# Plan de proyecto software

## Introducción

Esta es la introducción…

## Planificación temporal del proyecto

Gráfico con todas las iteraciones:

### 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.





### 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.





### 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.





### 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.





### 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.

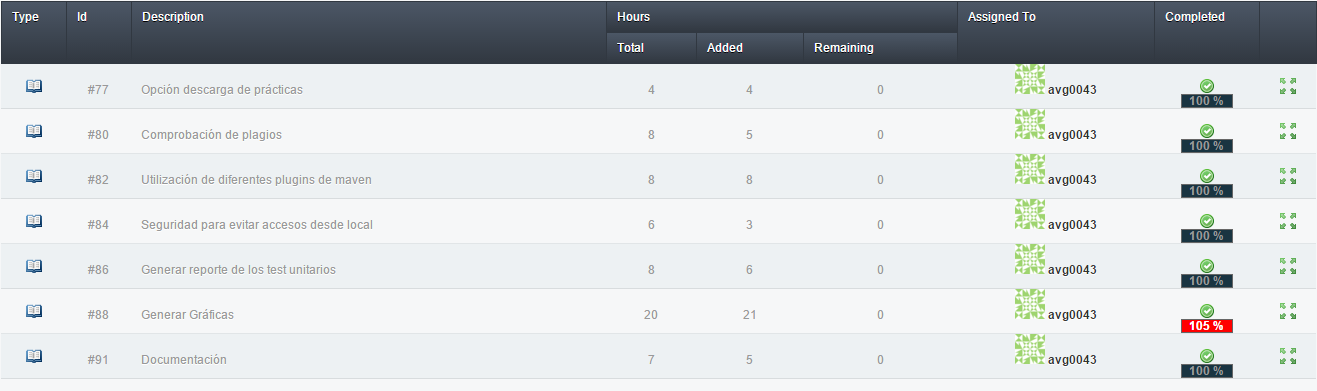




### Iteración 6 (26 Nov 2015 a 10 Dic 2015)

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”.





### 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.

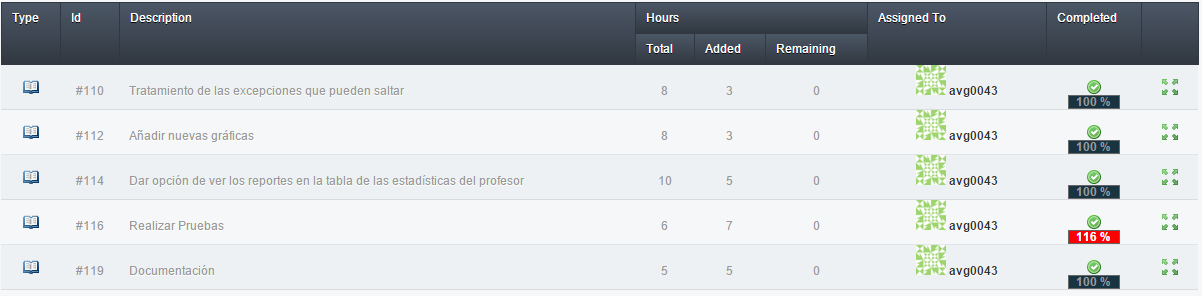


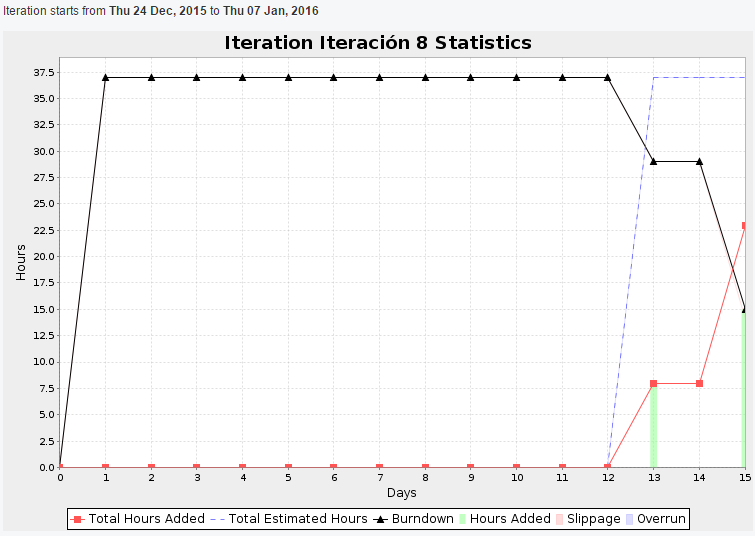


### Iteración 8 (24 Dic 2015 a 7 Ene 2016)

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”.





## Estudio de viabilidad

### Estudio de viabilidad legal

### 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 [1], 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 € |

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:

#### Análisis de costes - beneficio

# 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 el curso**: El profesor accederá a Moodle con sus credenciales y podrá crear un curso.
* **RF-M02: Crear la tarea:** El profesor creará en Moodle y dentro del curso creado, 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-M03: 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.
* **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.
* **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> | |

#### Pertenecientes a Moodle

|  |  |  |
| --- | --- | --- |
| **RF-M01** | **Crear el curso** | |
| **Descripción** | Crear un curso en Moodle para la posterior creación de tareas dentro del mismo. | |
| **Precondición** | Acceder a Moodle con los datos de acceso del profesor. | |
| **Secuencia Normal** | Paso | Acción |
| 1 | El profesor accede a Moodle con sus datos. |
| 2 | Crea el curso, estableciendo su nombre y la información que desee. |
| **Post-condición** | El nuevo curso queda creado y accesible para entrar. | |
| **Excepciones** | Paso | Acción |
| 1 | Si el profesor introduce sus datos de acceso incorrectos, no podrá acceder a Moodle. |
| **Importancia** | Alta. | |
| **Frecuencia** | Baja. | |
| **Urgencia** | Alta. | |

|  |  |  |
| --- | --- | --- |
| **RF-M02** | **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. | |

|  |  |  |
| --- | --- | --- |
| **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. | |

|  |  |  |
| --- | --- | --- |
| **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. | |

