



INTRODUÇÃO AO FLUTTER

JOÃO REICHERT

- iOS Developer @ Brave.ag
- Eng. Controle e Automação @ UFPel

<https://github.com/reeichert/>
joao.reichert@brave.com

CONTEÚDO

- Recap 
- O que é Flutter 
- Live Code 

RECAP

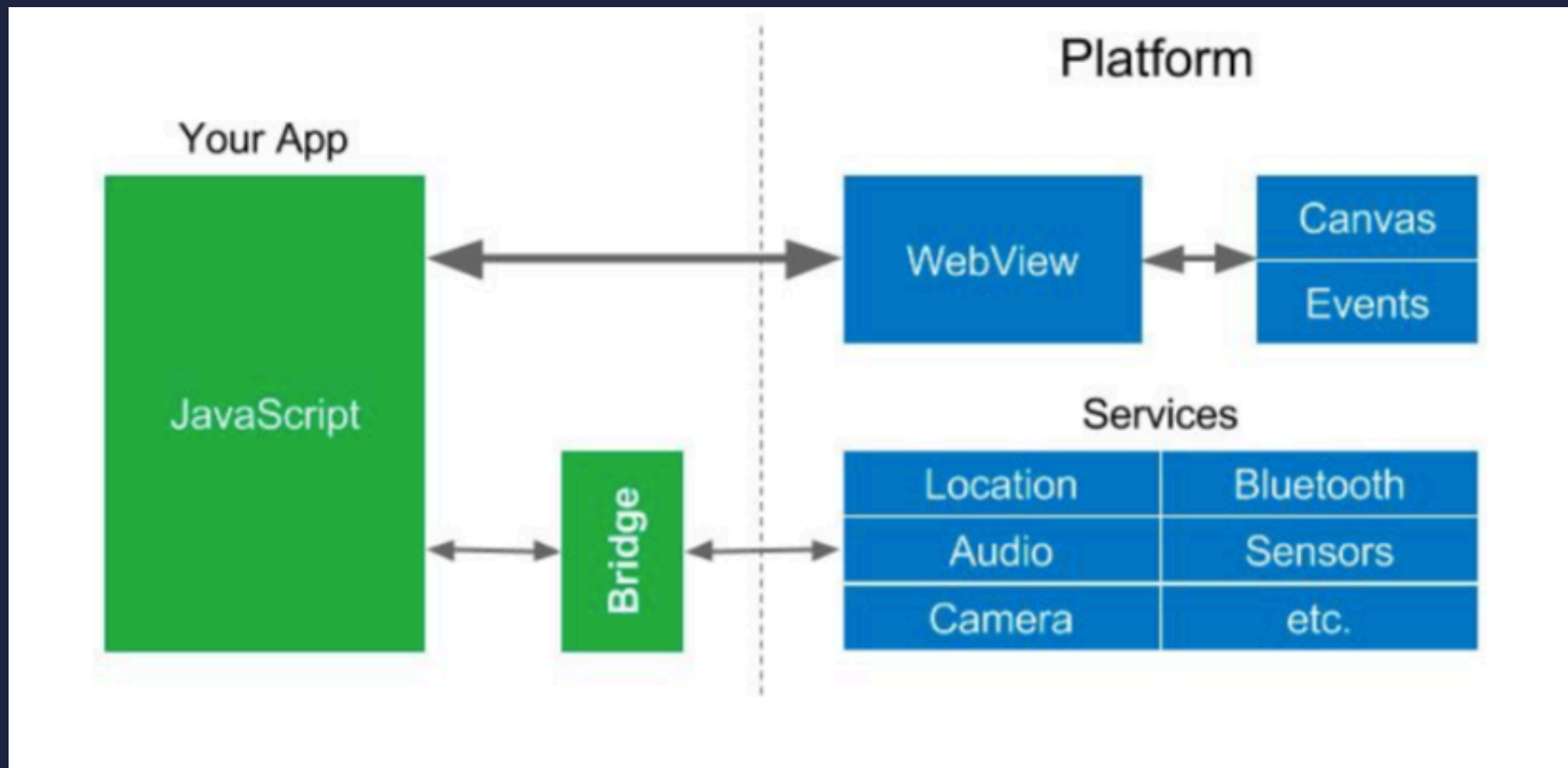


WRAPPER WEBVIEW



APACHE
CORDOVA™

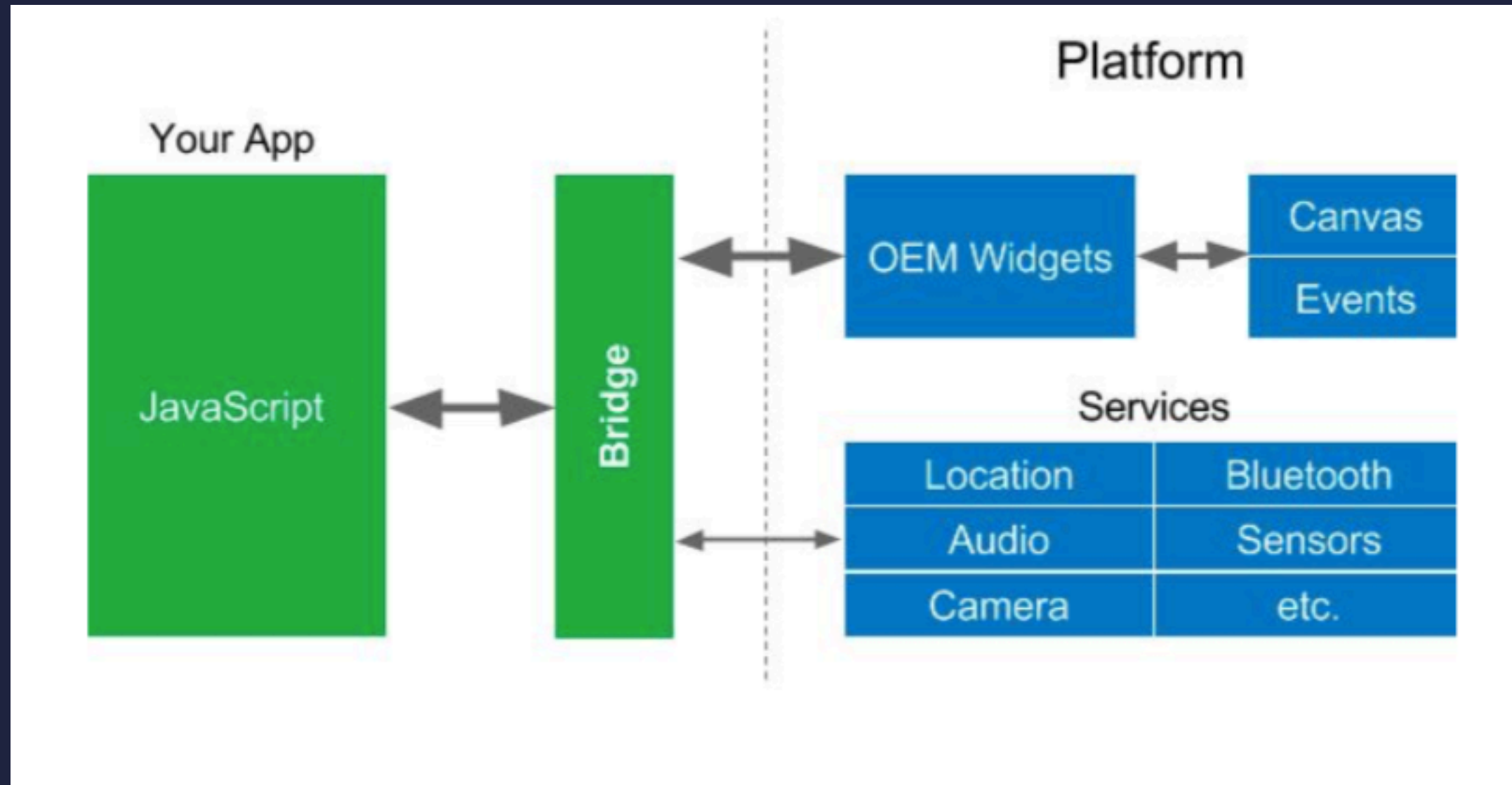
WRAPPER WEBVIEW



TRANSPILER



TRANSPILER



