
Curso de Flutter

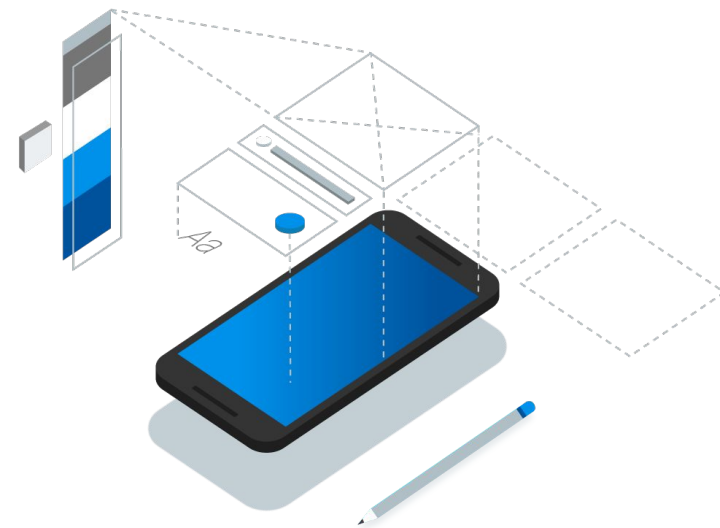
Anahí Salgado



¿Qué es Flutter?

Flutter

SDK de Google creado para diseñar **interfaces nativas iOS y Android**.





2017

alpha



Febrero 2018

beta

Flutter 1.0

Diciembre 2018



Flutter

Utiliza el lenguaje de programación **Dart** y fue creado para programadores de interfaces móviles

