

AULA

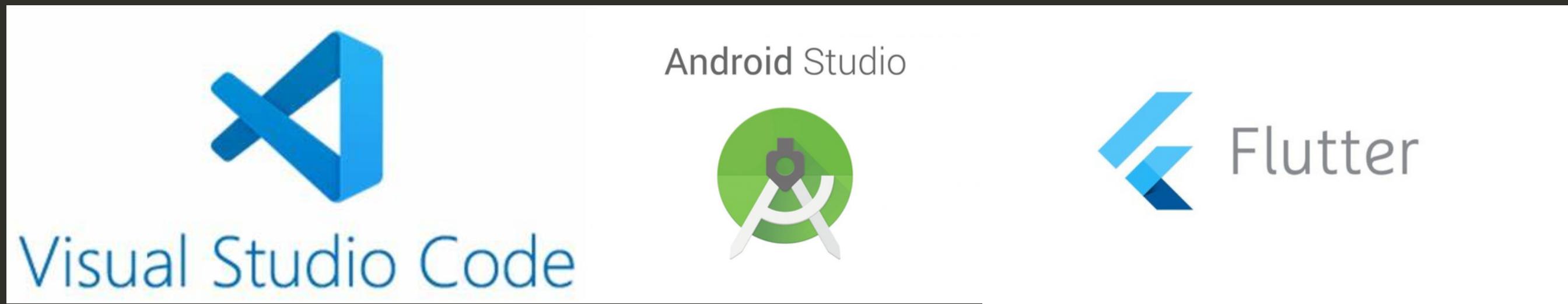
# PRIMEIRO APP



**Nessa apresentação iremos ver um passo a passo de  
uma aplicação flutter.**

# Para configurar tudo precisaremos de :

- Criar uma pasta específica para os projetos em flutter
- Atualizar Flutter



**Agora iremos fazer passo a passo,  
siga atentamente as instruções**

# Abra o prompt de comando, como Administrador e digite o comando ‘flutter upgrade’, isso deve atualizar o framework todo

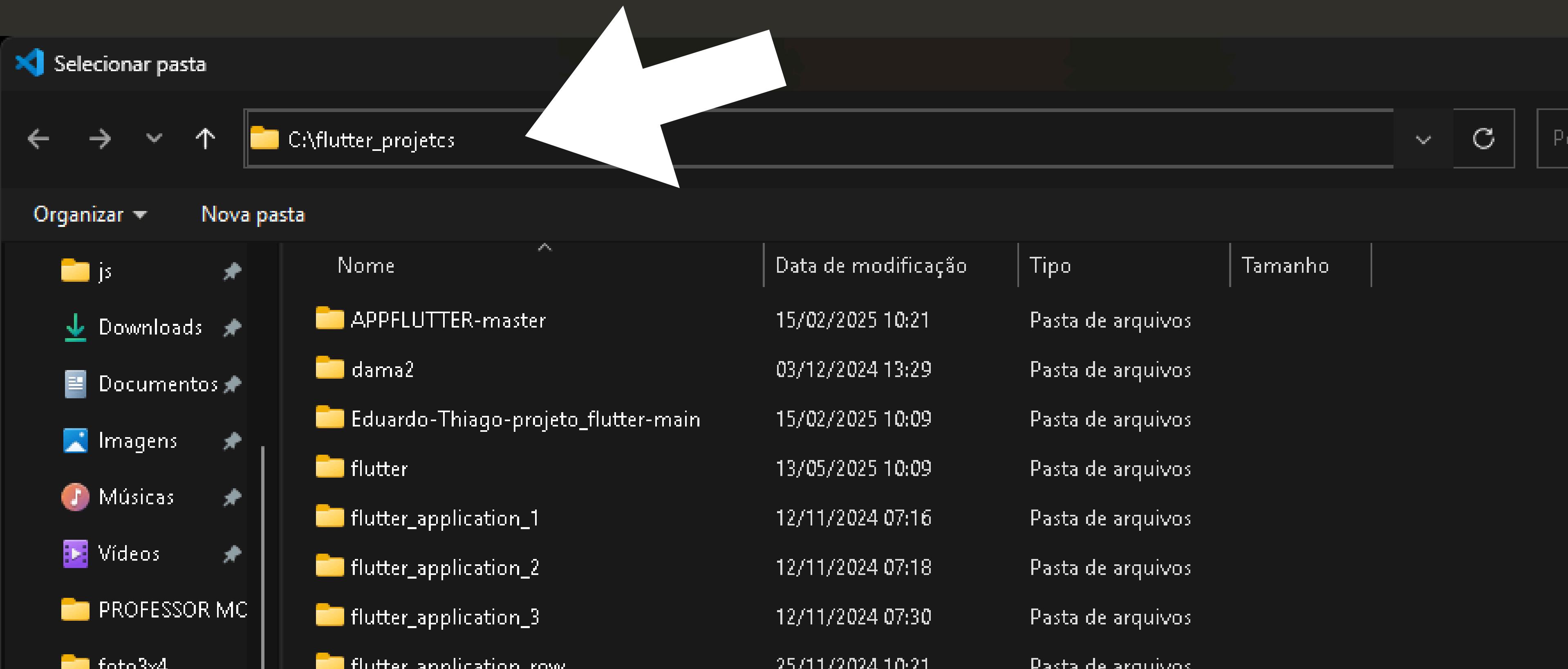
```
