



Ciclo Formativo de Grado Superior  
Desarrollo de Aplicaciones  
Multiplataforma

**TRABAJO FIN DE CICLO**

**“Aplicación de análisis estadístico sobre los  
resultados del Campeonato del Mundo de Fórmula 1”**

# Índice general

Tabla de contenido.....	2
CAPÍTULO I: DEFINICIÓN DEL PROYECTO .....	4
1.- INTRODUCCIÓN.....	5
2. DEFINICIÓN DEL PROBLEMA .....	5
2.1 Definición del problema real.....	6
2.2 Definición del problema técnico .....	6
3. Antecedentes .....	8
3.1 Antecedentes y estado del arte.....	8
4. Posibles trabajos y solución elegida .....	10
4.1 Estudio del estándar API REST .....	10
4.2 Estudio de los paradigmas Frontend y Backend.....	12
4.3 Estudio del framework Angular .....	13
4.4 Estudio del protocolo HTTP .....	14
5. Restricciones.....	15
5.1 Factores de dato .....	15
5.2 Factores estratégicos.....	16
6. Recursos.....	16
CAPITULO II: OBJETIVOS .....	17
1. Objetivos formales .....	18
2. Objetivos operacionales.....	18
CAPITULO III: ESPECIFICACIÓN DE REQUISITOS .....	19
1.- DESCRIPCIÓN GENERAL.....	20
2.- ESPECIFICACIÓN DE REQUISITOS .....	20
CAPITULO IV: ESPECIFICACIONES DEL SISTEMA.....	21
1.- Descripción de la información.....	22
1.2 Descripción de los tipos de entidad.....	22

2.- Representación del flujo de la información .....	24
3.- Especificaciones del software utilizado para el desarrollo del proyecto .....	25
3.1 Framework Angular 7 y TypeScript .....	25
3.2 Framework Electron para aplicaciones de escritorio .....	25
3.3 Preprocesador CSS: SASS .....	26
3.4.- Funcionamiento de nuestro software cuando un usuario hace una petición .....	29
CAPITULO V: PRUEBAS .....	33
1. Pruebas específicas para comprobar el funcionamiento del software .....	34
CAPITULO VI: CONCLUSIONES .....	35
ANEXO I: MANUAL DE USUARIO .....	37
1. Acceso a la aplicación .....	38
1.1 Página principal .....	38
1.2 Apartado circuitos .....	41
1.3 Apartado constructores .....	44
1.4 Apartado pilotos .....	49
1.5 Apartado temporadas .....	54

# **CAPÍTULO I: DEFINICIÓN DEL PROYECTO**

### 1.- INTRODUCCIÓN

Desde hace un tiempo es indudable que el mundo del Motorsport ha vivido una transformación inmensa, solo hay que ver el auge de la ingeniería en las distintas competiciones de motor. Este auge en la mayoría de ocasiones deja en segundo plano a la persona que verdaderamente importa dentro de un coche o una moto, el piloto.

Si nos acordamos de un artículo publicado por el afamado medio [motorsport.com](https://www.motorsport.com), en una de sus noticias ponen en valor el trabajo de un ingeniero en la Fórmula 1, cito textualmente:

“Soy ingeniero de carrera. Hay un grupo de seis ingenieros que trabajan conmigo en un mismo coche. Siendo claro, es mi responsabilidad hacer que el coche funcione lo más rápido posible”

Leyendo esto nos puede chocar un poco, que sea un ingeniero el que tenga que hacer el coche rápido, lo más lógico para alguien que no conozca el mundo de la Fórmula 1 es pensar que el coche lo deben hacer rápido en todo caso los mecánicos, pero no es así.

La labor de los ingenieros en la Fórmula 1 es clave, ya que hoy día, un coche es prácticamente un ordenador con ruedas. Y todos los aspectos del coche son controlados a través de la [telemetría](#).

Viendo un poco lo que se ha expuesto ahora nos toca poner en contexto la situación de la otra parte, al aficionado. Sinceramente llevo siendo aficionado de la Fórmula 1 desde hace prácticamente 14 años y cada vez es más difícil comprender todo el mundillo, desde el funcionamiento de los coches, hasta los mismos resultados de los pilotos, ya que no solo se comparan entre sí, sino que incluso son comparados con coches de otras temporadas y con otra reglamentación.

Viviendo esta situación como aficionado la idea de este proyecto es centrarse en la problemática de los resultados y sus estadísticas. Queremos hacer que un aficionado a este deporte no tenga que buscar entre cientos de páginas para encontrar los resultados deportivos de su piloto o constructor favorito.

Aparte del propósito anterior, la finalidad del proyecto es exponer todos los datos que haya disponibles relacionados con la Fórmula 1, de forma clara, concisa y directa y lo mejor de todo, de forma automatizada y al momento. Para ello nos basaremos en distintas fuentes de código abierto y gratuitas, creando una herramienta fácil de usar y sin ningún coste.

### 2. DEFINICIÓN DEL PROBLEMA

En este apartado se va a describir las características del problema a resolver con el desarrollo e implementación del proyecto en cuestión. Para poder diferenciar bien los problemas y analizarlos de forma más exhaustiva vamos a distinguir entre Problema Real, visión que tiene el usuario del problema (que no debe saber cómo está hecha nuestra aplicación) y Problema Técnico, visión específica que se dará para los desarrolladores interesados en el proyecto.

### 2.1 Definición del problema real

El problema estudiado en el proyecto es la prevención del desconocimiento por parte de la población en general y de forma más específica a los aficionados al mundo de la Fórmula 1.

El problema que se le plantea al usuario es no saber dónde consultar los resultados deportivos de sus atletas favoritos, debido principalmente a dos motivos básicos: falta de credibilidad de algunas aplicaciones y ausencia de sitios o aplicaciones dedicadas exclusivamente a los resultados que los usuarios desean.

### 2.2 Definición del problema técnico

En este apartado se pretende dar una visión del problema desde el punto de vista técnico. Indicando todas las características más relevantes en cuanto a desarrollo y componentes utilizados.

#### 2.2.1 Funcionamiento

El software desarrollado para este proyecto debe conseguir un objetivo fundamental, proporcionar a los usuarios una aplicación web para poder consultar los resultados históricos y actualizados del campeonato de Fórmula 1, reuniendo datos de pilotos, constructores, y circuitos.

Para conseguir tal resultado se usan datos de una API de dominio público la cual provee datos “planos” de todas las áreas citadas anteriormente. Nuestra aplicación se encargará de mapear esos datos y mostrárselos al usuario de una manera amigable y clara.

El software está dirigido al público en general, esto quiere decir que no se necesitan ningún tipo de conocimientos técnicos, aunque el usuario objetivo será el aficionado más acérrimo al campeonato de Fórmula 1. La aplicación va a estar desplegada en un entorno web, lo que la hace accesible desde todos los dispositivos que puedan acceder a un navegador web.

#### 2.2.2 Entorno

Nuestro aplicativo se encuentra dividido en tres entornos que abarcan los aspectos principales a tener en cuenta: entorno hardware, entorno software y el entorno de usuario.

- **Entorno hardware:** Al ser un proyecto principalmente centrado en el desarrollo del software el entorno hardware es menos importante. Aun así se debe especificar que al ser una aplicación que se mueve directamente en el navegador, el único hardware necesario para que funcione sería cualquier dispositivo que cuente con un navegador web. De todas formas, para que esta aplicación sea accesible desde varios ordenadores al mismo tiempo y por varios usuarios a la vez será desplegada en un servidor web.
- **Entorno software:** Nuestro software ha sido desarrollado con el lenguaje de programación TypeScript (Derivado del tan famoso JavaScript), usando como framework de desarrollo el tan aclamado Angular. Para ello hemos tenido que instalar Node.js (entorno de ejecución de JavaScript, por hacer un símil, parecido al SDK de Java) que trae un potente gestor de paquetes integrado (npm, Node Package Manager).
- **Entorno de usuario:** El usuario final al que va destinado este software no debe tener ningún conocimiento previo, ya que nuestra herramienta es bastante intuitiva y de fácil uso.

#### 2.2.3 Vida esperada

El desarrollo del software, desde sus inicios es conocido por tener un ámbito de aplicación percedero, el software que se ha desarrollado mitiga éstos efectos al mínimo ya que, si la API

de la que consume no tiene cambios a gran escala, nuestra aplicación seguirá funcionando con normalidad (hay que tener en cuenta que la API de la que consume es experimental, por lo que en las primeras etapas de la aplicación puede ser que sea más propensa a errores). La vida de nuestro software, como la de cualquier otra aplicación, va a venir determinada por los cambios y el mantenimiento realizado, pero, sobre todo, como hemos indicado antes por los cambios en la API.

### ***2.2.4 Ciclo de mantenimiento***

El sistema desarrollado no necesita de mantenimiento alguno, salvo el que implicaría su actualización respecto a nuevos conceptos de aplicación. Además, nuestra aplicación consta de una interfaz flexible y reutilizable para próximas versiones de este software. Lo único es que se deberá comprobar el correcto funcionamiento de la aplicación cada cierto tiempo, y sobre todo, al final y al inicio de la temporada del campeonato, ya que ahí la aplicación puede tener ciertos problemas para adaptarse a la nueva temporada.

El software aquí desarrollado puede ser utilizado como punto de partida para futuros trabajos y mejoras del mismo que impliquen nuevas funcionalidades. En trabajos futuros, capítulo 5 de este documento Conclusiones, se determinará diferentes usos que puede llegar a tener este software y las modificaciones que puede sufrir para llevarlos a cabo.

### ***2.2.5 Estandarización***

El sistema se ha desarrollado en el lenguaje de programación TypeScript, que al ser un lenguaje derivado de JavaScript (conocido por su flexibilidad y uso extendido) y diseñado bajo la tutela de Microsoft cumple todos los estándares que cualquier lenguaje de programación debe tener hoy día; programación orientada a objetos, funcional, asíncrona, etc...

La interfaz ha sido diseñada siguiendo los estándares que hay en el mercado, de manera que el usuario se familiarice con ella lo más rápido posible.

El código fuente está totalmente documentado, cualquier usuario con conocimientos de programación en estos lenguajes puede analizarlo sin dificultad. Por supuesto esta documentación del código ayuda a realizar las futuras mejoras y actualizaciones.

### ***2.2.6 Calidad y fiabilidad***

El término calidad se ha referido a la concordancia existente entre los requisitos funcionales y de rendimiento debidamente establecidos, la adaptación a los estándares de desarrollo seleccionados explícitamente documentados y a las características implícitas esperadas de todo software desarrollado profesionalmente. Nuestro sistema se ajusta a la definición de calidad, por lo que podemos afirmar que la posee. Se han conseguido los requisitos establecidos, así como un gran nivel de estandarización. El contenido se ha desarrollado a partir de multitud de fuentes. Tras sucesivas pruebas de control, podemos afirmar que el código fuente del sistema es correcto, lo que nos garantiza en un alto porcentaje la ausencia de fallos de ejecución o la correcta actuación del programa si estos ocurren.

### 3. Antecedentes

#### 3.1 Antecedentes y estado del arte

A continuación, se muestra el resultado de la investigación que se ha realizado para determinar que herramientas hay actualmente en el mercado que realicen una función parecida a la nuestra, explicaremos cómo funcionan (si hay documentación al respecto) y en qué nuestra aplicación las mejora.

Viendo que el campo donde nuestro aplicativo se centra es un campo bastante cubierto por los medios, solo basta buscar "Resultados Fórmula 1" para que aparezcan delante de ti cientos de resultados, esto es en parte porque casi todos los medios de comunicación cubren los resultados del campeonato o de las carreras, pero no llegan al mismo nivel de análisis que nuestro software.

Para evitar tener que analizar todos esos sitios nos vamos a centrar en el sitio oficial de la Fórmula 1 y en las distintas aplicaciones que hay relacionadas directamente con los datos, y que están desarrolladas basándose en la misma API de la que consumimos nosotros.

##### ***3.1.1 Página oficial de la Fórmula 1 ([www.formula1.com](http://www.formula1.com))***

Como era de esperar es la alternativa más completa a la hora de encontrar resultados sobre el campeonato. Analizando la sección que tiene por objetivo hacer lo mismo que nuestra aplicación, podemos hacer el siguiente resumen:

- Apartado "Races" (Carreras): Nos muestra una lista con las carreras del campeonato del año en curso, junto con el ganador de cada una de ellas (si lo hubiera), además de venir indicadas con un pequeño sombreado las que ya se han disputado. Cuando se pulsa encima de cualquiera de las carreras nos lleva a una página con el detalle de esta con los resultados y la información del circuito.
- Apartado "Standings" (Resultados): Vemos una serie de listas con información acerca de los resultados de pilotos, equipos, y (de nuevo) carreras. La información viene dispuesta de una forma que puede llegar a confundir al usuario ya que se muestra en una combinación de listas que el usuario debe combinar para ver los resultados que desee. Dentro de este apartado posee una sección dedicada exclusivamente a un "archivo", donde están los resultados de todas las temporadas anteriores a la que se encuentre en curso. También incluye un apartado de "premios" donde se pueden consultar los "Driver of the Day" y el premio "DHL Fastest Pit Stop Award".
- Apartado "Drivers" (Pilotos): Accedemos a una lista de los pilotos de esta temporada, viendo su nombre, número y fotografía. Además nos la lista con las posiciones del campeonato de pilotos del año en curso, por lo que de un vistazo podemos ver una gran cantidad de información.
- Apartado "Teams" (Equipos): Nos lleva a una vista con un diseño bastante elaborado en la cual se listan los equipos oficiales del campeonato para esa temporada, en cada entrada de esta lista tenemos el nombre del equipo, los pilotos actuales, el número de podiums y el número de títulos que tiene el equipo en cuestión. Al pulsar encima de cada equipo nos llevará a una página dedicada exclusivamente al mismo, donde vemos estadísticas más detalladas del equipo.

Vistos estos apartados se ha realizado una valoración general del sitio, por lo que bajo nuestro punto de vista es quizás la alternativa más vistosa, ya que el diseño es espectacular, además que tenga el respaldo de ser un sitio oficial otorga de mucha confianza a los usuarios.



