

1. Introducción

Este proyecto surge de la necesidad de proporcionar a los usuarios una forma sencilla de explorar productos, agregarlos a una lista compartida y almacenarlos en la nube. La aplicación está diseñada para facilitar la gestión de la lista de la compra familiar en un entorno digital, dejando atrás el papel y ofreciendo información detallada de cada artículo.

2. Análisis del problema

2.a Problemática

Los usuarios tienen dificultades para gestionar la lista de la compra de manera estructurada en notas físicas o digitales. Se necesita una lista compartida para poner fin a esto.

2.b Clientes potenciales

- Compradores frecuentes que desean organizar sus compras.
- Usuarios que investigan productos antes de comprarlos.
- Familias con varios miembros que usan listas conjuntas.

2.c Análisis DAFO

Debilidades:

- Dependencia de Firebase para el almacenamiento.
- Requiere conexión a internet para la sincronización.

Amenazas:

- Competencia con otras aplicaciones similares.

Fortalezas:

- Detalles alimenticios de cada producto.
- Sincronización en la nube.
- Interfaz intuitiva.

Oportunidades:

- Monetización mediante anuncios.
- Expansión a otros mercados y funcionalidades.

2.d Monetización y beneficios(Posibles implementaciones)

- **Publicidad:** Se pueden incluir anuncios que no molesten.
- **Modelo premium:** Acceso básico gratuito y premium con funciones adicionales.

3. Diseño de la solución

3.a Tecnologías elegidas

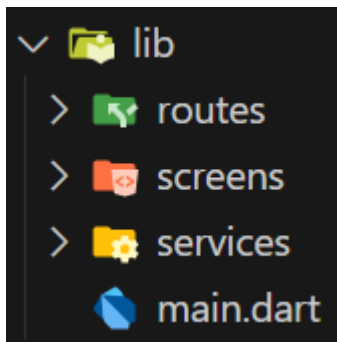
- **Flutter** para el desarrollo de la aplicación.
- **Dart** como lenguaje de programación.
- **Firebase** para almacenamiento en la nube.
- **Spoonacular API** para la obtención de productos.

3.b Arquitectura

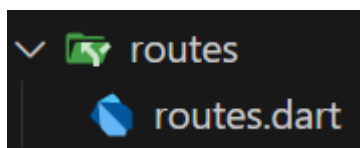
- **Modelo Cliente-Servidor:** La app se comunica con una API para obtener productos y usa Firebase para el almacenamiento.

3.c Diagrama de clases

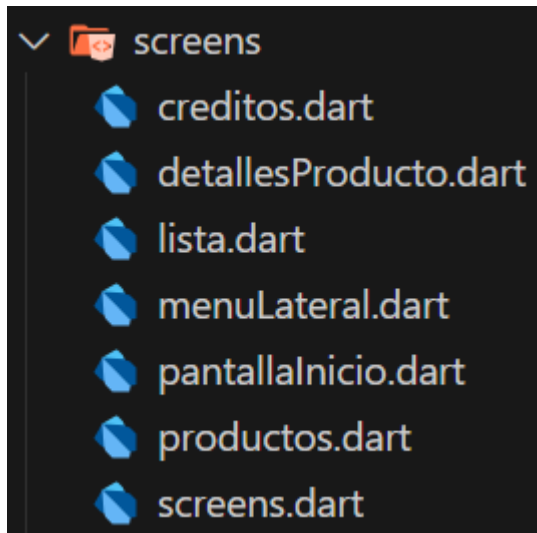
La estructura de clases se basa en la clase principal(main) y tres carpetas principales:



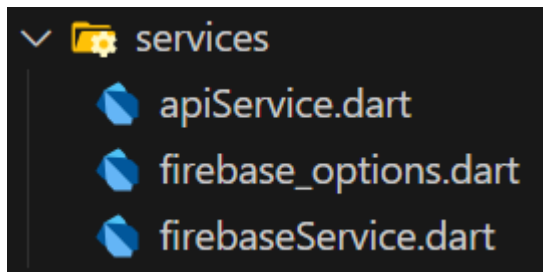
- **Routes:** contiene un archivo con las rutas nombradas de todas las pantallas.



- **Screens:** contiene todas las pantallas.



- **Services:** Contiene las clases donde se realiza la conexión a la API y a Firebase.



3.d Diagrama E/R

La estructura de la base de datos es simplemente una tabla 'productos' que contiene un array con los id's de los productos que se añaden a la lista.

4. Información importante.

- En el anteproyecto se especificaba que la aplicación iba a ser del supermercado 'Mercadona', pero finalmente no ha podido ser así por varios problemas con la API de éste supermercado y el resultado final se ha visto afectado minoritariamente. La API de la que se obtienen los productos no es de ningún supermercado real, ya que tampoco he encontrado ninguno que la ofrezca. Por ello, la API usada es principalmente de recetas de cocina, que contiene muchos ingredientes e información nutricional sobre ellos, desvinculando el proyecto de 'Mercadona'. De esta manera el resultado final es prácticamente lo que se planteó, quizás un poco menos útil por el hecho de que no estás tratando con productos de un supermercado real, pero ya no había manera de replantear el proyecto y creo que así he solucionado de una manera bastante buena el problema que se me planteó.
- Los productos están en inglés, por lo que al buscarlos deberás tenerlo en cuenta aunque no tienes que introducir los nombres completos, con una o varias letras se muestran resultados igualmente.

