

**Balloon Simulator**

ies comercio | Tutor: Raúl Bonachía Castillo

TFG Desarrollo de aplicaciones web

Álvaro Carbajo Alcalde

**2022**

índice

Introducción

**Globos Arcoíris** es una empresa riojana, localizada en Cuzcurrita de Río Tirón, del sector turístico dedicada a actividades con **globos aerostáticos**. Este proyecto busca realizar una aplicación web para dicha empresa en la cual se puedan simular los vuelos que se desarrollan por La Rioja Alta.



El objetivo del proyecto será diseñar e implementar una **aplicación web** en la que se puedan simular de la forma más realista posible vuelos en globo aerostático. La aplicación contará con varias secciones para la visualización de los datos que se recojan durante la simulación, rutas del trayecto recorrido…

La web tendrá también un apartado específico que permitirá realizar **consultas meteorológicas** que ayudan a los pilotos con la planificación de los vuelos y les permite seleccionar el punto despegue idóneo para un trayecto agradable y disfrutable.

Objetivos

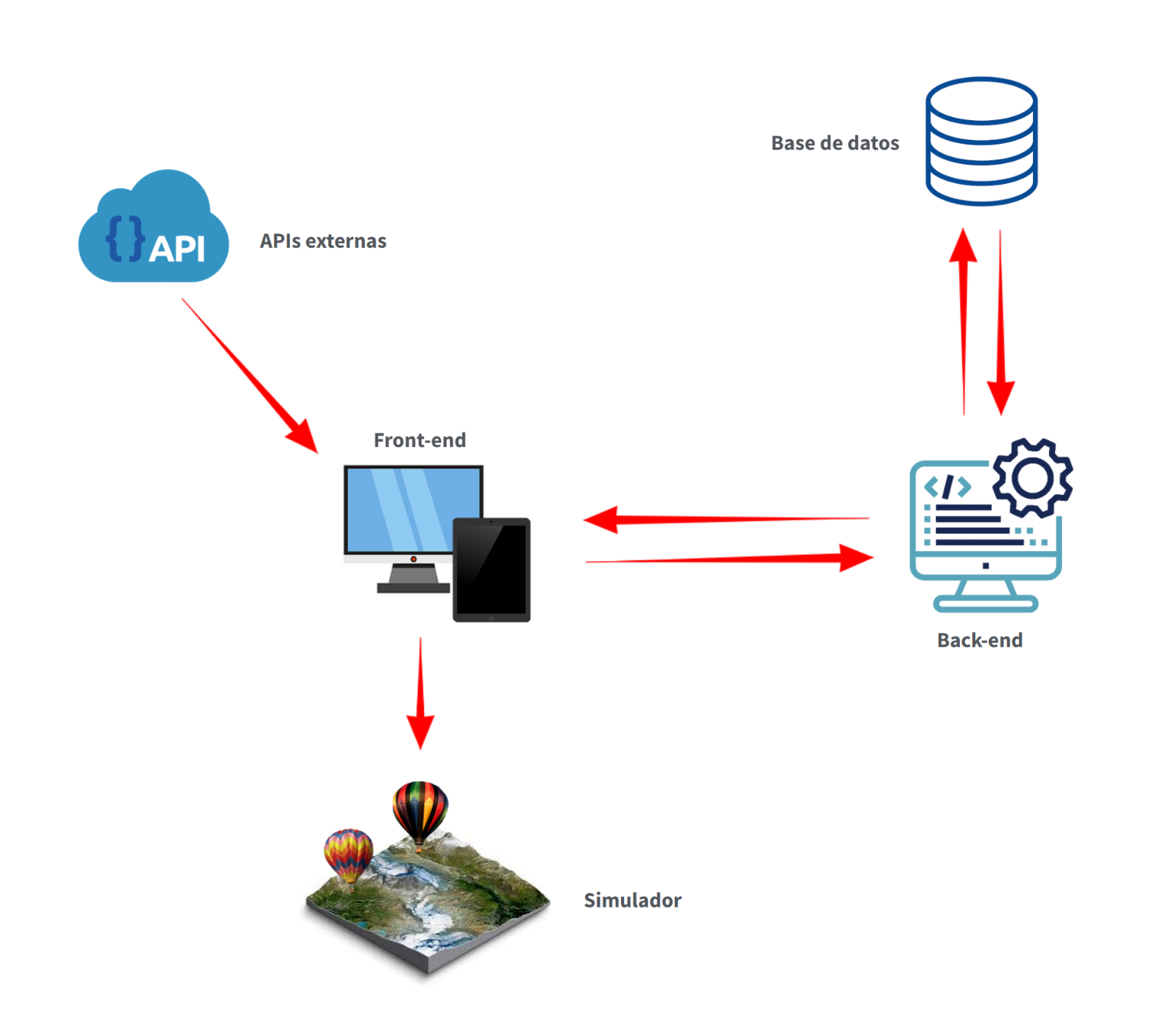
Estos son los objetivos que me he propuesto para la realización de este proyecto:

* Crear un simulador con un mapa de la Rioja Alta, con unas físicas realistas y controles basados en el manejo real de globo aerostático.
* Desarrollar un front-end que aloje el simulador y varias páginas con utilidades.
* Diseñar e implementar una base de datos que permita guardar datos de vuelos, puntos de despegue…
* Servirse de alguna API de terceros para poder visualizar en tiempo real la ruta seguida por el aerostato y usarla además para elegir punto de despegue.
* Listar los vuelos efectuados en un simulador y extraer datos de los mismos como su duración, velocidades medias…
* Mostrar las rutas realizadas en simulaciones anteriores en un mapa interactivo.
* Desarrollar un back-end que permita gestionar CRUDs de las tablas de la base de datos para insertar, visualizar o modificar la información.
* Implementar una API que permita al front-end comunicarse con el back-end mediante peticiones.

Diseño

La web estará formada por **cuatro partes** que funcionarán en conjunto para cumplir con los objetivos especificados. Estas partes son: el simulador, un front-end, un back-end y un sistema **gestor de base de datos**.

El **back-end** contendrá dos partes principales, un API para comunicarse con el front-end y una serie de vistas que permitirán a un usuario administrador gestionar la base de datos. Sera el encargado de intercambiar información con la base de datos.

 El **front-end** se servirá del back-end y de APIs externas para obtener los datos necesarios para el funcionamiento de la web. Esta parte estará dirigida al usuario final y será la encargada de lanzar el **simulador**.

base de datos



Para la gestión de datos utilizaré **MariaDB**. Una base datos relacional de código abierto alternativa a MySQL que nace tras la compra de este por parte de Oracle. He escogido MariaBD sobre **otras opciones** como MySQL o SQLServer porque es software libre y tiene un mayor rendimiento que MySQL.

Back-end



En el back-end estará desarrollado en **Laravel**, un framework de **PHP** muy potente y muy orientado desarrollo servidor. Utiliza como lenguajes HTML, PHP y CSS. También usa **Blade**, un motor de plantillas simple que otorga muchas facilidades para desarrollar vistas.

 Desarrollare un API que permita al front-end comunicarse con los datos y una serie de vistas orientadas a un administrador que realice la gestión de los datos. Me serviré de la estructura **MVC** que aporta Laravel para organizar el código de forma correcta y escalable. Para las vistas usaré Blade, HTML y **Bootstrap**. Bootstrap es un framework de CSS que ofrece un conjunto de herramientas de código abierto para el diseño de sitios web. He elegido Bootstrap porque puedes instalarlo por defecto en Laravel y ahorra mucho tiempo de maquetación de estilos.