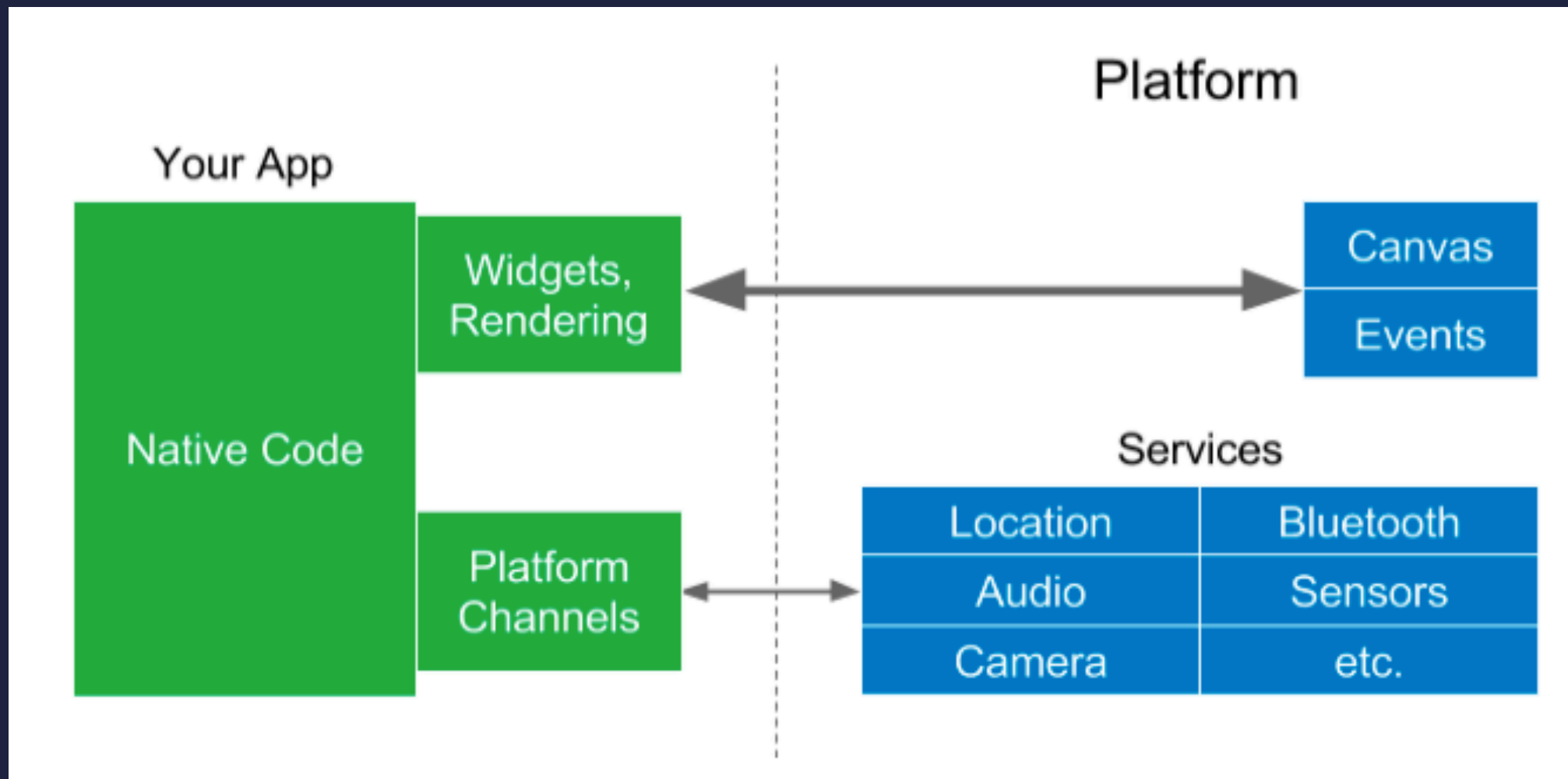


FLUTTER

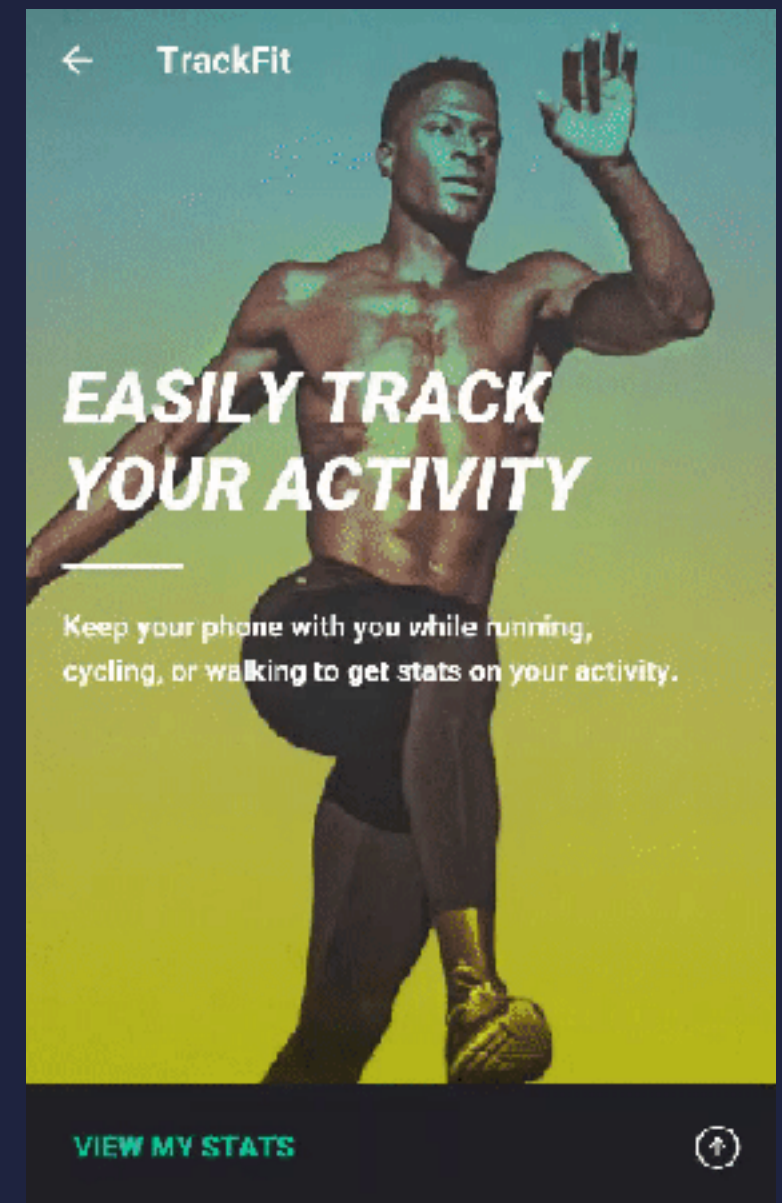
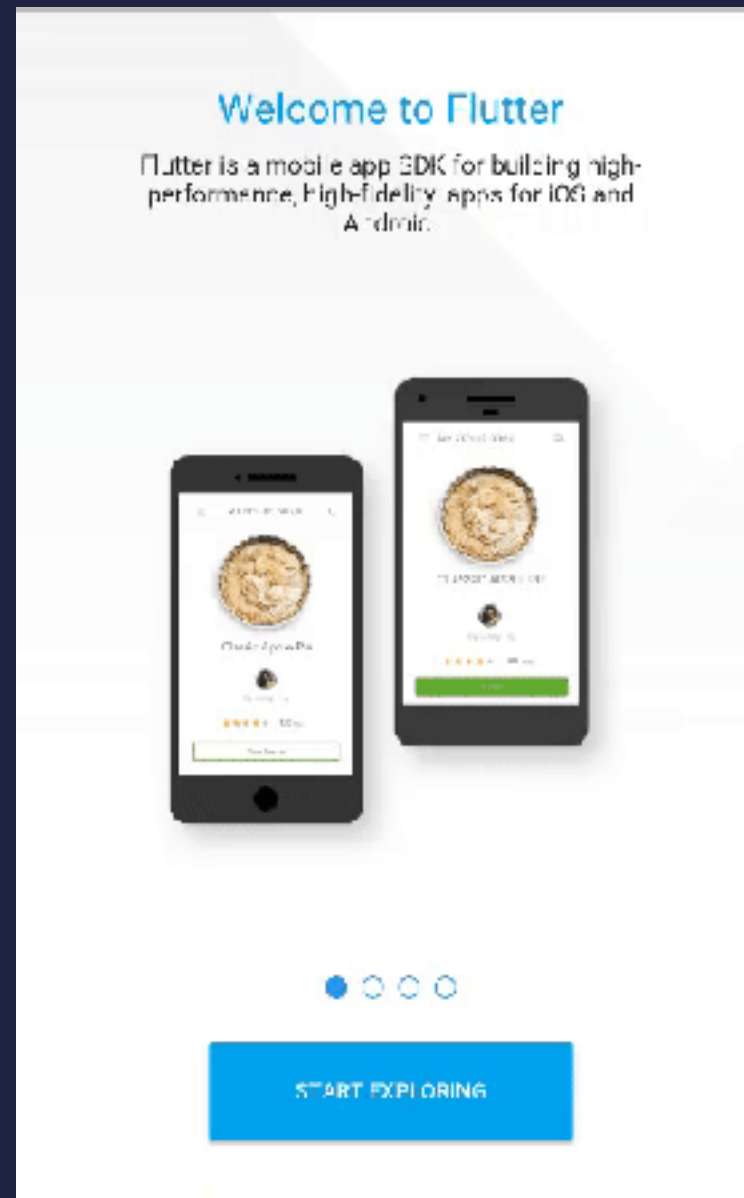
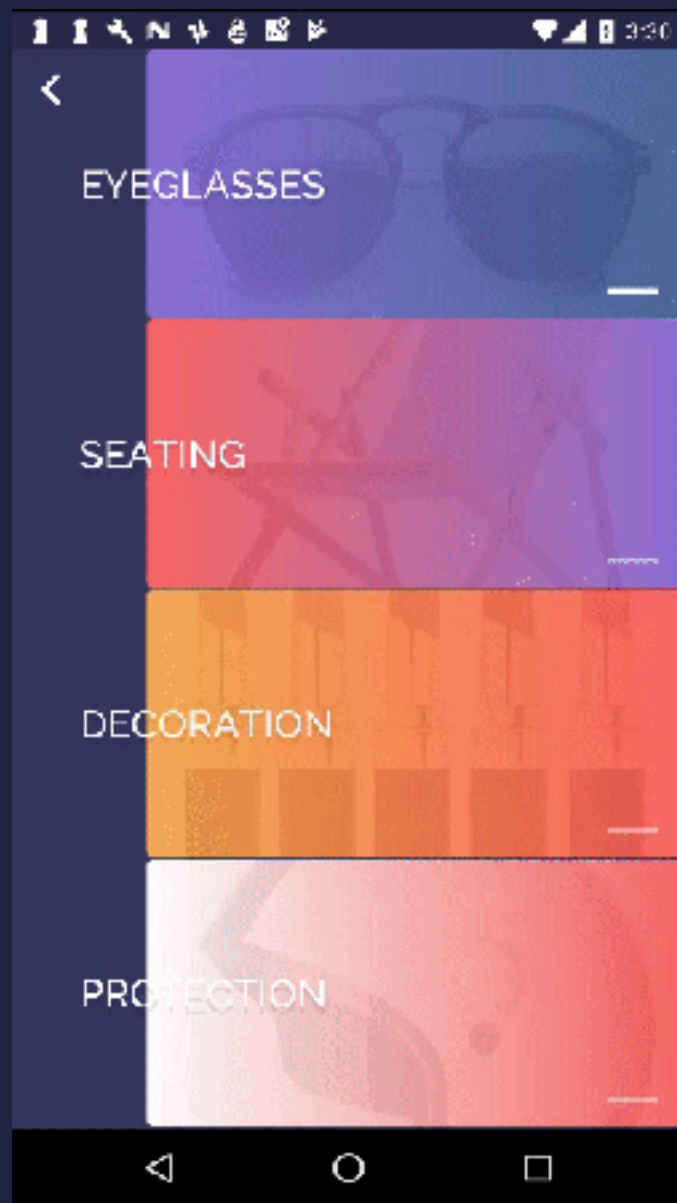
O QUE É FLUTTER? 🤔

