Google

GDGMontevideo

@gdgmontevideo

Flutter101

Todo es un widget

Martin Peruchena
ResponsiveIT

GDG Montevideo

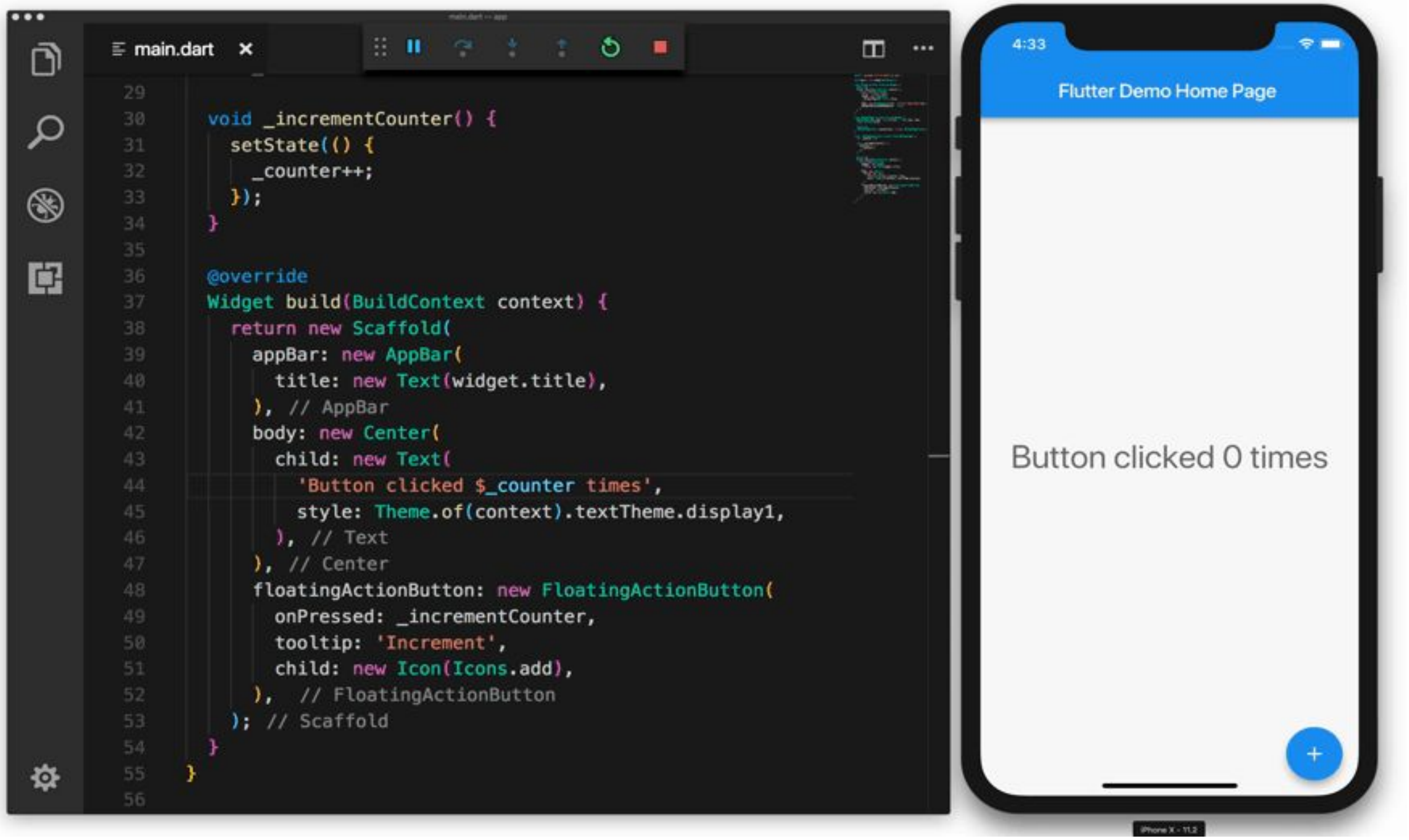
Características

Desarrollo rápido

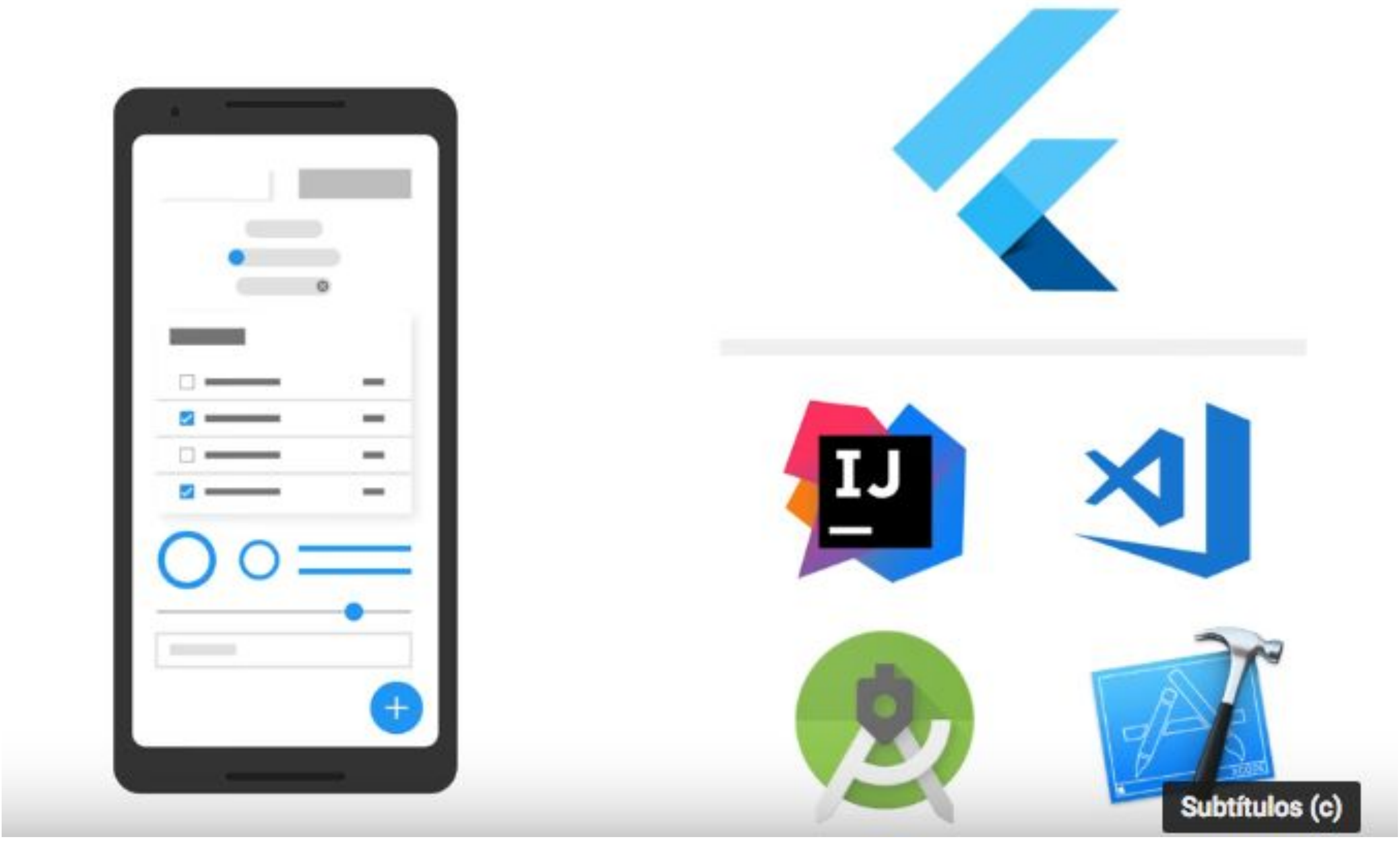
UI Flexible y expresiva

Desarrollo unificado

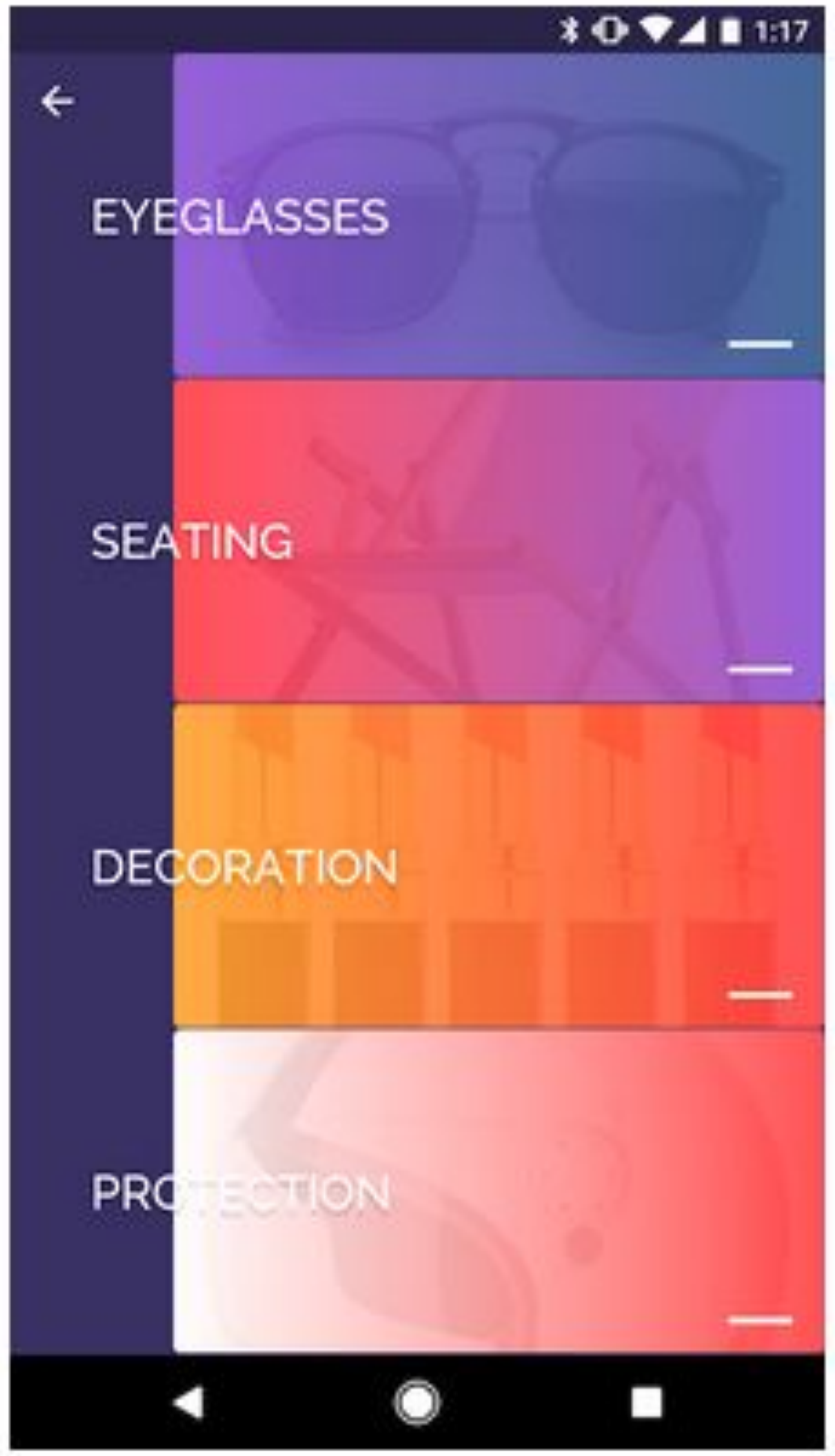
