

# PROYECTO InfoAutos

*Aplicación de Información y Comparación de Autos*



- Nombre del Proyecto: Infoautos.
- Ciclo Formativo: 2º Desarrollo de aplicaciones multiplataforma.
- Tutor: Carlos Alberto Sánchez Yufera.
- Curso: 2024-2025

## INDICE

<b>PROYECTO InfoAutos</b>	<b>1</b>
<b>RESUMEN</b>	<b>2</b>
<b>PALABRAS CLAVE</b>	<b>3</b>
<b>INTRODUCCIÓN</b>	<b>3</b>
<b>ESTADO DEL ARTE</b>	<b>4</b>
<b>ESTUDIO DE VIABILIDAD</b>	<b>5</b>
<b>ESTUDIO DE MERCADO</b>	<b>5</b>
<b>ANÁLISIS DE REQUISITOS</b>	<b>6</b>
<b>DISEÑO</b>	<b>8</b>
<b>CODIFICACIÓN</b>	<b>12</b>
<b>DOCUMENTACIÓN</b>	<b>14</b>
<b>DESPLIEGUE</b>	<b>18</b>
<b>HERRAMIENTAS DE APOYO</b>	<b>19</b>
<b>CONCLUSIONES</b>	<b>19-20</b>
<b>BIBLIOGRAFIA</b>	<b>20</b>

## RESUMEN

El presente proyecto consiste en el desarrollo de una aplicación móvil llamada **InfoAutos**, diseñada para facilitar la consulta, comparación y gestión de vehículos de distintas marcas y características técnicas. La app permite a los usuarios visualizar fichas detalladas, guardar favoritos, y comparar dos coches de manera visual e intuitiva.

El objetivo principal es ofrecer una solución sencilla y funcional que permita tanto a particulares como a profesionales del sector automotriz acceder rápidamente a datos relevantes de coches almacenados en una base de datos local. InfoAutos ha sido desarrollada con **Flutter** y **SQLite**, garantizando un rendimiento óptimo y compatibilidad multiplataforma, y se apoya en un diseño oscuro moderno con uso de logos e íconos representativos para mejorar la experiencia del usuario.

## PALABRAS CLAVE

Flutter, SQLite, vehículos, coches, comparación, favoritos, app móvil, DAM, diseño UI, SharedPreferences

## INTRODUCCIÓN

En la actualidad, con la enorme variedad de vehículos disponibles en el mercado, resulta cada vez más difícil para los usuarios comparar modelos y tomar decisiones de compra informadas. Muchos compradores potenciales se ven obligados a consultar múltiples sitios web, catálogos o incluso visitar concesionarios solo para reunir información técnica básica sobre distintos coches. Este proceso es tedioso, poco eficiente y muchas veces termina generando confusión o decisiones impulsivas.

**InfoAutos** nace con el propósito de simplificar este proceso. Se trata de una aplicación móvil que permite al usuario visualizar fichas detalladas de vehículos, compararlos de forma simultánea y guardar sus modelos favoritos, todo desde un solo lugar. Gracias a una interfaz intuitiva, oscura y bien estructurada, el usuario puede consultar rápidamente especificaciones como tipo de motor, consumo, transmisión, potencia, precio, entre otros, sin necesidad de acudir a fuentes externas.

El principal **objetivo del proyecto** es desarrollar una herramienta multiplataforma, accesible y visualmente clara que ayude a los usuarios a gestionar la información de coches de forma cómoda y ordenada. A través de funcionalidades como la comparación de vehículos, la marcación de favoritos o la búsqueda inteligente por marca y modelo, se pretende agilizar el proceso de selección y análisis automovilístico.

Durante el diseño inicial del proyecto se evaluaron distintas soluciones: el uso de páginas web responsivas, aplicaciones móviles con conexión a APIs de terceros o aplicaciones basadas en formularios con bases de datos externas. Sin embargo, tras analizar las ventajas e inconvenientes, se optó por una **aplicación móvil desarrollada con Flutter y basada en SQLite local**, por ser una opción ligera, rápida y funcional que no requiere conexión a internet constante y garantiza compatibilidad tanto con Android como iOS.

A lo largo del desarrollo del proyecto se intentarán resolver algunas dudas como: ¿es posible mantener una experiencia fluida con grandes volúmenes de datos en local?, ¿cómo gestionar correctamente los favoritos sin alterar la base de datos original?, ¿qué nivel de personalización es recomendable ofrecer al usuario sin recargar la interfaz?

