

MEMORIA PROYECTO FINAL



RUMBO BICI

Choose your course

Aplicación web que ofrece a los usuarios una plataforma digital donde puedan registrar y organizar sus rutas en bici.

Rodrigo Martínez Delgado

CIFP Juan de Colonia

CFGS Desarrollo de Aplicaciones Web

13/03/2025

Contenido

| | |
|--|----|
| 1. Introducción | 1 |
| 1.1. Descripción | 1 |
| 1.2. Justificación | 3 |
| 2. Planificación | 3 |
| 2.1. Requisitos | 3 |
| 2.1.1. Requisitos funcionales | 3 |
| 2.1.2. Requisitos no funcionales | 4 |
| 2.2. Recursos | 4 |
| 2.2.1. Recursos hardware | 4 |
| 2.2.2. Recursos software | 5 |
| 2.3. Planificación temporal | 5 |
| 2.4. Planificación económica | 6 |
| 3. Tecnologías | 6 |
| 4. Desarrollo y secuenciación temporal | 9 |
| 4.1. Diseño | 9 |
| 4.1.1. Diagrama de aplicación | 9 |
| 4.1.2. Diagrama E/R y esquema BBDD | 10 |
| 4.1.3. Diseño de interfaces | 10 |
| 4.2. Pruebas | 12 |
| 5. Conclusiones | 12 |
| 5.1. Grado de cumplimiento de los requisitos fijados | 12 |
| 5.2. Propuestas de mejora o ampliaciones futuras | 13 |
| 6. Guías | 13 |
| 7. Referencias | 13 |

Índice de figuras

Índice de tablas

1. Introducción

1.1. Descripción

El proyecto consiste en el desarrollo de una aplicación web full stack cuyo propósito es registrar y gestionar las actividades realizadas en bicicleta, fomentando al mismo tiempo su uso como medio de ocio saludable y sostenible.

A través de la plataforma, los usuarios pueden visualizar y clasificar las rutas realizadas, agrupándolas por categorías predefinidas.

La aplicación facilita un seguimiento eficiente del historial de rutas, permitiendo almacenar, consultar, editar y eliminar la información relacionada con cada actividad.

Para su implementación, se ha optado por una arquitectura basada en el patrón Modelo-Vista-Controlador (MVC), favoreciendo una clara separación de responsabilidades.

