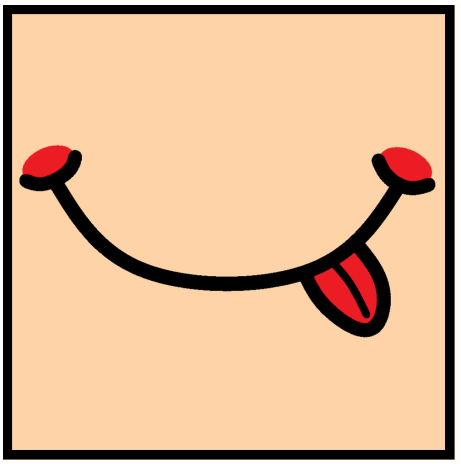
Hambrout

Manual técnico



Índice

Introducción

Hambrout es una aplicación que permite explorar lugares de comida de interés cerca del usuario, además de ofrecer múltiples recetas por si prefiere quedarse en casa y creación de listas tales como la de la compra o sus bebidas favoritas.

A la hora de ver lugares cercanos, el usuario puede elegir entre 4 tipos de establecimientos de comida tales como restaurantes o bares. El radio de distancia es modificable. Una vez que tenga los lugares podrá ver su calificación sobre 5, la distancia andando y en coche, dirección y si está abierto.

Si hablamos de las recetas, el cliente podrá filtrarlas por origen y podrá añadirlas a favoritos. Éstas se pueden consultar en una pestaña a parte. De cada receta podrá ver su dificultad, origen, tipo de plato (entrante, principal o postre), número de comensales, tiempo de preparación, ingredientes y elaboración.

Análisis del problema

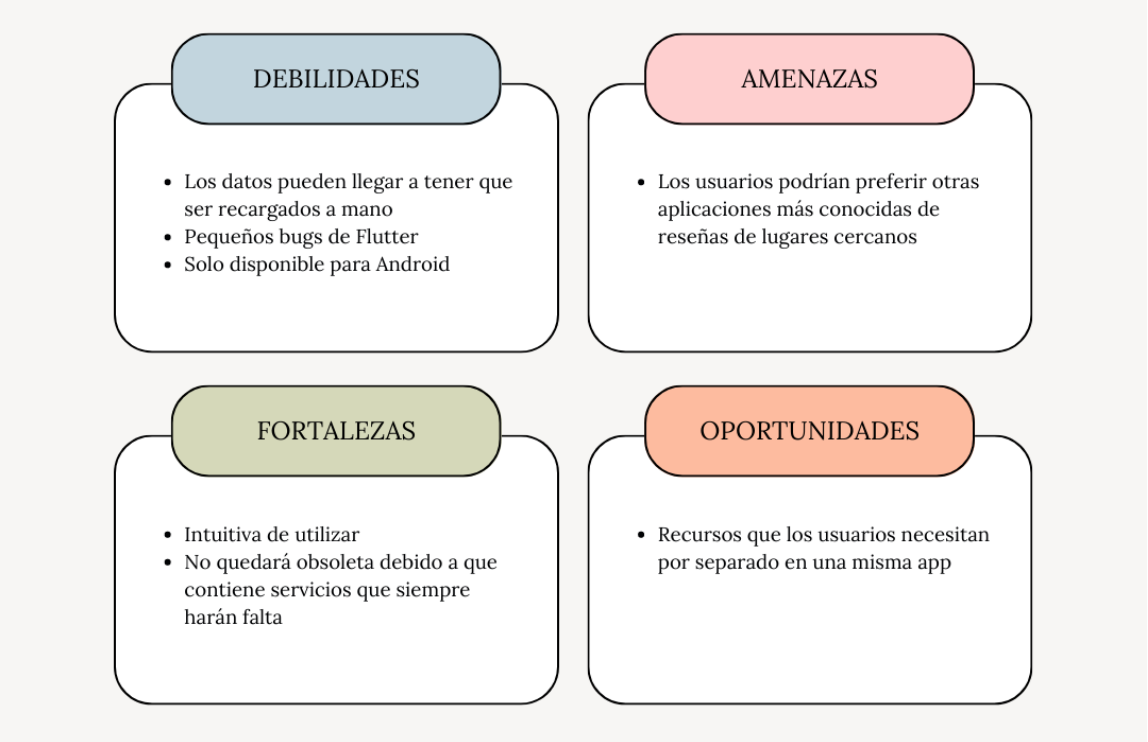
Problemática

Obtener una aplicación donde rápidamente se pueda ver lugares para comer en un rango determinado, además de recetas accesibles y de fiabilidad.