Desarrollo rápido

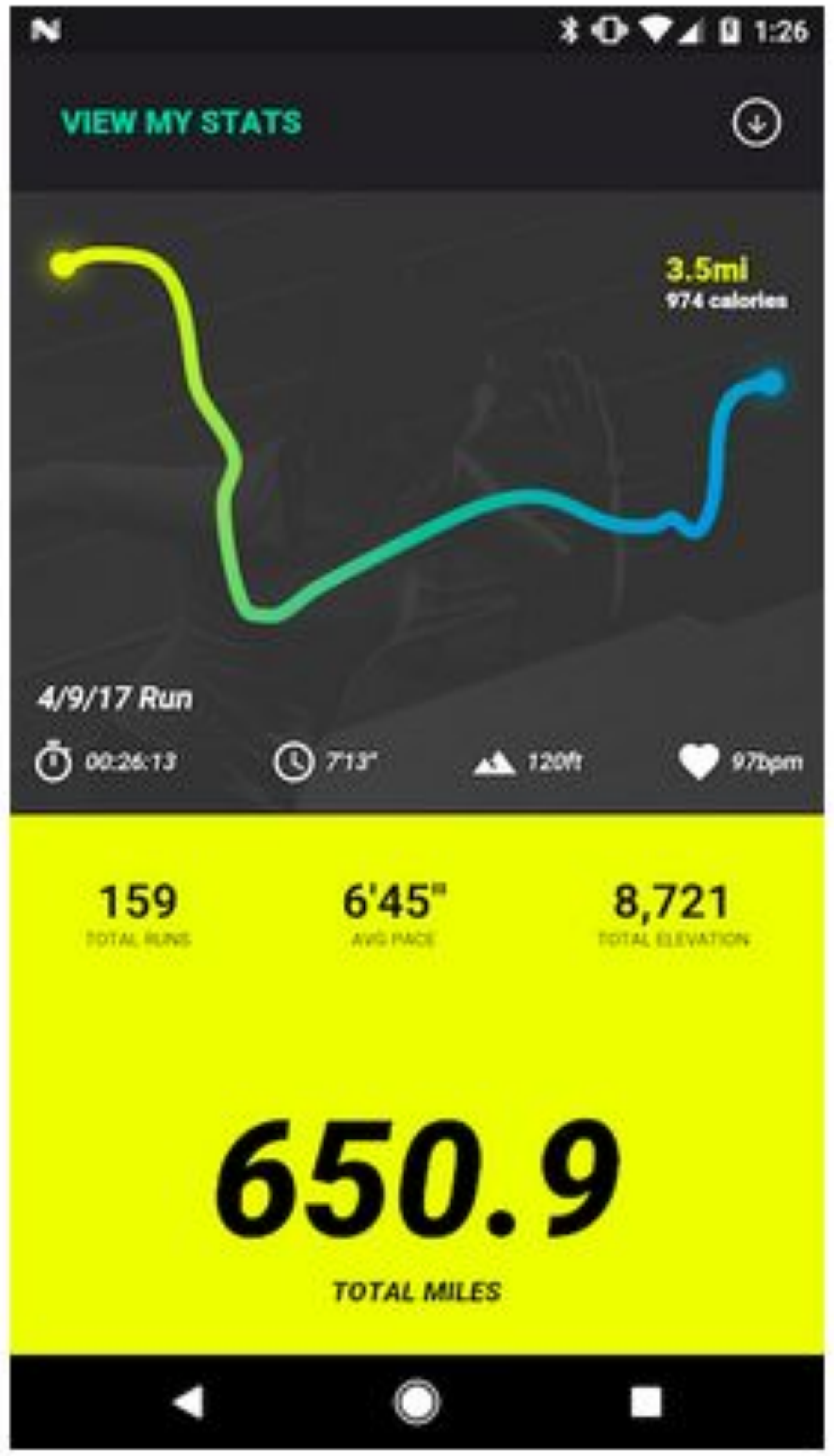


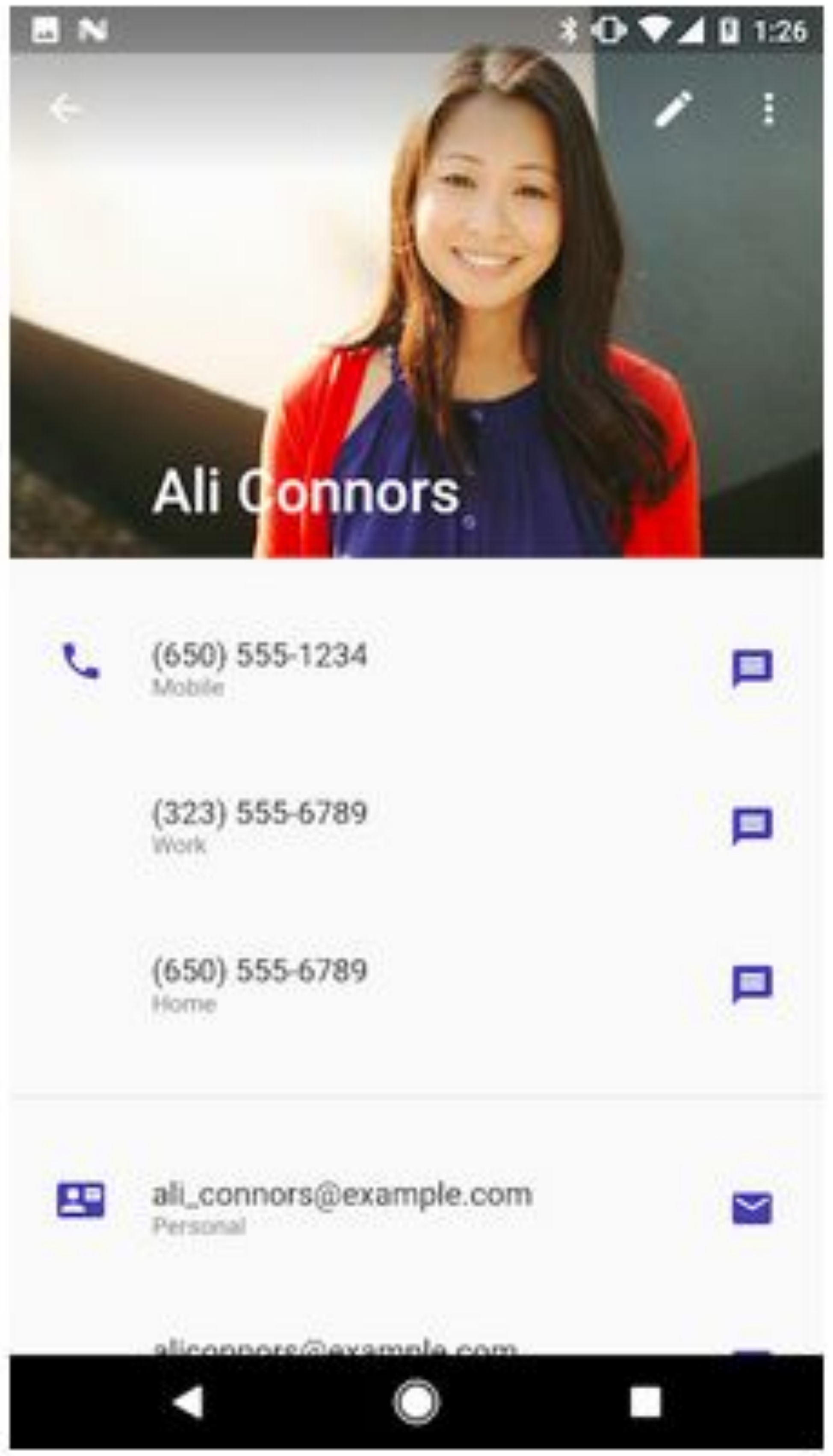
```
  void _incrementCounter() {
    setState(() {
      _counter++;
    });
  }

  @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
  }
}
```



UI flexible y expresiva







Ali Connors

