

Manual de Usuario - Aplicación de Fórmula 1



Aplicación de Fórmula 1
Versión 1.0

Sergio Fernández Buceta

1. Introducción.....	3
2. Pantallas de la Aplicación.....	3
2.1 Pantalla de Inicio.....	3
2.3 Carreras por Años.....	5
2.5 Escuderías.....	7
2.6 Estadísticas de Victorias.....	8
3. API Utilizada.....	10
4. Widgets Nuevos, Dependencias y sus Usos.....	11
Widget 1: Curved_Navigation_Bar.....	11
Widget 2: Graphic.....	12
Widget 3: Loading_indicator.....	13
Widget 4: Sidebarx.....	14
Dependencia 1: Ionicons.....	16

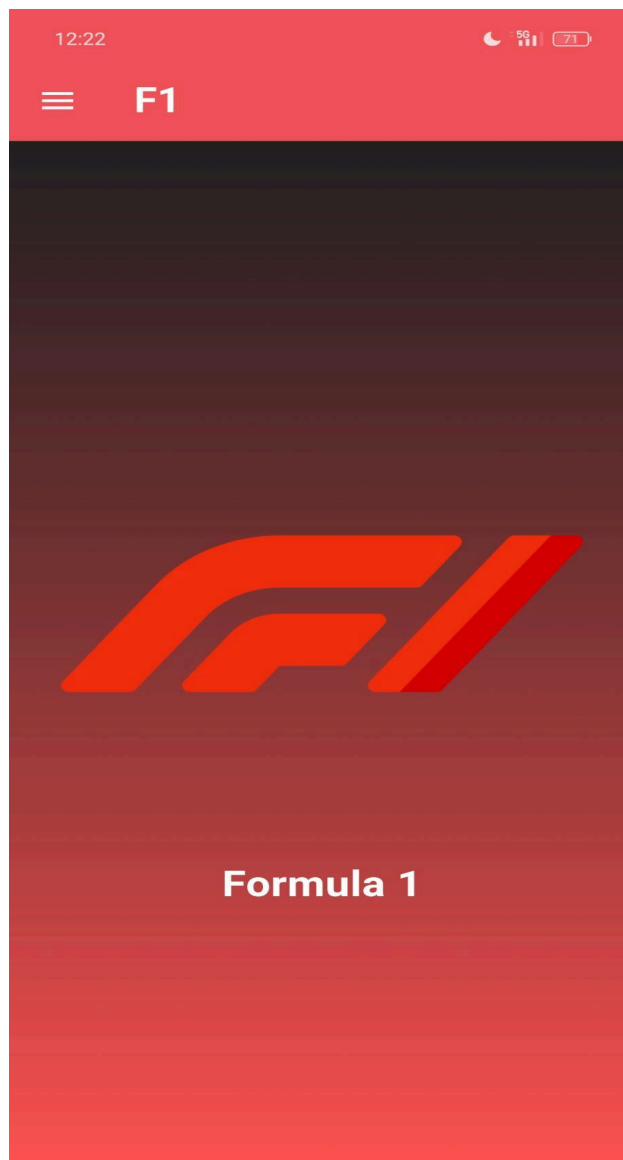
1. Introducción

Esta aplicación permite a los usuarios explorar información detallada sobre la Fórmula 1, incluyendo pilotos, escuderías, carreras por año y estadísticas de victorias. A través de una interfaz intuitiva, los usuarios pueden navegar entre diferentes secciones utilizando un menú lateral.

2. Pantallas de la Aplicación

2.1 Pantalla de Inicio

La pantalla de inicio muestra una presentación de la aplicación y un menú lateral que permite acceder a las diferentes secciones.



2.2 Búsqueda de Pilotos

Esta pantalla permite buscar pilotos de Fórmula 1 por nombre. Al seleccionar un piloto, se accede a su perfil con información detallada.

