Flutter - Aula 1

Exercícios 1 a 5

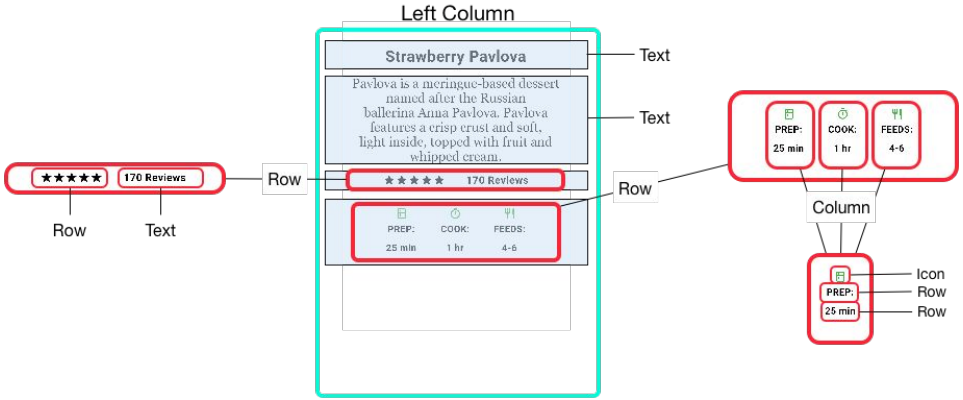
Fernando Duwe

v 1.0

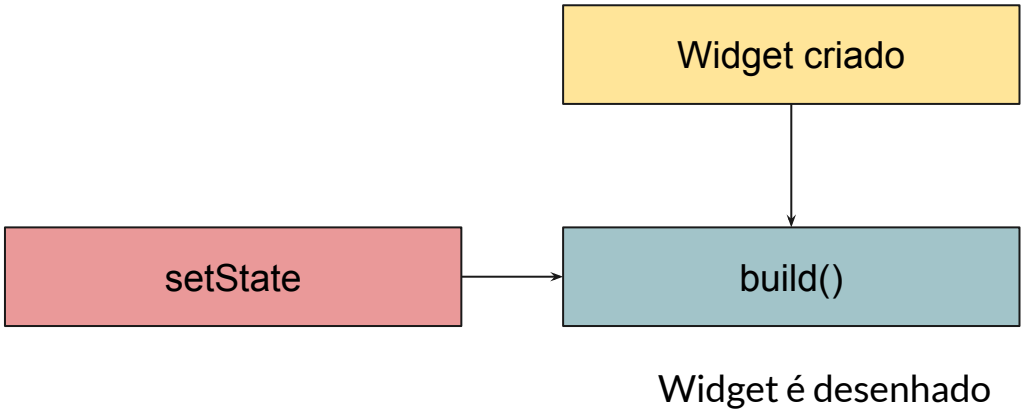
- Vamos seguir o passo a passo no site do Flutter;

- Android Studio; Ver mais
- Dependências; Ver mais
- Fluter CLI - Flutter Command-Line Tool; Ver mais

- Componentes;
- Como um conjunto de Lego;
- Widgets de Layout: Servem para determinar a estrutura de layout;
- Widgets de Interface;



- StatefulWidget: Possui controle de estados;
- StatelessWidget: Não possui controle de estados;
- setState;