## ESTADO DEL ARTE

El desarrollo de esta aplicación se justifica en la necesidad de proporcionar a los usuarios un **acceso centralizado a información confiable sobre vehículos**, sin necesidad de visitar múltiples sitios web. La inclusión de un comparador permitirá comparar diferentes modelos de coches de manera eficiente, y acompañado con la función de favoritos mejorará la experiencia del usuario.

### Aportaciones del proyecto:

- Interfaz sencilla y clara.
- Comparación directa de dos coches en paralelo.
- Funcionamiento totalmente **offline**.
- Carga de datos desde un archivo JSON.
- Posibilidad de destacar diferencias entre coches (ampliación futura).
- Integración de API's (ampliación futura).

## ESTUDIO DE VIABILIDAD

### - Análisis DAFO:

Fortalezas	Debilidades
App Multiplataforma	No uso de APIs externas (limitación de datos)
Funciona sin conexión	Catálogo limitado a los coches del JSON
Interfaz clara y fácil de usar	Sin funciones avanzadas como búsquedas web
Oportunidades	Amenazas
Usuarios que buscan comparar sin publicidad	Competencia de apps comerciales más completas
Ampliación futura con datos externos	Necesidad de actualizar el catálogo manualmente

## ESTUDIO DE MERCADO

Actualmente, las plataformas de comparación de coches como **Coches.net** o **Carwow** dominan el sector, pero se centran en el aspecto comercial, están llenas de anuncios y requieren conexión constante. Muchos usuarios buscan alternativas **más limpias, rápidas y privadas**.

Este proyecto cubre un hueco: una app **ligera, con comparador, sin publicidad**, ideal para estudiantes, comparadores ocasionales o usuarios que no desean registrarse en plataformas comerciales.

### - Viabilidad Temporal:

El desarrollo se ha planteado de forma progresiva, organizando los módulos (interfaz, lógica, base de datos, persistencia) de forma que permita completar el proyecto en el tiempo disponible. A pesar de haber descartado el uso de APIs por ser todas de pago, se ha adaptado el diseño para cargar la información desde un JSON local.

### - Planificación Temporal:

Fase	Duración
Preparación de datos y estructura local	2 semanas
Diseño UI/UX	3 semanas
Desarrollo de funciones básicas	4 semanas
Pruebas y ajustes	2 semanas
Documentación y lanzamiento	2 semanas

## ANÁLISIS DE REQUISITOS

### - Requisitos funcionales:

#### - Frontend:

- Mostrar una lista de coches de forma resumida (marca, modelo, motor...).
- Al tocar un coche, mostrar una vista detallada con sus especificaciones completas.
- Posibilidad de marcar o desmarcar coches como favoritos.
- Acceder a una pestaña de favoritos para ver los coches guardados.
- Acceder a una pestaña de comparador, con botones para seleccionar dos coches y ver sus fichas lado a lado.
- Posibilidad de quitar un coche del comparador con un botón de eliminación.
- Muestra de logos de cada marca de coche, si están disponibles en los recursos.

