# Conceptos Teóricos

## Learning Management System (LMS)

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

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

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

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

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

## Plataforma Moodle

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

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

Dentro de las características generales de Moodle encontramos:

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

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

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

# Técnicas y herramientas

## Técnicas de desarrollo

### LTI

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

Este estándar se trata de un concepto fundamental a la hora de realizar el trabajo fin de grado, ya que es el que va a permitir que nuestra aplicación establezca una conexión con Moodle.

### Modelo-vista-controlador (MVC)

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

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

* **Modelo**: Representa la parte de la aplicación que implementa la lógica de negocio. Es la capa responsable de gestionar los datos: procesamiento, recuperación, validación, asociación…
* **Vista**: La vista hace una representación de los datos del modelo, es decir, se trata de la interfaz de usuario.
* **Controlador**: Se encarga de gestionar las peticiones recibidas por parte de los usuarios, y es la capa encargada de responder la ayuda solicitada apoyándose tanto en el modelo como en la vista.

### Desarrollo ágil de software

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

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

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

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

#### SCRUM

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

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

Dentro de SCRUM existen varios roles:

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

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

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

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

## Herramientas de software

### Lenguajes

#### HTML5

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

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

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

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

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

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

#### PHP

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

Algunas de las características de este lenguaje son las siguientes:

* Es considerado un lenguaje fácil de aprender.
* Tiene capacidad para conectarse con la mayoría de los motores de base de datos.
* Lenguaje libre y abierto.
* Multitud de documentación, foros y ejemplos.
* No requiere definición de tipos de variables, permite técnicas de programación orientada a objetos y manejo de excepciones.

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

#### Bootstrap

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

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

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

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

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

#### XML

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

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

Con una explicación breve, ha sido utilizado en el TFG para las siguientes tareas:

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

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

### Herramientas

#### GitHub

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

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

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

#### XP-DEV

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

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

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

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

#### Moodle

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

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

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

Por lo anteriormente citado y por ser la plataforma utilizada en *UBUVirutal* de la Universidad de Burgos, Moodle ha sido utilizado para poder realizar el TFG.

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

#### WampServer

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

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

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

#### MySQL

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

Algunas de sus características son las siguientes:

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

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

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

#### HeidiSQL

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

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

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

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

#### Astah

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

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

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

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

#### FileSync

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

Ha sido utilizada para mantener actualizado el código fuente que se encuentra alojado en el subdirectorio GitHub del directorio Documentos.

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

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

# Aspectos relevantes del desarrollo

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

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

## Elección de la forma de validar el consumer key

El objetivo principal de las primeras semanas fue intentar conseguir una conexión entre Moodle y el servicio web. Para ello se empleó una librería lti, la cual se descargó y se realizó una primera investigación de la misma.

Dicha librería lo que nos iba a permitir era llevar a cabo una comparación de los consumer\_key y secret, que se establecían en la tarea creada en Moodle, con los valores consumer\_key y secret correctos que se encontraban almacenados en la base de datos. Si dichos valores coincidían, entonces la conexión entre Moodle y el servicio web podría realizarse correctamente, en caso contrario no.

Sin embargo empezaron a surgir problemas en el tema de realizar la comparación del consumer\_key, principalmente debido a que la librería utilizaba métodos de acceso a la base de datos que ya estaban obsoletos y otros trozos de código que no se conseguían entender correctamente.

Por ello, este proceso se realizará de otra forma. Para ver el proceso completo: ENLACE AL APARTADO CORRESPONDIENTE DEL DOCUMENTO ANEXOS.

## Elección del framework CakePHP

Las primeras semanas fueron empleadas para investigar acerca de Moodle, el protocolo OAuth, y sobre todo de la librería LTI que iba a permitir establecer una conexión entre Moodle y la propia aplicación o servicio web.

Tras conseguir realizar esta conexión, es cuando empezó realmente la programación del código durante aproximadamente 2-3 semanas. Durante estas primeras semanas no se habló de utilizar ningún patrón para realizar la aplicación, sin embargo posteriormente en una reunión el tutor propuso que usara el patrón MVC (Modelo-Vista-Controlador) y como tal también me indicó que podría usar algún framework que me facilitara la implementación de dicho patrón y es donde se propuso usar el framework CakePHP.

Debido a esta propuesta, me dediqué durante varios días a investigar, realizar tutoriales y pequeñas aplicaciones que se basaban en el framework.

Finalmente tras haber adquirido un pequeño conocimiento inicial acerca del framework y tras observar que me podría facilitar en numerosos aspectos la implementación del patrón MVC, decidí adaptar todo el código que tenía ya realizado para que utilizara CakePHP.

## Elección de Maven

Dado que la aplicación consistía principalmente en un proceso de autocorrección de las prácticas subidas por los alumnos, a las cuales iban a aplicarse los test subidos por el profesor, era necesaria una herramienta que permitiría gestionar esta implementación. Por ello como opciones posibles había dos: Ant y Maven.

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

También otro factor que me ayudó a decantarme por Maven fue que este es más actual que Ant.

## Elección de Bootstrap

Las primeras semanas se dejó un poco de lado el diseño de la interfaz, centrándonos principalmente en la creación de los controladores, tablas de la base de datos…

En una reunión, el tutor puso como tarea comenzar a realizar pequeños diseños de interfaz. Y es por tanto cuando se planteó la idea de usar Bootstrap, de la cual ya me sonaba de haberlo leído en algunos artículos de internet pero sin embargo que nunca la había utilizado sobre una aplicación.

Tras leer información en la web oficial y realizar pequeños ejemplos, tomé la decisión de utilizar dicha librería ya que me pareció que era relativamente sencilla de utilizar y que me podría ayudar a realizar el diseño de la interfaz de la aplicación.

## Elección de los plugins para los reportes (reducir y mover a anexos, instalación herramientas)

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

Por ello los plugins que se seleccionaron para generar estos reportes fueron los siguientes:

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

## Elección de la forma de añadir los plugins

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

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

1. Abrir el fichero pom.xml con una función propia de php.
2. Crear las etiquetas correspondientes para cada uno de los plugins.
3. Añadir dichas etiquetas al pom.xml y guardarlo.

Por lo tanto si en un futuro se quisiera añadir un nuevo plugin, sería necesario editar el método mencionado.

((los botoness de reportes habría que añadirles))

## Elección de la herramienta para los plagios

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

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

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