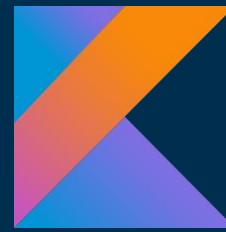


Native

Hibrido

Cross Platform

Native



Hibrido

Cross Platform

Native

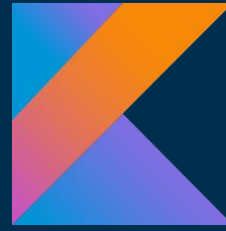


Hibrido



Cross Platform

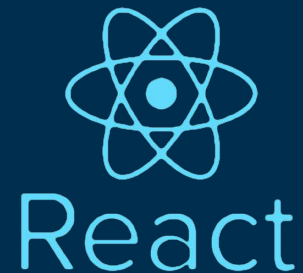
Native



Hibrido



Cross Platform

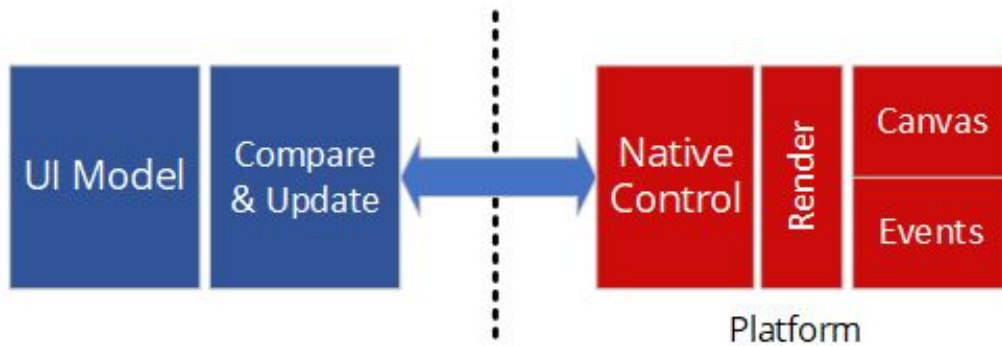


Cross Platform



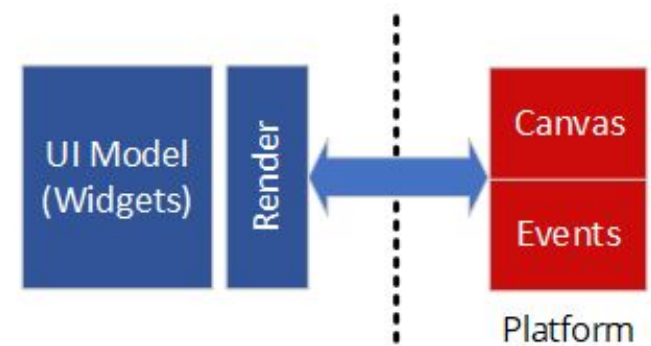
Cross Platform

Other Frameworks



V
S

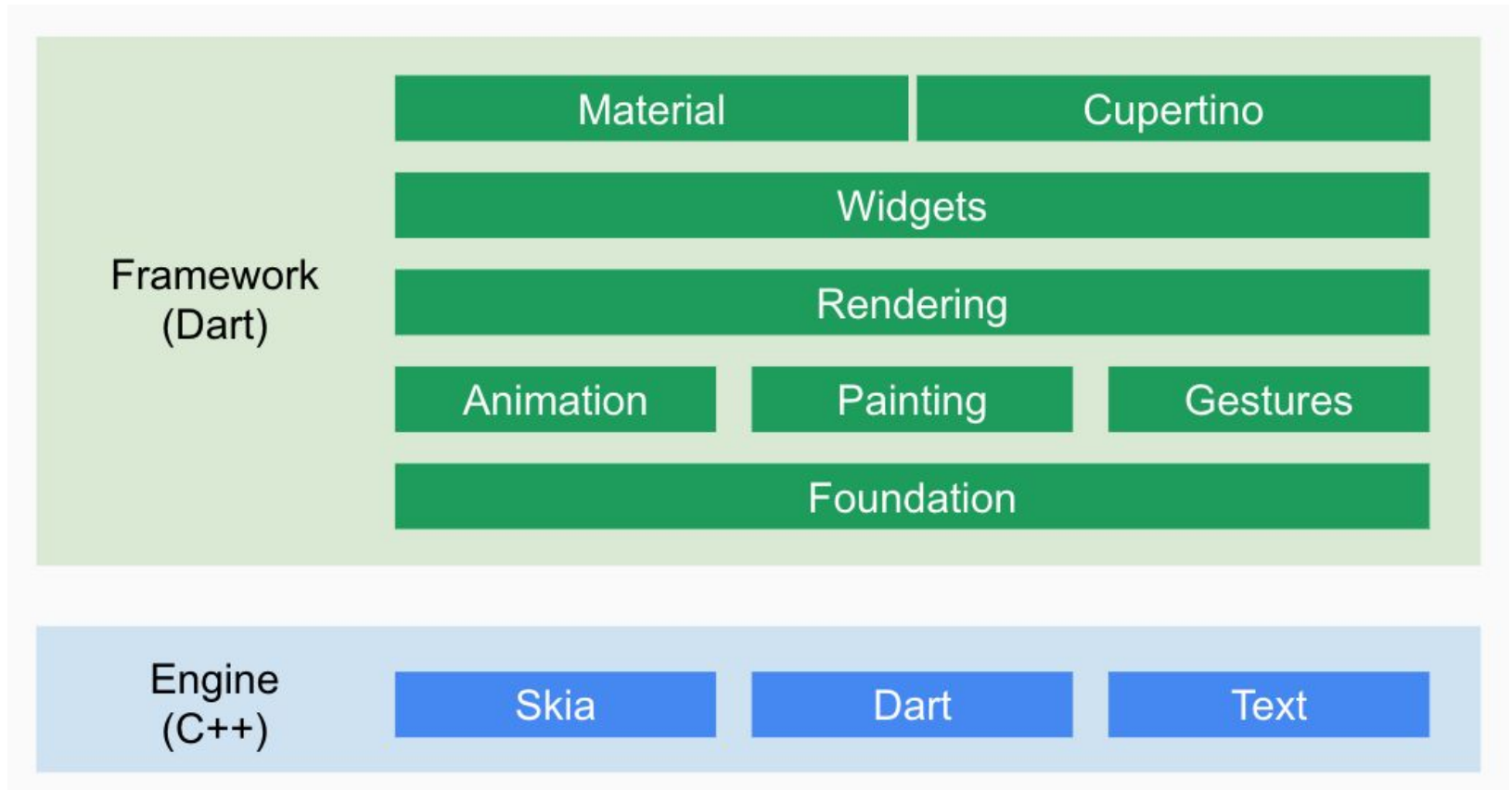
Flutter



Cross Platform



Cross Platform





Fuchsia



Fuchsia



Dart y Flutter



Dart

<https://www.dartlang.org/>

Usos de Dart



Flutter
Mobile



Web
Código que corre
en el navegador
AngularDart




Server
Aplicaciones de
lado del servidor



Ya he trabajado con Dart



Ya he trabajado con Dart




Programación Orientada a
Objetos Java, C++, etc.






Ya he trabajado con Dart



Programación Orientada a
Objetos Java, C++, etc.

Desarrollador Móvil



Android y iOS Nativo
React Native
Xamarin
Ionic


```
// Define a function.  
printInteger(int aNumber) {  
    print('The number is $aNumber.');// Print to console.  