MaterialApp

Ver mais

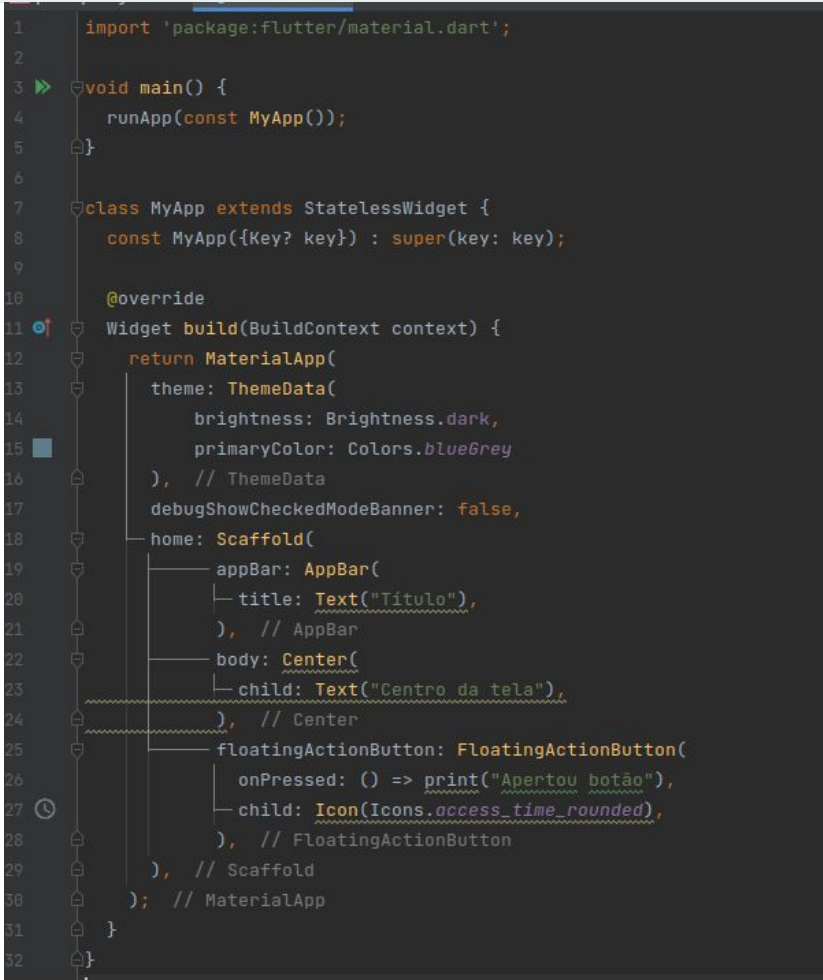
- Indica que o App utiliza Material Design
- Primeiro Widget chamado
- Configure temas

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

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData(
          brightness: Brightness.dark,
          primaryColor: Colors.blueGrey
      ),  // ThemeData
      debugShowCheckedModeBanner: false,
      home: Scaffold(
        appBar: AppBar(
          title: const Text('MaterialApp Theme'),
        ),  // AppBar
        body: Text("La Belle de Jour")
      ),  // Scaffold
    );  // MaterialApp
  }
}
```

Scaffold

Ver mais

- Estrutura padrão de layout
- appBar
- body
- floatingActionButton



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

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData(
          brightness: Brightness.dark,
          primaryColor: Colors.blueGrey
      ),  // ThemeData
      debugShowCheckedModeBanner: false,
      home: Scaffold(
          appBar: AppBar(
            title: Text("Título"),
          ),  // AppBar
          body: Center(
            child: Text("Centro da tela"),
          ),  // Center
          floatingActionButton: FloatingActionButton(
            onPressed: () => print("Apertou botão"),
            child: Icon(Icons.access_time_rounded),
          ),  // FloatingActionButton
      ),  // Scaffold
    );  // MaterialApp
  }
}
```

Ver mais

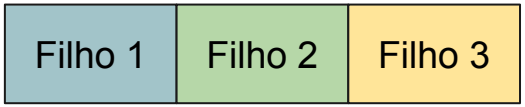
- Componente básico de layout
- padding
- margin

```
child: Container(
  child: Text(":)", style: TextStyle(color: Colors.black, fontSize: 32)),
  padding: EdgeInsets.all(32),
  decoration: BoxDecoration(
    shape: BoxShape.circle,
    color: Colors.amberAccent,
  ),  // BoxDecoration
```

Row

Ver mais

- Mostra os widgets filhos horizontalmente




```
body: Row(
  children: [
    Text("Texto 1"),
    Text("Texto 2"),
    Text("Texto 3")
  ],
), // Row
```

