

Adicionar Firebase no Flutter.

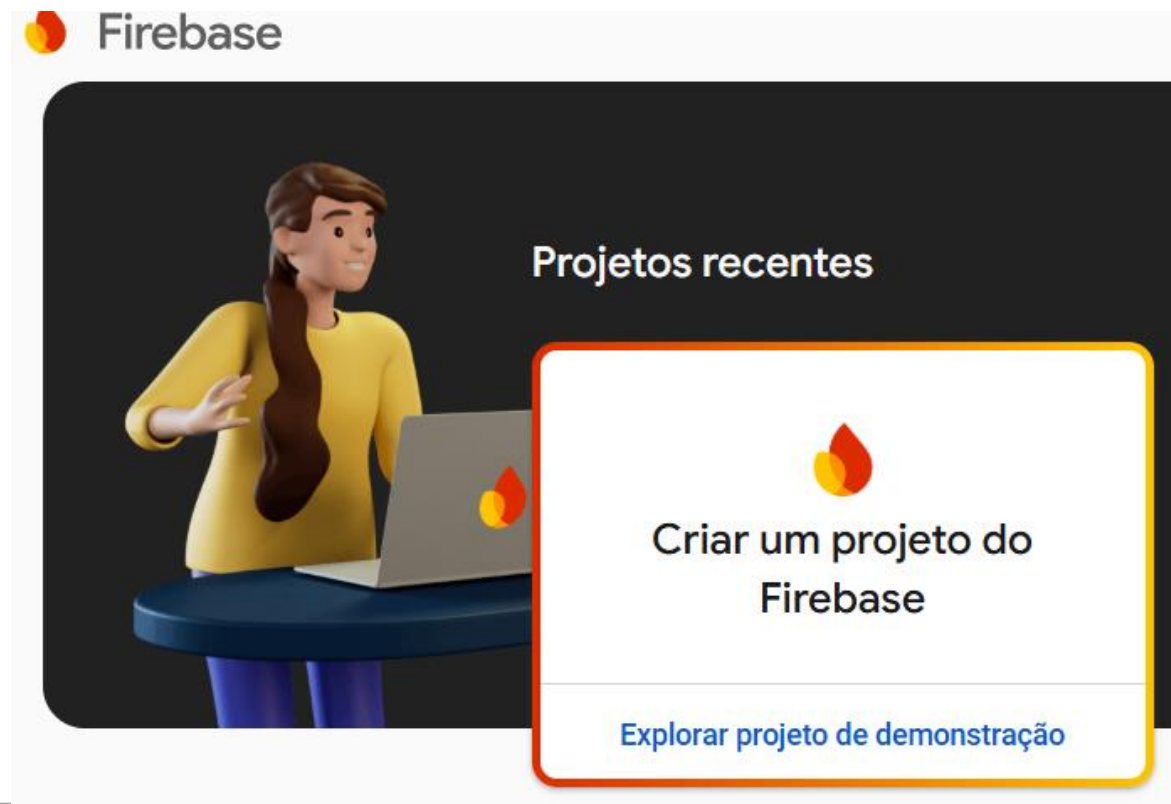


Prof Vinicius Almeida.



1º Passo.

- Criar um projeto.



2º Passo.

- Na interface do Firebase crie um projeto flutter.



3º Passo.

- Instalar a CDI do firebase.



firebase-tools-instant-win.exe

163 MB • 6 minutos atrás

4º Passo.

- Executar CLI .
- Aceitar opções com Y.
- Logar no navegador.

[illegible]

5º Passo.

- Adicionar variaveis de ambiente.

`C:\Users\vihha\AppData\Local\Pub\Cache\bin`

Adicione o comando : `npm install -g firebase-tools`

6º Passo.

Coloque estes 2 comandos na raiz do seu projeto.
Cada projeto tem comandos diferentes.

