



Jheferson condori

MANUEL INSTALACION FLUTTER





Wardiere Inc.

Table of Contents

01 Instalaciones en el equipo

02 Extensiones de VSCode

03 Instalación de Flutter

04 Instalar el Emuladores

INSTALACIONES EN EL EQUIPO



git



POSTMAN



Instalaciones en el equipo

1. [Git](#)

```
git config --global user.name "Tu nombre"
git config --global user.email "Tu correo"
```

2. Crear [cuenta en GitHub](#)

3. [VSCode - Visual Studio Code](#)

4. [Postman](#)

5. [Android Studio](#)

6. Sólo Mac [xCode](#)

7. [Flutter SDK](#)

8. [NodeJS](#)

9. [Docker Desktop](#)

10. [Table Plus](#) <-- Visor de base de datos

**Crear su cuenta
en Github**

Download then install Flutter

To install Flutter, download the Flutter SDK bundle from its archive, move the bundle to where you want it stored, then extract the SDK.

1. Download the following installation bundle to get the latest stable release of the Flutter SDK.

`flutter_windows_3.24.3-stable.zip`

INSTALAR EL EMULADORES

Activitus Bar v0.0.47

Gruntfuggly | 383,129 | ★★★★★ (36)

Save some real estate by recreating the

[Disable](#) [Uninstall](#) ☒ Auto Update

[DETAILS](#) [FEATURES](#) [CHANGELOG](#)

Error Lens v3.20.0

Alexander | 4,848,534 | ★★★★★ (163)

Improve highlighting of errors, warnings and other language diagnostics.

[Disable](#) [Uninstall](#) ☒ Auto Update 



Paste JSON as Code v23.0.170

quicktype | 2,365,828 | ★★★★★ (46)

Copy JSON, paste as Go, T

[Disable](#) [Uninstall](#)

[DETAILS](#) [FEATURES](#)

Terminal v0.0.10

Jun Han | 1,633,297 | ★★★★★ (15)

Terminal for Visual Studio Code

[Disable](#) [Uninstall](#) ☒ Auto Update 



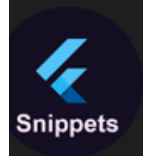
Awesome Flutter Snippets v4.0.1

Neevash Ramdial (Nash) | 673,181 | ★★★★★ (4)

Awesome Flutter Snippets is a collection snippets and shortcuts for commonly used Flutter functions and classes

[Disable](#) [Uninstall](#) ☒ Auto Update

[DETAILS](#) [FEATURES](#) [CHANGELOG](#)




bloc v6.6.6

Felix Angelov | 570,672 | ★★★★★ (17)


Support for the bloc library and provides tools for effectively creating blocs for both Flutter and AngularDart apps.

[Disable](#) [Uninstall](#) ☒ Auto Update 

Dart v3.96.0

Dart Code  [dartcode.org](#) | 10,084,142 | ★★★★★ (81)

Dart language support and debugger for Visual Studio Code.

[Disable](#) [Uninstall](#) [Switch to Pre-Release Version](#) ☒ Auto Update 

[DETAILS](#) [FEATURES](#) [CHANGELOG](#)

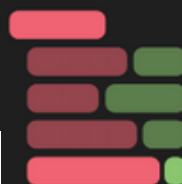
[chat](#) [discord](#) [twitter](#) [dartcode](#) [help](#) [contribute](#)

Pubspec Assist v2.3.2


Jeroen Meijer | 648,665 | ★★★★★ (15)

Easily add and update dependencies to your Dart and Flutter project.

