

O MATERIAL DESIGN

Material Design é uma linguagem de design desenvolvida pela Google. Seu intuito é criar um padrão de design para aplicações Android, fazendo assim com que todas as aplicações Android, sejam elas nativas do S.O, ou sejam elas desenvolvidas por outros desenvolvedores e baixadas através da loja de aplicativos da Google, o Google Play.

Assim, toda e qualquer aplicação irá conter um padrão de *design*, e será mais fácil para o usuário utilizar essas aplicações, pois já estará acostumado com a posição dos elementos e suas funções na *interface*.

ESQUELETO DA APLICAÇÃO MATERIAL DESIGN

Mas antes de iniciarmos nossos estudos sobre os Widgets Material Design, tenha em mente que sempre utilizaremos o seguinte esqueleto.

Esse esqueleto contém a estrutura base de uma aplicação Material Design no Flutter.

O objeto **MaterialApp** é responsável por informar ao Flutter que estamos iniciando uma aplicação *Material Design*. É nela que informamos nosso tema, ou seja, o estilo padrão dos componentes, como cores, estilos e tamanho de fonte, assim como outras propriedades relacionadas à aparência da aplicação.

Também é nele que informamos o título da aplicação, assim como o sistema de rotas e a página inicial da nossa aplicação.

```

1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(MaterialApp(
5     title: "Exemplo Flutter",
6     theme: ThemeData(
7       primarySwatch: Colors.green,
8     ),
9     home: Home(),
10  ));
11}
12
13 class Home extends StatefulWidget {
14   @override
15   _HomeState createState() => _HomeState();
16}
17
18 class _HomeState extends State<Home> {
19   @override
20   Widget build(BuildContext context) {
21     return Container(
22       child: , // Aqui colocaremos nossos Widgets
23     );
24   }
25}
26

```

Figura 1 Esqueleto da aplicação Material Design Fonte: Própria do autor

É importante ressaltar que em nossos exemplos utilizaremos nosso esqueleto. Assim como utilização resumida das propriedades, assim resultando em menos código em cada exemplo prático.

TEXTBUTTON

O *TextButton* é um *Widget* utilizado para criação de botões simples. Esse *Widget* apresenta um botão contendo apenas um texto simples, sem fundo e com efeito de wave (Onda).

Esse *Widget* possui algumas propriedades importantes que devem ser conhecidas por qualquer desenvolvedor Flutter. Vejamos quais são elas.

Propriedade child

A propriedade *child* é uma das principais propriedades do *Widget TextButton*. Como o próprio nome já sugere, define quem será o *Widget* filho deste. Normalmente utilizamos o *Widget Text*.

A screenshot of a code editor with a dark background and light-colored text. The code is written in Dart and defines a Scaffold widget. It has a ScaffoldAppBar with a title 'Aplicação Exemplo' and a body containing a TextButton with the text 'Um simples botão'. The code is as follows:

```
1 Scaffold(  
2   appBar: AppBar(  
3     title: Text("Aplicação Exemplo"),  
4   ),  
5   body: TextButton(  
6     child: Text("Um simples botão"),  
7   ),  
8 )
```

Propriedade onPressed

A propriedade *onPressed*, como o próprio nome já sugere, é a propriedade que define a ação que deverá ser executada assim que esse *Widget* for pressionado.

Essa propriedade deverá receber um objeto *Function*, esse contando a lógica a ser executada no momento do clique sobre o *Widget*.

```
1 Scaffold(  
2   appBar: AppBar(  
3     title: Text("Aplicação Exemplo"),  
4   ),  
5   body: TextButton(  
6     child: Text("Um simples botão"),  
7     onPressed: () {},  
8   ),  
9 )
```

Propriedade style

A propriedade *style*, como o próprio nome já sugere, define o estilo do *Widget*, permitindo a adição de cor de fundo, cor do texto e muito mais.

Essa propriedade recebe um objeto *ButtonStyle*. Normalmente utilizamos a função *styleFrom*, a qual retorna um objeto *ButtonStyle*.

Propriedade primary

A propriedade *primary* define a cor do texto do botão. Essa propriedade recebe um objeto *Color*.

```
1 Scaffold(  
2   appBar: AppBar(  
3     title: Text("Aplicação Exemplo"),  
4   ),  
5   body: TextButton(  
6     child: Text("Um simples botão"),  
7     onPressed: () {},  
8     style: TextButton.styleFrom(  
9       primary: Colors.green,  
10    ),  
11  ),  
12 )
```

Propriedade background

A propriedade *background* define a cor de fundo do botão. Essa propriedade recebe um objeto *Color*.



```
1 Scaffold(  
2   appBar: AppBar(  
3     title: Text("Aplicação Exemplo"),  
4   ),  
5   body: TextButton(  
6     child: Text("Um simples botão"),  
7     onPressed: () {},  
8     style: TextButton.styleFrom(  
9       primary: Colors.white,  
10      backgroundColor: Colors.green,  
11    ),  
12  ),  
13)
```

Propriedade elevation

A propriedade *elevation* é utilizada para dar ênfase ao *Widget TextButton*. Essa propriedade cria uma sombra ao redor do *Widget*.



```
1 Scaffold(  
2   appBar: AppBar(  
3     title: Text("Aplicação Exemplo"),  
4   ),  
5   body: TextButton(  
6     child: Text("Um simples botão"),  
7     onPressed: () {},  
8     style: TextButton.styleFrom(  
9       primary: Colors.green,  
10      elevation: 5  
11    ),  
12  ),  
13)
```

Propriedade shadowColor

A propriedade *shadowColor* é utilizada para dar cor à sombra criada ao redor do *Widget*.



```
1 Scaffold(  
2   appBar: AppBar(  
3     title: Text("Aplicação Exemplo"),  
4   ),  
5   body: TextButton(  
6     child: Text("Um simples botão"),  
7     onPressed: () {},  
8     style: TextButton.styleFrom(  
9       primary: Colors.green,  
10      elevation: 5,  
11      shadowColor: Colors.greenAccent,  
12    ),  
13  ),  
14)
```