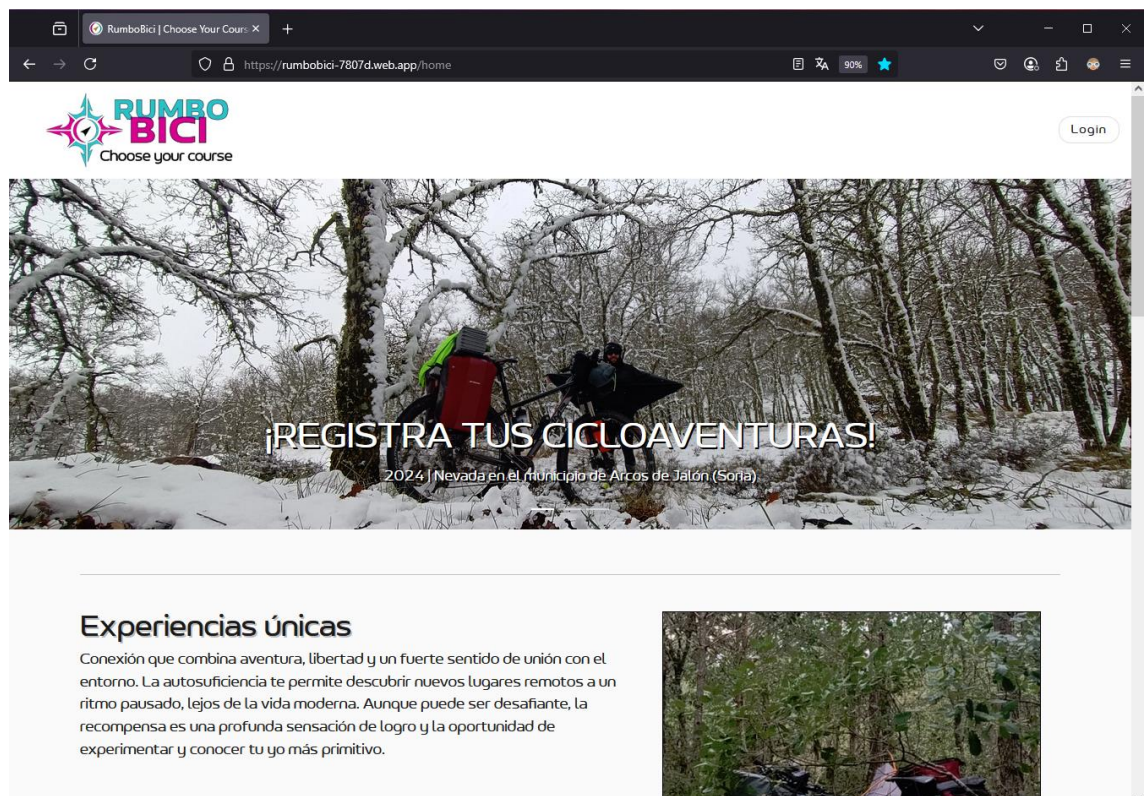


Ilustración 1-1. Pantalla inicial de la aplicación (Desktop).



Ilustración 1-2. Pantalla inicial de la aplicación (Mobile).

1.2. Justificación

Buscando promover estilos de vida saludables, sostenibles y conectados con el entorno, el uso de la bicicleta ha cobrado una relevancia especial, no solo como medio de transporte, sino también como una herramienta de descubrimiento personal y social.

La práctica del ciclismo permite conocer nuevos lugares, compartir experiencias con otras personas y reconectar con uno mismo, convirtiéndose en una forma accesible y enriquecedora de ocio activo.

La aplicación nace con el objetivo de apoyar y potenciar este estilo de vida, ofreciendo a los usuarios una plataforma digital donde puedan registrar y organizar sus rutas.

Al permitir clasificar las actividades por categorías, registrar fechas y acceder al historial completo de recorridos, la herramienta no solo cumple una función práctica, sino que también motiva a seguir explorando, superándose y construyendo una comunidad en torno a la bicicleta.

2. Planificación

2.1. Requisitos

2.1.1. Requisitos funcionales

- Perfil de usuario:
 - Acceso al perfil: Los usuarios podrán acceder a su perfil donde se muestra la información de las rutas que han creado.
 - Visualización de categorías disponibles: Podrán ver las categorías disponibles en la plataforma y asociar sus rutas a una de ellas.
 - Historial de rutas: Los usuarios podrán ver un historial completo de todas las rutas que han creado.
- Gestión de rutas:
 - Creación de rutas: Los usuarios podrán crear nuevas rutas, proporcionando información como el lugar, fecha, distancia y categoría.
 - Edición de rutas: Se podrá editar las rutas previamente creadas.

- Eliminación de rutas: Los usuarios podrán borrar rutas que ya no deseen mantener en su historial.
- Visualización de rutas creadas: Se podrán consultar todas las rutas que se han creado, tanto como por categoría como un listado general.

2.1.2. Requisitos no funcionales

- Usabilidad:
 - Interfaz intuitiva y fácil de navegar.
 - Diseño responsivo compatible con dispositivos móviles y tablets.
- Rendimiento:
 - Tiempo de carga optimizado en condiciones normales de red.
 - Capacidad de soportar múltiples usuarios sin afectar al rendimiento.
- Seguridad:
 - Protección de datos mediante cifrado SSL.
 - Uso de Spring JPA, este utiliza JPQL (Java Persistence Query Language) con consultas parametrizadas internamente, que impiden que las entradas del usuario sean interpretadas como código SQL.
- Escalabilidad:
 - Arquitectura preparada para ampliar funcionalidades.
 - Base de datos optimizada para manejar grandes volúmenes de datos.

2.2. Recursos

2.2.1. Recursos hardware

- Desarrollo:
 - Portátil Asus GL553VD-DM078T.
 - Monitor BENQ GL2460.
 - Teclado Keychron K10P-H3.
 - Ratón Razer Deathadder v2 X.
- Servidores en Producción:
 - 10Gb de almacenamiento (Firebase).
 - Google Cloud (Firebase).
 - 512 MB RAM / 2 vCPU por servicio (Railway).

2.2.2. Recursos software

- Diseño gráfico:
 - GIMP v2.10.38.
 - Inkscape v1.4.
- Entorno de desarrollo local:
 - Tomcat.
 - NG Serve.
 - Brave.
 - Mozilla Firefox.
 - Postman.
- IDE y Editores de código:
 - Visual Studio Code.
 - Notepad++.
- Control de versiones:
 - Git & GitHub.
- Base de datos:
 - MySQL Workbench.

