
Documentación del Proyecto: Fórmula 1

Proyecto

Fórmula 1

Aplicación web para consultar información de la temporada de Fórmula 1, incluyendo calendarios de carreras, circuitos, pilotos y escuderías, con estadísticas detalladas y datos históricos.

Objetivo

Ofrecer a los usuarios una plataforma donde puedan:

- Consultar las próximas carreras y su calendario completo.
- Ver detalles de cada circuito y Gran Premio.
- Explorar información de pilotos y escuderías, incluyendo estadísticas, victorias y campeonatos.
- Acceder a datos históricos y temporada actual de cada piloto y escudería.

Tecnologías utilizadas

Backend

Categoría	Tecnología
Lenguaje	Java
Framework	Spring Boot
Persistencia	JPA / Hibernate
Base de datos	MySQL (o MariaDB)
Dependencias	Spring Web, Spring Data JPA, Lombok

Frontend

Categoría	Tecnología
Lenguaje	TypeScript / JavaScript
Framework	Angular
Estilos	CSS

Estructura del Backend

3.1 Modelos (Entities)

- **Calendario:** Representa la temporada con lista de circuitos.
- **Circuito:** Información de cada circuito, fecha, longitud, récords, victorias, etc.
- **Escudería:** Datos históricos, temporada actual, pilotos oficiales y estadísticas.
- **Piloto:** Información personal, datos deportivos, estadísticas y escudería actual.

3.2 DTO

- **PilotoEscuderiaDTO:** Combina información del piloto con su escudería actual para mostrar en el frontend.

3.3 Repositorios

- **CalendarioRepository**
- **CircuitoRepository**
- **EscuderiaRepository**
- **PilotoRepository**

3.4 Servicios

- **CalendarioService:** Gestión de calendarios, ordenando circuitos por fecha.
- **CircuitoService:** CRUD de circuitos y búsqueda por nombre de GP.
- **EscuderiaService:** CRUD de escuderías, búsqueda por nombre y piloto.
- **PilotoService:** CRUD de pilotos, búsqueda por escudería y conversión a DTO.

3.5 Controladores

- **CalendarioController:** CRUD y consulta por temporada.
- **CircuitoController:** CRUD y búsqueda de circuitos por GP.
- **EscuderiaController:** CRUD, búsqueda por nombre y piloto.

- **PilotoController:** CRUD, búsqueda por escudería y DTO.

Endpoints principales

Calendarios

Método	Endpoint	Descripción
GET	/api/calendarios	Listar todos los calendarios.
GET	/api/calendarios/{id}	Obtener calendario por ID.
POST	/api/calendarios	Crear calendario con lista de circuitos.
PUT	/api/calendarios/{id}	Actualizar calendario.
DELETE	/api/calendarios/{id}	Eliminar calendario.
GET	/api/calendarios/temporada/{temporada}	Obtener calendario por temporada.

Circuitos

Método	Endpoint	Descripción
GET	/api/circuitos	Listar todos los circuitos.
GET	/api/circuitos/{id}	Obtener circuito por ID.
POST	/api/circuitos	Crear circuito.

POST	/api/circuitos/lista	Crear lista de circuitos.
PUT	/api/circuitos/update/{id}	Actualizar circuito.
DELETE	/api/circuitos/eliminar/{id}	Eliminar circuito.
GET	/api/circuitos/gp/{nombreGP}	Buscar circuito por nombre de GP.

Escuderías

Método	Endpoint	Descripción
GET	/api/escuderias	Listar todas las escuderías.
GET	/api/escuderias/{id}	Obtener escudería por ID.
POST	/api/escuderias	Crear escudería.
PUT	/api/escuderias/{id}	Actualizar escudería.
DELETE	/api/escuderias/{id}	Eliminar escudería.
GET	/api/escuderias/nombre/{nombre}	Buscar escudería por nombre.
GET	/api/escuderias/piloto/{pilotoId}	Obtener escudería de un piloto.

Pilotos

Método	Endpoint	Descripción
GET	/api/pilotos	Listar todos los pilotos.
GET	/api/pilotos/{id}	Obtener piloto por ID.
POST	/api/pilotos	Crear piloto.
PUT	/api/pilotos/{id}	Actualizar piloto.
DELETE	/api/pilotos/{id}	Eliminar piloto.
GET	/api/pilotos/escuderia/{escuderiaId}	Obtener pilotos por escudería.
GET	/api/pilotosDto/{id}	Obtener información de piloto + escudería en DTO.

Base de datos

- **Nombre:** `formula1_jpa`
- **Pre-requisito:** Debe estar creada en MySQL antes de correr el backend.
- **Tablas** generadas automáticamente por JPA: `Calendario`, `Circuito`, `Escudería`, `Piloto`.

Estructura del Frontend

6.1 Componentes

- **CalendarioComponent:** Muestra el calendario completo de la temporada y próximos GP.
- **CircuitoComponent:** Detalle de cada circuito, récords, longitud, victorias, mapa del circuito.

- **EscuderiaComponent**: Información de la escudería, pilotos, estadísticas y temporada actual.
- **PilotoComponent**: Información del piloto, estadísticas y escudería asociada.

6.2 Servicios

- **CalendarioService**: Consume endpoints de calendarios.
- **CircuitoService**: Consume endpoints de circuitos.
- **EscuderiaService**: Consume endpoints de escuderías.
- **PilotoService**: Consume endpoints de pilotos y DTO.

6.3 Rutas

- **/calendario** – Vista del calendario de la temporada.
- **/circuitos** – Lista y detalle de circuitos.
- **/escuderias** – Lista y detalle de escuderías.
- **/pilotos** – Lista y detalle de pilotos.

6.4 Estilos

- **CSS** para diseño *responsive* y contenedores limpios.
- Tarjetas de pilotos y escuderías con imágenes y estadísticas.
- No hay *scroll* horizontal en pantallas normales.

Instalación y ejecución

Backend

1. **Clonar** el repositorio usando `git clone <url-del-repositorio>`.
2. **Crear la base de datos** en MySQL con `CREATE DATABASE formula1_jpa`.
3. **Configurar** el archivo `application.properties` con tus credenciales de MySQL (URL, usuario y contraseña), asegurándose de que `spring.jpa.hibernate.ddl-auto` esté en `update`.
4. **Ejecutar** la aplicación con **Spring Boot**, desde el IDE o desde la terminal con `./mvnw spring-boot:run`.
 - La API estará disponible en `http://localhost:8080/api`.

Frontend

1. **Ingresa**r a la carpeta del frontend con `cd formula1-frontend`.
2. **Instalar** las dependencias usando `npm install`.
3. **Ejecutar** la aplicación Angular con `ng serve`.
4. **Acceder** desde el navegador en `http://localhost:4200`.