[Disable](#) [Uninstall](#) ☒ Auto Update 



Lorem ipsum v1.3.1

Daniel Imms  [growingwiththeweb.com](#) | 880,130 | ★★★★★ (14)

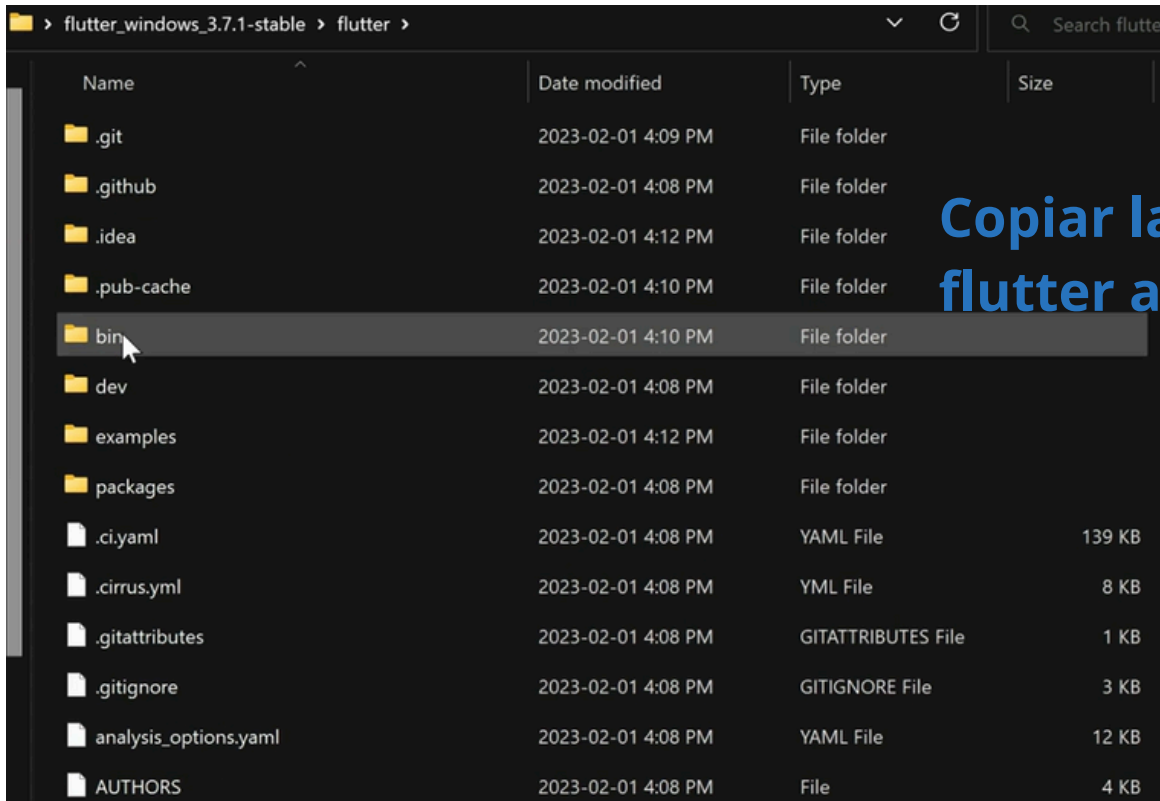
Generates and inserts lorem ipsum text

[Disable](#) [Uninstall](#) ☒ Auto Update 

[DETAILS](#) [FEATURES](#) [CHANGELOG](#)

