DESENVOLVIMENTO MOBILE MULTIPLATAFORMA

FLUTTER & FIREBASE

JOÃO REICHERT

- iOS Developer @ Brave.ag
- Eng. Controle e Automação @ UFPel
- 5 anos trabalhando com iOS

https://github.com/reeichert/
joao.reichert@brave.ag

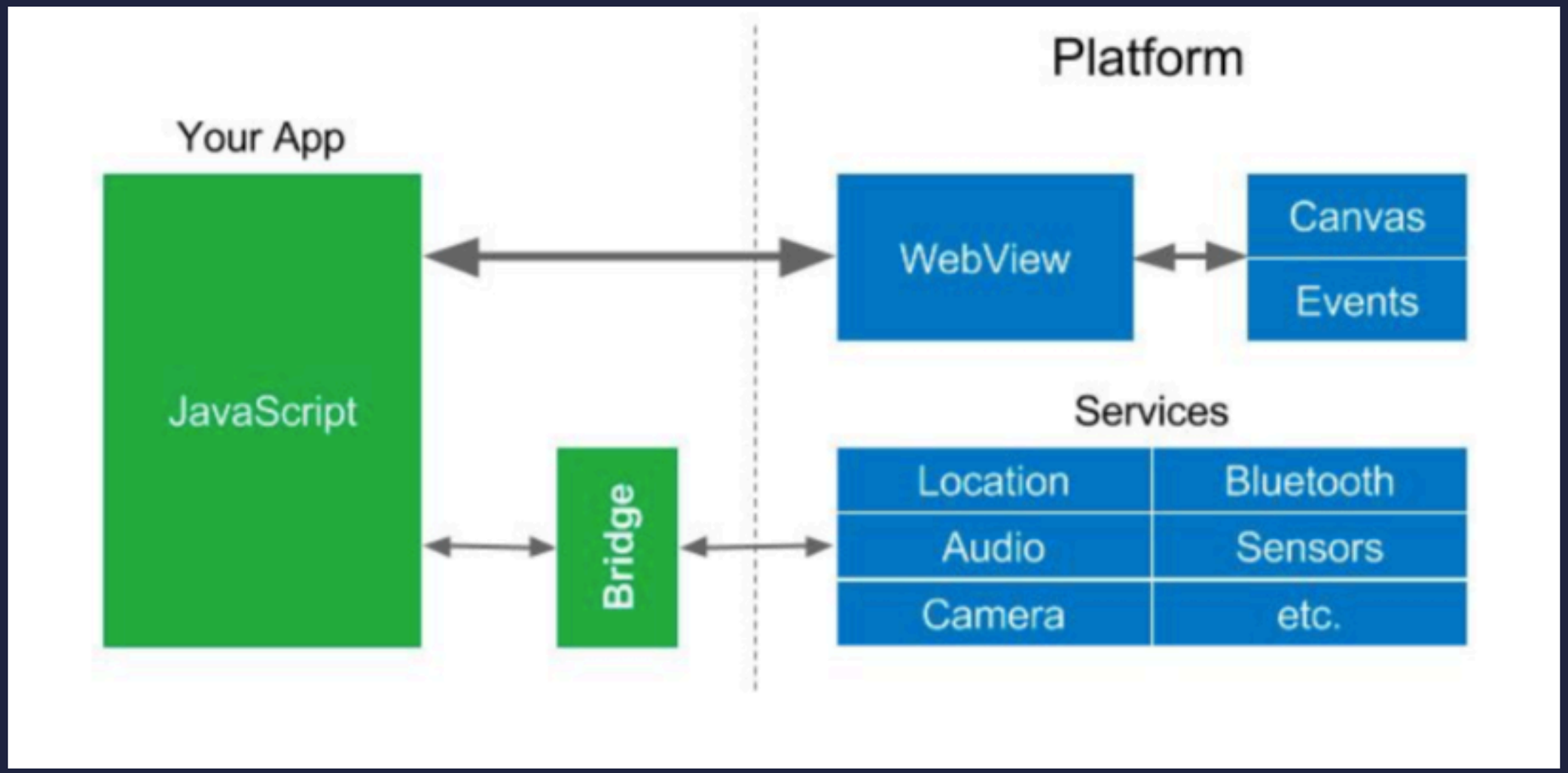
CONTEÚDO

- Recap 
- O que é Flutter 
- O que é Firebase 
- Code

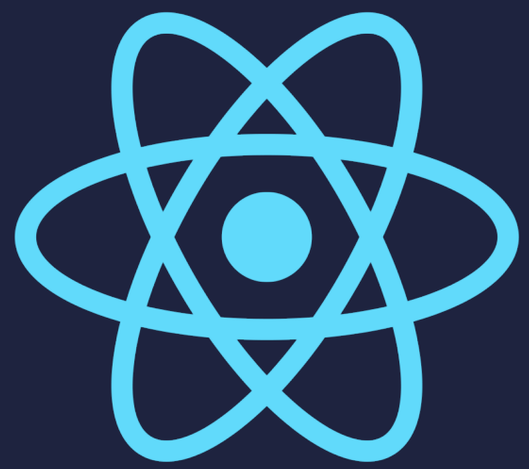
RECAP

WRAPPER WEBVIEW

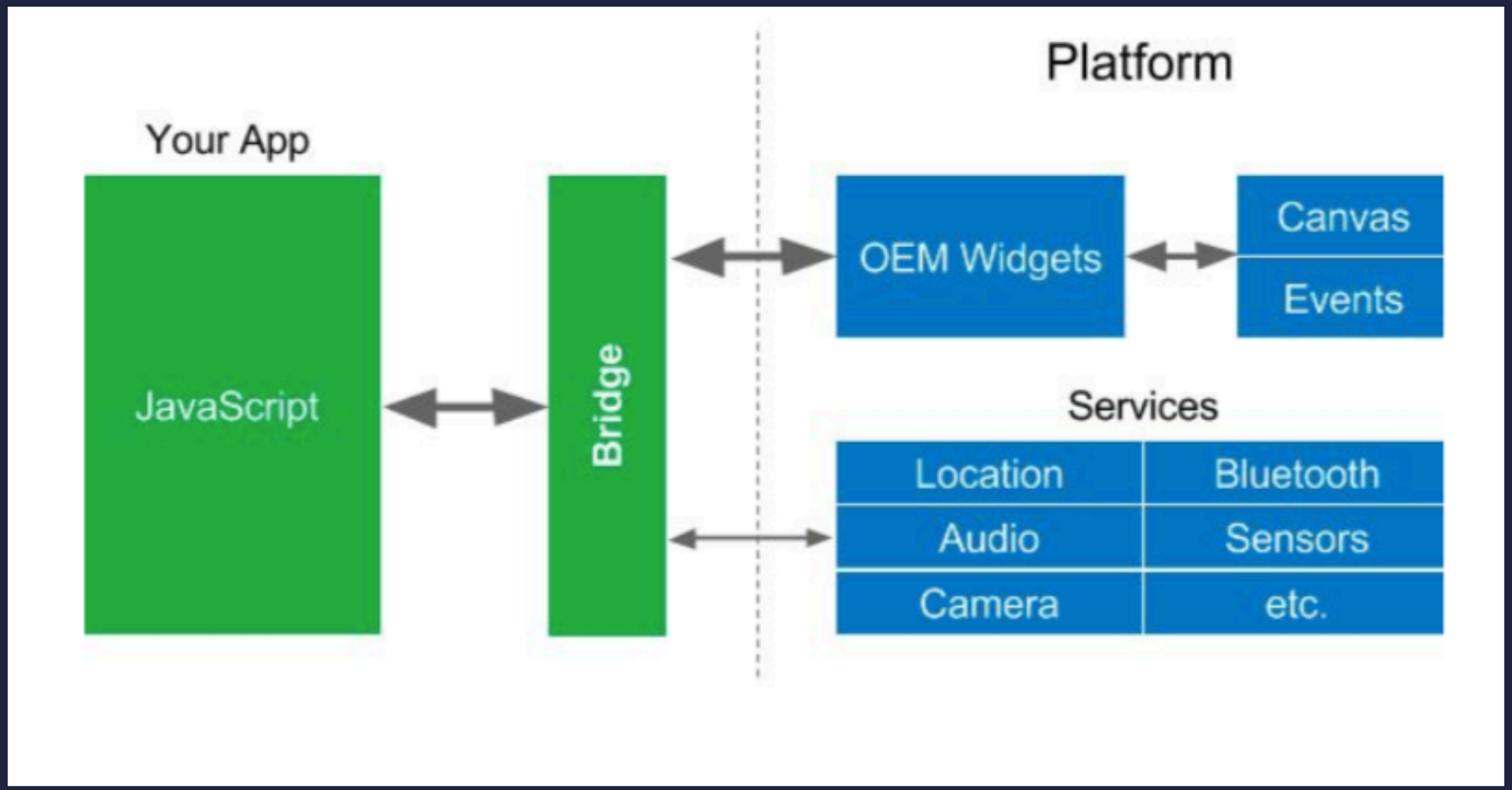
WRAPPER WEBVIEW



TRANSPILER



TRANSPILER


Platform
Your App
JavaScript
Bridge
OEM Widgets
Canvas
Events
Services
Location
Bluetooth
Audio
Sensors
Camera
etc.

FLUTTER

O QUE É FLUTTER? 🤔

O QUE É FLUTTER? 🤔

- Um SDK para desenvolver apps multiplataforma*

O QUE É FLUTTER? 🤔

- Um SDK para desenvolver apps multiplataforma*
- Foco em apps de alta performance, modernos e bonitos

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

O QUE É FLUTTER? 🤔

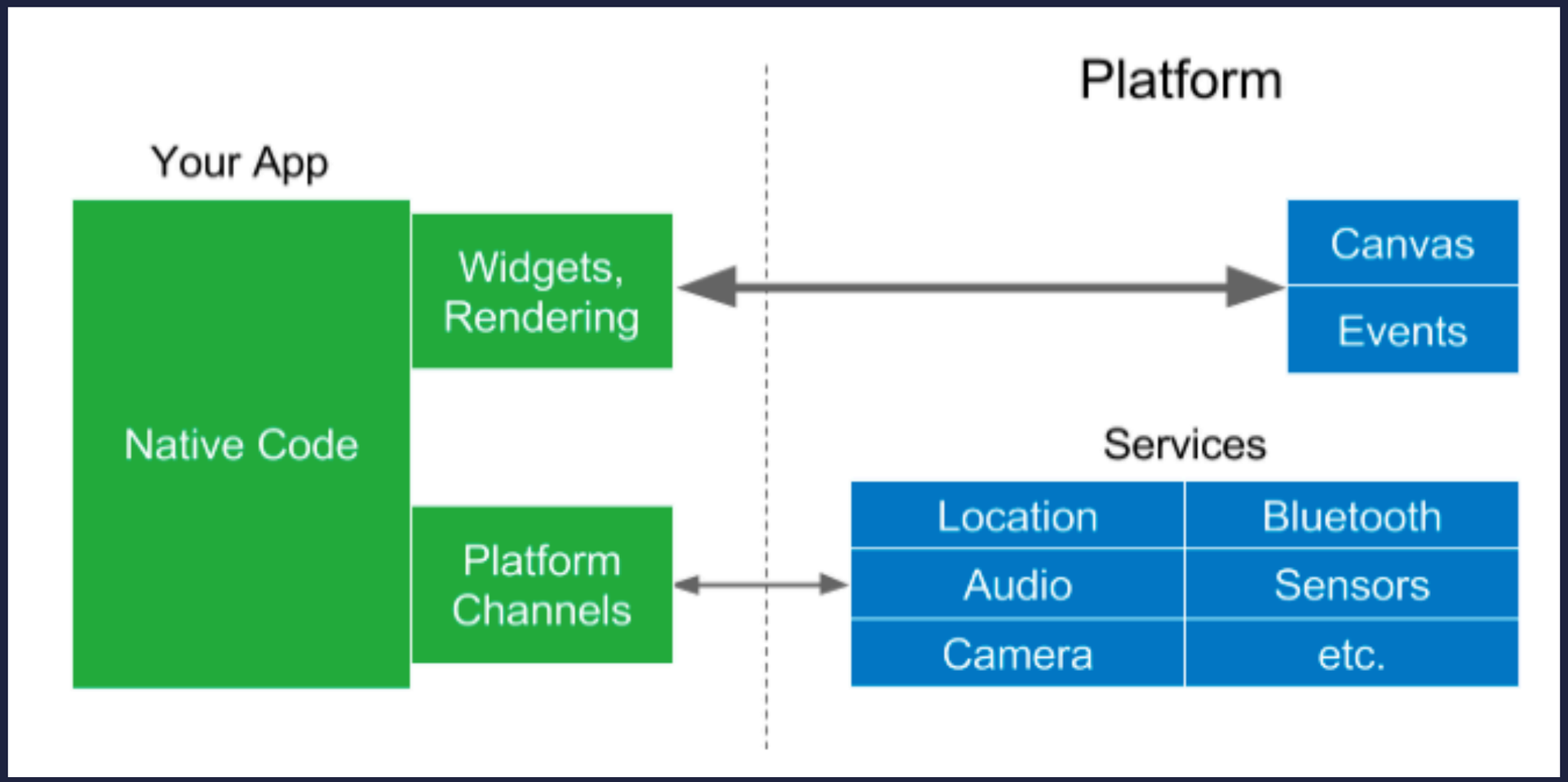
- Um SDK para desenvolver apps multiplataforma*
- Foco em apps de alta performance, modernos e bonitos
- Desenvolvido pelo Google, é gratuito e open-source

O QUE É FLUTTER? 🤔

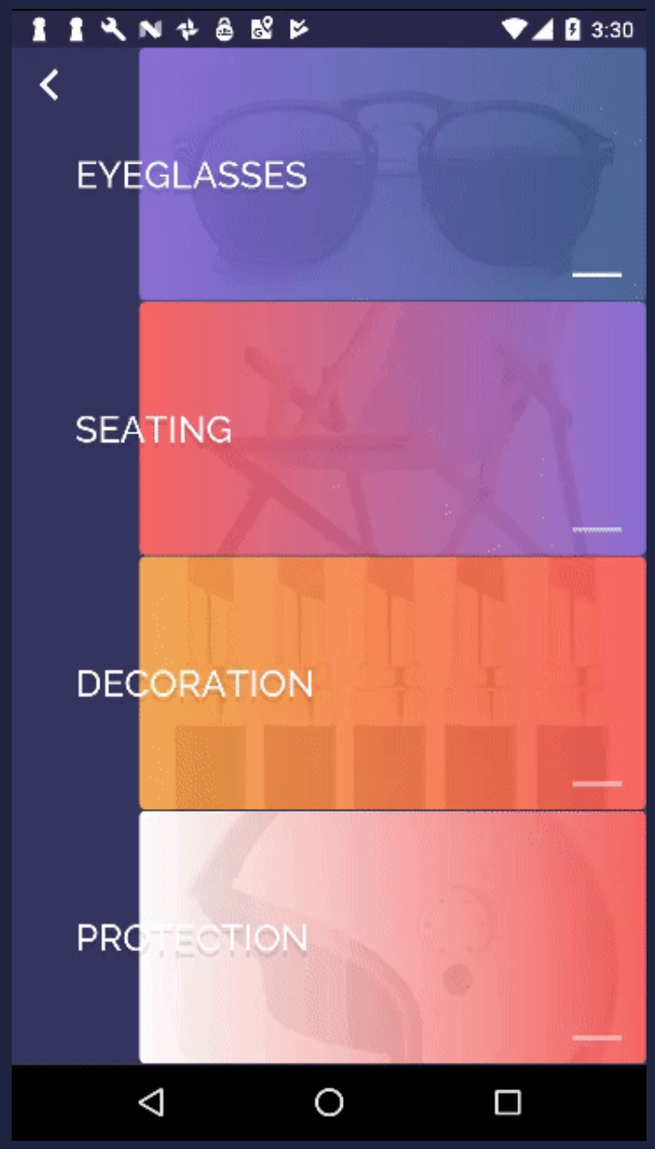
- Um SDK para desenvolver apps multiplataforma*
- Foco em apps de alta performance, modernos e bonitos
- Desenvolvido pelo Google, é gratuito e open-source

