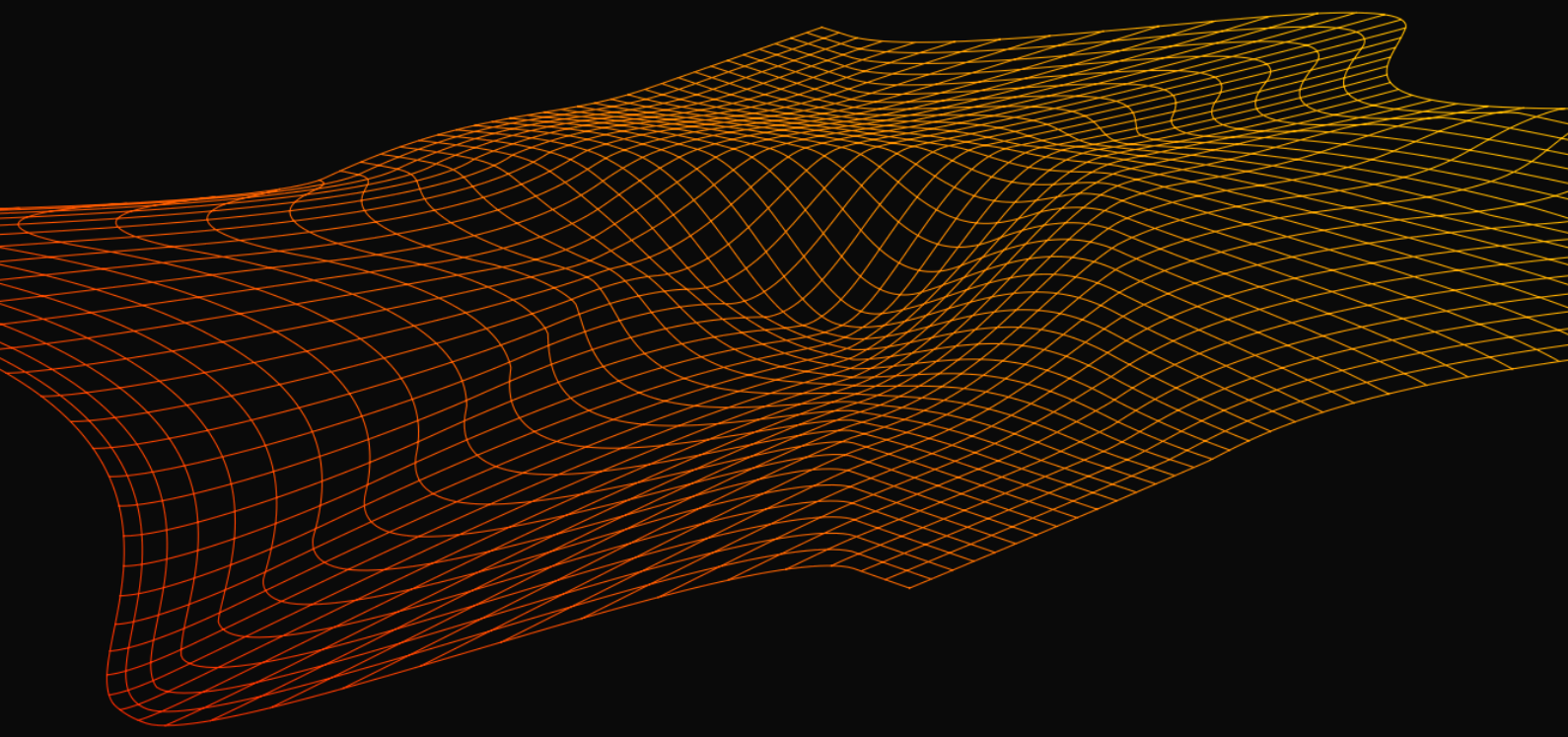
Programación de Servicios y Procesos

Proyecto Carfix upkeep



PSP - 2º DAM

CPIFP Los Enlaces

Creado por: Jorge Andrés Arroyo Celis

Fecha: 31/02/2025

Ciclo Formativo: 2º DAM-D 2024-25

Contenido

Programación de Servicios y Procesos

Práctica - Proyecto de Aplicación Carfix upkeep

Resumen Ejecutivo

Introducción

El proceso para hallar una falla en un coche es muy extenso e ineficiente, por ello, hemos desarrollado una aplicación móvil para Android que permite consultar y gestionar estos códigos de forma rápida y eficiente, utilizando Java y Firebase.

