

# **DCC917A – TÓPICOS ESPECIAIS III: DESENVOLVIMENTO DE APLICATIVOS MÓVEIS**

**AULA 05**

**Carlos Bruno Oliveira Lopes**

*Engenheiro de Computação*

*Mestre em Ciência da Computação*

# Flutter – básico

## Layout

- Os widgets de layout ajudam a organizar a interface de usuário e a estrutura da aplicação de várias maneiras.
  - Elas permitem construir o esqueleto do aplicativo.
  - Alguns exemplos de recursos de layout:
    - MaterialApp, Scaffold, Center, Row, Column, Expanded, Align e Text
- O layout no Flutter se resume principalmente a um estrutura da grade,
  - Linhas (Row)
  - Colunas (Column)
    - Elas podem ter mais filhos com disposições em horizontal ou vertical (novas linhas e colunas)

# Flutter – básico (Layout)

Código exemplo:

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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: "Flutter Playground",
      home: Scaffold(
        body: Center(
          child: Row(children: [
            Text("Child1"),
            Text("Child2"),
            Text("Child3")
          ])));
  }
}
```

# Flutter – básico (Layout)

## Aplicativo: layout básico

- **material.dart**: permite o uso de widgets do estilo Material;
- **main()**: instancia a classe `MyApp` e passa a instância para a função `runApp()`;
- **MyApp**: é um `StatelessWidget` (a App não precisa de nenhum tipo de estado);
- **build**: é um método requerido que produz um único widget de tipo `MaterialApp`;
  - Ela implementa boa parte da estrutura da app.
- **title**: é um propriedade da widget, ou seja, uma string usada pelo dispositivo para identificar o aplicativo para o usuário.

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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: "Flutter Playground",
      home: Scaffold(
        body: Center(
          child: Row(children: [
            Text("Child1"),
            Text("Child2"),
            Text("Child3")
          ]),
        ),
      ),
    );
  }
}
```

# Flutter – básico (Layout)

## Aplicativo: layout básico

### – Outras propriedades:

- **textAlign**: permite determinar como o texto deve ser alinhado horizontalmente;
- **textScaleFactor**: informa ao Flutter o número de pixels de fonte para cada unidade de pixel lógica e dessa forma dimensiona o texto;
- **mainAxisAlignment**: centraliza as widgets filhas;
- **Expanded**: faz o filho preencher todo o espaço disponível;
- **Align**: permite alinhar o filho no seu interior, e opcionalmente permite ser redimensionado de acordo com o tamanho do filho.
- **Column**: é similar ao `Row` exceto que é na direção vertical.

# Flutter – básico (Layout)

Aplicativo: layout básico (Mais exemplos)

– **Container**: permite organizar a UI.

- Podemos combinar várias widgets;

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

void main() {
  runApp(MaterialApp(
    title: "Frases do dia",
    home: Container(
      color: Colors.white,
    ),
  ));
}
```

# Flutter – básico (Layout)

## Aplicativo: layout básico (Mais exemplos)

– **children**: permite instanciar um lista de widgets

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

```
void main() {  
  runApp(MaterialApp(  
    title: "Frases do dia",  
    home: Column( //cria colunas  
      children: <Widget>[  
        Text("T1"),  
        Text("T2"),  
        Text("T3")  
      ],  
    ),  
  ));  
}
```

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

```
void main() {  
  runApp(MaterialApp(  
    title: "Frases do dia",  
    home: Row( //cria colunas  
      children: <Widget>[  
        Text("T1"),  
        Text("T2"),  
        Text("T3")  
      ],  
    ),  
  ));  
}
```

# Flutter – básico (Layout)

## Aplicativo: layout básico (Mais exemplos)

### – **style**: permite aplicar formatações ou estilos

- **TextStyle**: permite formatar o texto;
- **fontSize**: permite definir o tamanho da fonte;
- **fontStyle**: permite definir o estilo da fonte;
- **fontWeight**: permite definir o “peso do texto”;
- **letterSpacing**: permite definir o espaçamento entre as letras;
- **wordSpacing**: permite definir o espaçamento entre as palavras;
- **decoration**: aplica decorações no texto;
- **decorationColor**: aplica a cor na decoração;
- **decorationStyle**: aplica um estilo da decoração;
- **color**: aplica um cor no texto.

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

void main() {
  runApp(MaterialApp(
    title: "Frases do dia",
    home: Container(
      color: Colors.white,
      child: Column(
        children: <Widget>[
          Text(
            "Aula de layout e estilo.",
            style: TextStyle(
              fontSize: 25,
              fontStyle: FontStyle.italic,
              fontWeight: FontWeight.w100,
              letterSpacing: 10, // espaçamento entre letras
              wordSpacing: 5, //espaçamento entre palavras
              decoration: TextDecoration.underline,
              decorationColor: Colors.greenAccent,
              decorationStyle: TextDecorationStyle.dotted,
              color: Colors.lightBlue,
            ),
          ),
        ],
      ),
    ),
  ));
}
```