Clientes potenciales

Usuarios que estén de turismo en una ciudad o simplemente no se decidan. También personas que busquen recetas para comer en casa.

Análisis DAFO



Monetización y beneficios

En este momento la aplicación no permite monetización ni hay un servicio que lo pueda justificar. Sin embargo, se plantea añadir pequeños anuncios a lo largo de la aplicación y una suscripción mensual con beneficios tales como servicio a domicilio desde los locales que lo permitan tramitado desde la aplicación o una opción que permita dar un resultado aleatorio en las listas por si el usuario está indeciso.

Diseño de la solución

Tecnologías elegidas

Han sido las siguientes:

- **Lenguaje:** Dart
- **Framework:** Flutter
- **IDE:** Android Studio
- **Guardado de datos:** Firebase Firestore
- **API:** Google Cloud

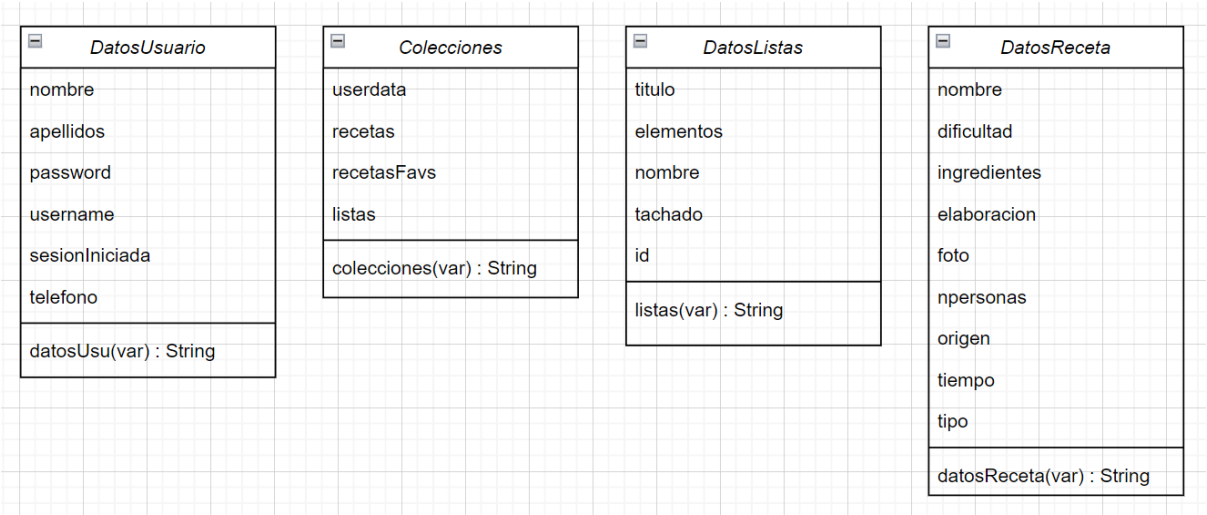
Arquitectura

Cliente - Servidor

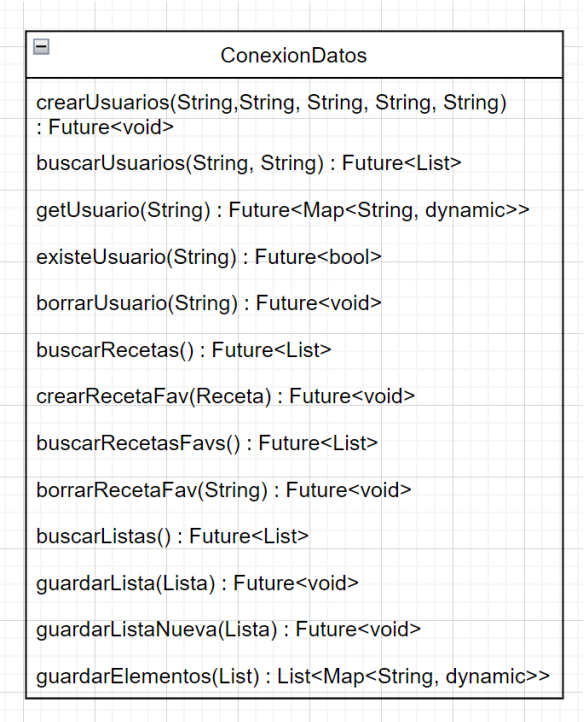
Diagrama de clases

❖ Tipos enumerados

Cada clase *enum* contiene los nombres de los campos de los documentos de Firebase y su método devuelve un string con el valor del nombre del campo.