- Um **SDK** para desenvolver apps **multiplataforma***
 - Foco em apps de **alta performance**, modernos e bonitos
 - Desenvolvido pelo **Google**, é **gratuito e open-source**
-
- ★ **Multiplataforma (iOS, Android e Fuchsia)**

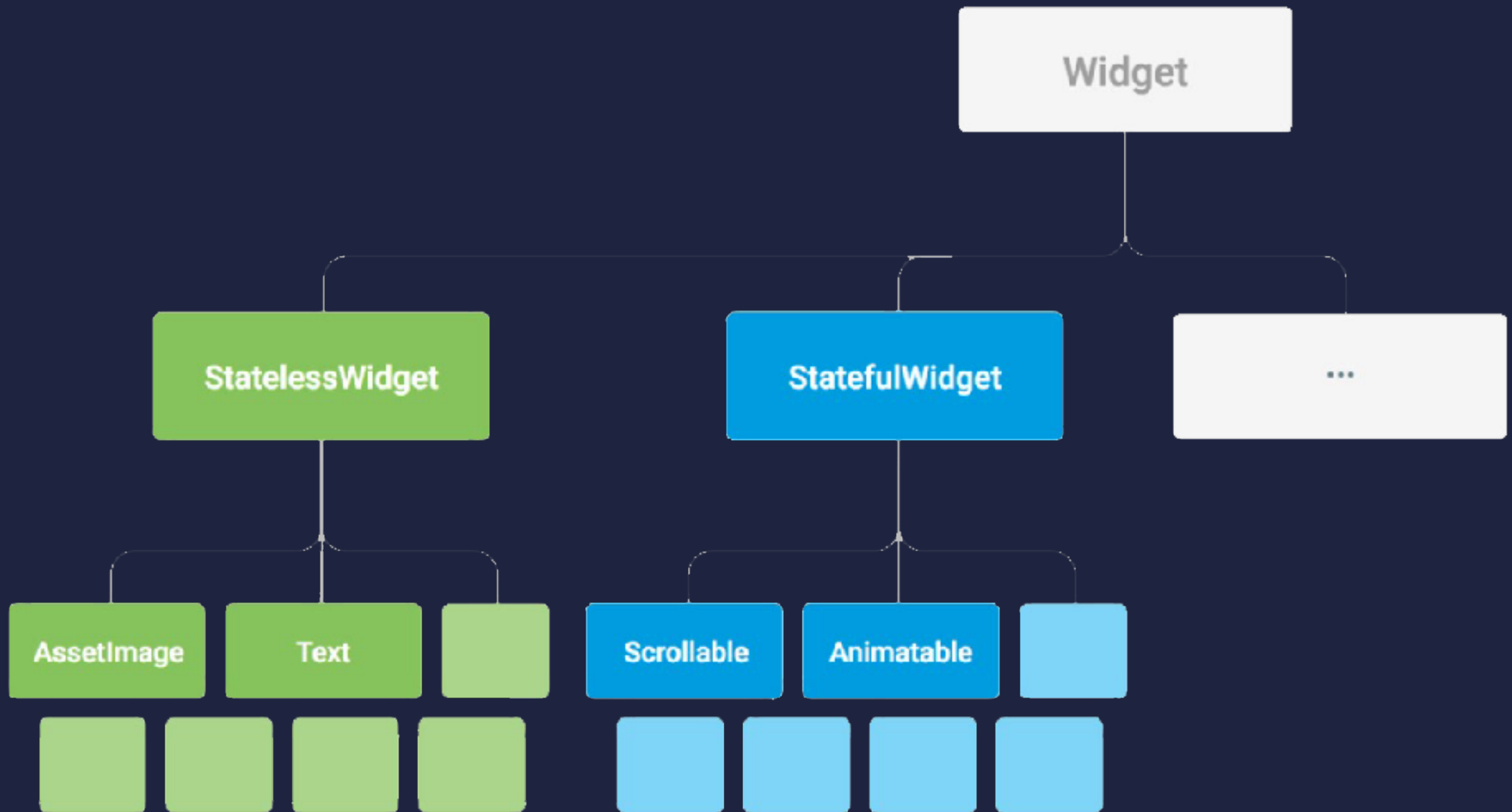
FLUTTER



EXEMPLOS



TUDO É WIDGET



WIDGETS

StatelessWidget

```
class HelloWorldPage extends StatelessWidget {  
  @override Widget build(BuildContext context) {  
    return Text('Hello World');  
  }  
}
```