2 Instalar e executar a CLI do FlutterFire

Em qualquer diretório, execute o comando:

```
$ dart pub global activate flutterfire_cli
```



Em seguida, na raiz do diretório do seu projeto do Flutter, execute o comando:

```
$ flutterfire configure --project=climateste
```



Com isso, seus apps são registrados automaticamente por plataforma com o Firebase, e um arquivo de configuração `lib/firebase_options.dart` é adicionado ao seu projeto do Flutter.

[Anterior](#)

[Próxima](#)

7º Passo.

- Importe 2 bibliotecas do firebase e modifique a sua função main para esperar a inicialização.

```
import 'package:flutter/material.dart';
```

```
import 'package:firebase_core/firebase_core.dart';  
import 'firebase_options.dart';
```

```
void main() async {
```

```
  WidgetsFlutterBinding.ensureInitialized();
```

```
  await Firebase.initializeApp(  
    options: DefaultFirebaseOptions.currentPlatform,
```

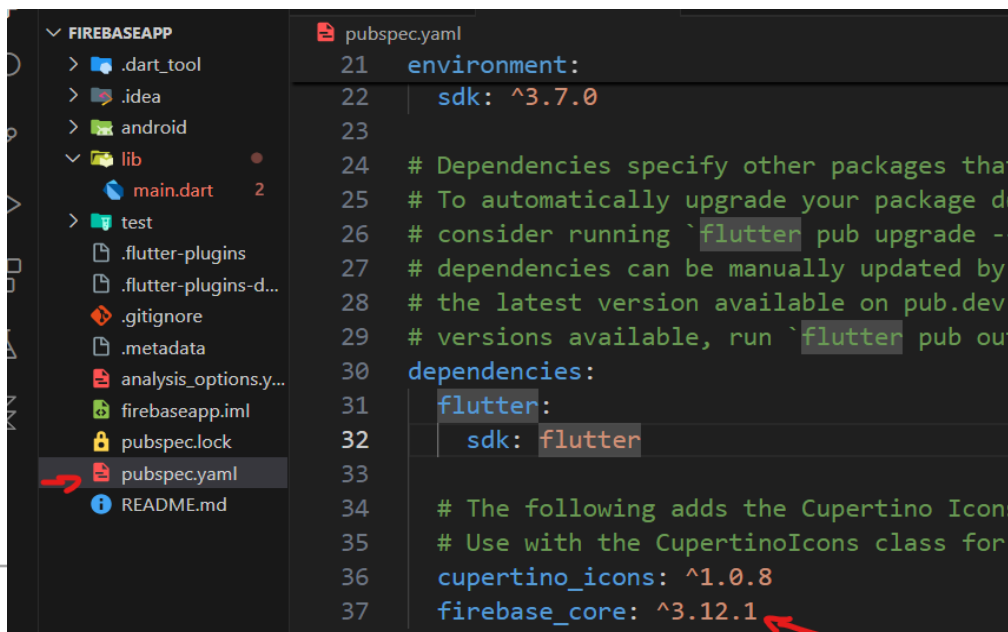
```
  );
```

```
  runApp(const MyApp());
```

```
}
```


8º Passo.

- No pubspec.yaml importe as bibliotecas do firebase.
- firebase_core: ^3.12.1



The screenshot shows an IDE interface with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'FIREBASEAPP' with various files and folders. The 'pubspec.yaml' file is selected and highlighted with a red arrow. The code editor displays the contents of 'pubspec.yaml', showing the 'environment' section with 'sdk: ^3.7.0' and the 'dependencies' section with 'flutter: sdk: flutter' and 'firebase_core: ^3.12.1'. A red arrow points to the 'firebase_core: ^3.12.1' line in the dependencies section.

```
pubspec.yaml
21 environment:
22   sdk: ^3.7.0
23
24 # Dependencies specify other packages that
25 # To automatically upgrade your package de
26 # consider running `flutter pub upgrade --
27 # dependencies can be manually updated by
28 # the latest version available on pub.dev
29 # versions available, run `flutter pub out
30 dependencies:
31   flutter:
32     sdk: flutter
33
34 # The following adds the Cupertino Icons
35 # Use with the CupertinoIcons class for
36 cupertino_icons: ^1.0.8
37 firebase_core: ^3.12.1
```

9º Passo.