12:22

5G

71

←

fern

⊗



Fernando Alonso

12:22

5G

71

←



Fernando Alonso

#14

Spanish

Estadísticas de carrera

Grandes Premios

404

Campeonatos del mundo

2

Podios

106

Total de puntos

2337

Mejores resultados

Mejor resultado en carrera

P1

Mejor resultado en clasificación

P1

Información personal

Nacimiento

Oviedo, Spain

Edad

43

2024

Aston Martin F1 Team



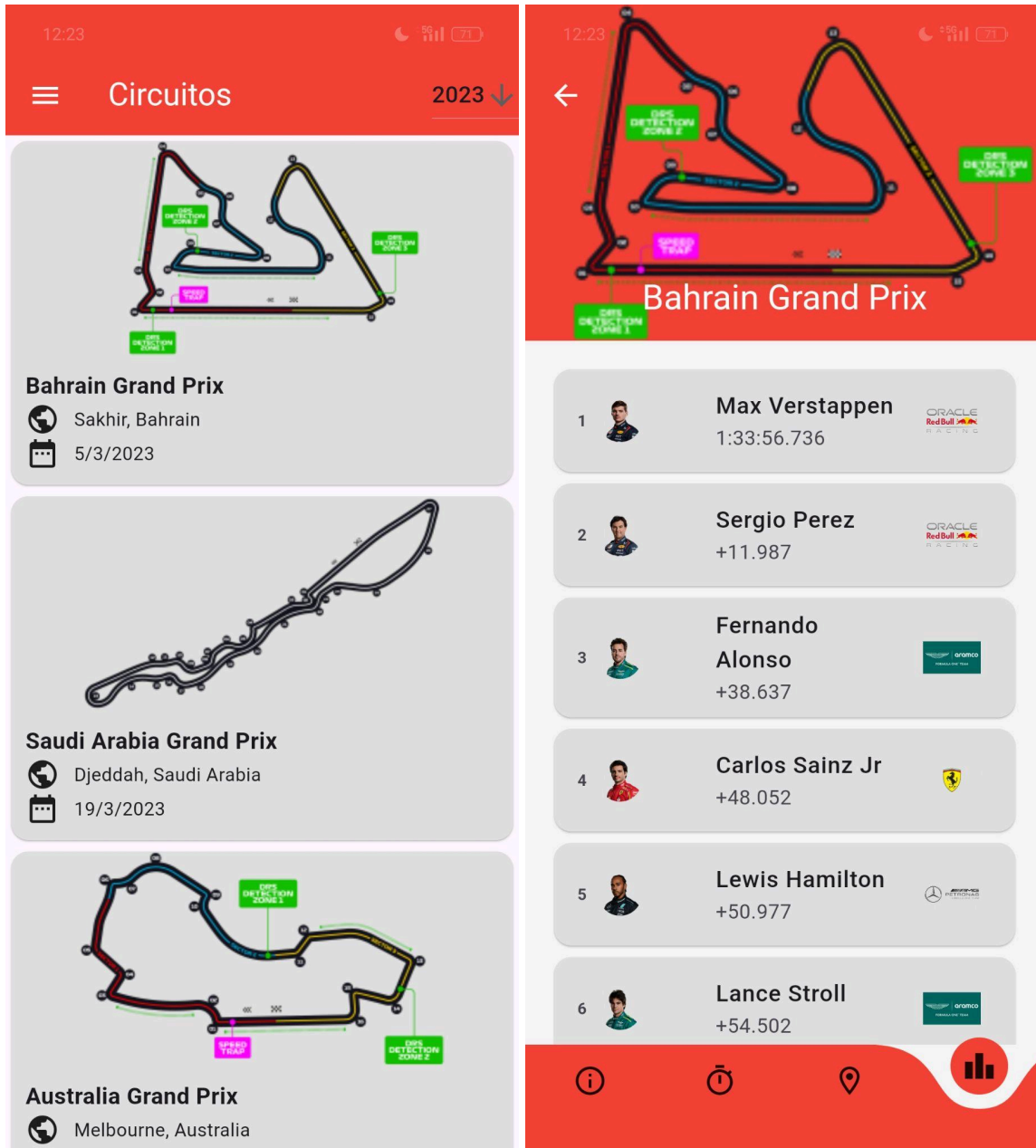
2023

Aston Martin F1



2.3 Carreras por Años

Se muestra una lista de temporadas de la Fórmula 1. Al seleccionar un año, se muestran las carreras de esa temporada. Pulsando sobre una de ellas, podrás acceder al detalle, donde dispondrás de un menú en la parte baja, pudiendo acceder al resultado de dicha carrera.



2.4 Pilotos por Año

Permite visualizar los pilotos que participaron en cada temporada. Desde aquí se puede acceder al detalle de cada piloto.