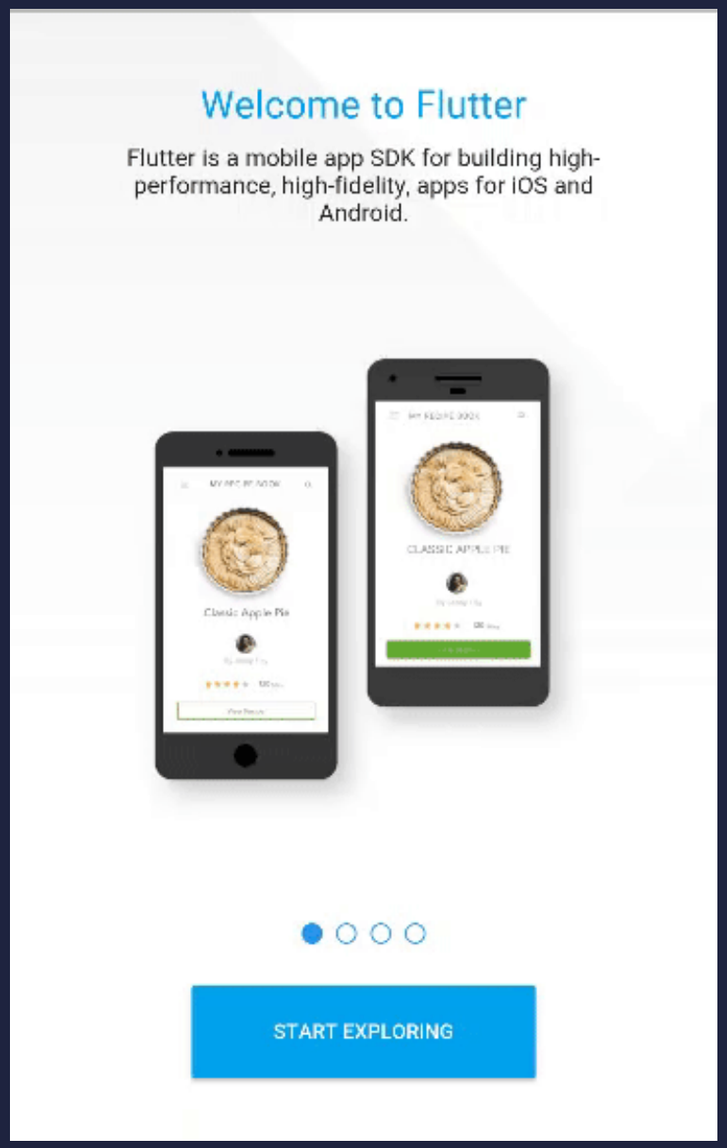
★ Multiplataforma (iOS, Android e Fuchsia)

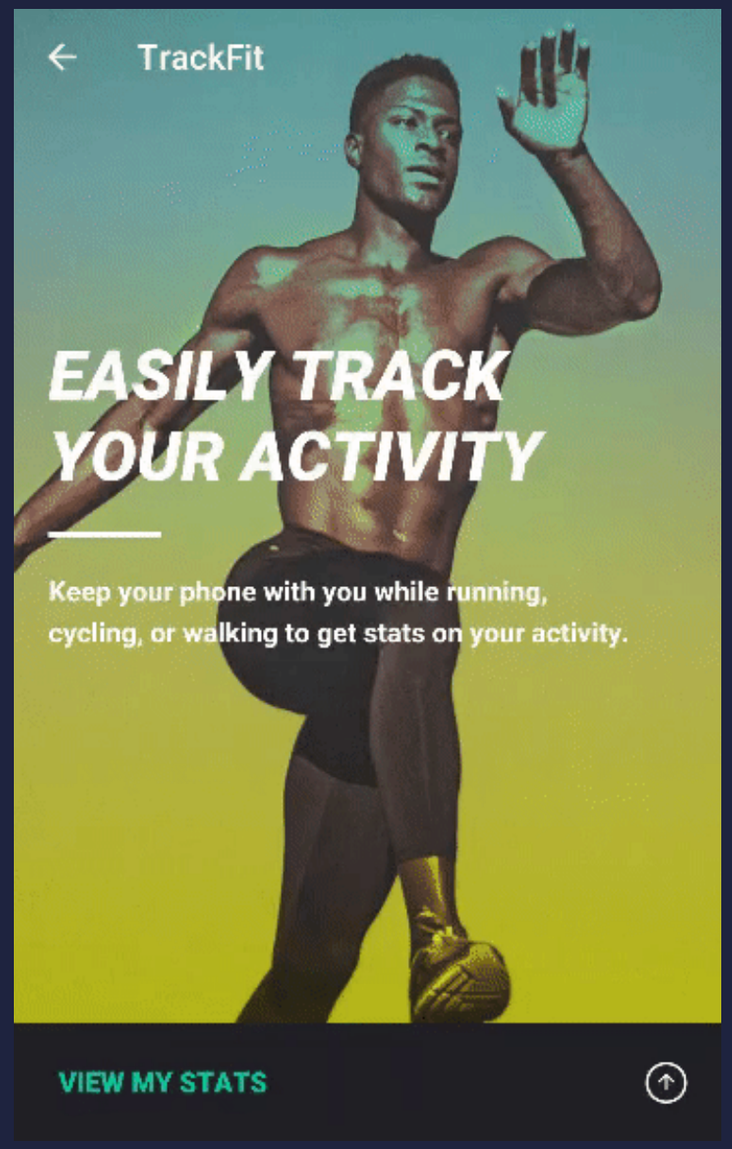
FLUTTER


Platform
Your App
Widgets,
Rendering
Native Code
Platform
Channels
Canvas
Events
Services
Location
Bluetooth
Audio
Sensors
Camera
etc.

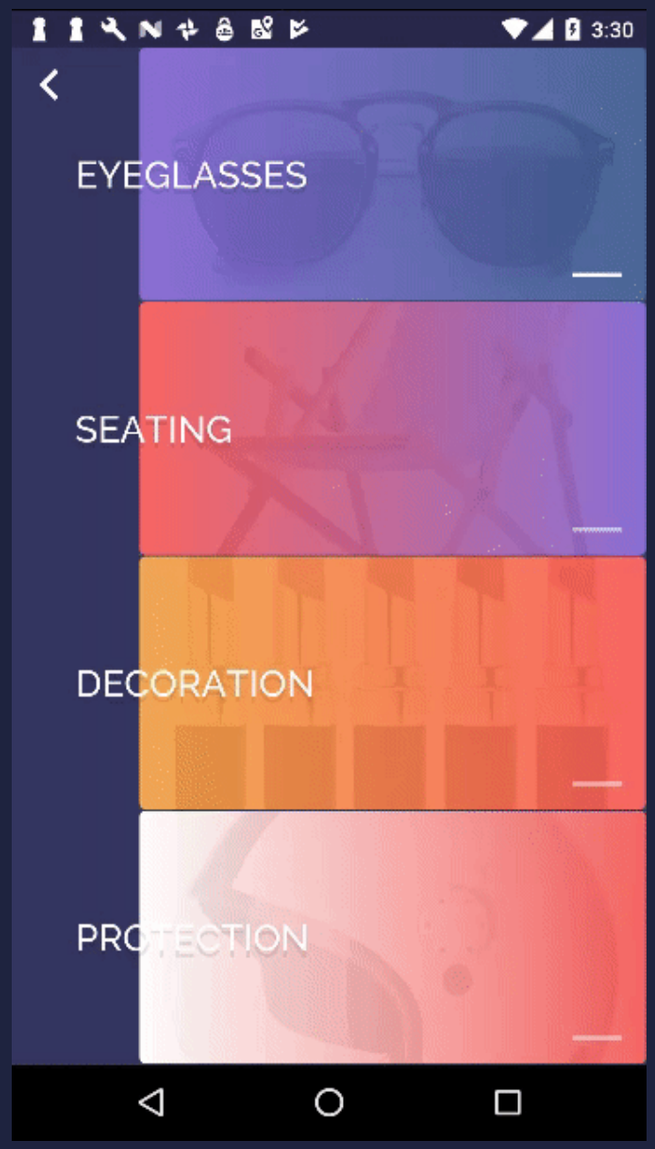
EXEMPLOS

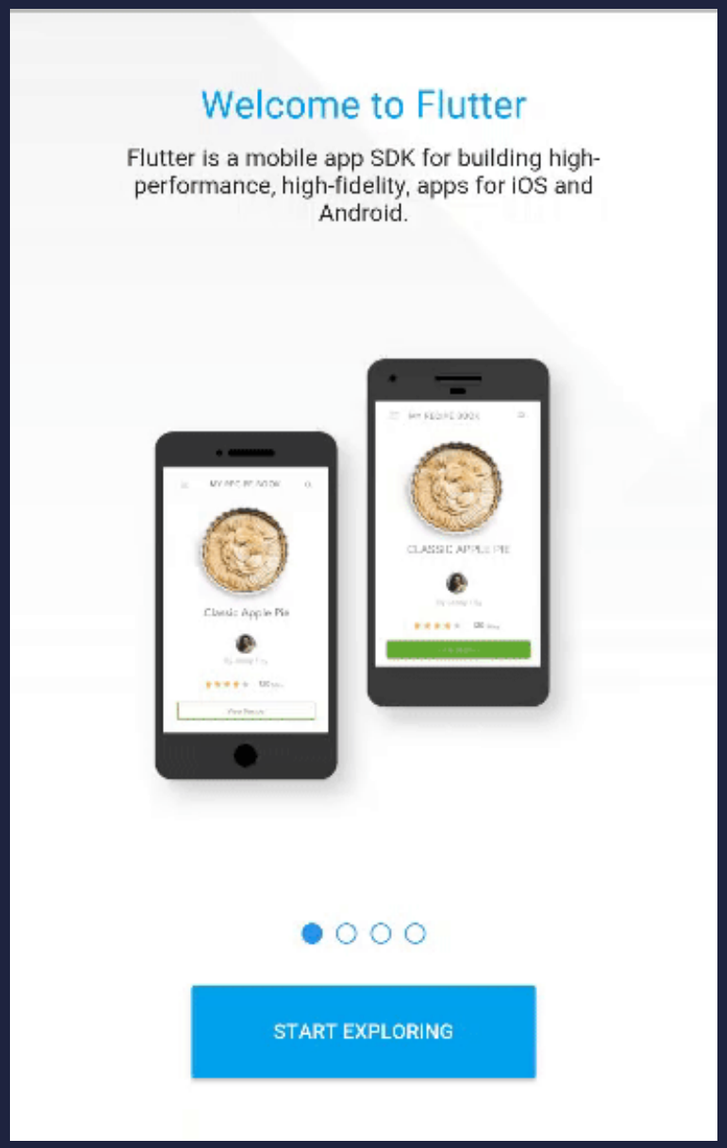

3:30
EYEGLASSES
SEATING
DECORATION
PROTECTION

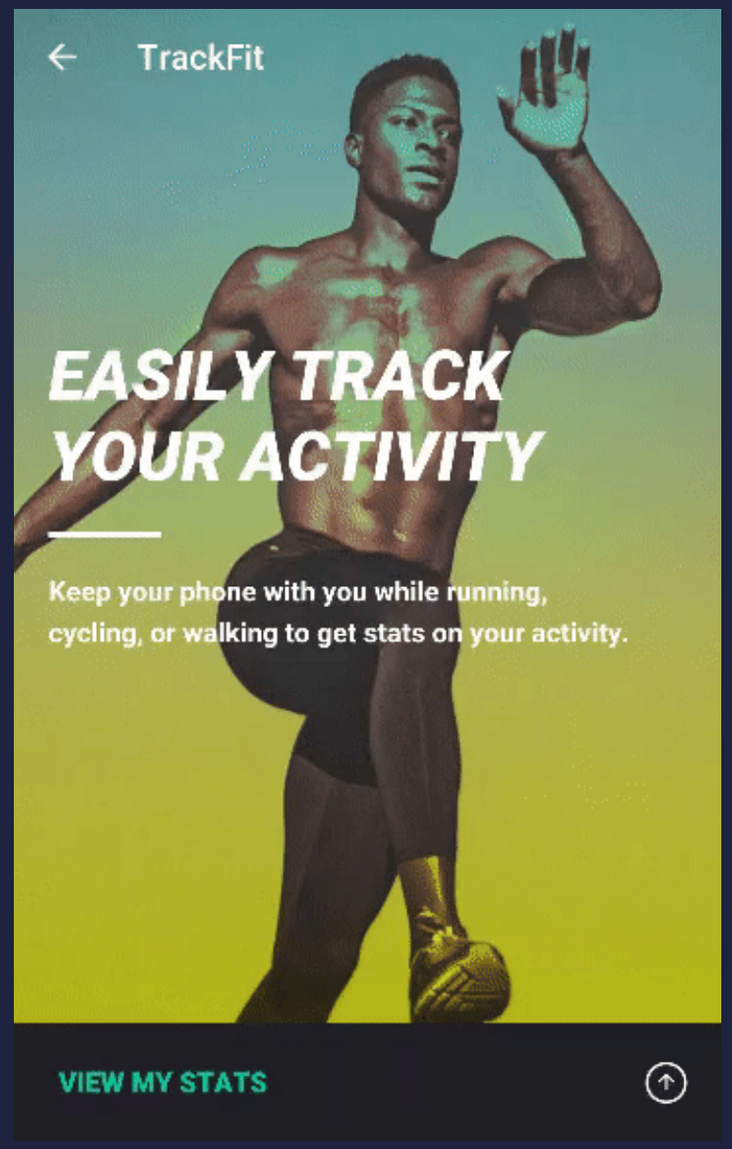

Welcome to Flutter
Flutter is a mobile app SDK for building high-performance, high-fidelity, apps for iOS and Android.
START EXPLORING


TrackFit
EASILY TRACK YOUR ACTIVITY
Keep your phone with you while running, cycling, or walking to get stats on your activity.
VIEW MY STATS

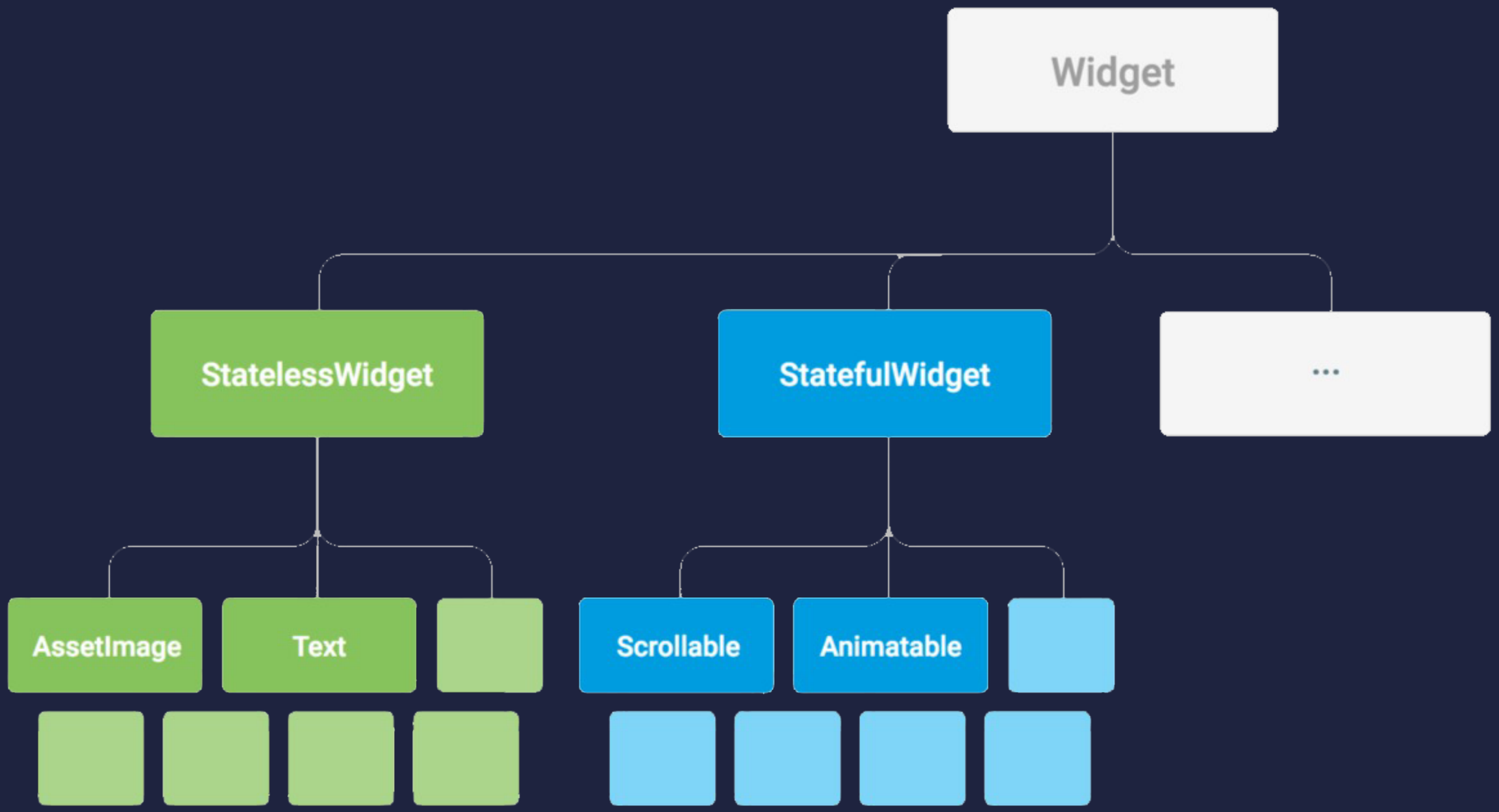
EXEMPLOS


3:30
EYEGLASSES
SEATING
DECORATION
PROTECTION


Welcome to Flutter
Flutter is a mobile app SDK for building high-performance, high-fidelity, apps for iOS and Android.
START EXPLORING


