

EXEMPLO B

Crie um aplicativo no Android Studio.

File | New Flutter Project | Flutter Application

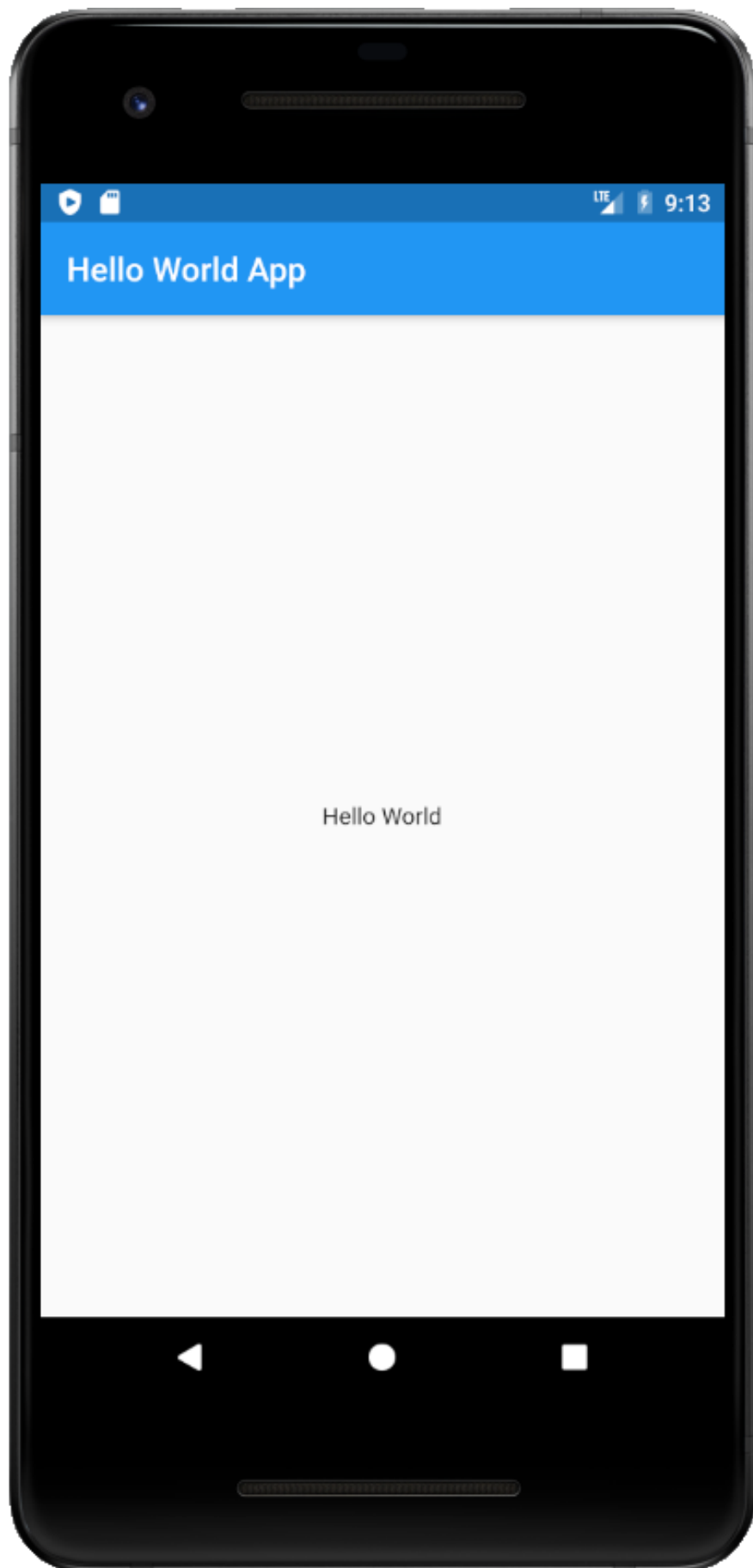
Copie o exemplo do HelloWorldApp na Wikipédia no verbete Flutter.