Por otro lado, encontramos un poco difícil encontrar algunos detalles en la información (quizás por el diseño tan recargado de la página), y puede que para los usuarios que no estén acostumbrados al diseño de la página les cueste un poco moverse por ella.

Otro aspecto negativo que hemos encontrado es el de tener tanto artículo de noticias y opinión, ya que los usuarios que quieran saber solo los resultados quizás les llegue a molestar un poco.

En comparación a este sitio, nuestra aplicación está más centrada en los resultados propiamente dichos, sin ofrecer más distracciones al usuario. Aun así pensamos que nuestra aplicación puede estar más enfocada, o que sea de más utilidad, a usuarios familiarizados con la estadística.

### ***3.1.2 Formula Graph ([www.formulagraph.com](http://www.formulagraph.com))***

Nada más visitar este sitio lo primero que nos llama la atención es su simpleza, ya que todos los datos mostrados en la misma son a través de una gráfica por lo que tiene sus ventajas y sus inconvenientes.

Se puede alternar entre ver los resultados de una temporada en concreto o ver los resultados históricos de un determinado constructor o piloto. Aunque la disposición de la información es buena, la valoración se ve un poco lastrada por la falta de datos.

Por ejemplo, se echa en falta datos más específicos de los pilotos, tales como lugar de nacimiento, número de carreras, etc. Lo mismo se puede llegar a decir de los constructores de los cuales solo se puede saber los resultados a lo largo de su historia, en el campeonato de constructores, información a todas luces, escasa.

### ***3.1.3 F1 Fansite ([www.f1-fansite.com](http://www.f1-fansite.com))***

Antes de empezar el análisis de este sitio, debemos referirnos a su descripción:

"On F1-Fansite.com you can find all F1 results of each and every held Formula 1 grand prix. You can also find all championships since the beginning of Formula 1 in 1950 until the current season. This page displays every F1 season. Behind each year you will find the F1 results of each F1 GP and the drivers and constructors championship tables for each F1 season.

En F1-Fansite.com puede encontrar todos los resultados de F1 de todos y cada uno de los grandes premios de Fórmula 1 celebrados. También puede encontrar todos los campeonatos desde el comienzo de la Fórmula 1 en 1950 hasta la temporada actual. Esta página muestra cada temporada de F1. Detrás de cada año, encontrarás los resultados de F1 de cada GP de F1 y las mesas de campeonato de pilotos y constructores para cada temporada de F1."

Leído esto tenemos que decir que es fiel a lo que la página ofrece, ya que en materia de datos es bastante competente, teniendo una cantidad abundante, quizás su talón de Aquiles sea el diseño, que recuerda a los de las primeras páginas de Internet y su adaptación a distintos dispositivos es bastante mediocre.

### **3.1.4 “4Mula1” (<http://www.4mula1.ro/>)**

En esta ocasión esta aplicación tiene mejor diseño que la analizada anteriormente, aún así sigue sin convencer debido a los altos estándares de diseño que se han impuesto hoy día. En lo relativo a los datos vemos como tiene unos datos bastante precisos, aunque se echa de menos algo más de contenido multimedia, tales como vídeos, imágenes de equipos y pilotos, etc.

No se ha encontrado nada relacionado con los circuitos del campeonato, ni del actual ni de las temporadas anteriores, no sabemos si es que realmente no dispone de ellos o que simplemente el diseño no permite acceder a los mismos debido a su poca usabilidad.

## **4. Posibles trabajos y solución elegida**

Visto lo que hay actualmente en el mercado se llega a la conclusión de que ningún sitio o aplicación que esté en producción es perfecta, por lo cual en nuestro desarrollo deberá tener la unión de una fuente de datos confiable, una programación fuerte y estructurada con unos estándares de calidad. Para ello se ha llegado a la siguiente conclusión:

Nada más empezar a pensar en la realización del proyecto lo primero que se nos ocurrió es desarrollar un sistema completo, es decir configurar un servidor propio con alguno de los lenguajes de Backend más conocidos (Java, Python, PHP) y ocuparnos del modelado de entidades, relación de estas y lo más importante de todo, conseguir una consistencia en los datos. Esto además tenía otro inconveniente incluso peor, el inmenso volumen de datos. Y es que para tener alojados en un servidor tal cantidad de datos (recordemos que nuestra herramienta debe ofrecer registros de todos los campeonatos de Fórmula 1 desde el principio, año 1950) en el desarrollo debe primar la alta disponibilidad y la confiabilidad cosa que nosotros, debido a la falta de tiempo para desarrollar el proyecto (apenas mes y medio quitando el tiempo que nos llevaría aprender todas las tecnologías necesarias para acometer el proyecto) y al poco conocimiento que tenemos en la tecnología desestimamos desde el primer momento.

En cuanto al Frontend, es algo que había que hacer si o sí, ya que era lo que queríamos al inicio de todo, una interfaz amigable para el usuario, sencilla y con una orientación multiplataforma. Para ello se pensó desde un principio en Angular, debido a su alta popularidad y su conocida robustez en cuanto a la estructura de código. Dicho esto, en los siguientes apartados se detallará en profundidad como se ha llevado a cabo el desarrollo de este.

Antes de iniciar el proyecto, y para la realización de este se han tenido que investigar una serie de tecnologías. El resultado de este estudio se ha plasmado en los siguientes apartados.

### **4.1 Estudio del estándar API REST**

Buscando una definición sencilla, REST es cualquier interfaz entre sistemas que use HTTP para obtener datos o generar operaciones sobre esos datos en todos los formatos posibles, como XML y JSON. Es una alternativa en auge a otros protocolos estándar de intercambio de datos como SOAP (Simple Object Access Protocol), que disponen de una gran capacidad pero también mucha complejidad. A veces es preferible una solución más sencilla de manipulación de datos como REST.

REST cambió por completo la ingeniería de software a partir del 2000. Este nuevo enfoque de desarrollo de proyectos y servicios web fue definido por [Roy Fielding](#), el padre de la especificación HTTP y uno de los referentes internacionales en todo lo relacionado con la Arquitectura de Redes, en su estudio '[Architectural Styles and the Design of Network-based Software Architectures](#)'. En el campo de las APIs, REST (Representational State Transfer-Transferencia de Estado Representacional) es, hoy en día, el alfa y omega del desarrollo de servicios de aplicaciones.

### Características de REST

- **Protocolo cliente/servidor sin estado:** cada petición HTTP contiene toda la información necesaria para ejecutarla, lo que permite que ni cliente ni servidor necesiten recordar ningún estado previo para satisfacerla. Aunque esto es así, algunas aplicaciones HTTP incorporan memoria caché.
- Las operaciones más importantes relacionadas con los datos en cualquier sistema REST y la especificación HTTP son cuatro: POST (crear), GET (leer y consultar), PUT (editar) y DELETE (eliminar).
- Los objetos en REST siempre se manipulan a partir de la URI. Es la URI y ningún otro elemento el identificador único de cada recurso de ese sistema REST. La URI nos facilita acceder a la información para su modificación o borrado.
- Interfaz uniforme: para la transferencia de datos en un sistema REST, este aplica acciones concretas (POST, GET, PUT y DELETE) sobre los recursos, siempre y cuando estén identificados con una URI. Esto facilita la existencia de una interfaz uniforme que sistematiza el proceso con la información.
- **Sistema de capas:** arquitectura jerárquica entre los componentes. Cada una de estas capas lleva a cabo una funcionalidad dentro del sistema REST.
- **Uso de "hipermedios":** "hipermedia" es un término acuñado por [Ted Nelson](#) en 1965 y que es una extensión del concepto de hipertexto. Ese concepto llevado al desarrollo de páginas web es lo que permite que el usuario puede navegar por el conjunto de objetos a través de enlaces HTML. En el caso de una API REST, el concepto de hipermedia explica la capacidad de una interfaz de desarrollo de aplicaciones de proporcionar al cliente y al usuario los enlaces adecuados para ejecutar acciones concretas sobre los datos.

Ejemplo de respuesta de una petición a una API REST según el principio HATEOAS:

```
{
  "id": 78,
  "nombre": "Juan",
  "apellido": "García",
  "coches": [
    {
      "coche": "http://miservidor/concesionario/api/v1/clientes/78/coches/1033"
    },
    {
      "coche": "http://miservidor/concesionario/api/v1/clientes/78/coches/3889"
    }
  ]
}
```

### Ventajas que ofrece REST para el desarrollo

- **Separación entre el cliente y el servidor:** el protocolo REST separa totalmente la interfaz de usuario del servidor y el almacenamiento de datos. Eso tiene algunas ventajas cuando se hacen desarrollos. Por ejemplo, mejora la portabilidad de la interfaz a otro tipo de plataformas (MUY importante en nuestro proyecto), aumenta la escalabilidad de los proyectos y permite que los distintos componentes de los desarrollos se puedan evolucionar de forma independiente.
- **Visibilidad, fiabilidad y escalabilidad:** La separación entre cliente y servidor tiene una ventaja evidente y es que cualquier equipo de desarrollo puede escalar el producto sin excesivos problemas. Se puede migrar a otros servidores o realizar todo tipo de cambios en la base de datos, siempre y cuando los datos de cada una de las peticiones se envíen de forma correcta. Esta separación facilita tener en servidores distintos el front y el back y eso convierte a las aplicaciones en productos más flexibles a la hora de trabajar.
- **La API REST siempre es independiente del tipo de plataformas o lenguajes:** la API REST siempre se adapta al tipo de sintaxis o plataformas con las que se estén trabajando, lo que ofrece una gran libertad a la hora de cambiar o probar nuevos entornos dentro del desarrollo. Con una API REST se pueden tener servidores PHP, Java, Python o Node.js. Lo único que es indispensable es que las respuestas a las peticiones se hagan siempre en el lenguaje de intercambio de información usado, normalmente XML o JSON.

## 4.2 Estudio de los paradigmas Frontend y Backend

Según los últimos estudios cada uno de nosotros pasa más de 4 horas al día frente a un ordenador, durante las cuales interactuamos con un número “X” de páginas web. No podemos negar que la web forma parte de nuestro día a día, aunque va más allá de los diseños agradables y animaciones, pues por detrás encontramos una serie de tecnologías que hacen que la magia sea posible. Estas tecnologías, se encargan de que todo funcione correctamente para que el usuario no tenga ningún problema al mandar una petición, consulta u obtener datos. Estamos hablando de los lados frontend y backend que se unen para cumplir un mismo objetivo: buscar la satisfacción del usuario.

El **Frontend** se enfoca en el usuario, en todo lo que podemos interactuar y lo que vemos mientras navegamos. Así, nuestra web busca causar una buena impresión y agradar al usuario, para lo cual utiliza HTML, CSS y JAVASCRIPT. Buena experiencia de usuario, inmersión y usabilidad, son algunos de los objetivos que busca un buen frontend y hoy en día existen una gran variedad de frameworks, preprocesadores y librerías que nos ayudarán en esta tarea. Para un frontend la creatividad es el recurso más valioso, ya que tendrá que tomar fuentes, colores, imágenes y todos los recursos de los cuales disponga para crear sitios agradables que se vean bien en todos los dispositivos y resoluciones.

Por otro lado, tenemos el **Backend** enfocado en hacer que todo lo que está detrás de un sitio web funcione correctamente. Toma los datos, los procesa y los envía al usuario, además de encargarse de las consultas o peticiones a la Base de Datos, la conexión con el servidor, entre otras tareas que debe realizar en su día a día. Cuenta con una serie de lenguajes y herramientas que le ayudan a cumplir con su trabajo como PHP, Ruby, Python, JavaScript, SQL, MongoDB, MySQL, etc, estos son usados para crear sitios dinámicos. Como en muchos sitios la información se encuentra en constante cambio o actualización, una buena capacidad de respuesta y una velocidad óptima del sitio son responsabilidades que un backend debe de afrontar.

Ya que entendimos las funciones de cada uno, podemos afirmar que ninguno de los dos lados es mejor o menos importante que el otro, pues se complementan con el fin de cumplir el mismo objetivo...

### 4.3 Estudio del framework Angular

Es la herramienta que utilizaremos principalmente en nuestro proyecto, así que nos centraremos de lleno en ella en este apartado.

**Angular** es un **framework** de desarrollo para JavaScript creado por Google. La finalidad de Angular es facilitarnos el desarrollo de aplicaciones web SPA y además darnos herramientas para trabajar con los elementos de una web de una manera más sencilla y optima.

Una aplicación web **SPA** creada con Angular es una web de una sola página, en la cual la navegación entre secciones y páginas de la aplicación, así como la carga de datos, se realiza de manera dinámica, casi instantánea, asincrónamente haciendo llamadas al servidor (backend con un API REST) y sobre todo sin refrescar la página en ningún momento.

Antes de la aparición de este framework este tipo de aplicaciones web de una sola página se hacían usando solo JavaScript y jQuery, lo que era muy complejo, tedioso y difícil de mantener. Aparte de que necesitaba gran cantidad de código para hacer cualquier cosa. Este es el problema que viene a resolver Angular, convirtiendo el desarrollo de una web SPA en un proceso muy sencillo, muy limpio, más orientado a objetos y con una arquitectura basada en componentes y buenas prácticas.

Otra ventaja que tiene este framework es que está respaldado por Google y tiene una comunidad brutalmente grande detrás.

#### Ventajas de usar angular

##### 1. Facilidad de uso

Dado que Angular es un framework, ofrece muchas funcionalidades de serie que una simple biblioteca. Con otro software similar, lo más común es tener que usar varias bibliotecas de terceros a la hora de desarrollar una aplicación. Lo más probable es que se necesite algunas adicionales para hacer el routing (enrutado de la aplicación web), para la gestión de dependencias, para realizar llamadas a APIs REST, para hacer testing, etc...

Con Angular ya sabes desde el primer momento cómo organizar el código, como se realizan las diferentes tareas que necesitas, la arquitectura de la aplicación...

##### 2. TypeScript

Aunque se puede programar con ECMAScript (estándar de JavaScript) “puro”, el equipo de Angular decidió que haría todo el desarrollo con el lenguaje TypeScript, y casi toda la documentación utilizan este lenguaje.

