# 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.

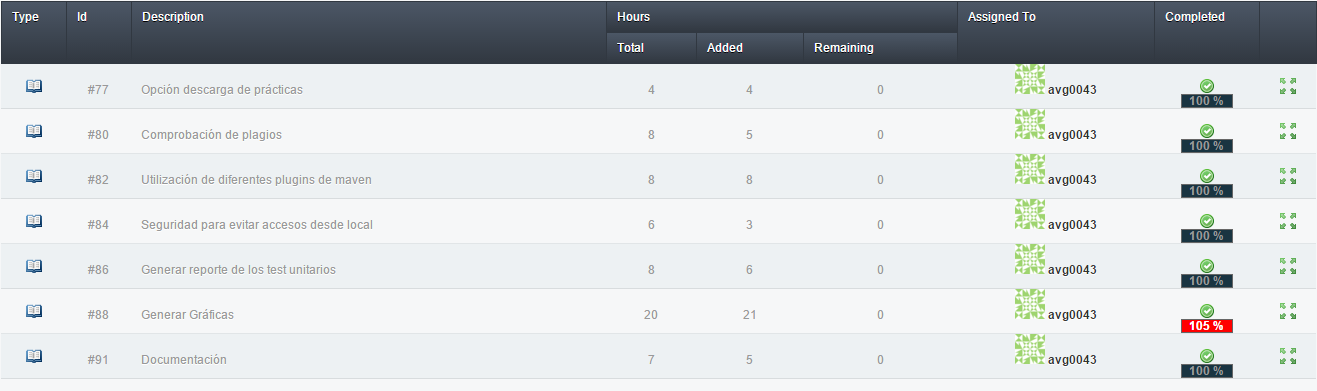




### Iteración 6 (26 Nov 2015 a 10 Dic 2015)

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.





# 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 la actividad:** El profesor creará en Moodle y dentro del curso creado, una actividad de tipo “herramienta externa” que enlazará con el servicio web estableciendo la URL, key y secret correspondiente. Cuando esta actividad sea seleccionada, se redirigirá a dicho servicio web. Tal enlazamiento es permitido gracias al LTI.
* **RF-M02: Configurar la actividad:** El profesor en todo momento podrá configurar o editar la actividad creada: cambiar su nombre, url, ocultarla, ponerla visible…

#### Rol profesor y alumno

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

### Propios del Servicio web

#### Rol profesor

* **Registrarse en el servicio web**: El profesor rellenará el formulario para registrarse y obtendrá los parámetros LTI necesarios (URL, consumer key y secret) para crear en Moodle la tarea de tipo “herramienta externa”.
* **Configurar los parámetros de la tarea**: El profesor configurará los parámetros correspondientes de la tarea para que posteriormente los alumnos puedan subir sus prácticas.
* **Consultar sus datos**: El profesor podrá consultar sus datos (nombre, apellidos, correo…).
* **Ver alumnos registrados**: El profesor podrá ver los alumnos que están registrados en el servicio web.
* **Subir test**: El profesor podrá subir los test que desee desde el formulario de subida de ficheros. Estos test se aplicarán posteriormente a las prácticas subidas por los alumnos.
* **Comprobar plagios**: El profesor podrá comprobar si existen plagios entre las prácticas subidas por los alumnos.
* **Descargar prácticas**: El profesor podrá descargar todas las prácticas subidas por los alumnos.
* **Consultar resultados de las prácticas:** El profesor podrá ver los diferentes intentos de prácticas subidas por los alumnos y visualizar si han pasado los test o no.
* **Visualizar gráficas:** El profesor podrá ver diferentes gráficas para tener una visión global referente a las diferentes prácticas subidas por los alumnos (violaciones de código cometidas, prácticas que pasan los test…).