}  
  
// This is where the app starts executing.  
main() {  
    var number = 42; // Declare and initialize a variable.  
    printInteger(number); // Call a function.  
}
```



Sintáxis

JavaScript - React

```
const Title = styled.h1`  
  color: palevioletred;  
  font-size: 1.5em;  
  text-align: center;  
`;  
;
```

```
<Wrapper>  
  <Title>Hello World, this is my first styled component!</Title>  
</Wrapper>
```

Dart

```
final Title = Text(  
  "Hola Mundo",  
  textAlign: TextAlign.center,  
  style: TextStyle(  
    fontSize: 13.0,  
    color: Color(0xFFa3a5a7)  
  ),  
);
```

Dart - JavaScript React

```
final Title = Text(  
  "Hola Mundo",  
  textAlign: TextAlign.center,  
  style: TextStyle(  
    fontSize: 13.0,  
    color: Color(0xFFa3a5a7)  
  ),  
);
```

```
const Title = styled.h1`  
  color: palevioletred;  
  font-size: 1.5em;  
  text-align: center;  
`;
```

Dart

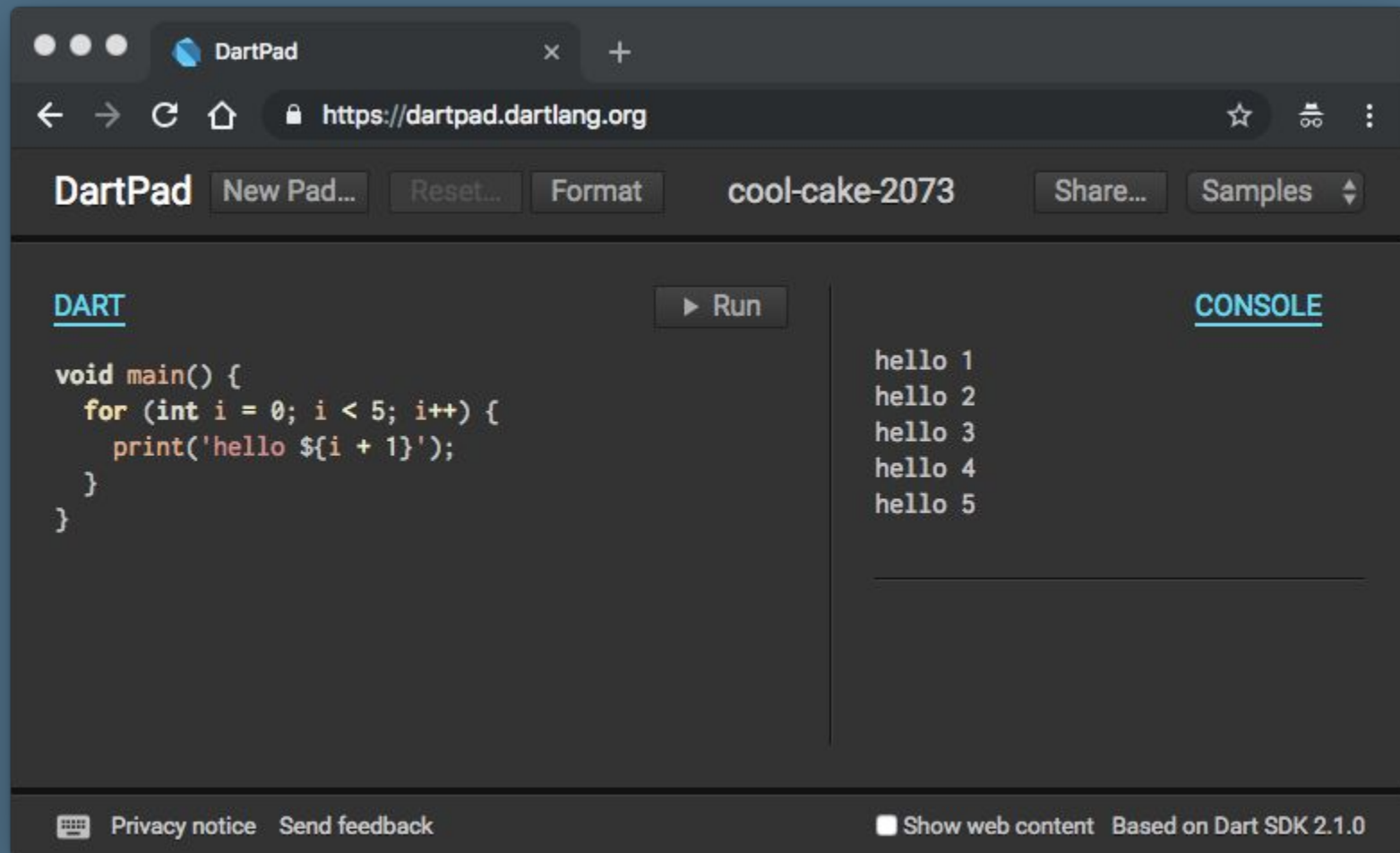
```
class HolaMundo extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    // TODO: implement build  
    return null;  
  }  
}
```