12:23

☾ 5G 71

≡

Pilotos

2023 ↓



Max Verstappen
575 puntos

ORACLE
Red Bull
RACING



Sergio Perez
285 puntos

ORACLE
Red Bull
RACING



Lewis Hamilton
234 puntos





Fernando Alonso
206 puntos





Charles Leclerc
206 puntos





Lando Norris
205 puntos





Carlos Sainz Jr
200 puntos





George Russell
175 puntos





Oscar Piastri
97 puntos



12:23

☾ 5G 71

←



Max Verstappen
#1 Dutch

Estadísticas de carrera

Grandes Premios

Campeonatos del mundo

Podios

Total de puntos

209

4

112

3023.5

Mejores resultados

Mejor resultado en carrera

Mejor resultado en clasificación

P1

P1

Información personal

Nacimiento

Edad

Hasselt, Belgium

27

2024

Red Bull Racing

ORACLE
Red Bull
RACING

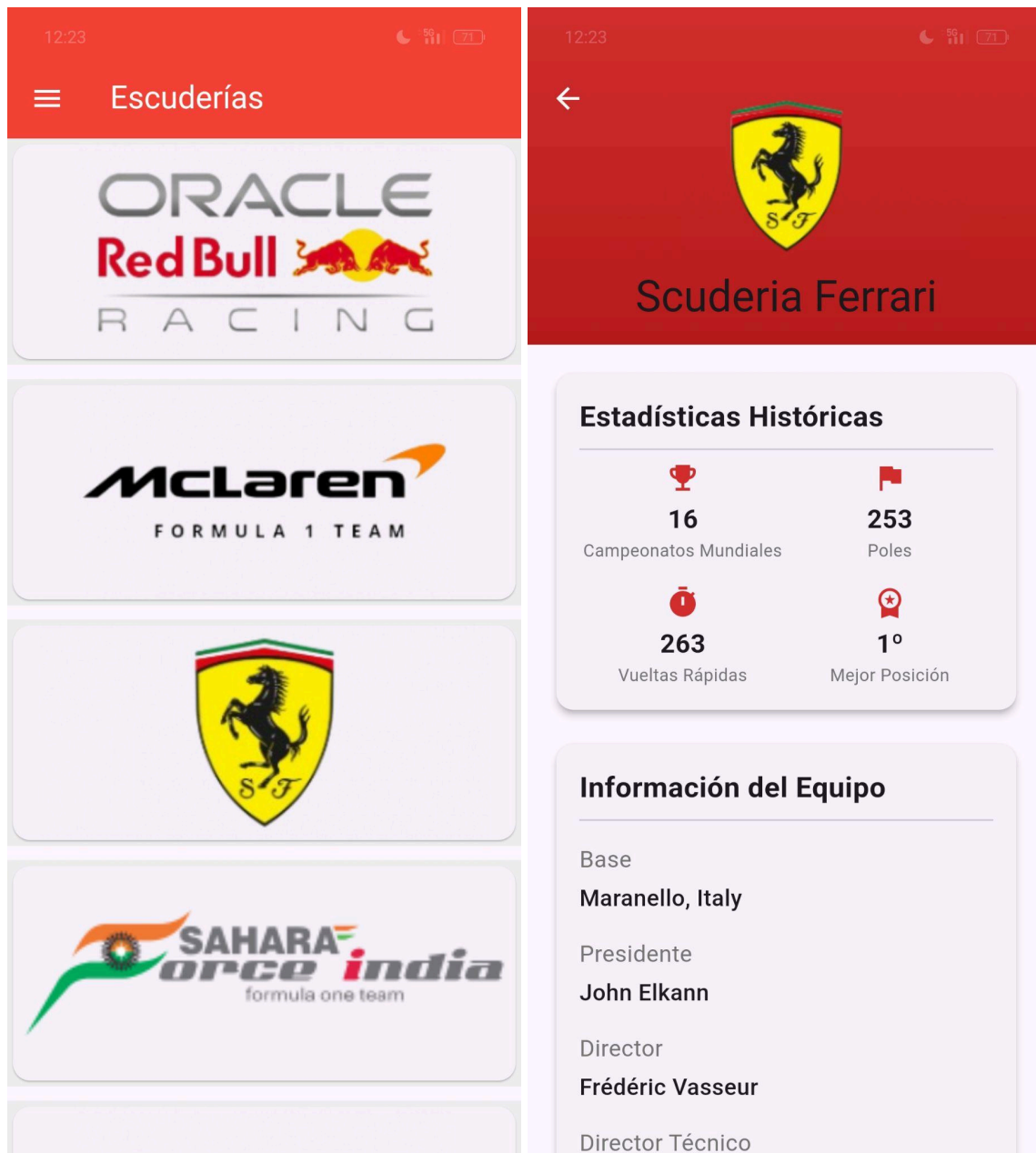
2023

Red Bull Racing

ORACLE
Red Bull
RACING

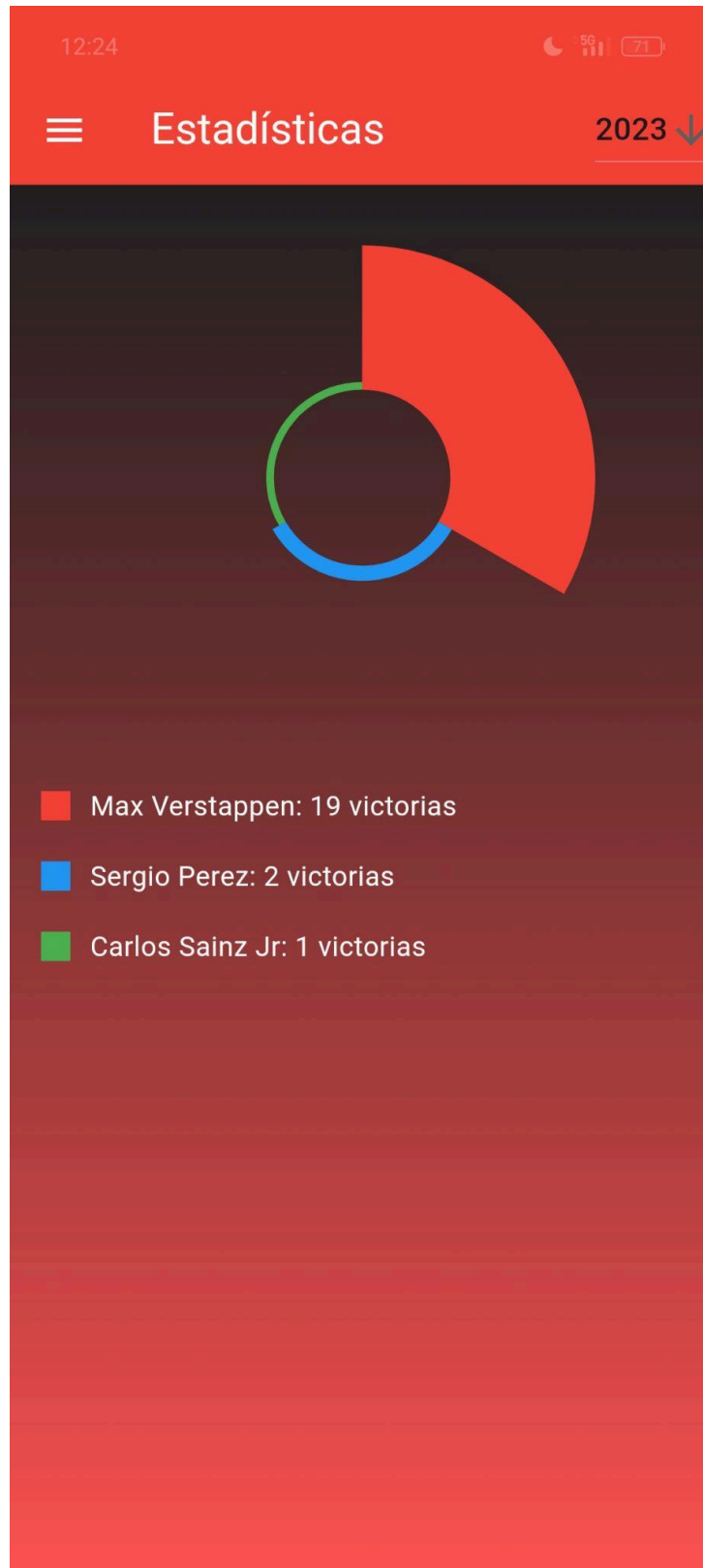
2.5 Escuderías

Lista de escuderías de Fórmula 1 con acceso a información detallada de cada una.