#### Rol alumno

* **RF-SW05: Subir práctica:** El alumno podrá subir la práctica desde el correspondiente formulario de subida de ficheros para su proceso de corrección.
* **RF-SW06: Borrar la entrega de la práctica:** Si el alumno ha subido en la entrega de la práctica una práctica que no quería subir, podrá reemplazarla por otra.
* **Consultar reportes:** El alumno podrá consultar los reportes que se generan tras el proceso de corrección de la práctica. Los reportes pertenecientes a los plugins JDepend y JavanCSS siempre les podrá ver, mientras que los de los plugins PMD y FindBugs únicamente cuando la práctica subida presente violaciones en el código.
* **Consultar gráficas:** El alumno podrá visualizar las gráficas ofrecidas para observar la evolución que va teniendo tras realizar cada intento de subida de práctica.

### Requisitos de restricción

* **RR-01:** No permitir que prácticas que no compilen puedan ser subidas.
* **RR-02:** Permitir corregir prácticas realizadas en otros lenguajes diferentes a Java.

## 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 del Servicio 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 del Servicio 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”**
  + Rol profesor:



* + 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 la actividad** | |
| **Descripción** | Crear una actividad de tipo “herramienta externa” que al ser clicada, redirija al servicio web. | |
| **Precondición** | Acceder a Moodle con las credenciales 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 credenciales, y entra al curso en el que desea crear la actividad. |
| 2 | Crea una actividad de tipo “herramienta externa” estableciendo la URL, key y secret del servicio web con el que se va a enlazar y guarda los cambios. |
| **Post-condición** | La nueva actividad aparece en el curso para poder ser accedida. | |
| **Excepciones** | Paso | Acción |
| 1 | Si el profesor introduce credenciales incorrectos, no podrá acceder a Moodle. |
| 2 | Si el curso no está creado, no se podrá crear la actividad. |
| 3 | Si introduce alguno de los parámetros (URL, key, secret) incorrectamente, no podrá acceder a la actividad. |
| **Importancia** | Alta. | |
| **Frecuencia** | Baja. | |
| **Urgencia** | Alta. | |

|  |  |  |
| --- | --- | --- |
| **RF-M02** | **Configurar la actividad** | |
| **Descripción** | La actividad creada podrá ser configurada o editada por el profesor. | |
| **Precondición** | Tener creada la actividad. | |
| **Secuencia Normal** | Paso | Acción |
| 1 | El profesor accede a Moodle con sus credenciales y pulsa en “editar” sobre la actividad. |
| **Post-condición** | La actividad queda configurada. | |
| **Excepciones** | Paso | Acción |
| 1 | Si el profesor introduce credenciales incorrectos, no podrá acceder a Moodle. |
| **Importancia** | Alta. | |
| **Frecuencia** | Baja. | |
| **Urgencia** | Alta. | |

|  |  |  |
| --- | --- | --- |
| **RF-M03** | **Seleccionar actividad** | |
| **Descripción** | El usuario, bien sea el profesor o el alumno, clica sobre la actividad de Moodle y es redirigido al servicio web. | |
| **Precondición** | La actividad 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 actividad. |
| **Post-condición** | El usuario accede al servicio web en función de su rol. | |
| **Excepciones** | Paso | Acción |
| 1 | La actividad no está configurada correctamente. |
| **Importancia** | Alta. | |
| **Frecuencia** | Alta. | |
| **Urgencia** | Alta. | |

#### Pertenecientes al Servicio web

|  |  |  |
| --- | --- | --- |
| **RF-SW01** | **Crear la opción de subida de ficheros** | |
| **Descripción** | El profesor creará la opción de subida de ficheros para que los alumnos puedan subir sus prácticas. | |
| **Precondición** | El profesor accede correctamente a la actividad que redirige al servicio web. | |
| **Secuencia Normal** | Paso | Acción |
| 1 | El profesor accede a la actividad. |
| 2 | Crea la opción para subir ficheros. |
| **Post-condición** | La opción para subir ficheros queda creada. | |
| **Excepciones** |  | |
| **Importancia** | Alta. | |
| **Frecuencia** | Media. | |
| **Urgencia** | Alta. | |