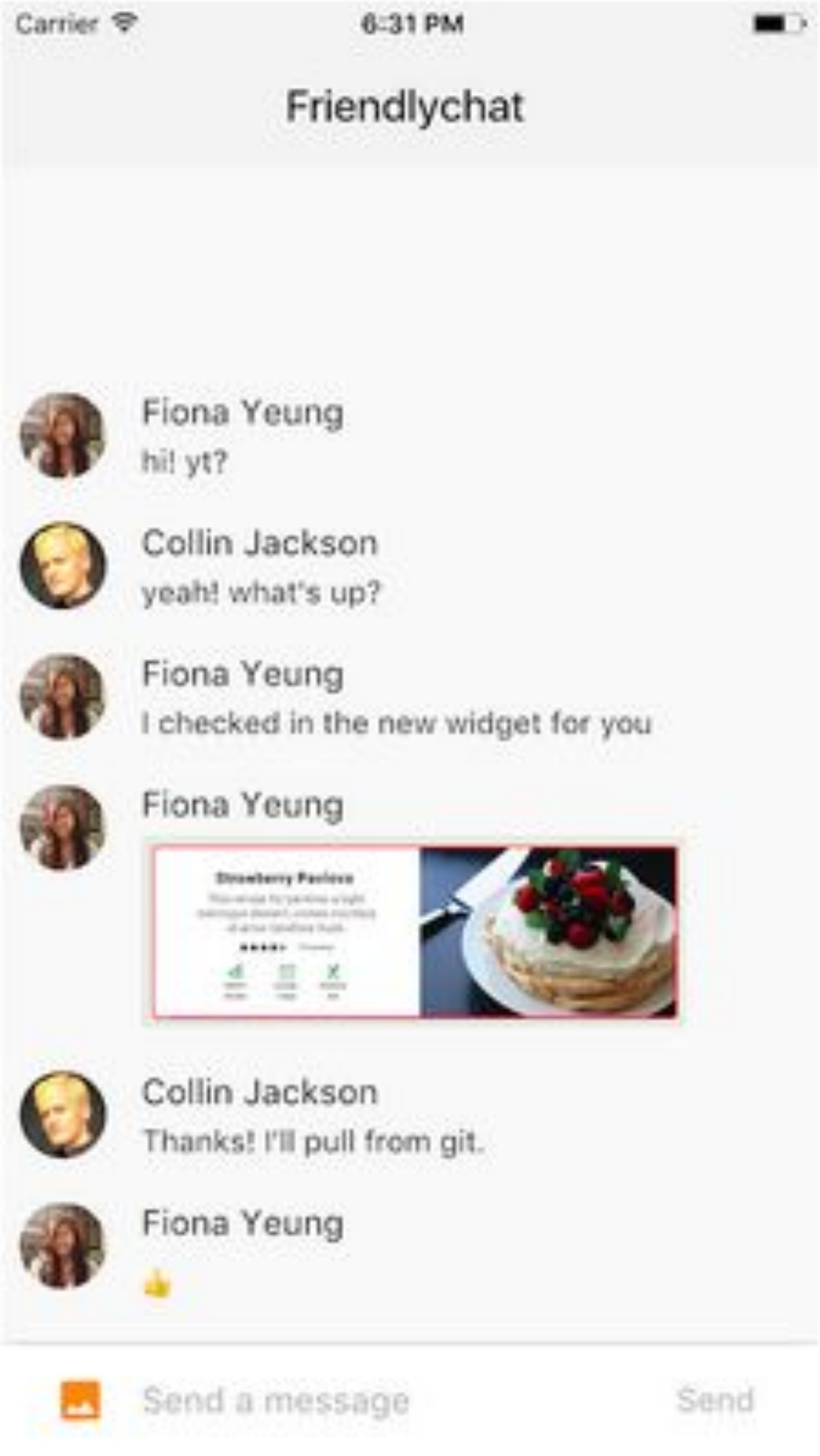
(650) 555-1234
Mobile

(323) 555-6789
Work

(650) 555-6789
Home

ali_connors@example.com
Personal

aliconnors@example.com



Fiona Yeung
hi! yt?

Collin Jackson
yeah! what's up?

Fiona Yeung
I checked in the new widget for you

Fiona Yeung

Collin Jackson
Thanks! I'll pull from git.