Administrator: Prompt de Comando - flutter upgrade
Microsoft Windows [versão 10.0.26100.4351]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Windows\System32> flutter upgrade
Flutter is already up to date on channel stable
Flutter 3.32.4 • channel stable • https://github.com/flutter/flutter.git
Framework • revision 6fba2447e9 (6 days ago) • 2025-06-12 19:03:56 -0700
Engine • revision 8cd19e509d (6 days ago) • 2025-06-12 16:30:12 -0700
Tools • Dart 3.8.1 • DevTools 2.45.1

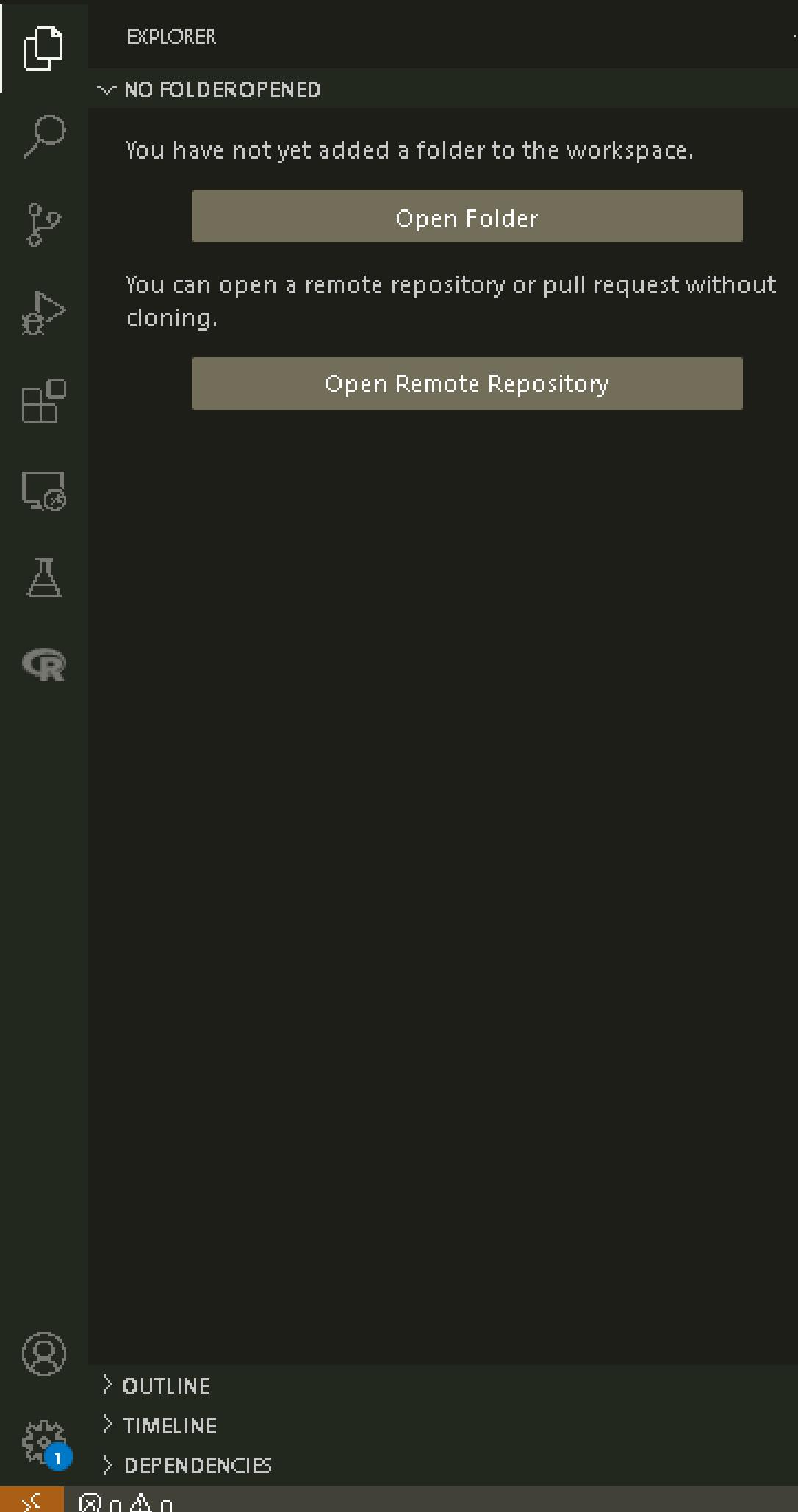
C:\Windows\System32>
```