|  |  |  |
| --- | --- | --- |
| **RF-SW02** | **Establecer los test para corregir la práctica** | |
| **Descripción** | El profesor establece los test que van a permitir corregir la práctica. | |
| **Precondición** | Los test deben estar creados correctamente. | |
| **Secuencia Normal** | Paso | Acción |
| 1 | El profesor accede a la actividad. |
| 2 | Establece los test. |
| **Post-condición** | Los test quedan establecidos | |
| **Excepciones** |  | |
| **Importancia** | Alta. | |
| **Frecuencia** | Media. | |
| **Urgencia** | Alta. | |

|  |  |  |
| --- | --- | --- |
| **RF-SW03** | **Cambiar los test** | |
| **Descripción** | El profesor podrá cambiar los test siempre que lo desee. | |
| **Precondición** | El nuevo test a establecer debe de estar creado correctamente. | |
| **Secuencia Normal** | Paso | Acción |
| 1 | El profesor accede a la actividad correspondiente y cambia los test que desee. |
| **Post-condición** | Los test quedan cambiados. | |
| **Excepciones** | Paso | Acción |
| 1 | No hay ningún test creado previamente. |
| **Importancia** | Media. | |
| **Frecuencia** | Media. | |
| **Urgencia** | Media. | |

|  |  |  |
| --- | --- | --- |
| **RF-SW04** | **Consultar estadísticas de resultados** | |
| **Descripción** | El profesor puede visualizar los resultados obtenidos tras la corrección. | |
| **Precondición** | Al menos una práctica ha tenido que ser corregida previamente. | |
| **Secuencia Normal** | Paso | Acción |
| 1 | El profesor accede a la actividad. |
| 2 | Selecciona la opción de consultar estadísticas. |
| **Post-condición** | Las estadísticas son mostradas. | |
| **Excepciones** |  | |
| **Importancia** | Media. | |
| **Frecuencia** | Baja. | |
| **Urgencia** | Media. | |

|  |  |  |
| --- | --- | --- |
| **RF-SW05** | **Subir la práctica** | |
| **Descripción** | El alumno deberá poder subir la práctica para el posterior proceso de corrección. | |
| **Precondición** | La actividad debe estar creada correctamente. | |
| **Secuencia Normal** | Paso | Acción |
| 1 | El alumno accede a la actividad y sube la práctica para ser corregida. |
| **Post-condición** | La práctica queda corregida. | |
| **Excepciones** | Paso | Acción |
| 1 | La práctica presenta errores de compilación. |
| **Importancia** | Alta. | |
| **Frecuencia** | Alta. | |
| **Urgencia** | Alta. | |

|  |  |  |
| --- | --- | --- |
| **RF-SW06** | **Borrar la entrega de la práctica** | |
| **Descripción** | El alumno podrá rectificar de una práctica subida y decidir borrarla de la entrega. | |
| **Precondición** | Ha tenido que subir una práctica previamente. | |
| **Secuencia Normal** | Paso | Acción |
| 1 | El alumno ya se encuentra dentro de la actividad y borra la entrega de la práctica que había subido. |
| **Post-condición** | La práctica queda borrada de la entrega. | |
| **Excepciones** |  |  |
| **Importancia** | Media. | |
| **Frecuencia** | Baja. | |
| **Urgencia** | Media. | |

|  |  |  |
| --- | --- | --- |
| **RF-SW07** | **Corregir la práctica** | |
| **Descripción** | La práctica que ha subido el alumno pasará el proceso de corrección. | |
| **Precondición** | Ha tenido que subir la práctica. | |
| **Secuencia Normal** | Paso | Acción |
| 1 | La práctica subida pasa por el proceso de corrección. |
| **Post-condición** | La práctica queda corregida. | |
| **Excepciones** |  |  |
| **Importancia** | Alta. | |
| **Frecuencia** | Alta. | |
| **Urgencia** | Alta. | |