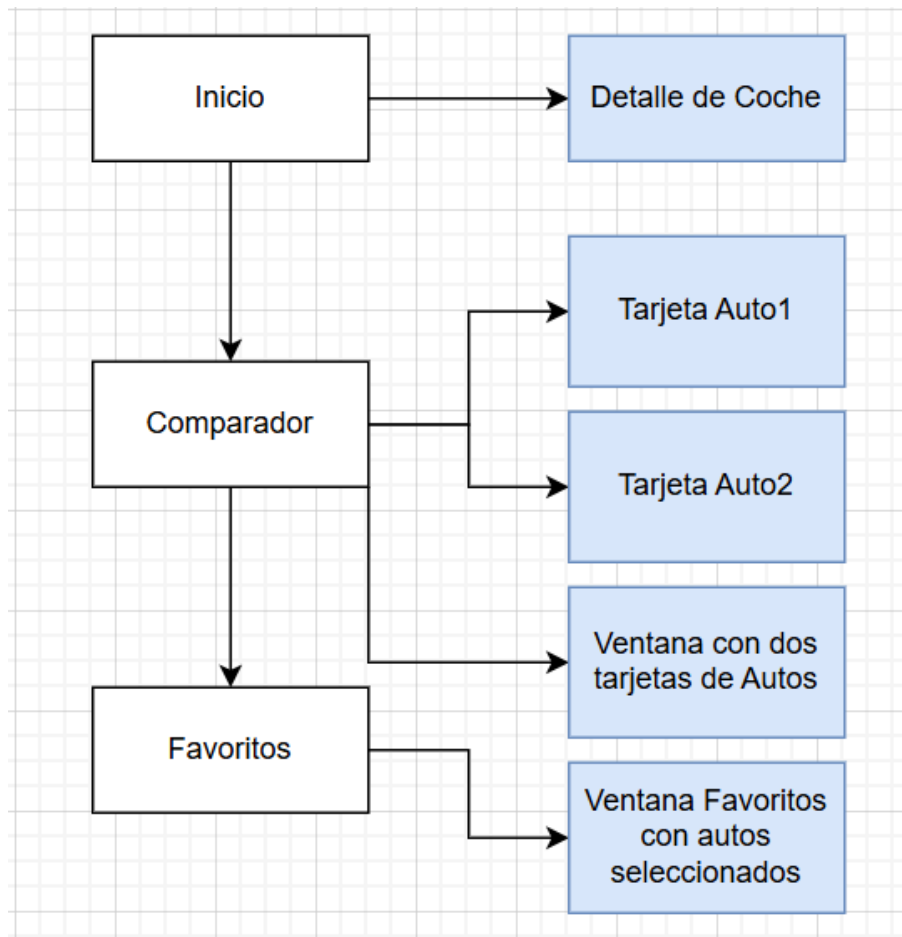
#### - Backend (lógica interna con Provider)

- Gestionar el estado de los coches cargados desde JSON y almacenados en SQLite.
- Controlar el estado de los coches favoritos (guardarlos localmente con **Shared Preferences**).
- Controlar el estado de los coches seleccionados para comparar.

#### - Base de Datos

- Estructura local en SQLite con campos bien definidos para cada coche.
- Carga inicial de datos desde un archivo coches.json.
- No se requiere conexión a servidores ni autenticación.

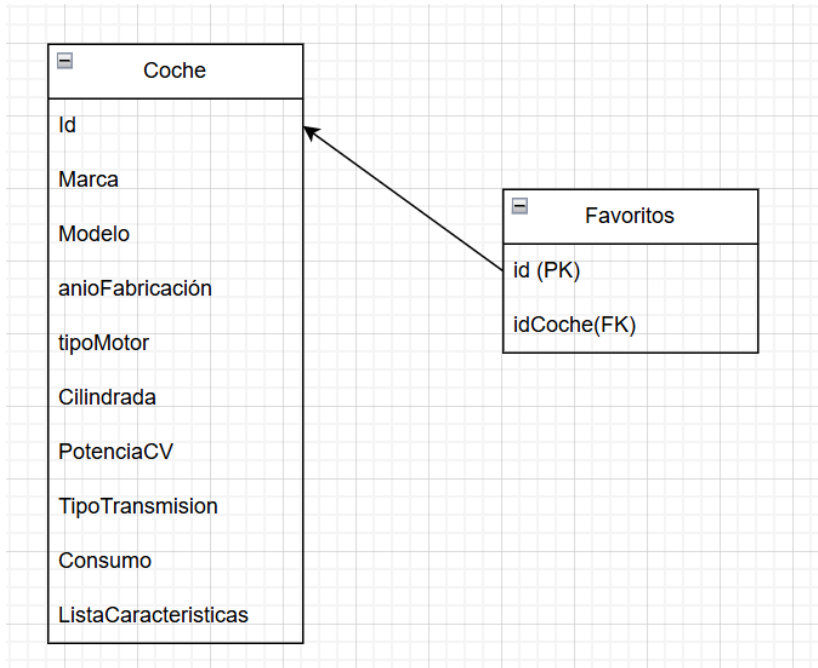
- Diagrama casos de uso





## DISEÑO

- Diseño conceptual entidad-relación:

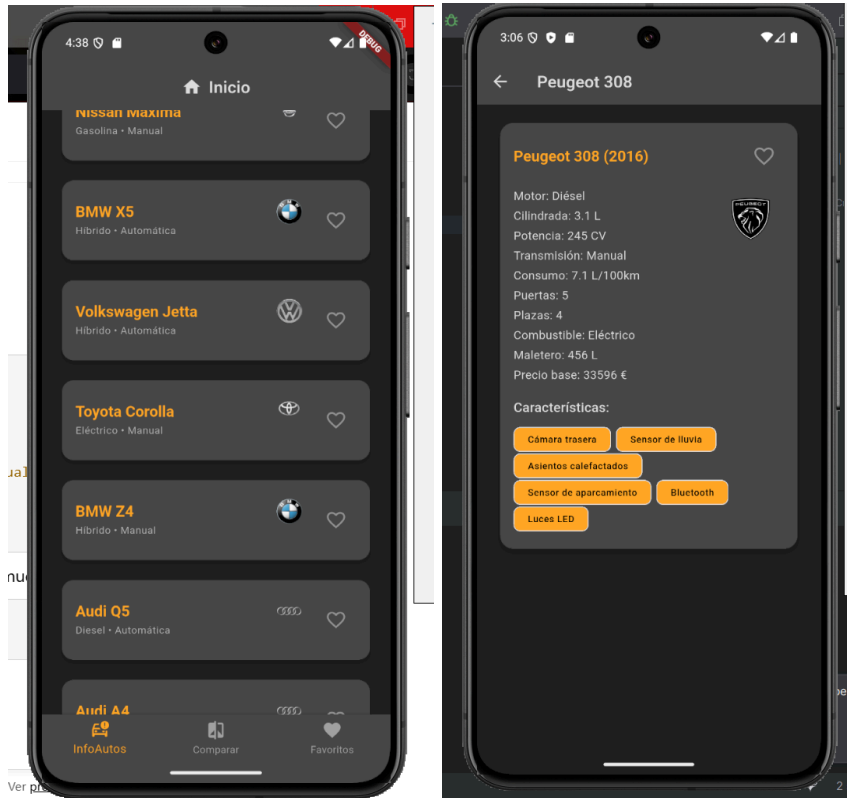


- Descripción tabla y campos:

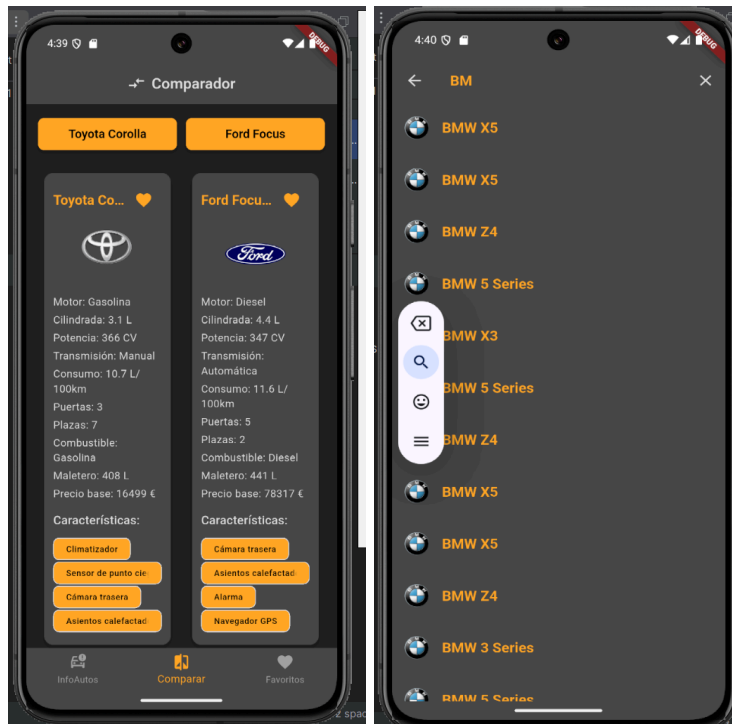
```
'coches',
{
  'marca': coche.marca,
  'modelo': coche.modelo,
  'añoFabricacion': coche.añoFabricacion,
  'tipoMotor': coche.tipoMotor,
  'cilindrada': coche.cilindrada,
  'potenciaCV': coche.potenciaCV,
  'tipoTransmision': coche.tipoTransmision,
  'consumo': coche.consumo,
  'listaCaracteristicas': coche.listaCaracteristicas.join(', '),
  'numPuertas': coche.numPuertas,
  'numPlazas': coche.numPlazas,
  'tipoCombustible': coche.tipoCombustible,
  'volumenMaletero': coche.volumenMaletero,
  'precioBase': coche.precioBase
},
```