Una de las ventajas de usar TypeScript en vez de JavaScript es la consistencia en la documentación. Por ejemplo, ES6 (o sea, ECMAScript 2015) ofrece varias formas diferentes de declarar un objeto, lo cual puede llegar a confundir. Con TypeScript esto no pasa, y toda la sintaxis y la manera de hacer las cosas en el código es la misma, lo que añade coherencia a la información y a la forma de leer el código.

Aunque Angular no te obliga a usar TypeScript, el equipo del core de Angular sí que lo ha adoptado y en la documentación sugiere usar TypeScript por defecto. Esta consistencia debería ayudar a evitar la confusión y la sobrecarga en la toma de decisiones derivadas de empezar con Angular.

Todo esto conlleva a largo plazo, en un mejor y más fácil mantenimiento de las aplicaciones.

### 3. Componentes web

Un componente en Angular es una porción de código que es posible reutilizar en otros proyectos de Angular sin apenas esfuerzo, lo que permite un desarrollo de aplicaciones mucho más ágil, pasando de un costoso MVC a una especie de puzle con los componentes.

El diseño de Angular adopta el estándar de los [componentes web](#). Se trata de un conjunto de APIs que te permiten crear nuevas etiquetas HTML personalizadas, reutilizables y auto-contenidas, que luego puedes utilizar en otras páginas y aplicaciones web. Estos componentes personalizados en navegadores modernos y con cualquier biblioteca o framework de JavaScript que funcione con HTML.

El [soporte actual](#) de este estándar por parte de los navegadores es muy reducido (básicamente Chrome), aunque existen [polyfills](#) para suplir esta carencia.

## 4.4 Estudio del protocolo HTTP

El Protocolo de Transferencia de Hipertexto (Hypertext Transfer Protocol) es un sencillo protocolo cliente-servidor que articula los intercambios de información entre los clientes Web y los servidores HTTP. La especificación completa del protocolo HTTP 1/0 está recogida en el RFC 1945. Fue propuesto por Tim Berners-Lee, atendiendo a las necesidades de un sistema global de distribución de información como el World Wide Web.

Desde el punto de vista de las comunicaciones, está soportado sobre los servicios de conexión TCP/IP, y funciona de la misma forma que el resto de los servicios comunes de los entornos UNIX: un proceso servidor escucha en un puerto de comunicaciones TCP (por defecto, el 80), y espera las solicitudes de conexión de los clientes Web. Una vez que se establece la conexión, el protocolo TCP se encarga de mantener la comunicación y garantizar un intercambio de datos libre de errores.

HTTP se basa en sencillas operaciones de solicitud/respuesta. Un cliente establece una conexión con un servidor y envía un mensaje con los datos de la solicitud. El servidor responde con un mensaje similar, que contiene el estado de la operación y su posible resultado. Todas las operaciones pueden adjuntar un objeto o recurso sobre el que actúan; cada objeto Web (documento HTML, fichero multimedia o aplicación CGI) es conocido por su URL.

### Etapas de una transacción HTTP

Para profundizar más en el funcionamiento de HTTP, veremos primero un caso particular de una transacción HTTP; en los siguientes apartados se analizarán las diferentes partes de este proceso.

Cada vez que un cliente realiza una petición a un servidor, se ejecutan los siguientes pasos:

- Un usuario accede a una URL, seleccionando un enlace de un documento HTML o introduciéndola directamente en el campo Location del cliente Web.
- El cliente Web descodifica la URL, separando sus diferentes partes. Así identifica el protocolo de acceso, la dirección DNS o IP del servidor, el posible puerto opcional (el valor por defecto es 80) y el objeto requerido del servidor.
- Se abre una conexión TCP/IP con el servidor, llamando al puerto TCP correspondiente. Se realiza la petición. Para ello, se envía el comando necesario (GET, POST, HEAD,...), la dirección del objeto requerido (el contenido de la URL que sigue a la dirección del servidor), la versión del protocolo HTTP empleada (casi siempre HTTP/1.0) y un conjunto variable de información, que incluye datos sobre las capacidades del browser, datos opcionales para el servidor,...
- El servidor devuelve la respuesta al cliente. Consiste en un código de estado y el tipo de dato MIME de la información de retorno, seguido de la propia información.
- Se cierra la conexión TCP.

Este proceso se repite en cada acceso al servidor HTTP. Por ejemplo, si se recoge un documento HTML en cuyo interior están insertadas cuatro imágenes, el proceso anterior se repite cinco veces, una para el documento HTML y cuatro para las imágenes.

## 5. Restricciones

En este apartado se expondrán todas aquellas restricciones existentes en el ámbito del diseño que condicionan la elección de una u otra alternativa.

### 5.1 Factores de dato

En este punto se enumeran todas aquellas restricciones que son impuestas por el propio proyecto, son las siguientes:

- **Tiempo:** El tiempo de desarrollo del proyecto está estrechamente relacionado a la fecha de entrega de este. Es decir, aproximadamente la primera quincena de junio, dentro del curso académico 2018/2019. Dicho tiempo se ve aún mas limitado teniendo en cuenta las nuevas tecnologías usadas en el proyecto.
- **Recursos humanos:** Para este proyecto se cuenta con un solo desarrollador, que en este caso es el alumno.
- **Recursos hardware y software:** En cuanto al hardware un ordenador personal y en cuanto a software ha sido bienvenido todo el contenido relacionado que se ha encontrado por Internet.
- **Limitaciones de tiempos de ejecución:** Estas limitaciones son casi inexistentes, ya que la aplicación desarrollada es bastante liviana.

### 5.2 Factores estratégicos

A diferencia de las anteriores, estas restricciones son impuestas por el propio desarrollador del proyecto.

- Se utilizará una metodología basada en etapas para realizar este proyecto, basada en el desarrollo de las metodologías ágiles. Entre las ventajas destaca la simplicidad, la temporalización y la potencia de este tipo de metodologías.
- El lenguaje de programación seleccionada para la programación de la interfaz será [TypeScript](#). Se ha seleccionado por ser un lenguaje orientado a objetos de alto nivel, multiplataforma, libre, seguro y recomendado por la comunidad de programadores. Además proporciona todo lo que queremos conseguir con el proyecto, que es simpleza y estabilidad.
- El framework de desarrollo elegido para hacer el desarrollo ha sido Angular en su versión 7.2.15 y será programado a través del entorno de desarrollo de Visual Studio Code todo ello instalado en una máquina con Windows 10.
- El software desarrollado será desplegado en un [Dyno](#) de [Heroku](#) (ligeros contenedor de aplicaciones), lo que nos aporta robustez y la confianza que da una plataforma como Heroku.
- La herramienta para realizar la documentación será el editor Microsoft Office Word, también instalado en una máquina con Windows 10.
- El paradigma de desarrollo del proyecto escogido ha sido el de desarrollo de prototipos, lo que permite ir desarrollando funcionalidades básicas de la aplicación, a esto se le llama MVP (Minimum Viable Product) para después en fases intermedias del proyecto poner en común todas estas características e integrarlas para construir la aplicación.

## 6. Recursos

En este apartado se presentan los recursos con los que se cuenta para la realización del proyecto. Se han dividido en recursos humanos, recursos software y recursos hardware.

- **Recursos humanos:** El equipo de personas con los que cuenta el proyecto está formado por: D. Juan Antonio Sánchez Luque, alumno de segundo curso del Ciclo Formativo de Grado Superior de Desarrollo de Aplicaciones Multiplataforma. El alumno será el encargado de llevar a cabo el desarrollo del proyecto, así como la documentación conveniente del mismo.
- **Recursos Hardware:** Se utilizará el equipo informático del proyectista que posee las siguientes características: Procesador Intel Core i5, 8 GB de RAM y 1 TB de disco duro.
- **Recursos Software:** Sistema Operativo Windows 10 Pro, con las herramientas necesarias para el desarrollo de aplicaciones en Angular, lector de documentos PDF Adobe Acrobat, software de edición de textos Microsoft Word, plataformas webs para realizar la búsqueda bibliográfica y de antecedentes.



# **CAPITULO II: OBJETIVOS**

A continuación, vamos a señalar los diferentes objetivos del proyecto. Para ello distinguiremos entre objetivos formales (aspectos generales sobre el contenido del proyecto) y objetivos operacionales (objetivos que debe cumplir el software desarrollado).

### 1. Objetivos formales

El objetivo fundamental de este Trabajo fin de Ciclo es desarrollar una interfaz que permita a los usuarios consultar los resultados del campeonato del mundo de Fórmula 1, no solo los más recientes si no un set completo de resultados históricos, desde el inicio del campeonato.

Para ello se ha tenido que realizar un estudio de los proveedores de datos existentes, que sean de código abierto y que además sean gratis.

Toda la interfaz estará basada en componentes, es decir, cada parte de la interfaz tendrá su propia lógica interna, aunque de forma transparente al usuario actuará como un conjunto, es decir la misma no se notará desarticulada entre sí.

Como objetivo final, se desarrolla este trabajo para superar el módulo “Proyecto de desarrollo de aplicaciones multiplataforma” de la titulación Técnico Superior en Desarrollo de Aplicaciones Multiplataforma cursado en el C.D.P Zafiro Enseñanza.

### 2. Objetivos operacionales

Incorporar el mayor número de opciones para permitir el uso y manipulación del software. Para ello se ha desarrollado una interfaz intuitiva y fácil de utilizar con elementos rápidos de reconocer y estandarizados como son; botones, zonas dotadas de funcionalidad, vínculos entre conceptos relacionados, manual de ayuda, etc.

Los objetivos operacionales de esta herramienta informática son:

- Crear una interfaz rápida, intuitiva y usable para poder consultar todos los datos de manera eficaz.
- Realizar un análisis estadístico de los datos obtenidos, para poder transformarlos y representarlos de forma que el usuario los pueda comprender con el menor esfuerzo posible.
- Gestionar los datos recibidos a través de peticiones HTTP.
- Asegurar el correcto funcionamiento de la interfaz ante varias casuísticas, ya sean provocadas por el usuario, o por el origen de datos usado.
- Utilizar tecnologías para desarrollar software a medida como Angular

# **CAPITULO III: ESPECIFICACIÓN DE REQUISITOS**

### 1.- DESCRIPCIÓN GENERAL

En el capítulo anterior se ha ofrecido una visión global del problema a resolver, se ha llevado a cabo una descripción del sistema objetivo a desarrollar y se ha definido el tipo de información a manejar y la funcionalidad que el sistema debe aportar. En esta parte se resume dicha información mediante el empleo de una especificación técnica orientada a definir qué debe hacer el sistema. Para la realización de este proyecto se ha considerado más adecuado el uso de un desarrollo incremental, iterando las fases de análisis, diseño y codificación para cada componente independiente del sistema desarrollar. Por último, se realizará la misma secuencia de pasos para la unión de componentes en el sistema global.

### 2.- ESPECIFICACIÓN DE REQUISITOS

En este apartado se van a citar los requisitos que nuestro el desarrollo de nuestro proyecto debe cumplir. Se describirán de forma detallada y concisa para que, cualquier lector con conocimientos o no del tema que trata nuestro proyecto, con solo leer este apartado tenga una visión general y global de qué se pretende con la ejecución de este proyecto.

Nuestro sistema debe cumplir con los siguientes requerimientos:

- Desarrollo de una interfaz sencilla para la representación de datos estadísticos.
- Estandarización del código empleado, siguiendo las últimas especificaciones de uso de los mismos.
- El entorno debe ser fácil de manejar, de forma que al usuario le resulte lo más similar posible a la realidad representada.
- Se debe garantizar el funcionamiento de nuestra aplicación en la mayoría de los navegadores del mercado.
- El sistema debe ser desarrollado para que las futuras actualizaciones y mejoras se puedan realizar sin que el coste de las mismas sea demasiado tedioso o complejo

# **CAPITULO IV: ESPECIFICACIONES DEL SISTEMA**

### 1.- Descripción de la información

En este punto se va a describir la estructura de la información que maneja nuestro software. Como ya se ha comentado anteriormente nuestro proyecto hace uso de la API experimental llamada "[Ergast Developer API](#)". Esta AP se describe como un servicio web experimental que proporciona un registro histórico de datos sobre carreras automovilísticas para fines no comerciales. Este API proporciona datos de Fórmula 1, desde el comienzo del campeonato en 1950.

Aunque la API no sea un desarrollo propio del proyecto, como tenemos acceso al código y se conoce como está hecha se ha visto conveniente explicar el funcionamiento interno de la misma.

#### 1.2 Descripción de los tipos de entidad

A continuación, queda reflejada la estructura interna de la API que estamos consumiendo, esta información la podemos encontrar en [su página web](#). Nosotros hemos hecho una descripción más extensa, explicando todas las entidades y los atributos que tienen cada uno.

**Entidad CIRCUITS:** Representa los diferentes circuitos que están o han estado alguna vez en el campeonato del mundo de Fórmula 1. También se incluyen los circuitos que hayan hecho apariciones esporádicas en el campeonato.