|  |  |  |
| --- | --- | --- |
| **RF-SW08** | **Guardar los resultados** | |
| **Descripción** | Los resultados del proceso de corrección de la práctica son guardados. | |
| **Precondición** | La práctica ha sido corregida. | |
| **Secuencia Normal** | Paso | Acción |
| 1 | Se guardan los resultados de la corrección de la práctica. |
| **Post-condición** | Los resultados quedan guardados. | |
| **Excepciones** |  |  |
| **Importancia** | Alta. | |
| **Frecuencia** | Alta. | |
| **Urgencia** | Alta. | |

|  |  |  |
| --- | --- | --- |
| **RF-SW09** | **Mostrar los resultados** | |
| **Descripción** | Los resultados tras el proceso de corrección de la práctica son mostrados. | |
| **Precondición** | La práctica ha sido corregida y sus resultados han sido guardados previamente. | |
| **Secuencia Normal** | Paso | Acción |
| 1 | Se muestran los resultados obtenidos del proceso de corrección de la práctica. |
| **Post-condición** | Los resultados son mostrados. | |
| **Excepciones** |  |  |
| **Importancia** | Alta. | |
| **Frecuencia** | Alta. | |
| **Urgencia** | Alta. | |

# Especificación de diseño

## Introducción

## 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.

* **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. |
| 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. |
| 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”.
* **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. |
| 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.

* **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. |
| resultado | TINYINT(1) | Indica si el intento realizado ha pasado los test o no. |
| fecha\_intento | DATETIME | Fecha en la que se ha realizado el intento. |

*\* 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”.
* **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.

*\* UNIQUE:* (tarea\_id, nombre)

*\* Foreign key (FK):*

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

# Manual del programador

## 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)

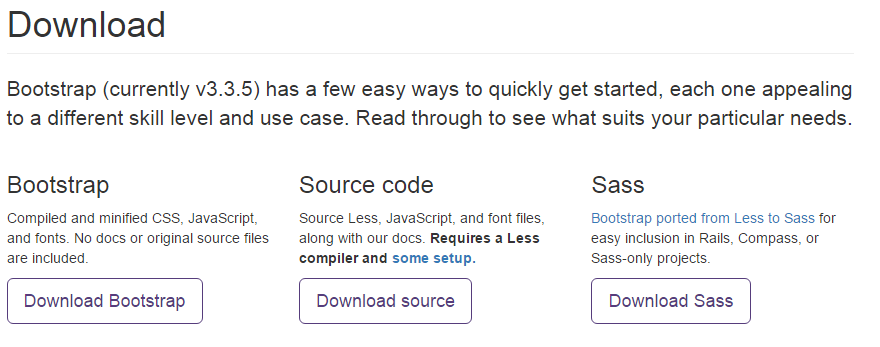
Para facilitar la creación del diseño de nuestras interfaces, se utilizará la librería Bootstrap.

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:



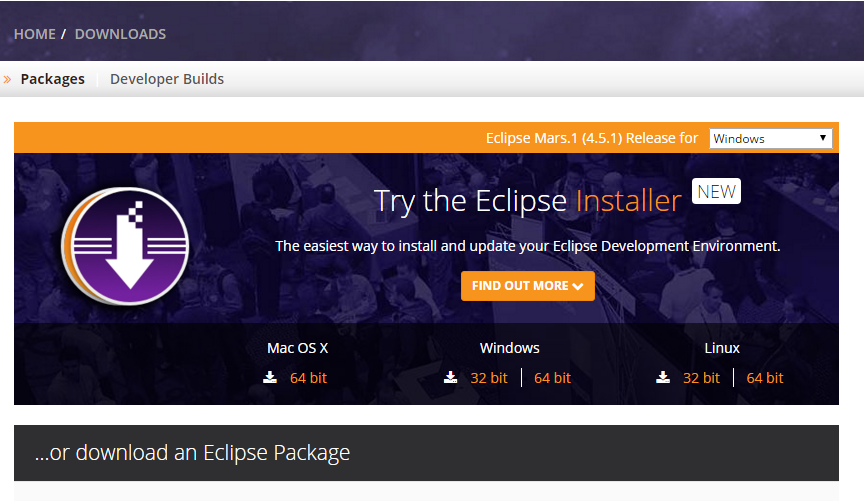
1. Se nos descargará un .zip el cual tendremos que extraer en las correspondientes carpetas (\css y \js) de nuestra carpeta “webroot” creada automáticamente tras crear nuestra aplicación utilizando el framework CakePHP.