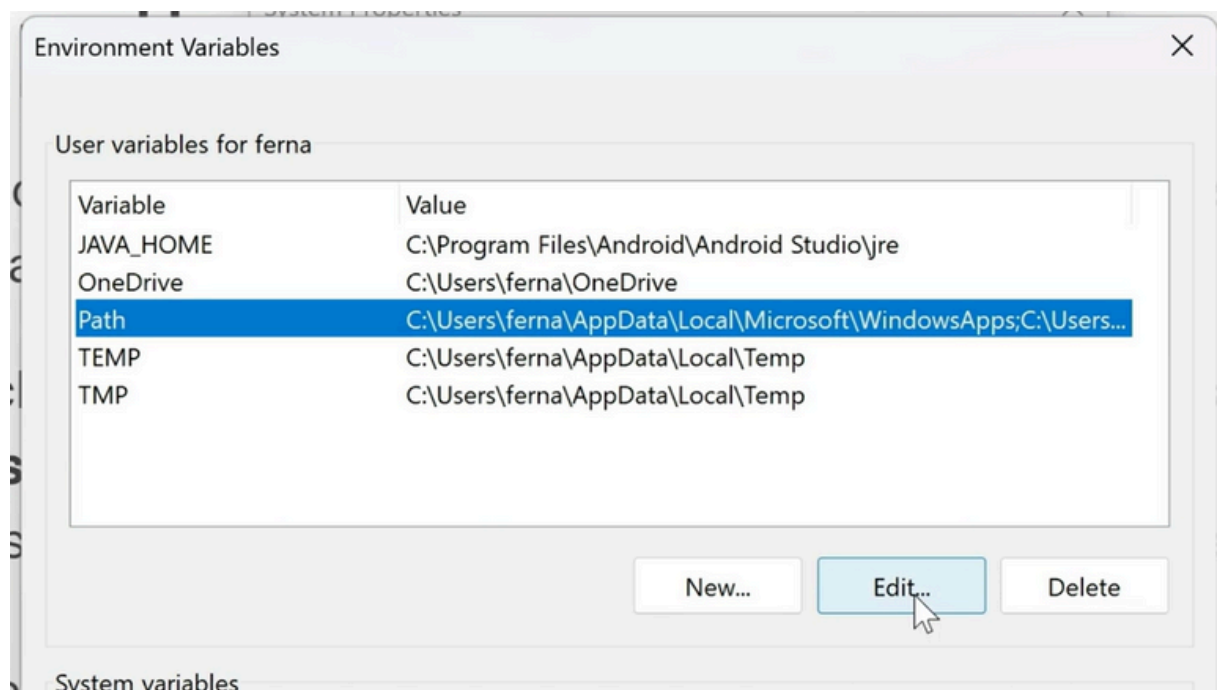


INSTALACIÓN DE FLUTTER



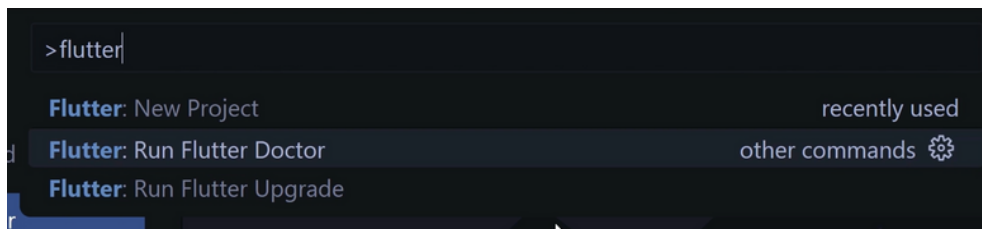
Copiar la carpeta flutter al disco C.

