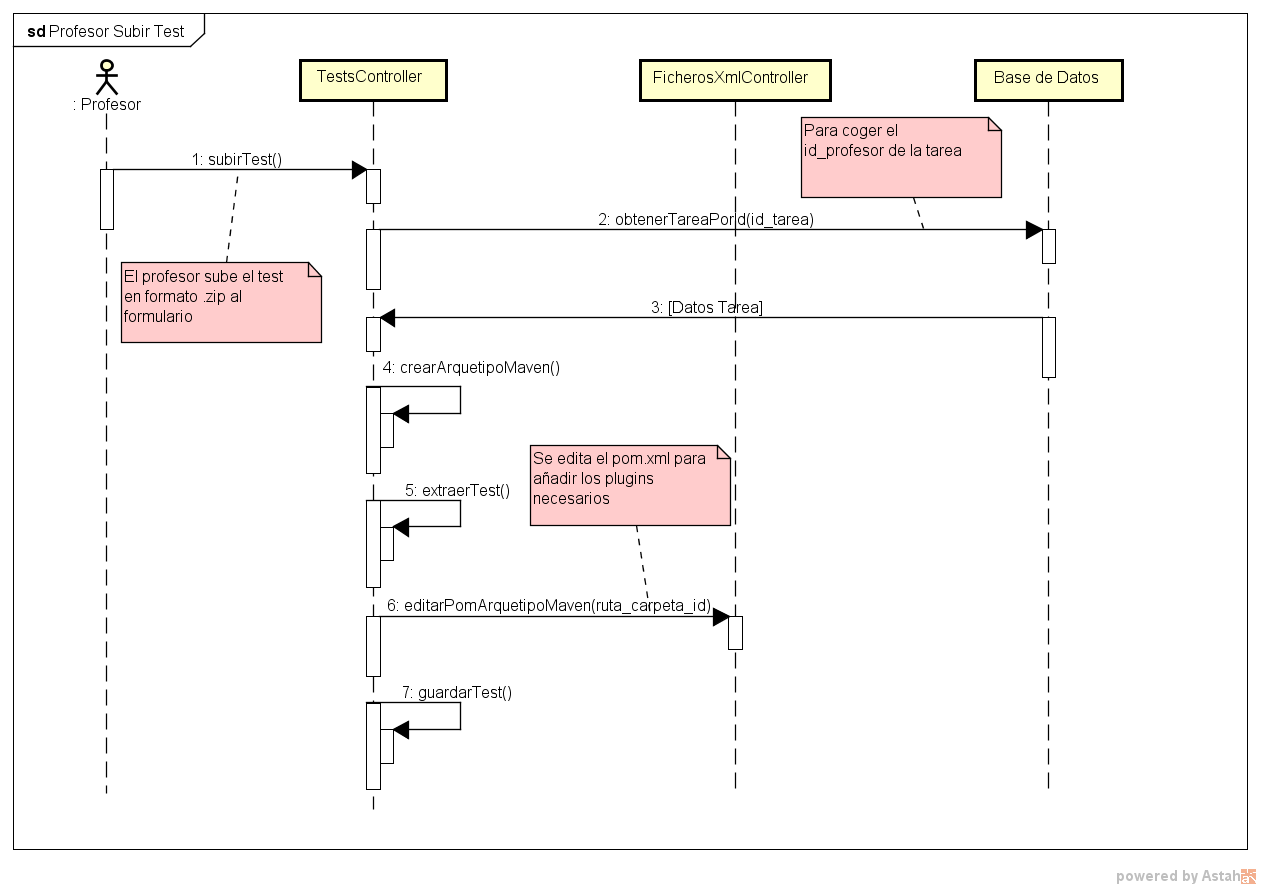


Ilustración : Diagrama de secuencia "Profesor sube test"

* Diagrama de secuencia: “Alumno sube práctica”