Nesse exemplo já estava atualizado

# Agora abra o File Explorer e no disco C: crie uma pasta flutter\_projects



**Agora vamos criar o primeiro projeto**



# Abram o Vs Code

Show All Commands `Ctrl + Shift + P`

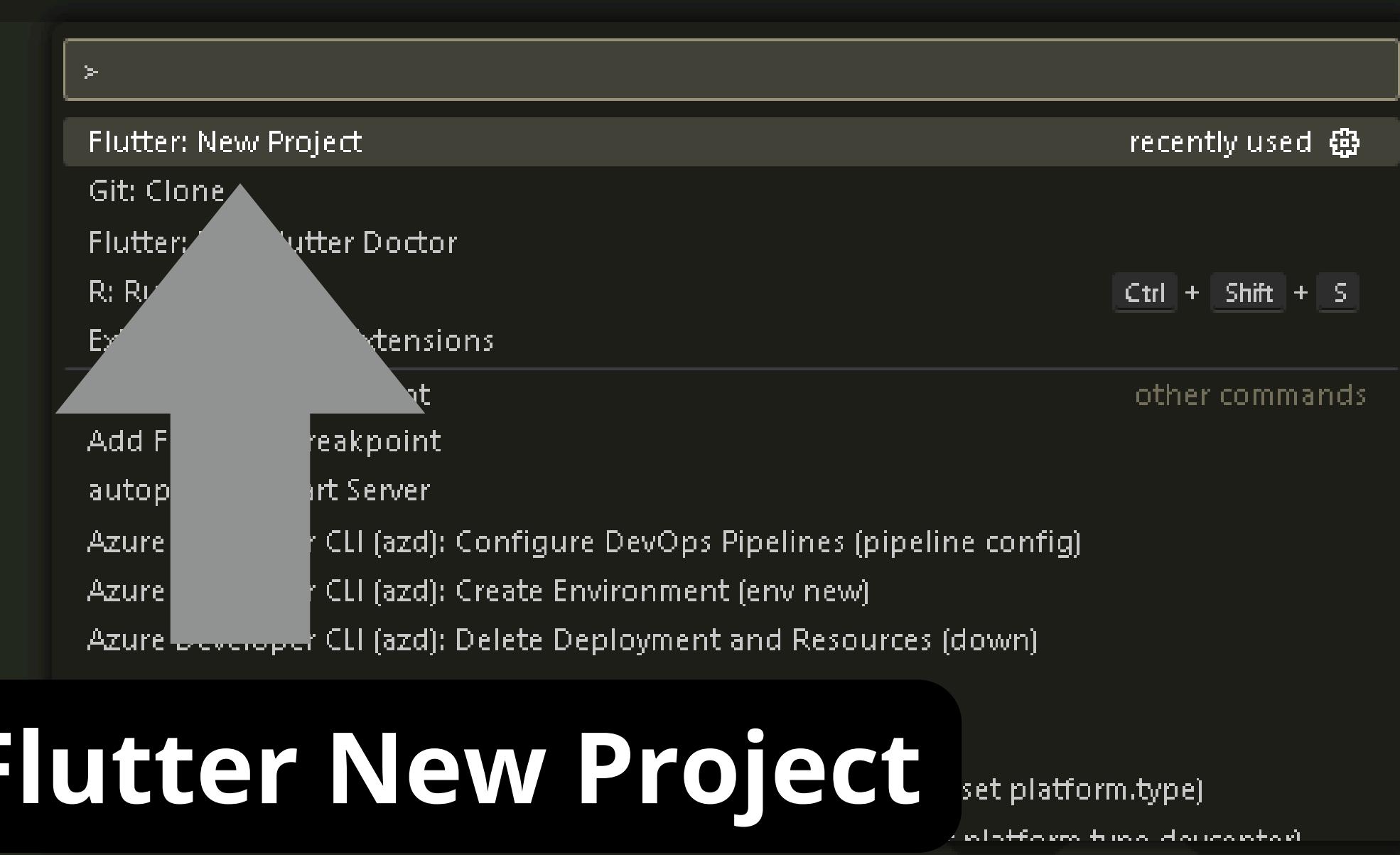
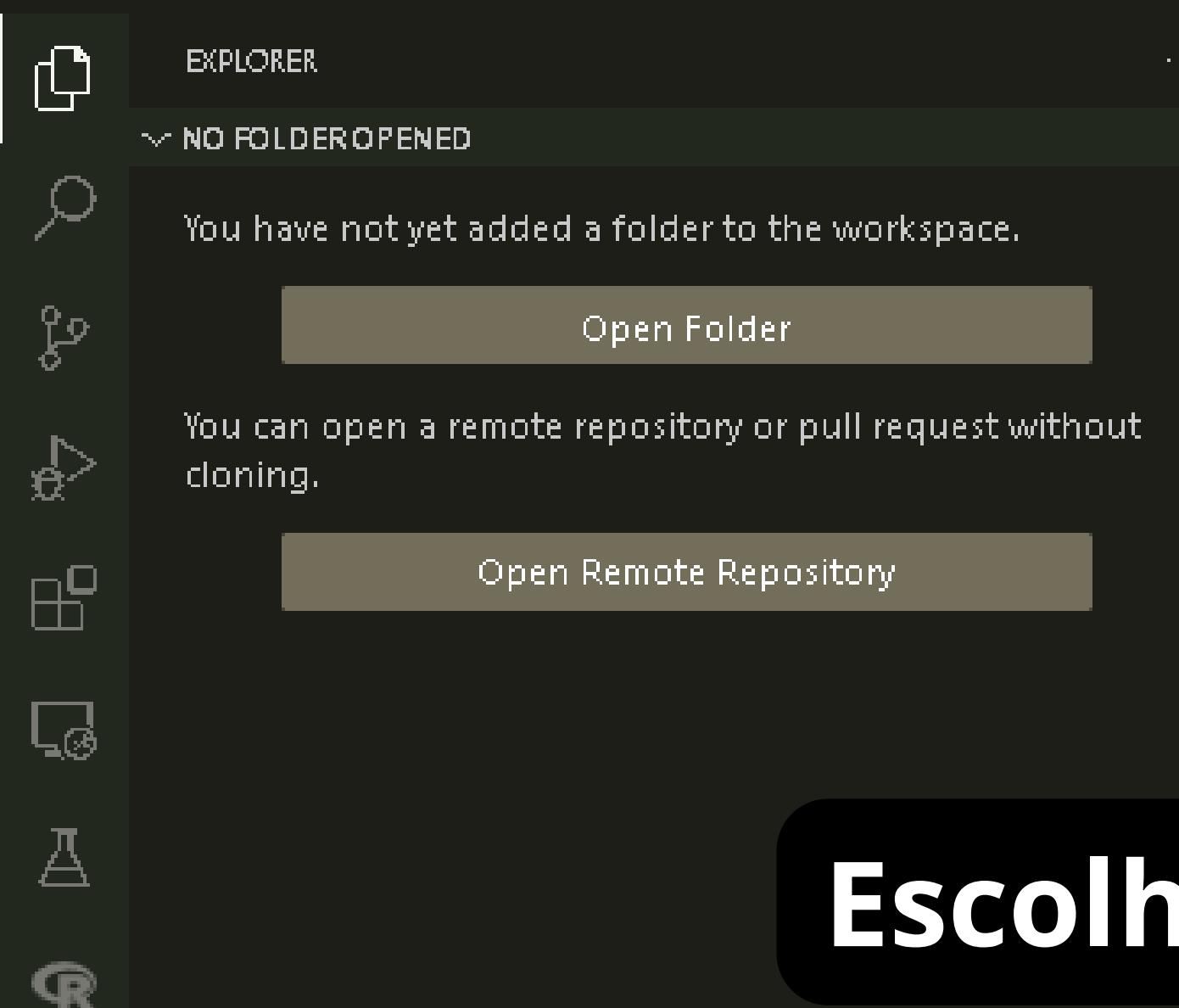
Go to File `Ctrl + P`

Find in Files `Ctrl + Shift + F`

Toggle Full Screen `F11`

Show Settings `Ctrl + ,`

**Agora pressione as teclas  
Ctrl+Shift+p ao mesmo tempo**



# Escolha Flutter New Project

set platform.type)

platform type documentation

Show All Commands **Ctrl + Shift + P**

Go to File **Ctrl + P**

Find in Files **Ctrl + Shift + F**

Toggle Full Screen **F11**

Show Settings **Ctrl + ,**



EXPLORER

NO FOLDER OPENED

You have not yet added a folder to the workspace.

Open Folder

You can open a remote repository or pull request without cloning.

Which Flutter template?

Application

A Flutter application with descriptive comments and tests.

Applications

Empty Application

A Flutter application without descriptive comments or tests.

Skeleton Application

A List View / Detail View Flutter application that follows community best practices.

Plugin

A shareable Flutter project containing an API in Dart code with a platform-specific imple...