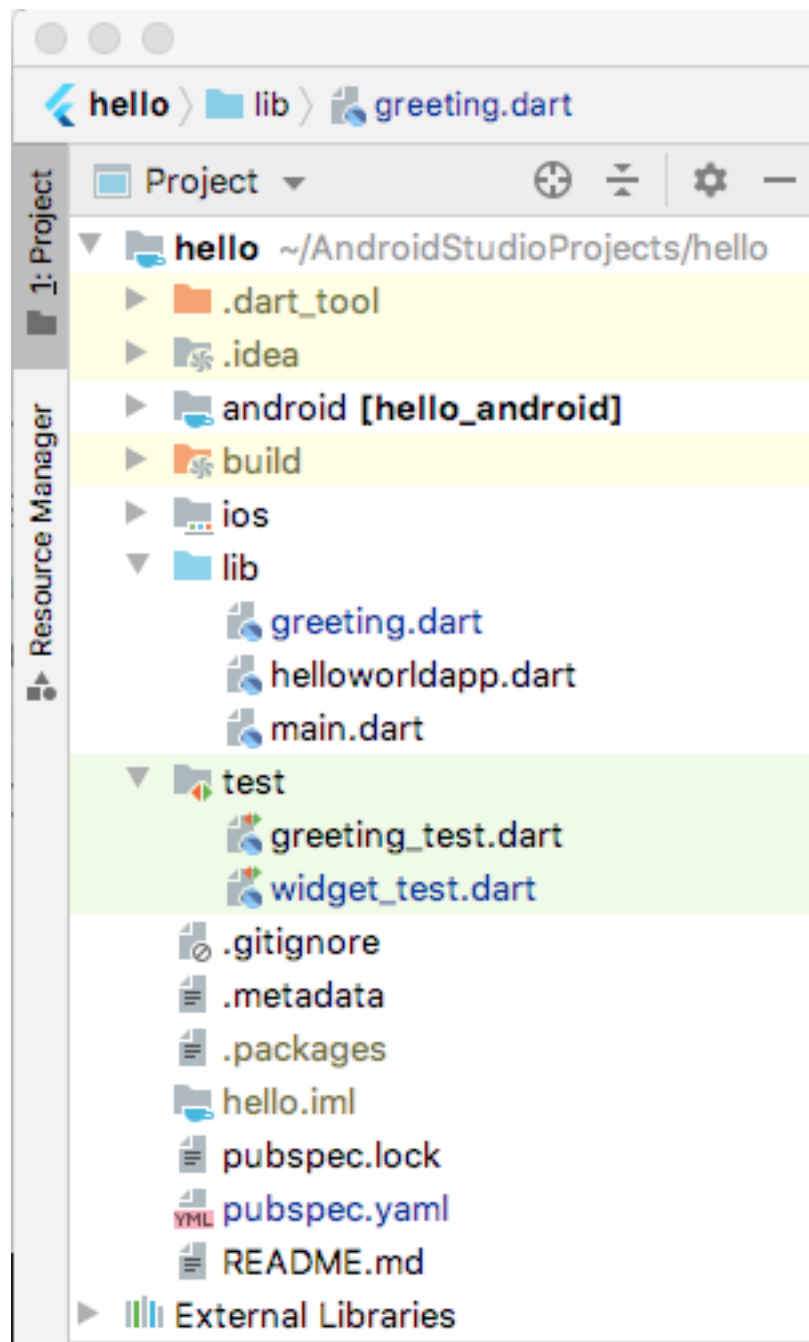
Crie uma classe Greeting.

Crie um teste unitário para a classe Greeting.

Coloque as classes Greeting e HelloWorldApp fora do arquivo main.dart/

Crie um teste de interface de usuário (teste de *widget*) para o aplicativo.





```
class Greeting {  
  String salutation;  
  
  Greeting() {  
    salutation = "Hello World";  
  }  
}
```

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

class HelloWorldApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {

    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Hello World App',

      home: Scaffold(
        appBar: AppBar(
          title: Text('Hello World App'),
        ),

        body: Center(
          child: Text('Hello World'),
        ),
      ),
    );
  }
}
```

```
import 'package:flutter/material.dart';  
import 'helloworldapp.dart';  
  
void main() => runApp(HelloWorldApp());
```

```
import 'package:test/test.dart';
import 'package:hello/greeting.dart';

void main() {
  test('Hello World Salutation', () {
    expect(Greeting().salutation, 'Hello World');
  });
}
```

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

import 'package:hello/helloworldapp.dart';

void main() {
  testWidgets('Hello smoke test', (WidgetTester tester) async {
    await tester.pumpWidget(HelloWorldApp());

    expect(find.text('Hello World'), findsOneWidget);

  });
}
```


name: hello

description: A new Flutter application.

publish_to: 'none'

version: 1.0.0+1

environment:

sdk: ">=2.7.0 <3.0.0"

dependencies:

flutter:

sdk: flutter

cupertino_icons: ^0.1.3

dev_dependencies:

test:

flutter_test:

sdk: flutter

flutter:

uses-material-design: true

DESAFIO B1

Crie um novo projeto. Altere o aplicativo para que as classes sejam organizadas em arquivos separados. Acrescente uma classe Counter e altere o programa para usar um objeto Counter no lugar do tipo int. Acrescente um teste de unidade para a classe Counter.

Utilize os projetos dos DESAFIO A. Utilize a classe Counter do material sobre testes unitários do Flutter.dev.

DESAFIO B2

Altere o aplicativo para incrementar ou decrementar o contador. Acrescente um teste que indica que o valor mínimo do contador é zero. Altere a aplicação para desabilitar ou esconder o botão de decremento.

DESAFIO B3

Altere a aplicação para apresentar dois contadores, dois objetos da classe Counter.

DESAFIO B4

Crie uma classe `Adder`, que armazena dois números e permite obter a soma desses números. Crie testes unitários para classe `Adder`. Em Flutter, crie uma tela com um rótulo, duas caixas de texto e um botão. O usuário pode informar valores inteiros e pressionar para obter o resultado da soma. Crie um teste de widget para a tela da aplicação.

Utilize a página da API Flutter e do Material Design Components. Utilize o material do [Flutter.dev](https://flutter.dev) para testes.