- Mockups:

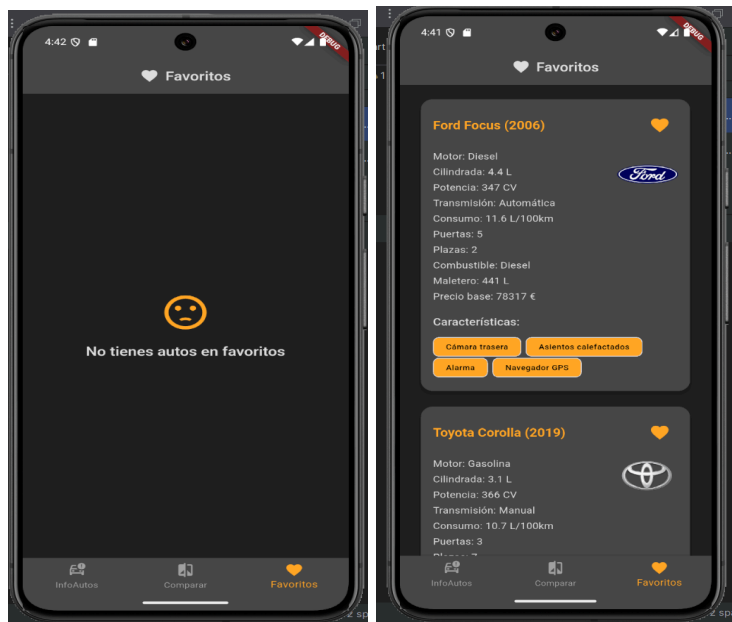
- Pantalla Inicio:



- Pantalla comparador:



- Pantalla favoritos:



## CODIFICACIÓN

Para el desarrollo del proyecto **InfoAutos**, se han seleccionado tecnologías modernas, multiplataforma y ampliamente utilizadas en el desarrollo móvil actual:

- **Lenguaje principal: Dart**
  - Elegido por ser el lenguaje utilizado por el framework Flutter.
  - Sencillo, con tipado estático, y gran compatibilidad con entornos multiplataforma.
- **Framework: Flutter**
  - Permite crear aplicaciones nativas tanto para Android como iOS desde una única base de código.
  - Alta productividad y personalización en el diseño UI/UX.
  - Amplia comunidad y gran soporte de documentación.
- **Gestión de estado: Provider**
  - Librería oficial recomendada por Flutter para la gestión del estado de la aplicación.
  - Permite una gestión clara y reactiva del estado global como los coches, favoritos o comparador.
- **Persistencia local: Sqflite + SharedPreferences**
  - **Sqflite**: para almacenar los datos de los coches en una base de datos SQLite local.
  - **SharedPreferences**: para guardar los favoritos del usuario sin necesidad de persistirlos en una tabla extra.
- **Diseño visual: Material Design**
  - Se ha seguido la guía de diseño Material para asegurar una experiencia coherente, intuitiva y atractiva.

- **Seguridad:**

Aunque se trata de una aplicación sin backend ni datos sensibles, se han tomado algunas medidas básicas:

- No se guarda información personal del usuario.
- La persistencia local se limita a datos técnicos del vehículo.
- La gestión del estado está encapsulada mediante **Provider** y no se expone nada innecesario.
- Se evita cualquier tipo de conexión remota que pueda comprometer la seguridad del dispositivo.

- **Desarrollo multiplataforma:**

Gracias a Flutter, el proyecto está preparado para funcionar tanto en **Android** como en **iOS**, con una única base de código.

Se han seguido prácticas compatibles con ambas plataformas, como:

- Interfaces responsive adaptadas a diferentes tamaños de pantalla.
- Uso de widgets estándar para garantizar compatibilidad nativa.
- No se han utilizado paquetes dependientes de Android/iOS específicos.

## DOCUMENTACIÓN

### - Introducción a la aplicación:

*InfoAutos* es una aplicación móvil desarrollada con el objetivo de facilitar al usuario la consulta y comparación de vehículos mediante una interfaz intuitiva y moderna. La aplicación permite visualizar especificaciones técnicas, características clave y precios base de diversos modelos de coches.

