# 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.

## Aceptación del proyecto

Como punto de partida, tenía como objetivo realizar un trabajo de fin de grado que no dependiese de otros proyectos o que no consistiera en realizar mejoras sobre trabajos anteriores.

Partiendo de esta idea y sin tener como objetivo ofrecer una idea de trabajo, me reuní con el tutor correspondiente para escucharle en sus propuestas de trabajos de fin de grado. Me propuso dos trabajos fin de grado, pero sin embargo el que más me llamó la atención fue el trabajo actual de autocorrección de prácticas realizadas en el lenguaje Java.

Me explicó un poco por encima los requisitos y el funcionamiento que tendría el trabajo fin de grado. El principal problema o duda que se me planteó fue que el lenguaje que se iba a utilizar era PHP y yo le desconocía por completo.

Finalmente tras varias semanas investigando y realizando tutoriales sobre PHP, y también sobre Moodle, acepté la propuesta de dicho trabajo fin de grado.

## 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, y tras hacerle la correspondiente propuesta al profesor, se planteó que el proceso de comparar el consumer\_key establecido en la tarea de Moodle con el consumer\_key correcto que se encontraba en la base de datos, se hiciera previamente antes de hacer la llamada a dicha librería, y que esta únicamente se encargara de comprobar el secret correcto y de proporcionarnos el objeto LTI con la información del usuario que accedía a Moodle. Tras realizar dicha implementación, se consiguió realizar la conexión.

## 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.