



DISTER

LIQIANG YANG, MARC

Índice

GUÍA TÉCNICA.....	3
INTRODUCCIÓN.....	3
REQUISITOS DEL SISTEMA.....	3
INSTALACIÓN:.....	3
Configuración del proyecto de Firebase:.....	4
Configuración del RapidAPI:.....	5
DEPENDENCIAS DEL PROYECTO.....	6
Core y Configuración.....	6
Autenticación y Usuarios.....	6
Servicios en la Nube.....	6
Gestión de Estado.....	6
Internacionalización y Localización.....	6
Interfaz de Usuario (UI).....	6
Iconos y estilo:.....	6
Navegación:.....	7
Indicadores y progreso:.....	7
Texto:.....	7
Imágenes:.....	7
Carruseles y componentes visuales:.....	7
Fecha y hora:.....	7
Splash screen:.....	7
Acciones Externas.....	7
ESTRUCTURA DEL PROYECTO.....	8

GUÍA TÉCNICA

INTRODUCCIÓN

Dister es una aplicación multiplataforma desarrollada con Flutter, diseñada desde 0 ofreciendo una experiencia moderna y minimalista. En este manual te guiará a través de las principales funcionalidades de la aplicación y cómo sacar provecho de ellas.

REQUISITOS DEL SISTEMA

Antes de usar la aplicación:

- Dispositivos compatibles: Android versión: 12^ (se puede utilizar en versiones anteriores pero habrá animaciones o pantallas de carga que no estarán disponibles).
- Conexión a Internet: Requerida para las funcionalidades de autenticación y sincronización de datos.
- Espacio de almacenamiento: Al menos 500MB.
- API Key: [Firebase](#), [RapidApi](#)

INSTALACIÓN:

Para la instalación de esta aplicación, como aún no se ha publicado en las Stores, tienes dos opciones:

- Descargar la aplicación mediante la APK proporcionada por el link de GitHub [\[Link\]](#)
- En caso de querer modificar la aplicación deberás descargar el proyecto mediante el mismo [\[Link\]](#), o creando una pull request en caso de que quieras contribuir.

Esta aplicación se puede desplegar tanto como en web o android, antes de arrancar tendrás que ejecutar el siguiente comando en la carpeta del proyecto:

```
flutter pub get
```

Una vez ejecutado seguramente te salga este error:

```
Error detected in pubspec.yaml:  
No file or variants found for asset: .env.  
  
Failed to compile application.  
  
Exited (1).
```

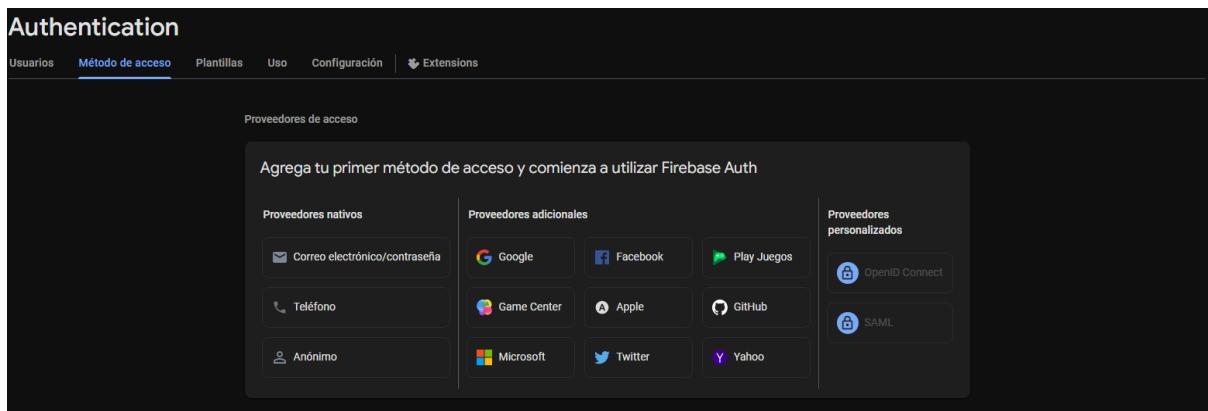
¡NO TE ASUSTES! Es normal, esto se debe a que necesitas crear el archivo .env en la ruta del proyecto y usar tus API KEYS.

