  
API F1

Contenido

[Introducción 3](#_Toc128337574)

[JQuery 4](#_Toc128337575)

[Obtener datos. 4](#_Toc128337576)

[Presentar datos 4](#_Toc128337577)

[Interactividad 5](#_Toc128337578)

[Efectos y animaciones 5](#_Toc128337579)

[Bootstrap 6](#_Toc128337580)

[Chart.js 7](#_Toc128337581)

[LeafletJS 8](#_Toc128337582)

[PaginationJS 9](#_Toc128337583)

[Funcionamiento de la web. 10](#_Toc128337584)

[Página principal 10](#_Toc128337585)

[Pilotos 10](#_Toc128337586)

[Escuderías 12](#_Toc128337587)

[Circuitos 13](#_Toc128337588)

[Referencias 14](#_Toc128337589)

# Introducción

Este proyecto es una aplicación web que utiliza la API de F1 para mostrar información actualizada y relevante sobre el campeonato mundial de Fórmula 1.

La API de F1 proporciona una gran cantidad de datos sobre los equipos, pilotos, carreras, resultados y estadísticas de la temporada actual y de temporadas anteriores. Este proyecto utiliza esta información para crear una experiencia interactiva que permite a los usuarios explorar los detalles de cada carrera, seguir el desempeño de su equipo o piloto favorito y mantenerse actualizados sobre las últimas noticias y actualizaciones del campeonato.

Este documento proporciona una guía detallada sobre cómo utilizar nuestra aplicación web, incluyendo información sobre cómo acceder a la API de F1, cómo realizar solicitudes para obtener datos específicos y cómo visualizar la información de manera efectiva.

# JQuery

## Obtener datos.

Para las peticiones asíncronas al servidor se ha utilizado el método getJSON de JQuery. Este método se usa para obtener los datos JSON del servidor utilizando AJAX.

Texto

Descripción generada automáticamente

Este método hace una petición al servidor y obtiene una “respuesta” que serían los datos JSON con los que se trabajará.

## Presentar datos

Para presentar los datos se utilizan eventos como:

* $(“boton”).click para ejecutar acciones cuando se pulsa un botón.
  + 
* $(“document).ready ejecutar acciones cuando se ha cargado el documento.
  + 

Para la manipulación del DOM se utilizan las funciones:

* Append() para insertar contenido en HTML.
  + Texto

    Descripción generada automáticamente
* Empty() para eliminar el contenido HTML del documento
  + 

## Interactividad

Se ha puesto a disposición del usuario un selector donde elegir la temporada donde obtener los datos.

Texto

Descripción generada automáticamente con confianza media

Este desplegable contiene todas las temporadas de las que se tienen registros. El propio desplegable es una petición de los años al servidor.

Texto

Descripción generada automáticamente

Este código obtiene las temporadas del servidor y va introduciendo los años en los <option> del formulario.

## Efectos y animaciones

Se ha añadido una sutil animación a la hora de presentar las “tarjetas” con los datos solicitados. Esta animación es fadeIn(), esta animación muestra los datos como apareciendo en la web.



# Bootstrap

A la hora de introducir los componentes HTML en la web se han utilizado los aportados por el framework Bootstrap.

Básicamente todos los componentes que se encuentran en la web son de Bootstrap ya que ofrece plantillas responsivas liberando de la tediosa tarea de tener que diseñarlo a mano.

Por ejemplo, el componente que mas se repite en toda la web es la barra de navegación.

Texto

Descripción generada automáticamente

Imagen que contiene Texto

Descripción generada automáticamente

Patrón de fondo

Descripción generada automáticamente

# Chart.js

Esta librería permite crear gráficos con un diseño moderno en la web. En este proyecto se ha utilizado cuando se pulsa sobre un piloto para ver su posición media y la media de puntos obtenidos.

Gráfico, Gráfico de barras

Descripción generada automáticamente

Texto

Descripción generada automáticamente

El gráfico introducido es un gráfico con dos barras donde indican la media del piloto.

# LeafletJS

Leaflet es una librería de código abierto que aporta un mapa interactivo a la web. En esta web se ha utilizado para indicar la ubicación cuando se selecciona un circuito.

Mapa

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Se le aporta una latitud y una longitud y ya nos sitúa en el lugar, adicionalmente se puede agregar un marcador en el lugar deseado con una descripción.

# PaginationJS

Pagination es un plugin que aporta una forma de paginar los resultados obtenidos en la web de manera simple. Este plugin usa jQuery y Bootstrap para mostrar las páginas en el documento.

Texto

Descripción generada automáticamente

Para usar este plugin hay que indicar un contenedor para los datos. Recibe la petición AJAX, se le indica lo que tiene que buscar y el tamaño de página. Después contiene un bucle que recorrerá el documento pasado. En este bucle se van introduciendo los datos.

Tabla

Descripción generada automáticamente con confianza media

# Funcionamiento de la web.

## Página principal

La página principal contiene un menú con las opciones de la web. Aquí se puede seleccionar lo que se desea consultar.

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

## Pilotos

Aquí se pueden visualizar las alineaciones de pilotos que tenía la competición en determinado año, para ello hay un desplegable que contiene los años en los que hay disponibilidad de datos. El usuario selecciona un año y pulsa en buscar.

Texto

Descripción generada automáticamente con confianza media

Al pulsar en buscar aparecerán los pilotos que competían ese año. Se puede acceder a su Wikipedia o pulsando en su nombre a mas información acerca de dicho piloto.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Al pulsar sobre un piloto nos abrirá una página con dos botones. Uno para ver el histórico de resultados de ese piloto o todas las temporadas donde ha corrido.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Si se pulsa en resultados aparecerán todos los resultados del piloto de mas antiguos a más nuevos y un gráfico con la media de su trayectoria.

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Si se pulsa en temporadas aparecerán todas las temporadas en activo del piloto.

Gráfico

Descripción generada automáticamente con confianza media

Los resultados se pueden paginar seleccionando en el desplegable el numero de resultados que se quiere mostrar por pantalla.

## Escuderías

Aquí se pueden visualizar las escuderías que han competido en la F1 según el año seleccionado. Para ello hay disponible un selector para indicar el año del cual ver los resultados.

Texto

Descripción generada automáticamente

Una vez seleccionado el año y pulsado el botón de buscar aparecerán las escuderías. Aquí se puede acceder a la Wikipedia de la escudería o pulsando en el nombre a más información acerca de esta.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Cuando se pulsa en una escudería se accede a una página donde muestra la tabla final de la temporada seleccionada y como quedó la escudería resaltada en azul.

Tabla

Descripción generada automáticamente

## Circuitos

En esta página se pueden ver todos los circuitos, que han existido en la F1.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Al pulsar en el nombre de uno accede a una página donde se muestra su ubicación en el mapa y un marcador con un enlace para acceder a su página de la Wikipedia.

Diagrama

Descripción generada automáticamente

# Referencias

Bootstrap. (2023). *Bootstrap*. Obtenido de https://getbootstrap.com/

Chart.js. (2023). *Chart.js*. Obtenido de https://www.chartjs.org/

Ergast. (2023). *Ergast*. Obtenido de http://ergast.com/mrd/

Leaflet. (2023). *Leaflet*. Obtenido de https://leafletjs.com/

Pagination.js. (2023). *Pagination.js*. Obtenido de https://pagination.js.org/

Postman. (2023). *GetPostman*. Obtenido de https://documenter.getpostman.com/view/11586746/SztEa7bL