### Eclipse for PHP Developers (BORRAR)

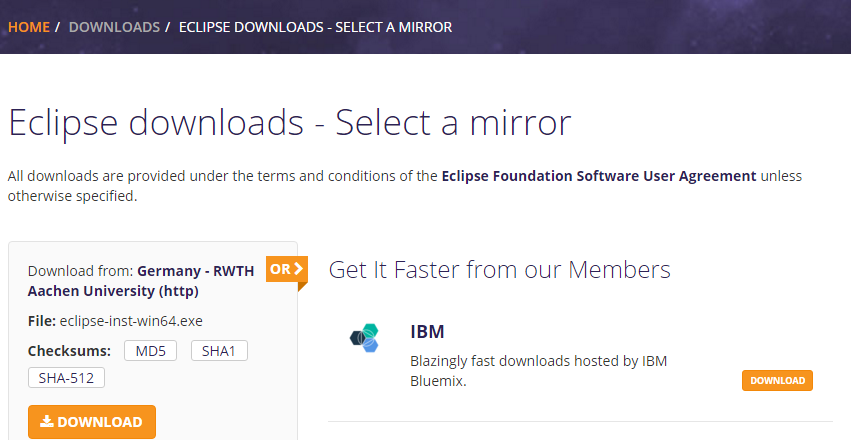
Para editar y realizar todo el código de la aplicación se ha utilizado el IDE Eclipse for PHP Developers, aunque podría haberse utilizado cualquier otro.

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.

### Webmin (BORRAR?????)

La instalación de esta herramienta nos va a permitir la configuración de los sistemas vía web, que en nuestro caso la utilizaremos para crear dos servidores o sitios:

* Uno para Moodle
* Y otro para las pruebas

El proceso para llevar a cabo la instalación va a ser el siguiente:

1. Descargamos el zip que contiene todo el contenido de Webmin, para el sistema operativo Windows, desde el siguiente enlace:

<http://www.webmin.com/download.html>

1. Dicho zip le descomprimimos en el disco C, quedándonos así la siguiente carpeta:

“C:\webmin”

1. Instalamos la última versión de ActivateState Perl (x86) desde el siguiente enlace:

<http://www.activestate.com/activeperl/downloads>

1. Descargamos el programa process.exe desde el siguiente enlace:

<http://retired.beyondlogic.org/solutions/processutil/processutil.htm>

1. Se descargará un zip el cual le extraeremos en el disco C, y de esta forma tendremos la siguiente carpeta:

“C:\process203”

1. Para que sea reconocible, añadiremos dicha ruta a la variable PATH del sistema:



1. Instalamos el Win32:Daemon desde línea de comandos. Para ello abrimos la consola, accedemos al directorio C:\webmin y ponemos el comando “ppm install Win32-Daemon”, y el resultado debería ser el siguiente:



1. Creamos el directorio “C:\temp”.
2. Para acabar con la instalación de Webmin abrimos la consola y accedemos al directorio “C:\webmin” y ejecutamos el comando “perl setup.pl”, momento en el que se nos formularán una serie de preguntas que las contestaremos de la siguiente manera:

* **Config file directory [/etc/webmin]:** C:\webmin\cfg
* **Log file directory [/var/webmin]:** C:\webmin\log

(tras contestar a esta pregunta se nos mostrarán varias veces el mensaje “El sistema no puede encontrar la ruta especificada” pero no hay problema ya que no es ningún tipo de error y la instalación continúa).

* **Web server port (default 10000):** Pulsamos ENTER
* **Login name (default admin):** Pulsamos ENTER
* **Login password:** admin
* **Password again:** admin
* **Start Webmin at boot time (y/n):** y

1. Nos dirá que Webmin ha sido instalado correctamente y además nos proporciona el enlace web con que accederemos a Webmin:



### Hosts Virtuales

Se va a proceder a la creación de dos Hosts Virtuales en Apache:

* Uno para Moodle de nombre “moodle”.
* Otro para las Pruebas de nombre “pruebas”.