Para solucionar este error debes seguir los siguientes pasos:

Configuración del proyecto de Firebase:

Por si no queréis leer las instrucciones, este vídeo puede servir de ayuda [\[Link\]](#), en caso de que prefieras leer tienes la documentación oficial de Firebase [\[Link\]](#) que está bien estructurada.

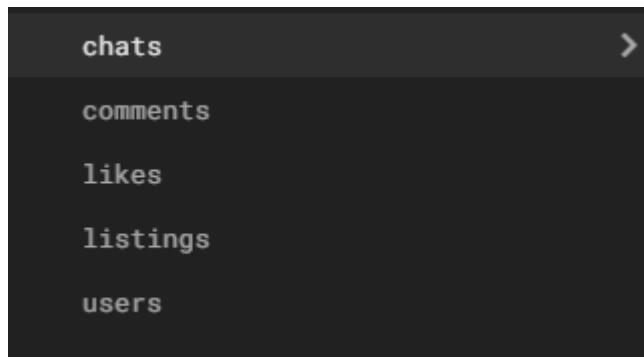
Una vez finalizada la configuración deberás configurar el Firebase Auth, para ello tienes que ir a la consola de Firebase y activar el Firebase Auth y la opción de proveedores nativos, Correo Electrónico/ Contraseña.



Activando el Auth, te permite hacer el registro y login de la aplicación, en este proyecto solo está configurado para el proveedor Correo/Contraseña en caso de que quieras configurar más proveedores se puede ver en la imagen que firebase permite numero de teléfono, Google...

Aparte de esto tendrás que activar el Firebase Cloud, es una base de datos NOSQL, que permite guardar “colecciones”, este lo usaremos para guardar los Posts de los usuarios, los datos de usuarios como Username y descripción u otras cosas...

En nuestro caso tenemos estas colecciones:



¡ATENCIÓN! (Opcional) Realizar la configuración del Firebase Storage, este necesita obligatoriamente la tarjeta de crédito o débito, sirve para poder almacenar las fotos de perfil o del propio post. Cabe destacar que el servicio es gratuito, te ofrecen 300€ gratis y 3 meses de prueba, sin embargo, si llegas a gastar ese dinero

se te cobrará automáticamente. En caso de que no quieras realizar este paso, puedes optar por otras plataformas como subirlas a IMGUR...

Configuración del RapidAPI:

Antes de empezar la configuración de la RapidAPI, vamos a explicar un poco en qué consiste RapidApi: es una plataforma que actúa como un centro de APIs, ofreciendo acceso a una amplia variedad de APIs públicas y privadas. Permite a desarrolladores encontrar, probar y conectarse a diferentes APIs con una única clave y panel de control.

En concreto usaremos la API Real-Time Product Search la versión 7 [[Link](#)], porque nos permite elegir el idioma, el país, que productos quieras buscar... Esta API es gratuita, solo que tiene **Límites**, tienes 1000 llamadas tipo test al mes y 100 llamadas al mes para la producción de la aplicación.

No requiere cuenta bancaria por lo tanto es un extra a tener en cuenta.

Cuando tengas todo tienes que llenar todo esto: es el archivo [.env] que tiene que estar en la ruta absoluta del proyecto.

```
FIREBASE_API_KEY_WEB=
FIREBASE_APP_ID_WEB=
FIREBASE_MESSAGING_SENDER_ID=
FIREBASE_PROJECT_ID=
FIREBASE_AUTH_DOMAIN=
FIREBASE_STORAGE_BUCKET=
FIREBASE_MEASUREMENT_ID_WEB=

FIREBASE_API_KEY_ANDROID=
FIREBASE_APP_ID_ANDROID=

FIREBASE_API_KEY_IOS=
FIREBASE_APP_ID_IOS=
FIREBASE_IOS_BUNDLE_ID=

FIREBASE_API_KEY_MACOS=
FIREBASE_APP_ID_MACOS=
FIREBASE_IOS_BUNDLE_ID_MACOS=

FIREBASE_API_KEY_WINDOWS=
FIREBASE_APP_ID_WINDOWS=
FIREBASE_MEASUREMENT_ID_WINDOWS=
```

RAPIDAPI_KEY=

DEPENDENCIAS DEL PROYECTO

La aplicación Dister utiliza una serie de plugins para extender sus funcionalidades y mejorar la experiencia del usuario. A continuación, se detalla cada uno de ellos y su propósito:

Core y Configuración

- [firebase_core](#): Inicializa y configura Firebase.
- [flutter_dotenv](#): Carga variables de entorno desde archivos `.env`.