- Adicione outra biblioteca: cloud_firestore

```
cupertino_icons: ^1.0.8  
firebase_core: ^3.12.1  
cloud_firestore: ^5.6.5
```

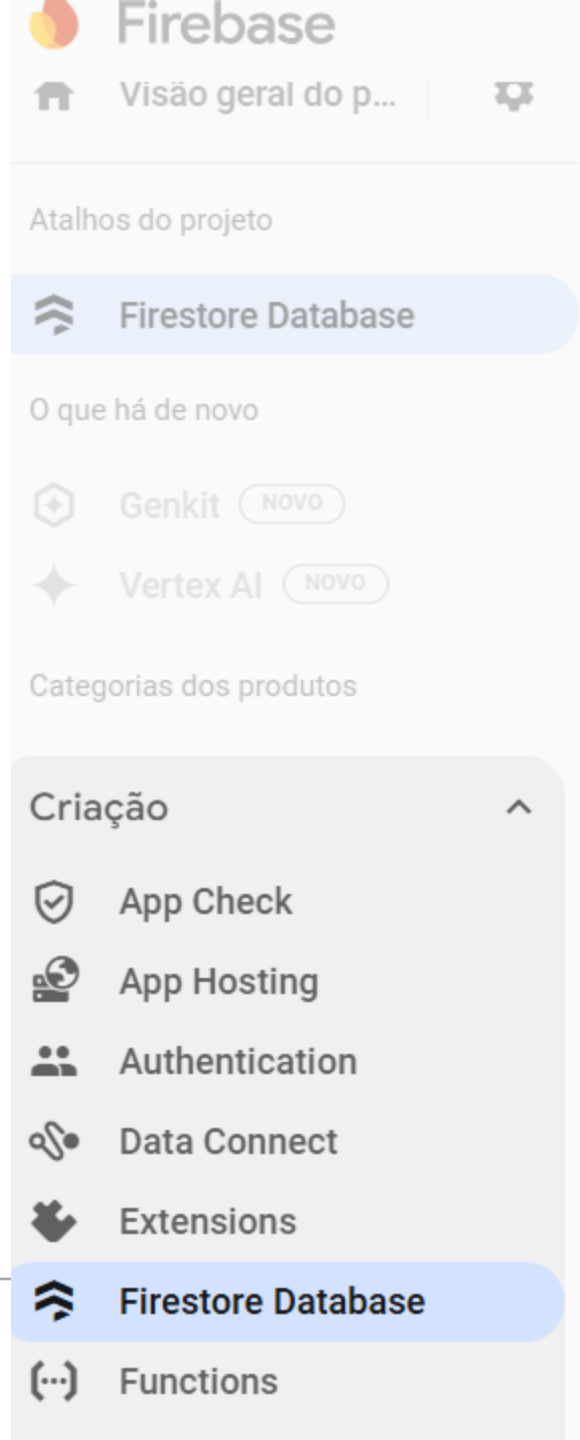
10º Passo.

- Na classe que você criar depois do @override adicione a função, ela faz iniciar o firebase, caso você queira adicionar uma função que o firebase constroi deve-se colocar aqui dentro.

- `@override`
- `void initState() {`
- `super.initState();`
-
- `}`

11º Passo.

- Crie um database.



12º Passo.

- Deixe o banco de dados para produção. E ative o write para true.

```
1 rules_version = '2';
2
3 service cloud.firestore {
4     match /databases/{database}/documents {
5         match /{document=**} {
6             allow read, write: if true;
7         }
8     }
9 }
```