Configuramos el Path en variables de entorno de sistema

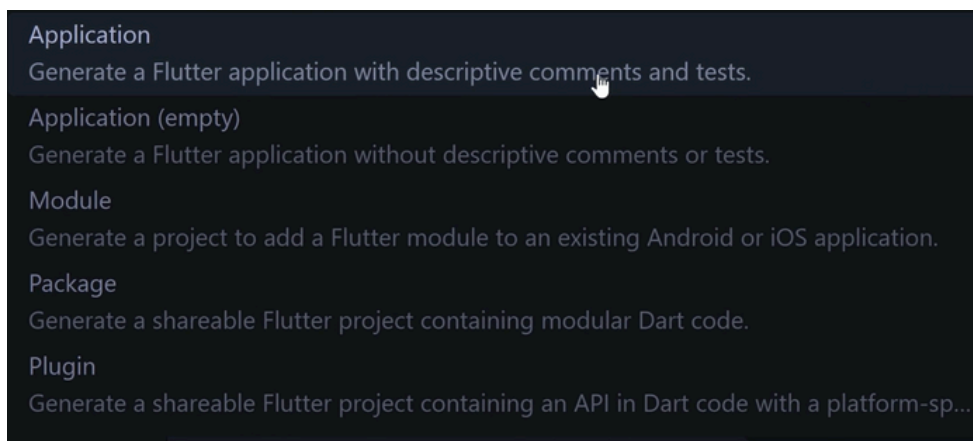


INSTALAR EL EMULADORES

Configurar Flutter doctor

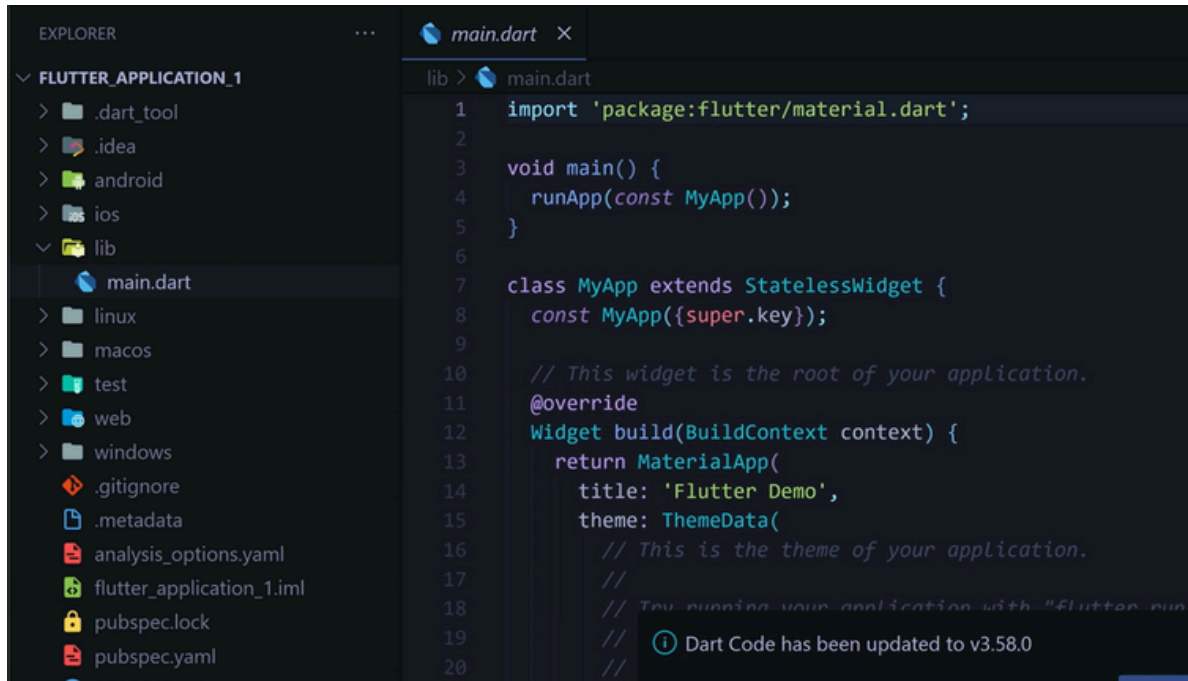


Seleccionamos los comentarios descriptivos



INSTALAR EL EMULADORES

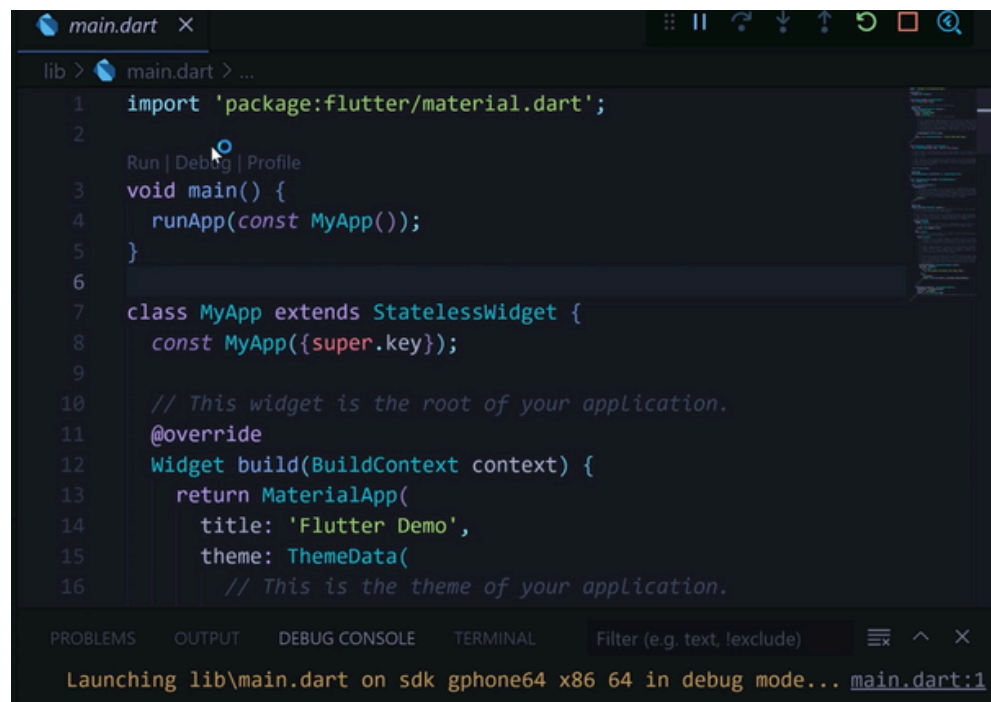
Eso hara que se descarguen los
templeates para usar flutter



The screenshot shows an IDE interface. On the left, the 'EXPLORER' panel displays a project structure for 'FLUTTER_APPLICATION_1'. The 'lib' directory is expanded, showing 'main.dart' selected. The main editor area displays the content of 'main.dart', which includes imports, a main function, and a StatelessWidget class named MyApp. A notification at the bottom right states 'Dart Code has been updated to v3.58.0'.

```
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(const MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   const MyApp({super.key});
9
10  // This widget is the root of your application.
11  @override
12  Widget build(BuildContext context) {
13    return MaterialApp(
14      title: 'Flutter Demo',
15      theme: ThemeData(
16        // This is the theme of your application.
17        //
18        // You can run your application with "flutter run"
19        // on a mobile device, and you'll see the Flutter
20        // logo, and a UI with the theme of your application.
```