❖ Conexión con firebase

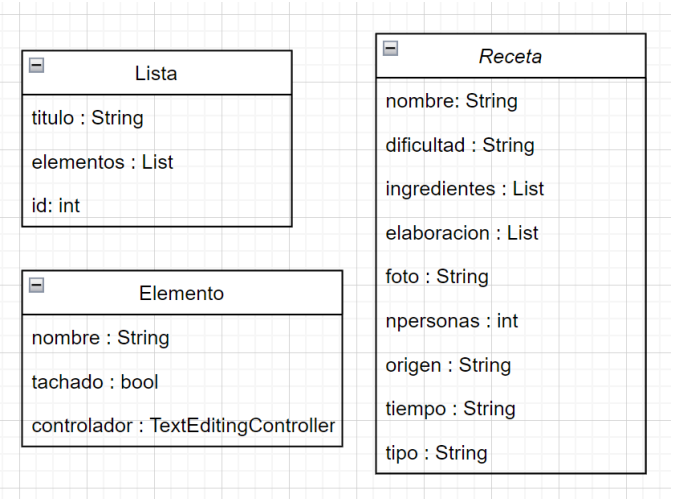


Esta clase contiene los métodos que conectan con Firebase Firestore.

❖ Modelos

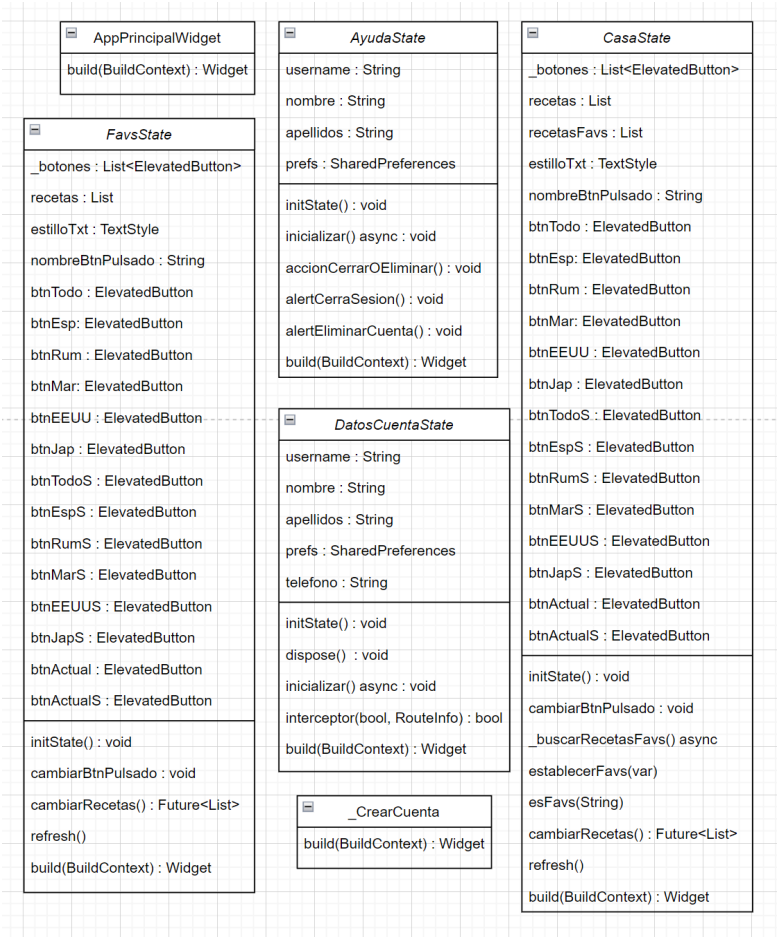
Estas clases contienen los atributos de los modelos de lista, elemento y receta.

Elemento es un modelo a parte que determina los atributos para los elementos de las listas. Es una clase a parte debido a que cada elemento es un objeto de tipo Map.