Fiona Yeung
👍

Desarrollo unificado

Moon Cycle
Moon Cycle

Canales de plataforma

```
    :
Future<Null> getBatteryLevel() async {
  var batteryLevel = 'unknown';
  try {
    int result = await methodChannel.invokeMethod('getBatteryLevel');
    batteryLevel = 'Battery level: $result%';
  } on PlatformException {
    batteryLevel = 'Failed to get battery level.';
  }
  setState(() {
    _batteryLevel = batteryLevel;
  });
}
```

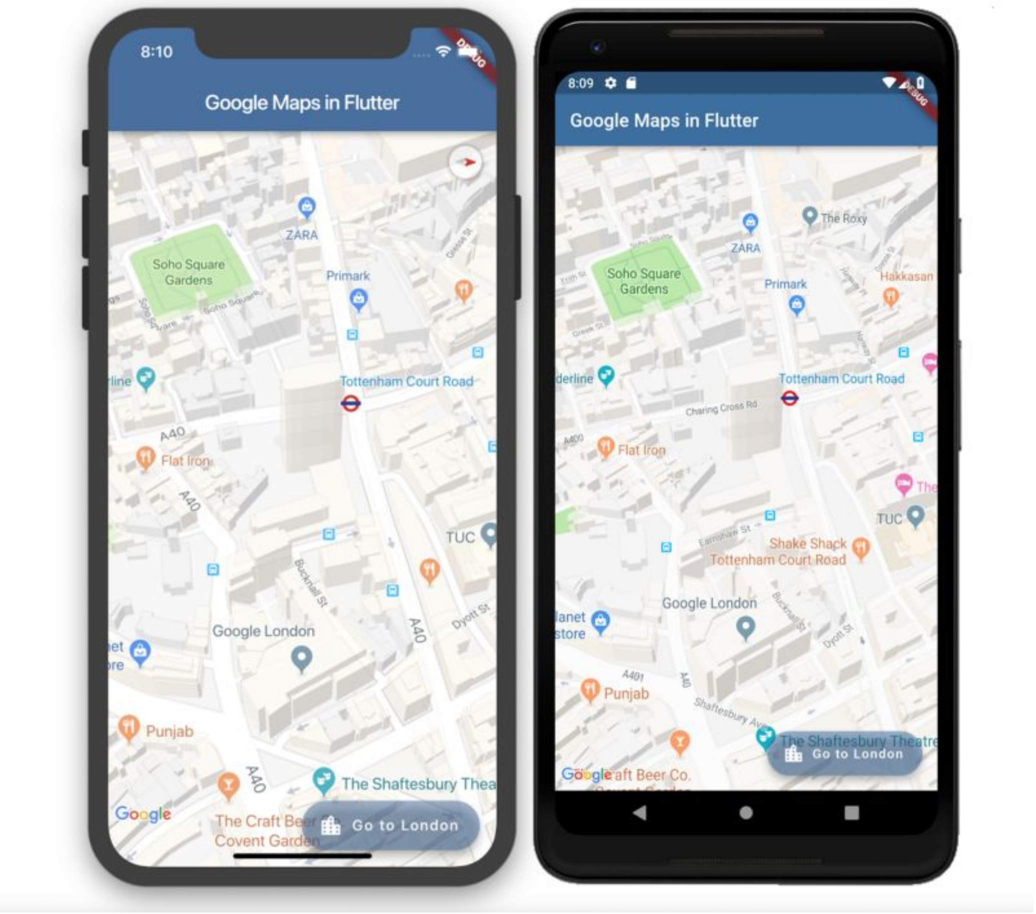
https://github.com/flutter/flutter/tree/master/examples/platform_channel

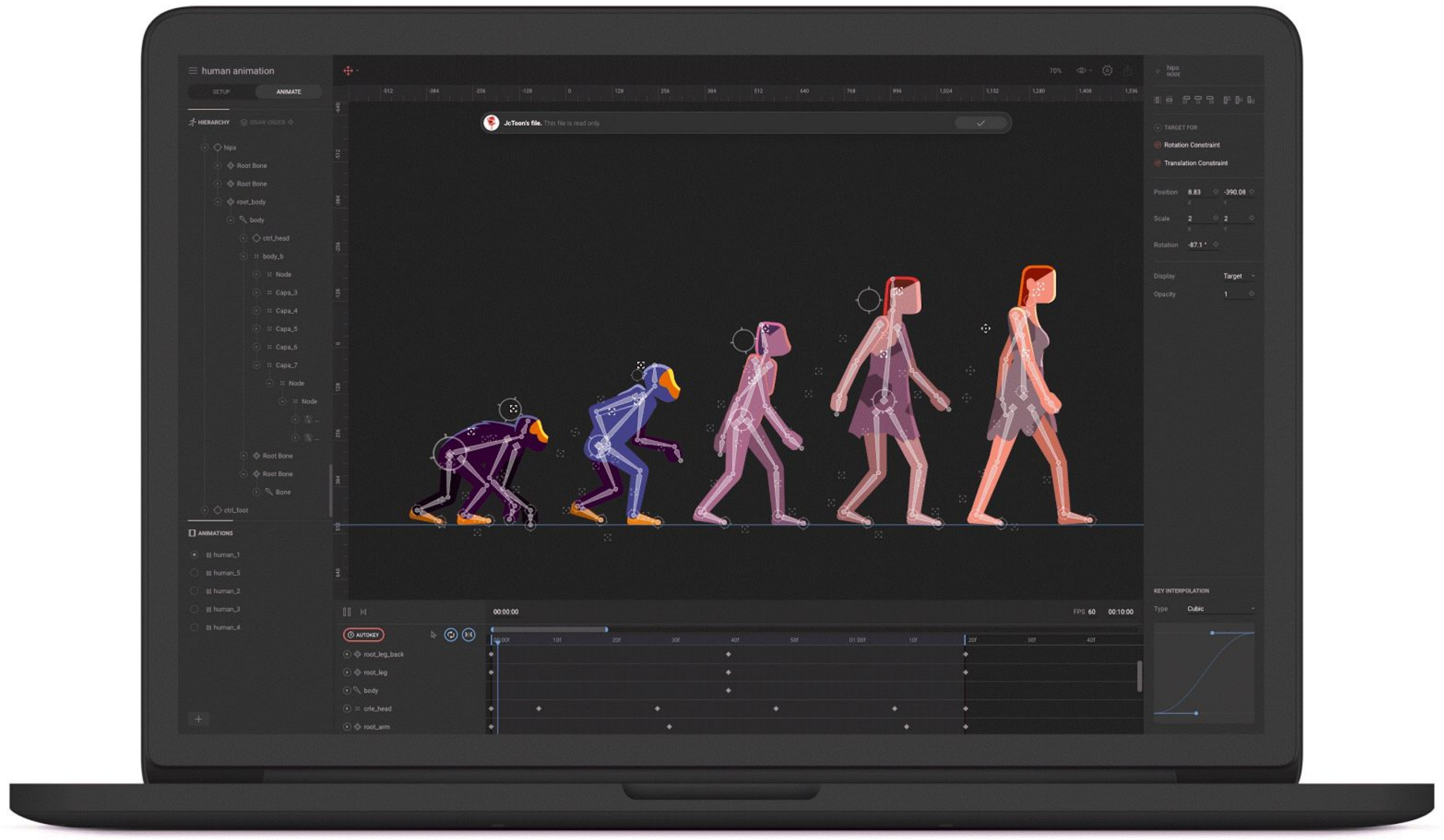
Flutter 1.0

Announcing

Dart 2.1

- Faster tools, smaller output
- Compile-time type checking
- Int to double value inference
- New mixin syntax & support