Precionamos F5 para comenzar el
desarrollo

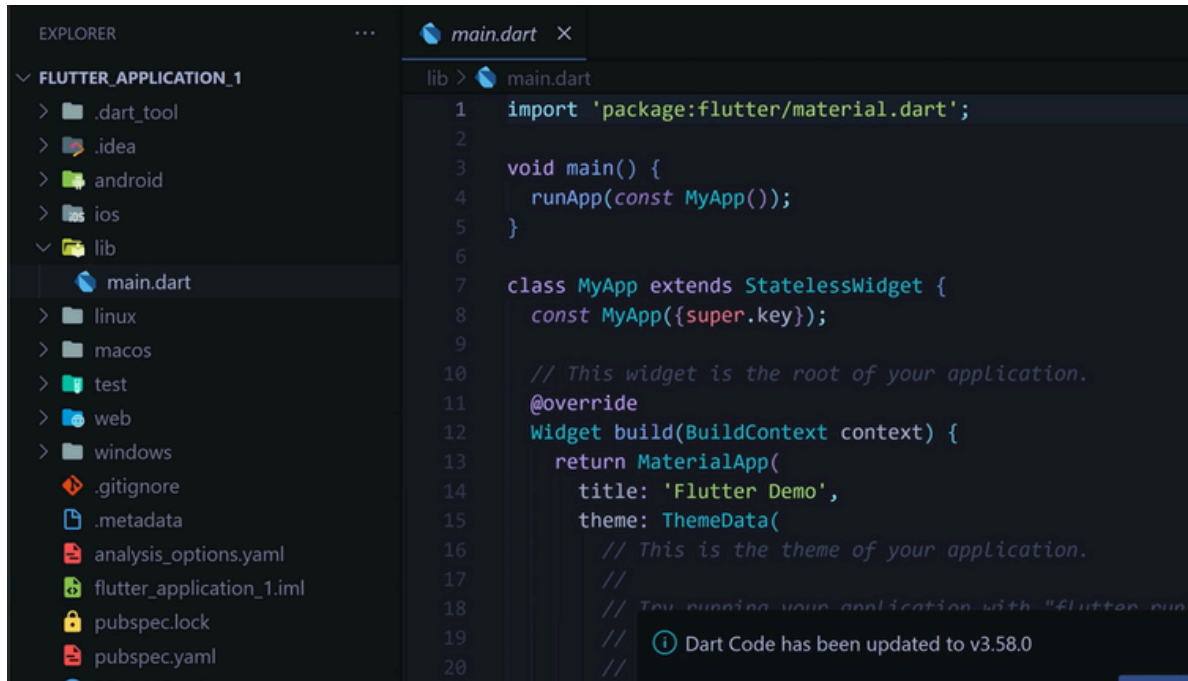


The screenshot shows the same IDE interface as before, but with the 'Run' button (a green play icon) highlighted in the top toolbar. The main editor area displays the content of 'main.dart'. The bottom status bar shows the message 'Launching lib\main.dart on sdk gphone64 x86 64 in debug mode... main.dart:1'.

```
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(const MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   const MyApp({super.key});
9
10  // This widget is the root of your application.
11  @override
12  Widget build(BuildContext context) {
13    return MaterialApp(
14      title: 'Flutter Demo',
15      theme: ThemeData(
16        // This is the theme of your application.
```


INSTALAR EL EMULADORES

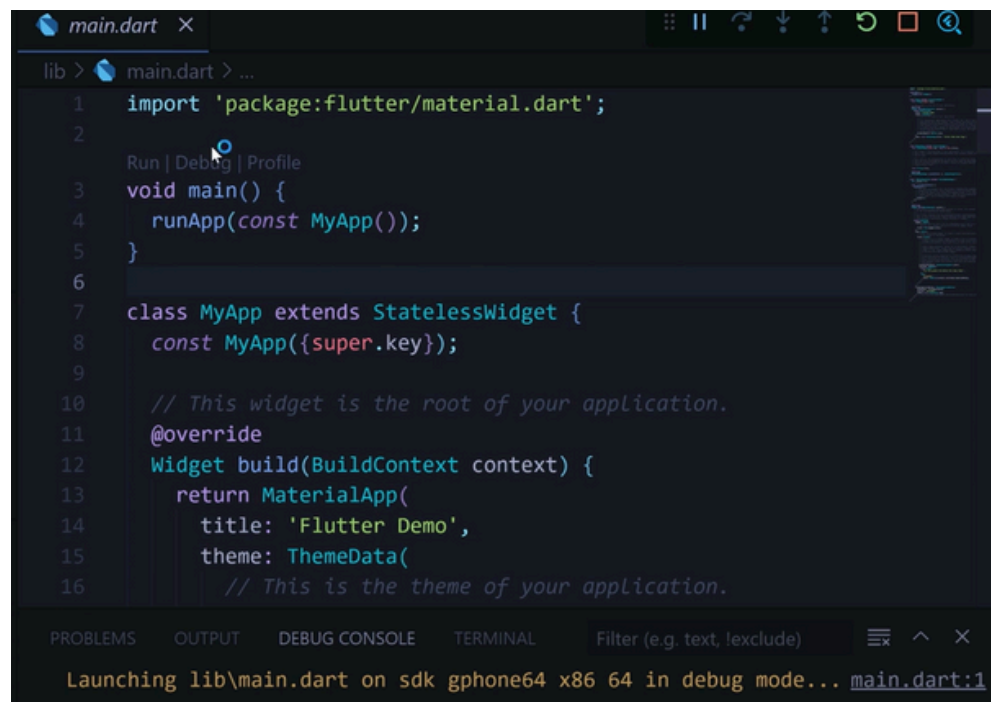
Eso hara que se descarguen los
templeates para usar flutter