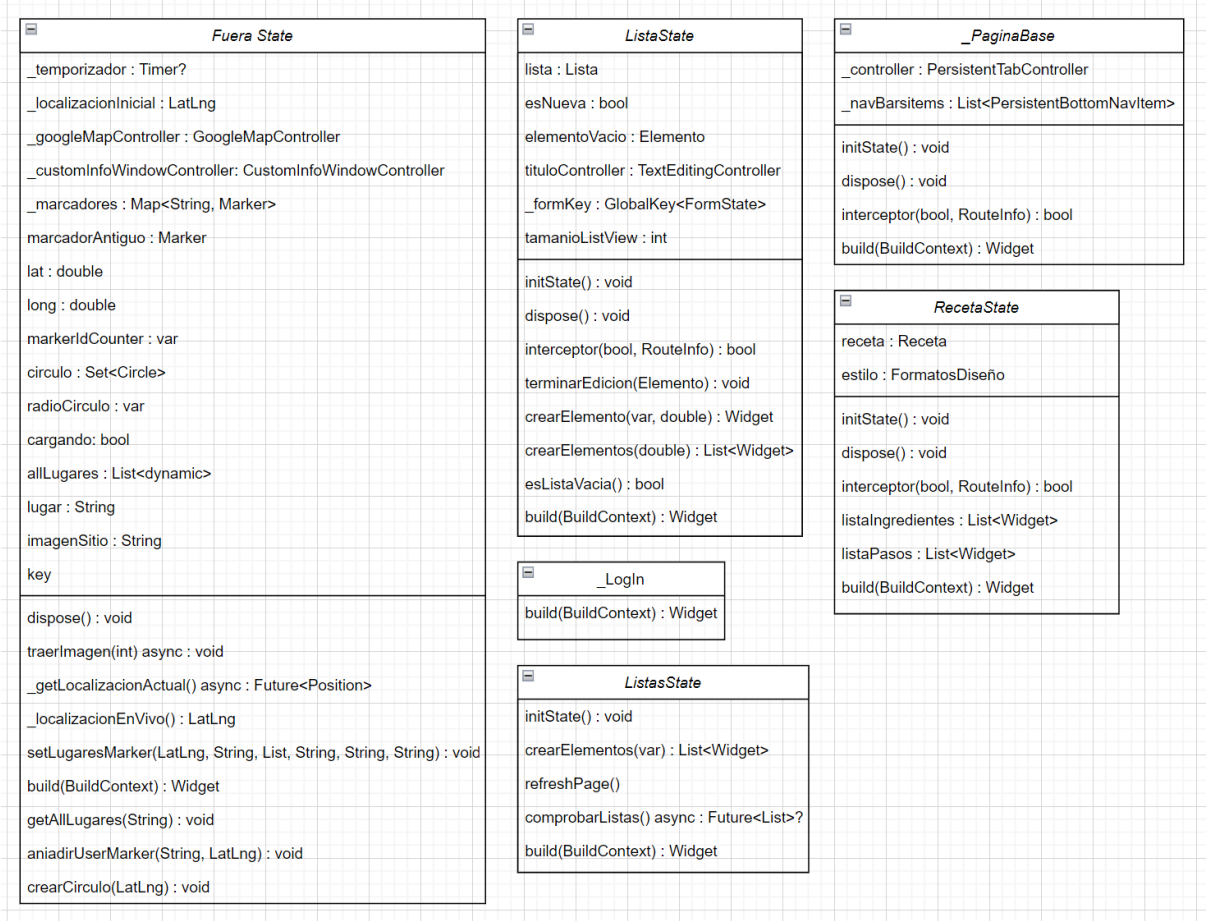
❖ Vistas

(App principal, Ayuda, Casa, Crear cuenta, Datos cuenta, Favs)



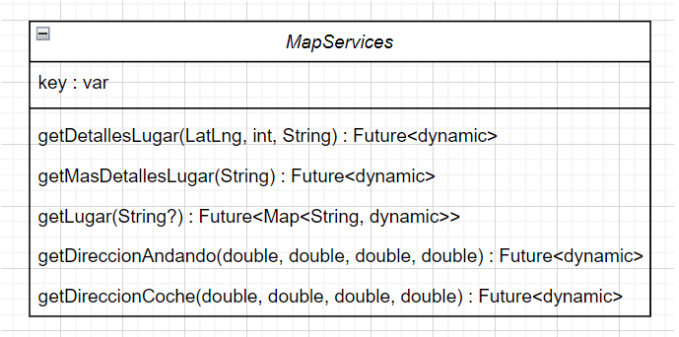
Las clases de vistas contienen todos los elementos necesarios para construir la visual. Cada clase extiende de la clase State<T>. T es un widget creado para cada State que extiende de StatelessWidget.