La base de datos de vehículos está integrada localmente en la aplicación utilizando SQLite. Una de las funcionalidades principales es la posibilidad de marcar vehículos como favoritos. Estos se guardan localmente mediante SharedPreferences, asegurando que las preferencias del usuario se mantengan tras cerrar la aplicación o reiniciar el dispositivo.

Además, se incluye un comparador que permite enfrentar dos vehículos seleccionados, mostrando sus atributos en paralelo. Esta herramienta es especialmente útil para evaluar diferencias técnicas y tomar decisiones más informadas.

La aplicación ha sido desarrollada en **Flutter**, lo que permite una experiencia fluida y adaptable en distintos dispositivos Android. La gestión del estado se realiza mediante el patrón **Provider**, facilitando el mantenimiento y la escalabilidad del código.

Desde el punto de vista visual, la aplicación utiliza un diseño basado en tarjetas, con una estética clara y moderna. Se ha añadido el logotipo de la marca de cada coche, aportando identidad visual sin necesidad de cargar imágenes complejas.

## - **Manual de instalación:**

Este manual describe el proceso necesario para instalar la aplicación *InfoAutos* en un dispositivo Android. Está orientado a usuarios con conocimientos básicos en el uso de smartphones.

### Requisitos previos

- Dispositivo móvil con sistema operativo Android 7.0 (Nougat) o superior.
- Al menos 50 MB de espacio libre de almacenamiento interno.
- Opción de instalación de aplicaciones desde orígenes desconocidos activada (solo en caso de instalar el archivo APK manualmente).

### Instalación desde archivo APK (recomendado para pruebas o distribución manual)

1. Descargar el archivo *infoAutos.apk* desde el repositorio proporcionado o desde un medio de confianza (por ejemplo, Google Drive, GitHub Releases, etc.).
2. En el dispositivo Android, localizar el archivo descargado usando un explorador de archivos.
3. Pulsar sobre el archivo APK. El sistema solicitará permiso para instalar aplicaciones desde esta fuente (si aún no está habilitado).
4. Conceder permiso temporal o permanente para esta fuente.
5. Pulsar en “Instalar” y esperar a que finalice el proceso.
6. Una vez instalada, pulsar en “Abrir” para iniciar la aplicación, o encontrarla posteriormente en el menú de aplicaciones bajo el nombre *InfoAutos*.

## - **Manual de Usuario:**

### **Descripción General:**

La aplicación **InfoAutos** permite a los usuarios consultar, comparar y gestionar información técnica sobre distintos vehículos. Su interfaz está diseñada para ser clara y funcional, tanto para aficionados al motor como para usuarios generales que deseen explorar o comparar modelos. El usuario puede marcar vehículos como favoritos y acceder rápidamente a ellos desde una pestaña específica.

### **Requisitos Previos:**

- Dispositivo con sistema operativo **Android 7.0 o superior**.
- Aplicación instalada correctamente (ver Manual de Instalación).

### **Navegación Principal:**

La app se organiza mediante una barra inferior con tres secciones:

- **Inicio:** Lista completa de vehículos disponibles.
- **Comparador:** Herramienta para seleccionar y comparar dos vehículos.
- **Favoritos:** Sección personalizada con los vehículos marcados como favoritos.

### **Busqueda de vehiculos:**

En las secciones de inicio y comparador, el usuario puede utilizar el buscador integrado:

- Al pulsar el botón de selección, se abre un buscador por marca o modelo.
- Los resultados incluyen el **logo del fabricante, modelo, tipo de motor y transmisión**.



## Funcionalidades:

### Inicio

- Presenta una **lista resumida de vehículos**.
- Al pulsar sobre una tarjeta, se accede a una **ficha completa del vehículo** con información detallada: motor, potencia, consumo, características, etc.
- Cada tarjeta incluye un **botón de favorito (icono de corazón)**. Al pulsarlo, el coche se añade o elimina de la lista de favoritos.

### Comparador

- Permite **seleccionar dos coches diferentes** mediante un buscador.
- Muestra sus fichas completas **lado a lado**.
- Incluye un botón para **quitar** cada coche del comparador de forma independiente.