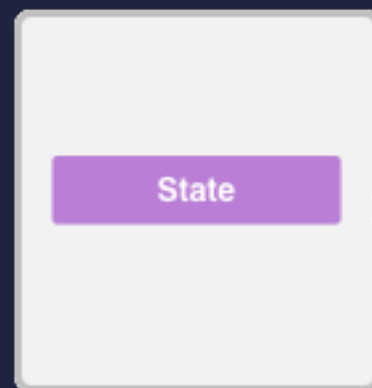
WIDGETS

StatefulWidget

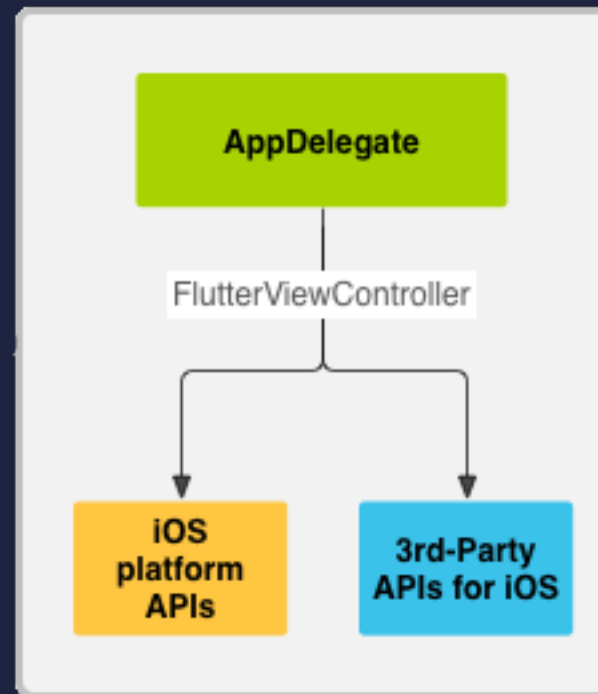
```
class Counter extends StatefulWidget {  
  @override  
  _CounterState createState() => _CounterState();  
}  
  
class _CounterState extends State<Counter> {  
  int _count = 0;  
  
  _increment() {  
    setState(() {  
      _count++;  
    });  
  }  
  
  @override  
  Widget build(BuildContext context) {  
    return Column(  
      children: [  
        Text('Current count is: $_count'),  
        RaisedButton(  
          onPressed: _increment,  
          child: Text('Increment the count!')  
        )  
      ],  
    );  
  }  
}
```

PLATFORM CHANNELS

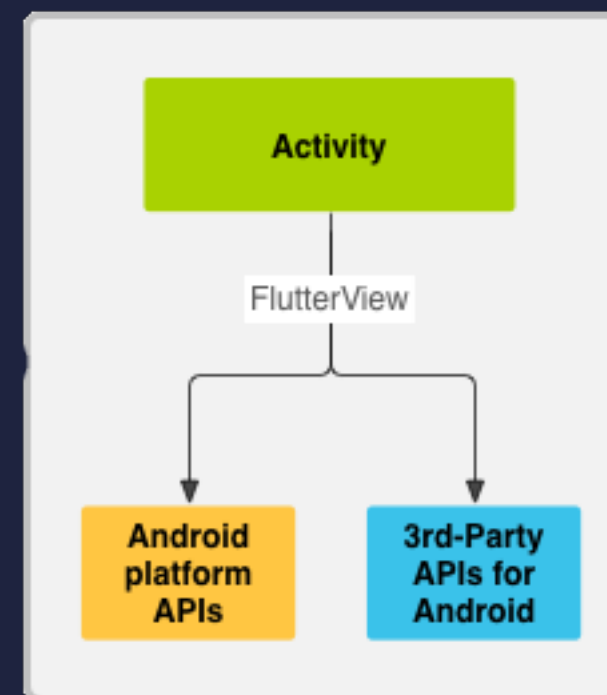
Flutter App (client)



MethodChannel



iOS Host



Android Host

PLATFORM CHANNELS

Kotlin

```
import android.os.Bundle
import io.flutter.app.FlutterActivity
import io.flutter.plugin.common.MethodChannel

class MainActivity() : FlutterActivity() {
    private val CHANNEL = "samples.flutter.io/battery"

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        MethodChannel(flutterView, CHANNEL).setMethodCallHandler { call, result ->
            // TODO
        }
    }
}
```

PLATFORM CHANNELS

Kotlin

```
MethodChannel(flutterView, CHANNEL).setMethodCallHandler { call, result ->
    if (call.method == "getBatteryLevel") {
        val batteryLevel = getBatteryLevel()
        if (batteryLevel != -1) {
            result.success(batteryLevel)
        } else {
            result.error("UNAVAILABLE", "Battery level not available.", null)
        }
    } else {
        result.notImplemented()
    }
}
```

PLATFORM CHANNELS

Swift

```
let controller : FlutterViewController = window?.rootViewController as! FlutterViewController;
let batteryChannel = FlutterMethodChannel.init(name: "samples.flutter.io/battery",
                                              binaryMessenger: controller);

batteryChannel.setMethodCallHandler({
    (call: FlutterMethodCall, result: FlutterResult) -> Void in
        // Handle battery messages.
    });
```