TrackFit
EASILY TRACK YOUR ACTIVITY
Keep your phone with you while running, cycling, or walking to get stats on your activity.
VIEW MY STATS

TUDO É WIDGET



WIDGETS

StatelessWidget

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

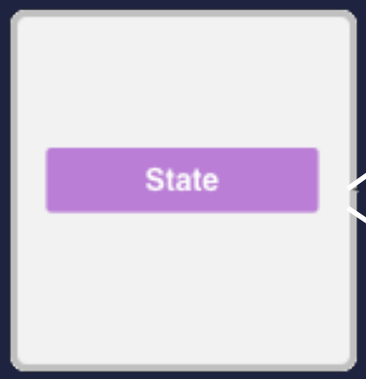
```dart
class Counter extends StatefulWidget {
  @override
  _CounterState createState() => _CounterState();
}

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!')
        )
      ]
    );
  }
}
```

WIDGETS

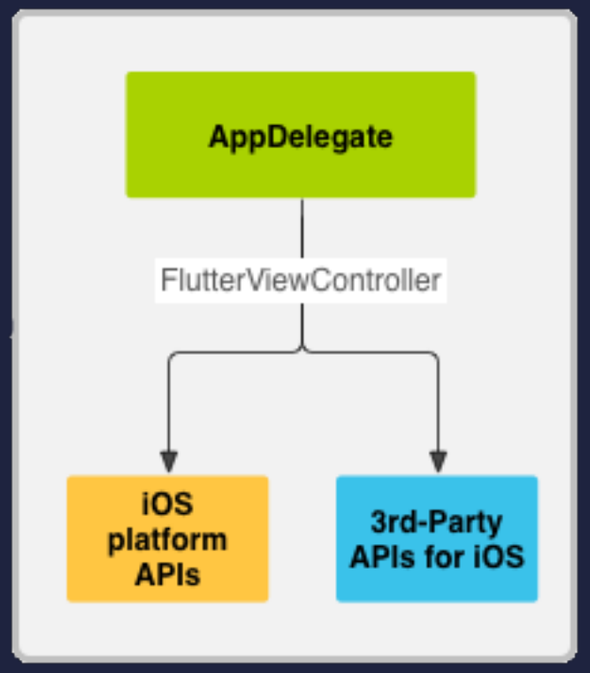
StatefulWidget

PLATFORM CHANNELS

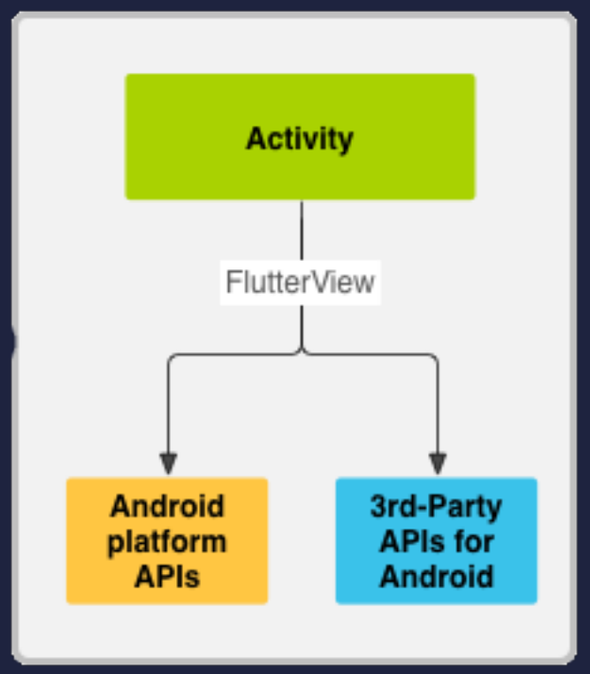
Flutter App (client)



MethodChannel



iOS Host



Android Host

PLATFORM CHANNELS

Kotlin

```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

```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

```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

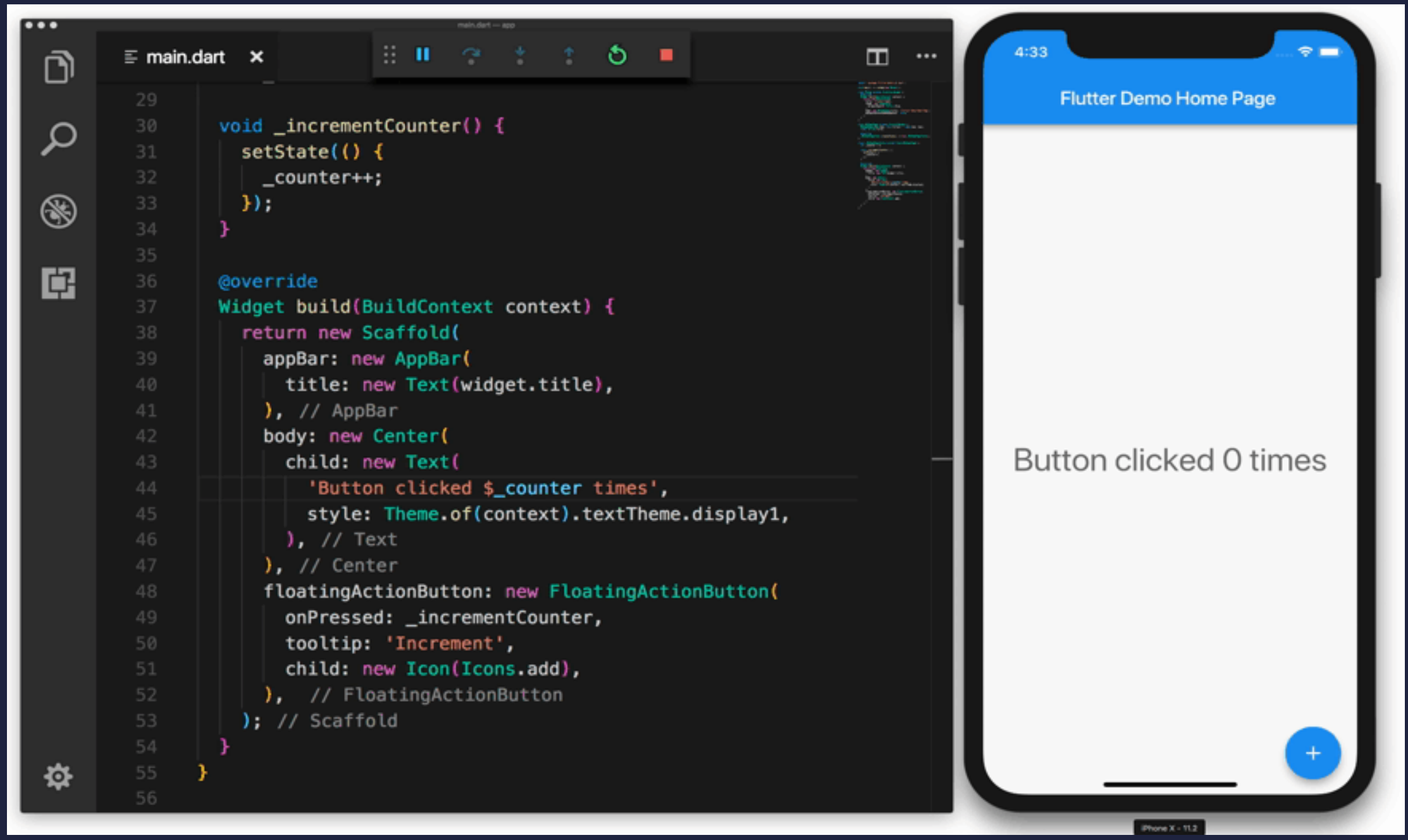
```
batteryChannel.setMethodCallHandler({
  (call: FlutterMethodCall, result: FlutterResult) -> Void in
  if ("getBatteryLevel" == call.method) {
    self.receiveBatteryLevel(result: result);
  } else {
    result(FlutterMethodNotImplemented);
  }
});
```

PLATFORM CHANNELS

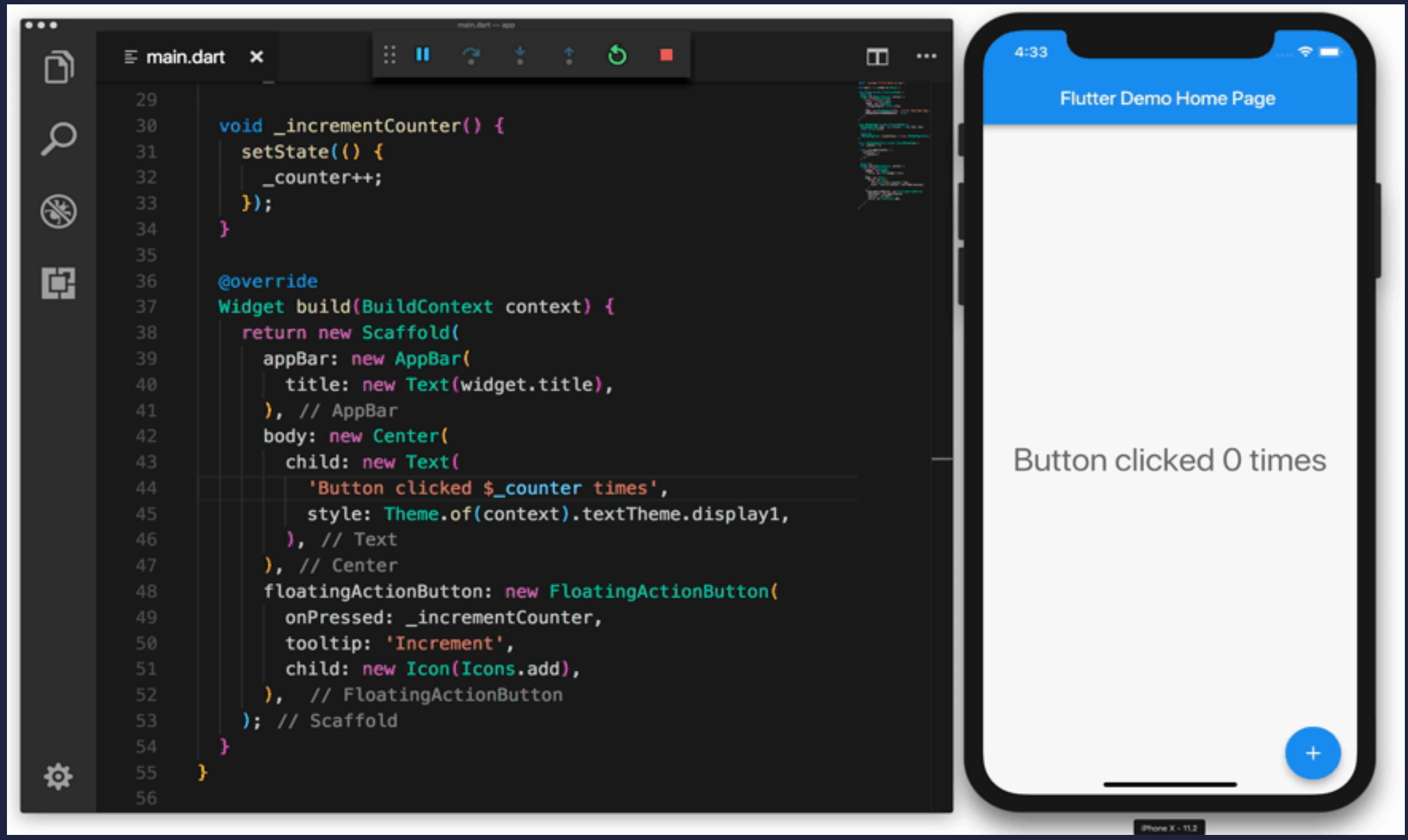
Flutter