### Favoritos

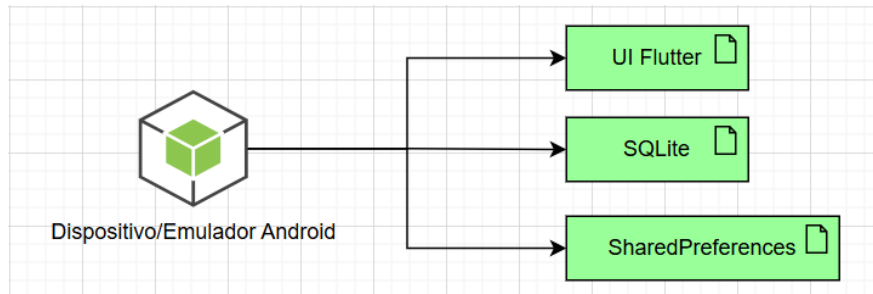
- Muestra una **lista personalizada** con los vehículos que el usuario ha marcado.
- También ofrece la posibilidad de consultar los detalles completos y eliminar vehículos de la lista con un solo toque.

### Guardado de favoritos

Los vehículos favoritos se almacenan en el dispositivo mediante **SharedPreferences**. Esta información **se conserva incluso tras cerrar o reiniciar la aplicación**.

## DESPLIEGUE

### - Diagrama de despliegue:



### - Descripción de la instalación:

La aplicación **InfoAutos** se ha desarrollado utilizando **Flutter**, lo que permite compilarla como una aplicación nativa para Android en formato **.apk**.

#### Pasos de despliegue en entorno local:

##### 1. Generación del APK:

- Desde Android Studio, se utiliza la opción **Build > Build Bundle(s) / APK(s) > Build APK(s)**.

##### 2. Instalación en un dispositivo Android:

- El archivo .apk puede transferirse al dispositivo mediante USB, correo electrónico o servicios en la nube (Drive, etc.).
- En el dispositivo móvil, se debe habilitar la opción **"Instalar aplicaciones de fuentes desconocidas"**.
- Al abrir el archivo .apk, se lanza la instalación.

##### 3. Ejecutar la app:

- Una vez instalada, la app aparece en el menú principal del dispositivo bajo el nombre **"InfoAutos"**.

## HERRAMIENTAS DE APOYO

- **Android Studio**
- **Flutter SDK**
- **Dart**
- **Git (Local)**
- **SQLite**
- **SharedPreferences**
- **Emulador Android Studio**

## CONCLUSIONES

El proyecto **InfoAutos** ha cumplido de forma satisfactoria los objetivos marcados al inicio del desarrollo. Se ha implementado una aplicación funcional que permite consultar una base de datos de vehículos, compararlos y gestionar favoritos, todo ello con una interfaz intuitiva y adaptada a distintos dispositivos Android.

Durante el proceso se han abordado aspectos clave como el diseño UX, la gestión del estado con Provider, la persistencia de datos con SQLite y la organización modular del código, lo que ha facilitado un desarrollo estructurado y mantenible.

Una de las limitaciones encontradas fue la imposibilidad de integrar APIs externas, debido a que la mayoría de servicios disponibles eran de pago. Esta circunstancia obligó a optar por una base de datos local integrada, lo cual, si bien limita la actualización automática de los datos, garantiza el funcionamiento autónomo de la aplicación sin requerir conexión.

A pesar de ello, el sistema está preparado para futuras ampliaciones. Entre las posibles mejoras que se podrían implementar destacan:

- **Integración de una API externa** (gratuita o mediante suscripción) para obtener información actualizada de vehículos.
- **Sistema de registro e inicio de sesión** para personalizar la experiencia del usuario.
- **Expansión del catálogo** de vehículos, incluyendo más marcas, imágenes y características.
- Añadir funciones como **filtros avanzados**, **notificaciones** o **favoritos sincronizados entre dispositivos**.

En resumen, InfoAutos ha demostrado ser un proyecto viable y funcional. Ha servido como ejercicio práctico de desarrollo móvil con un lenguaje que no conocía y como base sólida para futuras mejoras más ambiciosas.

## BIBLIOGRAFÍA

- **Curso Flutter y Dart Aristidevs:**  
 **FLUTTER y DART desde CERO**  **Curso Completo en ESPAÑOL - Android & i...**
- Documentación oficial Flutter: <https://docs.flutter.dev>
- Documentación Provider: <https://pub.dev/packages/provider>
- StackOverflow: <https://stackoverflow.com/>
- Chat GPT: Para generar el archivo JSON y corregir ciertos errores.