2.3. Planificación temporal

| Descripción de la tarea | Nº de horas |
|---|-------------|
| Planificación, estrategia y documentación | 10 |
| Creación de identidad visual | 10 |
| Diseño UI/UX | 10 |
| Desarrollo frontend | 30 |
| Desarrollo backend | 30 |
| Integraciones y pruebas | 5 |
| Despliegue y comprobaciones | 5 |
| TOTAL HORAS | 100 |

Tabla 2-1. Planificación temporal del proyecto.

2.4. Planificación económica

Teniendo en cuenta que el trabajo será realizado por un trabajador regulado por el XVIII Convenio Colectivo Estatal de Empresas de Consultoría, Tecnologías de la Información y Estudios de Mercado y de la Opinión Pública, en el Área 3 (Desarrollo de Software, Programación y Explotación de Sistemas), Grupo D, Nivel 3.

Salario Anual Base: Según las tablas salariales del convenio, el salario anual para el Grupo D, Nivel 3, es de 17.309,88 euros.

Horas Anuales de Trabajo: El convenio establece una jornada anual de 1.800 horas. $17.309,88 / 1.800 \text{ h} \approx 9,62 \text{ euros/hora}$

| Descripción de la tarea | Precio (€) |
|---|------------|
| Planificación, estrategia y documentación | 96,20 |
| Creación de identidad visual | 96,20 |
| Diseño UI/UX | 96,20 |
| Desarrollo frontend | 288,60 |
| Desarrollo backend | 288,60 |
| Integraciones y pruebas | 48,10 |
| Despliegue y comprobaciones | 48,10 |
| TOTAL (€) | 962 |

Tabla 2-2. Presupuesto del proyecto.

3. Tecnologías

Tecnologías de desarrollo

- Identidad visual:
 - SVG.
 - webp.
 - XCF.
- Frontend:
 - Bootstrap 5.
 - Angular v19.
 - HTML5.
 - SCSS.
 - TypeScript.
- Backend:
 - Spring Boot v3.4.4.
 - Java v21.0.6.
 - Spring Web.
 - Spring Data JPA.

- MySQL Driver.
- Base de datos:
 - MySQL.
- Despliegue:
 - Railway - Backend
 - Docker
 - Linux
 - Github.
 - Firebase - Frontend
 - HTTPS + SSL automático
 - **Firebase CLI**
 - Angular SPA (Single Page App).

La aplicación es una solución full stack basada en **Angular** para el frontend y **Spring Boot** con MySQL para el backend.

Angular, desarrollado en TypeScript, se encarga de la interfaz de usuario e interacción con el usuario. Se comunica con el backend mediante peticiones HTTPS a una API Rest (Application Programming Interface - Representational State Transfer).

Spring Boot gestiona la lógica del negocio y expone los endpoints necesarios para que el frontend pueda acceder o enviar datos.

Toda la información se almacena y gestiona en una base de datos MySQL, a la que el backend accede utilizando JPA (Java Persistence API). Esta arquitectura permite una clara separación entre la interfaz y el servidor, facilitando el mantenimiento y la escalabilidad de la aplicación.

A continuación, una breve descripción de los lenguajes y marcos de trabajo utilizados:

Angular v19.

Angular es un framework de desarrollo frontend basado en TypeScript que permite crear aplicaciones web dinámicas y escalables.

Angular 19 continúa mejorando el rendimiento y la experiencia del desarrollador, incorporando nuevas optimizaciones, mejoras en el renderizado y herramientas más intuitivas para el desarrollo moderno. El enfoque modular y basado en componentes facilita la creación de aplicaciones web mantenibles y eficientes.

[Angular Oficial](#)

HTML 5 (HyperText Markup Language 5).

Última versión del lenguaje de marcado utilizado para estructurar contenido en la web. Ofrece nuevas características, como etiquetas semánticas (ej. <header>, <footer>, <article>) que **mejoran la accesibilidad y el SEO**, además de soportar elementos multimedia nativos como video y audio.

[HTML5 Manz.dev](#)

SCSS (Sassy CSS)

SCSS es una extensión de CSS que añade funcionalidades avanzadas como variables y anidamiento entre otras, facilitando una escritura más limpia, estructurada y reutilizable del código de estilos. Es parte del preprocesador Sass y permite mantener proyectos grandes de forma más eficiente y escalable.

[Sass: Syntactically Awesome Style Sheets](#)

TS (TypeScript).