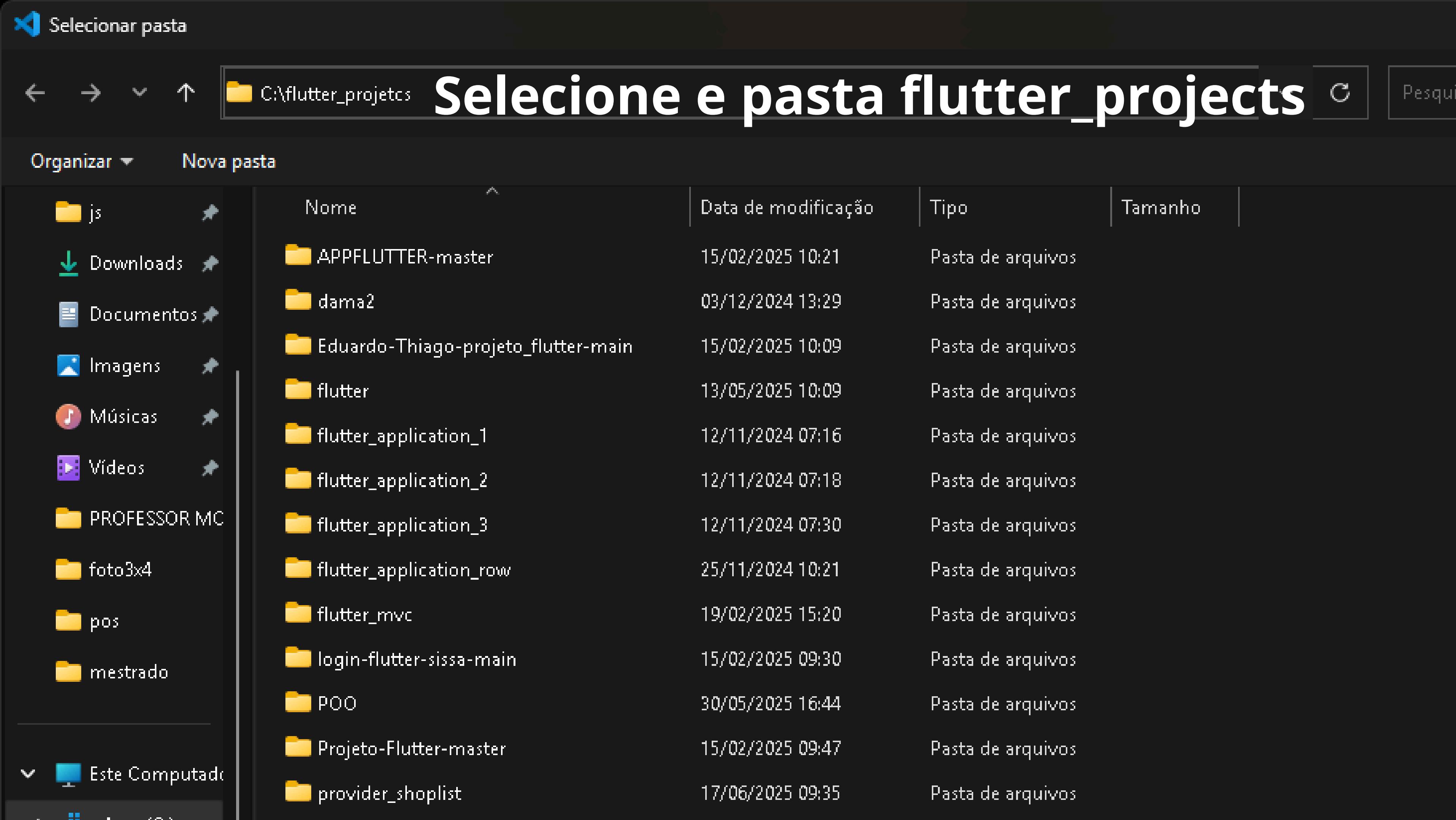
# Escolha a segunda opção: Empty Application