(Fuera, Lista, Listas, LogIn, Página base, Receta)



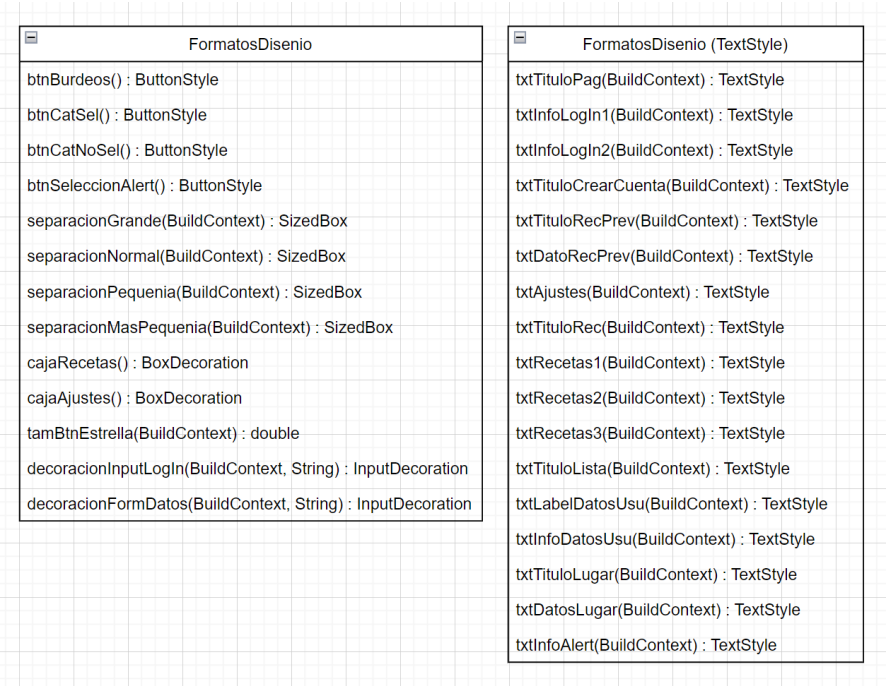
❖ Servicio de Maps

Esta clase contiene los métodos para hacer las llamadas a la API. El atributo de clase “key” es la clave de la API.

