Marzo, 22 2025.

**MANUAL TECNICO PADILLAROUTE.**

v 1.0.0

TABLA DE CONTENIDO

[**INTRODUCCION** 4](#_Toc193796873)

[**CAPITULO 1: ARQUITECTURA.** 5](#_Toc193796874)

[**A) PATRON ARQUITECTONICO.** 5](#_Toc193796875)

[**CAPITULO 2: BASE DE DATOS.** 7](#_Toc193796876)

[**B) REALTIME DATABASE.** 7](#_Toc193796877)

[**C) OBJECTBOX STORE.** 7](#_Toc193796878)

[**CAPITULO 3: BACKEND.** 7](#_Toc193796879)

[**A) APIS.** 7](#_Toc193796880)

[**i.** **MAPS.** 7](#_Toc193796881)

[**ii.** **GEOLOCATOR.** 7](#_Toc193796882)

[**B) MODELOS.** 7](#_Toc193796883)

[**i.** **USUARIOS.** 7](#_Toc193796884)

[**ii.** **VEHICULOS.** 7](#_Toc193796885)

[**iii.** **PARADAS.** 7](#_Toc193796886)

[**iv.** **RUTAS.** 7](#_Toc193796887)

[**v.** **VIAJES.** 7](#_Toc193796888)

[**vi.** **INCIDENTES.** 7](#_Toc193796889)

[**vii.** **LOGS.** 7](#_Toc193796890)

[**C) SERVICIOS.** 7](#_Toc193796891)

[**i.** **CONECTOR REALTIME DATABASE.** 7](#_Toc193796892)

[**ii.** **CONECTOR OBJECTBOX STORE.** 7](#_Toc193796893)

[**iii.** **AUTENTICACIÓN.** 7](#_Toc193796894)

[**iv.** **NOTFICACIÓNES.** 7](#_Toc193796895)

[**v.** **GESTION DE USUARIOS.** 7](#_Toc193796896)

[**vi.** **GESTION DE VEHICULOS.** 7](#_Toc193796897)

[**vii.** **GESTION DE PARADAS.** 7](#_Toc193796898)

[**viii.** **GESTION DE RUTAS.** 7](#_Toc193796899)

[**ix.** **GESTION DE VIAJES.** 8](#_Toc193796900)

[**x.** **GESTION DE INCIDENTES.** 8](#_Toc193796901)

[**xi.** **LOGS DEL SISTEMA.** 8](#_Toc193796902)

[**CAPITULO 4: INTERFAZ (FRONTEND).** 8](#_Toc193796903)

[**A) PANTALLAS.** 8](#_Toc193796904)

[**B) WIDGETS PERSONALIZADOS.** 8](#_Toc193796905)

[**CAPITULO 5: SEGURIDAD.** 8](#_Toc193796906)

[**A) AUTENTICACIÓN DE IDENTIDAD DE USUARIOS.** 8](#_Toc193796907)

[**B) BUENAS PRACTICAS.** 8](#_Toc193796908)

[**i.** **ENCRIPTACION DE CONTRACSEÑAS.** 8](#_Toc193796909)

[**C) PERMISOS**. 8](#_Toc193796910)

# **INTRODUCCION**

Este manual técnico describe el diseño, arquitectura y funcionamiento de [nombre de la aplicación], una aplicación desarrollada en **Flutter** con almacenamiento local en **ObjectBox** y sincronización con Firebase Realtime **Database**.

El objetivo de esta aplicación es [breve descripción del propósito de la app]. Para lograrlo, se ha diseñado con una arquitectura modular y escalable, permitiendo una fácil expansión y mantenimiento.

A lo largo de este documento, se detallarán los componentes clave**, la estructura de la base de datos, las interacciones con Firebase y las mejores prácticas utilizadas en su desarrollo**. Este manual está dirigido a desarrolladores, ingenieros de software y cualquier persona interesada en comprender el funcionamiento interno de la aplicación.

***Nota:******se presupone que quien esté leyendo esto tiene, de antemano, conocimientos en el stack mencionado anteriormente y usado para este proyecto. Por lo cual, no nos centraremos en explicar cosas como la estructura de un proyecto de Flutter, qué es Firebase, patrones de diseño, etc.***

# **ARQUITECTURA.**

## **PATRON ARQUITECTONICO.**

Este proyecto utiliza una versión personalizada del patrón de diseño arquitectónico **Modelo-Vista-Controlador**; dentro de lo poco que se personalizó se encuentran los nombres de los componentes: **Modelo-Pantalla-Servicio**.

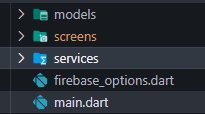


Imagen 1.1 Patrón arquitectónico de la aplicación.

* **Models** contiene los modelos de datos necesarios para trabajar con las bases de datos local y remota (Realtime Database y ObjectBox Store respectivamente).
* **Screens**, justo como lo están pensado, contiene las pantallas y widgets utilizados para construir la interfaz gráfica de la aplicación.
* **Services** guarda toda la lógica de negocios; conectores a bases de datos, servicios para realizar operaciones CRUD, servicios de autenticación de identidad, generadores de archivos, entre otros. Mas adelante se explorará a profundidad.

## **FLUJO DE DATOS.**

Revisemos cómo funciona el sistema. Se pueden distinguir tres actores distintos: un encargado de la administración de recursos como usuarios, vehículos, reportes, rutas, etc; otro cuya tarea es llevar el registro de los viajes en tiempo real; finalmente la base de datos remota, que es el principal centro de almacenamiento de datos.