Dart - Java

```
class HolaMundo extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    // TODO: implement build  
    return null;  
  }  
}
```

```
public class MenuActivity extends AppCompatActivity {  
  
  @Override  
  protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_menu)  
  }  
}
```

<https://dartpad.dartlang.org/>



Flutter en **Android**, iOS y **Xamarin**

Android



View	Widget	
XML	Widget tree	

iOS



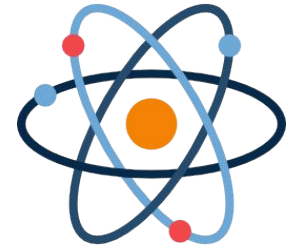
UIView	Widget	
Storyboard	Widget tree	

Xamarin



Element	Widget	
XAML	Widget tree	

React Native



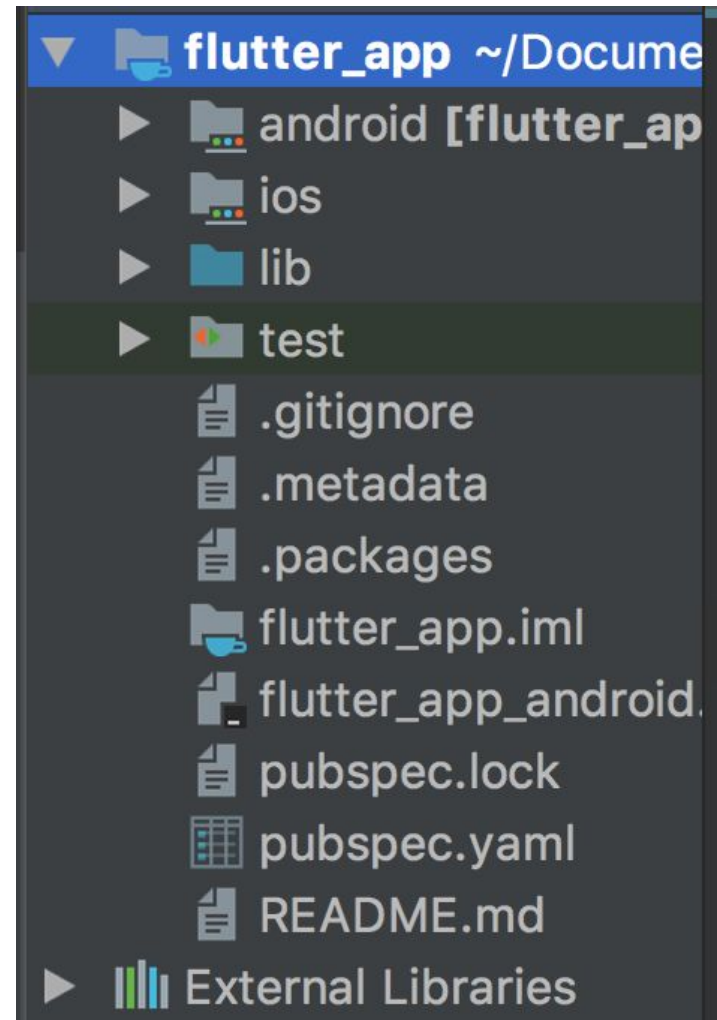
UIView	Widget	
Storyboard	Widget tree	

Apps en Flutter

<https://startflutter.com/>

Composición de un proyecto en Flutter

Proyecto Flutter



android



ios



lib



android

FlutterActivity, AndroidManifest y todos los archivos que corresponden a un proyecto Android

ios

lib

android

FlutterActivity, AndroidManifest y todos los archivos que corresponden a un proyecto Android

ios

FlutterAppDelegate, info.plist y todos los archivos que corresponden a un proyecto iOS

lib

android

FlutterActivity, AndroidManifest y todos los archivos que corresponden a un proyecto Android

ios

FlutterAppDelegate, info.plist y todos los archivos que corresponden a un proyecto iOS

lib

Aquí es donde vive la aplicación Flutter. Concentramos los archivos .dart

pubspec.yaml

Archivo de configuración escrito en
YAML

Es un formato de serialización de datos
legible por humanos inspirado en
lenguajes como XML

pubspec.yaml

```
name: newtify
version: 1.2.3
description: >-
  Have you been turned into a newt?  Would you like to be?
  This package can help. It has all of the
  newt-transmogrification functionality you have been looking
  for.
author: Natalie Weizenbaum <nweiz@google.com>
homepage: https://newtify.dartlang.org
documentation: https://docs.newtify.com
environment:
  sdk: '>=2.0.0 <3.0.0'
dependencies:
  efts: ^2.0.4
  transmogrify: ^0.4.0
dev_dependencies:
  test: '>=0.6.0 <0.12.0'
```


pubspec.yaml

```
dependencies:  
  flutter:  
    sdk: flutter  
  
  cupertino_icons: ^0.1.0  
  connectivity: ^0.3.0  
  
dev_dependencies:  
  flutter_test:  
    sdk: flutter
```

pubspec.yaml

```
main.dart widget_test.dart textFun.dart ! pubspec.yaml x
48 # https://flutter.io/assets-and-images/#from-packages
49
50 # To add custom fonts to your application, add a fonts section here,
51 # in this "flutter" section. Each entry in this list should have a
52 # "family" key with the font family name, and a "fonts" key with a
53 # list giving the asset and other descriptors for the font. For
54 # example:
55 fonts:
56   - family: Monoton
57     fonts:
58       - asset: fonts/Monoton-Regular.ttf
59   - family: Raleway
60     fonts:
61       - asset: fonts/Raleway-Black.ttf
62       - asset: fonts/Raleway-ExtraBoldItalic.ttf
63         weight: 800
64 #
65 # For details regarding fonts from package dependencies,
66 # see https://flutter.io/custom-fonts/#from-packages
67
```