#### 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. | |
| **Excepciones** | Paso | Acción |
| 1 | El profesor no rellena correctamente los campos del formulario (campos vacíos, datos incorrectos). |
| **Importancia** | Alta. | |
| **Frecuencia** | Media. | |
| **Urgencia** | Alta. | |

|  |  |  |
| --- | --- | --- |
| **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. | |

|  |  |  |
| --- | --- | --- |
| **RF-AW03** | **Subir test** | |
| **Descripción** | El profesor podrá subir los test que desee desde el correspondiente formulario de subida de ficheros. | |
| **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. |
| 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. |
| **Importancia** | Alta. | |
| **Frecuencia** | Media. | |
| **Urgencia** | Alta. | |

|  |  |  |
| --- | --- | --- |
| **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. | |

|  |  |  |
| --- | --- | --- |
| **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. | |

|  |  |  |
| --- | --- | --- |
| **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. | |

|  |  |  |
| --- | --- | --- |
| **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. | |

* **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. | |

|  |  |  |
| --- | --- | --- |
| **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. | |

|  |  |  |
| --- | --- | --- |
| **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. | |

|  |  |  |
| --- | --- | --- |
| **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. | |

|  |  |  |
| --- | --- | --- |
| **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. | |

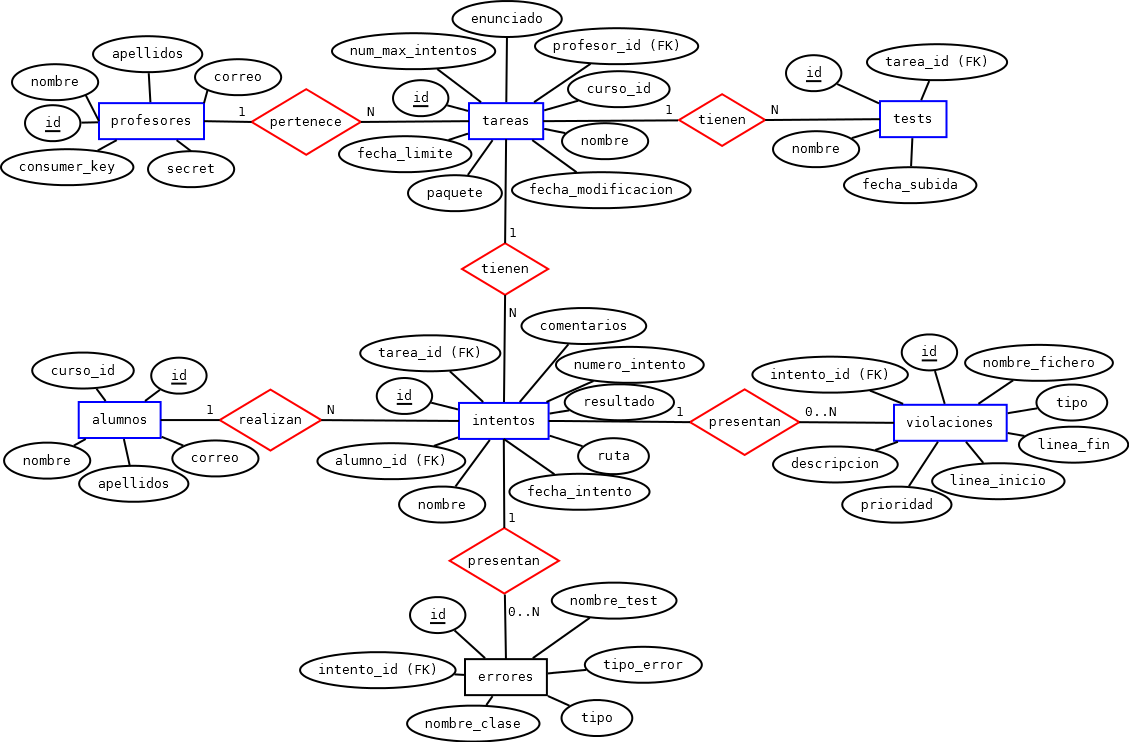
# Especificación de diseño

## 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



#### 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. |

*\* 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. |

*\* 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. |

*\* 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. |

*\* 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. |

*\* 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. |

*\* 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.

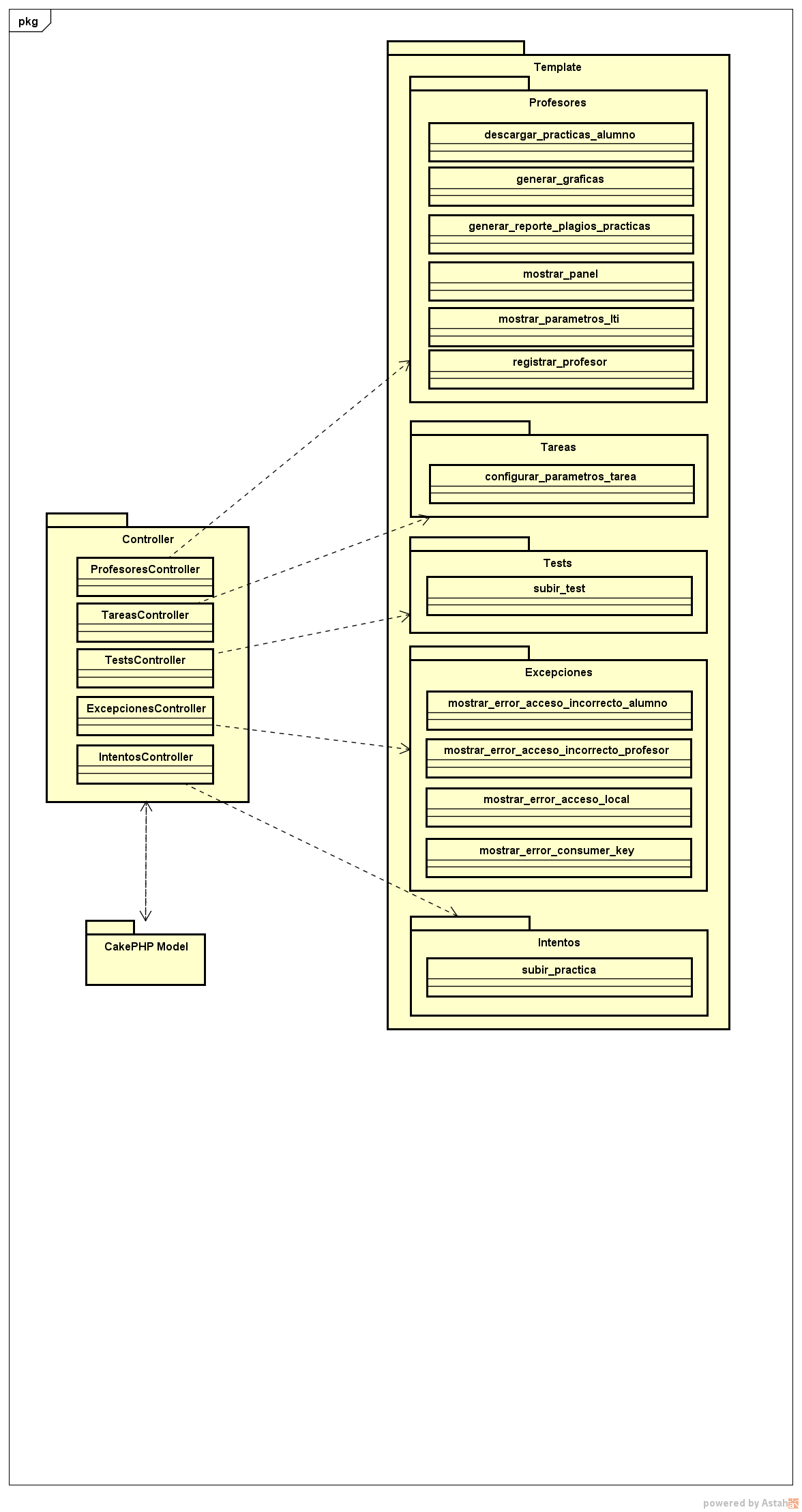
|  |  |  |
| --- | --- | --- |
| **Atributo** | **Tipo** | **Descripción** |
| id | INT | Código que identifica al error de forma única. |
| intento\_id | INT | Código que identifica al intento de subida de práctica asociada. |
| nombre\_clase | VARCHAR | Nombre del fichero .java en el que se ha cometido 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. |