Propuesta de solución

La app ofrece una base de datos con códigos de avería, proporcionando descripciones y soluciones en tiempo real gracias a Firebase Firestore. Diseñada en Android Studio, permite búsquedas rápidas y acceso en la nube. Ademas cuenta con una sincronizacion en segundo plano.

Valor y mercado objetivo

Dirigida a mecánicos, técnicos y propietarios de vehículos, esta herramienta solventa el uso del scanner y/o ayuda del mismo.

Ventajas competitivas

- Actualización en tiempo real mediante Firebase.
- Interfaz intuitiva para una navegación eficiente.
- Accesibilidad y portabilidad en dispositivos Android.
- Uso offline gracias a workmanager que controla servicios en segundo plano. Además se guardan datos en cache

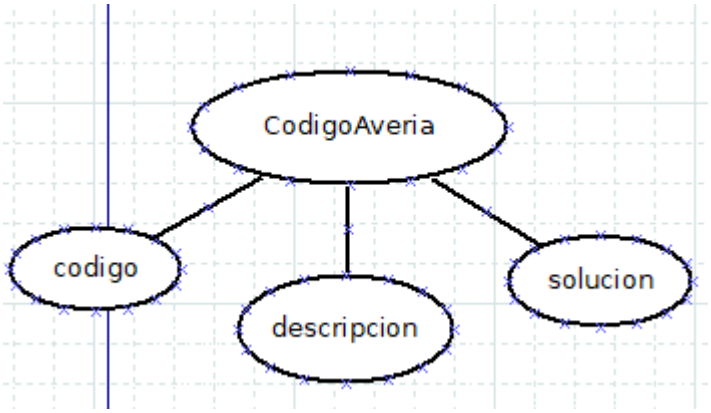
Próximos pasos

Realizaremos pruebas con usuarios para optimizar la experiencia y exploramos mejoras como diagnósticos guiados. Esta aplicación busca revolucionar el acceso a la información sobre códigos de avería, agilizando el proceso de reparación.

Análisis y diseño

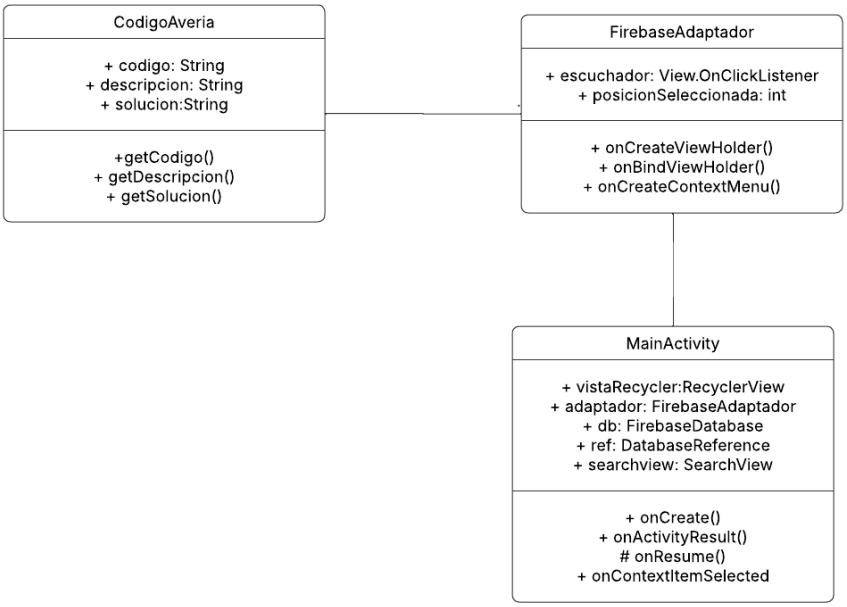
4.1 Modelado de datos. Análisis y diseño de la base de datos