Programación Declarativa en Flutter

Flutter toma su principal
inspiración en
React

**React utiliza un
estilo de
programación
declarativa**

Declarativa vs. Imperativa

Imperativa

```
// Imperative style  
b.setColor(red)  
b.clearChildren()  
ViewC c3 = new ViewC(...)  
b.add(c3)
```


Declarativa

```
// Declarative style  
return ViewB(  
  color: red,  
  child: ViewC(...),  
)
```

Imperativa

```
// Imperative style  
b.setColor(red)  
b.clearChildren()  
ViewC c3 = new ViewC(...)  
b.add(c3)
```

Declarativa

```
// Declarative style  
return ViewB(  
    color: red,  
    child: ViewC(...),  
)
```

```
// Imperative style
b.setColor(red)
b.clearChildren()
ViewC c3 = new ViewC(...)
b.add(c3)
```

ViewA a

ViewB b

ViewC c1

ViewC c2

```
// Declarative style
return ViewB(
  color: red,
  child: ViewC(...),
)
```

ViewA a

ViewB b

ViewC c1

Estructura de un programa en Flutter

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

Widgets de Material Design provistos por SDK Flutter



```
void main() => runApp(App());
```

La función runApp toma el widget y lo sirve



```
class App extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      title: 'My Flutter App',  
      home: Home(),  
    );  
  }  
}
```

Widget llamado App
Método build crea la interfaz


```
class App extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      title: 'My Flutter App',  
      home: Home(),  
    );  
  }  
}
```

El punto inicial para general una aplicación en MaterialDesign

```
class App extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      title: 'My Flutter App',  
      home: Home(),  
    );  
  }  
}
```

El título: task manager
Home: Lo que muestra la app


```
class Home extends StatefulWidget {  
  @override  
  State<StatefulWidget> createState() {  
    return _HomeState();  
  }  
}
```

```
class Home extends StatefulWidget {  
  @override  
  State<StatefulWidget> createState() {  
    return _HomeState();  
  }  
}
```

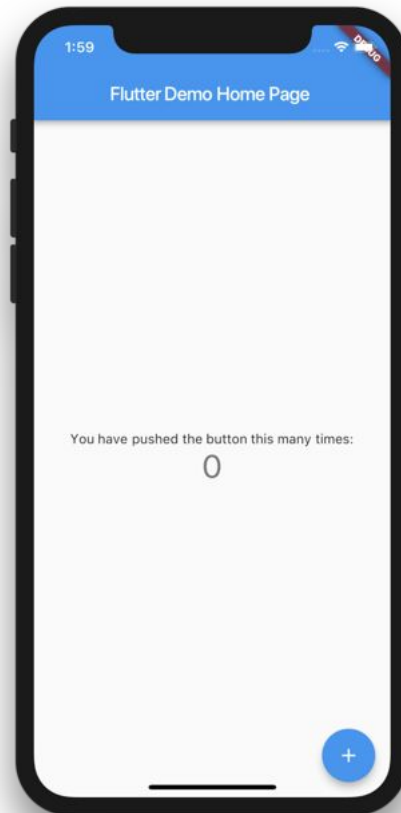
```
class _HomeState extends State<Home> {  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: Text('My Flutter App'),  
      ),  
    ),  
  },  
}
```

```
return Scaffold(  
  appBar: AppBar(  
    title: Text('Sample Code'),  
  ),  
  body: Center(  
    child: Text('You have pressed the button $_count times.'),  
  ),  
  bottomNavigationBar: BottomAppBar(  
    child: Container(height: 50.0,),  
  ),  
  floatingActionButton: FloatingActionButton(  
    onPressed: () => setState(() {  
      _count++;  
    }),  
    tooltip: 'Increment Counter',  
    child: Icon(Icons.add),  
  ),  
  floatingActionButtonLocation: FloatingActionButtonLocation.centerDocked,  
);  
}
```

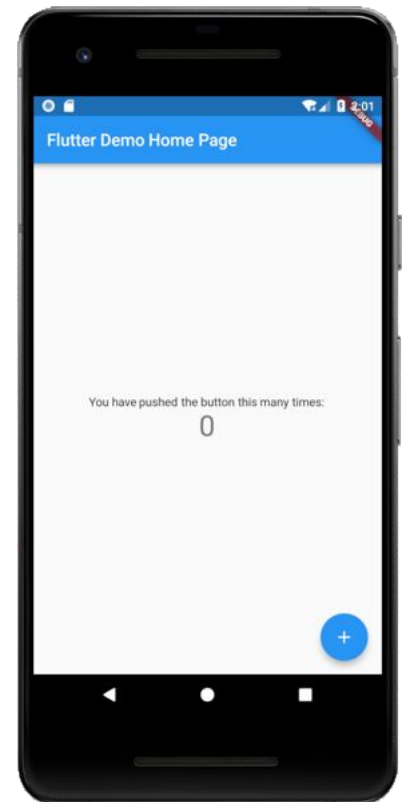
```
return Scaffold(  
  appBar: ,  
  body: ,  
  bottomNavigationBar: ,  
  floatingActionButton: ,  
  floatingActionButtonLocation: ,  
);
```



```
return Scaffold(  
  appBar: ,  
  body: ,  
  bottomNavigationBar: ,  
  floatingActionButton: ,  
  floatingActionButtonLocation: ,  
);
```



iPhone X - 11.3



Widgets básicos

“

*In Flutter, almost
everything is a widget*

”