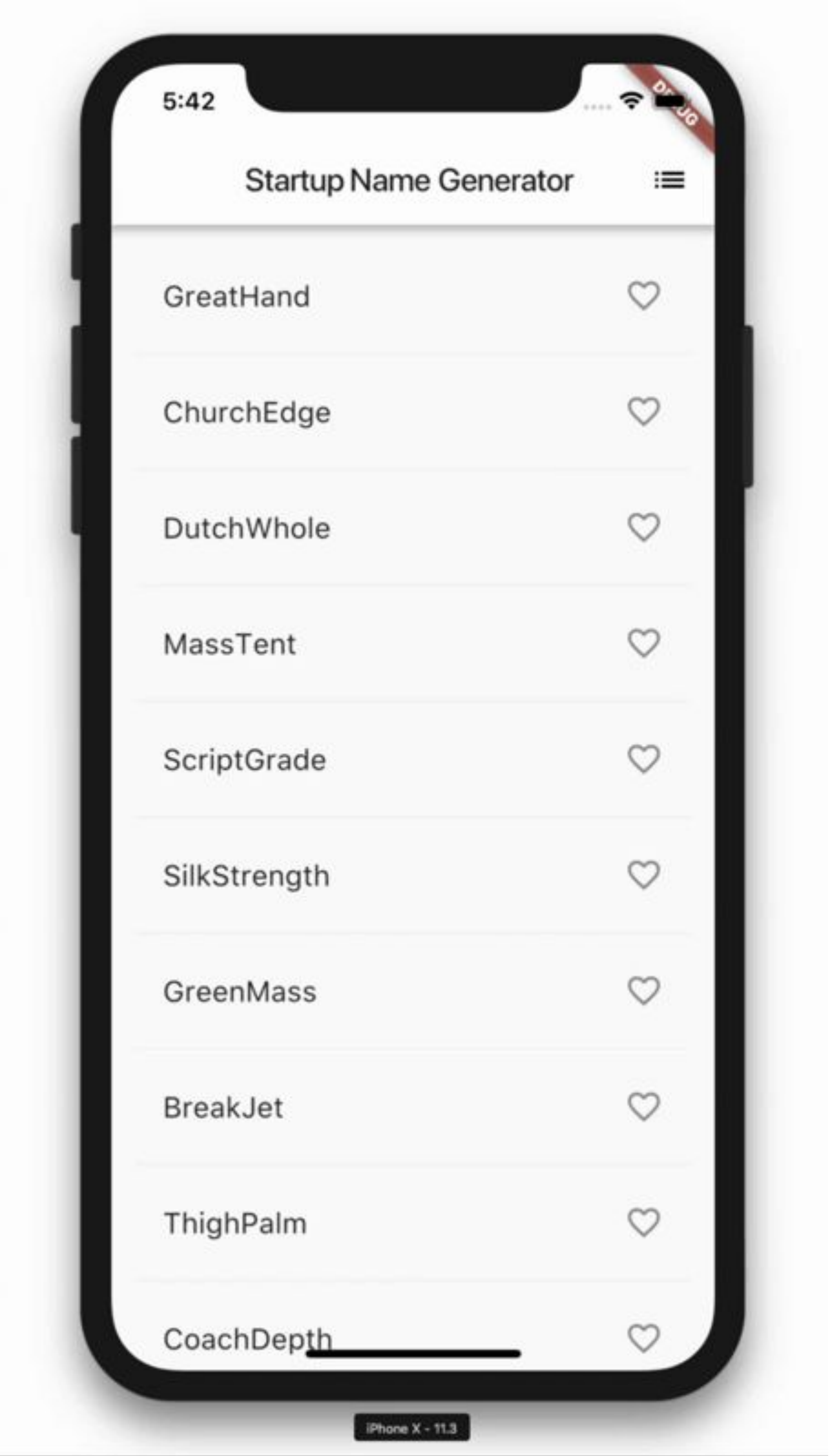


Hello World

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

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

  class MyApp extends StatelessWidget {
   @override
   Widget build(BuildContext context) {
    return MaterialApp(
     title: 'Welcome to Flutter',
     home: Scaffold(
      appBar: AppBar(
       title: Text('Welcome to Flutter'),
      ),
      body: Center(
       child: Text('Hello World'),
      ),
     ),
    );
   }
  }
```

Write Your First App

- Generar nombres para una startup
- Mostrar listado de los nombres
- El usuario puede seleccionar y deseleccionar nombres

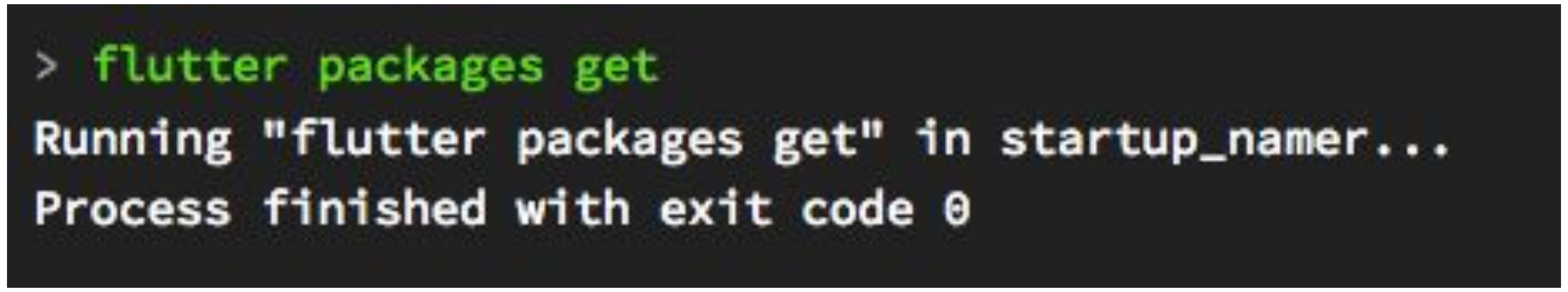


Startup Name Generator

- GreatHand ♡
- ChurchEdge ♡
- DutchWhole ♡
- MassTent ♡
- ScriptGrade ♡
- SilkStrength ♡
- GreenMass ♡
- BreakJet ♡
- ThighPalm ♡
- CoachDepth ♡

Write Your First App