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

*\* Foreign key (FK):*

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

### 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 [2].

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 [3].

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 [4].

### C:\Users\Álvaro\Dropbox\TFG\Diagrama de clases\Class Diagram0.pngDiagrama de clases

**Explicación del diagrama de clases**

La explicación del diagrama de clases y un poco el papel que tienen 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:

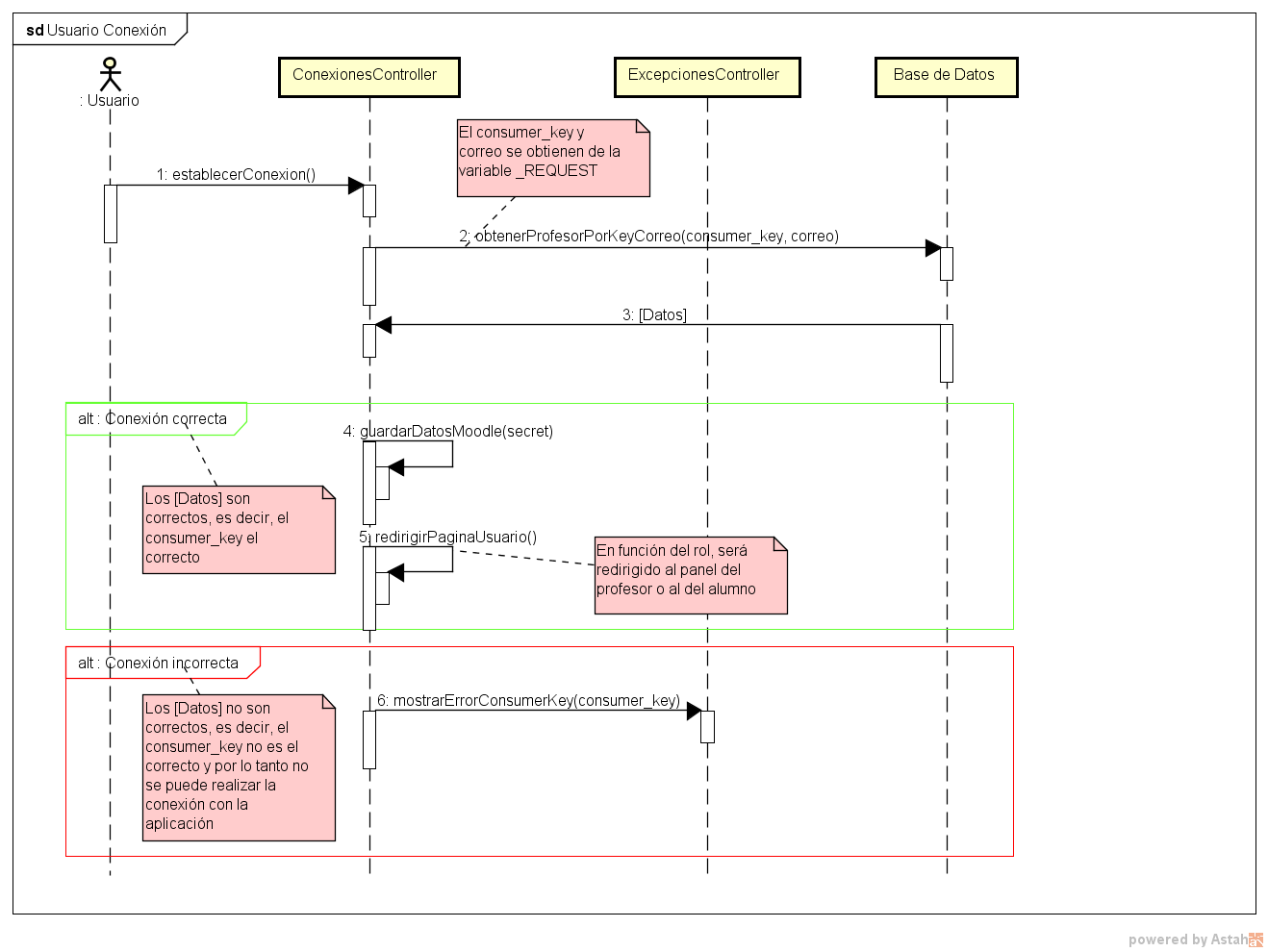
* Si el que accede a la aplicación es un **profesor** entonces:
  1. Si es su primer acceso, será redirigido a “TareasController” para que rellene el formulario obligatorio de configuración de parámetros de la tarea.
  2. En los siguientes accesos, será redirigido al panel principal del profesor que 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.