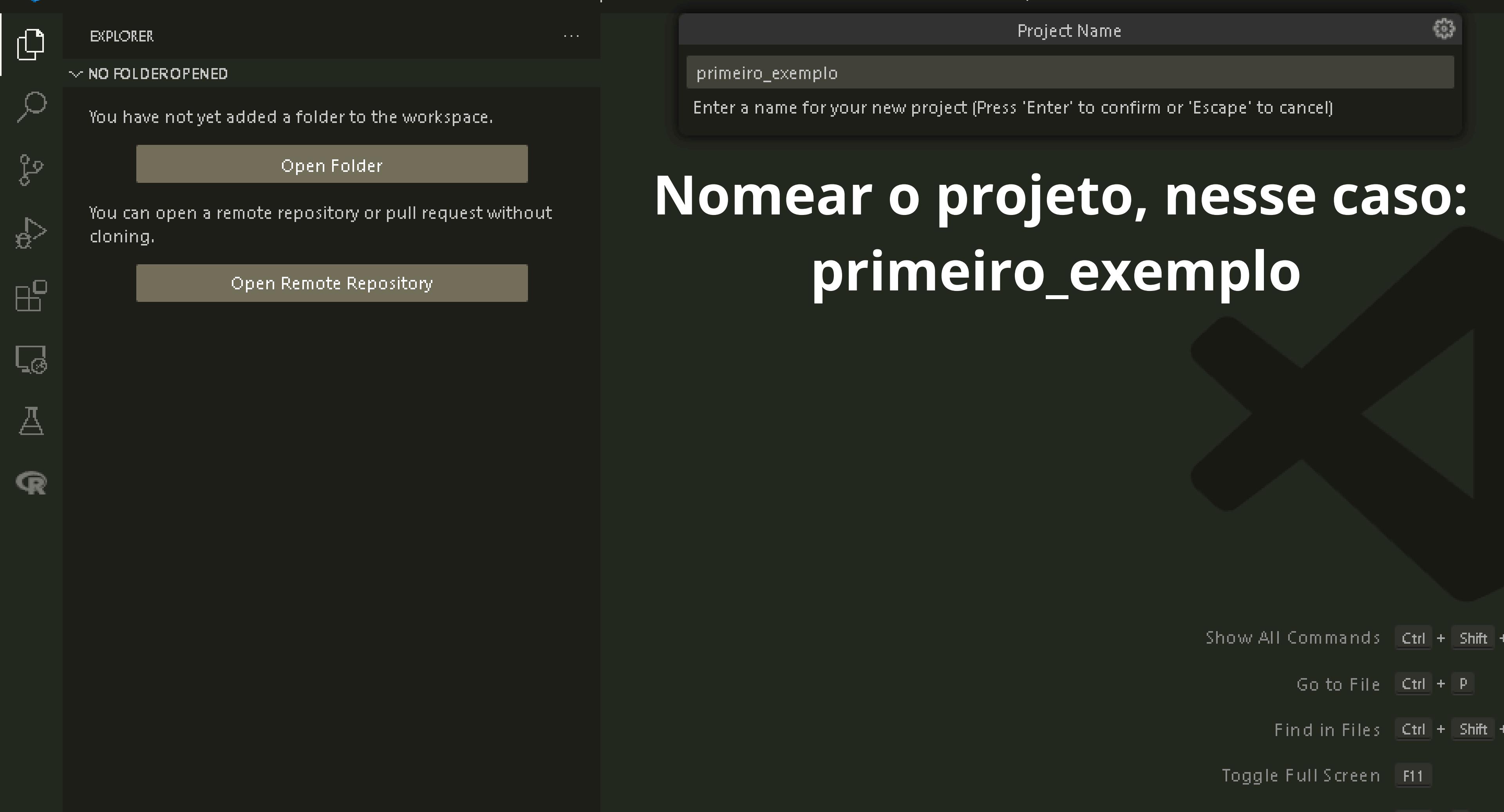
Show All Commands **Ctrl + Shift + P**

Go to File **Ctrl + P**

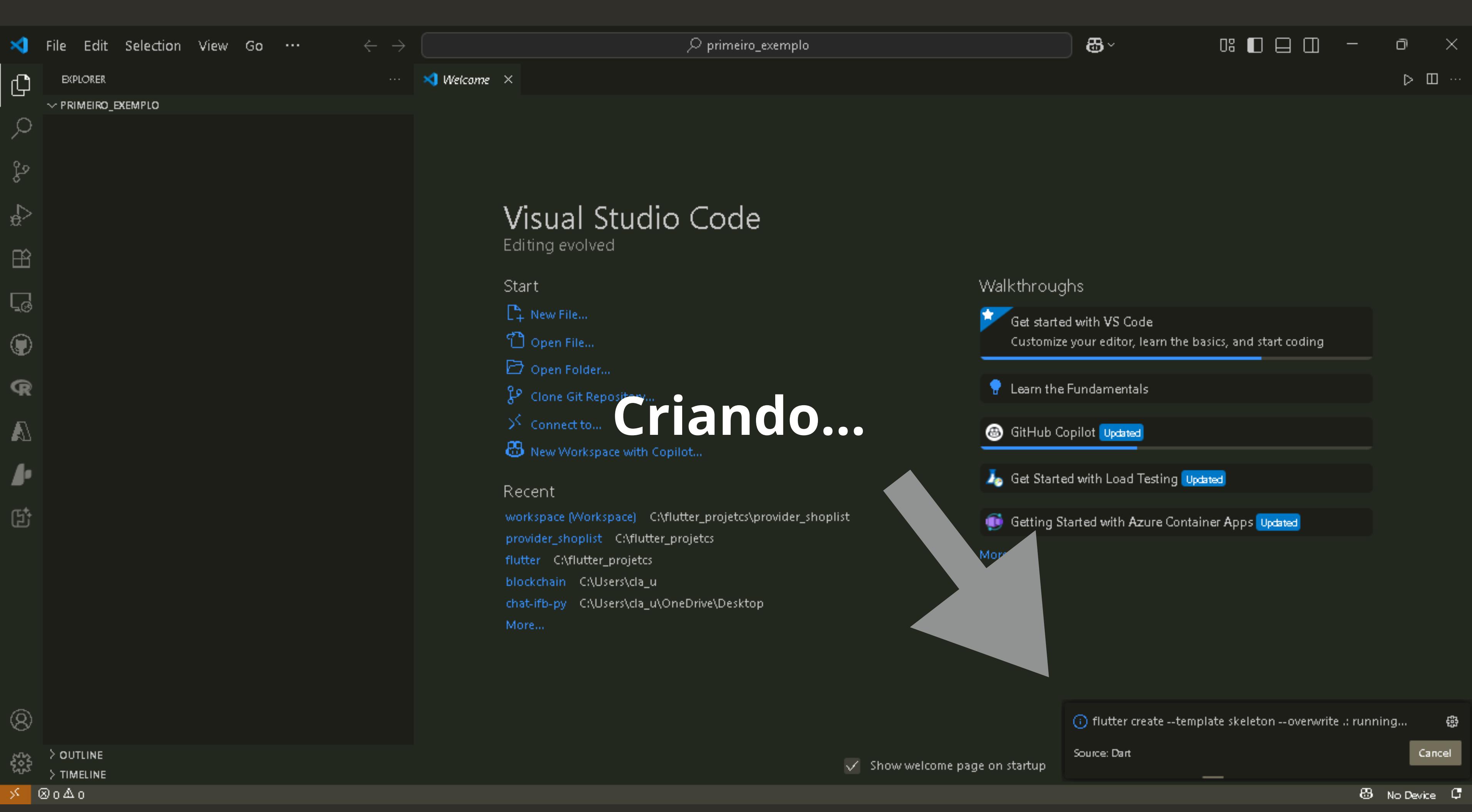
Find in Files **Ctrl + Shift + F**

Toggle Full Screen **F11**





**Nomear o projeto, nesse caso:  
primeiro\_exemplo**



The screenshot shows a Flutter project named "primeiro\_exemplo" open in an IDE. The project structure in the Explorer panel includes ".dart\_tool", ".idea", "android", "ios", and a "lib" folder containing "main.dart". The "main.dart" file is selected in the editor, displaying the following Dart code:

```
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(const MainApp());
5 }
6
7 class MainApp extends StatelessWidget {
8   const MainApp({super.key});
9
10 @override
11 Widget build(BuildContext context) {
12   return const MaterialApp(
13     title: 'Flutter Demo',
14     theme: ThemeData(
15       colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
16       useMaterial3: true,
17     ),
18   );
19 }
```

A large black semi-transparent overlay with the text "Terminou!" (Completed!) is centered over the code editor.

