**CAPÍTULO 2**

# **TECNOLOGÍAS WEB EMPLEADAS**

Como bien se ha planteado en los objetivos, queremos que esta aplicación se trate de una aplicación multiplataforma. Debido a esto creo que la opción más conveniente para poder lograr este objetivo debe ser una aplicación web responsive para que si sea accesible para el máximo número de usuarios. Existen varios frameworks para el desarrollo web. Por una parte, tenemos el desarrollo en PHP, Python con frameworks como Django y C# que es la herramienta que he decidido utilizar.

La decisión de la tecnología es una de las decisiones más importantes a la hora de realizar un proyecto y por eso es necesario realizar un análisis previo de las opciones que tenemos a nuestra disposición, valorando las ventajas y las desventajas que esa tecnología nos pueda ocasionar.

|  |  |  |
| --- | --- | --- |
| Lenguaje | Pros | Contras |
| PHP | * Numerosos frameworks. * Una comunidad muy grande. * Compatible con prácticamente todos los hostings. | * Lento en comparación con el resto. * Más código para realizar la misma tarea que los demás. |
| Python | * Muy rápido. * Fácil de desarrollar. * Gestión interna de paquetes. | * Pocos hostings compatibles. * Necesidad de un framework para desarrollo web. |
| C# | * Mejor procesamiento paralelo * Totalmente orientado a objetos * Uso de Access incluido para el acceso a las bases de datos. | * Más lento que PHP * Compilación necesaria en el lado del servidor |

Tabla 2. Comparativa de lenguajes para desarrollo Web

Mi elección final para este proyecto ha sido C# por su facilidad a la hora de la generación de las clases, el tratamiento de los elementos de la base de datos y la integración de conexiones con los servidores de Microsoft. A parte, gracias al uso de Microsoft Visual Studio puedo emplear una estructura Modelo-Vista-Controlador de una manera mucho más sencilla y natural en el proyecto gracias a esta herramienta.

También hay que resaltar que el hecho de que C# se encuentre compilado en el lado del servidor, pese a ser una de las desventajas que tiene, ya que ralentiza el trabajo, supone un aumento de la seguridad, ya que lo que se almacena en el servidor es el ejecutable del proyecto.

## **2.1 HERRAMIENTAS CLOUD DE MICROSOFT (AZURE)**

De entre las herramientas que se van a utilizar para este proyecto hay que resaltar la herramienta de Azure. Obviamente, el hecho de desarrollar una aplicación web implica que debemos de tener un sitio en el cual alojar tanto la base de datos como la web en sí. En este caso al trabajar con C# y herramientas de Microsoft debemos tener un servidor con un Sistema Operativo Windows para que se pueda compilar y ejecutar correctamente en este.

Para obtener acceso a los servicios de Azure es necesario crearse una cuenta en el portal de Azure y dentro de este ya podremos crear los recursos necesarios para poder trabajar.



Ilustración 1. Captura del panel de Azure.

Aquí podemos ver una captura de la pantalla que nos encontramos en el panel de Azure. Como se puede observar tenemos varios recursos ya creados, el primero se trata del recurso donde se almacenará la base de datos de SQL con el nombre de *TFGDatabase*. El segundo recurso que observamos es el servidor de SQL, *wecalendar.* En el tercer recurso almacenaremos el Back-End de la web, *WCBackend,* y en el último se encontrará el Front-End, *wecalendar*.

El hecho de trabajar con la herramienta Visual Studio implica que este servicio de Azure ya se encuentra integrado dentro de esta, permitiéndome hacer subidas al servidor directamente desde esta aplicación seleccionando la dirección que desee.

Durante la mayor parte del desarrollo de la aplicación todas las pruebas serán ejecutadas en local, dándole uso a Azure para almacenar la base de datos y poder realizar los accesos. Una vez finalizada la fase de producción tanto el Back-End como el Front-End serán subidas al servidor de Azure.