# Flutter – básico (Layout)

## Aplicativo: layout básico (Mais exemplos)

### – FlatButton: tipo de botão simples

- **onPressed**: recebe um função com parâmetro que implementa o que botão deve fazer;
- **child**: defini a widget filha (no exemplo Text);

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

void main() {
  runApp(MaterialApp(
    title: "Frases do dia",
    home: Container(
      color: Colors.white,
      child: Column(
        children: <Widget>[
          FlatButton(
            onPressed: () {
              print("Botão pressionado!");
            },
            color: Colors.lightBlueAccent,
            child: Text(
              "Clique aqui",
              style: TextStyle(
                fontSize: 20,
                color: Colors.black,
                decoration:
TextDecoration.none
              ),
            ),
          ),
        ],
      ),
    ),
  ));
}
```

# Flutter – básico (Layout)

## Aplicativo: layout básico (Mais exemplos)

- **padding**: permite inserir espaçamento interno nas bordas;
- **margin**: permite inserir espaçamento externo nas bordas;
- **BoxDecoration**: permite formata a borda de um contêiner;

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

void main() {
  runApp(MaterialApp(
    // debugShowCheckedModeBanner: false,
    title: "Frases do dia",
    home: Container(
      // color: Colors.amber,
      padding: EdgeInsets.all(10),
      // padding: EdgeInsets.fromLTRB(10,30,10,30),
      margin: EdgeInsets.all(20),
      // margin: EdgeInsets.fromLTRB(10, 5, 10, 5),
      // margin: EdgeInsets.only(top: 20),
      decoration:
        BoxDecoration(
          border: Border.all(
            width: 1,
            color: Colors.green
          )
        ),
      child: Row(
        children: <Widget>[
          Text("t1"),
          Padding(
            padding: EdgeInsets.all(30),
            child: Text("t2"),
          ),
          Text("t3")
        ],
      ),
    ),
  ));
}
```

# Flutter – básico (Layout)

## Aplicativo: layout básico (Mais exemplos)

- **mainAxisAlignment**: alinhamento do elementos no eixo principal;
  - **start**: alinha no inicio;
  - **center**: alinha no centro;
  - **end**: alinha no final;
  - **space...**: distribui o elementos preenchendo com espaços
- **crossAxisAlignment**: alinhamento no eixo que cruza o principal;
  - **start**: alinha no inicio;
  - **center**: alinha no centro;
  - **end**: alinha no final;

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

void main() {
  runApp(MaterialApp(
    // debugShowCheckedModeBanner: false,
    title: "Frases do dia",
    home: Container(
      padding: EdgeInsets.only(top: 40),
      margin: EdgeInsets.all(10),
      decoration: BoxDecoration(
        border: Border.all(
          width: 3,
          color: Colors.white
        )
      ),
      child: Row(
        mainAxisAlignment:
MainAxisAlignment.spaceEvenly,
        crossAxisAlignment: CrossAxisAlignment.center,
        children: <Widget>[
          Text("t1"),
          Text("t2"),
          Text("t3")
        ],
      ),
    ),
  ));
}
```

# Flutter – básico (Layout)

```
# To add assets to your application, add an assets section, like this:  
# assets:  
#   - images/a_dot_burr.jpeg  
#   - images/a_dot_ham.jpeg  
assets:  
  - images/mesa.jpg  
  - images/parque.jpg
```

## Aplicativo: layout básico (Mais exemplos)

### – Adicionando imagens:

- Crie um pasta no projeto principal (como “**images**”) e adicione as imagens nessa pasta.
- Acesse o arquivo “**pubspec.yaml**” para realizarmos umas configurações:
  - Dentro do arquivo procure por “Flutter:” onde terá um comentário “To add ... assets:”, então adicione a configuração de recursos que estão sendo utilizado como exemplificado:
    - » assets:
    - » - images/mesa.jpg
    - » - images/parque.jpg
  - Por fim, clique “Pub get” para aplicar as mudanças.

Pub get

# Flutter – básico (Layout)

## Aplicativo: layout básico (Mais exemplos)

- **Image.asset**: permite inserir uma imagem passando o path;
- **fit**: permite configura como a imagem irá se ajustar ao contêiner;
  - **contain**: ajusta a imagem o quanto pode dentro do contêiner ao qual foi inserida;
  - **cover**: cobri toda área disponível contendo a imagem se necessário;
  - **fill**: preenche toda área disponível, mas pode distorce a imagem;
  - **fitHeight**: preenche segundo a altura;
  - **fitWidth**: preenche segundo a largura do contêiner;
  - **none**: não ajusta a imagem;
  - **scaleDown**: ajusta a imagem sem distorce de acordo com espaço disponível;

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

void main() {
  runApp(MaterialApp(
    // debugShowCheckedModeBanner: false,
    title: "Frases do dia",
    home: Container(
      padding: EdgeInsets.only(top: 40),
      margin: EdgeInsets.all(10),
      decoration: BoxDecoration(
        border: Border.all(
          width: 3,
          color: Colors.white
        )
      ),
      child: Image.asset(
        "images/mesa.jpg",
        // contain, cover, fill, fitHeight
        // fitWidth, none, scaleDown
        fit: BoxFit.scaleDown,
      )
    ),
  ));
}
```