- **circuitId:** Identificador y clave principal de la entidad, es un dato de tipo cadena y representar unívocamente a cada circuito almacenado en la base de datos. (**Varchar 255**)
- **name:** nombre oficial del circuito en cuestión. (**Varchar 255**)
- **location:** lugar de localización del circuito, más concretamente la ciudad donde está. (**Varchar 255**)
- **country:** representa el país donde está ubicado el circuito. (**Varchar 255**)
- **lat:** valor de la latitud de la localización del circuito. (**Float**)
- **lng:** valor de la longitud de la localización del circuito. (**Float**)
- **alt:** altitud del circuito, este dato es bastante importante, porque el rendimiento de los motores depende del mismo. (**Int 11**)
- **url:** dirección URL de la página de Wikipedia del circuito en cuestión. (**Varchar 255**)

**Entidad CONSTRUCTORS\_RESULTS:** Representa los resultados de un constructor en cuestión en una determinada carrera.

- **constructorResultsId:** identificador y clave principal de la entidad, es un dato de tipo entero y sirve para identificar al registro en cuestión. (**Int 11**)
- **raceId:** identificador de la carrera del registro. (**Int 11**)
- **constructorId:** identificador del constructor del registro. (**Int 11**)
- **points:** puntos obtenidos por el constructor en la carrera en cuestión. (**float**)
- **status:** estado de finalización del constructor (por ejemplo, si está descalificado, etc). (**Varchar 255**)

**Entidad CONSTRUCTOR\_STANDINGS:** Representa las posiciones de los constructores de la base de datos.

- **constructorStandingsId**: identificador y clave principal de la entidad, es un dato de tipo entero. **(Int 11)**
- **racelId**: identificador de la carrera del registro. **(Int 11)**
- **constructorId**: identificador del constructor del registro. **(Int 11)**
- **points**: puntos obtenidos por el constructor. **(float)**
- **position**: posición en la que ha quedado el constructor. **(int 11)**
- **positionText**: transformación del campo anterior a tipo texto. **(Varchar 255)**
- **wins**: número de victorias del constructor. **(Int 11)**

**Entidad CONSTRUCTORS:** Representa a cada uno de los constructores de la Fórmula 1.

- **constructorId**: identificador y clave principal de la entidad, es un dato de tipo entero. **(Int 11)**
- **name**: nombre del constructor en cuestión. **(Varchar 255)**
- **nationality**: nacionalidad del constructor. **(Varchar 255)**
- **url**: dirección URL de la página de Wikipedia del constructor en cuestión. **(Varchar 255)**

**Entidad DRIVER\_STANDINGS:** Representa las posiciones de los pilotos de la base de datos.

- **driverStandingsId**: identificador y clave principal de la entidad, es un dato de entero. **(Int 11)**
- **racelId**: identificador de la carrera del registro. **(Int 11)**
- **driverId**: identificador del piloto. **(Int 11)**
- **points**: puntos obtenidos por el constructor. **(float)**
- **position**: posición en la que ha quedado el piloto. **(int 11)**
- **positionText**: transformación del campo anterior a tipo texto. **(Varchar 255)**
- **wins**: número de victorias del piloto. **(Int 11)**

**Entidad DRIVERS:** Representa a cada uno de los pilotos del campeonato.

- **driverId**: identificador y clave principal de la entidad, es un dato de tipo entero. **(Int 11)**
- **number**: número identificativo del piloto, si lo tuviera. **(int 11)**
- **forename**: nombre del piloto. **(Varchar 255)**
- **surname**: apellido del piloto. **(Varchar 255)**
- **dob**: fecha de nacimiento del piloto. **(Date)**
- **nationality**: nacionalidad del piloto. **(Varchar 255)**
- **url**: dirección URL de la página de Wikipedia del piloto en cuestión. **(Varchar 255)**

**Entidad RACES:** Representa cada una de las carreras que hay almacenadas en la base de datos.

- **racelId:** identificador y clave principal de la entidad, es un dato de tipo entero. **(Int 11)**
- **year:** año de la carrera. **(Int 11)**
- **round:** número de carrera respecto al campeonato que se esté desarrollando. **(Int 11)**
- **circuitId:** identificador del circuito donde se corre la carrera. **(Int 11)**
- **name:** nombre oficial de la carrera. **(Varchar 255)**
- **date:** fecha de la carrera. **(Date)**
- **time:** hora de comienzo de la carrera. **(Time)**
- **url:** dirección URL de la página de Wikipedia de la carrera en cuestión. **(Varchar 255)**

**Entidad RESULTS:** Representa los resultados de finalización de cada carrera.

- **resultId:** identificador y clave principal de la entidad, es un dato de tipo entero. **(Int 11)**
- **racelId:** identificador de la carrera del registro. **(Int 11)**
- **driverId:** identificador del piloto. **(Int 11)**
- **constructorId:** identificador del constructor. **(Int 11)**
- **number:** número del piloto. **(Int 11)**
- **grid:** posición de salida. **(Int 11)**
- **position:** posición en la que ha quedado el piloto. **(int 11)**
- **positionText:** transformación del campo anterior a tipo texto. **(Varchar 255)**
- **points:** puntos obtenidos por el piloto. **(Float)**
- **laps:** número de vueltas. **(Int 11)**
- **time:** tiempo que se ha tardado en completar la carrera. **(Varchar 255)**
- **milliseconds:** tiempo en milisegundos que se ha tardado en completar la carrera. **(Int 11)**
- **fastestLap:** numero vuelta rápida del piloto. **(Int 11)**
- **statusId:** estado de finalización del piloto. **(Int 11)**

## 2.- Representación del flujo de la información

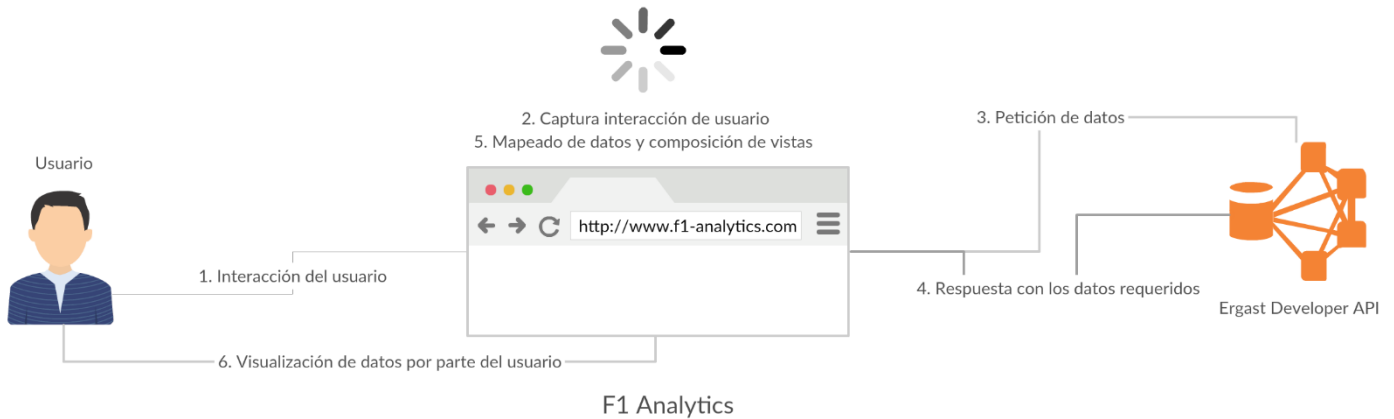
En este apartado se representa el flujo de información de nuestra aplicación, es decir, cómo la información descrita anteriormente se moverá por el sistema en relación a los eventos que en el mismo se producen.

Utilizaremos para ello los llamados diagramas de caso de uso. Un caso de uso representa una "descripción de un conjunto de acciones que ejecuta un sistema para producir un resultado observable de valor para un actor" [BOOC1999]. Básicamente indican qué va a hacer el sistema pero sin profundizar en el cómo lo va a hacer.

Los diagramas de caso de uso se utilizan para modelar el contexto del propio sistema, identificando en el proceso a los actores principales, modelando los requisitos funcionales del mismo desde los aspectos más genéricos hasta los más concretos que van a representar la funcionalidad interna de los anteriores. En esta documentación seguiremos un proceso de especificación progresivo, partiendo del contexto general del sistema y los subsistemas, refinando hasta casos de uso más concretos.



En los siguientes esquemas se modelan el contexto del sistema, el flujo de información, los actores principales, las funcionalidades aplicadas y los resultados de las acciones. Este esquema nos ayuda a comprender el funcionamiento general de la aplicación desarrollada en este proyecto.



### 3.- Especificaciones del software utilizado para el desarrollo del proyecto

Como ya se ha comentado a lo largo de los puntos y capítulos anteriores de este documento, para el desarrollo de este proyecto se ha utilizado diferentes tipos de software, desde sistemas operativos, pasando por software de desarrollo y finalizando con software de edición de textos. Vamos a describir las características de todo este software, cómo se ha usado, instalado y la utilidad que tiene para nuestro trabajo.

#### 3.1 Framework Angular 7 y TypeScript

El framework de desarrollo FrontEnd Angular ha sido el utilizado para crear la totalidad de la interfaz y toda su lógica; servicios, componentes, vistas, rutas, etc. Más concretamente se ha utilizado la versión 7.1.4. Como especificamos en puntos anteriores para usar este framework se necesita tener instalado NodeJS y el gestor de paquetes npm.

Hemos realizado un estudio más extenso de este framework en el apartado 1.3 Estudio del framework Angular

#### 3.2 Framework Electron para aplicaciones de escritorio

Debido a que la idea es que nuestra aplicación sea lo más accesible posible y que desde su concepción se planteó con un diseño responsive se ha decidido llevarla a todas las plataformas posibles. En este afán, se ha decidido hacer una aplicación de escritorio con el framework multiplataforma Electron. A continuación, se detallan un poco más las características del framework:

Como hemos comentado, Electron es un framework para JavaScript que permite el desarrollo de aplicaciones enriquecidas de escritorio mediante el uso de tecnologías web. Esta desarrollado por GitHub (lo que garantiza revisiones constantes), es de código abierto y multiplataforma (funciona bajo Linux, Mac y Windows).

Electron está basado en io.js y funciona bajo un subconjunto mínimo de librerías de Chromium. Además proporciona acceso a APIs nativas enriquecidas mediante el motor de Chromium, el cual es controlado mediante JavaScript. Esto no significa que enlace a librerías gráficas del sistema operativo, ya que su GUI se desarrolla mediante HTML, pero podemos aprovechar ciertas características de las librerías nativas mediante Chromium. Compañías como Microsoft, Facebook, Slack y Docker utilizan esta plataforma.

### 3.3 Preprocesador CSS: SASS

Un preprocesador de CSS es una herramienta que nos permite escribir pseudo-código CSS que luego será convertido a CSS real. Ese pseudo-código se conforma de variables, condiciones, bucles o funciones. Podríamos decir que tenemos un lenguaje de programación que genera CSS.

El objetivo de estos preprocesadores es tener un código más sencillo de mantener y editar. Los preprocesadores incluyen características tales como variables, funciones, mixins, anidación o modularidad.

SASS se desarrolla en 2007 (si desde 2007 hay pre-procesadores, muchos los descubrimos años después) , SASS ha convertido en el pre-procesador mas utilizado – y por una buena razón. A pesar de que ha existido durante casi una década los desarrolladores detrás SASS siguen ampliar su funcionalidad .

SASS sobresale al utilizarse con frameworks como Compass y Bourbon , ya que estos anexan la funcionalidad que quizá SASS debería tener de forma nativa.

#### Sintaxis

SASS permite el uso de dos sintaxis diferentes para crear sus archivos. La primera, conocida como SCSS (del inglés, Sassy CSS) .

Además, SCSS es capaz de entender la mayoría de hacks de CSS y la sintaxis específica de los navegadores, como por ejemplo la vieja sintaxis filter de Internet Explorer. Obviamente esta sintaxis básica de CSS se ha mejorado con las características de Sass.

Los archivos creados con esta sintaxis utilizan la extensión. scss.

```
// Variable
$primary-color: hotpink;

// Mixin
@mixin border-radius($radius) {
    -webkit-border-radius: $radius;
    -moz-border-radius: $radius;
    border-radius: $radius;
}

.my-element {
    color: $primary-color;
    width: 100%;
    overflow: hidden;
}

.my-other-element {
    @include border-radius(5px);
}
```

La segunda sintaxis, conocida como «sintaxis indentada» o simplemente «sintaxis sass» permite escribir los estilos CSS de manera más concisa. En este caso, el anidamiento de selectores se indica con tabulaciones en vez de con llaves y las propiedades se separan con saltos de línea en vez de con puntos y coma.

```
// Variable
!primary-color= hotpink

// Mixin
=border-radius(!radius)
  -webkit-border-radius= !radius
  -moz-border-radius= !radius
  border-radius= !radius

.my-element
  color= !primary-color
  width= 100%
  overflow= hidden

.my-other-element
  +border-radius(5px)
```

Los archivos creados con esta segunda sintaxis utilizan la extensión .sass

Algunos diseñadores consideran que esta segunda sintaxis es más sencilla de leer y más rápida de escribir que SCSS. En cualquier caso, las dos sintaxis tienen exactamente las mismas funcionalidades.

Una de las ventajas de SASS es que los archivos creados con una sintaxis pueden importar cualquier archivo creado con la otra sintaxis. Además, dispones de una utilidad para la línea de comandos que te permite convertir una sintaxis en otra.

### Características de Sass

- 100% compatible con CSS3
- Permite el uso de variables, anidamiento de estilos y mixins.
- Incluye numerosas funciones para manipular con facilidad colores y otros valores.
- Permite el uso de elementos básicos de programación como las directivas de control y las librerías.
- Genera archivos CSS bien formateados y permite configurar su formato.
- Integración con Firebug gracias a la extensión FireSass.

### 3.4.- Funcionamiento de nuestro software cuando un usuario hace una petición

Como el objetivo principal de nuestro proyecto es hacer una herramienta capaz de aunar la mayor cantidad de resultados del campeonato mundial de Fórmula 1 y sus resultados históricos, vamos a explicar de manera interna como se recogen estos datos, como son mapeados por el software y como son mostrados al usuario, desde que el usuario hace una petición, hasta que los datos aparecen delante de sus narices. Hay que indicar que todo se realiza mediante código de programación desde la interfaz principal desarrollada en Angular sin que el usuario sea consciente de las acciones que se están realizando. En el anexo de este documento “Manual de usuario”, se exponen más detalladamente los pasos que debe dar el usuario de la aplicación para conseguir este y otros objetivos.

Para llevar a cabo la explicación, dado que nuestra aplicación está separada en distintos componentes interrelacionados se irá explicado componente a componente como funciona:

- Apartado principal: Este apartado se puede dividir a su vez en varias partes bien diferenciadas. Justo debajo del [“hero”](#) lo primero que aparece en la interfaz son los resultados del gran premio anterior, para ello se debe hacer un cálculo de la fecha actual y compararla con la de todos los grandes premios de la temporada actual, una vez que se sabe cuál es el GP anterior se hace una petición a la API para obtener el resumen de los resultados de ese gran premio.

Otra parte principal de este componente principal es el apartado del próximo Gran Premio, en este pequeño apartado se comprueba algo parecido al punto anterior, pero comprobando cual es el próximo gran premio, además se inserta una cuenta regresiva para saber cuánto tiempo queda para el próximo Gran Premio.