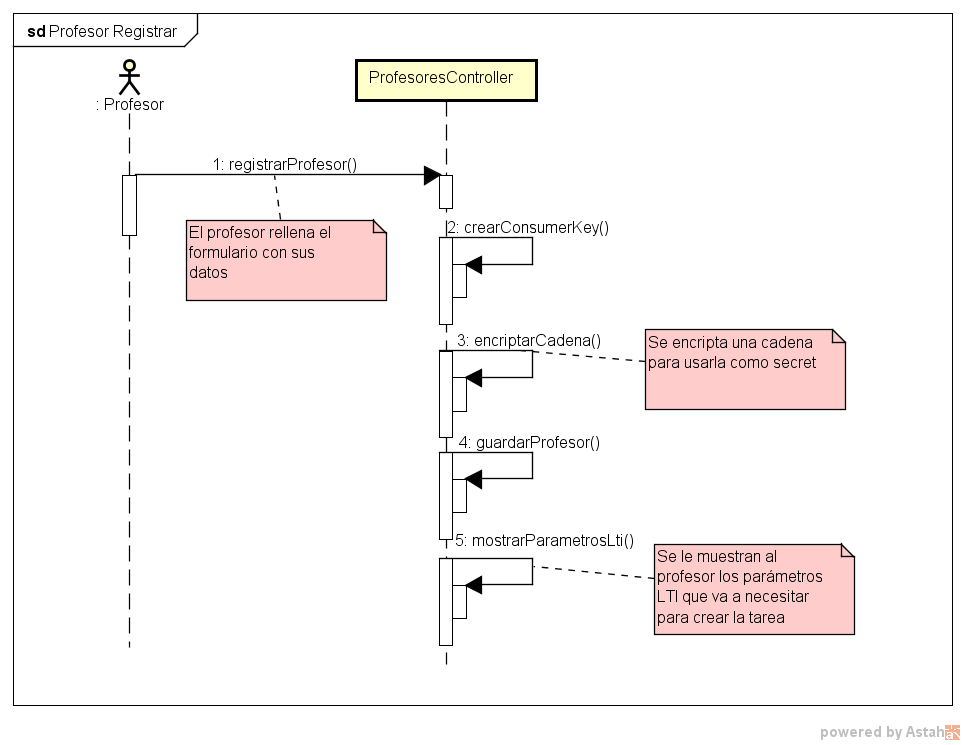
## Diseño procedimental

### Diagramas de secuencias

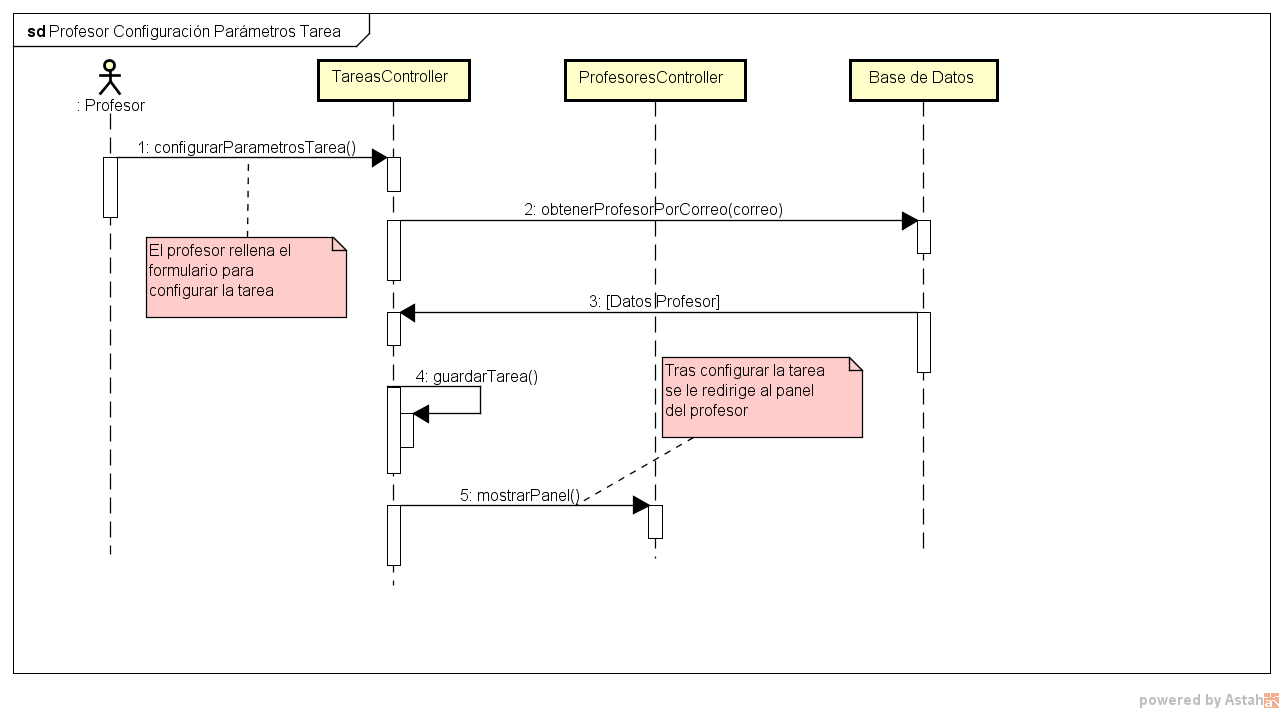
* Diagrama de secuencia: “Conexión del usuario (profesor/alumno)”