# Flutter – básico (Layout)

## Aplicativo: layout básico (Mais exemplos)

- **Scaffold**: estrutura de layout que implementa o visual básico do material designer;
  - Fornece uma barra superior, uma área de conteúdo e uma barra inferior.
  - Na estrutura há uma formatação predefinida.
- **AppBar**: implementa a barra superior;
- **body**: implementa o conteúdo principal (entre barra superior e inferior);
- **bottomNavigationBar**: implementa a barra inferior;

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

void main() {
  runApp(MaterialApp(
    debugShowCheckedModeBanner: false,
    title: "Frases do dia",
    home: Scaffold(
      appBar: AppBar(
        title: Text("Instagram"),
        backgroundColor: Colors.blueGrey,
      ),
      body: Padding(
        padding: EdgeInsets.all(15),
        child: Text("Conteúdo principal"),
      ),
      bottomNavigationBar: BottomAppBar(
        color: Colors.lightBlueAccent,
        child: Padding(
          padding: EdgeInsets.all(15),
          child: Row(
            mainAxisAlignment: MainAxisAlignment.spaceEvenly,
            children: [
              Text(
                "Texto1",
                style: TextStyle(fontWeight: FontWeight.w900)
              ),
              Text(
                "Texto 2",
                style: TextStyle(fontWeight: FontWeight.w900)
              ),
            ],
          ),
        ),
      ),
    ),
  ));
}
```

# Flutter – básico (Layout)

## Aplicativo: layout básico (Mais exemplos)

- **Stateless**: não atualiza a tela ou widget quando detectar uma mudança de estado (renderiza);
  - Widgets que não podem ser alterados;

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

void main() {
  runApp(MaterialApp(
    debugShowCheckedModeBanner: false,
    title: "Frases do dia",
    home: HomeStateless(),
  ));
}
```

```
class HomeStateless extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    var _titulo = "Instagram";
    return Scaffold(
      appBar: AppBar(
        title: Text(_titulo),
        backgroundColor: Colors.blueGrey,
      ),
      body: Padding(
        padding: EdgeInsets.all(15),
        child: Text("Contéudo principal"),
      ),
      bottomNavigationBar: BottomAppBar(
        color: Colors.lightBlueAccent,
        child: Padding(
          padding: EdgeInsets.all(15),
          child: Row(
            mainAxisAlignment: MainAxisAlignment.spaceEvenly,
            children: [
              Text(
                "Texto1",
                style: TextStyle(fontWeight:
FontWeight.w900)
              ),
              Text(
                "Texto 2",
                style: TextStyle(fontWeight:
FontWeight.w900)
              ),
            ],
          ),
        ),
      ),
    );
  }
}
```

# Flutter – básico (Layout)

## Aplicativo: layout básico (Mais exemplos)

- **Stateful**: atualiza a tela ou widgets quando detectar uma mudança de estado (renderiza);
  - Widgets que podem ser alterados;

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

void main() {
  runApp(MaterialApp (
    debugShowCheckedModeBanner: false,
    title: "Frases do dia",
    home: HomeStateful (),
  ));
}

class HomeStateful extends StatefulWidget {
  @override
  HomeStateful_State createState() =>
  HomeStateful_State ();
}
```

UFRR - Ciência da Computação

```
class HomeStateful_State extends State<HomeStateful>
{
  var _texto = "App Stateful";
  var _ctrl = 0;
  @override
  Widget build(BuildContext context) {
    var _titulo = "Instagram";
    print("Build foi chamado!");
    return Scaffold (
      appBar: AppBar (
        title: Text(_titulo),
        backgroundColor: Colors. blueGrey,
      ),
      body: Container (
        child: Column (
          children: [
            RaisedButton (
              onPressed: () {
                setState() { //atualiza UI
                  if (_ctrl == 0){
                    _texto = "App Flutter";
                    _ctrl = 1;
                  } else {
                    _texto = "App Stateful";
                    _ctrl = 0;
                  }
                }
              },
              color: Colors. lightBlueAccent,
              child: Text("Clique aqui"),
            ),
            Text("Nome: $_texto")
          ],
        ),
      ),
    );
  }
}
```

16



# Flutter – básico (Exercícios)

1. Tente criar a App com o designer demonstrado na figura ao lado. A sua App deve exibir frases aleatórias quando for clicado no botão “Nova Frase”. Gere no máximo 5 frases diferentes. Dica: use uma lista para armazenar as frases.