```
dependencies:
  flutter:
    sdk: flutter

  cupertino_icons: ^0.1.0
  english_words: ^3.1.0
```



```
> flutter packages get
Running "flutter packages get" in startup_namer...
Process finished with exit code 0
```

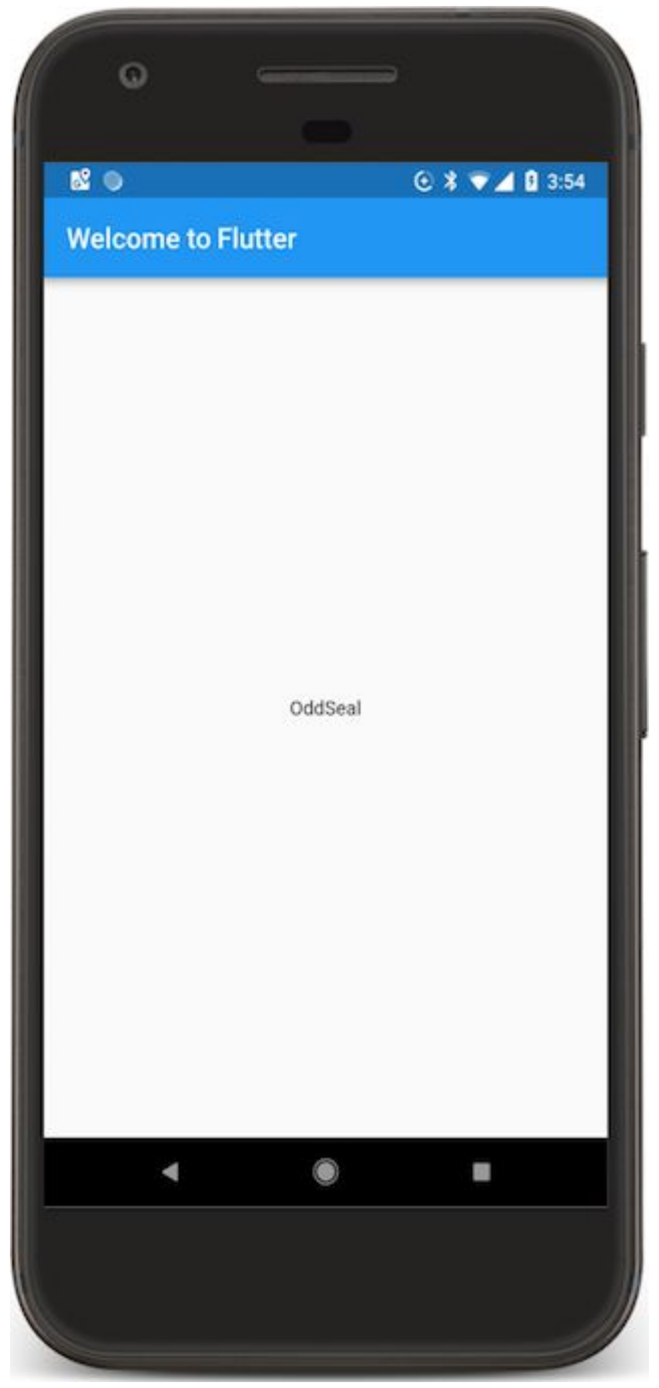
lib/main.dart

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

Write Your First App

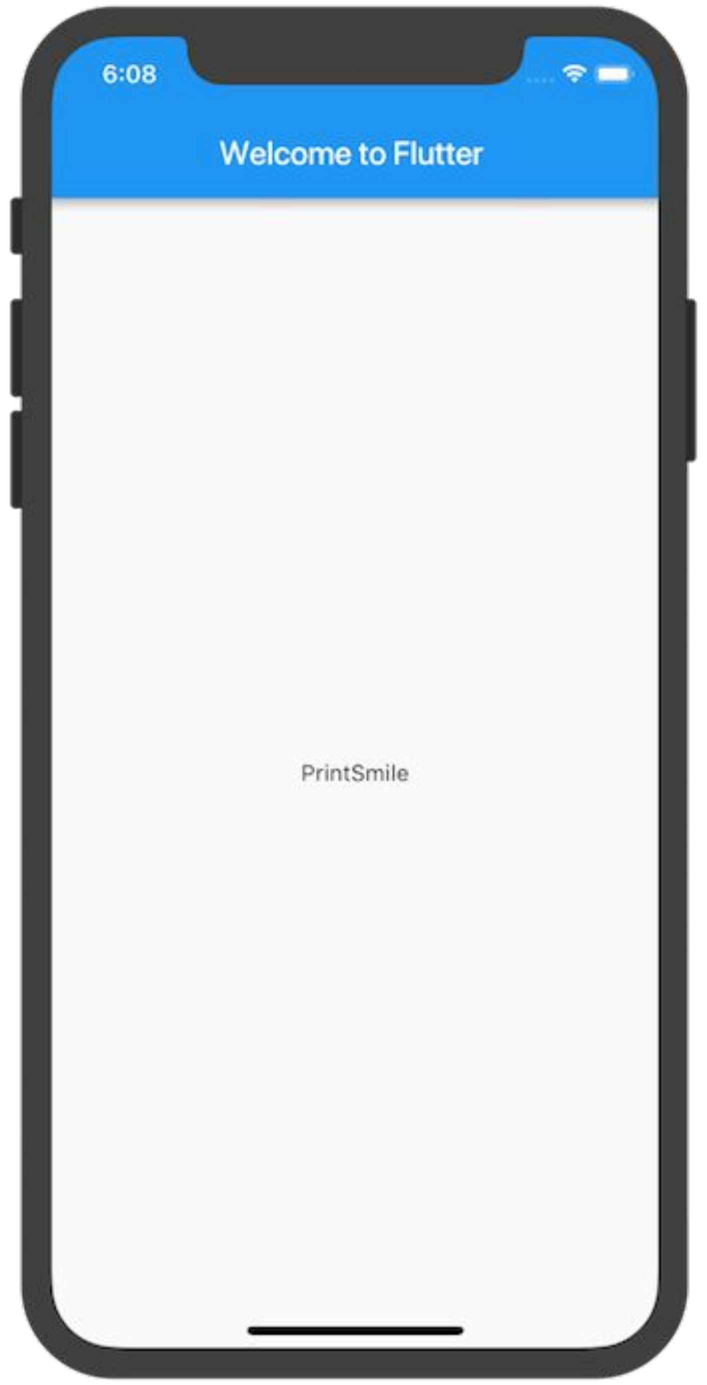
```
import 'package:flutter/material.dart';
import 'package:english_words/english_words.dart';

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    final wordPair = WordPair.random();
    return MaterialApp(
      title: 'Welcome to Flutter',
      home: Scaffold(
        appBar: AppBar(
          title: Text('Welcome to Flutter'),
        ),
        body: Center(
          child: Text(wordPair.asPascalCase),  // con este texto resaltado.
        ),
      ),
    );
  }
}
```

Write Your First App



Welcome to Flutter

OddSeal



Welcome to Flutter

PrintSmile

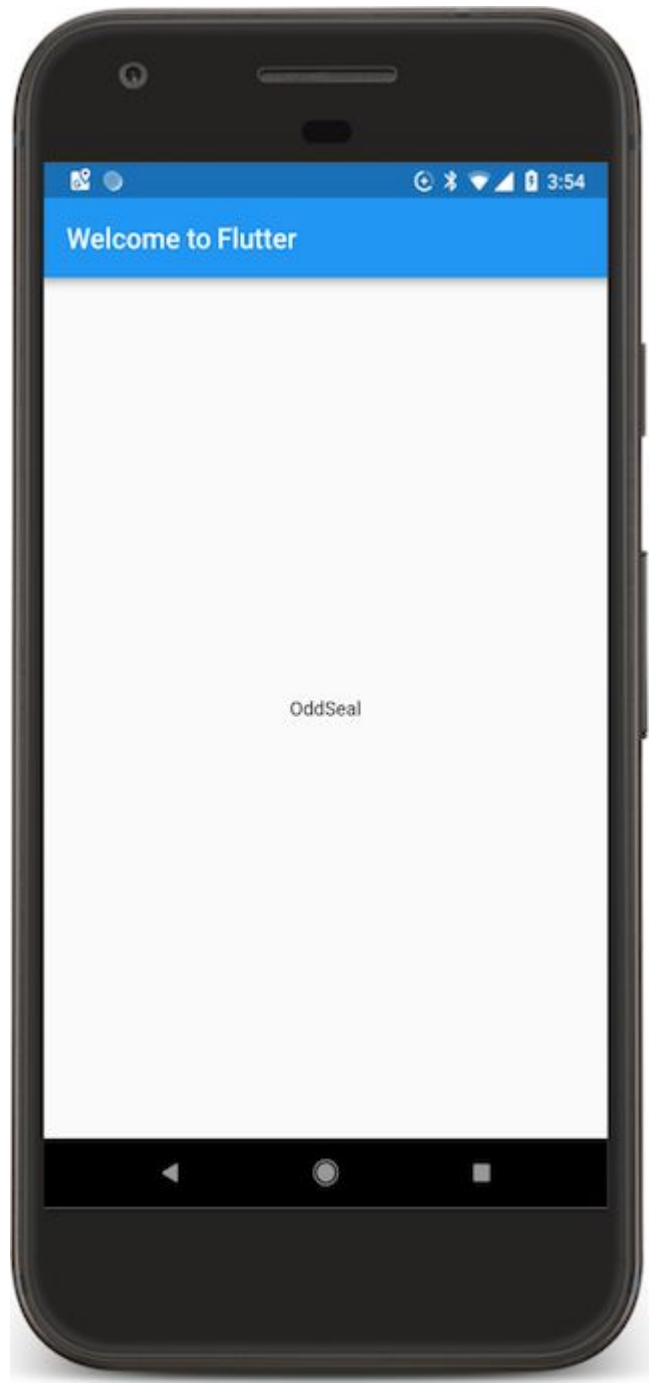
Write Your First App

```
class RandomWords extends StatefulWidget {
  @override
  RandomWordsState createState() => new RandomWordsState();
 }
```

```
class RandomWordsState extends State<RandomWords> {
  @override
  Widget build(BuildContext context) {
    final wordPair = WordPair.random();
    return Text(wordPair.asPascalCase);
  }
}
```

Write Your First App