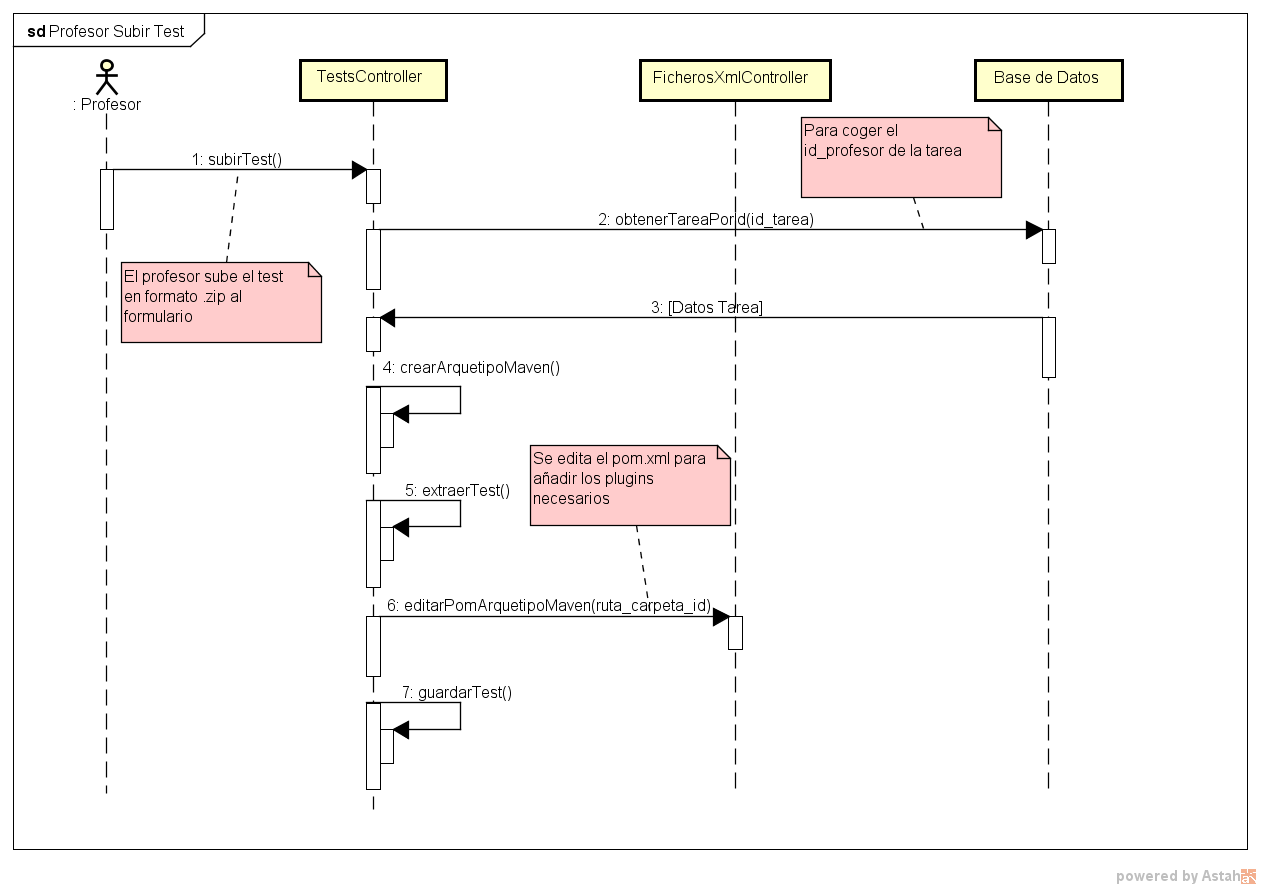
* Diagrama de secuencia: “Registro del profesor”



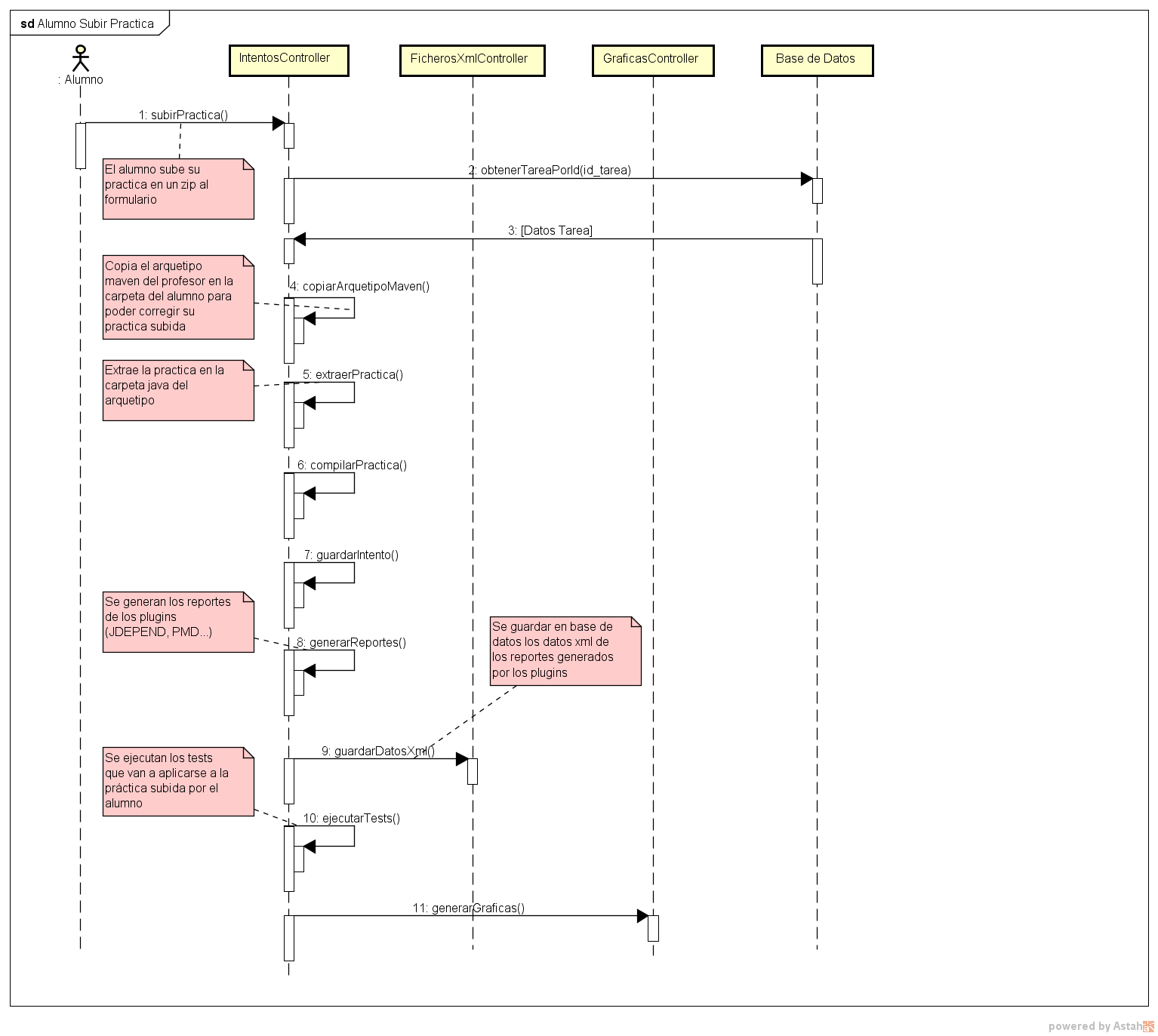
* Diagrama de secuencia: “Profesor configura parámetros de la tarea”