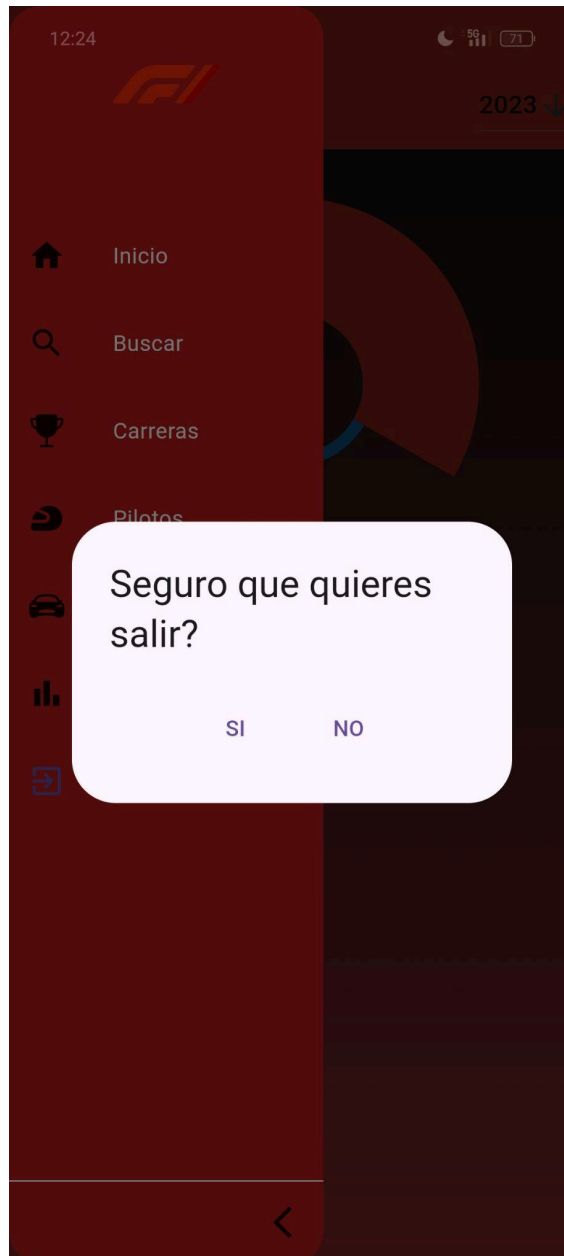
2.6 Estadísticas de Victorias

Muestra un gráfico de victorias por año, permitiendo conocer qué pilotos y escuderías fueron más exitosos.



2.7 Menú Lateral y Botón de Salida

El menú lateral facilita la navegación entre pantallas. El botón de salida permite cerrar la aplicación de manera segura.



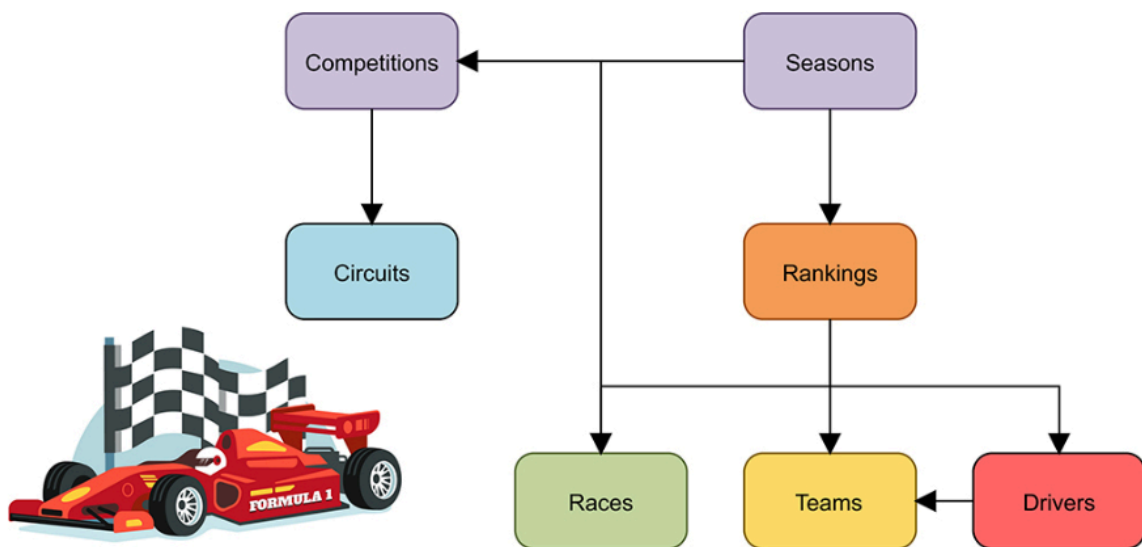
3. API Utilizada

La aplicación utiliza la API **RapidApiF1** para obtener datos sobre pilotos, carreras, escuderías y estadísticas de Fórmula 1. Uno de los problemas de la versión gratuita de esta api es que solo deja realizar 100 consultas al día, con un máximo de 10 por minuto.

Fuente:

<https://api-sports.io/documentation/formula-1/v1#section/Authentication/RAPIDAPI-Account>