The screenshot shows an IDE interface. On the left, the 'EXPLORER' panel displays a project structure for 'FLUTTER_APPLICATION_1'. The 'lib' directory is expanded, showing 'main.dart' selected. The main editor area displays the content of 'main.dart', which includes imports, a main function, and a StatelessWidget class named 'MyApp'. A notification at the bottom right states 'Dart Code has been updated to v3.58.0'.

```
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(const MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   const MyApp({super.key});
9
10  // This widget is the root of your application.
11  @override
12  Widget build(BuildContext context) {
13    return MaterialApp(
14      title: 'Flutter Demo',
15      theme: ThemeData(
16        // This is the theme of your application.
17        //
18        // You can run your application with "flutter run"
19        // from your terminal and open the application in the
20        // Dart Code IDE to see these widgets in action.
```

Precionamos F5 para comenzar el
desarrollo

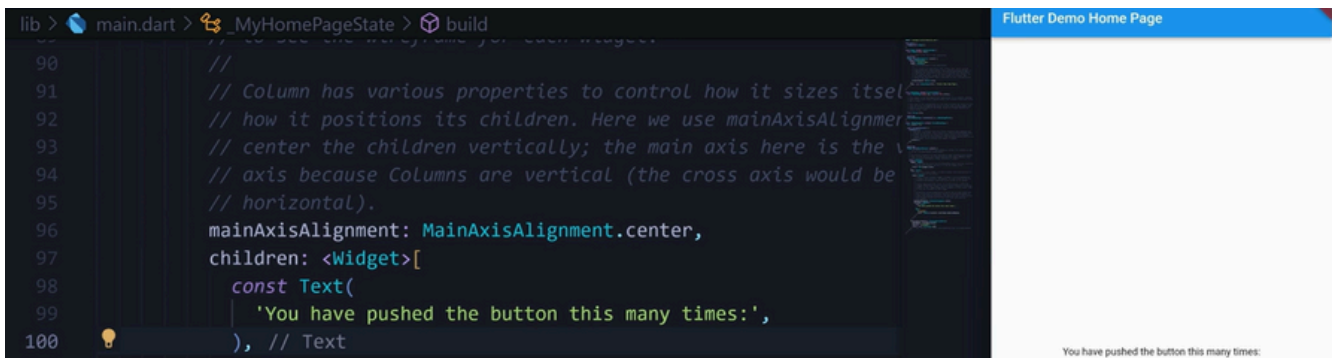


The screenshot shows the same IDE interface as before, but with the 'Run' button (a green play icon) highlighted in the top toolbar. The main editor area displays the content of 'main.dart'. At the bottom, the status bar shows 'Launching lib\main.dart on sdk gphone64 x86 64 in debug mode... main.dart:1'.

```
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(const MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   const MyApp({super.key});
9
10  // This widget is the root of your application.
11  @override
12  Widget build(BuildContext context) {
13    return MaterialApp(
14      title: 'Flutter Demo',
15      theme: ThemeData(
16        // This is the theme of your application.
```