En el siguiente apartado llamado “Driver of the day” se selecciona un piloto de todos los disponibles en la API, esta selección se hace dependiendo del día de la semana, del mes y en el año en el que estemos. De este piloto se obtiene el número de campeonatos, de victorias, de pódiums y poles y una pequeña descripción sacada de su página de Wikipedia.

Otra sección es la de últimas noticias donde se recaban noticias de la API [“News API”](#), para ello se acotan las noticias a la última semana y solo sobre temas relacionados con la fórmula 1.

- Apartado circuitos: En este apartado aparecen listados todos los circuitos que alguna vez han participado en una carrera puntuable para el campeonato del mundo de Fórmula 1. Estos circuitos aparecen ordenados y agrupados por orden alfabético. Para ello se hace una consulta con todos los circuitos del campeonato y después se mapean para poder agruparlos por sus iniciales.
- Apartado circuitos (detalle): Si el usuario pulsa sobre cualquier circuito del apartado anterior es llevado a otra pantalla con el detalle de ese circuito. Para ello se pasa por parámetro en la URL la id del circuito del que se quiere ver el detalle. Una vez redirigido nos aparece un apartado con la información general del circuito. En esta información

encontramos una foto del circuito (obtenida desde su página de Wikipedia), su localización (junto con la bandera de su país) y una breve descripción del mismo. Si el circuito seleccionado ha albergado más de 5 grandes premios de Fórmula 1 se generará un texto de forma automática con las estadísticas medias de ese circuito, estas estadísticas son básicamente son; el número de grandes premios albergados en este circuito, junto con los años de inicio y fin, además se refleja el piloto más exitoso en el circuito junto con el número de victorias, también se refleja quien ha sido el constructor más exitoso junto con el número de victorias.

Además, en el apartado de detalle se muestra la ubicación del circuito, haciendo uso de Google Maps, este apartado es realmente útil si se tiene planificado hacer algún viaje y ver el circuito en cuestión.

Por último, nos aparece una tabla con todos los eventos disputados en el circuito, con varios datos y detalles; por cada Gran Premio disputado aparece el nombre oficial del evento, la fecha, el ganador (bandera, nombre, equipo y el tiempo de finalización) y el piloto que se llevó la pole position (incluyendo su bandera y su equipo).

- **Apartado constructores:** En este apartado nada más entrar se puede ver la lista de constructores de la temporada actual, además se muestran los resultados del campeonato de constructores de la temporada actual. También se ha insertado un video explicativo de la temporada actual, donde se detallan aspectos más técnicos de cada uno de los constructores. A continuación, se listan todos los constructores que han participado alguna vez en el campeonato de constructores del mundial de Fórmula 1. De la misma forma que en el apartado de pilotos, los constructores se encuentran agrupados por orden alfabético.
- **Apartado constructores (detalle):** Este apartado está dedicado a mostrar detalles específicos del constructor seleccionado. Lo primero que encontramos es información básica del constructor. Esta información básica es la nacionalidad, y los pilotos actuales del equipo. Justo debajo tenemos la sección de estadísticas, estas estadísticas constan de los siguientes datos: número de victorias, numero de pole positions, numero de temporadas disputadas, numero de carreras disputadas, numero de campeonatos de pilotos ganados y numero de campeonatos de constructores ganados. Para obtener estas estadísticas se han tenido que hacer un conjunto de peticiones a la API y cuando se obtiene respuesta de todas las peticiones se hace un mapeado global de los datos, teniendo una gran dificultad técnica.

Después de este apartado encontramos el apartado de información general del constructor, aquí encontramos una foto representativa del equipo (obtenidas de su página de Wikipedia, a través de la API) junto con una breve descripción del mismo. Si no es posible encontrar ninguna foto del constructor se colocará una por defecto, lo mismo se hace con la descripción.

Después de esta información básica tenemos una serie de gráficas estadísticas, para ello se ha realizado un complejo mapeado de datos ya que la API no provee los datos completamente listos para la elaboración de estas gráficas.

También son obtenidos de la API los pilotos que han conducido para el constructor en cuestión, para ello se ha tenido que solicitar a la API datos de todas las temporadas que ha disputado el constructor y nuestro software ha tenido que filtrar y obtener los datos de los pilotos que han conducido para el constructor.

Otro apartado importante es el de “campeonatos de pilotos” en el que se refleja todos los campeonatos de pilotos que ha ganado el constructor, aquí se detalla el año del título y el piloto que lo consiguió.

Justo debajo tenemos una lista con los campeonatos de constructores que se ha llevado el equipo, junto a un enlace que nos llevará al detalle de esa temporada.

A continuación, y por último tenemos dos apartados aparentemente iguales pero analizados en detalle se puede comprobar que el primero de ellos hace referencia al listado de victorias del constructor y el segundo apartado a la lista de las poles positions que ha logrado. Este apartado quizás sea el más complejo de nuestro software ya que se han tenido que manejar una gran cantidad de datos, y conseguir una relación entre ellos ha sido bastante complicado.

- **Apartado pilotos:** Este apartado está destinado a ofrecer de manera clara los datos generales de todos los pilotos de la Fórmula 1. En primer lugar encontramos los pilotos de la temporada actual del campeonato y sus posiciones y puntos del campeonato de pilotos. Pulsando encima de cada uno de ellos o encima de su constructor la aplicación nos redirigirá al apartado correspondiente, ya sea al apartado de pilotos o al apartado de constructores.

Justo debajo encontramos una lista con todos los pilotos que alguna vez han participado en una Gran Premio oficial, ordenados y agrupados alfabéticamente. Básicamente cumple la función de índice, para poder elegir cualquier piloto e irse a la descripción detallada del piloto.

- **Apartado pilotos (detalle):** Una vez seleccionado un piloto en la vista anterior, el flujo de funcionamiento de la aplicación nos lleva a este componente. En el mismo vemos un pequeño apartado con información acerca del piloto, esta información está compuesta por su nombre completo, nacionalidad, lugar de nacimiento (obtenido de la API de Wikipedia), fotografía (obtenida de la API de Wikipedia) y una pequeña descripción (también obtenida de Wikipedia). Cabe mencionar que si alguno de los datos que se obtienen de Wikipedia no se recaban correctamente la aplicación colocará unos datos por defecto genéricos, además de un mensaje diciendo que ha habido un error en la carga de los datos.

Este componente también tiene una sección de estadísticas, que como en otros apartados de la aplicación se han tenido que obtener datos de varios puntos de la API, por lo que el mapeado de estas estadísticas tiene una gran dificultad técnica para nuestro software.

Justo debajo tenemos una serie de gráficas estadísticas sobre el rendimiento del piloto, las principales son: evolución de resultados por temporadas en el campeonato de pilotos, un histórico de las posiciones finales en carrera a lo largo de las temporadas y un histórico de las posiciones en parrilla a lo largo de las temporadas.

Finalmente se puede ver una pequeña tabla con los equipos en los que ha estado el piloto y los compañeros de equipo que ha tenido a lo largo de las temporadas. Estos

datos se han obtenido recorriendo todas las temporadas del piloto y viendo en que equipos ha estado, más tarde se ha comprobado que pilotos ha tenido ese equipo en las temporadas que ha estado el piloto en ese equipo.

- Apartado temporadas: Este apartado quizás sea el más simple de todos ya que muestra una lista de todas las temporadas de la Fórmula 1, junto con la cuenta del número de carreras que ha tenido cada temporada. Al pulsar sobre cualquier temporada nos llevaría al componente de detalle.
- Apartado temporadas (detalle): Cuando se entra en este componente y se tiene una temporada seleccionada se muestran los siguientes apartados; lo primero que vemos es los resultados del campeonato de pilotos en esa temporada (todos los pilotos, junto con su constructor y los puntos obtenidos en esa temporada).

A continuación encontramos un gráfico con la distribución de puntos por cada equipo de la temporada, en el campeonato de pilotos. Este gráfico es de tipo queso, y para confeccionarlo se ha tenido que obtener el porcentaje de puntos obtenidos por cada equipo (cosa que no provee la API directamente) y la cantidad total de puntos repartidos esa temporada (tampoco es proveído por la API).

Por último encontramos una lista con las carreras de la temporada y sus resultados, es decir, el piloto ganador, el constructor ganador, las vueltas y el tiempo de carrera.



# CAPITULO V: PRUEBAS

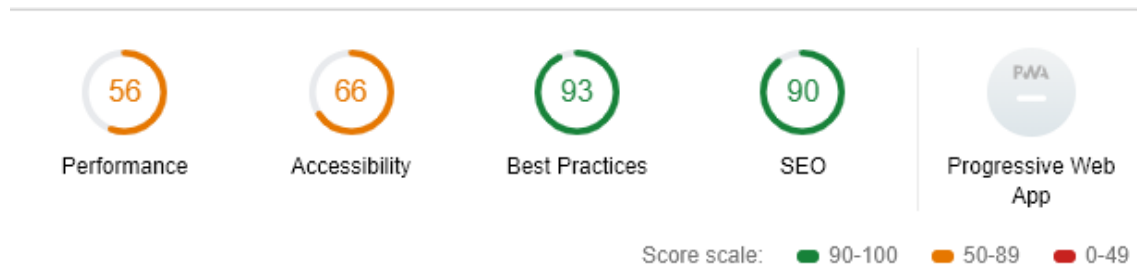
# 1. Pruebas específicas para comprobar el funcionamiento del software

## Pruebas en la aplicación web

Desde la aplicación web se han realizado todo tipo de pruebas para asegurar el máximo rendimiento de la misma. En ella se controla si el servicio de origen de datos está caído, tarda mucho en responder o simplemente la petición requerida no se puede satisfacer. También hay un control de errores en la parte respectiva a Wikipedia, ya que, aunque la API de Wikipedia está caracterizada por tener una alta disponibilidad, nunca se sabe cuándo se puede producir un fallo en el servicio.

En cuanto a las pruebas de la aplicación web en dispositivos móviles se puede afirmar que la interfaz se adapta perfectamente a todo tipo de dispositivos. Esto se puede afirmar ya que ha sido probada gracias a las herramientas de desarrollo de Google.

Como prueba adicional la aplicación ha sido auditada con la reconocida herramienta de Google [“LightHouse”](#) obteniendo los siguientes resultados:



Poniendo en valor estos resultados creemos que son unos resultados bastante buenos, si tenemos en cuenta el factor del tiempo de desarrollo del proyecto y el desconocimiento en las tecnologías empleadas en la realización del mismo.

# **CAPITULO VI: CONCLUSIONES**

El desarrollo de este proyecto ha tenido como objetivo final implementar una herramienta que permita a cualquier aficionado de la Fórmula 1, sin necesidad de tener demasiados conocimientos sobre el deporte, poder consultar los datos y las estadísticas de sus pilotos u equipos favoritos, sin necesidad de tener que evitar decenas de páginas donde lo principal no es lo que precisamente está buscando el usuario. La finalidad del proyecto es establecer esta herramienta como una referencia en cuanto a datos estadísticos de Fórmula 1.

En cuanto al uso de esta herramienta, podemos decir que es bastante simple ya que así fue pensada desde sus orígenes de desarrollo. También queremos puntualizar un aspecto importante de la misma, y es su adaptabilidad a cualquier dispositivo.

Debido a factores tales como el tiempo de desarrollo del proyecto y al poco conocimiento en las tecnologías trabajadas, se han dejado varias cosas en el tintero, que sin duda, en un futuro se pueden llegar a implementar de forma más o menos fácil en la aplicación, así que dejamos constancia de las mejoras propuestas:

- Teniendo los datos casi a tiempo real, sería de fácil implementación una liga fantasy, en la cual distintos usuarios formen grupos y elijan a un constructor o varios pilotos para un Gran Premio y que tras la carrera vean como han quedado y hacer una clasificación con los puntos obtenidos entre pilotos y constructor y crear una especie de competición con los usuarios.
- Añadir datos de la Fórmula E, cosa que implementó hace poco la API que usamos como origen de datos, pero que por falta de tiempo no se ha podido realizar.
- Implementación de un foro, donde los usuarios podrán comentar las carreras en directo.

## BIBLIOGRAFÍA

- La telemetría en la Fórmula 1. Fuente: <http://drive-smart.com/es/blog/2015/07/22/la-telemetria-en-la-f1-clave-para-el-ingeniero-trampa-para-el-piloto/>
- Estudio del estándar API REST: <https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos>
- Estudio protocolo HTTP: <http://neo.lcc.uma.es/evirtual/cdd/tutorial/aplicacion/http.html>
- Estudio de los paradigmas Frontend y Backend: <https://devcode.la/blog/frontend-y-backend/>
- Estudio del framework Angular: <https://victorroblesweb.es/2017/08/05/que-es-angular-y-para-que-sirve/>  
<https://www.campusmvp.es/recursos/post/las-5-principales-ventajas-de-usar-angular-para-crear-aplicaciones-web.aspx>

# **ANEXO I: MANUAL DE USUARIO**

# 1. Acceso a la aplicación

El acceso a la aplicación es tan sencillo como colocar la dirección URL de la misma en el navegador (en nuestro caso <http://localhost:4200/>), una vez haya cargado completamente la aplicación nos mostrará una vista como ésta:



## 1.1 Página principal

Si nos desplazamos por la página principal se pueden ver varias zonas bien diferenciadas:



En esta primera sección denominada “Actualidad” la aplicación nos ofrece varios datos, en la parte izquierda tenemos la información del último Gran Premio disputado. Desglosaremos la información de la imagen anterior; se obtiene el nombre oficial del Gran Premio (Canadian Grand Prix), el circuito donde se disputó (Circuit Gilles Villeneuve), después en pequeño se ve el número de carrera (en este caso 7) y la fecha.

En el mismo apartado tenemos un resumen de los resultados de esa carrera, donde se ven los 3 primeros clasificados. De estos clasificados se muestran la posición, su bandera identificando su nacionalidad y su abreviatura oficial.



Como se ve en la imagen anterior, si pulsamos encima de los nombres de los pilotos, nos llevaría a la página de detalle del piloto en cuestión. La página de detalle de los pilotos se trata en profundidad más adelante en el manual.