```dart
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;
  });
}
```

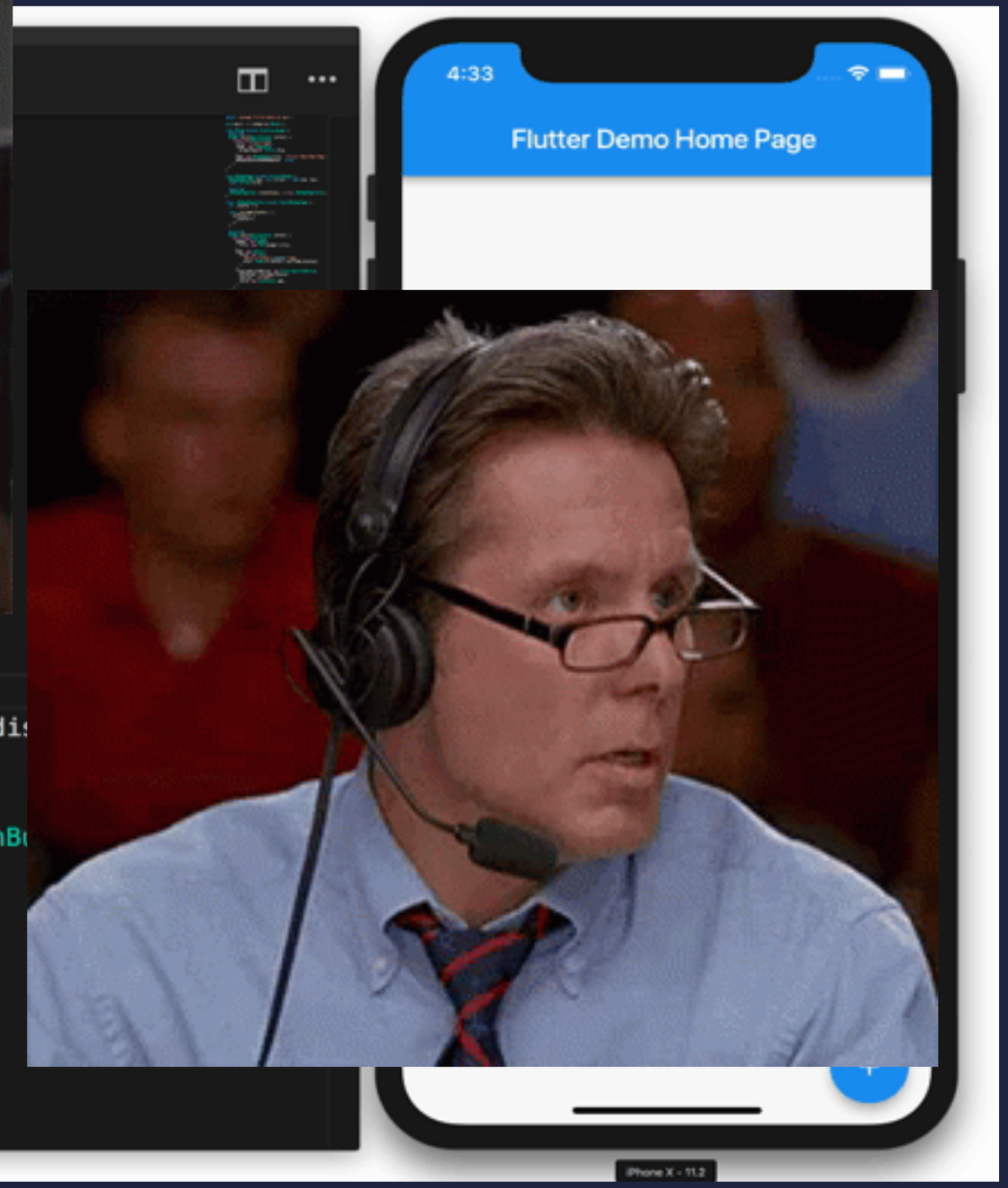
HOT RELOAD


main.dart
void _incrementCounter() {
setState(() {
_counter++;
});
}
@override
Widget build(BuildContext context) {
return new Scaffold(
appBar: new AppBar(
title: new Text(widget.title),
), // AppBar
body: new Center(
child: new Text(
'Button clicked $_counter times',
style: Theme.of(context).textTheme.display1,
), // Text
), // Center
floatingActionButton: new FloatingActionButton(
onPressed: _incrementCounter,
tooltip: 'Increment',
child: new Icon(Icons.add),
), // FloatingActionButton
); // Scaffold
}
}
4:33
Flutter Demo Home Page
Button clicked 0 times

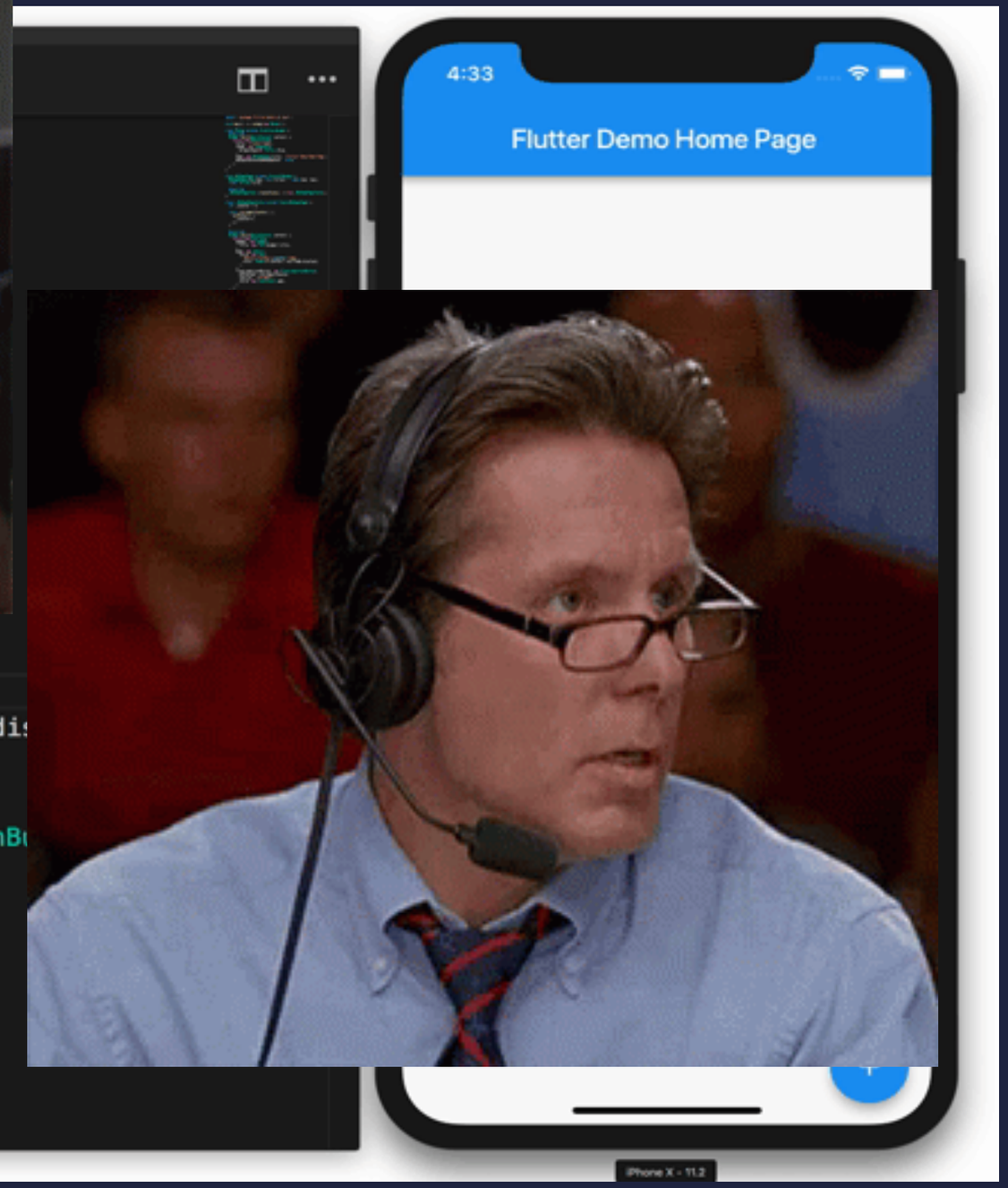
HOT RELOAD


main.dart
void _incrementCounter() {
setState(() {
_counter++;
});
}
@override
Widget build(BuildContext context) {
return new Scaffold(
appBar: new AppBar(
title: new Text(widget.title),
), // AppBar
body: new Center(
child: new Text(
'Button clicked $_counter times',
style: Theme.of(context).textTheme.display1,
), // Text
), // Center
floatingActionButton: new FloatingActionButton(
onPressed: _incrementCounter,
tooltip: 'Increment',
child: new Icon(Icons.add),
), // FloatingActionButton
); // Scaffold
}
}
4:33
Flutter Demo Home Page
Button clicked 0 times

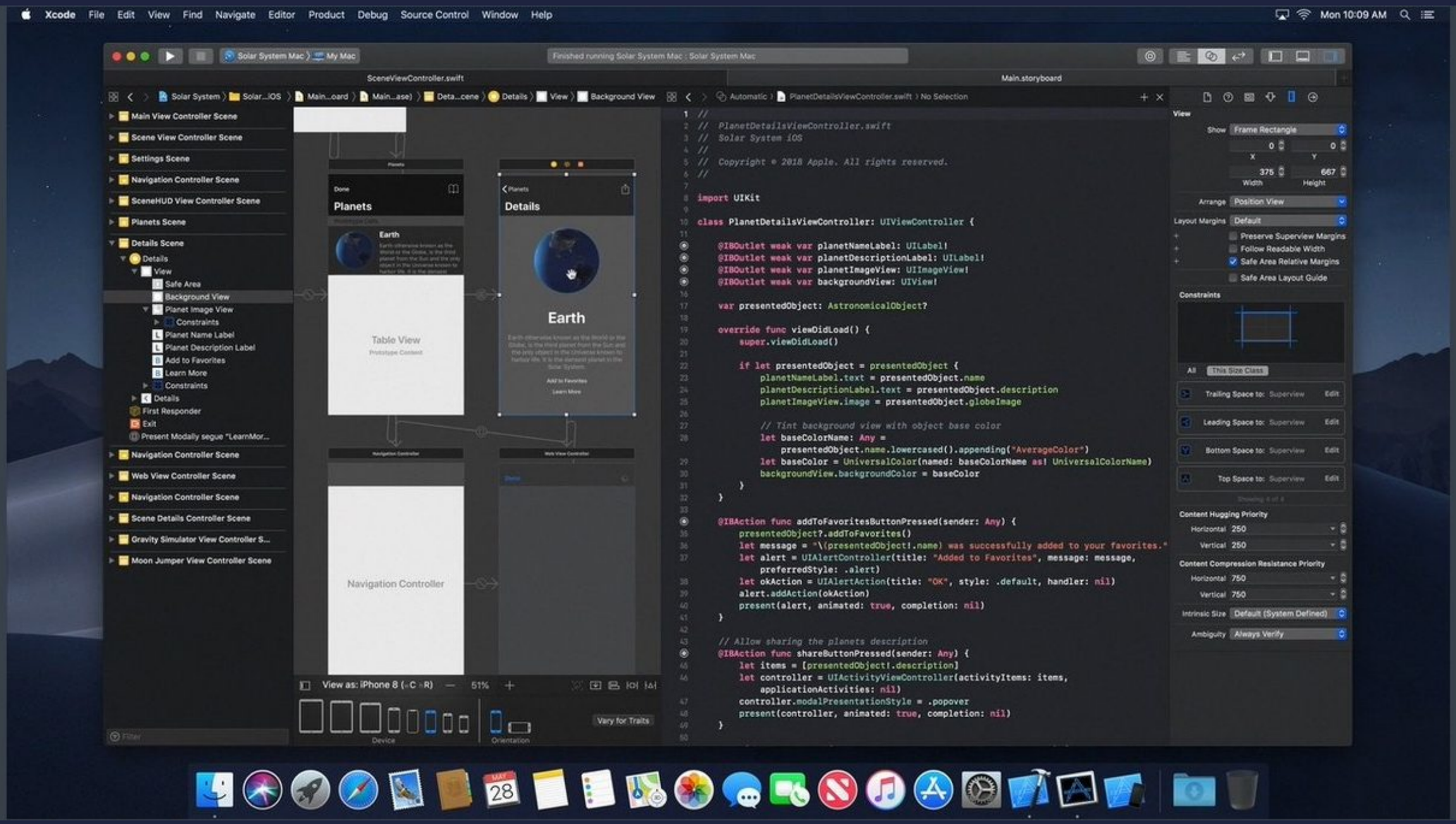
HOT RELOAD


4:33
Flutter Demo Home Page
body: new Center(
child: new Text(
'Button clicked $_counter times',
style: Theme.of(context).textTheme.dis
), // Text
), // Center
floatingActionButton: new FloatingActionBu
onPressed: _incrementCounter,
tooltip: 'Increment',
child: new Icon(Icons.add),
), // FloatingActionButton
); // Scaffold
}
}
iPhone X - 11.2

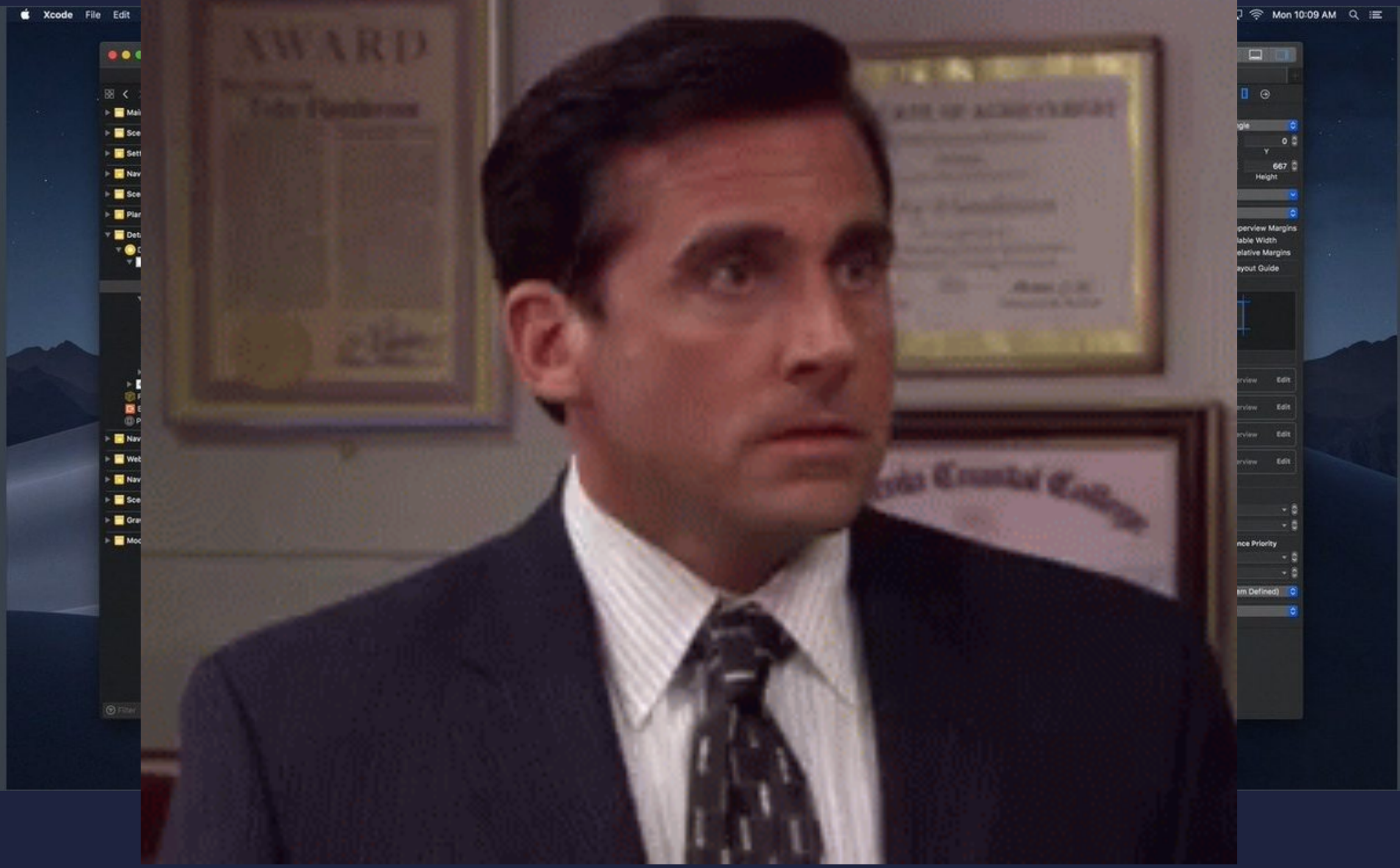
HOT RELOAD


4:33
Flutter Demo Home Page
body: new Center(
child: new Text(
'Button clicked $_counter times',
style: Theme.of(context).textTheme.dis
), // Text
), // Center
floatingActionButton: new FloatingActionBu
onPressed: _incrementCounter,
tooltip: 'Increment',
child: new Icon(Icons.add),
), // FloatingActionButton
); // Scaffold
}
}
iPhone X - 11.2

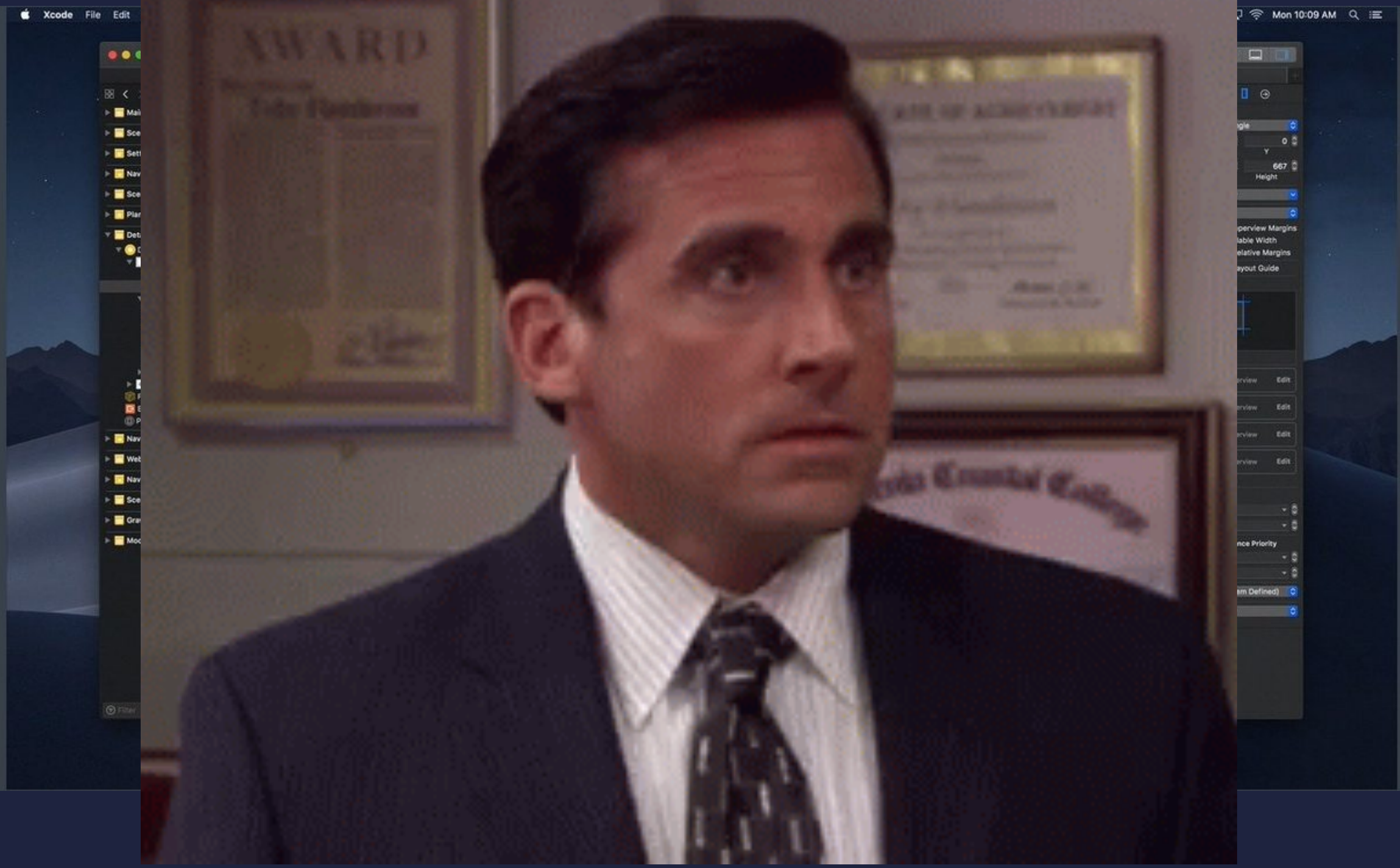
XCODE


Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Mon 10:09 AM
Solar System Mac 〉 My Mac
Finished running Solar System Mac : Solar System Mac
SceneViewController.swift
Main.storyboard
Main View Controller Scene
Scene View Controller Scene
Settings Scene
Navigation Controller Scene
SceneHUD View Controller Scene
Planets Scene
Details Scene
Details
View
Safe Area
Background View
Planet Image View
Constraints
Planet Name Label
Planet Description Label
Add to Favorites
Learn More
Constraints
Details
First Responder
Exit
Present Modally segue "LearnMor...
Navigation Controller Scene
Web View Controller Scene
Navigation Controller Scene
Scene Details Controller Scene
Gravity Simulator View Controller S...
Moon Jumper View Controller Scene
Planets
Earth
Table View
Details
Earth
Navigation Controller
View as: iPhone 8
51%
Device
Orientation
Vary for Traits
PlanetDetailsViewController.swift
No Selection
// PlanetDetailsViewController.swift
// Solar System iOS
// Copyright © 2018 Apple. All rights reserved.
import UIKit
class PlanetDetailsViewController: UIViewController {
@IBOutlet weak var planetNameLabel: UILabel!
@IBOutlet weak var planetDescriptionLabel: UILabel!
@IBOutlet weak var planetImageView: UIImageView!
@IBOutlet weak var backgroundView: UIView!
var presentedObject: AstronomicalObject?
override func viewDidLoad() {
super.viewDidLoad()
if let presentedObject = presentedObject {
planetNameLabel.text = presentedObject.name
planetDescriptionLabel.text = presentedObject.description
planetImageView.image = presentedObject.globeImage
// Tint background view with object base color
let baseColorName: Any =
presentedObject.name.lowercased().appending("AverageColor")
let baseColor = UniversalColor(named: baseColorName as! UniversalColorName)
backgroundView.backgroundColor = baseColor
}
}
@IBAction func addToFavoritesButtonPressed(sender: Any) {
presentedObject?.addToFavorites()
let message = "\(presentedObject!.name) was successfully added to your favorites."
let alert = UIAlertController(title: "Added to Favorites", message: message,
preferredStyle: .alert)
let okAction = UIAlertAction(title: "OK", style: .default, handler: nil)
alert.addAction(okAction)
present(alert, animated: true, completion: nil)
}
// Allow sharing the planets description
@IBAction func shareButtonPressed(sender: Any) {
let items = [presentedObject!.description]
let controller = UIActivityViewController(activityItems: items,
applicationActivities: nil)
controller.modalPresentationStyle = .popover
present(controller, animated: true, completion: nil)
}
View
Show Frame Rectangle
0 0
X Y
375 667
Width Height
Arrange Position View
Layout Margins Default
Preserve Superview Margins
Follow Readable Width
Safe Area Relative Margins
Safe Area Layout Guide
Constraints
All This Size Class
Trailing Space to: Superview Edit
Leading Space to: Superview Edit
Bottom Space to: Superview Edit
Top Space to: Superview Edit
Content Hugging Priority
Horizontal 250
Vertical 250
Content Compression Resistance Priority
Horizontal 750
Vertical 750
Intrinsic Size Default (System Defined)
Ambiguity Always Verify

XCODE



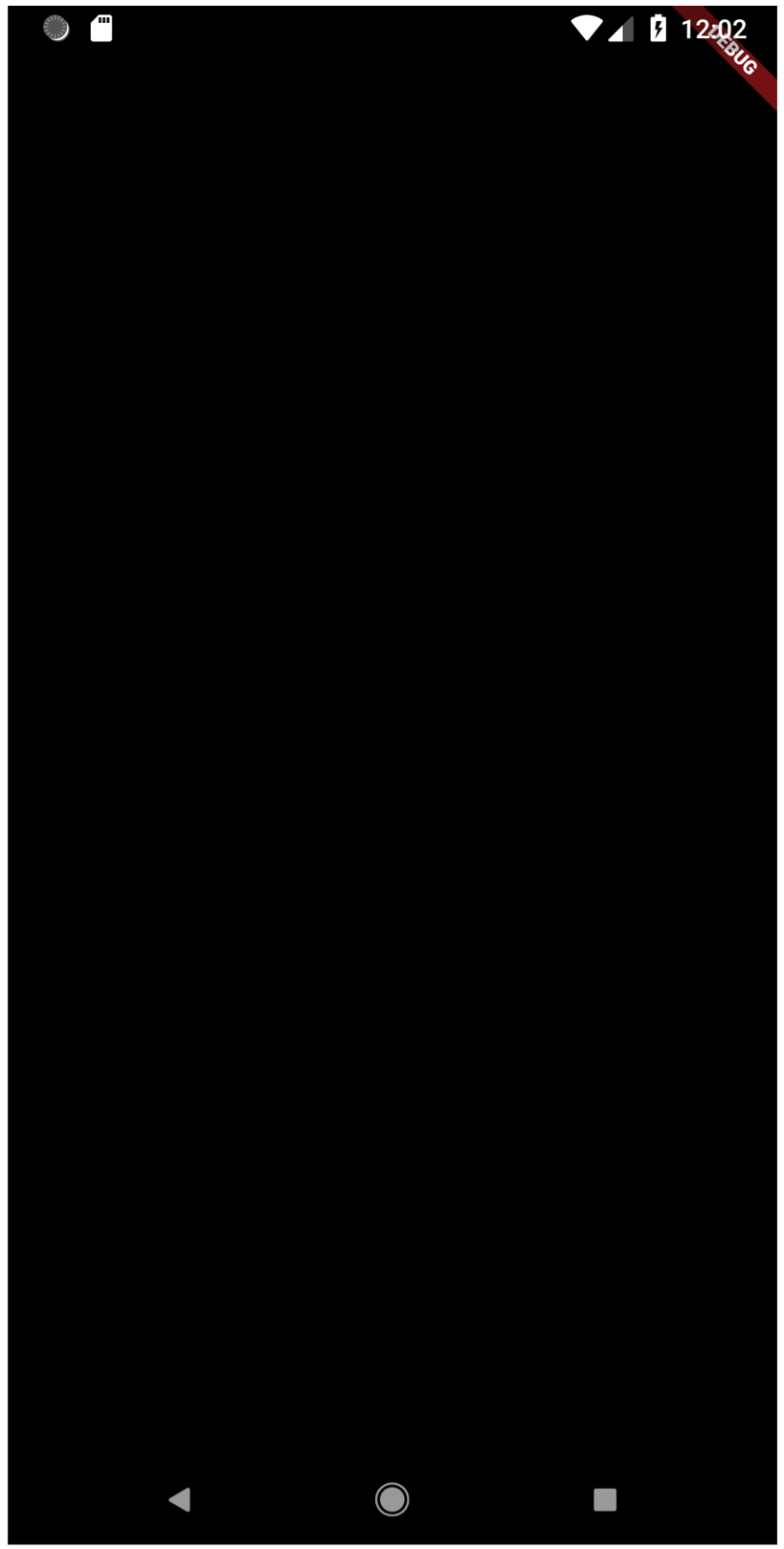
XCODE



ESTRUTURA

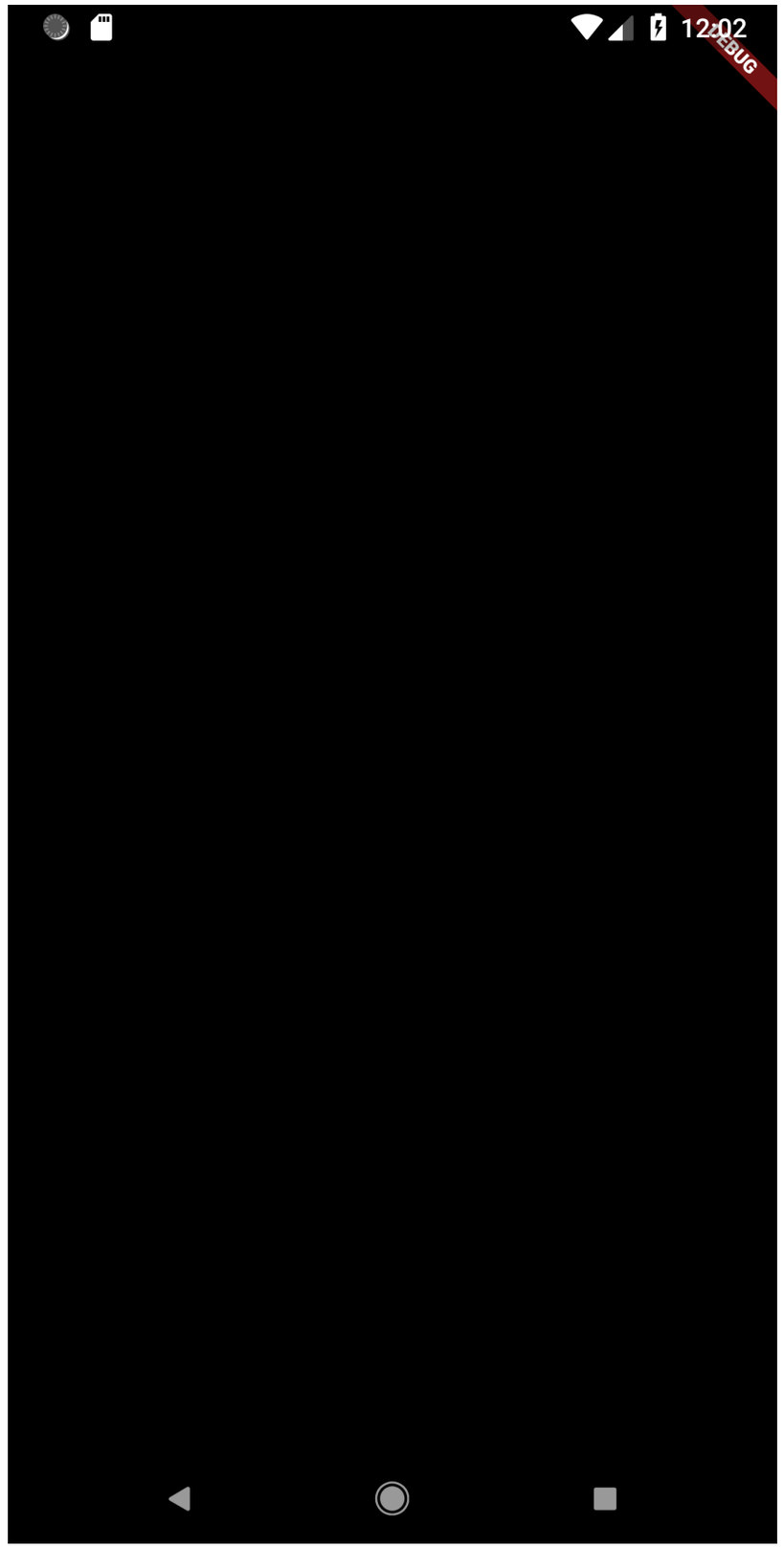
ESTRUTURA

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

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {

 @override
  Widget build(BuildContext context) {
    return MaterialApp();
  }
}
```