The terminal output below the editor shows:

```
PROBLEMS OUTPUT TERMINAL PORTS AZURE Filter flutter (primeiro_exemplo)
All done!
You can find general documentation for Flutter at: https://docs.flutter.dev/
Detailed API documentation is available at: https://api.flutter.dev/
If you prefer video documentation, consider: https://www.youtube.com/c/flutterdev
```

In order to run your empty application, type:

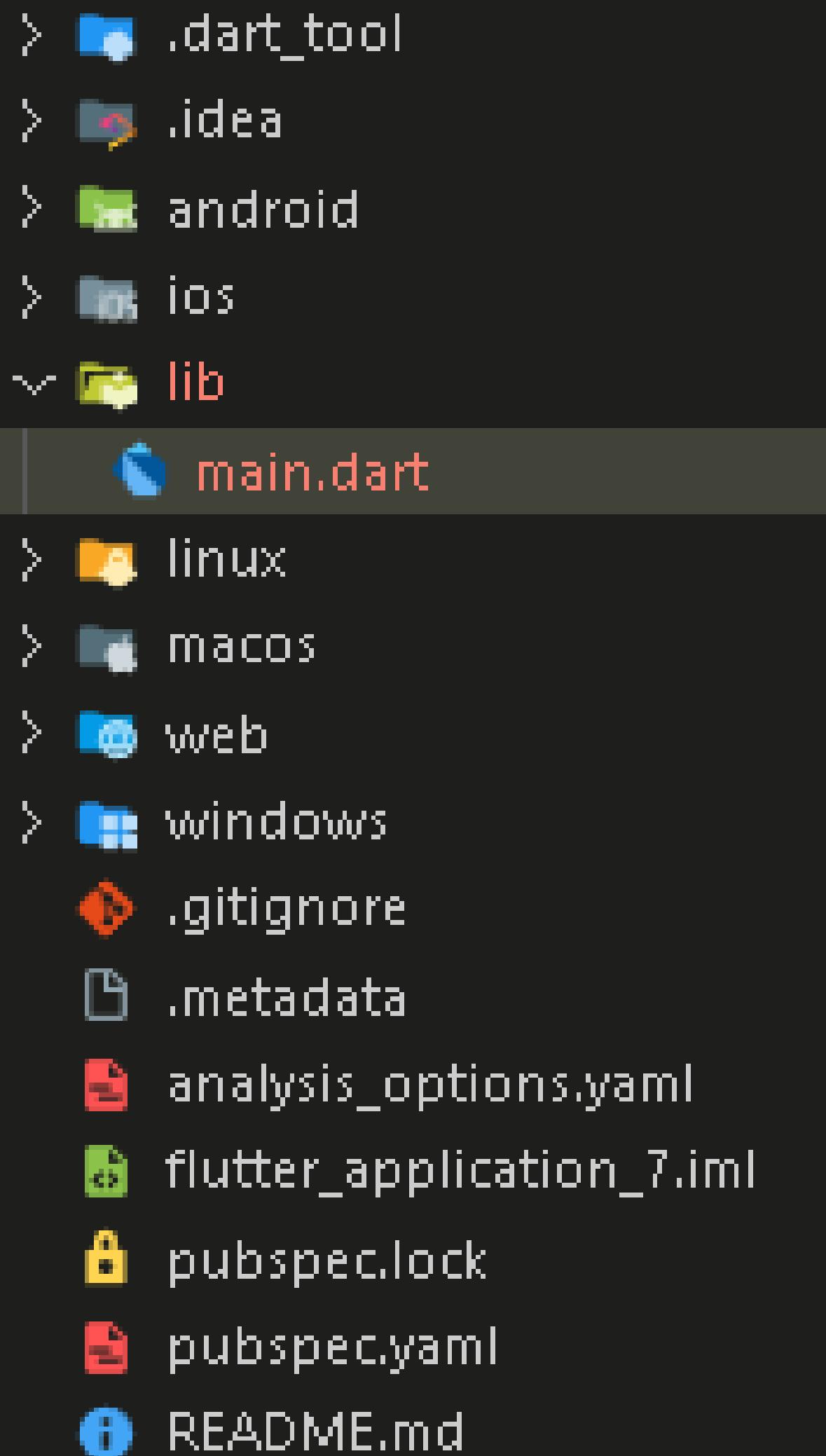
```
$ cd .
$ flutter run
```

Your empty application code is in .\lib\main.dart.

exit code 0

Bottom right corner message: Your Flutter project is ready! Press F5 to start running.

**Agora vamos ver a estrutura do projeto**



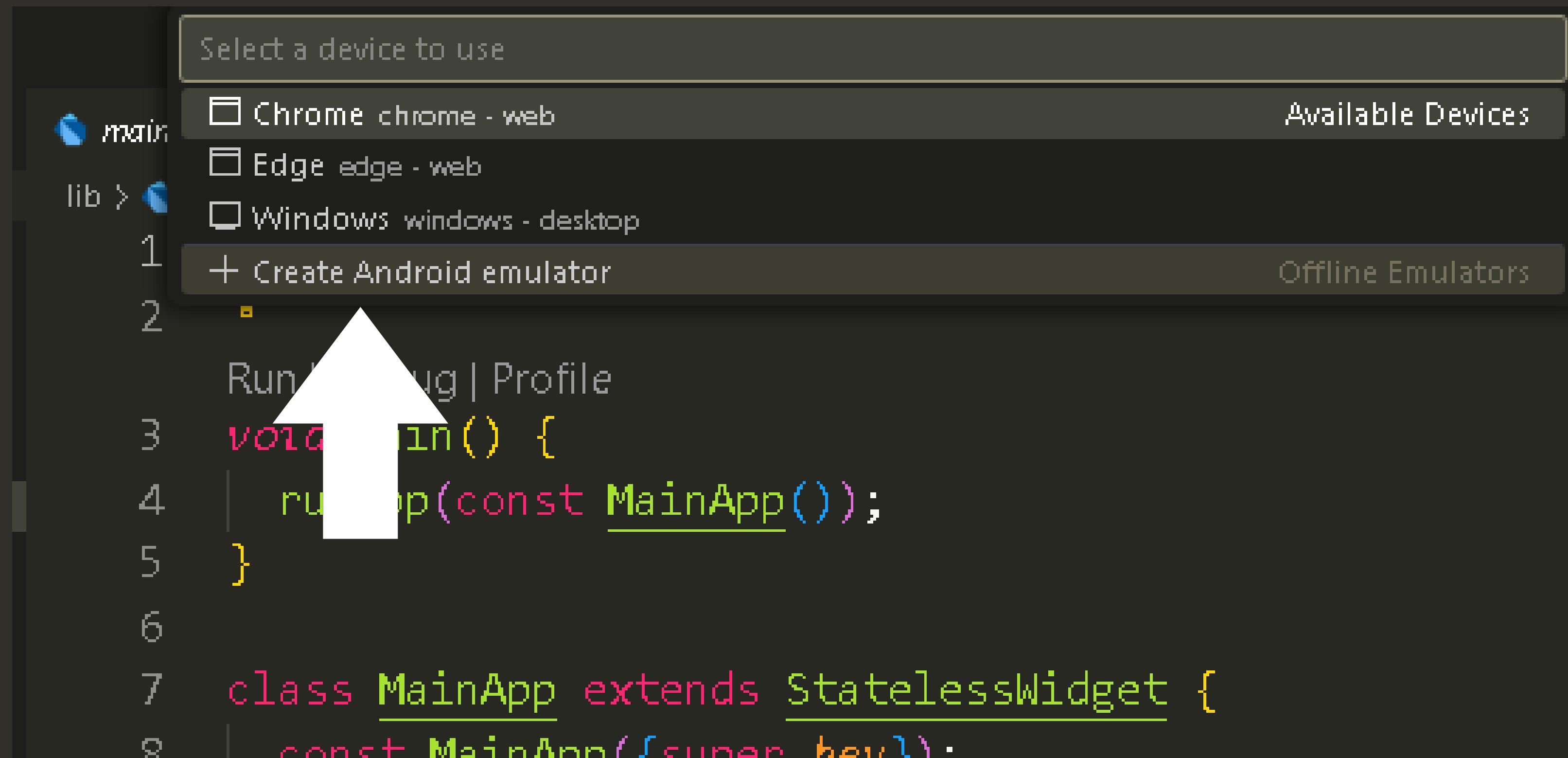
- **.dart\_tool**: gerenciamento de dependências e ferramentas.
- **.idea** (se usar Android Studio): Configurações específicas do Android Studio.
- **android**: Arquivos de configuração e código para a plataforma Android.
- **ios**: Arquivos de configuração e código para a plataforma iOS.
- **lib**: Onde reside o código principal do aplicativo, escrito em Dart.
- **linux**: Arquivos de configuração e código para a plataforma Linux.
- **macos**: Arquivos de configuração e código para a plataforma MacOS.
- **web** (se o projeto suportar web): Arquivos de configuração para a plataforma web.
- **Windows**: Arquivos de configuração e código para a plataforma Sistema Windows
- **.gitignore**: Define quais arquivos/pastas devem ser ignorados pelo controle de versão.
- **.metadata**: informações internas sobre o projeto.
- **analysis\_options.yaml**: Configurações para análise estática de código.
- **flutter\_application.iml**: importante para quem usa o Android Studio ou IntelliJ IDEA.
- **pubspec.lock**: Registra as versões exatas das dependências instaladas.
- **pubspec.yaml**: Arquivo de configuração do projeto.
- **README.md**: documentação do projeto