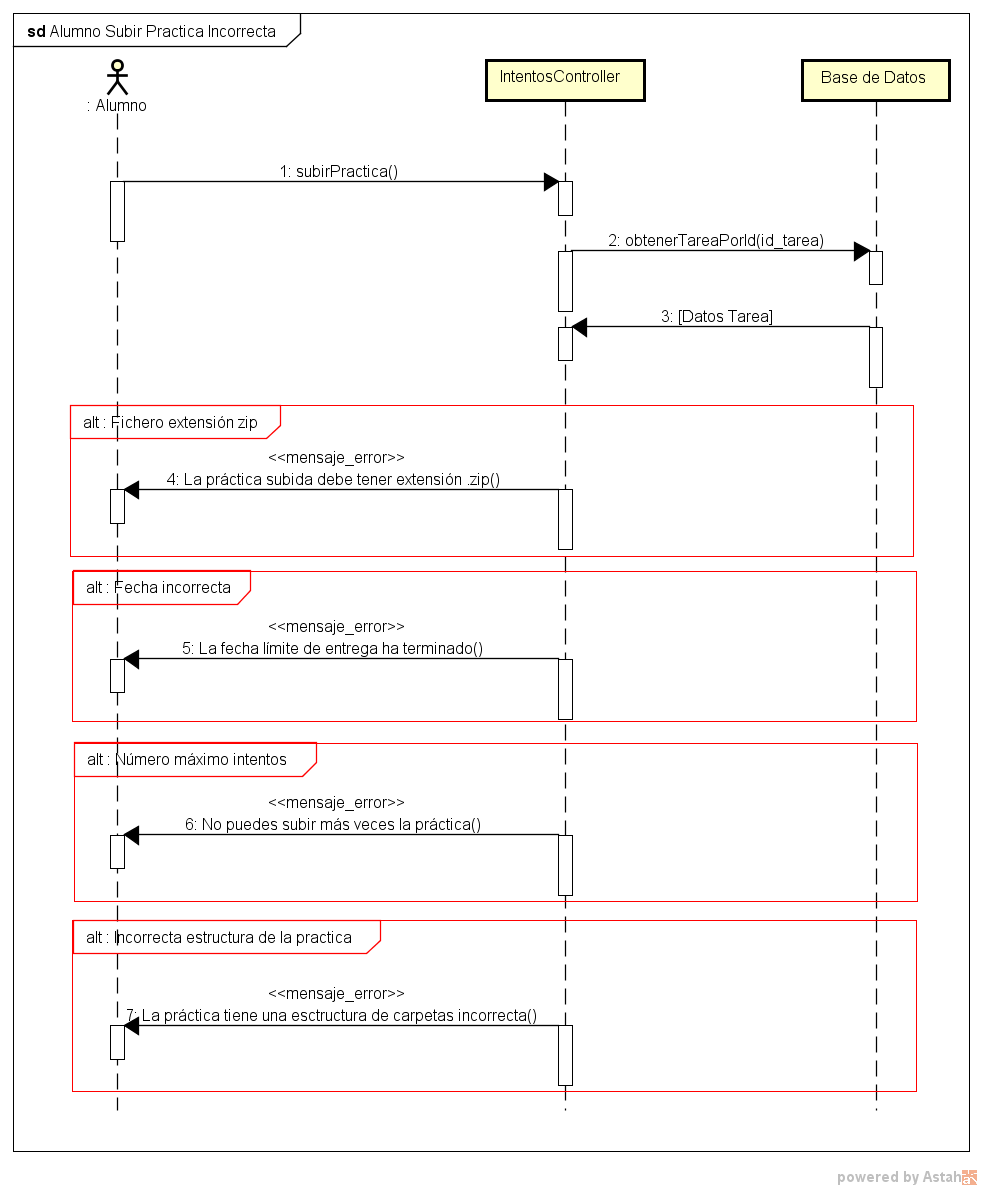
* Diagrama de secuencia: “Profesor sube test”



* Diagrama de secuencia: “Alumno sube práctica”



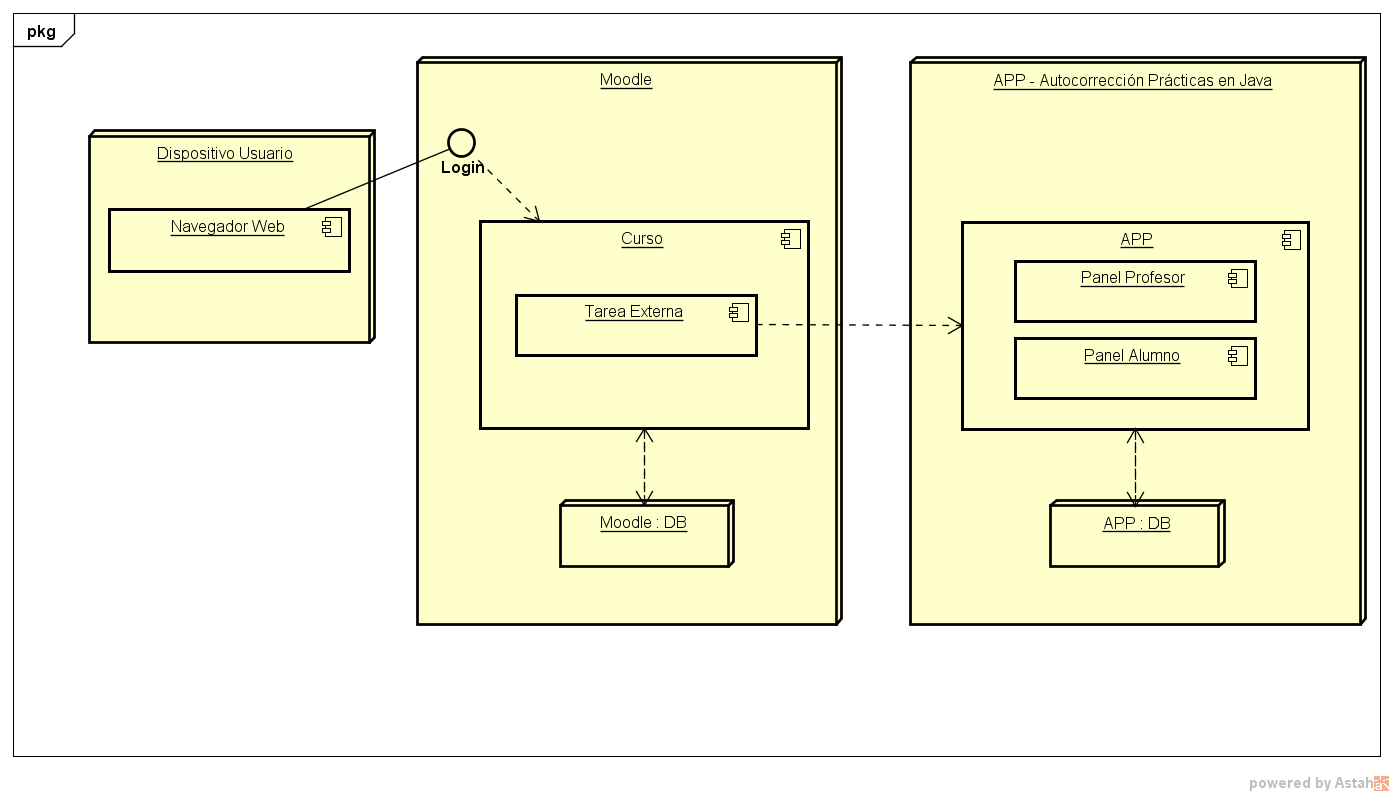
* Diagrama de secuencia: “Alumno sube práctica incorrecta”