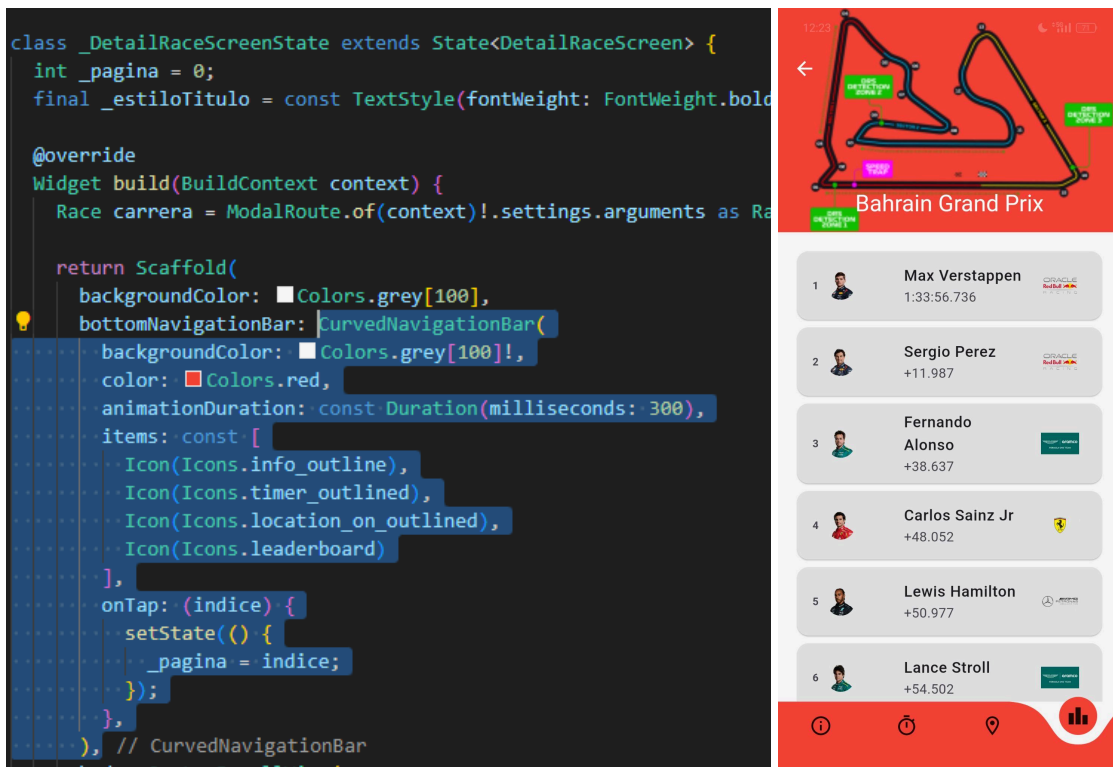
Arquitectura de la base de datos:



4. Widgets Nuevos, Dependencias y sus Usos

Widget 1: Curved_Navigation_Bar

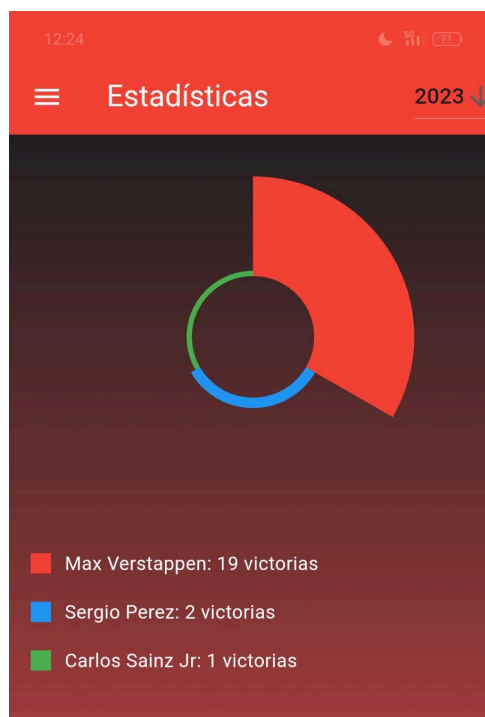
- **Ubicación:** Pantalla de detalle de Carreras
- **Descripción:** Permite mostrar un menú en la parte inferior de la pantalla, con forma curva y una pequeña animación al cambiar el elemento seleccionado
- **Cómo se encontró:** Buscando widgets populares en pub.dev.
- **Captura de Pantalla:**



Widget 2: Graphic

- **Ubicación:** Pantalla de Estadísticas de Victorias
- **Descripción:** Gráfico en forma circular que muestra el número de victorias de los pilotos en el año seleccionado.
- **Cómo se encontró:** Buscado en internet widgets que permitieran mostrar gráficos.
- **Captura de Pantalla:**