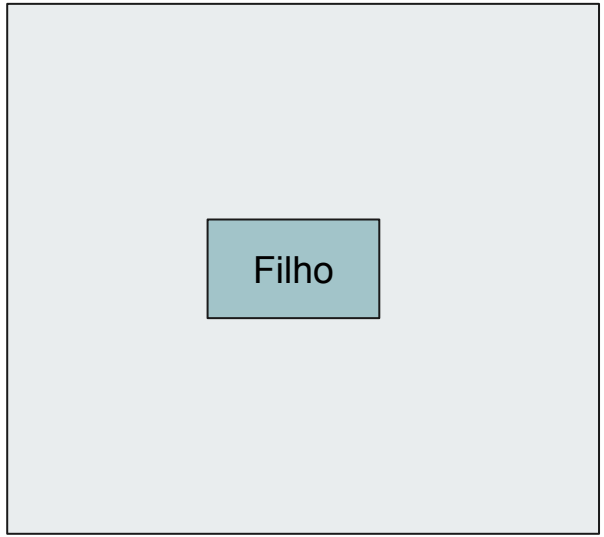
Column

Ver mais

- Mostra os widgets filhos verticalmente


```
body: Column(
    children: [
        Text("Texto 1"),
        Text("Texto 2"),
        Text("Texto 3")
    ],
), // Column
```

- Centraliza o widget filho na tela

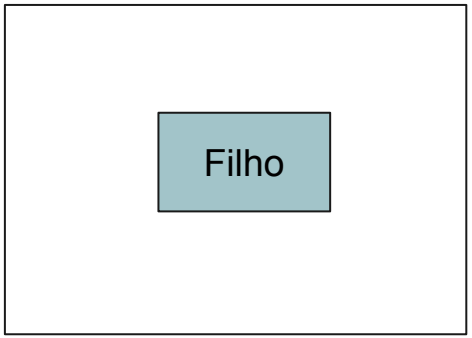



```
body: Center(
  child: Text("Texto centralizado"),
), // Center
```

Padding

Ver mais

- Adiciona um espaço ao redor do widget filho
- EdgeInsets




```
children: [
  Padding(
    padding: const EdgeInsets.fromLTRB(0, 0, 12, 0),
    child: Container(
      decoration: BoxDecoration(color: Colors.blue),
      height: 32,
      width: 32,
    ),  // Container
  ),  // Padding
```

- Texto

```
dy: Text("Flutter é legal :)",
  style: TextStyle(
      color: Colors.amberAccent,
      backgroundColor: Colors.white,
      fontWeight: FontWeight.bold,
      fontFamily: "Courier New",
      fontSize: 48)),  // TextStyle, Text
```

Exercício 1

Ver mais

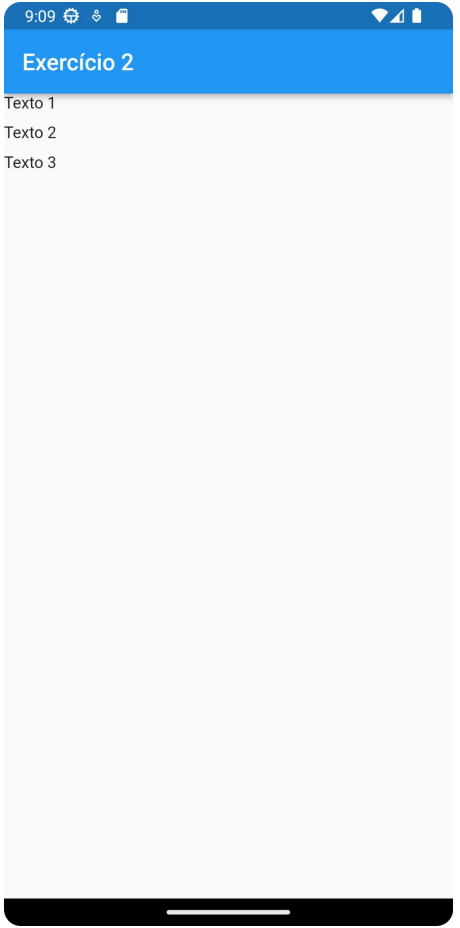
Crie uma tela, utilizando a estrutura padrão de layout (Scaffold).

Nesta tela, deve existir um texto centralizado (o que está escrito fica a seu critério).

Esse texto deve estar em negrito, com um tamanho de fonte igual a 24.

Exercício 2

Ver mais



Crie uma tela, utilizando a estrutura padrão de layout (Scaffold).

Nesta tela, deverão ser exibidos três textos, sendo exibidos um em cima do outro.

Adicione um espaço no texto do meio, para que os três não fiquem muito colados.