He rechazado **otras opciones** como Symfony porque Laravel tiene una curva de aprendizaje más sencilla y parece tener un mejor rendimiento. Laravel cuenta además con una documentación más concisa y permite separar con facilidad la parte de API con la parte web.

front-end

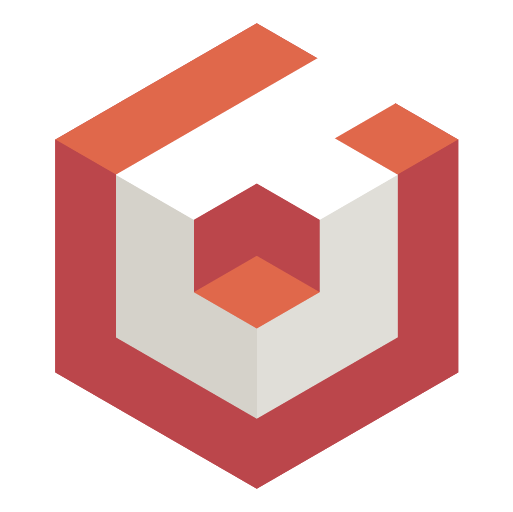


En el front-end **Angular** era la opción más clara, es un framework de JavaScript desarrollado por Google en 2010. Es uno de los frameworks front-end mas populares junto con React, sus grandes ventajas son la flexibilidad, la integración de gran cantidad de librerías de terceros y el uso de TypeScript.

**TypeScript** es un lenguaje de programación libre y de código abierto desarrollado por Microsoft, es un superconjunto de JavaScript, que esencialmente añade un tipado de clases. Es un lenguaje que al contrario que JavaScript está muy orientado a objetos, su desventaja es que requiere una compilación a JS previa a su ejecución.

Para las vistas no utilizaré ningún framework de CSS/HTML porque quiero diseñar las páginas de forma muy concreta y personalizada para el usuario final de la web.

Simulador



Para el simulador he escogido un framework llamado **Babylon.js**, es un motor 3D desarrollado por Microsoft escrito en **JavaScript**. Se basa en una escena que se renderiza embebida en un elemento canvas del HTML. Permite cargar modelos en 2 y 3 dimensiones, crear puntos de luz, cámaras, físicas…

He escogido a Babylon.js por encima de **otras opciones** comoThree.js o Pixi.js porque cuenta con una documentación mucho más extensa con ejemplos ejecutables y soporte de Microsoft.

Servicios de terceros

El front-end se nutrirá de una serie de **APIs** de terceros para extraer distintos datos. Para la creación del mapeado del simulador obtendré una serie de imágenes extraídas de **MapQuest**. MapQuest es un servicio de cartografía web.



Para obtener datos de altura de distintos puntos del mapeado realizare consultas a **OpenTopography**, una API que devuelve información geográfica como la altitud de unas coordenadas dadas.



Los mapas que mostrarán tanto la posición en tiempo real del globo como los puntos de despegue y las trayectorias usaré **Leaflet** una librería de JavaScript de código abierto que provee de herramientas útiles para crear mapas interactivos como creación de marcadores, rutas, etc.



Leaflet a su vez utiliza una API llamada **OpenStreetMap**, un proyecto colaborativo para crear mapas editables y libres. Tiene una base de datos muy amplia de carreteras, información geográfica, relieve…

Bibliografía

Las páginas y documentación usada durante la realización del proyecto:

Angular: <https://angular.io>

Laravel: <https://laravel.com>

Bootstrap: <https://getbootstrap.com>

Babylon.js: <https://www.babylonjs.com>

Leaflet: <https://leafletjs.com>

OpenStreetMap: <https://www.openstreetmap.org>

MapQuest: <https://developer.mapquest.com>

OpenTopography: <https://opentopography.org>

Stack Overflow: <https://stackoverflow.com>

Mockflow: <https://www.mockflow.com>

GitHub: <https://github.com>

GitHub Desktop: <https://desktop.github.com>