## Diseño arquitectónico

### Diagrama de despliegue

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



# 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.

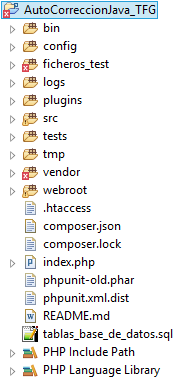
## Estructura de directorios

….

## Estructura del código fuente

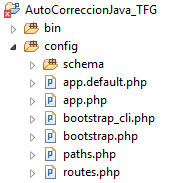
En esta sección se va a explicar brevemente los directorios que componen el código fuente que se ha utilizado para crear la aplicación.

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:



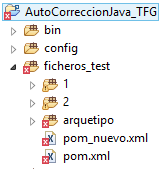
Las carpetas que no han sido utilizadas ni modificadas son: bin, logs, plugins y tmp.

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



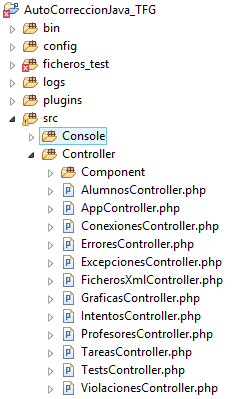
Dentro de este directorio el único fichero modificado ha sido el de “app.php”, el cual ha sido necesario configurar para poder establecer conexión con la base de datos utilizada en la aplicación, además de la base de datos adicional que se ha utilizado para las pruebas unitarias.

* Directorio AutoCorreccionJava\_TFG/**ficheros\_test:**



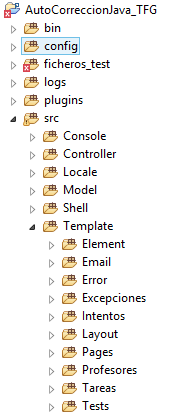
Dentro de esta carpeta van a estar una serie de ficheros que van a usarse para poder realizar algunas pruebas unitarias, ya que hay controladores que tienen métodos que realizan procesos de lectura y escritura sobre ficheros xml.

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



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

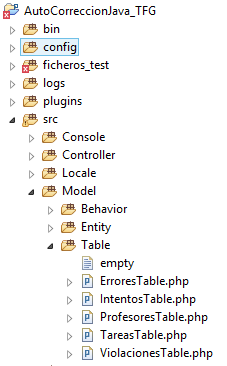
* Directorio AutoCorreccionJava\_TFG/**src/Template**:



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.

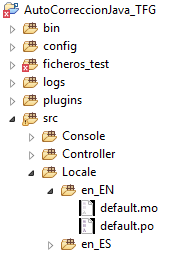
* Directorio AutoCorreccionJava\_TFG/**src/Model**:



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.

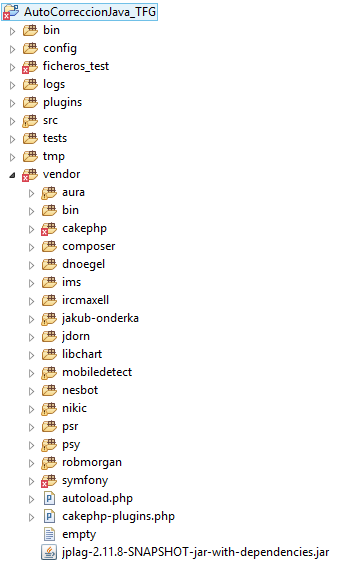
Por ello dentro de esta carpeta la única subcarpeta utilizada es la de “Table”, la cual contiene diferentes ficheros php los cuáles van a encargarse de realizar la validación de los datos introducidos en los campos de los formularios y de establecer relaciones entre las tablas de la base de datos para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

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



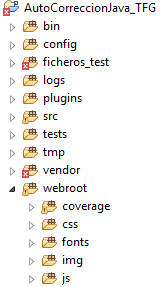
Contiene los ficheros .po que van a permitir realizar la Internacionalización de la aplicación.

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



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:**



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.

## 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:



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



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



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:



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



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”:



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:



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.

### Java JDK y JRE

### 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:



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.

### CakePHP (BORRAR)

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:



### Bootstrap (BORRAR)

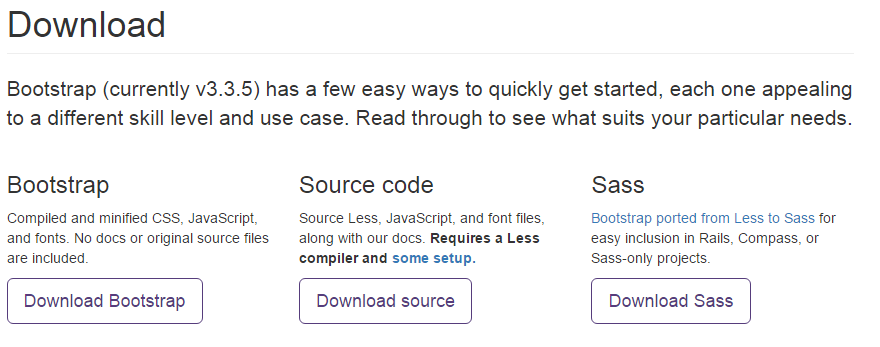
Para facilitar la creación del diseño de nuestras interfaces, se utilizará la librería Bootstrap.