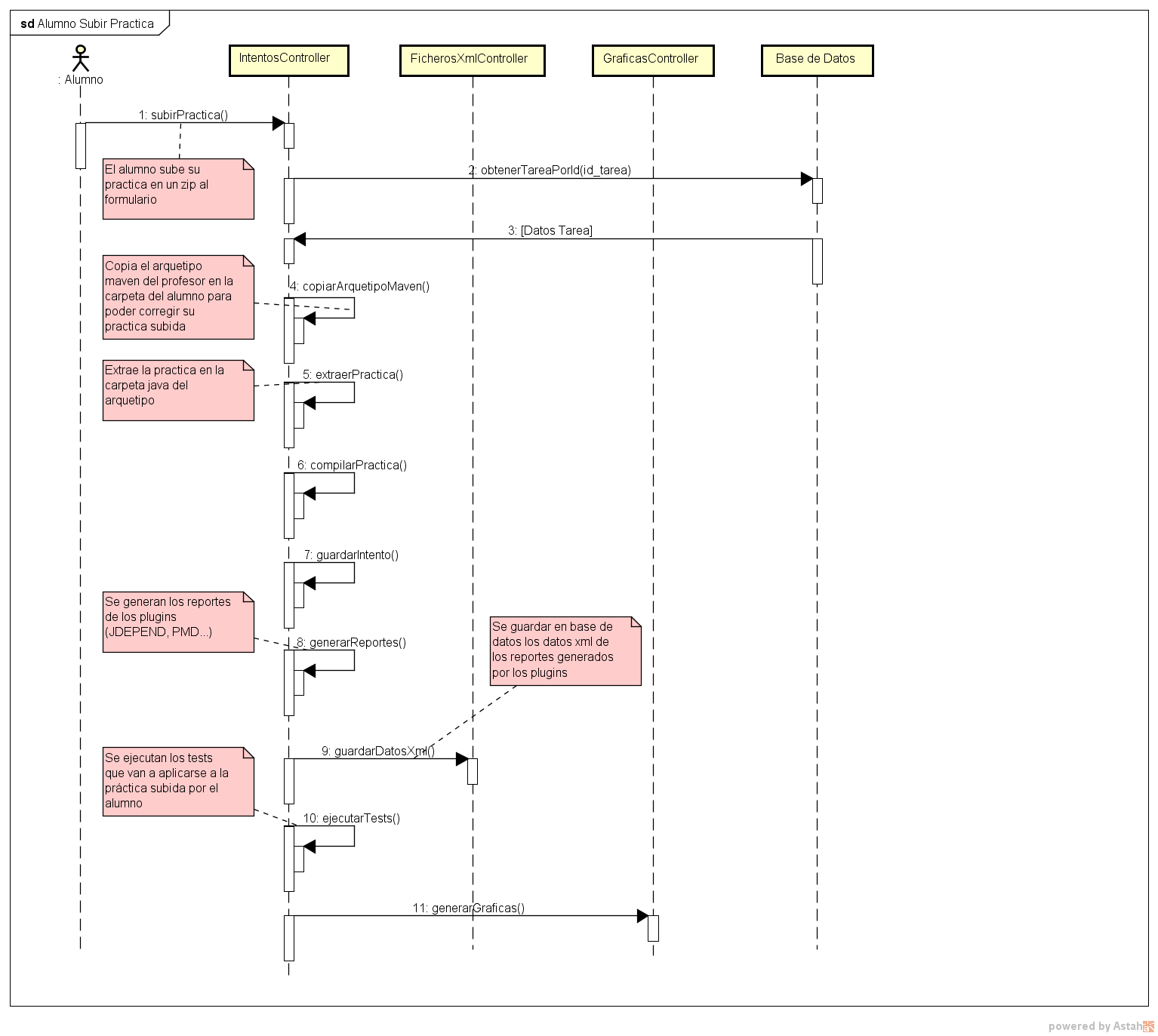


Ilustración : Diagrama de secuencia "Alumno sube práctica"

* Diagrama de secuencia: “Alumno sube práctica incorrecta”

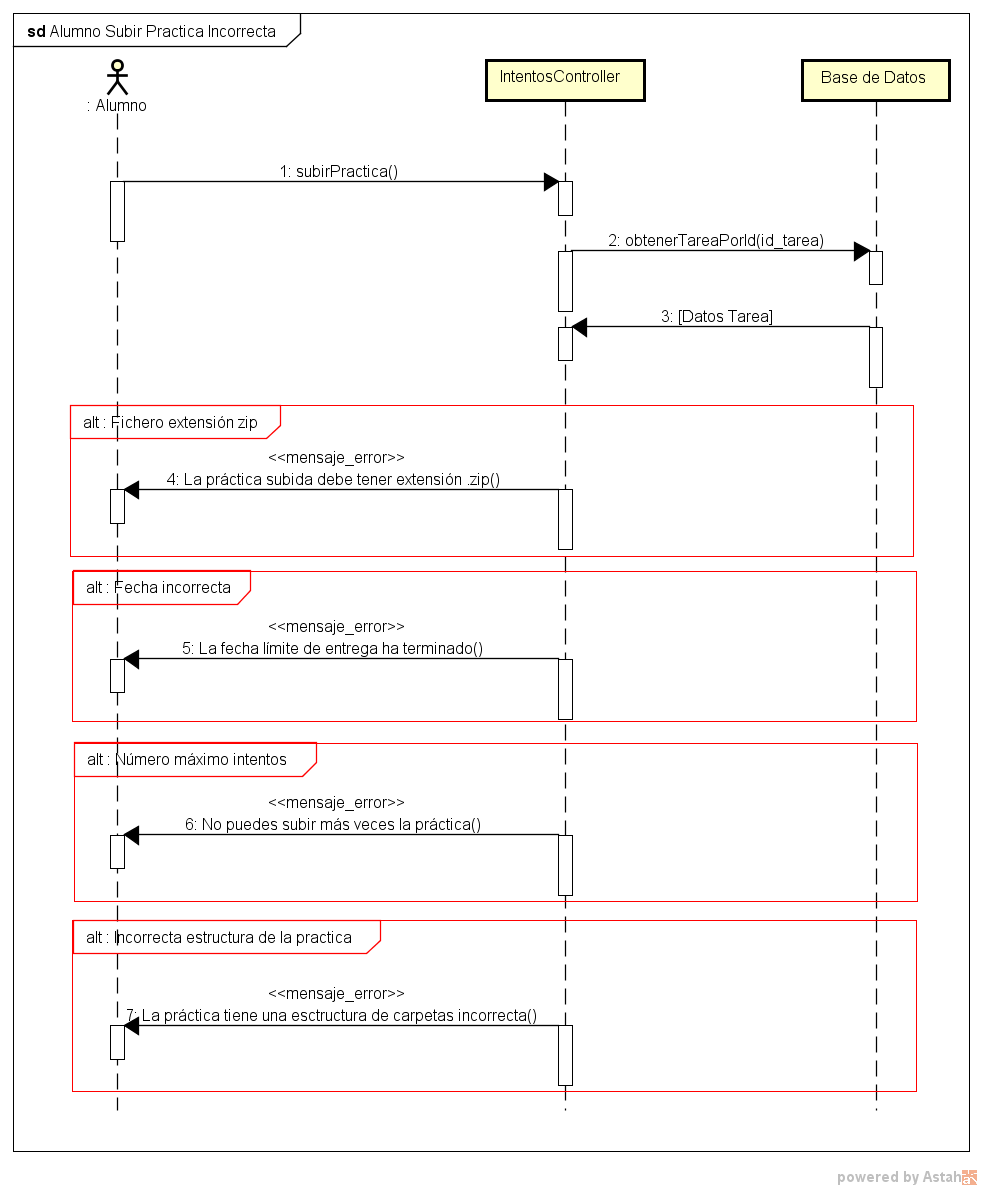


Ilustración : Diagrama de secuencia "Alumno sube práctica incorrecta"