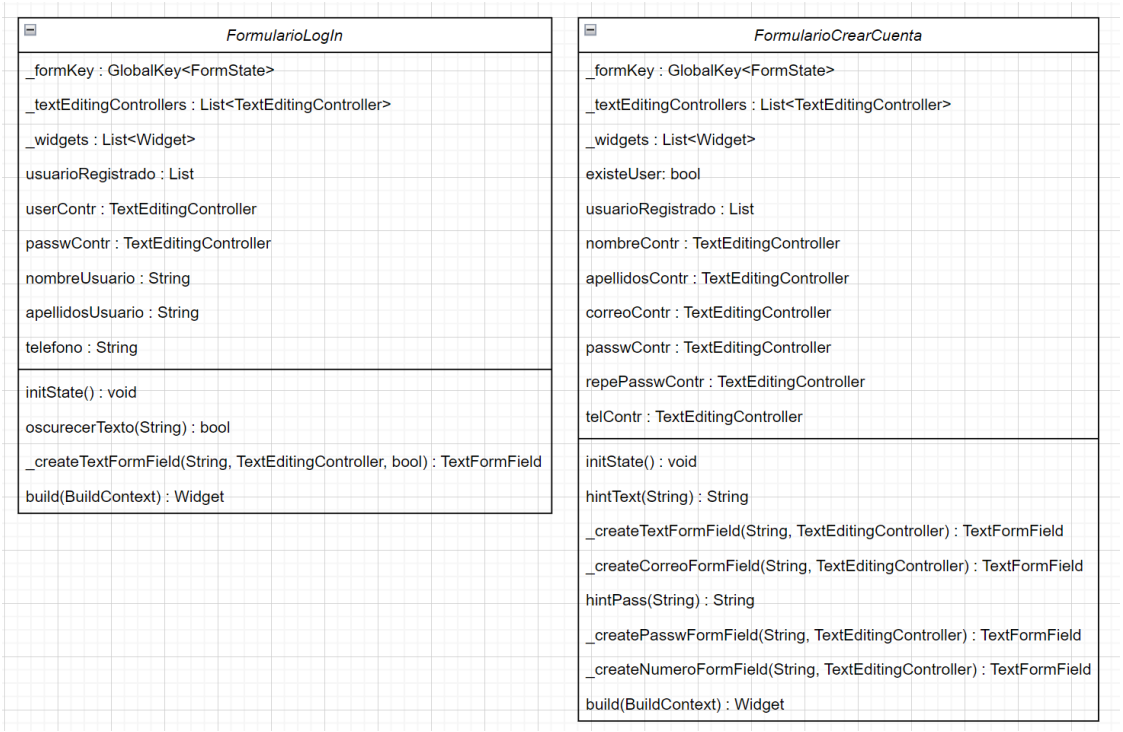


❖ Utils

La clase FormatosDisenio alberga casi todos los estilos de los widget, es decir, tiene todos los estilos que son más complejos o se usan más de una vez.



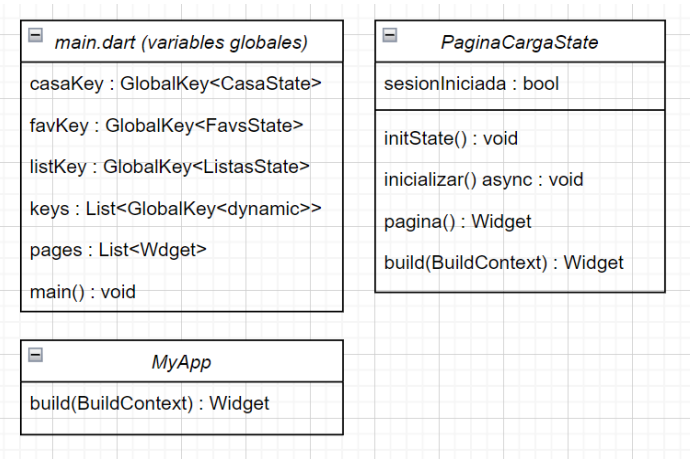
En en archivo formularios.dart tenemos dos clases: FormularioLogIn y FormularioCrearCuenta. Este archivo contiene las dos clases de formularios de la app.



❖ Main

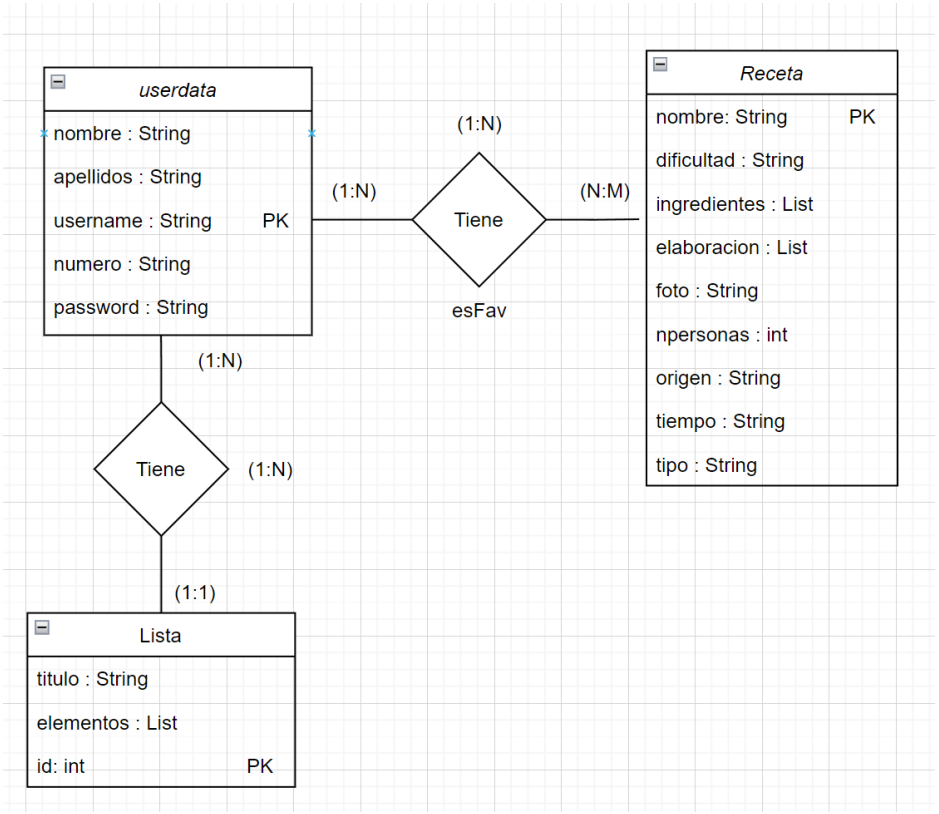
El archivo main.dart incluye varios elementos importantes. Al principio vemos tres declaraciones de GlobalKey que se incluyen en una lista. Estas GlobalKey se utilizan para refrescar las pantallas entre sí cuando hay un cambio.

