

# CONFIGURACIÓN FIREBASE EN FLUTTER

## Configura o actualiza Firebase CLI

- <https://firebase.google.com/docs/cli?hl=es-419#windows-npm>

## Instalar FlutterFire CLI

El FlutterFire CLI es una herramienta que facilita la configuración de Firebase en proyectos Flutter. Se debe instalar globalmente usando el siguiente comando:

- `flutter pub global activate flutterfire_cli`

## Agregar el directorio de paquetes globales de Dart al PATH

Para que el comando flutterfire sea reconocido en cualquier lugar, se debe agregar el directorio de paquetes globales de Dart a la variable de entorno PATH.

### En Windows:

1. **Encontrar el directorio de paquetes globales de Dart**

Por defecto, el directorio es:

`C:\Users\<TuUsuario>\AppData\Local\Pub\Cache\bin`

### Agregar la ruta al PATH

- Abre el menú **Inicio** y busca "Editar las variables de entorno del sistema".
- Haz clic en "**Variables de entorno...**".
- En la sección "**Variables de usuario**", selecciona la variable "**Path**" y haz clic en "**Editar**".
- Haz clic en "**Nuevo**" y pega la ruta al directorio de paquetes globales de Dart

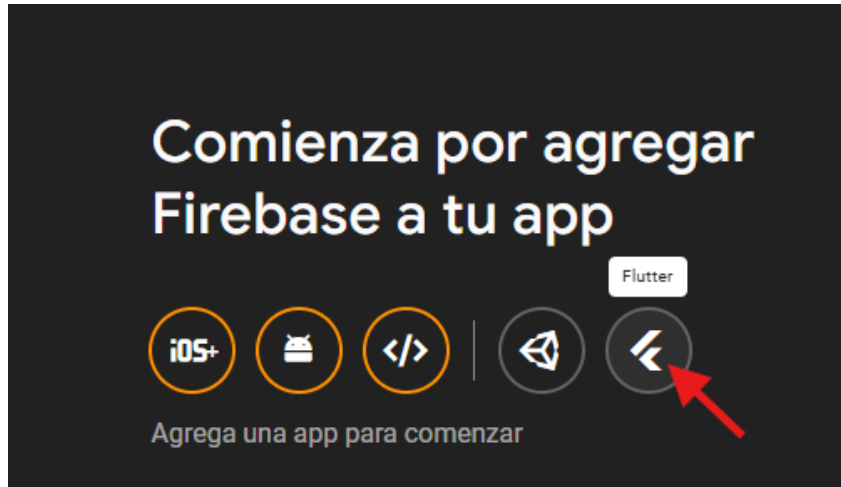
## Verificar la instalación de FlutterFire CLI

Para asegurar que flutterfire está correctamente instalado y reconocido, ejecuta:

- `flutterfire --version`

## Crear Proyecto en Firebase

- Agregar app



- Damos clic en Siguiente



- SEGUIR LOS PASOS

×

Agrega Firebase a tu app de Flutter

1

Prepara tu espacio de trabajo

2

Instala y ejecuta la CLI de FlutterFire

Desde cualquier directorio, ejecuta este comando:

\$ dart pub global activate flutterfire\_cli

Luego, en la raíz del directorio de tu proyecto de Flutter, ejecuta este comando:

\$ flutterfire configure --project=flutter-a575d

Esta acción registra automáticamente tus apps por plataforma en Firebase y agrega un archivo de configuración `lib/firebase_options.dart` al proyecto de Flutter.

Anterior

Siguiente

3

Inicializa Firebase y agrega complementos

- **Crear BD/tablas**
- Ir a la ruta "Compilacion / Firestore Database".
- Crear Base de Datos / modo prueba
- Iniciar una colección "categorías"
- Agrega campos
- \*nombre
- \*descripción

## Reemplazar archivos

- Agrega los archivos en la carpeta lib

Agrega las rutas

//! Rutas para el manejo de Categorías (CRUD)

```
GoRoute(  
  path: '/categoriasFirebase',  
  name: 'categoriasFirebase',  
  builder: (_, __) => const CategoriaFbListView(),  
),  
GoRoute(  
  path: '/categoriasfb/create',  
  name: 'categoriasfb.create',  
  builder: (context, state) => const CategoriaFbFormView(),  
),  
GoRoute(  
  path: '/categoriasfb/edit/:id',  
  name: 'categorias.edit',  
  builder: (context, state) {  
    final id = state.pathParameters['id']!;  
    return CategoriaFbFormView(id: id);  
  },  
),
```

- Agrega en custom\_drawer.dart

```
ListTile(  
  leading: const Icon(Icons.cloud),  
  title: const Text('Categorías Firebase'),  
  onTap: () => context.pushNamed('categoriasFirebase'),  
),
```

- Instala las dependencias.  
**flutter pub add firebase\_core**  
**flutter pub add cloud\_firestore**
- edita el archivo de configuración “Build.gradle”

**Agregar:**

```
Plugins{  
  id "com.google.gms.google-services" // debe tener esta línea  
}
```