## Diseño arquitectónico

### Diagrama de despliegue

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

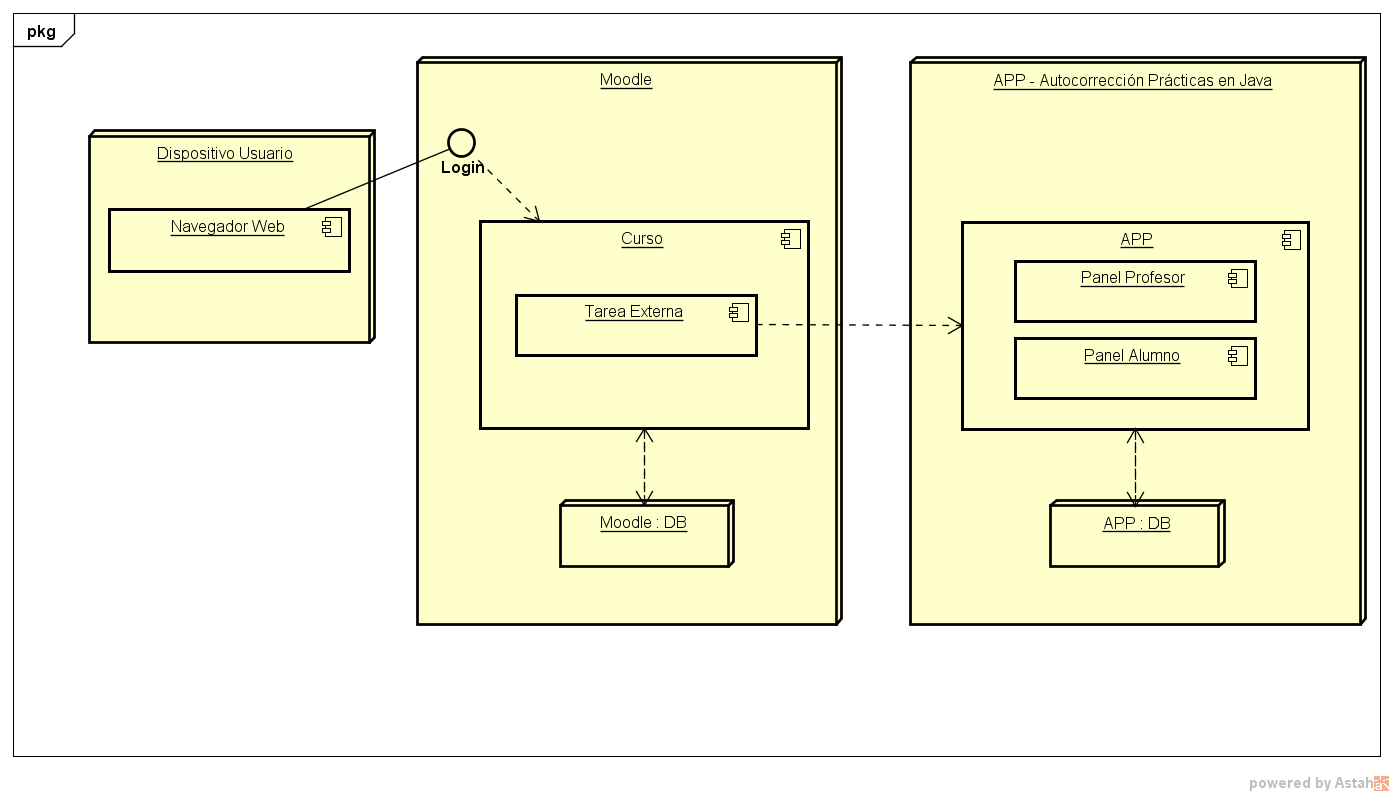


Ilustración : Diagrama de despliegue

# Manual del programador

## 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.

## Estructura de directorios

….

## 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 viene con una estructura de carpetas por defecto, y por lo tanto hay muchas carpetas que no han sido utilizadas ni modificadas. El directorio raíz de la aplicación sería el siguiente:

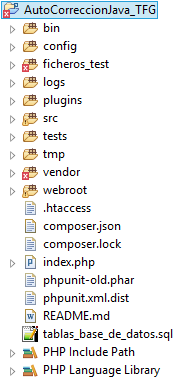


Ilustración : Directorio raíz de la aplicación

Las carpetas que no han sido utilizadas ni modificadas son: bin, logs, plugins y tmp.

* Directorio AutoCorreccionJava\_TFG/**config**:

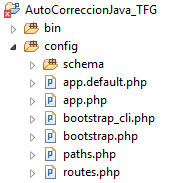


Ilustración : Directorio “config” de la aplicación

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:**

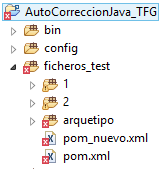


Ilustración : Directorio "ficheros\_test" de la aplicación

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**:

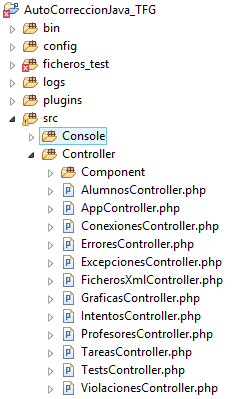


Ilustración : Directorio "src/Controller" de la aplicación

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**:

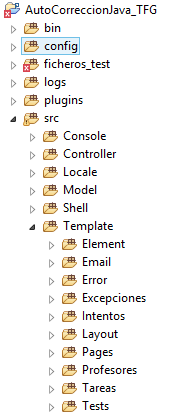


Ilustración : Directorio "src/Template" de la aplicación

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”.