```
class MyApp extends StatelessWidget {
 @override
 Widget build(BuildContext context) {
  final wordPair = WordPair.random();  // Borra esta línea
  return MaterialApp(
   title: 'Welcome to Flutter',
   home: Scaffold(
   appBar: AppBar(
    title: Text('Welcome to Flutter'),
   ),
   body: Center(
    //child: Text(wordPair.asPascalCase), // reemplazar el texto resaltado por ...
        child: RandomWords(),
       ),
      ),
     );
    }
}
```



Welcome to Flutter

OddSeal

Write Your First App

```
class RandomWordsState extends State<RandomWords> {
   final _suggestions = <WordPair>[];

  final _biggerFont = const TextStyle(fontSize: 18.0);
   ...
 }
```

```
class RandomWordsState extends State<RandomWords> {
   ...
   Widget _buildSuggestions() {
    return ListView.builder(
      padding: const EdgeInsets.all(16.0),
      itemBuilder: (context, i) {
        // Añade un widget divisor de un píxel de alto antes de cada fila en theListView.
        if (i.isOdd) return Divider();
        // La sintaxis "i ~/ 2" divide i entre 2 y devuelve un resultado entero.
        // Por ejemplo: 1, 2, 3, 4, 5 se convierte en 0, 1, 1, 2, 2.
        // Esto calcula el número real de pares de palabras en el ListView,
        // menos los widgets divider.
        final index = i ~/ 2;
        // Si ha llegado al final de los pares de palabras disponibles....
        if (index >= _suggestions.length) {
          // ... luego generar 10 más y agregarlos a la lista de sugerencias.
          _suggestions.addAll(generateWordPairs().take(10));
        }
        return _buildRow(_suggestions[index]);
      }
    );
  }
 }
```

```
Widget _buildRow(WordPair pair) {
   return ListTile(
     title: Text(
       pair.asPascalCase,
       style: _biggerFont,
     ),
   );
```

Write Your First App

```
class RandomWordsState extends State<RandomWords> {
  ...
  @override
  Widget build(BuildContext context) {
    final wordPair = WordPair.random(); // Eliminar estas dos lineas
    return Text(wordPair.asPascalCase);
    return Scaffold (
      appBar: AppBar(
        title: Text('Startup Name Generator'),
      ),
      body: _buildSuggestions(),
    );
  }
  ...
}
```

Write Your First App