Autenticación y Usuarios

- [firebase_auth](#): Autenticación de usuarios (email, Google, etc.).
- [shared_preferences](#): Almacenamiento local de configuraciones o datos simples.

Servicios en la Nube

- [cloud_firestore](#): Base de datos en tiempo real de Firebase.
- [firebase_storage](#): Almacenamiento de archivos en la nube con Firebase.
- [http](#): Solicitudes HTTP para consumir APIs REST.

Gestión de Estado

- [provider](#): Implementación del patrón Provider para manejar el estado de la app.

Internacionalización y Localización

- [flutter_localizations](#): Soporte para localización de la app.
- [intl](#): Herramientas de internacionalización (fechas, números, etc.).
- [translator](#): Traducción automática de texto.

Interfaz de Usuario (UI)

Iconos y estilo:

- [cupertino_icons](#): Íconos al estilo iOS.

Navegación:

- [sidebarx](#): Barra lateral personalizable.
- [crystal_navigation_bar](#): Barra inferior moderna y atractiva.

Indicadores y progreso:

- [smooth_page_indicator](#): Indicadores animados para carruseles/paginación.
- [step_progress_indicator](#): Indicador de progreso por pasos.

Texto:

- [animated_hint_textfield](#): Campos de texto con animaciones.
- [auto_size_text](#): Ajuste automático del tamaño del texto.

Imágenes:

- [image_picker](#): Selección o captura de imágenes.
- [cached_network_image](#): Descarga y almacenamiento en caché de imágenes.

Carruseles y componentes visuales:

- [carousel_slider](#): Carruseles interactivos de imágenes o contenido.
- [confirmation_success](#): Animaciones visuales de confirmación/exito.

Fecha y hora:

- [flutter_datetime_picker_plus](#): Selector de fecha y hora personalizable.

Splash screen:

- [flutter_native_splash](#): Generación de pantallas de carga personalizadas.

Acciones Externas

- [url_launcher](#): Apertura de URLs en el navegador o apps externas.

ESTRUCTURA DEL PROYECTO

```
Dister/
└── lib/          # Código principal de la aplicación
    ├── controllers/   # Controladores para lógica de negocio y servicios
    │   ├── comment_service/
    │   ├── favorite_service/
    │   ├── firebase/    # Servicios relacionados con Firebase
    │   ├── media/
    │   ├── provider/    # Proveedores para manejo de estado
    │   └── shared_prefs/
    ├── generated/     # Archivos generados automáticamente
    │   ├── l10n.dart    # Localización principal
    │   └── intl/        # Archivos de localización por idioma
    ├── l10n/          # Archivos de localización (ARB)
    │   ├── intl_en.arb
    │   └── intl_es.arb
    ├── models/         # Modelos de datos
    │   ├── api_post_model.dart
    │   ├── categorie_model.dart
    │   ├── category_icons_model.dart
    │   ├── chat_model.dart
    │   ├── comment_model.dart
    │   ├── highlight_model.dart
    │   ├── post_model.dart
    │   └── user_model.dart
    ├── screens/        # Pantallas de la aplicación
    │   ├── mobile/      # Pantallas para dispositivos móviles
    │   └── tablet/      # Pantallas para tablets
    ├── theme/          # Temas de la aplicación
    │   ├── dark_mode.dart
    │   └── light_mode.dart
    ├── widgets/         # Widgets reutilizables
    │   ├── api_post_container.dart
    │   ├── comment_card.dart
    │   ├── custom_dropdown.dart
    │   └── ...
    └── main.dart       # Punto de entrada de la aplicación
└── assets/          # Recursos estáticos (imágenes, fuentes, etc.)
└── test/            # Pruebas unitarias y de widgets
└── pubspec.yaml     # Configuración del proyecto
```