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

* Directorio AutoCorreccionJava\_TFG/**src/Model**:

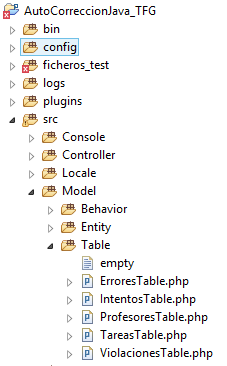


Ilustración : Directorio "src/Model" de la aplicación

Como se ha explicado en el apartado (DIAGRAMA DE PAQUETES, poner enlace), CakePHP no obliga a tener creadas 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**:

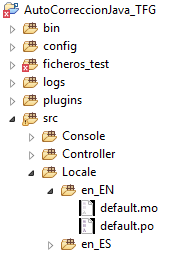


Ilustración : Directorio "src/Locale" de la aplicación

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

* Directorio AutoCorreccionJava\_TFG/**vendor**:

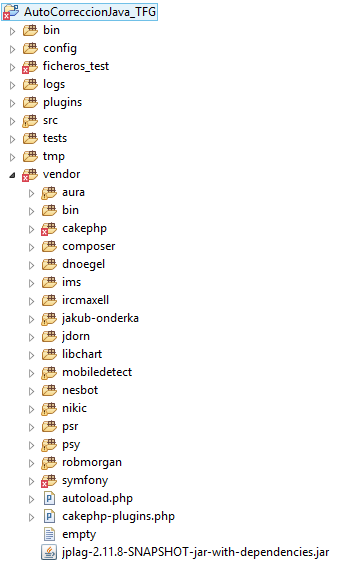


Ilustración : Directorio "vendor" de la aplicación

Dentro de esta carpeta van a estar las librerías externas, de las cuales las únicas que han sido añadidas son: “ims”, encargada de establecer la conexión LTI entre Moodle y la aplicación, “libchart”, librearía utilizada para la generación de gráficas y “jplag-2.11.8-SNAPSHOT-jar-with-dependencies.jar”, utilizado para poder generar el reporte de plagios de las prácticas.

* Directorio AutoCorreccionJava\_TFG/**webroot:**

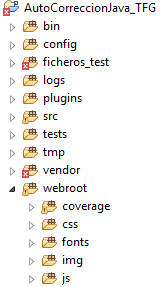


Ilustración : Directorio "webroot" de la aplicación

Contiene las carpetas con los ficheros css, javascript y las diferentes imágenes utilizadas en la aplicación. Además contiene la carpeta con los ficheros correspondientes a la cobertura de código. **MEJOR ELIMINAR LA CARPETA DE COBERTURA NO?**

## Instalación de herramientas

### Sistema operativo

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

### 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>

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



Ilustración : Instalar WampServer paso 1

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



Ilustración : Instalar WampServer paso 2

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



Ilustración : Instalar WampServer paso 3

1. 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.

### Base de datos

#### Creación de las bases de datos

Para realizar el proyecto van a ser necesarias crear dos bases de datos: una de nombre “moodle” en la que van a almacenarse todas las tablas que Moodle (que posteriormente se explicará su proceso de instalación) va a necesitar, y otra de nombre “autocorrecc\_tfg” en la cual van a guardarse las tablas propias de nuestra aplicación.

Los pasos a seguir son los siguientes:

1. Abrimos el navegador y accedemos a “localhost/phpmyadmin/”, entramos en la pestaña de “Bases de datos” y la pantalla que se nos mostrará será la siguiente:



Ilustración : phpMyAdmin

1. Creamos la base de datos de nombre “moodle” y cotejamiento “utf8\_general\_ci”:



Ilustración : phpMyAdmin creación BD Moodle

1. Pulsamos en “Crear” y ya tendremos la base de datos creada y necesaria para la instalación del Moodle que viene a continuación.

Para crear la base de datos “autocorrecc\_tfg” el proceso realizado sería el mismo.

#### HeidiSQL (BORRAR??)

Se utilizará la herramienta “HediSQL” para crear, eliminar, editar y visualizar las diferentes tablas de nuestra base de datos.

El proceso de instalación de dicha herramienta es el siguiente:

1. Accedemos a la web oficial de HeidiSQL cuyo enlace es el siguiente:

<http://www.heidisql.com/>

1. Accedemos a la pestaña “Downloads” y descargamos el instalador pinchando sobre el botón “HeidiSQL Installer”:



Ilustración : Descarga HeidiSQL

1. Una vez descargado el fichero .exe simplemente hay que ejecutarlo para proceder a su instalación, y en el momento que nos pidan conectarnos a una base de datos tendremos que introducir nuestra base de datos “autocorrecc\_tfg” creada en el apartado anterior.

#### Creación de las tablas

Gracias al script .sql 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 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 : HeidiSQL sql creación de tablas

1. 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.

### 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/>

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



Ilustración : Maven descarga

1. 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.

### 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…).
* **JDEPEND**: Ofrece información sobre métricas a nivel de paquetes (abstractness, inestability, afferent couplings…).
* **PMD**: Ofrece información acerca de violaciones de código cometidas (variables sin usar, métodos privados sin usar…).
* **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).

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

### 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/>

1. 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 : CakePHP Instalación

### Bootstrap (BORRAR)

Para facilitar la creación del diseño de nuestras interfaces, se utilizará la librería Bootstrap.

Su instalación es tan simple como:

1. Acceder a su web oficial:

<http://getbootstrap.com/>

1. Seleccionar “Download Bootsrap” y volver a seleccionar dicha opción desde la siguiente página:

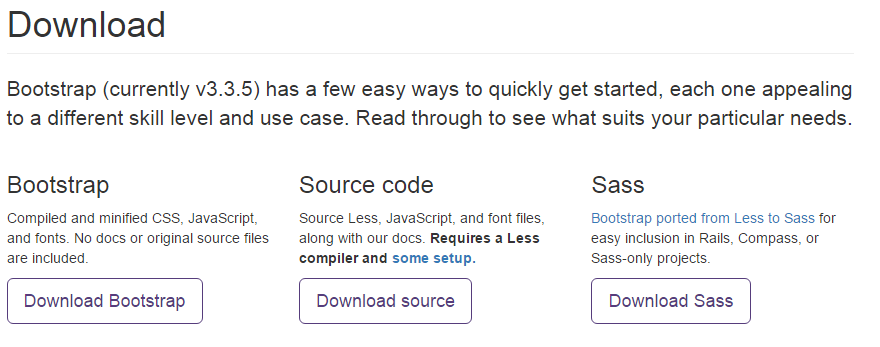
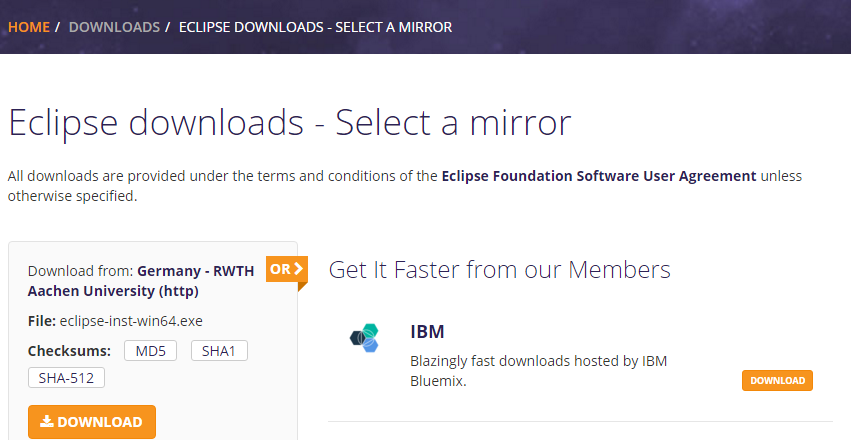


Ilustración : Bootstrap descarga

1. Se nos descargará un .zip el cual tendremos que extraer en las correspondientes carpetas (\css y \js) de nuestra carpeta “webroot” creada automáticamente tras crear nuestra aplicación utilizando el framework CakePHP.



### Moodle

La versión que va a ser instalada de Moodle es la 2.9.2.

El proceso de instalación es el siguiente:

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/>

1. Copiamos el fichero descomprimido en la siguiente carpeta:

“C:\wamp\www”

1. Abrimos el navegador e introducimos la siguiente URL para comenzar el proceso de instalación de Moodle:

“localhost/moodle”

1. Se nos mostrará la siguiente pantalla:



Ilustración : Moodle instalación seleccionar idioma

1. 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:



Ilustración : Moodle instalación ajuste base de datos

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



Ilustración : Moodle instalación datos admin

1. 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
2. La instalación de Moodle estaría ya realizada.

### 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>

1. Descomprimimos el zip descargado y le colocamos dentro del siguiente directorio:

“C:\wamp\www\moodle\local”

1. 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.

### PHPUnit

No ha sido necesario instalar PHPUnit, sino que simplemente se ha descargado el fichero .phar correspondiente a PHPUnit [6].

### 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 [7].

## Conexión LTI

Uno de los aspectos más fuertes e importantes de la aplicación es que debe poder conectarse con la plataforma *Moodle*, de manera que únicamente se pueda acceder a la aplicación a través de las tareas, de tipo “herramienta externa”, creadas por el profesor desde *Moodle*.

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 conectarse con *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, 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 pasará 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…).

## Pruebas

Las pruebas que se han llevado a cabo han sido dos: 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.

### Ejecutar pruebas

#### 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:

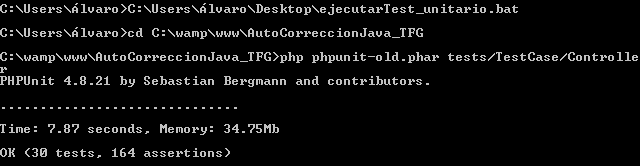


Ilustración : Salida cmd obtenida tras ejecuctar pruebas unitarias

#### 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:

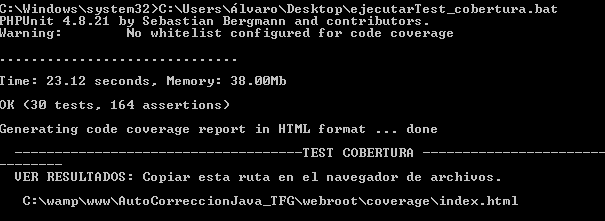


Ilustración : Salida cmd obtenida tras ejecutar pruebas de cobertura

1. 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\webroot\coverage\index.html*

1. Se abrirá una web con los resultados:

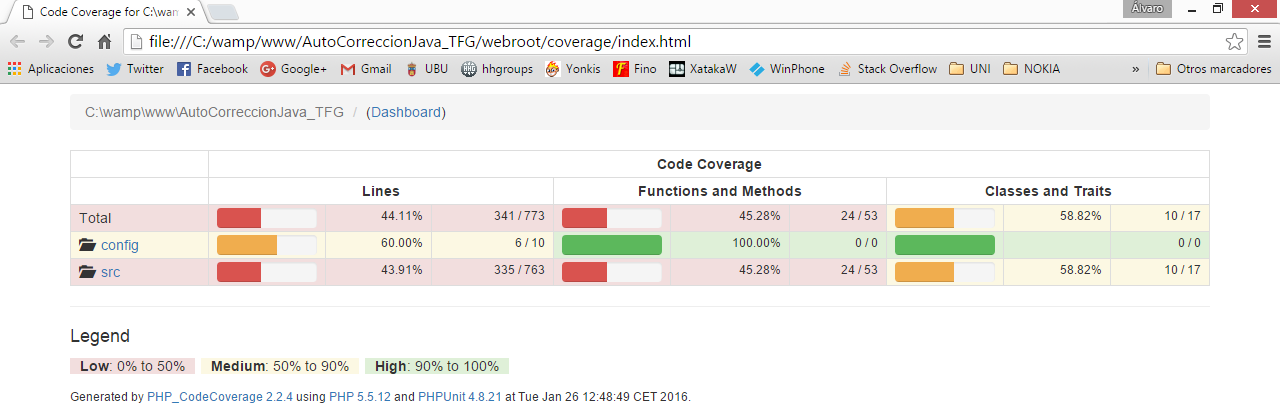


Ilustración : Resultado de test de cobertura

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:

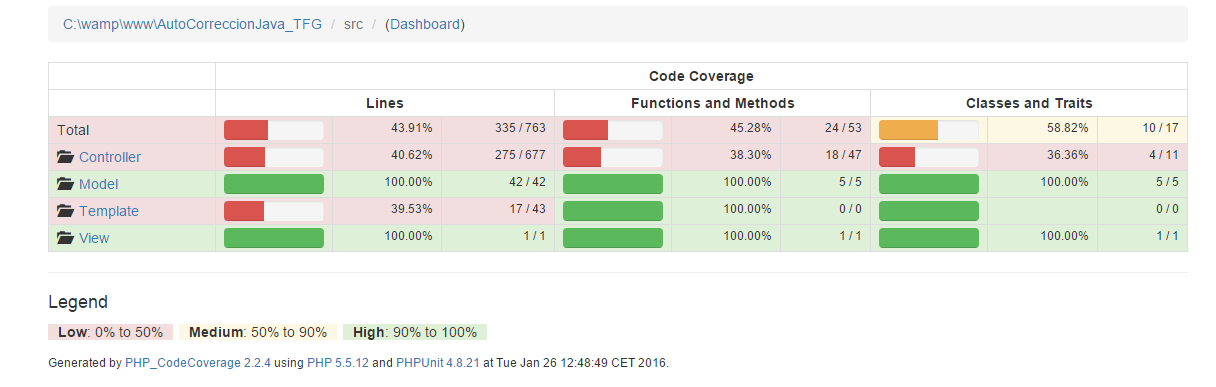


Ilustración : Resultado de test de cobertura carpeta "src"

### 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 [9].

Visualizando la anterior ilustración, 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.

Se ha intentado realizar el máximo número de pruebas unitarias, el problema que se ha presentado es que había controladores que tenían métodos privados, además de otros que tenían métodos de generación de gráficas, 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.

#### Probando el controlador

Como se ha mencionado anteriormente, no ha sido probar todas las funciones de los controladores ya que hay muchos que tienen métodos privados y otros métodos que no permitían 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 |

Tabla : Tabla pruebas unitarias

### 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 [10].

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:

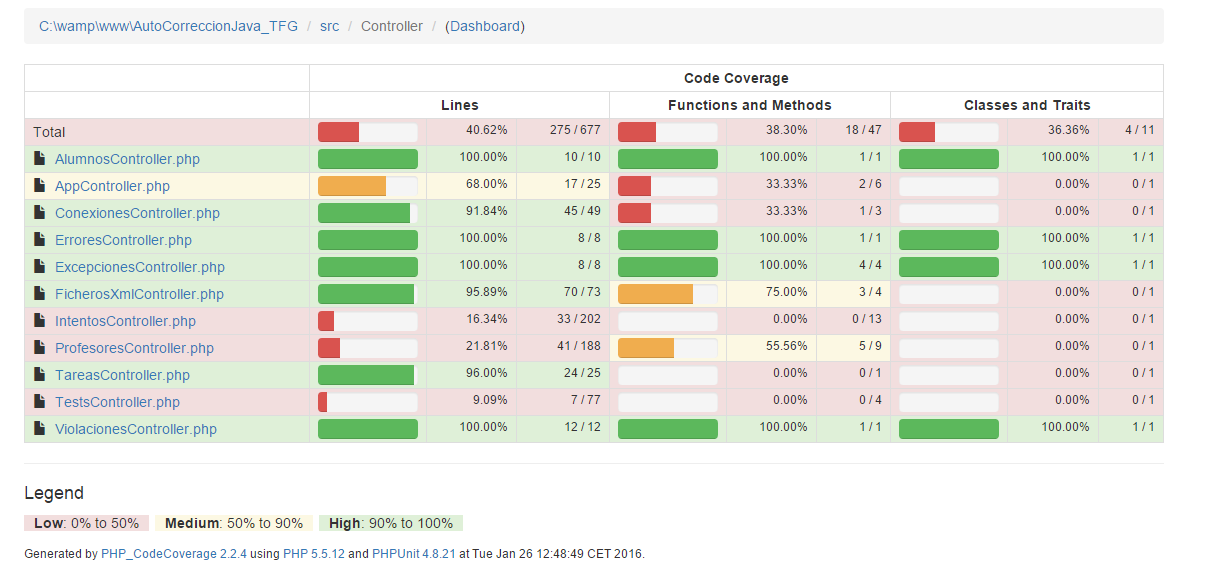


Ilustración : Cobertura de código de los controladores

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.

# 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.

## Manual del profesor

### Registrar profesor en la aplicación

#### 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 desde el siguiente enlace:

<http://localhost/AutoCorreccionJava_TFG/Profesores/registrar>

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



Ilustración : Formulario de registro del profesor

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

1. Una vez rellenado el formulario, pinchamos en “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 : Parámetros LTI obtenidos

#### 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 [8].

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 : 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 : Moodle selección herramienta externa

1. 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:

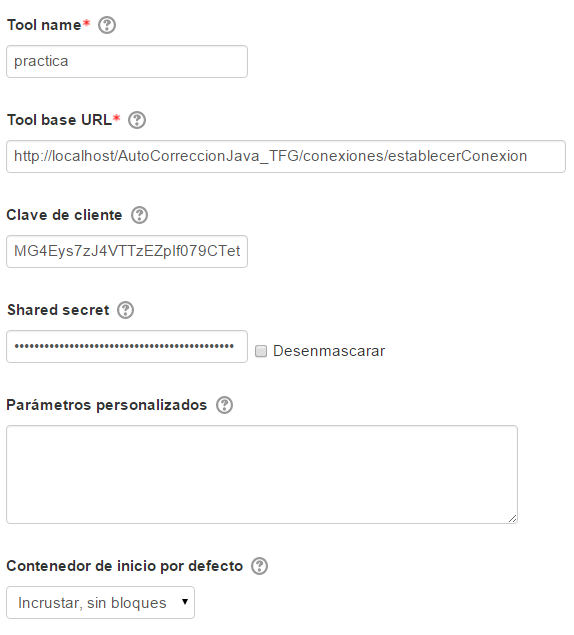


Ilustración : Moodle configuración parámetros de la tarea paso 1

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



Ilustración : 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.