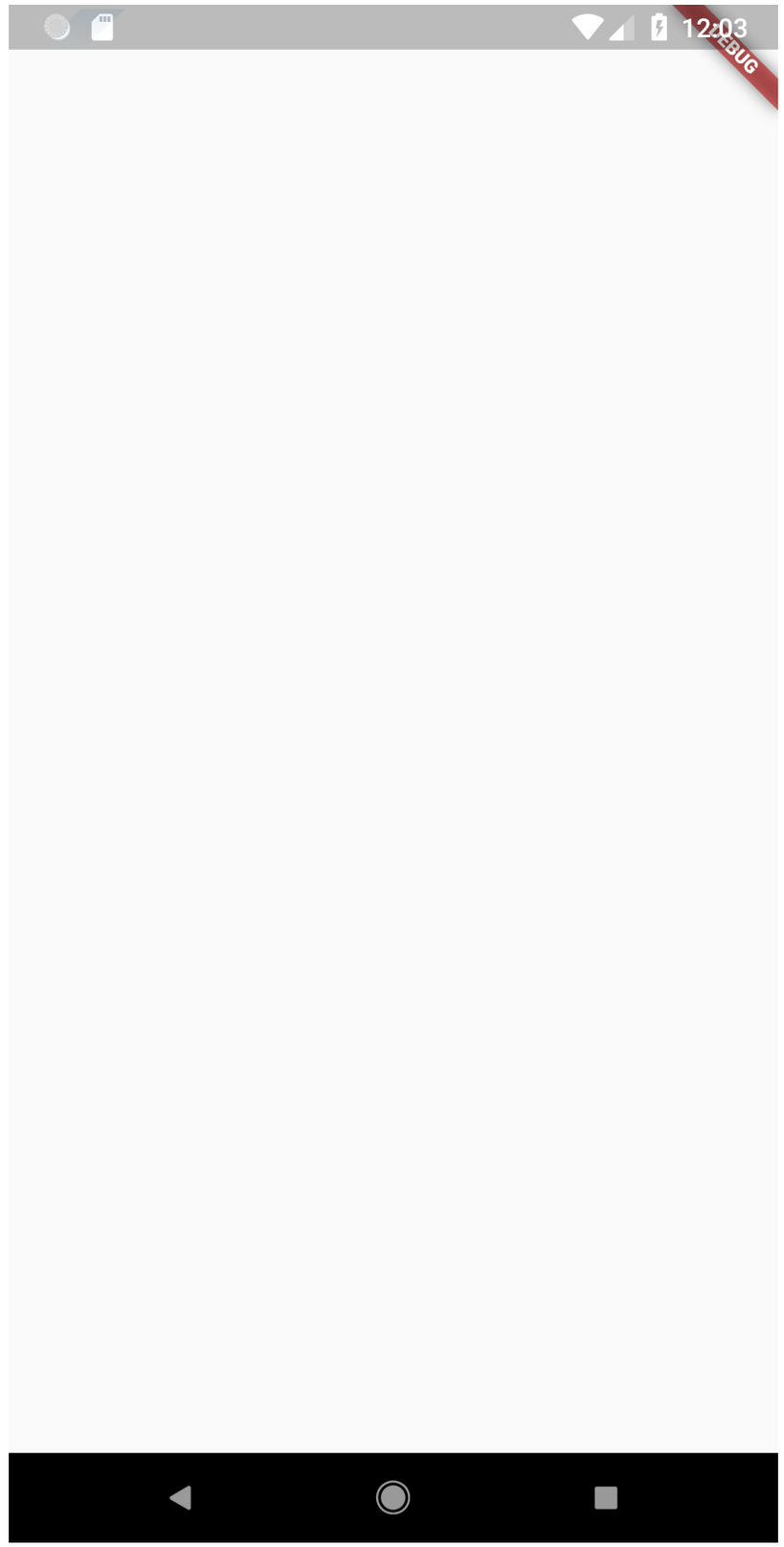
ESTRUTURA

```dart
class MyApp extends StatelessWidget {

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData(
        primarySwatch: Colors.teal,
      ),
      home: HomePage(),
    );
  }
}
```