Si nos centramos en la sección de la derecha nos encontramos una pequeña presentación de la página en general junto con una cuenta atrás. Esta cuenta atrás hace referencia al tiempo que queda para el próximo Gran Premio del campeonato. Este contador tiene una precisión hasta los segundos y está programado para que se adapte a la franja horaria del usuario.





En esta pequeña sección también aparece el nombre del siguiente evento, en este caso el French Grand Prix.


Si nos seguimos desplazando hacia abajo por la página nos topamos con la sección llamada "Driver of the day" en esta sección aparecerá cada día, de forma aleatoria, un piloto de Fórmula 1 (no tiene por qué ser actual, puede salir cualquiera de la historia del campeonato), además aparecerán algunos datos relativos al mismo como puede ser su nacionalidad, el número de campeonatos ganados, el número de victorias, el número de podios y el número de poles.




### DRIVER OF THE DAY




**Gianmaria Bruni** 




0 campeonatos



0 podiums



0 victorias



0 poles


Gianmaria "Gimmi" Bruni es un piloto italiano de automovilismo de velocidad, que se ha destacado en gran turismo como piloto oficial de Ferrari. Obtuvo títulos en el Campeonato FIA GT 2008, la Le Mans Series 2011, el Open Internacional de GT 2012, y el Campeonato Mundial de Resistencia 2013, además de un subcampeonato en la American Le Mans Series 2010, así como victorias de clase en las 24 Horas de Le Mans 2008, 2012 y 2014, las 12 Horas de Sebring 2010, Petit Le Mans 2011 y las 24 Horas de Spa 2008, entre otras competiciones de gran turismo.

También tenemos una breve descripción del piloto, donde aclarará al usuario quien es realmente.

Por último, tenemos una sección de noticias, básicamente esto permite al usuario ver las noticias directamente dentro de la aplicación, sin tener que recurrir a páginas de terceros para estar totalmente informados.


### Últimas Noticias

#### Horarios y dónde ver el Gran Premio de Canadá de F1 2019




"Este fin de semana se celebrará el séptimo gran premio de la temporada correspondiente al Mundial de F1. Hasta el momento, el dominio de Mercedes..."

#### Leclerc no da por perdido el título: "No me rendiré hasta que esté decidido"




"Charles Leclerc es quinto en el Mundial de F1 con sólo 57 puntos. Está a 80 de Lewis Hamilton (137). Una barbaridad teniendo en cuenta que sólo se..."

#### La FIA invitará también a los pilotos para discutir las normas de 2021



"El problema de la F1 es la democracia. Hay que poner de acuerdo a tanta gente que al final no se puede cambiar nada", decía Bernie..."

#### "La F1 es un poco vergonzosa, debería ser tan emocionante como MotoGP"




"George Russell alberga la esperanza de que las carreras de la F1 sean tan emocionantes como las de MotoGP después de que haya una nueva..."


Cuando se pulse encima de cada noticia nos llevará a la página en cuestión donde esté publicada la noticia, esta página se abrirá en una nueva pestaña para que el usuario no pierda la noción de donde estaba en nuestra aplicación.

Por último, tenemos una parte que es común a todos los apartados y esta es el denominado "footer". En él tenemos datos como la información legal de la página, enlaces a las redes sociales de la aplicación y una serie de accesos directos para poder navegar entre las distintas páginas.

Este sitio web no es oficial y no está asociado en ningún modo con el grupo de compañías de la Fórmula 1. F1, FORMULA ONE, FORMULA 1, FIA FORMULA ONE WORLD CHAMPIONSHIP, GRAND PRIX y marcas relacionadas son marcas registradas de Formula One Licensing BV.

#### Redes Sociales

 [Síguenos en Facebook](#)

 [Síguenos en Twitter](#)

#### Accesos directos

[Circuitos](#)  
[Constructores](#)  
[Pilotos](#)  
[Temporadas](#)  
[Ránkings](#)

© Juan Antonio Sánchez Luque Todos los derechos reservados

F1 Analytics: Análisis de resultados de Fórmula 1

40

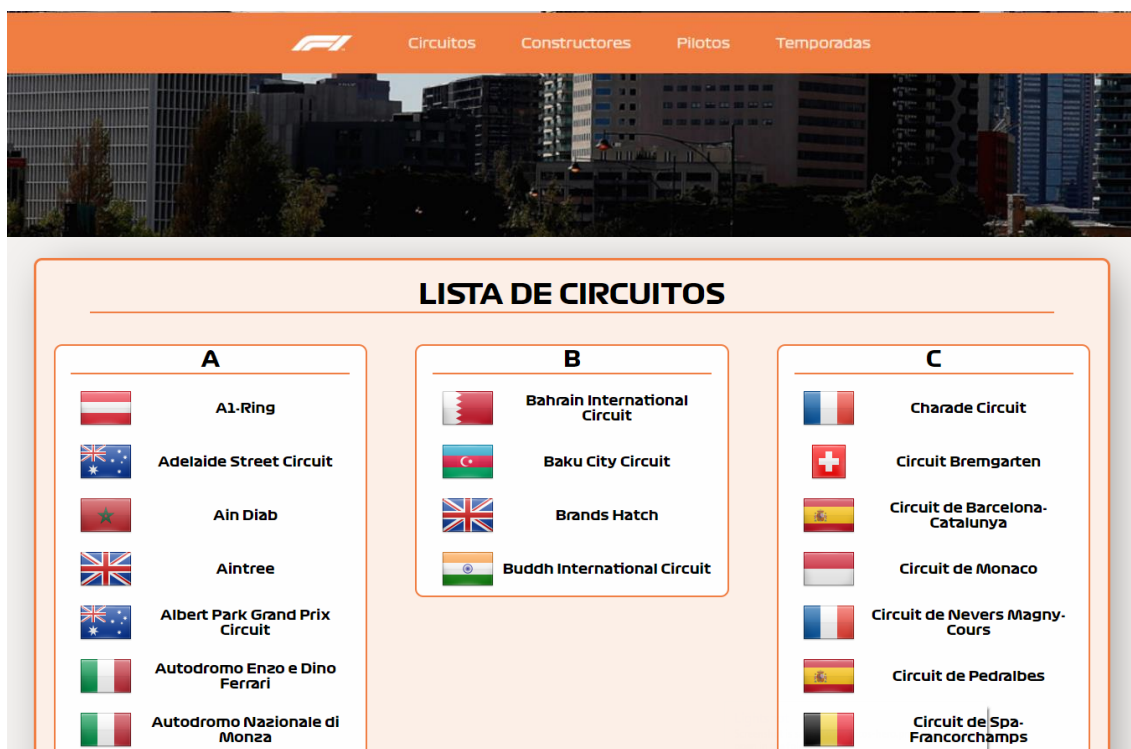


### 1.2 Apartado circuitos

Para acceder a este apartado de la aplicación, basta con hacer click en la barra de navegación superior o en el footer donde pone “Circuitos” como se puede ver en las siguientes imágenes:



Una vez que se pulse en uno de los dos enlaces se nos abrirá la página de circuitos, donde en una primera instancia nos aparecerá su respectivo título de sección junto con una lista de circuitos. Esta lista estará conformada por los circuitos que hayan albergado aunque sea una única carrera puntuable para el campeonato de Fórmula 1.



## Anexo I: Manual de usuario

Como se ve en la imagen anterior los circuitos se encuentran agrupados alfabéticamente lo que hace la tarea de buscar cualquier circuito mucho más fácil. Por cada circuito que aparece en la lista se puede ver que aparece una bandera de un país, esta bandera hace referencia al país donde está ubicado el circuito, por poner un ejemplo se ve de un simple vistazo que el circuito “A1- Ring” está ubicado en una ciudad de Austria.

Si se pulsa sobre cualquier circuito de la lista la aplicación nos llevaría al apartado específico de ese circuito. Para hacer la demostración en este manual se ha simulado que el usuario ha pulsado sobre el circuito llamado “Circuit de Barcelona-Catalunya”.



Como vemos en la imagen anterior el encabezado de la página cambiaría, mostrando el nombre del circuito seleccionado. Justo después del encabezado nos aparecería un apartado con información general del circuito, como se puede ver en la siguiente fotografía:

### Circuit de Barcelona-Catalunya

**Montmeló, Spain**



El Circuit de Barcelona-Catalunya en Montmeló ha albergado 29 Grandes Premios entre 1991 y 2019.

Con 6 victorias, Michael Schumacher es el piloto más exitoso en este circuito. El mejor constructor ha sido Ferrari con un total de 8 victorias en este trazado.

El Circuito de Barcelona-Cataluña, también conocido como Circuito de Montmeló o Circuito de Barcelona, es un autódromo situado en las poblaciones de Montmeló y Granollers, en la provincia de Barcelona, comunidad autónoma de Cataluña, España. Inaugurado en el año 1991, tiene un aforo de 131.000 espectadores, y acoge diversas competiciones entre las que destacan el Gran Premio de España de Fórmula 1 y el Gran Premio de Cataluña de Motociclismo del Campeonato Mundial de Motociclismo. También es uno de los circuitos más utilizados para los ensayos de los diferentes equipos europeos durante el receso de invierno.

[Ver más](#)

Analizando en detalle esta información sobre el circuito seleccionado se muestra una imagen representativa del circuito, ya sea una imagen del trazado en sí o del logo representativo del circuito. Como no podría ser de otra manera aparece el nombre del circuito junto con la bandera identificativa del país. Justo debajo aparece también una descripción del circuito. Si aún así el usuario desea ver más información acerca del circuito, si pulsa sobre el botón “Ver más” le llevará a la página de Wikipedia del circuito en cuestión.

## Anexo I: Manual de usuario

Si nos seguimos desplazando por la sección lo siguiente que nos encontramos es un apartado bastante útil para localizar el circuito. La sección es llamada “Ubicación” y como podemos ver en la imagen que hay justo debajo contiene un mapa situando exactamente el circuito en el mapa.



El usuario podrá interactuar con el mapa de varias formas, la primera es pulsando sobre el botón “Ampliar el mapa” situado en la esquina superior izquierda del mismo. Este botón ampliará el mapa y nos llevará directamente a Google Maps, donde se podrán usar todas las utilidades de la famosa herramienta de Google.

Otra opción es poder cambiar el tipo de mapa, así si se pulsa en el icono cuadrado de la esquina inferior izquierda podrá activarse el modo satélite del mapa.

En la esquina inferior derecha del mapa tenemos un par de botones donde se podrá ampliar o alejar el mapa a gusto del usuario.

Por último, pero no menos importante tenemos el apartado de eventos, como se puede ver en la siguiente imagen este apartado está destinado a reflejar todos los eventos albergados por el circuito en cuestión, junto con el piloto ganador y el que se llevó la pole position.

Eventos									
#	Evento		Ganador				Pole Position		
1	1991 Spanish Grand Prix	29 Septiembre		N. Mansell	Williams	1:38:41.541		G. Berger	McLaren
2	1992 Spanish Grand Prix	03 Mayo		N. Mansell	Williams	1:56:10.674		N. Mansell	Williams
3	1993 Spanish Grand Prix	09 Mayo		A. Prost	Williams	1:32:27.685		A. Prost	Williams
4	1994 Spanish Grand Prix	29 Mayo		D. Hill	Williams	1:36:14.3		M. Schumacher	Benetton
5	1995 Spanish Grand Prix	14 Mayo		M. Schumacher	Benetton	1:34:20.507		M. Schumacher	Benetton
6	1996 Spanish Grand Prix	02 Junio		M. Schumacher	Ferrari	1:59:49.307		D. Hill	Williams
7	1997 Spanish Grand Prix	25 Mayo		J. Villeneuve	Williams	1:30:35.896		J. Villeneuve	Williams
8	1998 Spanish Grand Prix	10 Mayo		M. Häkkinen	McLaren	1:33:38.3		M. Häkkinen	McLaren
9	1999 Spanish Grand Prix	30 Mayo		M. Häkkinen	McLaren	1:34:13.665		M. Häkkinen	McLaren

Si nos fijamos bien en la imagen anterior vemos como la primera fila aparece seleccionada y es que si queremos obtener más información acerca del piloto o constructor que haya ganado el Gran Premio o se haya llevado la pole podemos dirigirnos a la página del piloto o constructor pulsando encima del nombre de uno u otro respectivamente.

Como todas las secciones de la aplicación esta también tiene un footer, pero como en todas las secciones es el mismo y se ha explicado anteriormente no se ve necesario volver a explicarlo en esta sección.

### 1.3 Apartado constructores

Para acceder a este apartado de la aplicación, basta con hacer click en la barra de navegación superior o en el footer donde pone “Circuitos” como se puede ver en las siguientes imágenes:



Una vez que se pulse en uno de los dos enlaces se nos abrirá la página de constructores, donde en una primera instancia nos aparecerá su respectivo título de sección. Justo debajo de la cabecera de la página nos encontramos con dos apartados:



### CONSTRUCTORES TEMPORADA 2019

#	Constructor	Puntos
1	Mercedes	295
2	Ferrari	172
3	Red Bull	124
4	McLaren	30
5	Renault	28
6	Racing Point	19
7	Toro Rosso	17
8	Haas F1 Team	16
9	Alfa Romeo	13
10	Williams	0

### F1 2019: Equipos y pilotos

## FERRARI

SCUDERIA FERRARI

DEBUT 1950

BASE MARANELLO, ITALY

JEFE DE EQUIPO MATTHIA BINOTTO

JEFE TÉCNICO

MOTOR FERRARI

TÍTULOS MUNDIALES 16

MEJOR RESULTADO EN CARRERA 1 (X236)

POLES 215

**SEBASTIAN VETTEL** 5

FECHA DE NACIMIENTO 03/07/1987

GRANDES PREMIOS 220

MEJOR RESULTADO 33

POLES 111

**CHARLES LECLERC** 16

FECHA DE NACIMIENTO 16/10/1997

GRANDES PREMIOS 21

MEJOR RESULTADO 6 (X1)

POLES 0

FERRARI SF90

Tal como se puede ver en la imagen superior, a la izquierda tenemos una tabla con los constructores y sus resultados en la temporada actual de Fórmula 1. Como se especifica estas posiciones se refieren al Campeonato de Constructores. En este caso si se pulsa encima del nombre de cualquier equipo de esta pequeña lista la aplicación nos llevaría a la página dedicada completamente a ese constructor.