PLATFORM CHANNELS

Swift

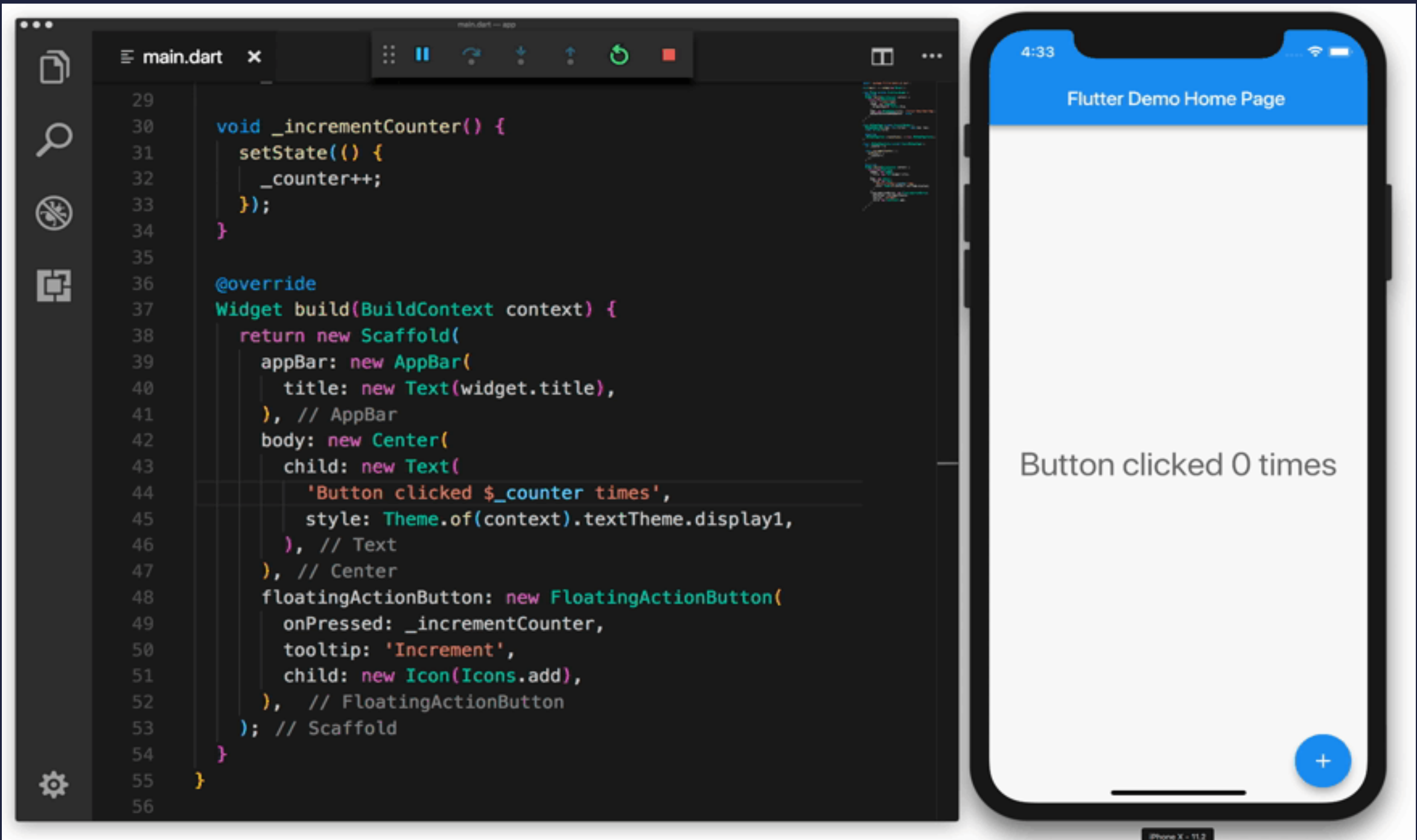
```
batteryChannel.setMethodCallHandler({
    (call: FlutterMethodCall, result: FlutterResult) -> Void in
    if ("getBatteryLevel" == call.method) {
        self.receiveBatteryLevel(result: result);
    } else {
        result(FlutterMethodNotImplemented);
    }
});
```

PLATFORM CHANNELS

Flutter

```
static const platform = const MethodChannel('samples.flutter.io/battery');  
// Get battery level.  
String _batteryLevel = 'Unknown battery level.';  
  
Future<Null> _getBatteryLevel() async {  
  String batteryLevel;  
  try {  
    final int result = await platform.invokeMethod('getBatteryLevel');  
    batteryLevel = 'Battery level at $result % .';  
  } on PlatformException catch (e) {  
    batteryLevel = "Failed to get battery level: '${e.message}'.";  
  }  
  
  setState(() {  
    _batteryLevel = batteryLevel;  
  });  
}
```