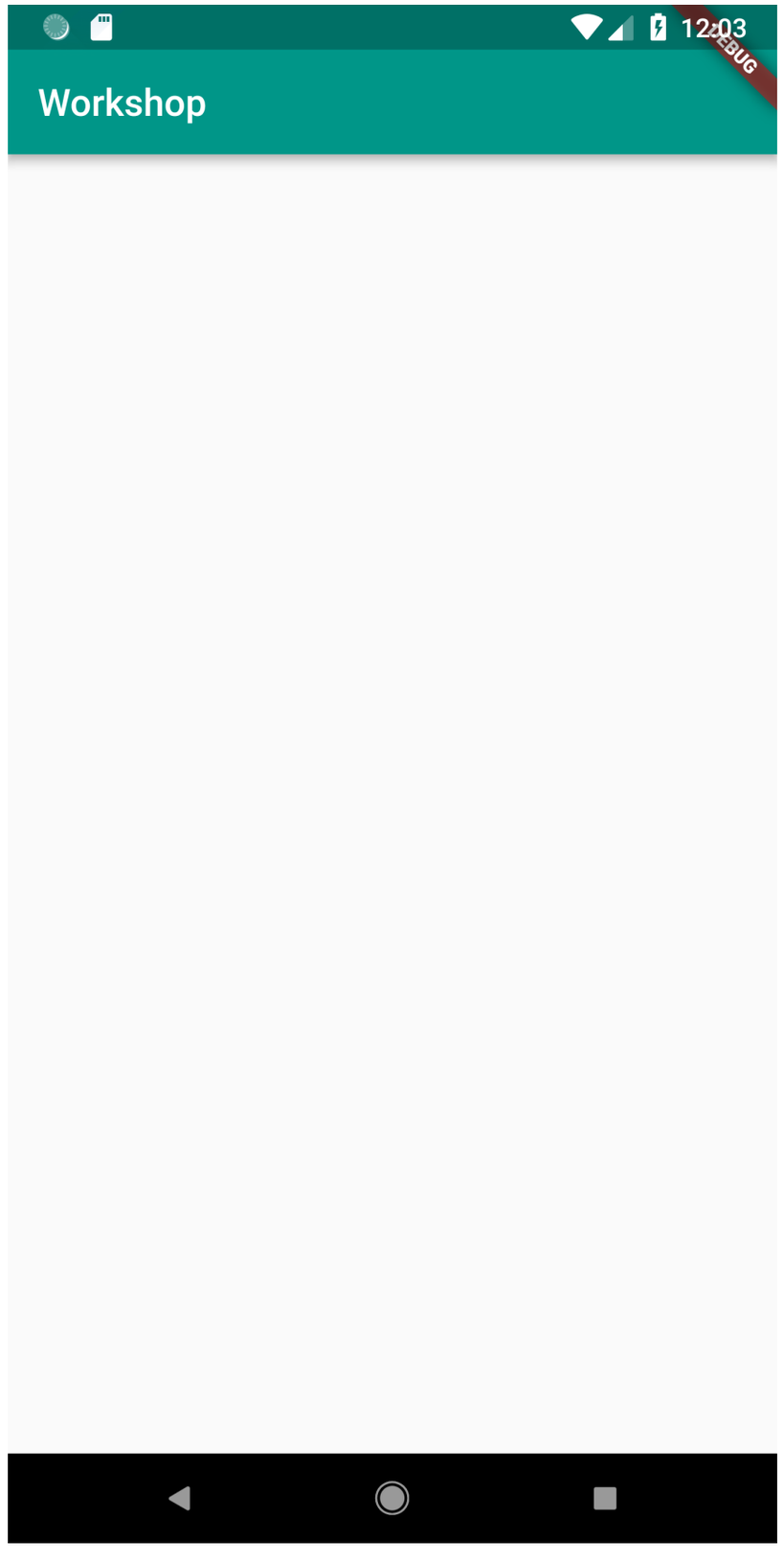
ESTRUTURA

```dart
class HomePage extends StatelessWidget {

  @override
  Widget build(BuildContext context) {
    return Scaffold();
  }
}
```



ESTRUTURA

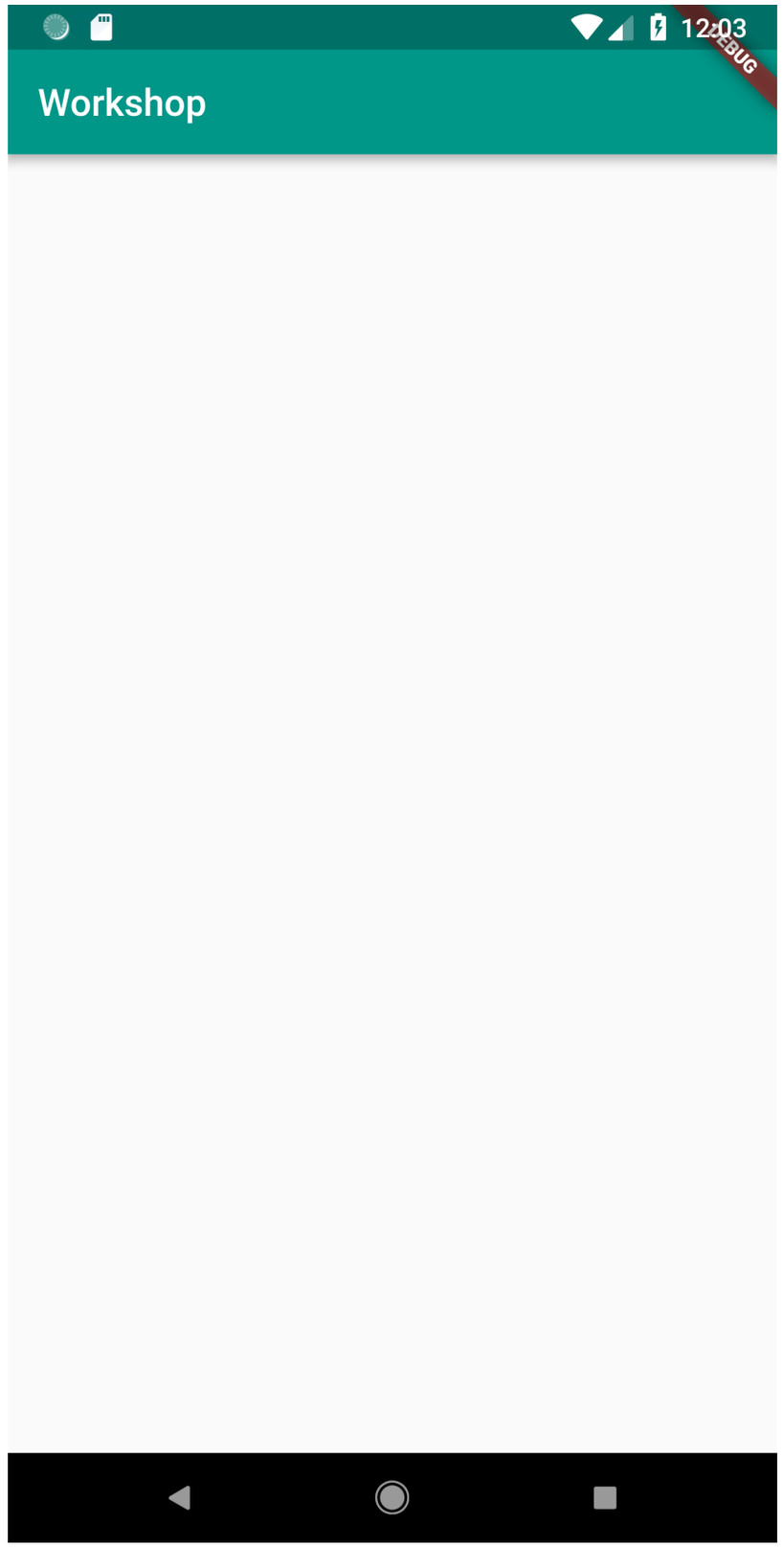
```dart
class HomePage extends StatelessWidget {

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Workshop'),
      )
    );
  }
}
```



ESTRUTURA

```dart
class MyApp extends StatelessWidget {

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData(
        primarySwatch: Colors.teal,
      ),
      home: HomePage(),
    );
  }
}
```

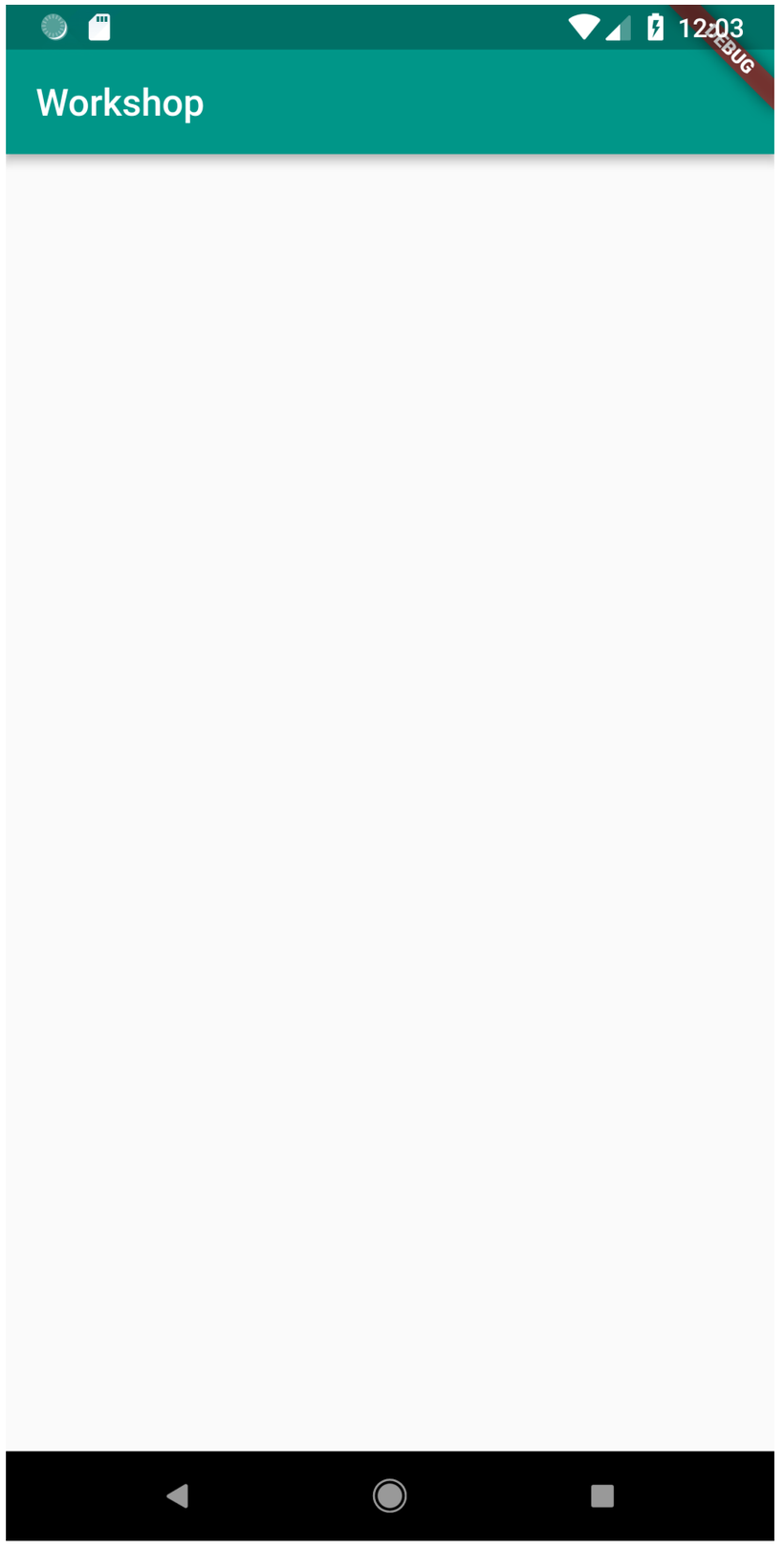
```dart
class HomePage extends StatelessWidget {

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Workshop'),
      )
    );
  }
```



ESTRUTURA

```dart
class HomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Workshop'),
      ), // AppBar

    );
  }
}
```

```
backgroundColor:
body:
bottomNavigationBar:
bottomSheet:
drawer:
endDrawer:
floatingActionButton:
floatingActionButtonAnimator:
floatingActionButtonLocation:
key:
persistentFooterButtons: <Widget…
primary:
```



ESTRUTURA

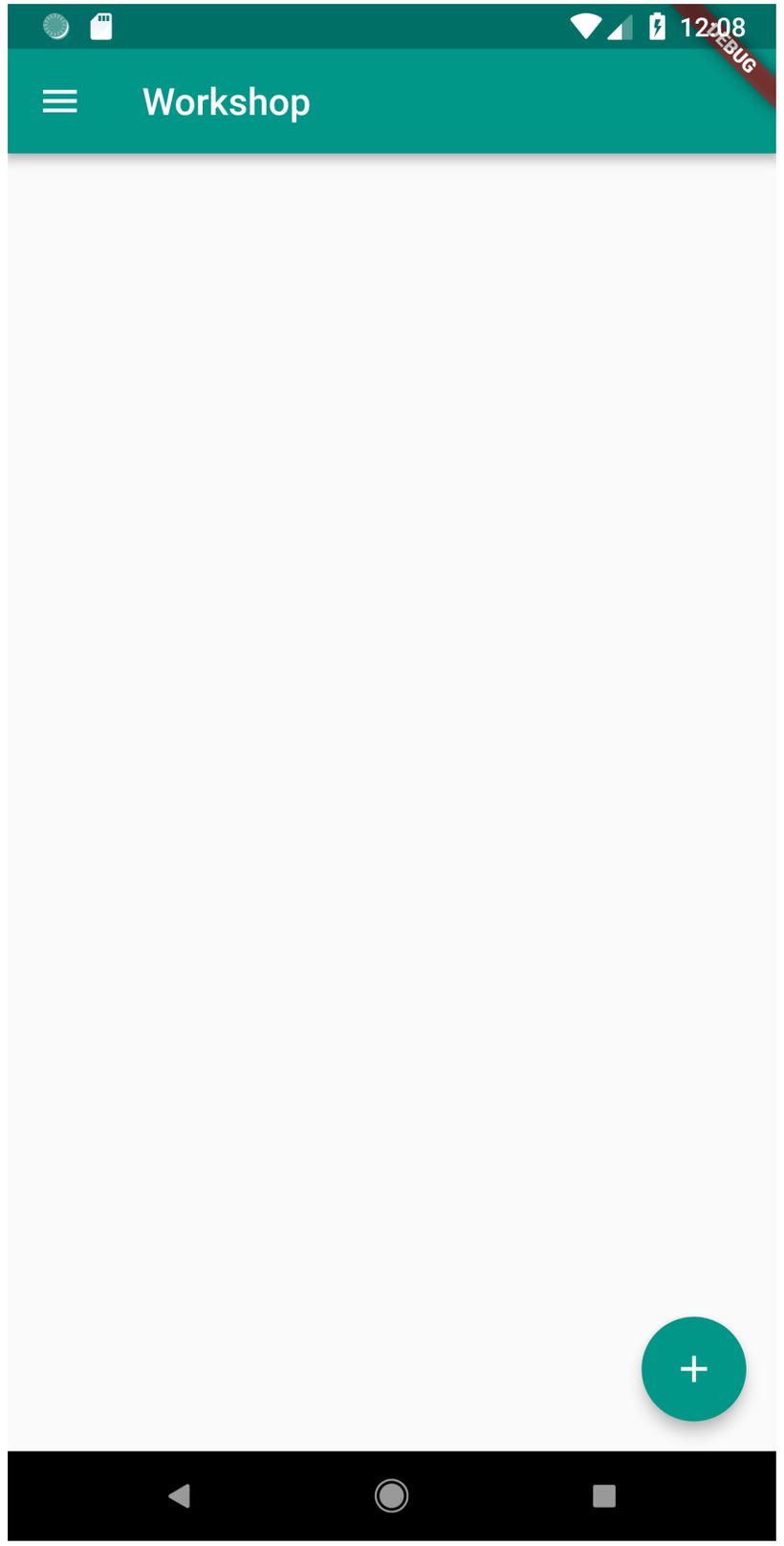
```dart
class HomePage extends StatelessWidget {

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Workshop'),
      ),
      drawer: Drawer()
    );
  }
}
```