A la derecha de la tabla tenemos un video introductorio, donde se explica de una forma más detallada todas las características de los constructores de la temporada actual. En el video aparecen características como el nombre técnico del monoplaza, el jefe técnico del equipo, el motor que lleva y algunas estadísticas sencillas.

Si nos seguimos desplazando hacia abajo nos encontramos con una gran lista de constructores. En esta lista aparecen todos los constructores que hayan participado en al menos una carrera puntuable para el campeonato de Fórmula 1.

### LISTA DE CONSTRUCTORES

#### A

- Adams
- AFM
- AGS
- Alfa Romeo
- Alita
- Amon
- Andrea Moda

#### B

- BAR
- Behra-Porsche
- Bellasi
- Benetton
- BMW
- BMW Sauber
- Boro

#### C

- Caterham
- Christensen
- Cisitalia
- Coloni
- Connaught
- Connew
- Cooper

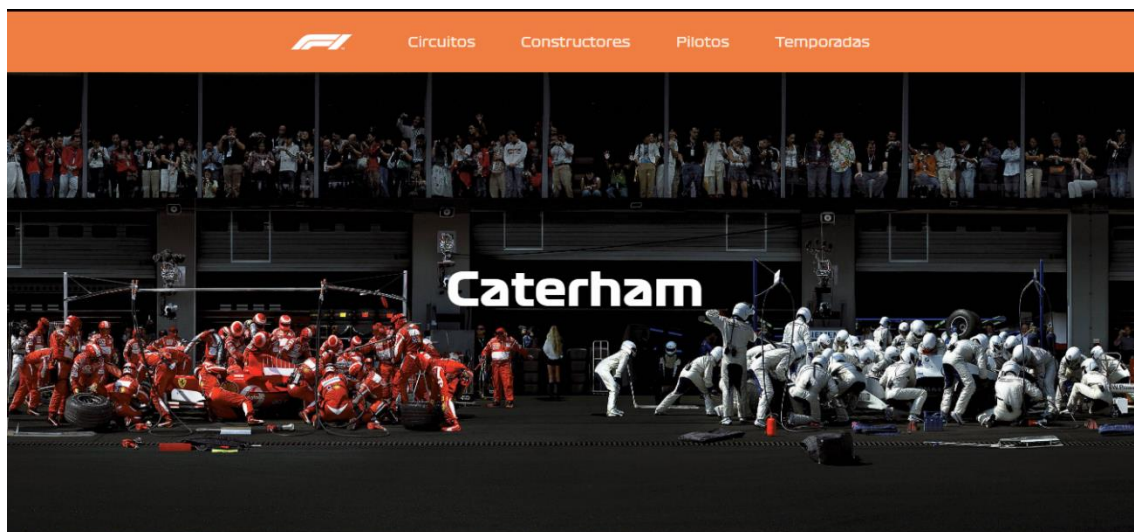
localhost4200/constructores/caterham

Como se puede ver, en esta lista ocurre como en las anteriores, los constructores se encuentran ordenados y agrupados por orden alfabético, lo que hace la búsqueda más sencilla. Al igual que en la sección de circuitos si se pulsa encima de cualquier constructor la aplicación nos llevará a la página específica del constructor seleccionado.



## Anexo I: Manual de usuario

Para realizar este manual se ha simulado que el usuario ha seleccionado ver el detalle del equipo “Caterham”, por lo que la aplicación habrá cambiado de vista y ahora se nos mostrará algo así:



Como en el resto de secciones en la cabecera aparecería el nombre del constructor seleccionado. Justo debajo aparecería información básica del constructor, junto con un cuadro con las estadísticas generales del mismo.

ESTADÍSTICAS	
Victorias	0 (0%)
Pole positions	0 (0%)
Temporadas	3
Carreras	56
Campeonatos de pilotos	0 (0%)
Campeonatos de constructores	0 (0%)

INFORMACIÓN BÁSICA	
Nacionalidad:	
	Malaysian

Como se puede ver en la imagen anterior, encontramos a la izquierda un apartado con información básica del equipo, como puede ser su nacionalidad. En este caso solo se muestra esta información, debido a que el equipo seleccionado no participa en el campeonato por lo que no tiene pilotos conduciendo para él. Si por el contrario el equipo seleccionado fuera de la temporad actual, el campo de información básica saldría de la siguiente manera:

INFORMACIÓN BÁSICA	
Nacionalidad:	
	French
Pilotos actuales	
	Nico Hülkenberg
	Daniel Ricciardo

Centrándonos en el apartado de la derecha, se puede visualizar una serie de estadísticas básicas del constructor seleccionado, estas contienen el número de victorias del equipo (junto con su porcentaje con respecto al total de carreras disputadas), el número de pole positions (junto con su porcentaje con respecto al total de carreras disputadas), el número de temporadas que ha participado como equipo oficial en el campeonato, el número de carreras disputadas, el número de campeonatos ganados (junto con su porcentaje respecto al número de temporadas disputadas) y el número de campeonatos de constructores (respecto al número de temporadas disputadas)

Justo debajo de esta sección de información básica nos encontramos con otra sección de información, pero esta es un poco más específica.

### ACERCA DE CATERHAM

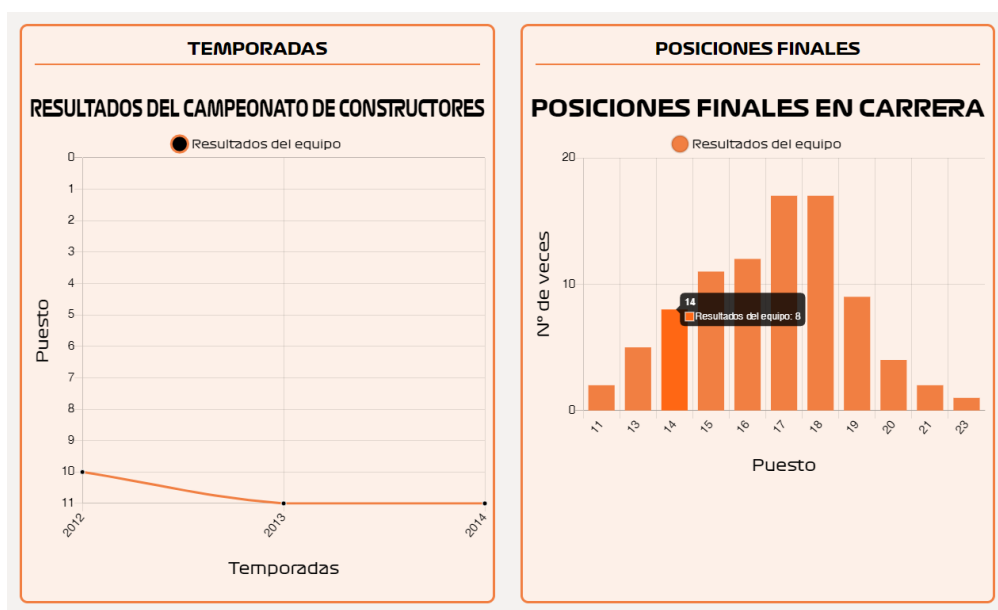


Caterham F1 Team fue una escudería malaya de Fórmula 1. Participó en las temporadas 2012, 2013 y 2014 sin sumar puntos. La base del equipo fue el anterior Team Lotus.

[Saber más](#)

Como vemos en la captura anterior, lo primero que se visualiza es una imagen representativa del equipo seleccionado, junto con una pequeña descripción del mismo. Si se desea saber algo más del equipo bastaría con hacer click en el botón de “Saber más” y este nos abriría la página de Wikipedia del equipo.

Si seguimos explorando esta sección justo debajo de la información del equipo nos encontramos con tres gráficas:



## Anexo I: Manual de usuario

En la primera de todas tenemos los resultados del equipo a lo largo de las temporadas. En el eje Y de la gráfica se encuentran representadas las posiciones de finalización en el campeonato de constructores y en el eje X estarían representadas las temporadas del campeonato en las que ha participado el equipo. Con esto obtenemos una comparativa histórica de resultados, sabiendo el rendimiento general del equipo.


En la gráfica de la derecha se nos representa una gráfica acumulativa de las posiciones finales en carrera. En el eje X de la gráfica se encuentran representados el puesto de finalización en carrera y en el eje Y estaría representado el acumulativo con el número de veces que el equipo ha quedado en esa posición.

Justo debajo de estas dos gráficas tenemos la tercera, que tiene la siguiente fisionomía:



En esta gráfica queda representado algo parecido a la anterior, pero con las posiciones de parrilla. En el eje X de la gráfica tenemos representados los puestos de salida (o de parrilla) y en el eje Y estaría representado el acumulativo con el número de veces que el equipo ha salido desde esa posición.

Continuando el manual de uso, justo debajo de esta gráfica nos encontramos con una tabla, donde se nos listan todos los pilotos que han conducido para el constructor seleccionado:

HAN CONDUCIDO PARA CATERHAM...			
#	Piloto		
1		Marcus Ericsson	ERI
2		Kamui Kobayashi	KOB
3		Heikki Kovalainen	KOV
4		André Lotterer	LOT
5		Vitaly Petrov	PET
6		Charles Pic	PIC
7		Will Stevens	STE
8		Giedo van der Garde	VDG




En esta tabla se detalla la nacionalidad del piloto, su nombre y su abreviatura oficial (en caso de que no tuviera aparecería el valor ---). Si pulsamos en el nombre del piloto la aplicación nos redirigiría a la página específica del piloto.

### 1.4 Apartado pilotos

Para acceder a este apartado de la aplicación, basta con hacer click en la barra de navegación superior o en el footer donde pone “Pilotos” como se puede ver en las siguientes imágenes:

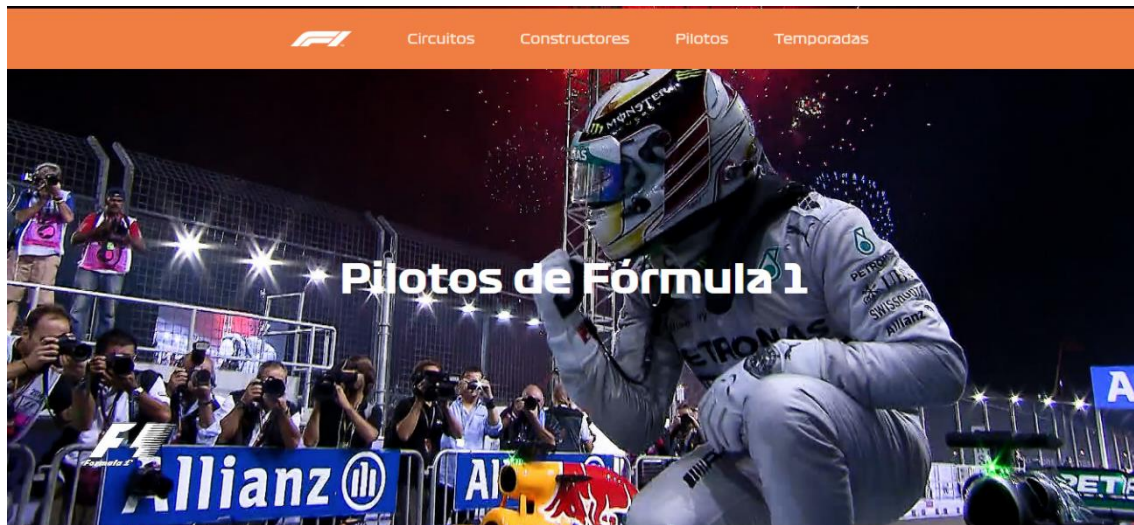


El enlace al apartado de los pilotos de la barra de navegación en este caso tiene algo peculiar y es que si dejamos el ratón encima del mismo aparecerá un menú desplegable, que tiene el siguiente estilo:



 Circuitos Constructores <b>Pilotos</b> Temporadas			
Temporada 2019			
 Alexander ALBON	 Lewis HAMILTON	 Kevin MAGNUSSEN	 George RUSSELL
 Valtteri BOTTAS	 Nico HULKENBERG	 Lando NORRIS	 Carlos SAINZ
 Pierre GASLY	 Robert KUBICA	 Sergio PEREZ	 Lance STROLL
 Antonio GIOVINAZZI	 Daniil KVIAT	 Kimi RAIKKONEN	 Max VERSTAPPEN
 Romain GROSJEAN	 Charles LECLERC	 Daniel RICCIARDO	 Sebastian VETTEL

En este menú desplegable como pone en la esquina superior izquierda nos aparecen todos los pilotos de la temporada 2019, si pulsamos encima de cualquiera de ellos la aplicación nos llevaría a la página dedicada al piloto en cuestión.