1. 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:



Ilustración : Moodle tarea creada

### Panel principal

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



Ilustración : 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:

### 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 le 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:

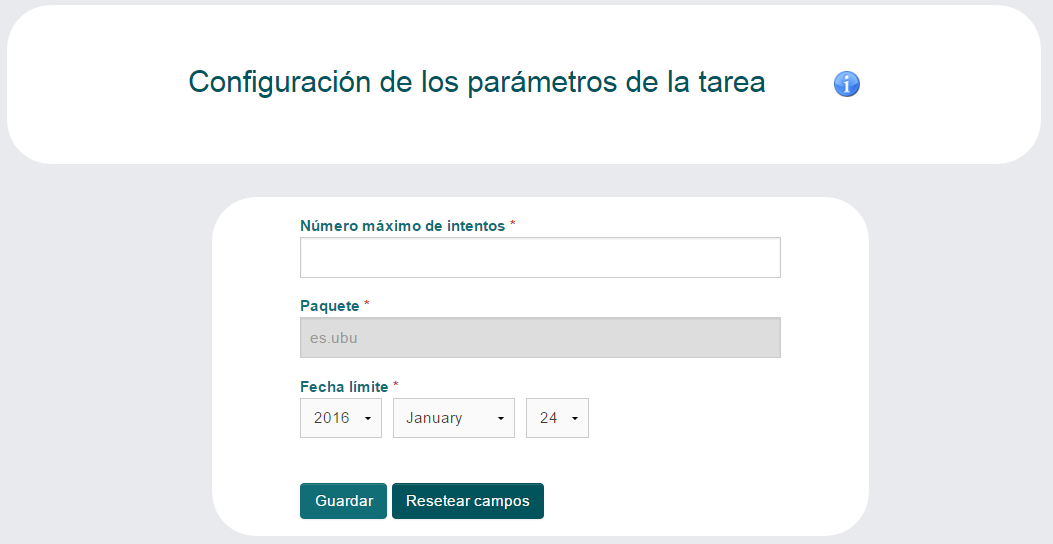


Ilustración : Panel del profesor de configuración de parámetros de la tarea

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

### 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:

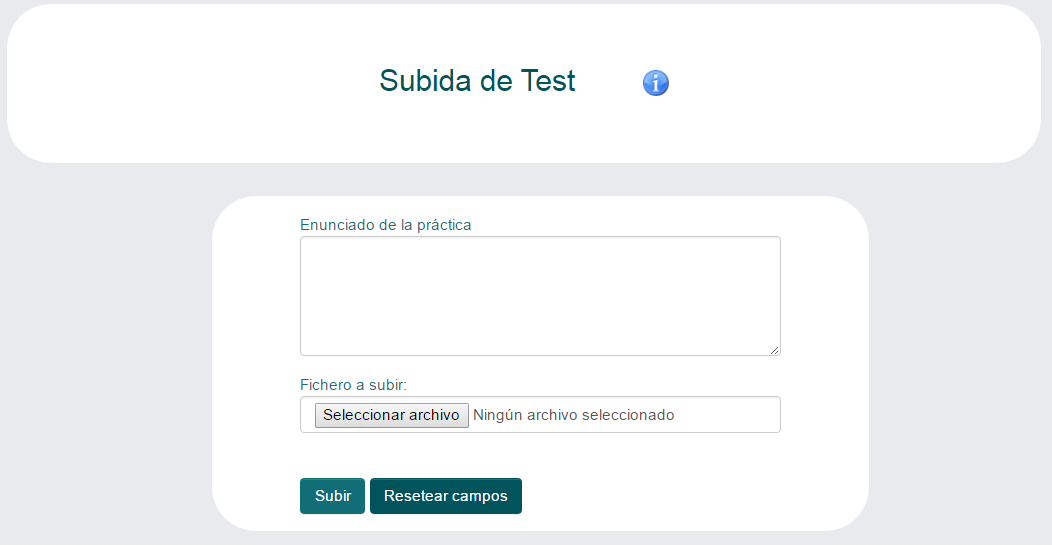


Ilustración : Panel del profesor de subida de test

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

### 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 filtraciones, así como ordenaciones de los datos de la misma. Además también se permite que la descarga de las prácticas de los alumnos en formato .zip.

#### Aspecto inicial

El aspecto de este panel es el siguiente:



Ilustración : Panel del profesor de estadísticas

#### 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 : Panel del profesor de estadísticas consluta de reportes

Como se trata de una práctica que sí que ha pasado los test, no el botón correspondiente al reporte de errores aparece desactivado.

### 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 diferentes 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.