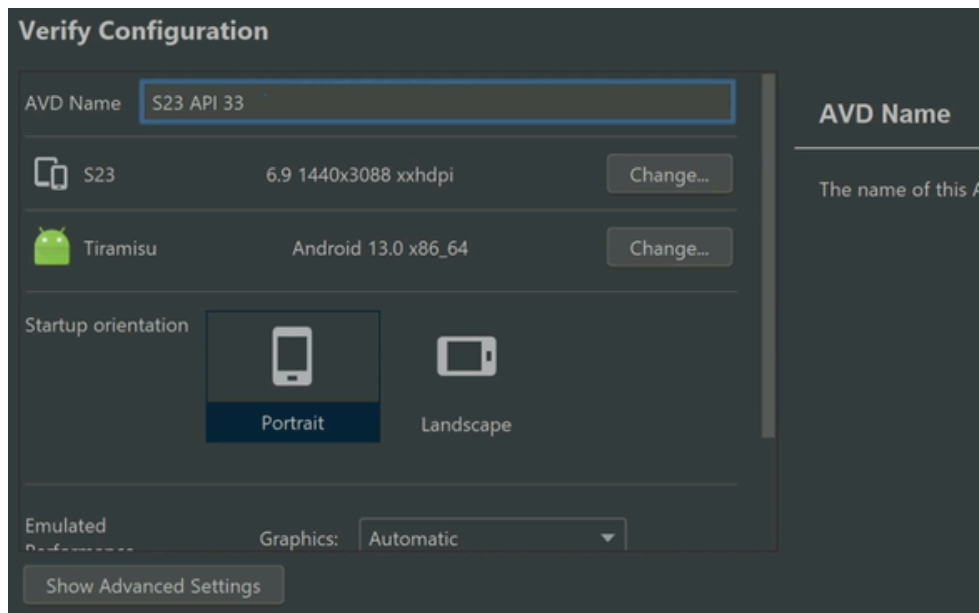

INSTALAR EL EMULADORES

Al ejecutar se abra el emulador

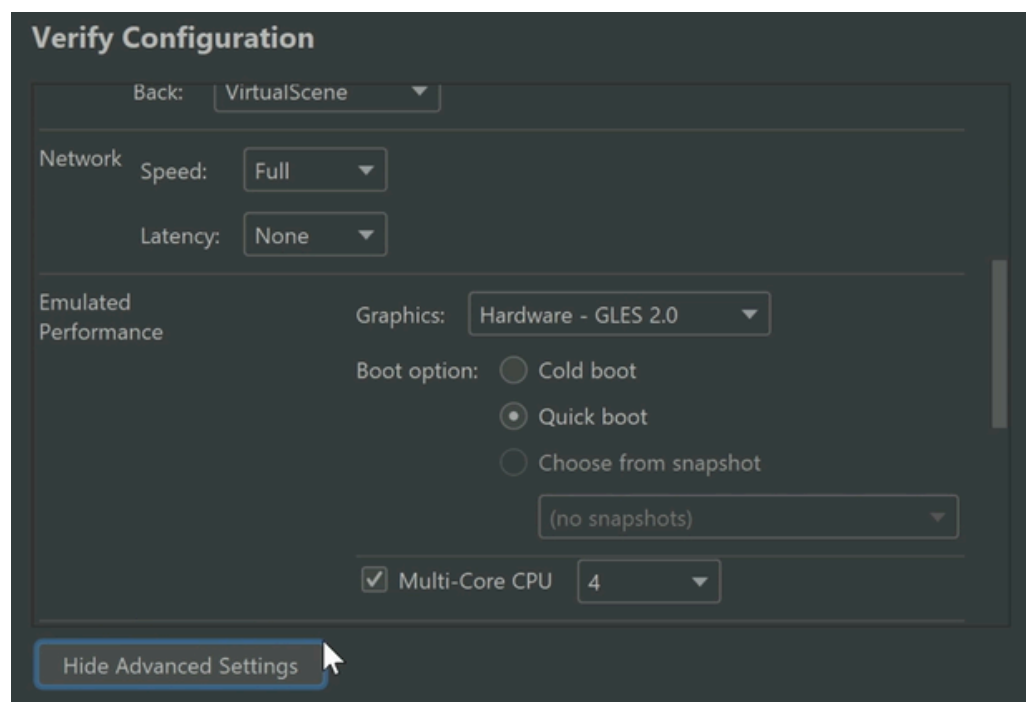


INSTALAR EL EMULADORES

Configuramos el api para ejecutar los paquetes



Configuramos las graficas



INSTALAR EL EMULADORES

Al ejecutar se ejecuta el emulador