Los pasos a seguir son los siguientes:

1. Creamos una carpeta como nivel superior que va a contener las carpetas de los dos hosts virtuales que vamos a crear. Esta carpeta principal recibirá el nombre de “servidores” y se va a encontrar ubicada en la siguiente ruta:

“C:\wamp\www\servidores”

1. Dentro de esta nueva carpeta, creamos dos subcarpetas para cada uno de nuestros hosts virtuales de nombres “moodle” y “pruebas”.
2. Editamos el fichero localizado en “C:\WINDOWS\system32\drivers\etc\hosts” y añadimos abajo del todo dos nuevas líneas que contentan el formato “127.0.0.1 nombreDelHostVirtual” para así registrar nuestros dos nuevos hosts virtuales. Es decir, quedaría de la siguiente forma:



Guardamos dicho fichero “hosts” y lo cerramos.

1. Abrimos el fichero principal de configuración de Apache, “httpd.conf”, localizado en la dirección “C:\wamp\bin\apache\apache2.4.9\conf\httpd.conf”. Una vez abierto procedemos a su edición, y para ello nos dirigimos a la sección “Supplemental configuration” y localizamos la siguiente sección:



1. Eliminamos el comentario # del principio de la segunda línea, quedándonos así la sección de la siguiente forma:



Guardamos dicho fichero y lo cerramos.

1. Abrimos el fichero “httpd-vhosts.conf” que se encuentra en “C:\wamp\bin\apache\apache2.4.9\conf\extra\httpd-vhosts.conf” cuyo aspecto es el siguiente:



1. Nos posicionamos en la línea 16 la cual está vacía, y colocamos el siguiente trozo de código:



Con esto lo que hacemos es dar los correctos permisos a nuestra carpeta principal “servidores” la cual va a contener las subcarpetas de los dos hosts virtuales.

1. A continuación modificaremos el código mostrado en el paso anterior número 6.

La primera etiqueta <VirtualHost> se refiere a nuestro servidor principal, mientras que las siguientes etiquetas <VirtualHost> van a ser para nuestro dos hosts virtuales.

Los únicos valores que tenemos que establecer van a ser el de “DocumentRoot” (el cual se refiere a la ubicación de nuestro host) y el de “ServerName” (que se trata del nombre que vamos a asignar a nuestro host).

Tras la modificación, este sería el resultado:



1. Para finalizar, guardamos y cerramos el fichero anterior “httpd-vhosts.conf” y reiniciamos Apache.

Para acceder a cualquiera de los dos hosts virtuales, lo podremos hacer de una de las siguientes formas (en este caso usando como ejemplo el sitio “pruebas”):

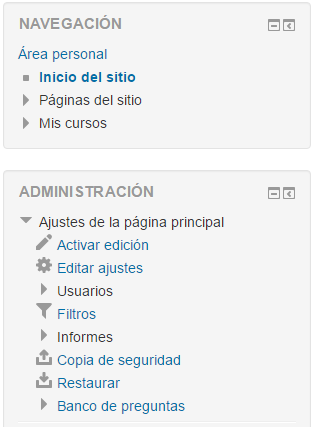
* <http://localhost/servidores/pruebas/>
* <http://pruebas/>

### Curso en Moodle (ESTO IRIA AQUÍ O EN EL MANUAL DEL DOCENTE)

En este apartado se va a explicar cómo va a crearse el curso de Moodle el cual va a contener la actividad de tipo “herramienta externa” que va a enlazar con nuestro servicio web.