ESTRUTURA

```dart
class HomePage extends StatelessWidget {

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Workshop'),
      ),
      drawer: Drawer()
    );
  }
}
```

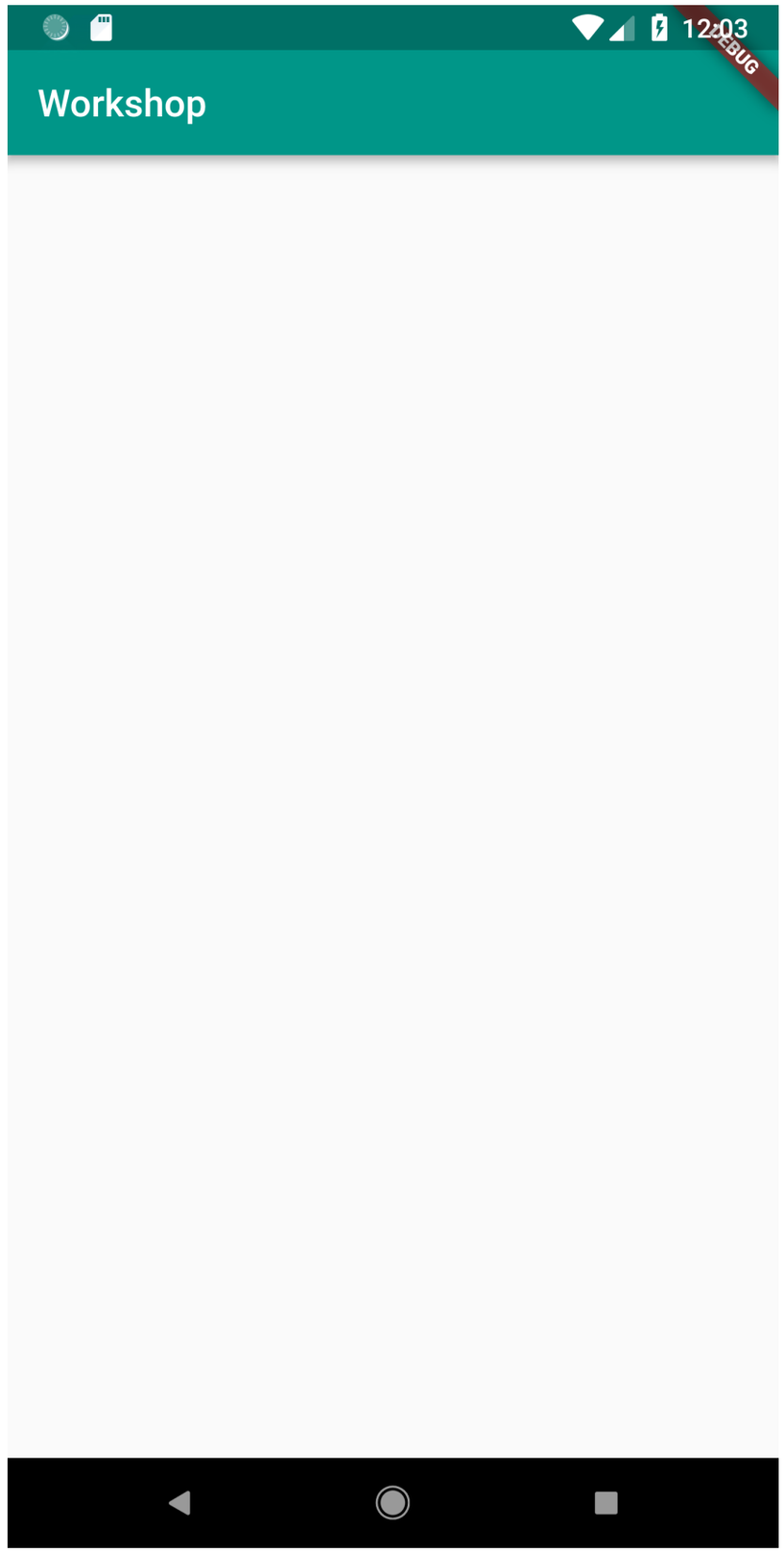
ESTRUTURA

```dart
class HomePage extends StatelessWidget {

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Workshop'),
      ),
      drawer: Drawer(),
      floatingActionButton: FloatingActionButton(
        child: Icon(Icons.add),
        onPressed: () {},
      )
    );
  }
}
```



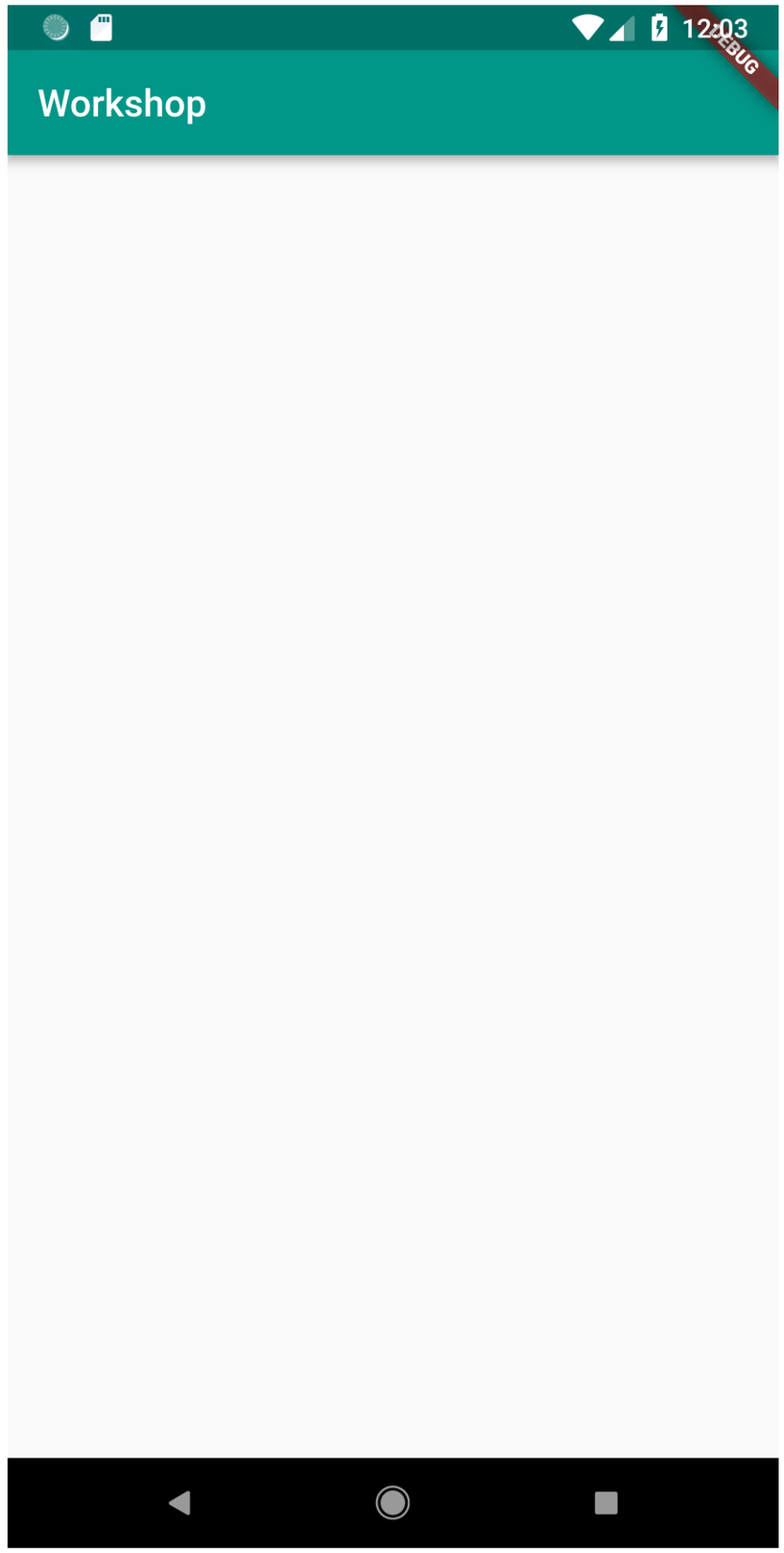
ESTRUTURA

```dart
class HomePage extends StatelessWidget {

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Workshop'),
      ),
      body: playersList(),
    );
  }

  Widget playersList() {
    return Container();
  }
}
```



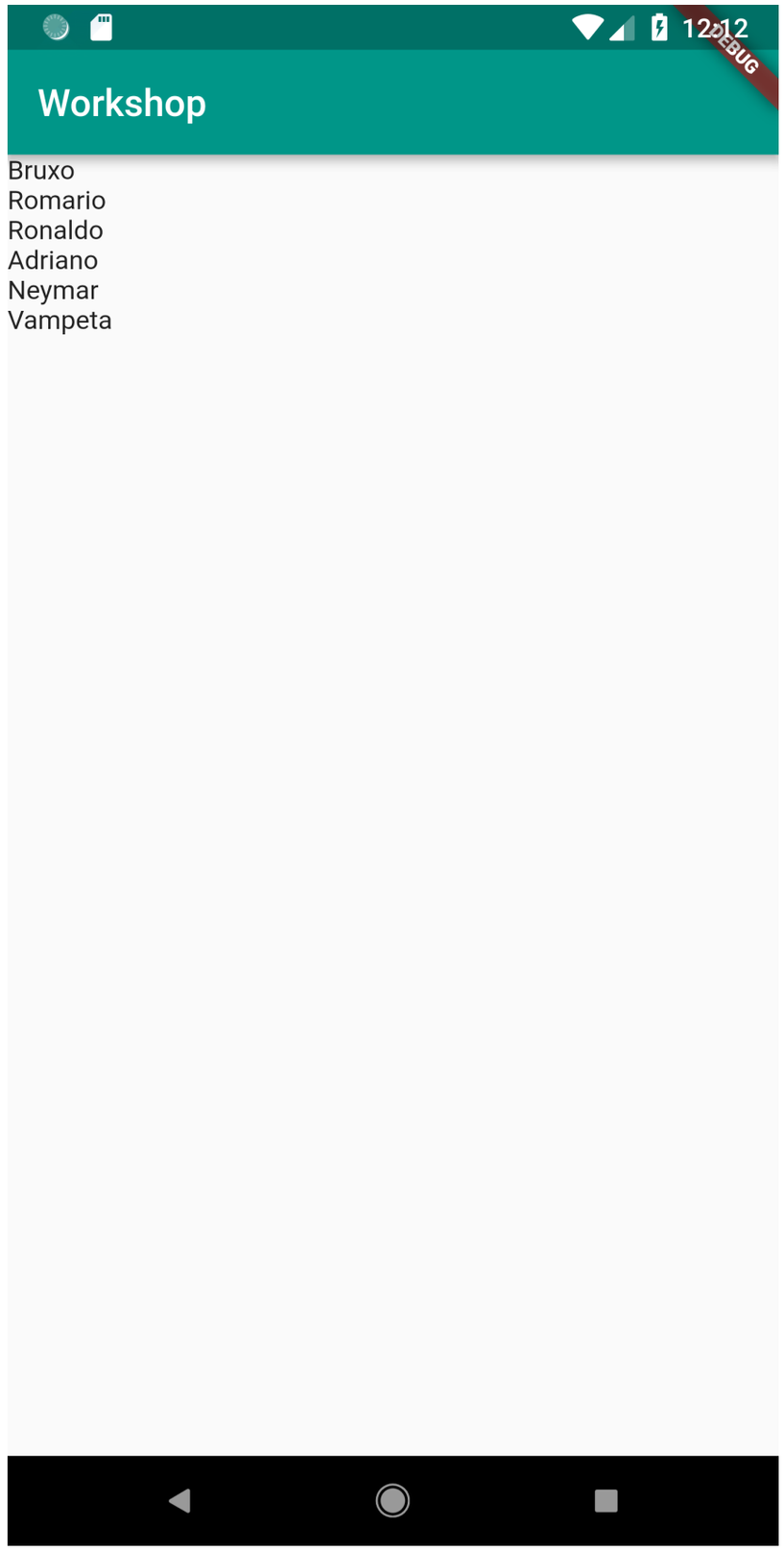
ESTRUTURA

```dart
class HomePage extends StatelessWidget {
  final players = [
    'Bruxo',
    'Romário',
    'Ronaldo',
    'Adriano',
    'Neymar',
    'Vampeta'
  ];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Workshop'),
      ),
      body: playersList(),
    );
  }

  Widget playersList() {
    return Container();
  }
}
```



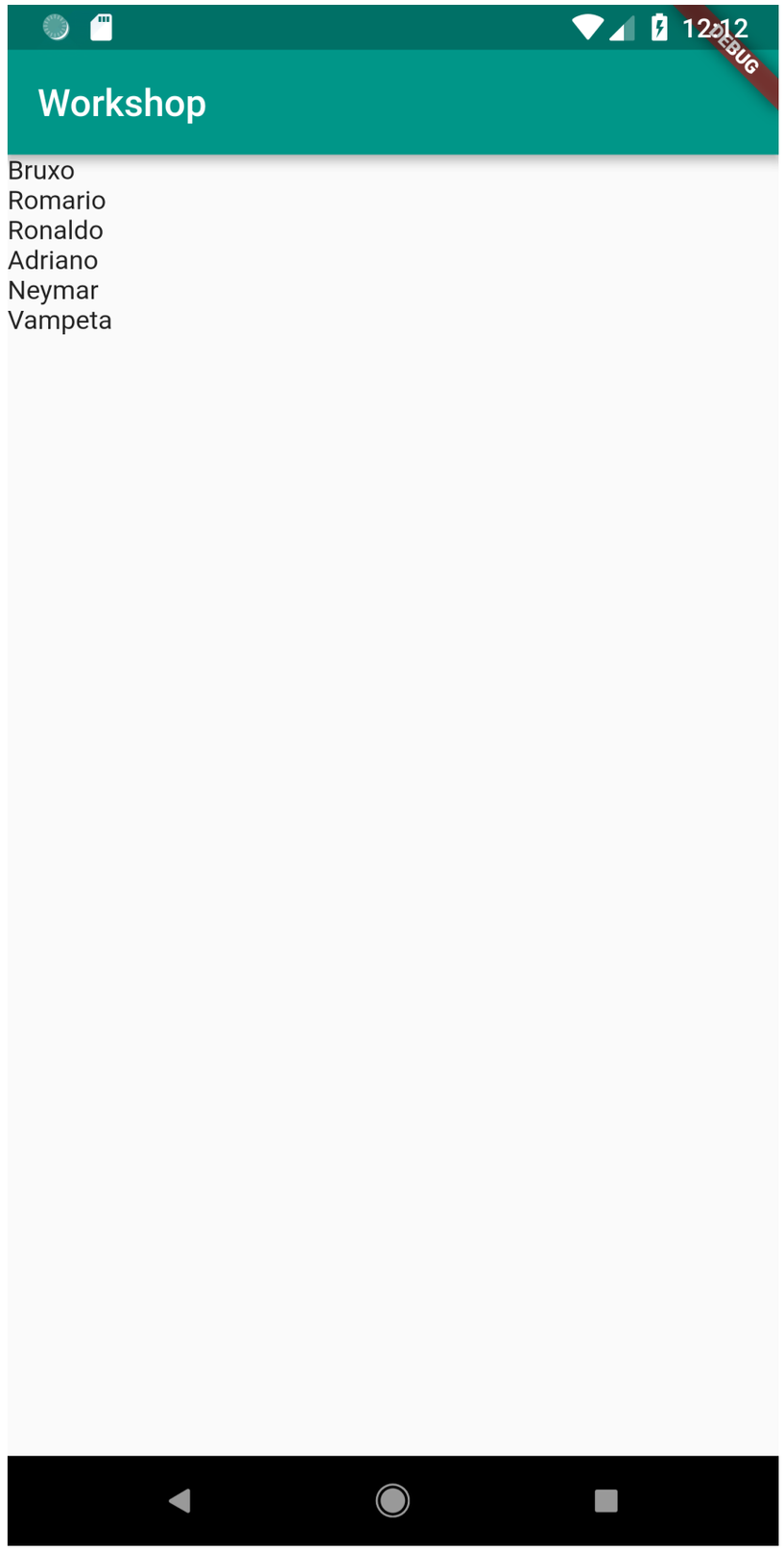
ESTRUTURA

```
Widget playersList() {
    return ListView.builder(
      itemCount: players.lenght,
      itemBuilder: (context, index) {
        return Text(players[index]);
      }
    );
  }
```