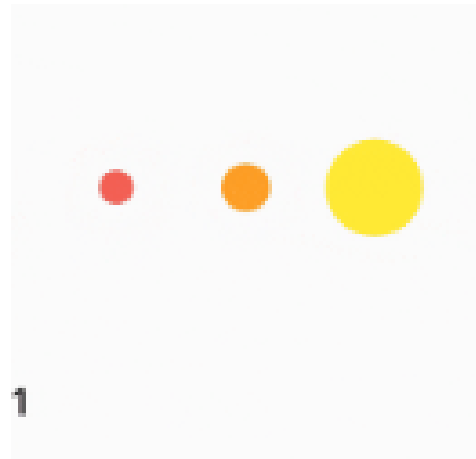
```
Widget _grafico({required List<DriverSeason> listaPilotos}) {  
  final colores = [  
    Colors.red,  
    Colors.blue,  
    Colors.green,  
    Colors.orange,  
    Colors.purple,  
    Colors.yellow,  
    Colors.cyan,  
  ];  
  
  final datosGrafico = listaPilotos  
    .where((piloto) => piloto.wins > 0)  
    .map((piloto) => {  
      'piloto': piloto.driver.name,  
      'victorias': piloto.wins,  
    })  
    .toList();  
  
  datosGrafico.sort((a, b) => {b['victorias'] as int}.compareTo(a['victorias'] as int));  
  
  return ListView(  
    children: [  
      SizedBox(  
        height: 300,  
        child: datosGrafico.isEmpty  
          ? Center(  
            child: Text(  
              'No hay victorias registradas',  
              style: TextStyle(color: Colors.white),  
            ), // Text  
          ) // Center  
          : Chart(  
            data: datosGrafico,  
            variables: {  
              'piloto': Variable(  
                accessor: (Map map) => map['piloto'] as String,  
              ), // Variable  
              'victorias': Variable(  
                accessor: (Map map) => map['victorias'] as int,  
              ), // Variable  
            },  
            marks: [  
              IntervalMark(  
                position: Varset('piloto') * Varset('victorias'),  
                color: ColorEncode(  
                  variable: 'piloto',  
                  values: colores,  
                ), // ColorEncode  
              ), // IntervalMark  
            ],  
            coord: PolarCoord(  
              startRadius: 0.3,  
              endRadius: 0.9  
            ), // PolarCoord  
          ), // Chart  
        ), // SizedBox  
      ...datosGrafico.asMap().entries.map((entry) {  
        final index = entry.key;  
        final piloto = entry.value['piloto'];  
        final victorias = entry.value['victorias'];  
        final color = colores[index % colores.length];  
      }  
    ],  
  );  
}
```



Widget 3: Loading_indicator

- **Ubicación:** En todas las pantallas que cargan datos de la api
- **Descripción:** Widget que se usa en lugar del Circular Progress Indicator y que presenta un mejor aspecto visual permitiendo seleccionar el que más nos guste.
- **Cómo se encontró:** Buscando widgets populares en pub.dev.
- **Captura de Pantalla:** Al ser complicado sacar una captura de esto, se adjunta la imagen del efecto usado(ballPulse).

```
class TeamsScreen extends StatelessWidget {  
  const TeamsScreen({super.key});  
  
  @override  
  Widget build(BuildContext context){  
    Provider? provider = Provider.of(context, listen: false);  
    return Scaffold(  
      appBar: AppBar(  
        title: const Text("Escuderias"),  
      ), // AppBar  
      drawer: SideBar(),  
      body: FutureBuilder(  
        future: provider.obtenerEscuderias(),  
        builder: (context, snapshot) {  
          if (snapshot.hasError) {  
            //print(snapshot.error);  
            return Center(  
              child: Text("ERROR AL DESCARGAR"),  
            ); // Center  
          } else if (!snapshot.hasData) {  
            return Center(  
              child: LoadingIndicator(  
                indicatorType: indicator.ballPulse,  
                colors: [Colors.red, Colors.green],  
              ), // LoadingIndicator  
            ); // Center  
          } else {  
            return Consumer<Provider?>(  
              builder: (context, provider, child) => Escuderias(listaEscuderias: provider.listaEscuderias),  
            ); // Consumer  
          }  
        },  
      ),  
    );  
  }  
}
```

