# Técnicas y herramientas

## Introducción

En este apartado se va a hablar de las diferentes herramientas y técnicas que he analizado y usado, mostrando las razones que me han influenciado a tomar la decisión de utilizarlo para la realización del proyecto. Debido a la gran cantidad de herramientas y técnicas que existen para resolver los problemas de la aplicación, he ido escogiendo varias de las herramientas disponibles y he cogido la que mejor se ajustaba.

En algún caso las decisiones han sido claras, pero en otros no he sabido muy bien cual escoger, por lo que me he decantado por lo que más conocía, y otras veces la decisión la he tomado por la sencillez de la técnica o herramienta.

## Framework

**CakePHP**

CakePHP hace que la construcción de páginas web sea más sencilla, rápida y que requiera menos código. Ofrece un acceso más flexible a las bases de datos y permite la construcción de sistemas más complejos.

Ventajas:

* Contiene una amplia documentación que cuenta con libros, API y Videos. Esta documentación está en varios idiomas, pero no está traducida completamente al español.
* El proceso de instalación es relativamente sencillo, ya sea, a través de Composer u Oven.

Desventajas

* Cake es muy pesado debido a que tiene muchas opciones activas, lo que hace que se pierda velocidad.

**Laravel**

Es un framework para desarrollar aplicaciones y páginas web con PHP 5 y PHP 7. Busca desarrollar código de forma elegante y simple, permitiendo muchas funcionalidades. Agrupa lo mejor de otros frameworks y usa las últimas características que se han incorporado a PHP.

Ventajas:

* Posee una gran documentación, con una guía de instalación bastante sencilla y detallada.
* Dentro de esta documentación, existe la posibilidad de utilizar un sistema de plantillas llamado Blade, lo cual mejora mucho el aspecto visual de la aplicación.

Desventajas

* Fuerte orientación hacia la programación estática (usa una abusiva cantidad de clases y métodos estáticos)

**Symfony**

Es un famework que sirve para la construcción de aplicaciones, que simplifica el desarrollo ya que automatiza los patrones que se utilizan para solucionar las tareas más comunes. Se crea código más legible y con mayor facilidad de mantenimiento.

Ventajas:

* Contiene una documentación extensa con los problemas y cuestiones más frecuentes
* La instalación se realiza a través de Composer, y también es bastante sencilla.
* Facilita el desarrollo de código legible.

Desventajas

* Si se busca un tutorial, con los creadores de Symfony, puede llegar a tener un coste de +1000€.
* Algo complejo para programadores inexpertos.
* Consultas complejas a la base de datos.

En mi caso, he decido quedarme con Laravel, puesto que he encontrado mucha más información y tutoriales más sencillos y concisos, que en los otros dos frameworks.

Otro factor que me ha ayudado a la elección ha sido, el sistema de plantillas Blade, ya que considero muy importante el aspecto visual de la aplicación y esta herramienta me va a ayudar a poder crear un diseño elegante y atractivo para el usuario final.