TypeScript es un superconjunto de JavaScript que añade tipado estático y otras funcionalidades como interfaces, clases y enumeraciones. Mejora la escalabilidad y mantenimiento de aplicaciones grandes, facilitando la detección de errores en tiempo de compilación. Es ampliamente usado en frameworks como Angular.

[TypeScript Language](#)

Bootstrap 5.

Bootstrap es un framework de **diseño frontend** que facilita la creación de interfaces web responsivas y estilizadas. Bootstrap 5 introduce mejoras, como la eliminación de dependencias de jQuery y nuevas clases de diseño, permite crear **páginas rápidas y adaptativas** para móviles, tabletas y escritorios.

[Bootstrap 5 Website](#)

Spring Boot v3.4.4.

Spring Boot es un framework Java que simplifica el desarrollo de aplicaciones backend al ofrecer configuración automática, un servidor embebido y una estructura modular. En esta aplicación, utiliza **Spring Web** para crear servicios

REST y **Spring Data JPA** junto con el driver de **MySQL** para interactuar con la base de datos de forma eficiente mediante entidades.

[Spring Boot](#)

Java 21

Java es un lenguaje de programación orientado a objetos, ampliamente utilizado para desarrollar aplicaciones empresariales, móviles y web. Es conocido por su portabilidad, seguridad y robustez. La versión 21, una versión de soporte a largo plazo (LTS).

[JDK21 Documentation](#)

MySQL.

MySQL es un sistema de **gestión de bases de datos relacional** muy popular. Se usa para almacenar, organizar y recuperar grandes cantidades de datos de manera eficiente. Es comúnmente utilizado junto con PHP en aplicaciones web.

[MySQL Documentation](#)

4. Desarrollo y secuenciación temporal

4.1. Diseño

4.1.1. Diagrama de aplicación

La arquitectura de la aplicación está compuesta por un frontend en Angular, que gestiona la interfaz de usuario y se comunica con el backend a través de una API REST, mediante peticiones HTTPS, utilizando métodos estándar como GET, POST, PUT y DELETE para acceder, crear, actualizar o eliminar datos.

El backend, desarrollado en Spring Boot, maneja la lógica de negocio y expone endpoints REST mediante Spring Web, mientras que Spring Data JPA se encarga de la interacción con la base de datos MySQL, que almacena los datos de forma persistente. Este enfoque asegura

una separación clara entre la presentación, la lógica de negocio y la persistencia de datos.