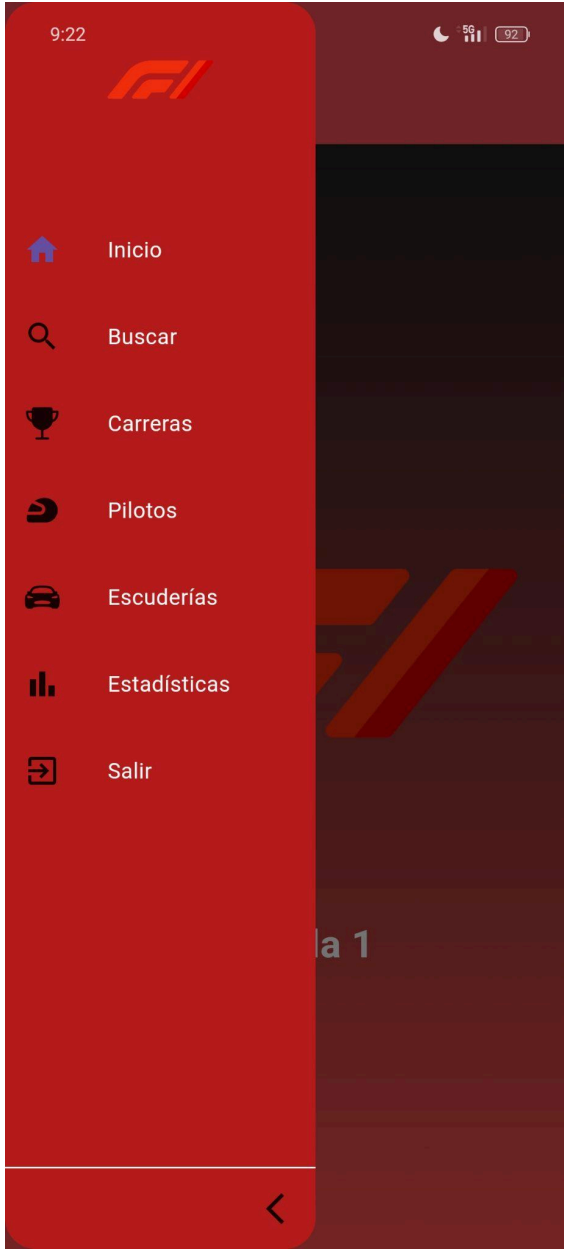


Widget 4: Sidebarx

- **Ubicación:** En las pantallas desde las que se puede acceder al drawer
- **Descripción:** Widget que se usa en lugar del drawer y que presenta un mejor aspecto visual, además de más opciones de configuración. Además, dispone de dos posiciones, una ampliada y otra contraída.
- **Cómo se encontró:** Buscando en internet widgets que sustituyeran al drawer.
- **Captura de Pantalla:**

```
return SidebarX(  
  controller: _controller,  
  theme: SidebarXTheme(  
    width: 100,  
    itemTextPadding: EdgeInsets.only(left: 30),  
    selectedItemTextPadding: EdgeInsets.only(left: 30),  
    textStyle: TextStyle(color: Colors.white),  
    selectedItemStyle: TextStyle(color: Colors.white),  
    decoration: BoxDecoration(  
      color: Colors.red[900],  
      borderRadius: BorderRadius.circular(20),  
    ), // BoxDecoration  
  ), // SidebarXTheme  
  extendedTheme: SidebarXTheme(  
    width: 200,  
  ), // SidebarXTheme  
  footerDivider: Divider(color: Colors.white, height: 1),  
  headerBuilder: (context, extended) {  
    return SizedBox(  
      height: 100,  
      child: Padding(  
        padding: EdgeInsets.all(16.0),  
        child: Image.asset("assets/images/f1.png"),  
      ), // Padding  
    ); // SizedBox  
  },  
  items: [  
    SidebarXItem(  
      icon: Icons.home,  
      label: "Inicio",  
      onTap: () {  
        Navigator.of(context).pop();  
        Navigator.of(context).pushNamed('home');  
      },  
    ), // SidebarXItem  
    SidebarXItem(  
      icon: Icons.search,  
      label: "Buscar",
```

```
    SidebarXItem(  
      icon: Ionicons.trophy,  
      label: "Carreras",  
      onTap: () {  
        Navigator.of(context).pop();  
        Navigator.of(context).pushNamed('race');  
      },  
    ), // SidebarXItem  
    SidebarXItem(  
      icon: Icons.sports_motorsports,  
      label: "Pilotos",  
      onTap: () {  
        Navigator.of(context).pop();  
        Navigator.of(context).pushNamed('driver');  
      },  
    ), // SidebarXItem  
    SidebarXItem(  
      icon: Ionicons.car_sport,  
      label: "Escuderías",  
      onTap: () {  
        Navigator.of(context).pop();  
        Navigator.of(context).pushNamed('teams');  
      },  
    ), // SidebarXItem  
    SidebarXItem(  
      icon: Icons.bar_chart,  
      label: "Estadísticas",  
      onTap: () {  
        Navigator.of(context).pop();  
        Navigator.of(context).pop();  
        Navigator.of(context).pushNamed('stats');  
      },  
    ), // SidebarXItem  
    SidebarXItem(  
      icon: Icons.exit_to_app,  
      label: "Salir",  
      onTap: () {  
        _showAlertDialog(context);  
      },  
    ),  
  ],  
);
```



Dependencia 1: Ionicons

- **Ubicación:** Se usan en el sidebarx, para los iconos de cada pantalla.
- **Descripción:** Aporta más iconos que los que tiene flutter como predeterminados.
- **Cómo se encontró:** Buscando widgets que aportaran más iconos.