4.1.1 Diagrama E/R

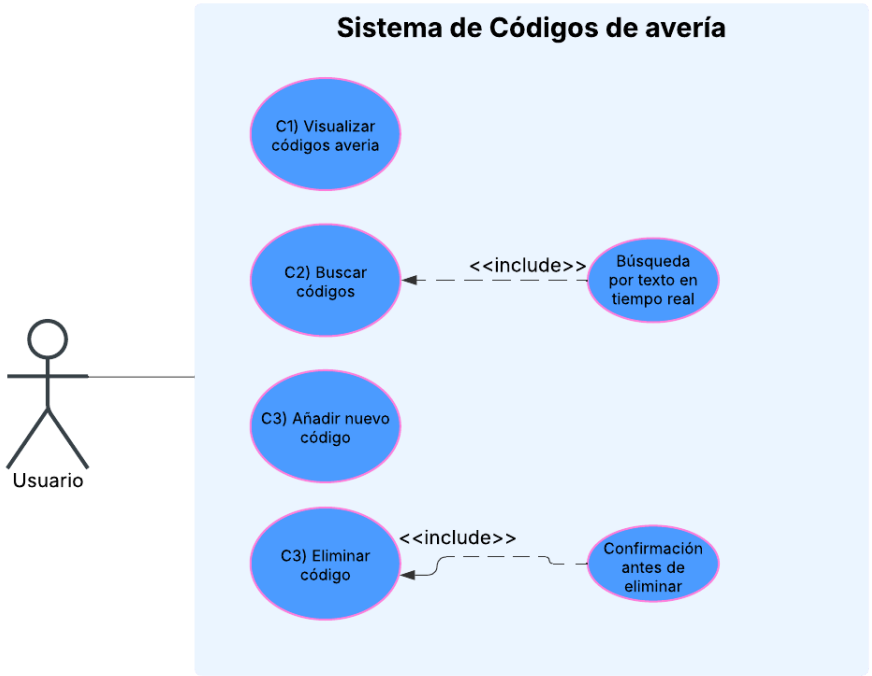


En este caso el modelo que se ha utilizado ha sido el NoSQL por lo tanto la relación que tiene con otra entidad es nula, aún así esa es la representación, cada Código Avería es una entrada independiente en Firestore.