#### Aspecto inicial

El aspecto de este panel sería el siguiente:



Ilustración : Panel del profesor de gráficas

#### Selección de gráficas

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

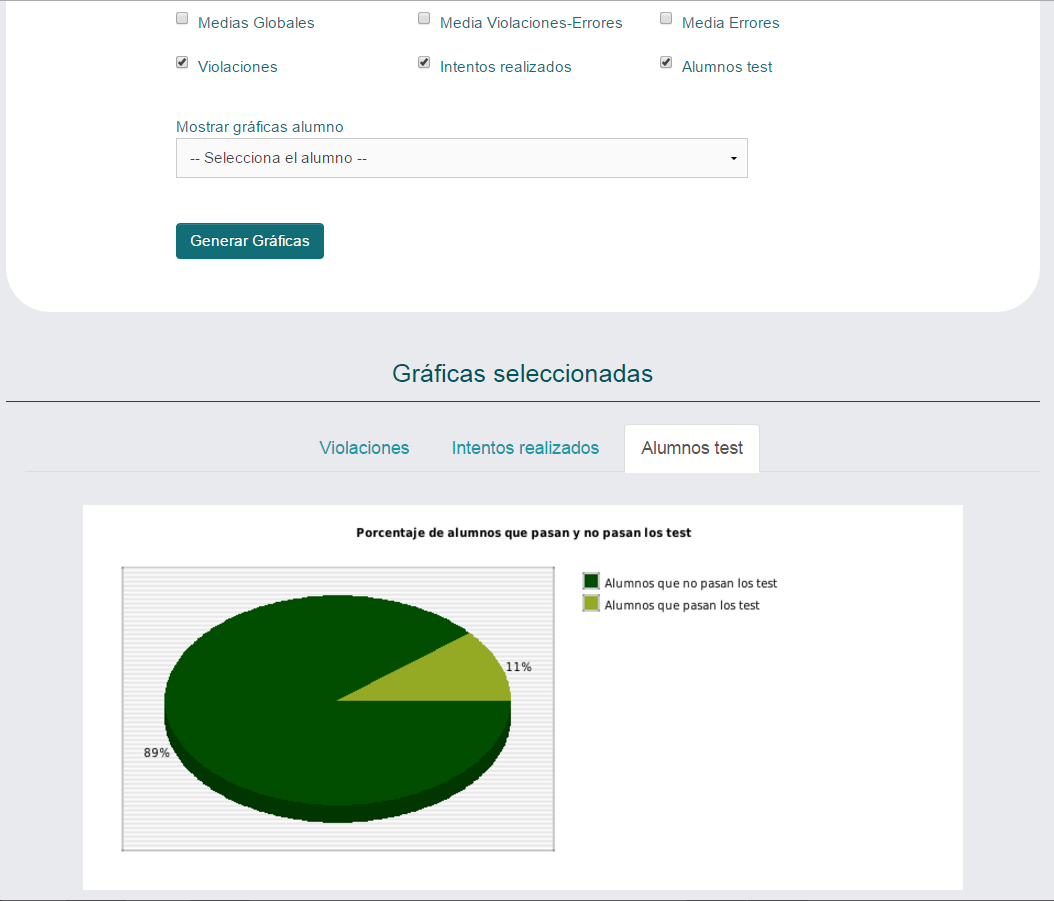


Ilustración : Panel del profesor de selección de gráficas

### 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.

#### Aspecto inicial

El aspecto de este panel es el siguiente:



Ilustración : Panel del profesor de comprobación de plagios

#### Comprobar plagios

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

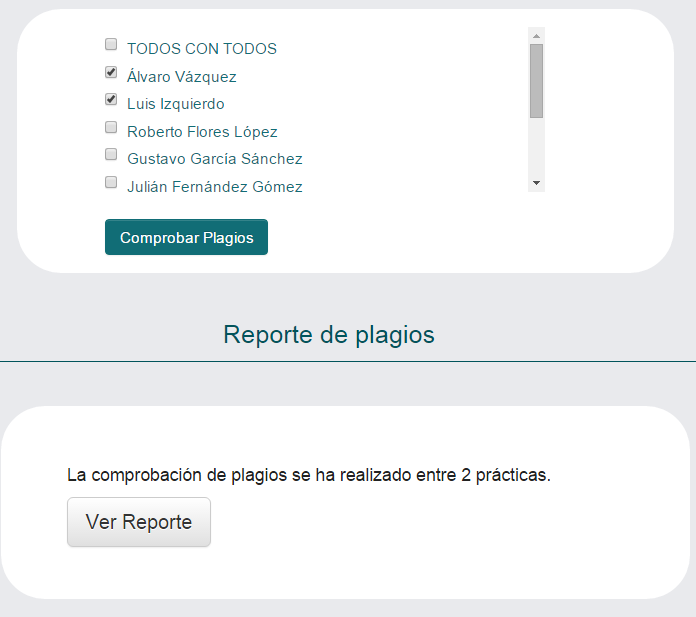


Ilustración : Panel del profesor de selección de plagios

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

## 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.

### 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.

#### Ninguna práctica subida

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



Ilustración : Panel principal del alumno

#### 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:



Ilustración : Panel del alumno visualización de gráficas y reportes

### Información y Enunciado

El alumno podrá consultar la información referente a qué características debe de tener la práctica que va a subir, 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:



Ilustración : Panel del alumno información y enunciado

### 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:



Ilustración : Panel del alumno parámetros de la tarea

### 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 : Panel del alumno error subida de práctica

### 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 : Panel del alumno ver reportes y gráficas

## 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 : Cambiar idioma

# 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] «Vistas - CakePHP cookbook». [En línea]. Disponible en: http://book.cakephp.org/3.0/en/views.html.

[4] «Controladores - CakePHP cookbook». [En línea]. Disponible en: http://book.cakephp.org/3.0/en/controllers.html.

[5] «Modelo - CakePHP cookbook». [En línea]. Disponible en: http://book.cakephp.org/3.0/en/orm/saving-data.html.

[6] «PHPUnit - descarga fichero .phar». [En línea]. Disponible en: https://phpunit.de/#download.

[7] «Eclipse for PHP Developers». [En línea]. Disponible en: http://www.eclipse.org/downloads/packages/eclipse-php-developers/heliosr.

[8] «Moodle - Creación de un curso». [En línea]. Disponible en: https://docs.moodle.org/all/es/A%C3%B1adiendo\_un\_nuevo\_curso.

[9] «Prueba unitaria», *Wikipedia, la enciclopedia libre*. 30-nov-2015.

[10] «Cobertura de código», *Wikipedia, la enciclopedia libre*. 01-sep-2015.