Nada más entrar a la sección de pilotos, como es ya habitual, nos aparecerá un encabezado con el nombre de la sección:





















Si seguimos examinando la página nos encontramos justo debajo del encabezado una tabla, la misma como su nombre indica contiene los pilotos que participan en el Campeonato de Pilotos de la temporada actual de Fórmula 1, su estructura es la siguiente:

PILOTOS TEMPORADA 2019				
#		Piloto	Constructor	Puntos
1		<a href="#">Lewis Hamilton</a>	<a href="#">Mercedes</a>	162
2		Valtteri Bottas	Mercedes	133
3		Sebastian Vettel	Ferrari	100
4		Max Verstappen	Red Bull	88
5		Charles Leclerc	Ferrari	72
6		Pierre Gasly	Red Bull	36
7		Carlos Sainz	McLaren	18

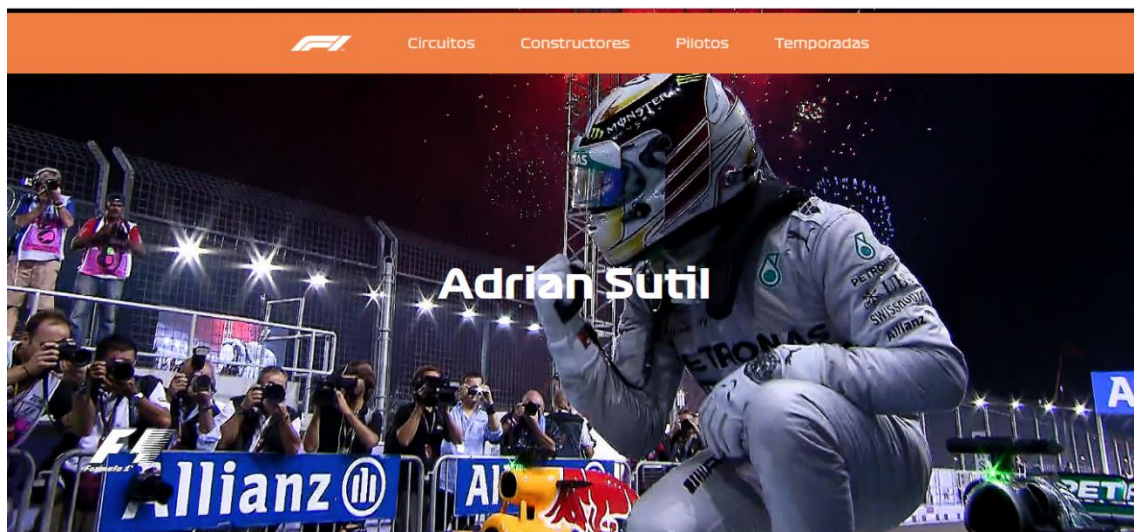
Aquí se nos muestra la posición del piloto, su nacionalidad (representada con la bandera), su nombre, el nombre de su equipo y los puntos conseguidos en el campeonato. Como en casi toda la aplicación si pulsamos encima del nombre del piloto o del constructor nos llevará a la página del piloto o constructor en cuestión respectivamente.

Debajo de esta tabla encontramos una lista con todos los pilotos que alguna vez han participado en al menos una carrera del campeonato de Fórmula 1.

LISTA DE PILOTOS		
A	B	C
 Adolf Brudes	 Basil van Rooyen	 Cal Niday
 Adolfo Cruz	 Bayliss Levrett	 Carel Godin de Beaufort
 <b>Adrian Sutil</b>	 Ben Pon	 Carl Forberg
 Adrián Campos	 Beppe Gabbiani	 Carl Scarborough
 Aguri Suzuki	 Bernard Collomb	 Carlo Abate
 Al Herman	 Bernard de Dryver	 Carlo Facetti

Como en el resto de los apartados la lista se encuentra ordenada y agrupada por orden alfabético por lo que la búsqueda de los pilotos se hace mucho más fácil. Si se pulsa encima de cualquier nombre la aplicación nos redirige a la página del piloto seleccionado.


Para la elaboración de este manual se ha simulado que el usuario ha pulsado sobre el nombre “Adrian Sutil”. Una vez que se ha pulsado en el nombre del piloto la aplicación nos lleva a una pantalla como esta:




Como vemos el nombre del piloto se coloca como título en la cabecera y se sustituye por el título de la sección general.

Justo debajo de la cabecera tenemos un apartado con la información general del piloto:

### Adrian Sutil



 **Starnberg**

Adrian Sutil es un piloto alemán de automovilismo de velocidad. Compitió en categorías internacionales como Fórmula 3 Euroseries y Fórmula 3 Japonesa de la que se proclamó campeón en 2006.

[Saber más](#)

Como vemos en la imagen anterior en este apartado aparecen datos del piloto como su nombre completo, su nacionalidad, el lugar de nacimiento, una fotografía identificativa y una breve descripción. Si el usuario quisiera obtener más información acerca del piloto solo tendría que pulsar el botón y este abriría la página de Wikipedia del piloto, con toda la información disponible.

Justo debajo de esta sección tenemos un cajón con varias estadísticas personales del piloto:

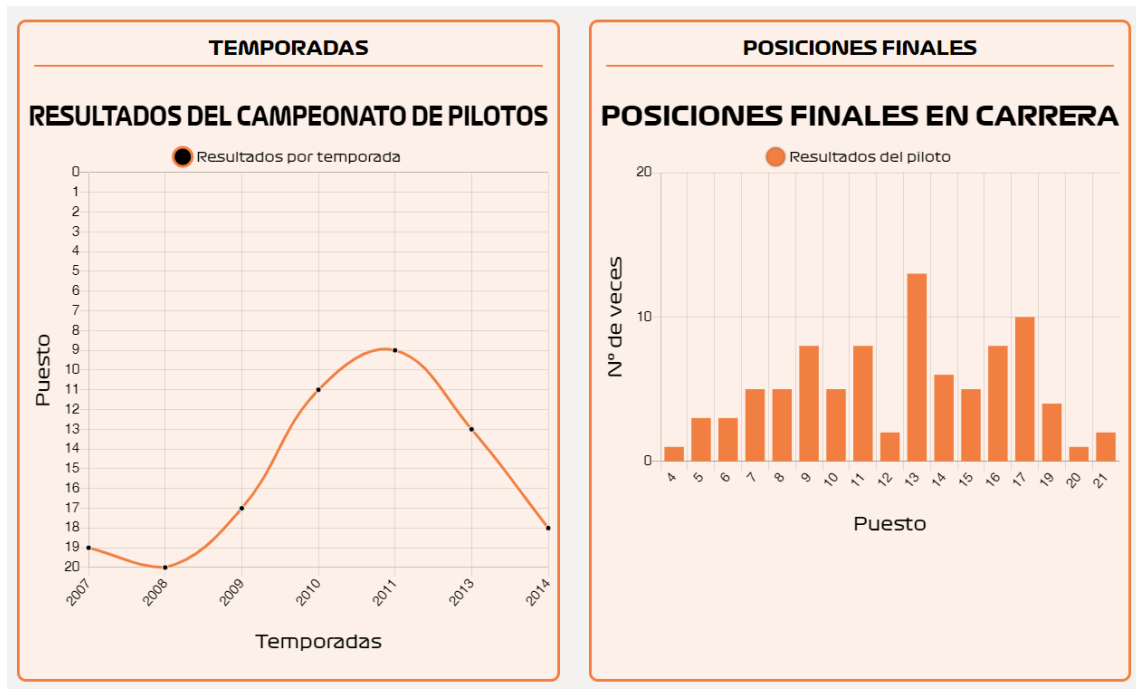
ESTADÍSTICAS	
Victorias	0 (0%)
Podiums	0 (0%)
Pole Positions	0 (0%)
Campeonatos	0 (0%)
Mejor resultado	4º
Temporadas	7
Carreras	128
Puntos conseguidos	124 (1 puntos/carrera)
Retiradas	39 (30.5%)

Este cajón de estadísticas cuenta con los siguientes datos:

- Numero de victorias (junto con el porcentaje relativo al número de carreras disputadas).
- Número de podios (junto con el porcentaje relativo al número de carreras disputadas).
- Número de pole positions (junto con el porcentaje relativo al número de carreras disputadas).
- Número de campeonatos ganados (junto con el porcentaje relativo al número de temporadas disputadas).
- Mejor resultado de su carrera.
- Número de temporadas disputadas.
- Número de carreras disputadas.

- Puntos conseguidos a lo largo de su carrera como piloto, además de una media de puntos por carrera.
- Numero de abandonos (junto con el porcentaje relativo al número de carreras disputadas).

Si seguimos explorando esta sección justo debajo de la información del piloto nos encontramos con tres gráficas:



En la primera de todas tenemos los resultados del piloto a lo largo de las temporadas. En el eje Y de la gráfica se encuentran representadas las posiciones de finalización en el campeonato de pilotos y en el eje X estarían representadas las temporadas del campeonato en las que ha participado el piloto. Con esto obtenemos una comparativa histórica de resultados, sabiendo el rendimiento general del piloto.

En la gráfica de la derecha se nos representa una gráfica acumulativa de las posiciones finales en carrera. En el eje X de la gráfica se encuentran representados el puesto de finalización en carrera y en el eje Y estaría representado el acumulativo con el número de veces que el piloto ha quedado en esa posición.

Justo debajo de estas dos gráficas tenemos la tercera, que tiene la siguiente fisionomía:



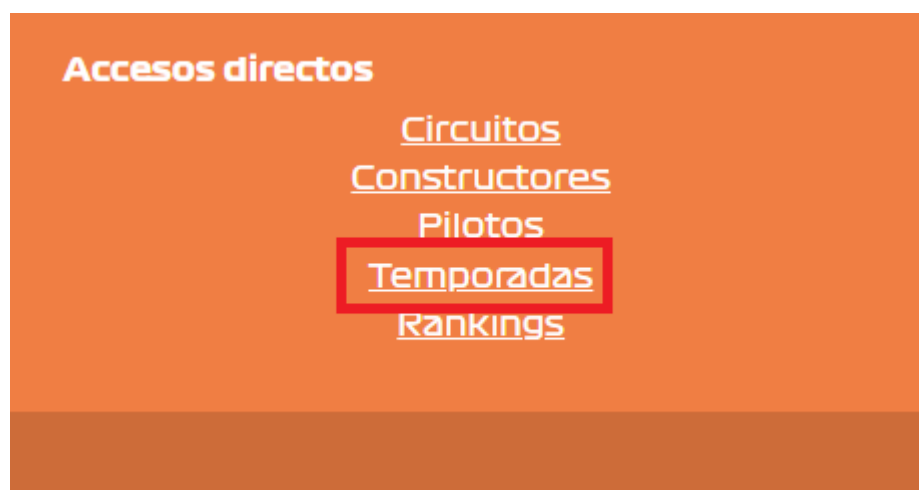
En esta gráfica queda representado algo parecido a la anterior, pero con las posiciones de parrilla. En el eje X de la gráfica tenemos representados los puestos de salida (o de parrilla) y en el eje Y estaría representado el acumulativo con el número de veces que el piloto ha salido desde esa posición.

Por último tenemos una tabla donde quedan reflejados todos los equipos por los que ha pasado el piloto y los compañeros de equipo que ha tenido cada año y en cada equipo. Cabe recordar que si se pulsa sobre el nombre de los pilotos, del constructor o incluso de la temporada la aplicación nos mostrará datos relativos a cada uno de ellos.

EQUIPOS Y COMPAÑEROS		
Temporada	Compañero	Equipo
2007	Christijan Albers Markus Winkelhock Sakon Yamamoto	Spyker
2008	<u>Giancarlo Fisichella</u>	<u>Force India</u>
2009	Giancarlo Fisichella Vitantonio Liuzzi	Force India
2010	Vitantonio Liuzzi	Force India
2011	Paul di Resta	Force India
2013	Paul di Resta	Force India
2014	Esteban Gutiérrez	Sauber

## 1.5 Apartado temporadas

Para acceder a este apartado de la aplicación, basta con hacer click en la barra de navegación superior o en el footer donde pone “Temporadas” como se puede ver en las siguientes imágenes:





Una vez que se pulse en uno de los dos enlaces se nos abrirá la página de temporadas, donde en una primera instancia nos aparecerá su respectivo título de sección. Justo debajo de la cabecera de la página nos encontramos con un apartado único y principal:



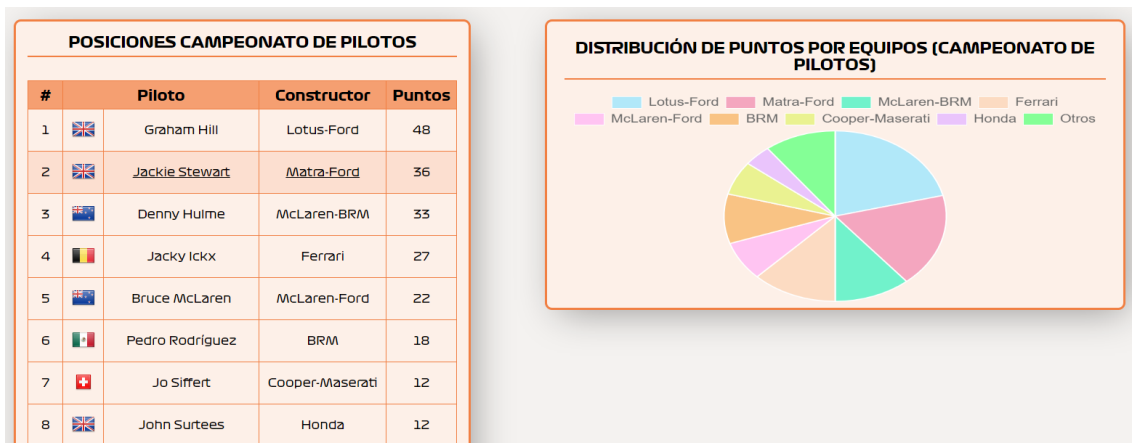
LISTA DE TEMPORADAS		
#	Temporada	Numero de carreras
1	1950	7
2	1951	8
3	1952	8
4	1953	9
5	1954	9
6	1955	7
7	1956	8
8	1957	8
9	1958	11

El componente reflejado en la imagen anterior es bastante simple, ya que nos muestra una lista con todas las temporadas de Fórmula 1 disputadas hasta la fecha, junto con un contador y el número de carreras que ha tenido esa temporada. Si se pulsa encima de cualquiera de las temporadas la aplicación nos mostrará los detalles de la temporada en cuestión.

Para la realización del manual se ha simulado que el usuario ha seleccionado la temporada 1968. Cuando se seleccione una temporada el título de la sección cambiará reflejando en él la temporada seleccionada:



Justo después del encabezado nos encontramos dos apartados:



En la sección de la izquierda vemos una tabla donde aparecen reflejadas las posiciones del Campeonato de Pilotos de la temporada seleccionada. Cabe recordar que si pulsamos encima del nombre del piloto o del equipo la aplicación nos llevará a la página detalle del constructor o el piloto respectivamente.

En el gráfico de la derecha vemos el reparto de puntos por equipos del campeonato de pilotos. Este reparto se ha calculado respecto al total de puntos repartidos en la temporada en cuestión. Si se pulsa en las etiquetas del gráfico se pueden ocultar los equipos que no se quieran visualizar.

Por último, nos encontramos un resumen de las carreras disputadas en la temporada, de cada carrera se muestra la fecha, el nombre oficial de la carrera, el circuito, el piloto ganador, el constructor ganador, el número de vueltas de la carrera y el tiempo de finalización de la misma.