Ver mais

- TextButton
- ElevatedButton
- OutlinedButton

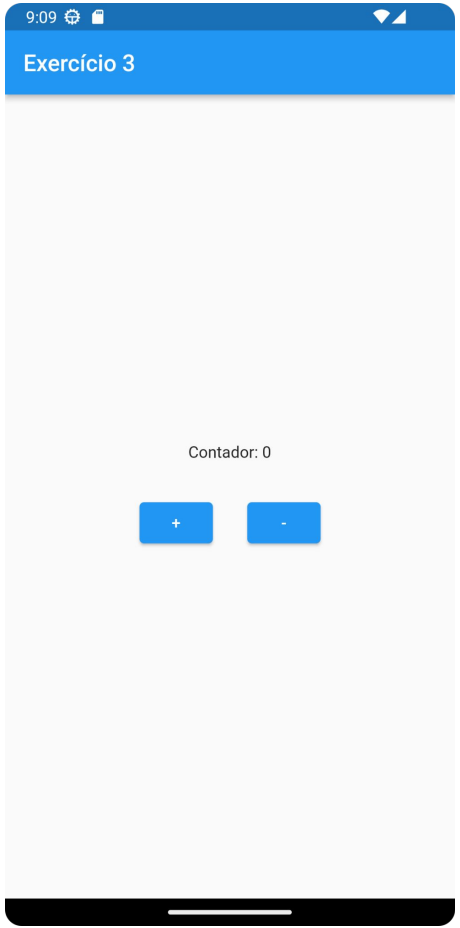
```
TextButton(
    onPressed: () {
      print("Cliquei em um TextButton");
    },
    child: Text("TextButton")),  // TextButton

ElevatedButton(
    onPressed: () => print("Cliquei em um ElevatedButon"),
    child: Text("ElevatedButton")),  // ElevatedButton

OutlinedButton(
    onPressed: clicouOutlineButton, child: Text("OutlineButton"))  // OutlinedButton
```

Exercício 3

Ver mais

Crie uma tela com um contador e dois botões.

- Um botão (+) irá incrementar o contador;
- Um botão (-) irá decrementar o contador;
- Exiba o contador em tela;



- Caixa de texto
- Controllers

Exercício 4

Ver mais

Vamos finalmente criar a nossa calculadora...

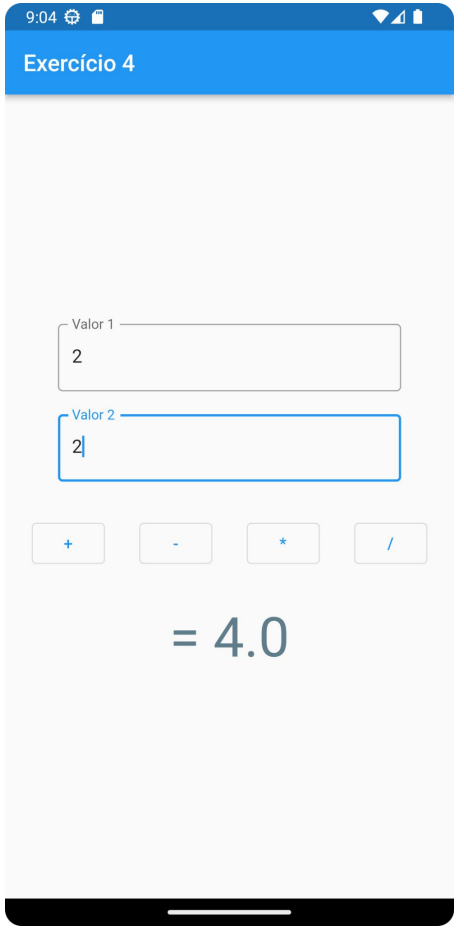
Ela deve possuir dois TextFields em campo, com os valores a calcular e quatro botões:

+

-

*

/



- Exiba imagens no seu app
- JPEG, PNG, GIF, WebP, BMP, WBMP
- Imagens da internet, com o NetworkImage
- Imagens dos seus assets, com AssetImage

```
children: [

  Image(
    image: AssetImage("assets/happy_santaclaus.png"),
  ),  // Image

  Image(
    image: NetworkImage("https://www.iconbolt.com/preview/facebook/remix-icon-fill/flutter.svg"),
  ),  // Image

],
```

Exercício 5

Ver mais

Vamos fazer um app de Natal!
Para isso, crie um app que exiba uma foto do papai noel,
e uma foto de uma árvore de natal.