```
class MyApp extends StatelessWidget {
    @override
    Widget build(BuildContext context) {
      return MaterialApp(
        title: 'Startup Name Generator',
        home: RandomWords(),
      );
    }
  }
```

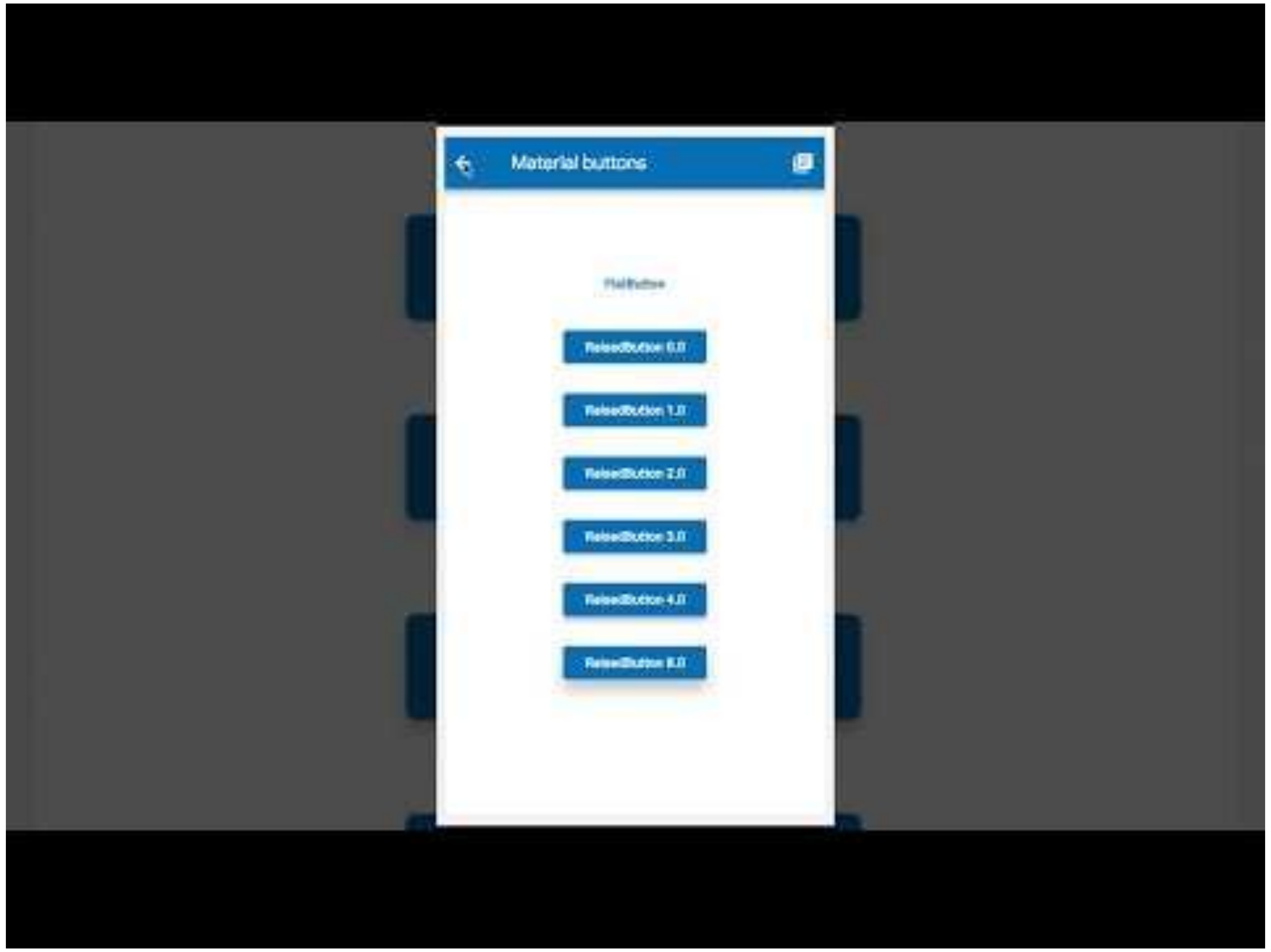
Empezando por...

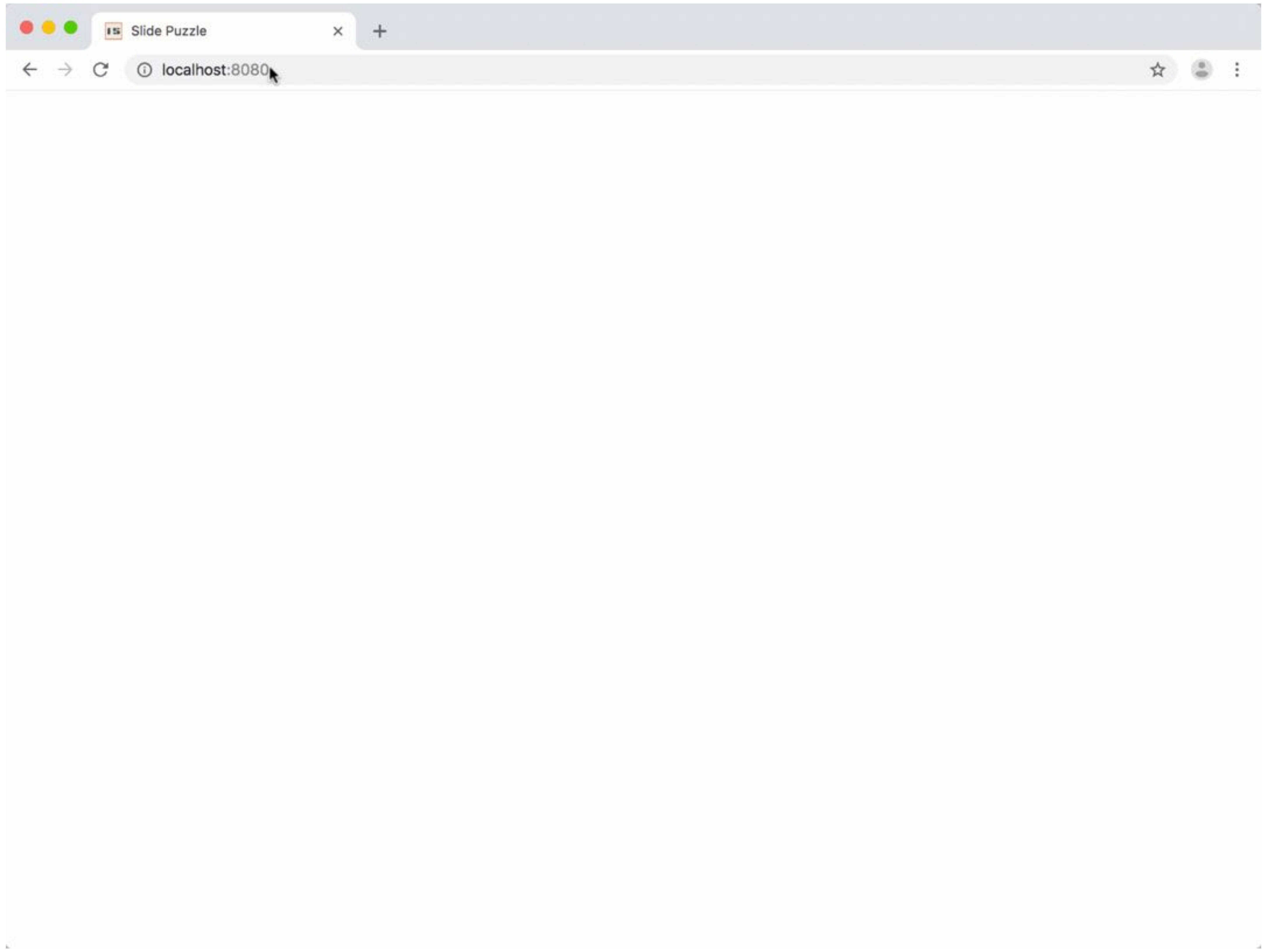
Instalar Flutter
developers.google.com/discovery/libraries

Google CodeLabs
https://codelabs.developers.google.com/?cat=Flutter

Flutter en Udacity
https://www.udacity.com/course/build-native-mobile-apps-with-flutter--ud905

HummingBird





Links

Flutter Live 2018

Sitio Oficial Flutter EN ES

https://flutterstudio.app/

https://material.io/tools/theme-editor/

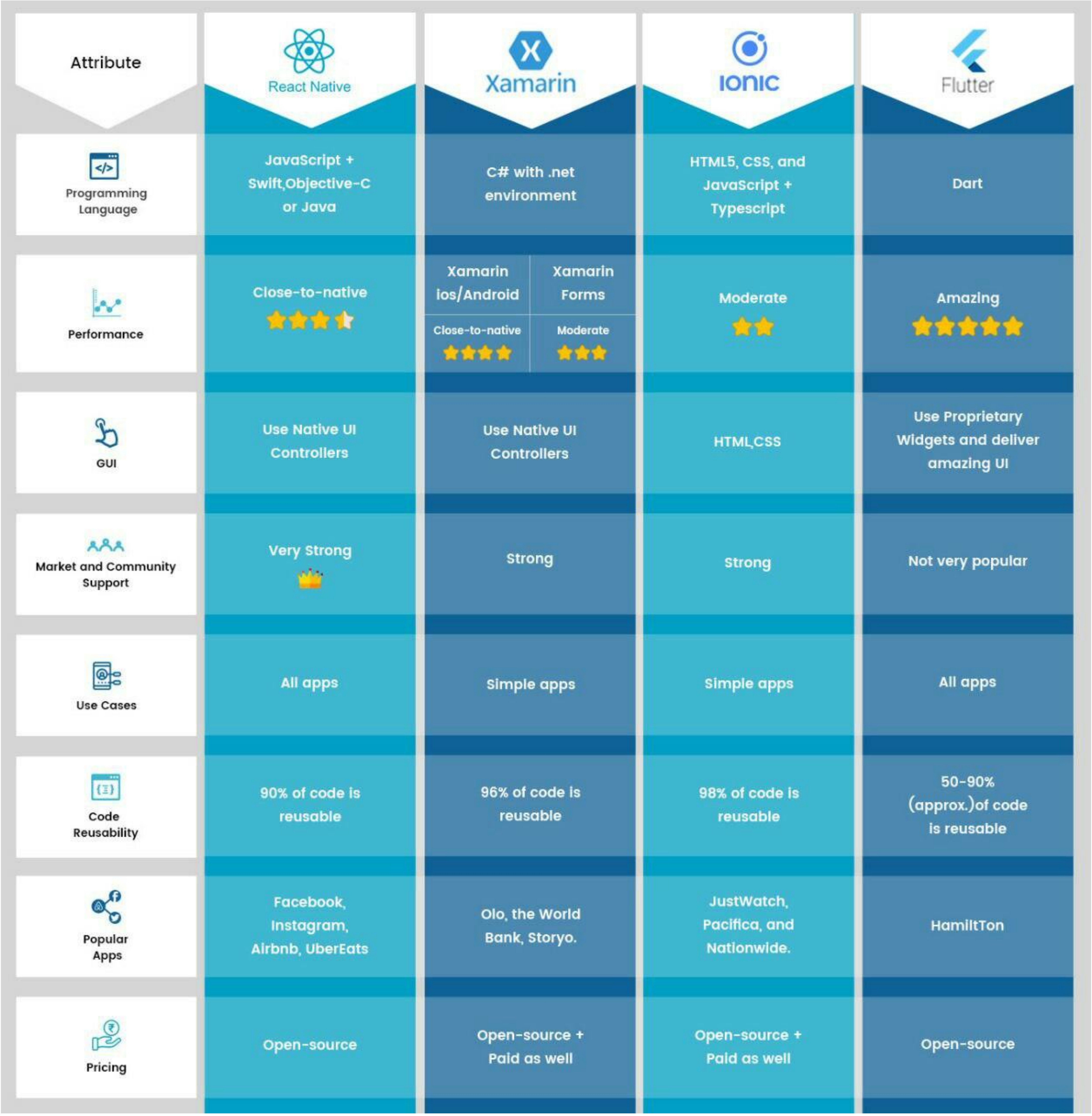
https://material.io/tools/color/

https://github.com/Solido/awesome-flutter

Catálogo de Widgets - Flutter

PorDondeEmpezamosEnFlutter@MediumByMartinPeruchena

Comparativa



Attribute	React Native	Xamarin	Ionic	Flutter
Programming Language	JavaScript + Swift,Objective-C or Java	C# with .net environment	HTML5, CSS, and JavaScript + Typescript	Dart
Performance	Close-to-native ★★★★½	Xamarin ios/Android Close-to-native ★★★★ Xamarin Forms Moderate ★★★	Moderate ★★	Amazing ★★★★★
GUI	Use Native UI Controllers	Use Native UI Controllers	HTML,CSS	Use Proprietary Widgets and deliver amazing UI
Market and Community Support	Very Strong 👑	Strong	Strong	Not very popular
Use Cases	All apps	Simple apps	Simple apps	All apps
Code Reusability	90% of code is reusable	96% of code is reusable	98% of code is reusable	50-90% (approx.)of code is reusable
Popular Apps	Facebook, Instagram, Airbnb, UberEats	Olo, the World Bank, Storyo.	JustWatch, Pacifica, and Nationwide.	HamiltTon
Pricing	Open-source	Open-source + Paid as well	Open-source + Paid as well	Open-source

Gracias! ¿Preguntas?

Martin Peruchena

@washsar

GDGMontevideo

@gdgmontevideo