Para ello primero se va a explicar cómo crear el curso, y posteriormente cómo se añadirá dicha actividad. Los pasos a seguir en la creación del curso son:

1. El profesor o administrador accede a Moodle con sus credenciales. (SOLO DEJA HACERLO AL ADMINISTRADOR..)
2. En el panel de “Navegación” selecciona en “Inicio del sitio”, y después en el panel de “Administración” selecciona la opción “Activar edición”:



1. Tras esto, aparecerá un nuevo botón “Agregar un nuevo curso” el cual clicaremos y a continuación rellenaremos la información correspondiente a nuestro curso.

El apartado “General” contendrá los siguientes campos:

* **Nombre completo del curso:** el nombre que queramos.
* **Nombre corto del curso:** el nombre que queramos.
* **Categoría del curso:** dejamos la que viene por defecto.
* **Visible:** dejamos lo que viene por defecto.
* **Fecha inicio del curso:** donde seleccionaremos el día actual.
* **Número ID del curso:** estableceremos el ID que queramos para nuestro curso.

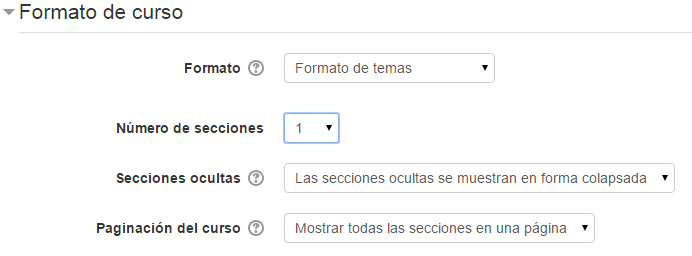
Deberá obtenerse una vista similar a esta:



En el apartado “Formato de curso” tendremos los siguientes campos:

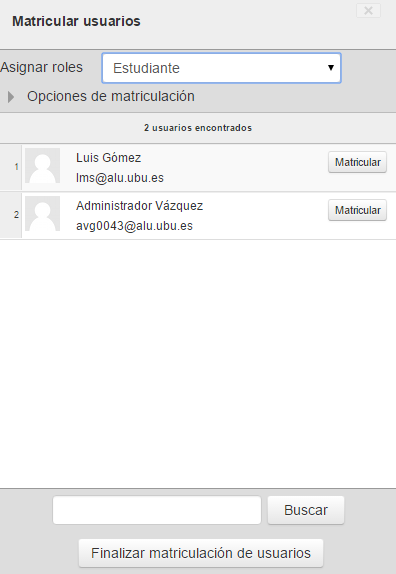
* **Formato:** seleccionaremos la opción “formato de temas”.
* **Número de secciones:** 1.
* **Secciones ocultas:** dejamos la opción que viene por defecto.
* **Paginación:** dejamos la opción que viene por defecto.

Deberemos obtener una vista similar a la siguiente:



El resto de apartados se dejarán tal cual están sin modificar nada. Finalmente pulsamos el botón “Guardar cambios y mostrar”.

1. A continuación seremos redirigidos a una nueva página en la que se procederá a la matriculación de los usuarios que queramos que tengan acceso a nuestro curso.



En primer lugar se seleccionará el rol que queramos asignar al nuevo usuario a matricular desde el desplegable “Asignar roles”, y después seleccionando “Matricular” el usuario quedará matriculado en el curso.

Una vez hayamos matriculado a todos los usuarios que queramos, finalizamos con el botón “Finalizar matriculación de usuarios” y después pinchando en el botón “Continuar al contenido del curso”.

1. La apariencia que tendrá nuestro curso será la siguiente:



1. Una vez creado nuestro curso, procederemos 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. Se nos mostrará un formulario similar al que nos apareció al crear el curso.

En este caso 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 en la que se encuentra nuestro servicio web. De esta forma esta actividad que estamos creando podrá conectarse con dicho servicio web.
    - Clave de cliente: Deberá ser proporcionada por el servicio web, para que así coincida que la que este tiene establecida en su configuración y así poder acceder al servicio web.
    - Shared secret: Deberá ser proporcionada por el servicio web, para que así coincida que la que este tiene establecida en su configuración y así poder acceder al servicio web.
    - 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 nuestro servicio 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 nuestro servicio 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…).

# 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].