Widgets Básicos

Texto

Row

Columna

Stack

Container

Widgets Básicos

Texto

Row

Columna

Stack

Container

Widgets Básicos

Texto

Row

Columna

Stack

Container

Widgets Básicos

Texto

Row

Columna

Stack

Container

Widgets con estado y sin estado

Un widget con estado

StatefulWidget

Usuario puede interactuar con él

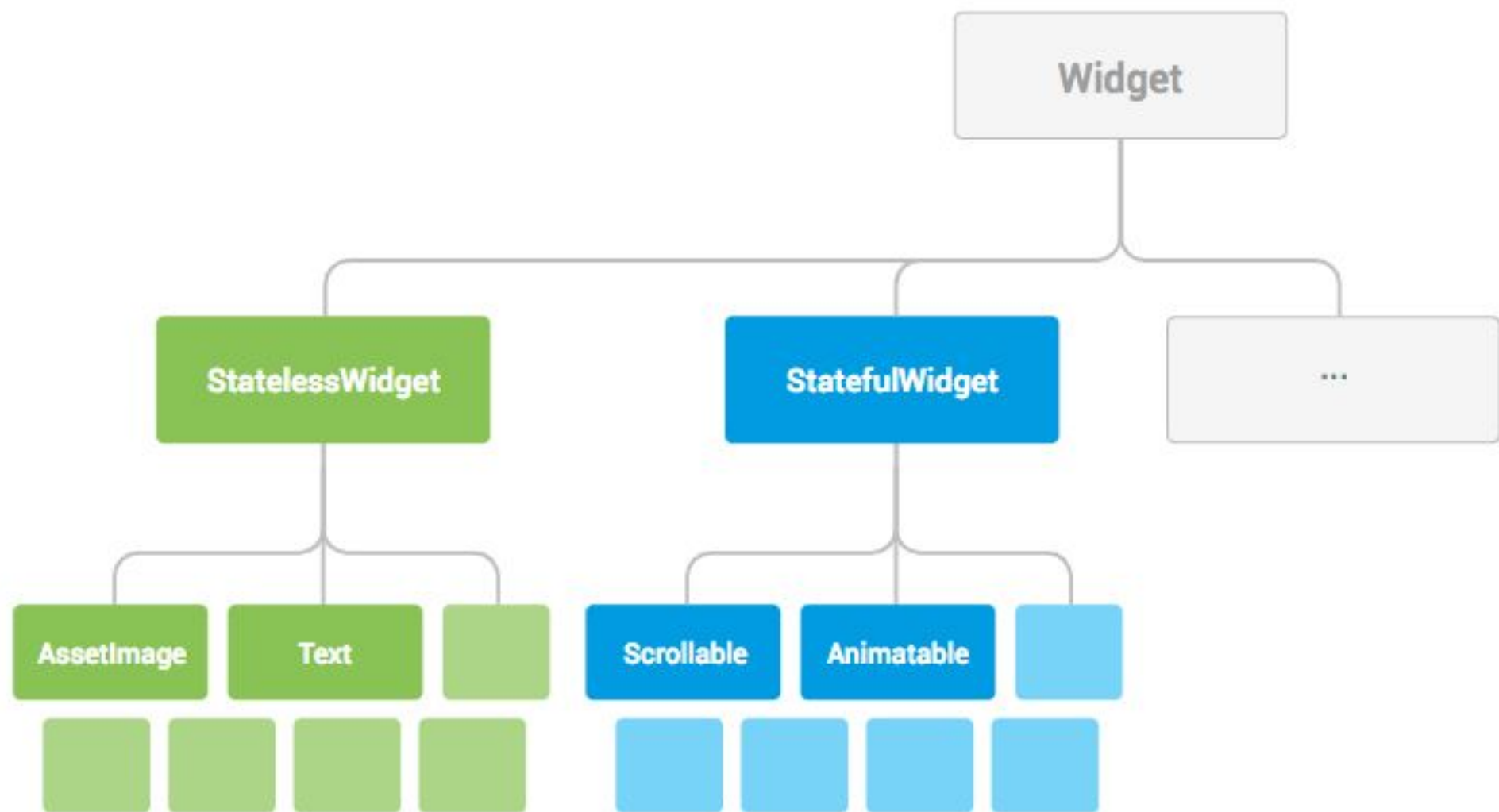
- Checkbox, Radio, Slider, Form

Un widget con estado

StatelessWidget

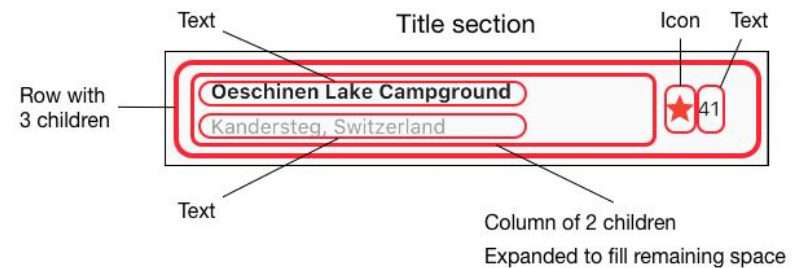
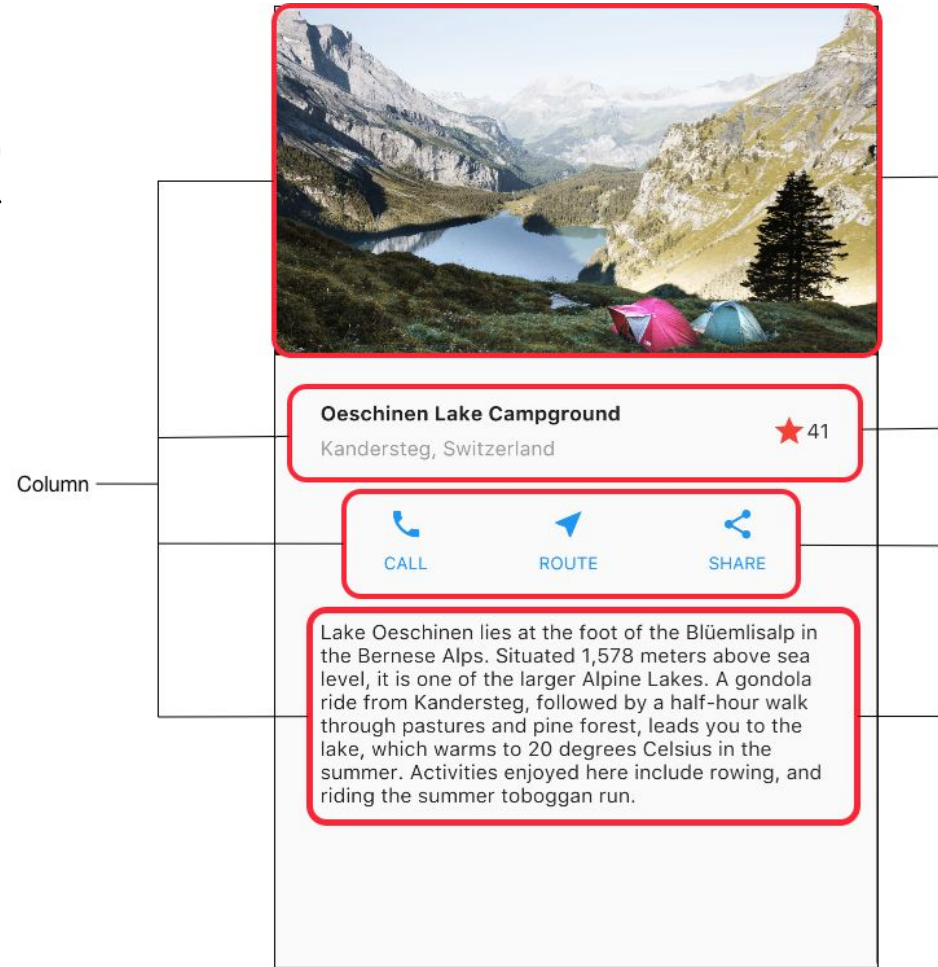
Elementos fijos que no interactúan con el usuario