**Agora vamos rodar o projeto.**

**Podemos executar com duas opções:**

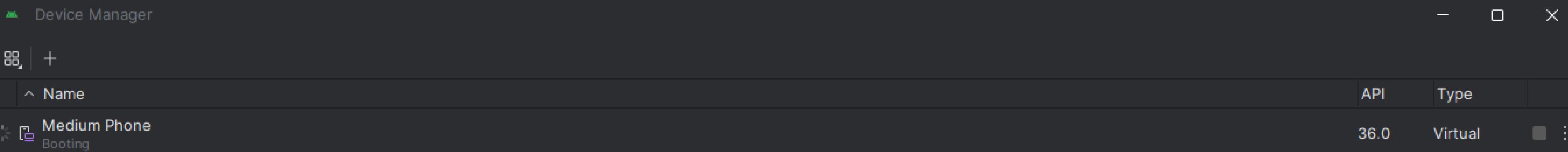
- Web
- Emulador Android

# Aperte F5, aqui temo várias opções, vamos começar pelo emulador Android



```
Select a device to use
Available Devices
main
lib >
1   + Create Android emulator
2
Run | Debug | Profile
3 void main() {
4   runApp(const MainApp());
5 }
6
7 class MainApp extends StatelessWidget {
8   const MainApp({super.key});
```

**Se preferir, pode abrir o Android Studio e no Virtual Device Manager pode ligar um emulador em separado.**



Name	API	Type
Medium Phone Booting	36.0	Virtual

No Configurations    main.dart    ...    ...    ...

lib > main.dart > ...

```
import 'package:flutter/material.dart';
void main() {
  runApp(const MainApp());
}

class MainApp extends StatelessWidget {
  const MainApp({super.key});

  @override
  Widget build(BuildContext context) {
    return const MaterialApp(
      home: Scaffold(
        body: Center(
          child: Text('Hello World'),
        ),
      ),
    );
  }
}
```

Run | Debug | Profile

void main() {  
 runApp(const MainApp());  
}  
  
class MainApp extends StatelessWidget {  
 const MainApp({super.key});  
  
 @override  
 Widget build(BuildContext context) {  
 return const MaterialApp(  
 home: Scaffold(  
 body: Center(  
 child: Text('Hello World'),  
 ),  
 ),  
 );  
 }  
}

IS    OUTPUT    TERMINAL    PORTS    AZURE

CONSOLE

```
g. text (exclude, \escape)
```

loading android-arm-profile/windows-x64 tools...  
loading android-arm-release/windows-x64 tools...  
loading android-arm64-profile/windows-x64 tools...  
loading android-arm64-release/windows-x64 tools...  
loading android-x64-profile/windows-x64 tools...  
loading android-x64-release/windows-x64 tools...  
...  
ching lib/main.dart on sdk gphone64 x86 64 in debug mode...

TERMINAL

```
PS C:\flutter_projects\primeiro_exemplo>
```

Fique monitorando o andamento, as vezes pode falhar pela demora do emulador. Nesse caso repetir a operação até conseguir. Emulador requer bastante processamento. Pode demorar mais de meia hora.