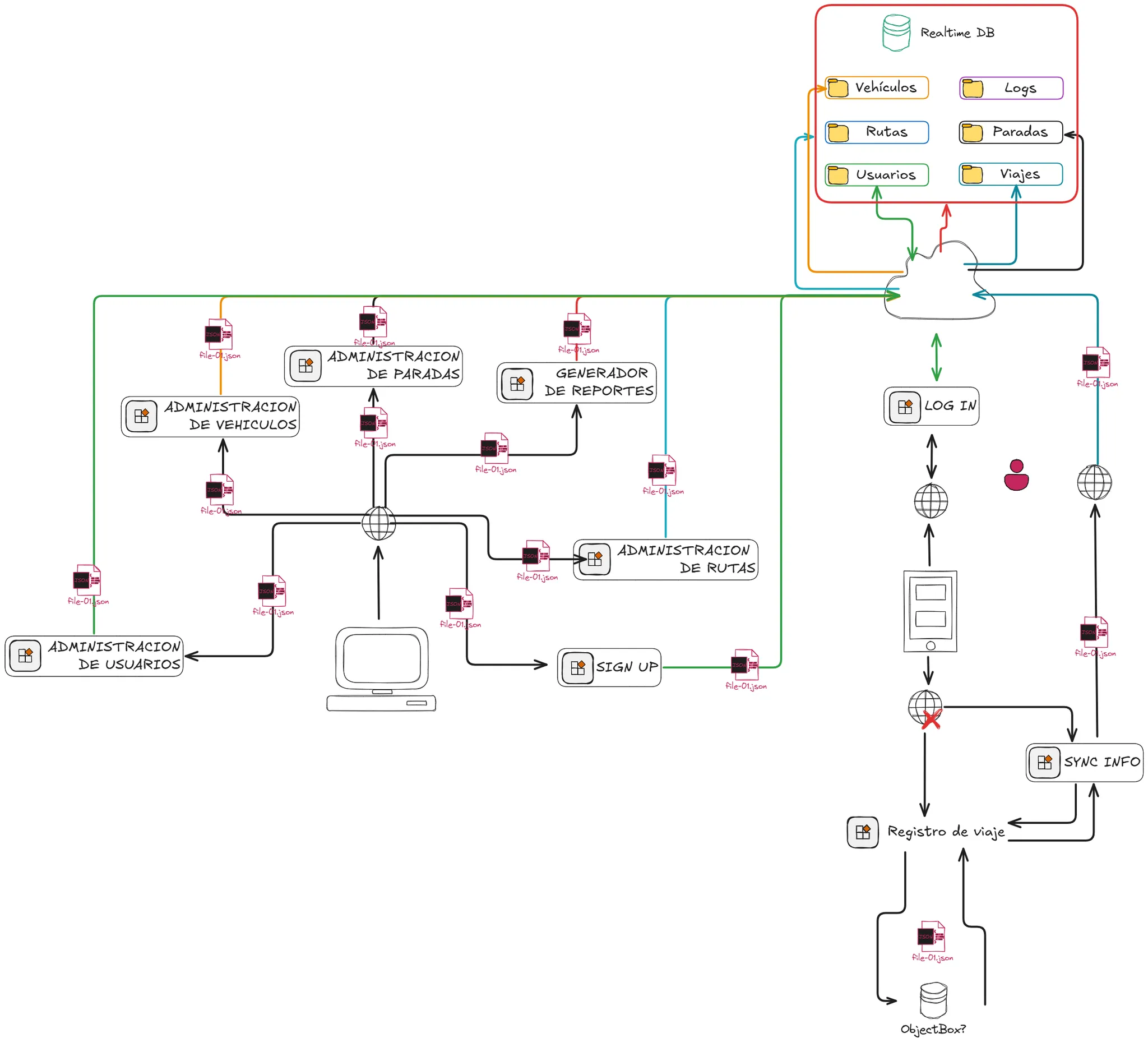


Imagen 1.2 Flujo de datos del sistema.

# **BASE DE DATOS.**

## **REALTIME DATABASE.**

## **OBJECTBOX STORE.**

# **BACKEND.**

## **APIS.**

### **MAPS.**

### **GEOLOCATOR.**

## **MODELOS.**

### **USUARIOS.**

### **VEHICULOS.**

### **PARADAS.**

### **RUTAS.**

### **VIAJES.**

### **INCIDENTES.**

### **LOGS.**

## **SERVICIOS.**

### **CONECTOR REALTIME DATABASE.**

### **CONECTOR OBJECTBOX STORE.**

### **AUTENTICACIÓN.**

### **NOTFICACIÓNES.**

### **GESTION DE USUARIOS.**

### **GESTION DE VEHICULOS.**

### **GESTION DE PARADAS.**

### **GESTION DE RUTAS.**

### **GESTION DE VIAJES.**

### **GESTION DE INCIDENTES.**

### **LOGS DEL SISTEMA.**

# **INTERFAZ (FRONTEND).**

## **PANTALLAS.**

## **WIDGETS PERSONALIZADOS.**

# **SEGURIDAD.**

## **AUTENTICACIÓN DE IDENTIDAD DE USUARIOS.**

## **BUENAS PRACTICAS.**

### **ENCRIPTACION DE CONTRACSEÑAS.**

## **PERMISOS**.

TABLA DE IMAGENES.

[Imagen 1.1 Patrón arquitectónico de la aplicación. 5](#_Toc193797651)

[Imagen 1.2 Flujo de datos del sistema. 6](#_Toc193797652)