- Icono
- Texto
- Contenedor con color

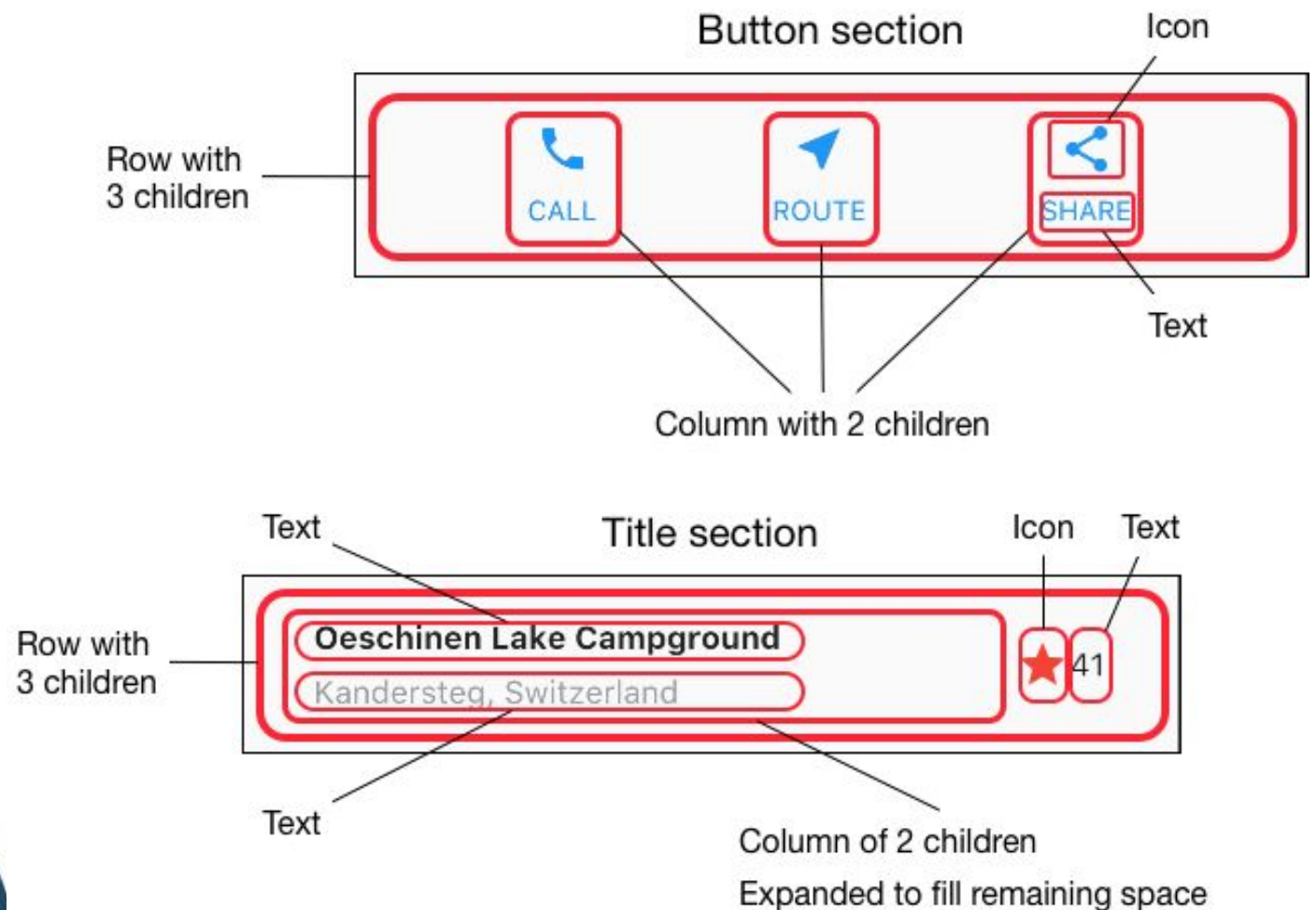


Layout

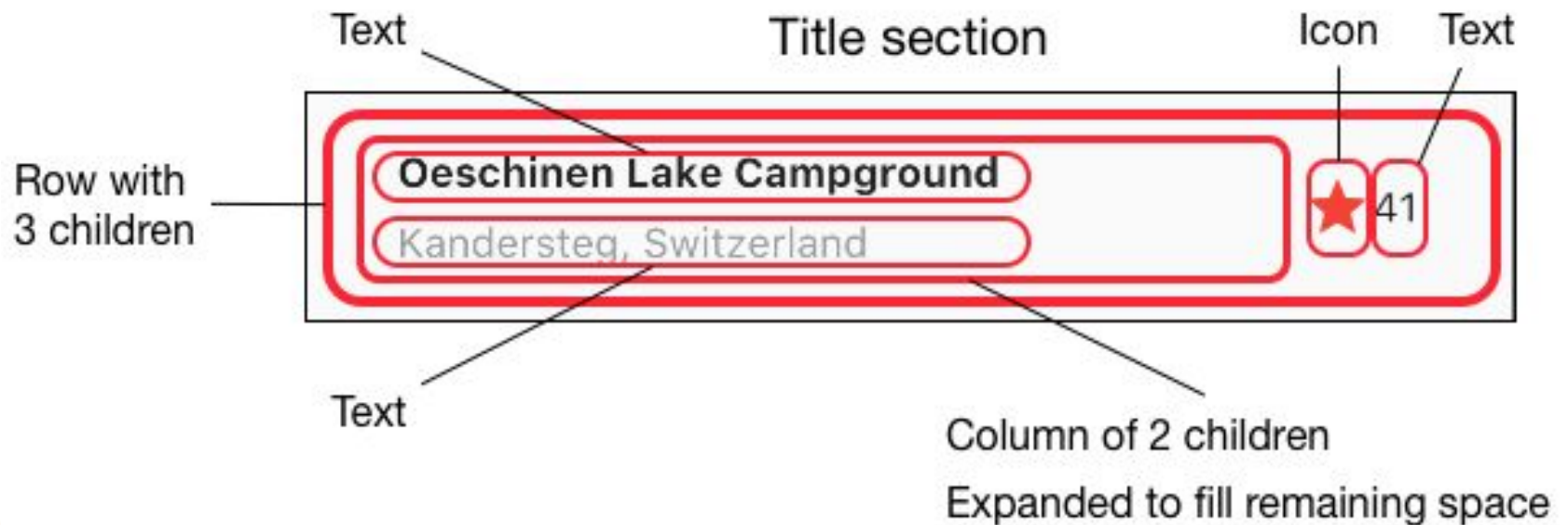
Diagrammar layout

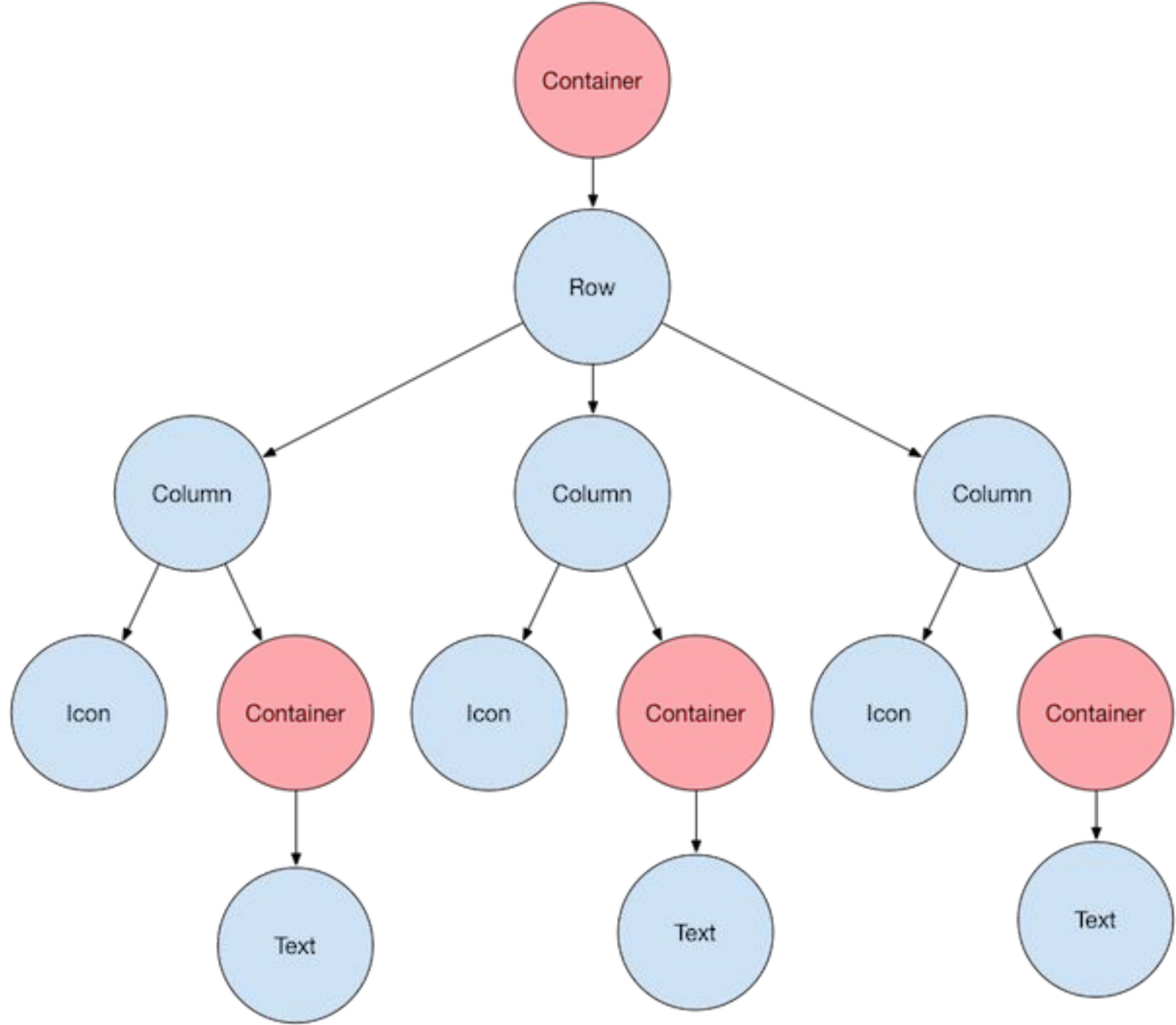


Diagrammar layout



Diagramar layout





Clase StatefulWidget

```
class PlatziTrips extends StatefulWidget {  
  @override  
  State<StatefulWidget> createState() {  
    // TODO: implement createState  
    return _PlatziTrips();  
  }  
}
```



```
class PlatziTrips extends StatefulWidget {  
  @override  
  State<StatefulWidget> createState() {  
    // TODO: implement createState  
    return _PlatziTrips();  
  }  
}
```

```
class _PlatziTrips extends State<PlatziTrips> {
```



```
@override
Widget build(BuildContext context) {
  // TODO: implement build

  return Scaffold(
    body: widgetsChildren[indexTap],
    bottomNavigationBar: Theme(
      data: Theme.of(context).copyWith(
        canvasColor: Colors.white,
        primaryColor: Colors.purple
      ),
    child: BottomNavigationBar(
      onTap: onTapTapped,
      currentIndex: indexTap,
```