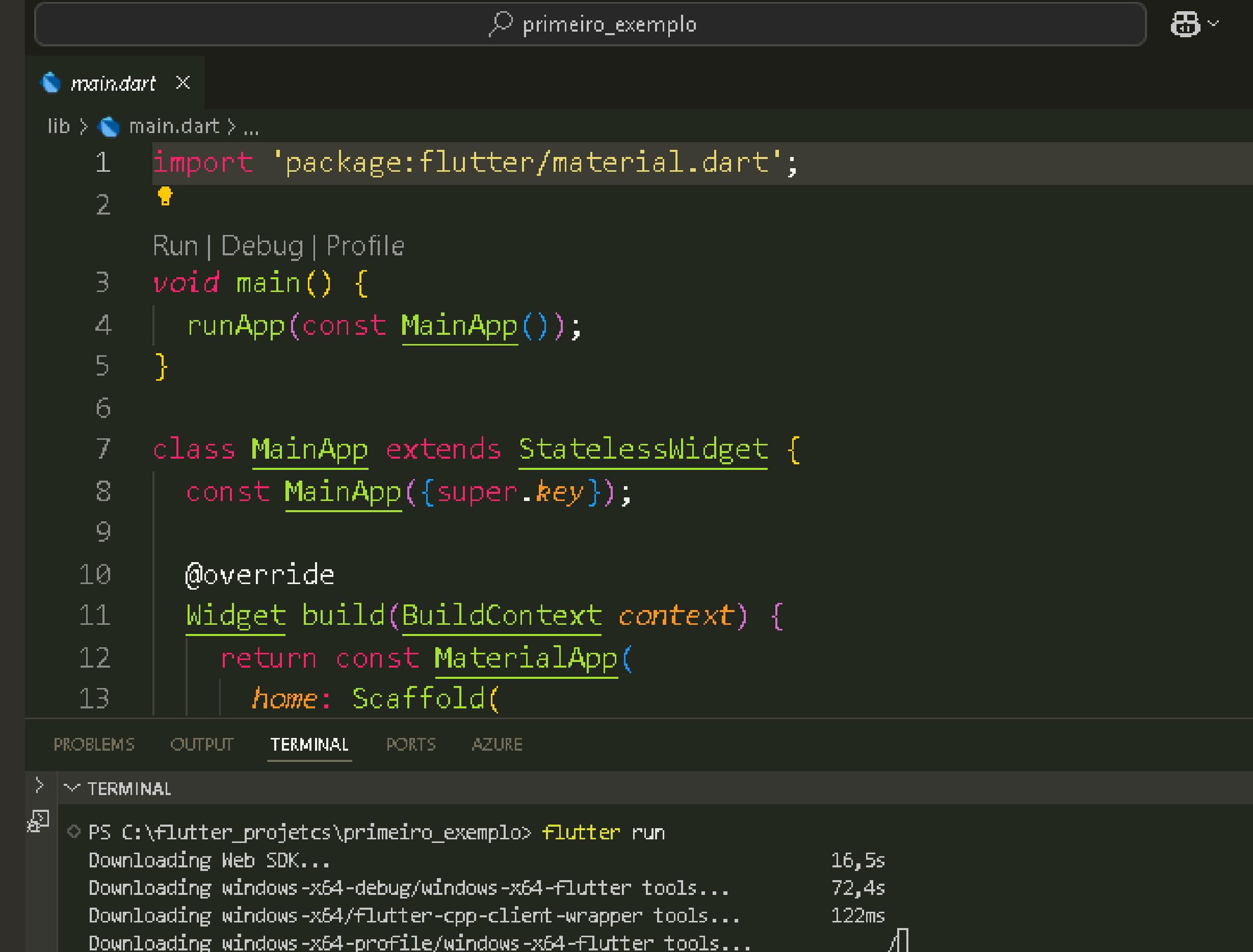


Flutter: Running Gradle task 'assembleDebug'...

Failed to launch flutter\_emulator: Error: Emulator di...

Outra maneira  
de rodar o app é  
abrir um new  
Terminal e no  
diretorio do  
projeto digitar  
**'flutter run'**

O emulador  
deve estar  
rodando



```
main.dart
```

```
lib > main.dart > ...
1 import 'package:flutter/material.dart';
2 void main() {
3   runApp(const MainApp());
4 }
5
6
7 class MainApp extends StatelessWidget {
8   const MainApp({super.key});
9
10 @override
11 Widget build(BuildContext context) {
12   return const MaterialApp(
13     home: Scaffold(

```

PROBLEMS    OUTPUT    TERMINAL    PORTS    AZURE

> ~ TERMINAL

PS C:\flutter\_projects\primeiro\_exemplo> flutter run

Time	Message
16,5s	Downloading Web SDK...
72,4s	Downloading windows-x64-debug/windows-x64-flutter tools...
122ms	Downloading windows-x64/flutter-cpp-client-wrapper tools...
	Downloading windows-x64-profile/windows-x64-flutter tools...

Rodando com  
sucesso.

Reparam os  
comando r e R  
para fazer o  
hotreload

The screenshot shows a Flutter project in a code editor. A context menu is open over the code, showing options like 'Available Devices' and 'Offline Emulators'. The code itself is a simple 'Hello World' application:

```
Widget build(BuildContext context) {  
  return const MaterialApp(  
    home: Scaffold(  
      body: Center(  
        child: Text('Hello World!'),  
      ), // Center  
    ), // Scaffold  
  ); // MaterialApp  
}
```

The 'TERMINAL' tab in the bottom left shows the command `flutter run` being executed, with the output indicating the app is running on an 'sdk gphone64 x86 64' device. The right side of the screen displays the 'Android Emulator - Medium\_Phone:5554' window, which shows the text 'Hello World!' on the screen. The emulator interface includes various control icons for power, volume, camera, search, and orientation.

```
PS C:\flutter_projects\primeiro_exemplo> flutter run  
Launching lib\main.dart on sdk gphone64 x86 64 in debug mode...  
Running Gradle task 'assembleDebug'...  
✓ Built build\app\outputs\flutter-apk\app-debug.apk  
Installing build\app\outputs\flutter-apk\app-debug.apk...  
I/flutter ( 7545): [IMPORTANT:flutter/shell/platform/android/android_context_gl_impeller.cc:  
Syncing files to device sdk gphone64 x86 64...  
  
Flutter run key commands.  
r Hot reload.  
R Hot restart.  
h List all available interactive commands.  
d Detach (terminate "flutter run" but leave application running).  
c Clear the screen  
q Quit (terminate the application on the device).
```

A Dart VM Service on `sdk gphone64 x86 64` is available at: <http://127.0.0.1:63109/PX6BrRNzmo>

Ln 1, Col 1 Spaces: 2 UTF-8 CR

Select a device to use

Chrome chrome - web

Available Devices

Edge edge - web

Windows windows - desktop

+ Create Android emulator

Offline Emulators

2

Run | Debug | Profile

```
3 void main() {  
4   runApp(const MainApp());  
5 }
```

6

```
7 class MainApp extends StatelessWidget {  
8   const MainApp({String key});
```

Existe, entre outras, a possibilidade de rodar para navegadores web. Ou usando o comando flutter run, escolher o navegador preferido

Fim

Chegamos ao fim desse tutorial, agora está pronto para desenvolver em Flutter, vamos os próximos passo.  
Parabéns por ter terminado a etapa!