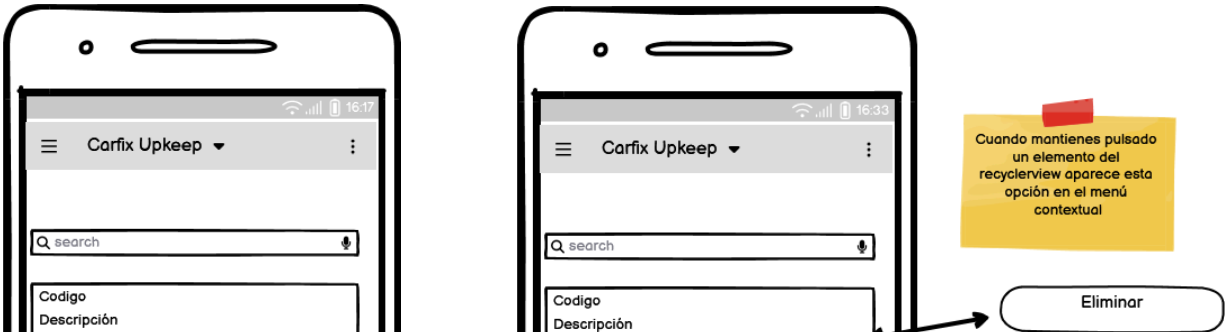
4.2 Diagrama de clases

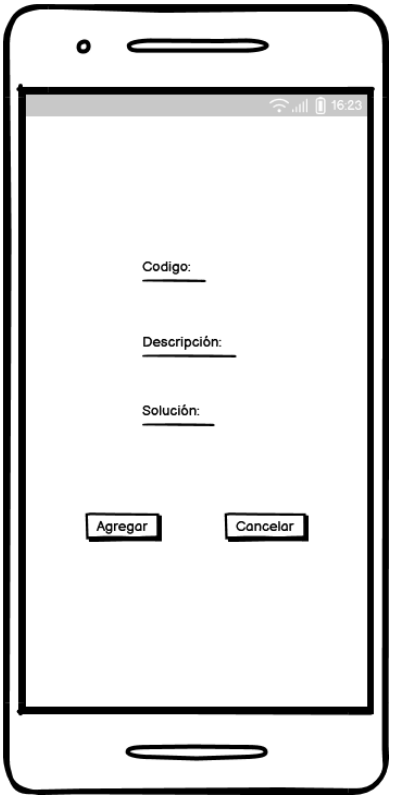


4.2 Diagrama de casos de uso



4.3 Análisis y diseño de la interfaz de usuario. Mockups.





4.4 Tecnologías/Herramientas usadas y descripción de las mismas

A lo largo del desarrollo de la aplicación, se han utilizado diversas herramientas y tecnologías. A continuación, se detallan las principales herramientas empleadas:

DIA (Diagramas)

Ha permitido diseñar la estructura de la base de datos y visualizar las relaciones entre las entidades de la aplicación.

Balsamiq Mockups

Se ha usado para la planificación visual clara de la experiencia de usuario (UX) y la disposición de los elementos de la interfaz (UI).

Android Studio

entorno de desarrollo que ha permitido realizar pruebas en emuladores y dispositivos físicos, optimizando el rendimiento y la experiencia de usuario.

Firebase

Se ha utilizado Firebase Firestore como base de datos NoSQL para almacenar y gestionar la información de los códigos de avería de los vehículos. Su capacidad en tiempo real y su integración sencilla con Android han facilitado la gestión eficiente de los datos.

Lucidchart

Se ha utilizado en el proyecto para diseñar diagramas de clases UML y la arquitectura de la aplicación, permitiendo una mejor planificación y documentación del sistema antes de su desarrollo.

4.4.2 Arquitectura de componentes de la aplicación

La arquitectura de la aplicación sigue el patrón Modelo-Vista-Controlador (MVC). A continuación, se detallan los principales componentes:

Cliente

Vista:

MainActivity: Pantalla principal con RecyclerView y SearchView

AddCodigo: Pantalla para añadir nuevos códigos

Layouts XML: Estructura visual

Menús: Principal y contextual para operaciones como añadir/eliminar

Controladores:

Activities: Manejan la lógica de presentación y respuesta a eventos

FirebaseAdaptador: Conecta los datos con la vista mediante RecyclerView

DescargarCodigosW: Se encarga de los procesos en segundo plano para la descarga de codigos de averia

Modelo:

CodigoAveria: Representa la estructura de datos

Servidor (Firebase)

Firebase Realtime Database:

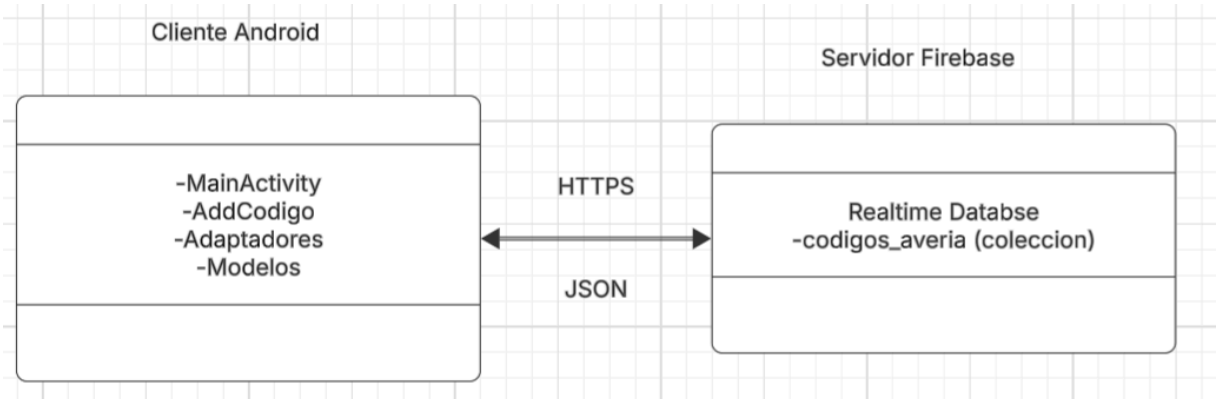
Almacena los "codigos_averia" en formato JSON

URL de referencia: "https://codigosaveriatfg-default-rtdb.europe-west1.firebasedatabase.app"

Servicios Firebase:

API REST para operaciones CRUD

Sistema de sincronización en tiempo real



Conclusiones

Cada vez que encontramos un proceso y este se puede acortar la mayor cantidad posible, ahí hemos dado en el clavo, como lo ha hecho esta app que facilita el diagnóstico y optimiza el mantenimiento de los vehículos, convirtiéndose en una herramienta clave tanto para mecánicos como para propietarios.

El impulso de Firebase ayuda a la administración de estos códigos lo que aumenta la respuesta y accesibilidad. Por último cabe aclarar que se han cumplido los objetivos propuestos y ha superado las expectativas, es un proyecto que cuenta con escalabilidad a largo plazo y posibilidad de funcionar offline.

Bibliografía/Webgrafía

- Asana. (2025). *Cómo redactar un resumen ejecutivo (incluye ejemplos)*. Recuperado el 20 de febrero de 2025, de Cómo redactar un resumen ejecutivo (incluye ejemplos) [2025] • Asana
- Cloud Firestore - Firebase Console. (s.f.). *CodigosAveriaTfg*. Recuperado el 26 de febrero de 2025, de CodigosAveriaTfg - Cloud Firestore - Firebase console
- Lucidchart. (s.f.). *Diagrama de clases avería*. Recuperado el 26 de febrero de 2025, de Diagrama de clases averia: Lucidchart
- Medium – Android Grid. (s.f.). *How to use FirebaseRecyclerAdpater with latest Firebase Dependencies in Android*. Recuperado el 26 de febrero de 2025, de How to use FirebaseRecyclerAdpater with latest Firebase Dependencies in Android | by Ankit Suda | Android Grid | Medium
- Peugeot. (s.f.). *Listado de Códigos de Avería Peugeot: Solución Rápida*. Recuperado el 26 de febrero de 2025, de ▷ Listado de Códigos de Avería Peugeot: Solución Rápida ✅
- Stack Overflow. (s.f.). *How to implement Firebase Recycler Adapter in newer version of Android 3.1 and higher?*. Recuperado el 26 de febrero de 2025, de java - How to implement Firebase Recycler Adapter in newer version of Android 3.1 and higher? - Stack Overflow
- profit.me. (2017). *Firebase for Android: Database tutorial*. Recuperado el 26 de febrero de 2025, de Firebase for Android: Database tutorial | en.proft.me
- Developer.android.(2024). *Arquitectura de apps: Capa de datos.* Recuperado el 26 de febrero de 2025, de Arquitectura de apps: Capa de datos - Cómo programar tareas con WorkManager - Android Developers | Android Developers