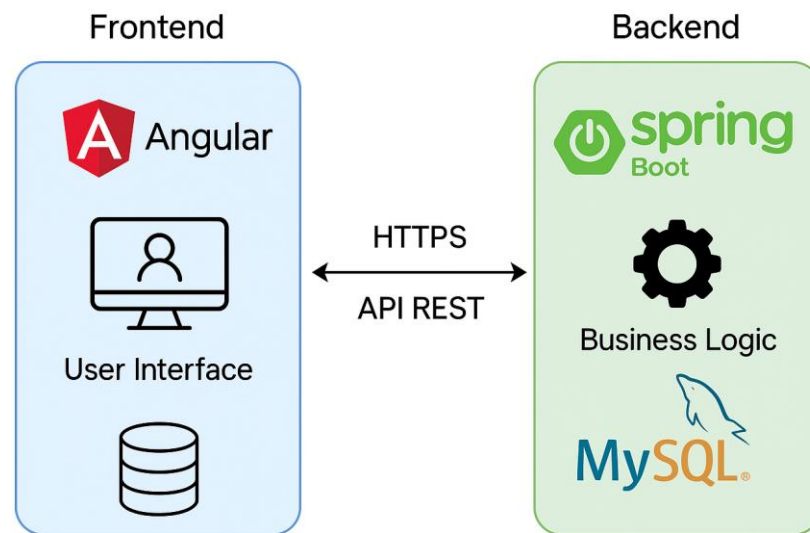


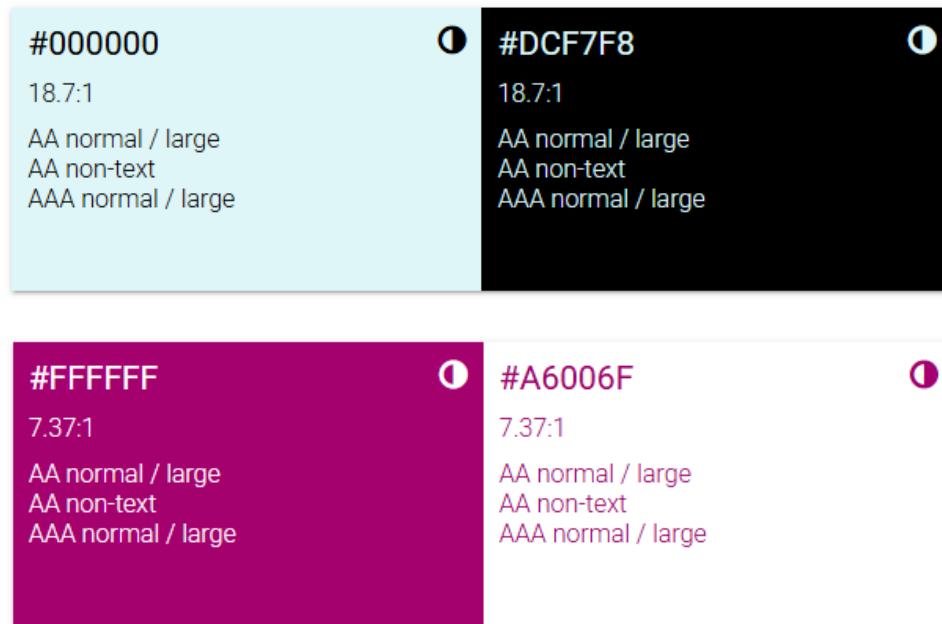
Ilustración 4-1. Diagrama de aplicación.

4.1.2. Diagrama E/R y esquema BBDD

4.1.3. Diseño de interfaces

Se prioriza un diseño limpio utilizando el blanco como color de fondo. El magenta que aporta energía y llama la atención, útil para destacar elementos interactivos o secciones clave de la interfaz y el turquesa transmite tranquilidad y claridad.

| | |
|--------------------|--------------------|
| #FFFFFF | #000000 |
| 21:1 | 21:1 |
| AA normal / large | AA normal / large |
| AA non-text | AA non-text |
| AAA normal / large | AAA normal / large |



En la estructura, el perfil de usuario cuenta con una barra de navegación lateral para navegar entre los distintos apartados.

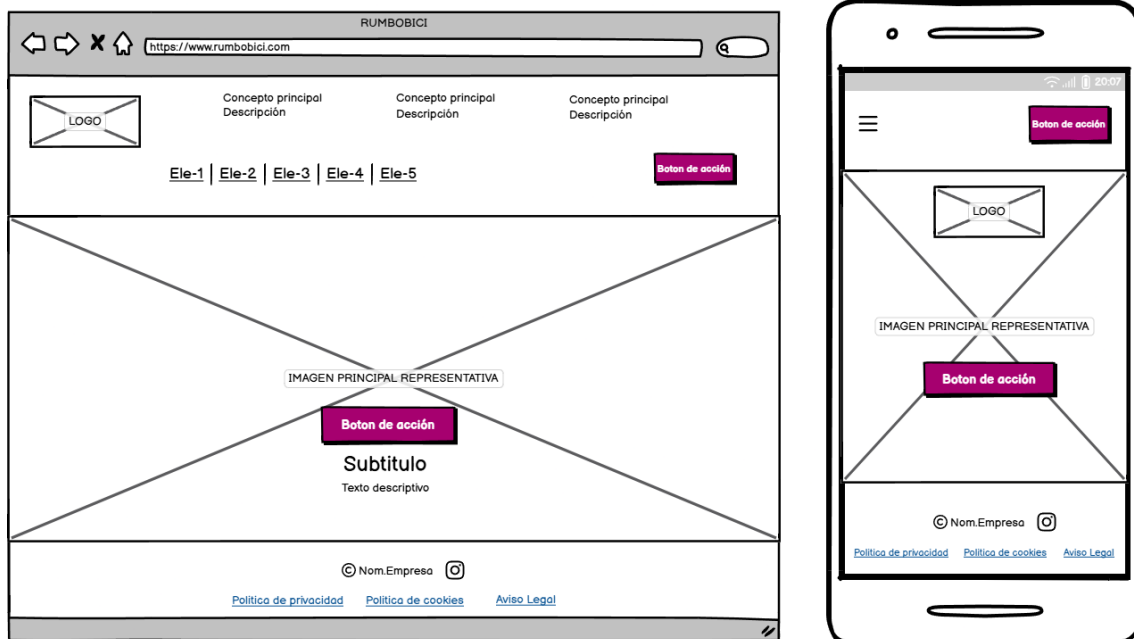


Ilustración 4-4. Wireframe Home page.

La tipografía elegida ha sido LT Wave cuyo diseño transmite una sensación de ligereza y dinamismo, ideal para un entorno digital enfocado en la experiencia del ciclismo y la conexión con el entorno natural.