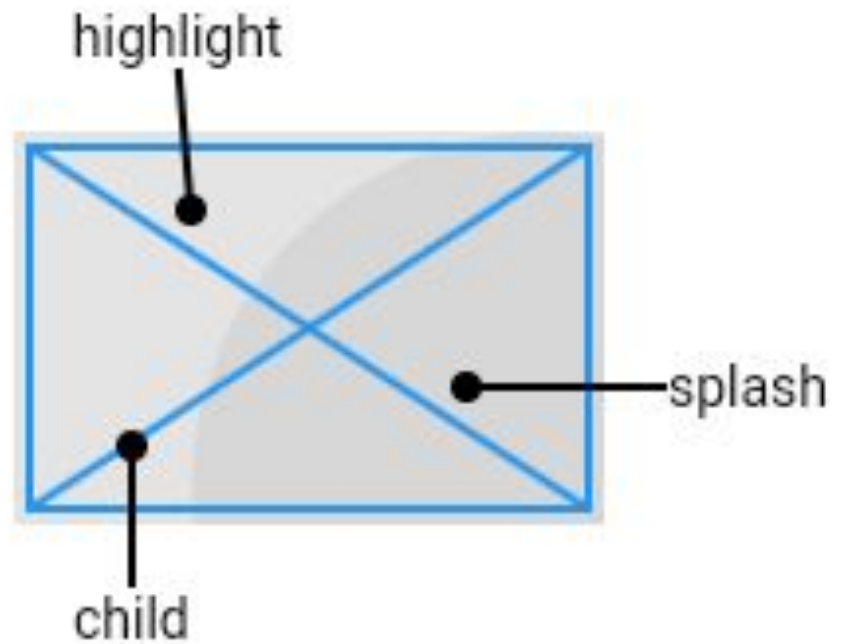
```
class _PlatziTrips extends State<PlatziTrips> {
```

```
    void onTapTapped(int index){
```

```
        setState(() {  
            indexTap = index;  
        });
```

```
    }
```

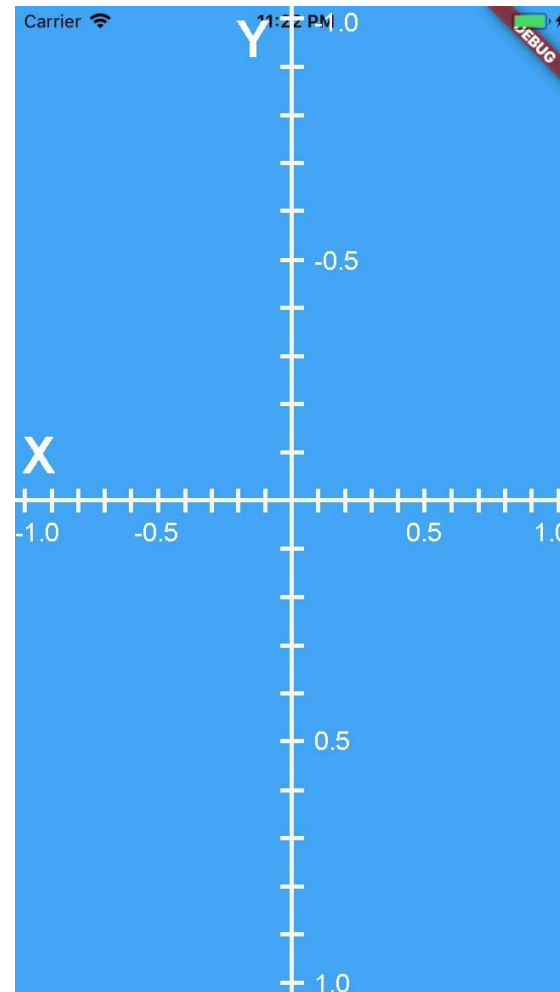
InkWell

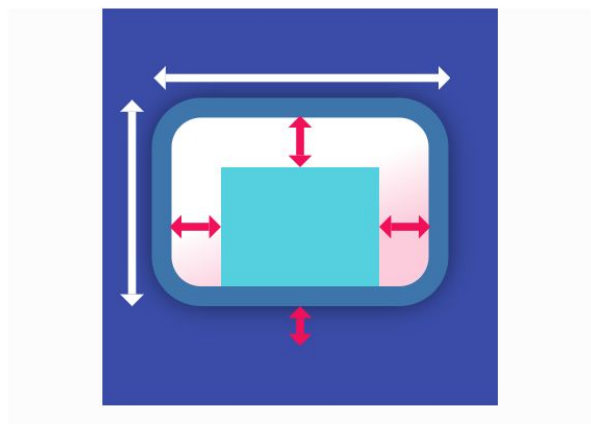


Area que responde a un touch

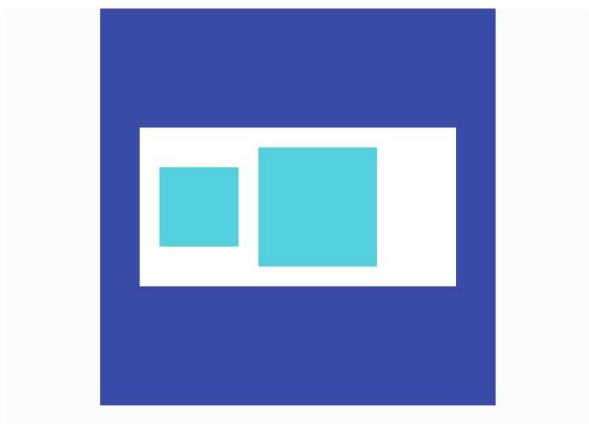
Widgets Básicos

Container

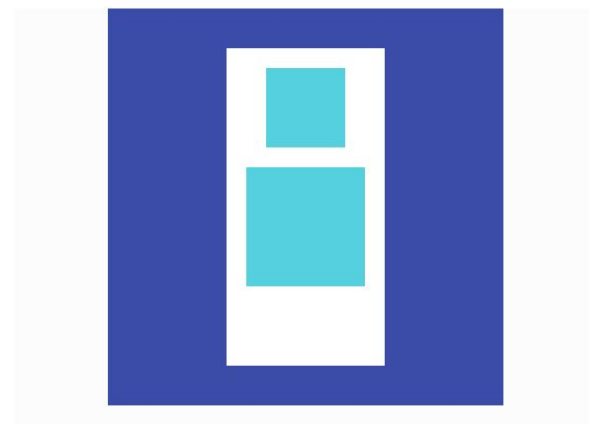




Container



Row



Column