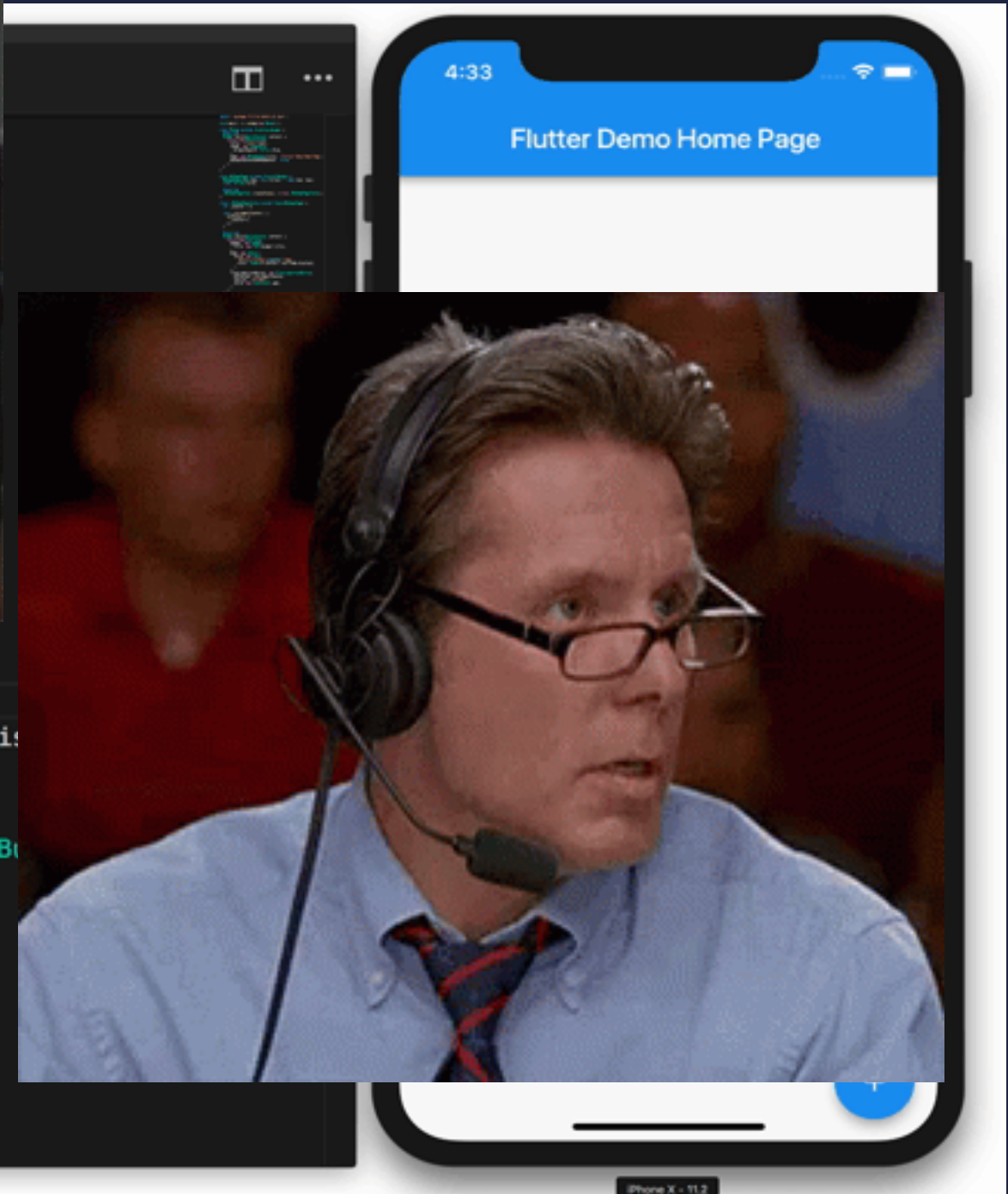
HOT RELOAD



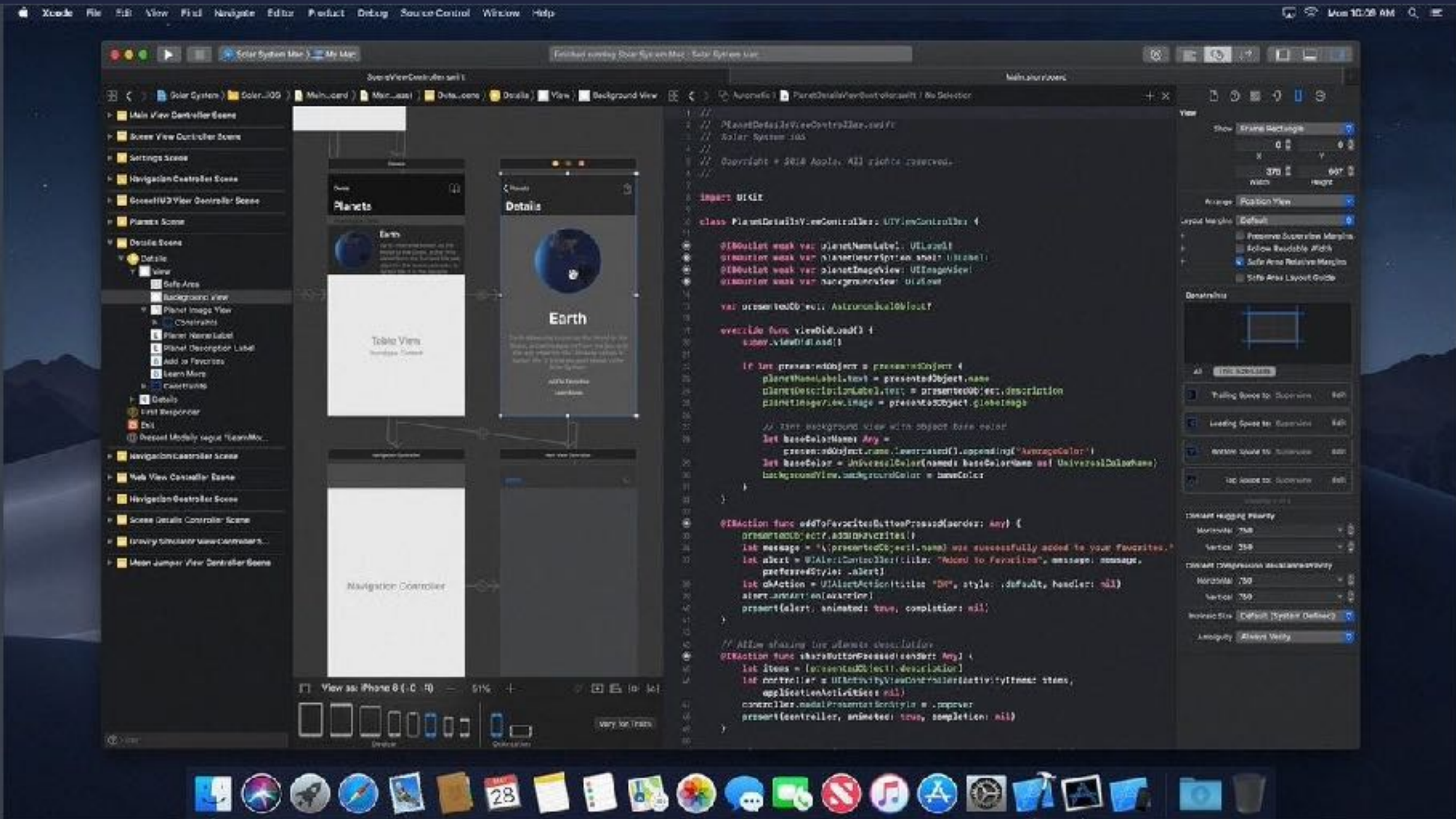
HOT RELOAD



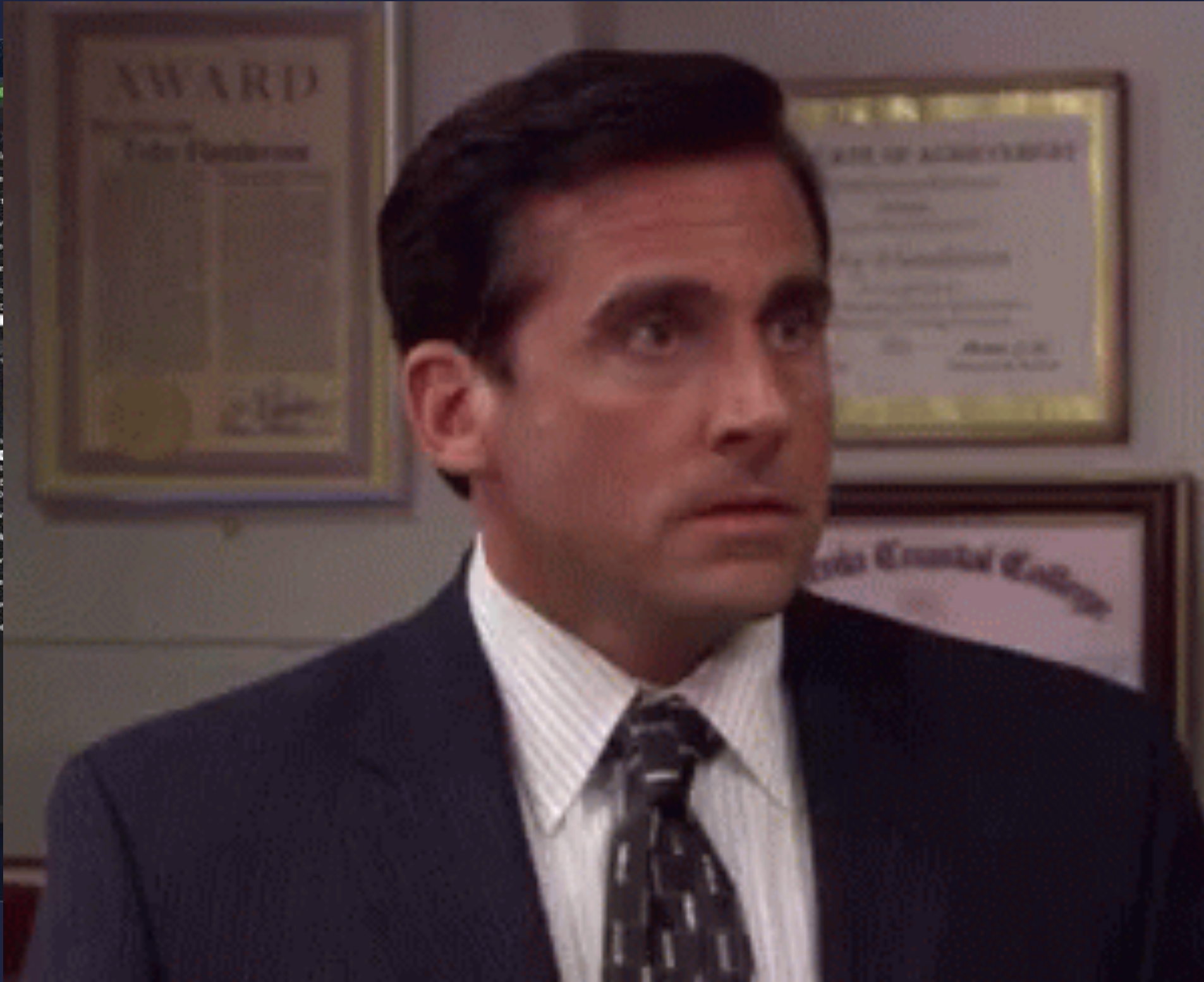
```
42   body: new Center(  
43     child: new Text(  
44       'Button clicked $_counter times',  
45       style: Theme.of(context).textTheme.display1,  
46     ), // Text  
47   ), // Center  
48   floatingActionButton: new FloatingActionButton(  
49     onPressed: _incrementCounter,  
50     tooltip: 'Increment',  
51     child: new Icon(Icons.add),  
52   ), // FloatingActionButton  
53 ); // Scaffold  
54 }  
55 }  
56
```



XCODE



XCODE



PERGUNTA

- Web?
- Android?
- iOS?
- React Native?

Get started

- 1: Install
- 2: Configure editor
- 3: Test drive
- 4: Write your first app
- 5: Learn more

Build UIs

- Tour the framework
- Widget catalog
- Cookbook
- Sample catalog
- Codelabs
- Build layouts - Tutorial
- Add interactivity - Tutorial
- Flutter for Web devs
- Flutter for Android devs
- Flutter for iOS devs
- Flutter for React Native devs
- Gestures
- Animations
- Box constraints
- Assets and images
- Internationalization
- Accessibility

Flutter Documentation

Note: Please take this 5-min survey to help improve Flutter!

Get Started

Set up your environment and start building.

[VISIT](#)

Widgets Catalog

Dip into the rich set of Flutter available in the SDK.

[VISIT](#)

New to Flutter?

Once you've gone through [Get Started](#), including [Write Your First Flutter App](#), here

- [Flutter for Android Developers](#)
Review these tips if you have Android experience.
- [HTML/CSS patterns](#)
Review these HTML -> Flutter analogs if you have web experience.

LIVE CODE



LINKS

- <https://flutter.io/>
- [What's Revolutionary about Flutter](#)
- [Reflectly — From React Native to Flutter](#)
- [Codelabs](#)

OBRIGADO!

PERGUNTAS?

Slides and Code

<https://github.com/reeichert/>

Contato

joao.reichert@brave.com