Aporta personalidad sin perder claridad, alineándose con la filosofía del proyecto.

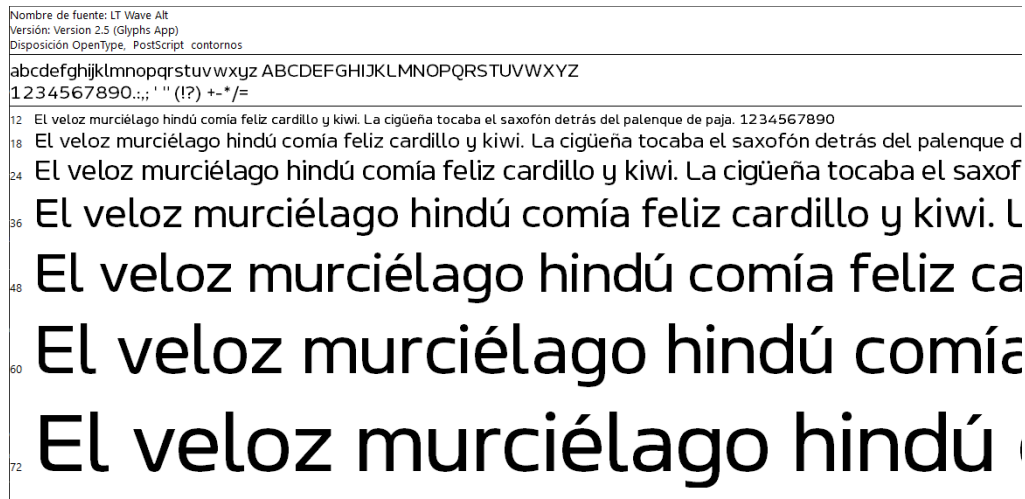


Ilustración 4-5. Tipografía.

4.2. Pruebas

A lo largo del desarrollo del proyecto se han llevado a cabo diversas **pruebas de usuario** con el objetivo de verificar el correcto funcionamiento de las funcionalidades principales de la aplicación.

Estas pruebas han permitido comprobar que el login de usuarios, así como la navegación entre las distintas secciones mediante enlaces y rutas, se realiza sin errores.

Del mismo modo, se ha validado la correcta visualización, creación, edición, eliminación y categorización de rutas, asegurando una experiencia fluida e intuitiva.

Estas comprobaciones garantizan que el sistema responde de forma adecuada ante las acciones del usuario, contribuyendo a una aplicación robusta y funcional.

5. Conclusiones

5.1. Grado de cumplimiento de los requisitos fijados

El desarrollo del proyecto ha alcanzado el cumplimiento de los requisitos establecidos, cubriendo de forma completa las funcionalidades previstas.

Además, se han implementado tecnologías y herramientas actuales que han enriquecido la solución final, mejorando tanto el rendimiento como la experiencia de usuario.

El resultado ha sido satisfactorio, no solo por cumplir con los objetivos marcados, sino también por superarlos con nuevas aportaciones técnicas que aportan valor añadido al proyecto.

5.2. Propuestas de mejora o ampliaciones futuras

Mecanismo de autenticación más robusto con Json Web Token.

- El backend valida las credenciales y genera un token firmado (JWT).
- El frontend (Angular) guarda el token en localStorage y lo envía en cada petición (en el header Authorization).
- El backend verifica ese token y autoriza al usuario.

Implementar un sistema de interacción entre los usuarios, como comentarios, valoraciones o la posibilidad de compartir rutas.

La incorporación de un panel de administración permitiría gestionar usuarios, rutas y contenidos de forma más eficiente. Con un rol específico de administrador, se podría controlar el acceso, realizar revisiones de contenido y mantener la integridad del sistema.

6. Guías

<https://rumbobici-7807d.web.app/home>

user: user1 | pass: Abc..123

7. Referencias

Frontend

Diseño

<https://balsamiq.com/>

<https://www.dafont.com/es/lt-wave.font>

Desarrollo

<https://getbootstrap.com/>

<https://angular.dev/>

<https://material.angular.io/components/categories>

Backend

Bases de datos

<https://www.edrawmax.com/>

<https://www.mysql.com/products/workbench/>

Desarrollo

<https://spring.io/projects/spring-boot>

<https://www.postman.com/>

Despliegue

<https://firebase.google.com/>

<https://railway.com/>

Diagramas

<https://app.diagrams.net/>