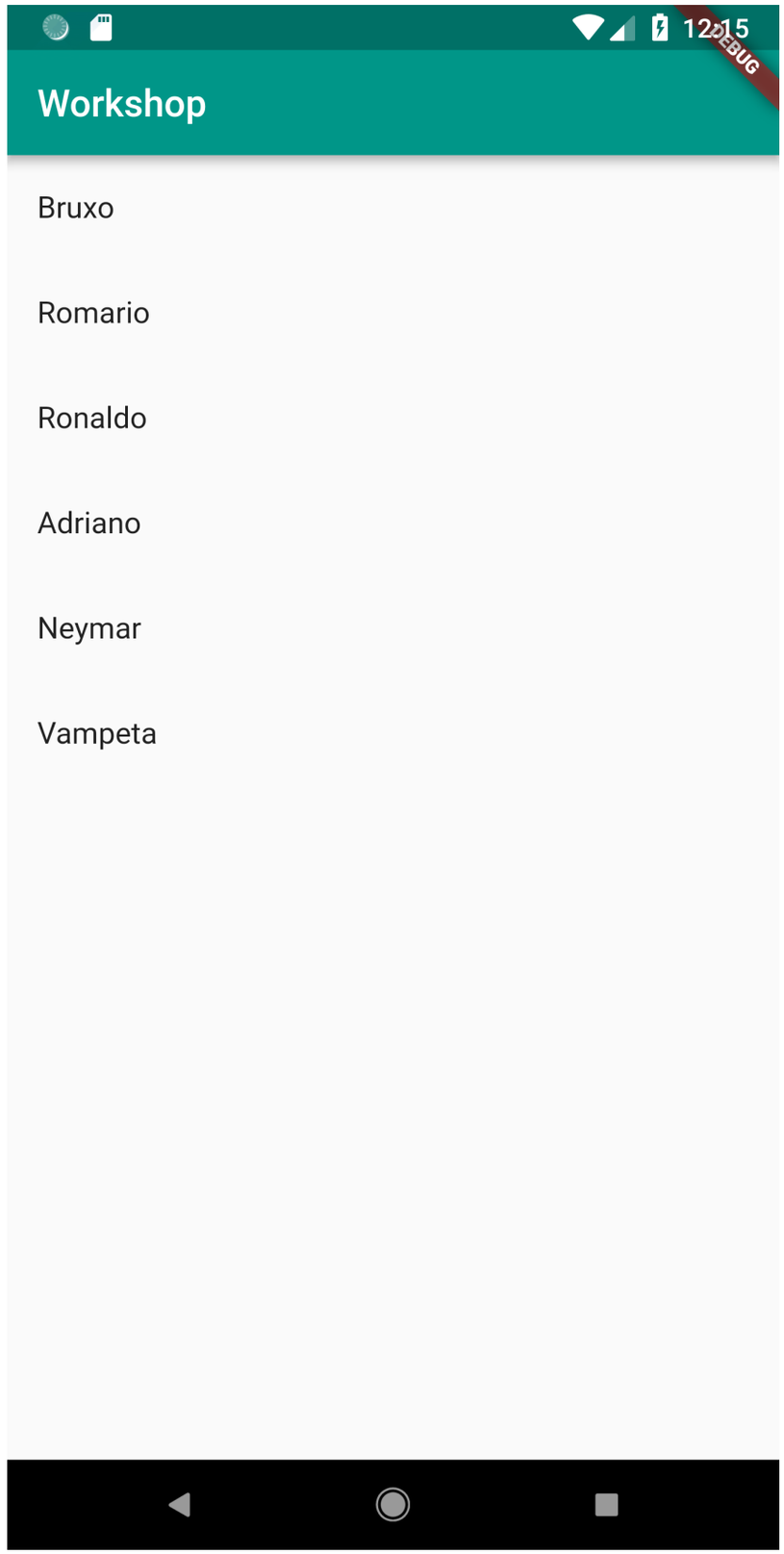
ESTRUTURA

```
Widget playerRow(String playerName) {
    return ListTile(
      title: Text(playerName)
    );
  }
```



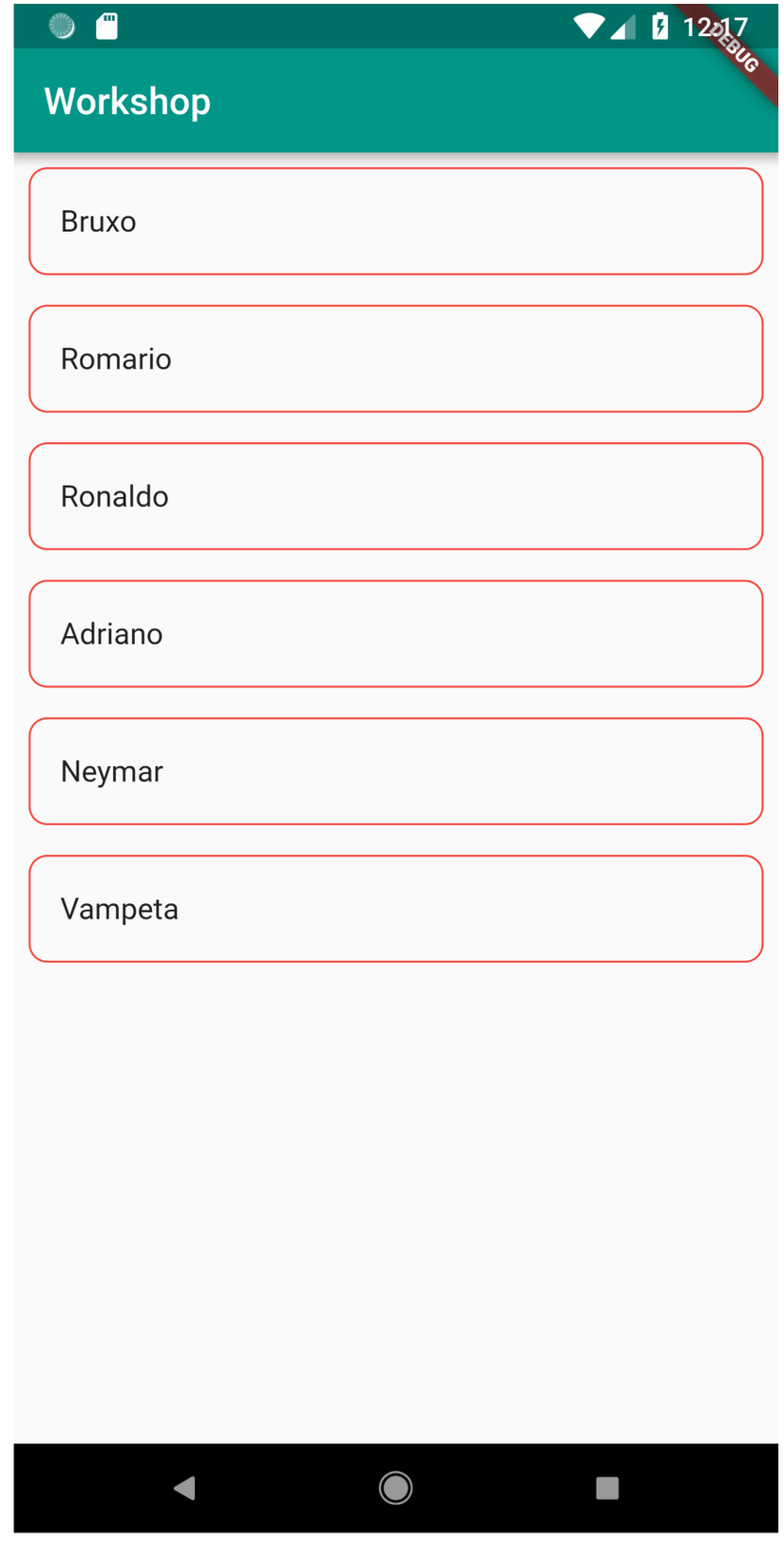
ESTRUTURA

```
Widget playersList() {
    return ListView.builder(
      itemCount: players.lenght,
      itemBuilder: (context, index) {
        return playerRow(players[index]);
      }
    );
  }

  Widget playerRow(String playerName) {
    return ListTile(
      title: Text(playerName)
    );
}
```



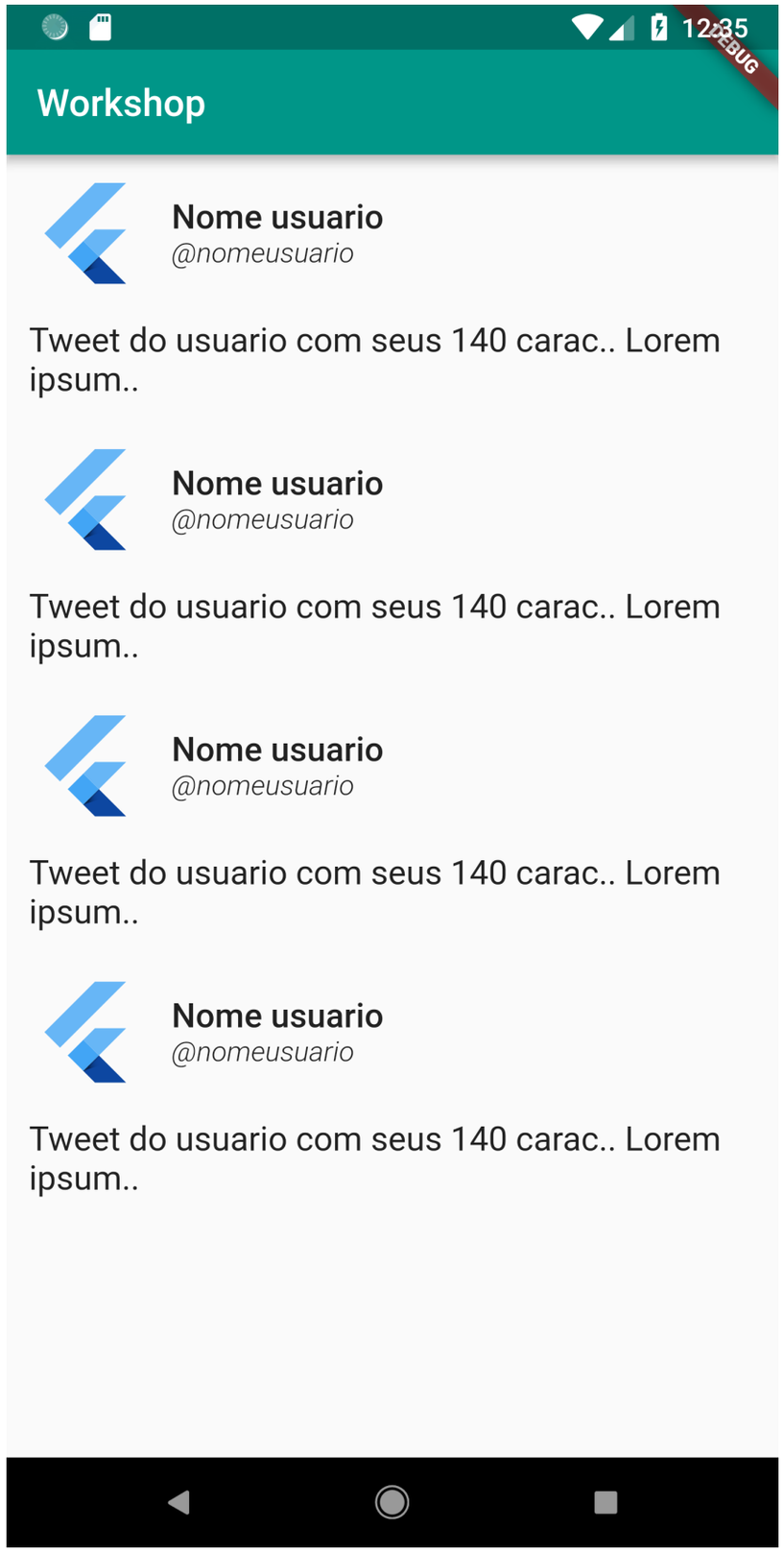
ESTRUTURA

```dart
Widget playerRow(String playerName) {
    return Padding(
      padding: const EdgeInsets.all(8.0),
      child: Container(
        decoration: BoxDecoration(
          border: Border.all(color: Colors.red),
          borderRadius: BorderRadius.circular(10.0)
        ),
        child: ListTile(
          title: Text(playerName),
        ),
      ),
    );
}
```



ESTRUTURA

```
Column(
   children: <Widget>[],
);

Row(
   children: <Widget>[],
);
```



FIREBASE

O QUE É FIREBASE? 🤔

Firebase nos ajuda a construir melhores apps mobile e a crescer o negócio

Construindo aplicativos de forma rápida, sem ter que gerenciar infra-estrutura

O QUE É FIREBASE? 🙄

Gratuito 💸 em sua maior parte e com preços acessíveis para escalar

Spark Plan	Flame Plan	Blaze Plan
Generous limits for hobbyists	Fixed pricing for growing apps	Calculate pricing for apps at scale
Free	$25/month	Pay as you go
		✓ Free usage from Spark plan included*

BUILD

- Realtime Database
  - Cloud Firestore (Beta)
- Authentication
- ML Kit(Beta)
- Hosting
- Cloud Storage
- Cloud Functions

IMPROVE

- Crashlytics
- Performance Monitoring
- Test Lab

GROW

- Google analytics
- Cloud messaging
- Dynamic links
- Remote config
- App indexing
- A/B Testing (Beta)
- Predictions (Beta)


FLUTTERFIRE

Available FlutterFire plugins

Plugin	Version	Firebase feature	Source code
cloud_firestore	pub v0.8.2+1	Cloud Firestore	packages/cloud_firestore
cloud_functions	pub v0.0.4	Cloud Functions	packages/cloud_functions
firebase_admob	pub v0.6.1+1	Firebase AdMob	packages/firebase_admob
firebase_analytics	pub v1.0.4	Firebase Analytics	packages/firebase_analytics
firebase_auth	pub v0.6.2+1	Firebase Authentication	packages/firebase_auth
firebase_core	pub v0.2.5+1	Firebase Core	packages/firebase_core
firebase_database	pub v1.0.5	Firebase Realtime Database	packages/firebase_database
firebase_dynamic_links	pub v0.1.0	Firebase Dynamic Links	packages/firebase_dynamic_links
firebase_messaging	pub v2.0.2	Firebase Cloud Messaging	packages/firebase_messaging
firebase_ml_vision	pub v0.2.0	Firebase ML Kit	packages/firebase_ml_vision
firebase_performance	pub v0.0.8	Firebase Performance Monitoring	packages/firebase_performance
firebase_remote_config	pub v0.0.6+1	Firebase Remote Config	packages/firebase_remote_config
firebase_storage	pub v1.0.4	Firebase Cloud Storage	packages/firebase_storage

CODE 👨‍💻

CODE 👨‍💻