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:



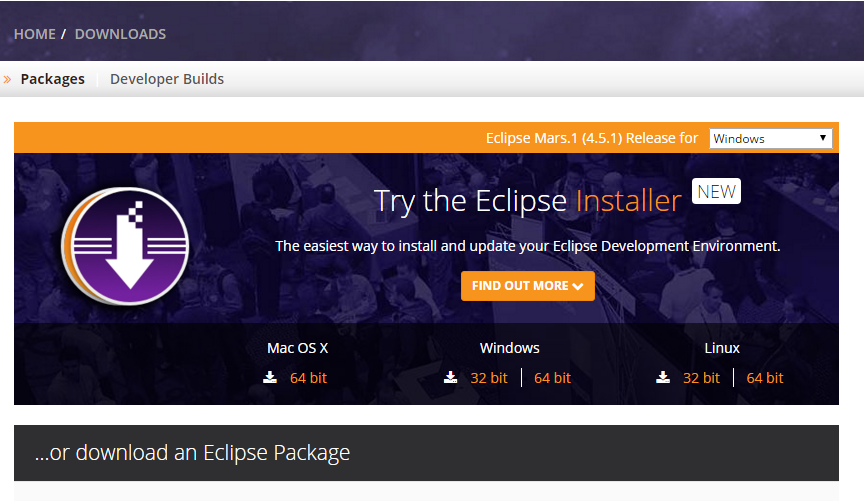
1. Se nos descargará un .zip el cual tendremos que extraer en las correspondientes carpetas (\css y \js) de nuestra carpeta “webroot” creada automáticamente tras crear nuestra aplicación utilizando el framework CakePHP.

### Eclipse for PHP Developers (BORRAR)

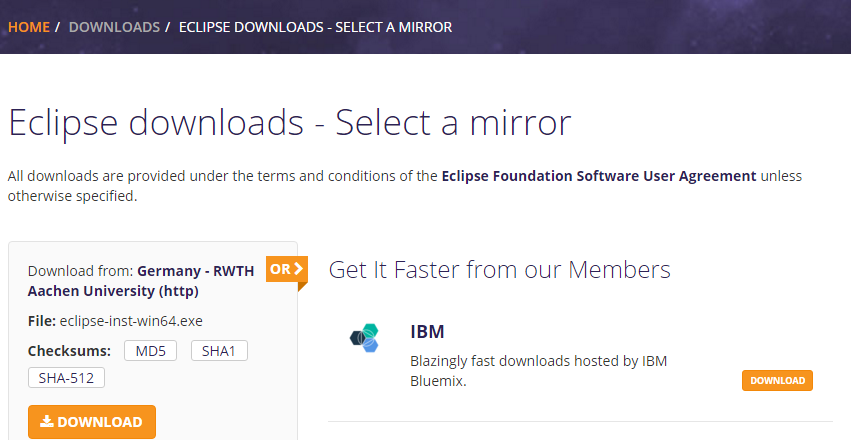
Para editar y realizar todo el código de la aplicación se ha utilizado el IDE Eclipse for PHP Developers, aunque podría haberse utilizado cualquier otro.

Para su instalación simplemente habría que:

1. Acceder a la URL <http://www.eclipse.org/downloads/> y seleccionar la arquitectura que se corresponda con nuestro sistema operativo:



1. Una vez seleccionada la opción correspondiente, nos aparecerá una nueva página en la que ya podremos finalmente descargar el instalador de Ecplise seleccionando el botón “Download” presente en la parte izquierda:



### 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:



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:



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:



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 [5].

### 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 [6].

### Tarea tipo “herramienta externa” en Moodle (ESTO IRIA AQUÍ O EN EL MANUAL DEL DOCENTE)

En este apartado se va explicar cómo se deben de crear en Moodle las tareas de tipo “herramienta externa”, que pertenecerían a un curso de Moodle, las cuales van a ser las que enlacen con la aplicación web. Estas tareas son imprescindibles, ya que son las que van a configurarse con los parámetros LTI necesarios que van a permitir enlazar con la aplicación.

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 [7].

Partiendo de que el curso ya está creado, la apariencia que veríamos al entrar al curso sería la siguiente:



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 nuestro servicio web. 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”.



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.
    - Clave de cliente (consumer key): Deberá ser proporcionada al profesor tras rellenar el formulario de registro de la aplicación web, y por lo tanto deberá ser establecida correctamente por el profesor ya que si no será imposible acceder a la aplicación.
    - Shared secret: Al igual que la clave del cliente, se le habrá sido proporcionada al profesor en su registro y por lo tanto ese secret proporcionado deberá ser el que sea establecido en este campo.
    - Parámetros personalizados: no rellenamos nada.
    - Contenedor de inicio por defecto: opción que viene por defecto.

El aspecto se tendrá que parecer al siguiente:



* + *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.



* 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:



## 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* 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*: que va a apuntar a la aplicación, y la cual va a permitir acceder a ella.
* *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 lanza una excepción, 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

### Ejecutar pruebas

### Pruebas unitarias

### Pruebas de cobertura de código

# Manual del usuario

## Manual del docente

### Registrar docente en el Servicio Web

Este es el primer paso que deberá de realizar el profesor para poder utilizar la aplicación.

Para proceder a este registro los pasos que ha de seguir son los siguientes:

1. Accedemos a Moodle y entramos en el enlace “registrar” y se nos redirigirá a la siguiente URL.

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

1. Completamos los campos del formulario tal y como se nos indican.
2. Una vez rellenado el formulario, pinchamos en “Registrar” y automáticamente se nos mostrarán los parámetros LTI, consumer\_key y secret, necesarios para que el profesor pueda crear una tarea de tipo “herramienta externa” desde Moodle.
3. Al crear esa tarea, el profesor ya podrá acceder correctamente al servicio web.

# Bibliografía

[1] «Bases y Tipos de Cotización 2015», *Crear-Empresas*. [En línea]. Disponible en: http://www.crear-empresas.com/bases-tipos-cotizacion. [Accedido: 05-ene-2016].

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

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

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

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

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

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