Si continuamos para abajo nos encontramos con la lista de las pantallas que tendrá la barra de navegación. En la declaración de tres de las clases se incluye la GlobalKey que le corresponde a cada una.



Luego podemos ver el método main que inicia la app y la clase MyApp. Esta clase toma como 'home' a la vista PaginaCarga ya que la aplicación, al iniciarse, comprueba si hay una sesión iniciada o no, por lo que con la pantalla de carga, sobreponemos la pantalla en negro que aparecería en su defecto.

Diagrama E/R



Consideraciones técnicas

Para desplegar la aplicación desde Android Studio hay que hacer lo siguiente:

1. Se necesita tener instalado Android Studio con la última versión.
2. Solo hace falta clonar el repositorio Hambrout del enlace en una carpeta al gusto y abrirla en Android Studio.
3. Para acceder a la base de datos Firebase me tienes que solicitar permiso mandando un correo a irenebarboteo@gmail.com y solo se permite visualizar.
4. Si quieres ejecutar la aplicación necesitas:
    a. Abrir un emulador en Android Studio o conectar un dispositivo físico al ordenador.
    b. Añadir una nueva configuración de ejecución de flutter indicando el archivo main.dart como main y nombrandolo, por ejemplo, hambroutMain.

La clave de la API es únicamente mía. Si cambias la cuenta de Firebase que va conectada a la aplicación necesitas generar una api en la cuenta de Google Cloud de ese gmail.

Documentación de la solución

Todo el código de la aplicación está en GitHub El repositorio es público por lo que podrás consultarlo cuando quieras.

El enlace al repositorio de Hambrout es este:
https://github.com/IBarC?tab=repositories

Enlaces de interés

Mi perfil de GitHub
Mi perfil de LinkedIn

Documentación de Flutter: https://docs.flutter.dev/
Documentación de Firebase: https://firebase.google.com/docs?hl=es-419
Documentación de API de Google Cloud:
https://cloud.google.com/apis/docs/overview?hl=es-419

Dependencias usadas:
https://pub.dev/packages/firebase_core
https://pub.dev/packages/cloud_firestore
https://pub.dev/packages/back_button_interceptor

https://pub.dev/packages/dio
https://pub.dev/packages/google_maps_flutter
https://pub.dev/packages/geolocator
https://pub.dev/packages/lottie (para cargar animaciones)
https://pub.dev/packages/http
https://pub.dev/packages/custom_info_window/versions

Recetas:
https://www.bonviveur.es/recetas/tag/cocina-internacional/

Páginas de las que he obtenido código:
https://www.youtube.com/watch?v=zDXpRTA55gE&list=LL&index=7
https://www.youtube.com/watch?v=MHuQd9MXva0
https://stackoverflow.com/questions/48481590/how-to-set-update-state-of-stateful widget-from-other-statefulwidget-in-flutter
https://stackoverflow.com/questions/16800540/how-should-i-check-if-the-input-is-an-email-address-in-flutter
https://es.stackoverflow.com/questions/244344/regexp-sobre-nombres-compuestos
https://www.youtube.com/watch?v=9v44lAagZCI
https://www.youtube.com/watch?v=x0QNPHYATj4&t=4878s
https://pub.dev/packages/custom_info_window/example

Todas las fotos utilizadas en la app son obtenidas de Freepik y FlatIcon.