



# Flutter

Wesley Dias Maciel

2021/02

# Prática 08

## Classe MaterialApp

Documentação: <https://api.flutter.dev/flutter/material/MaterialApp-class.html>

**Objetivo:** apresentar a classe MaterialApp. Uma instância da classe MaterialApp é um widget que envolve vários outros widgets que são normalmente necessários para aplicativos de design material (Material Design: <https://material.io/design>, <https://materializecss.com/>).

1) Crie um novo projeto Flutter com o exemplo abaixo, usando:

- a. Visual Studio Code, ou;
- b. <https://dartpad.dev/>, ou;
- c. <https://flutlab.io/editor>, ou;
- d. <https://flutterstudio.app/>, ou;
- e. <https://codemagic.io/>.

O exemplo apresenta um texto na tela. A função runApp () recebe uma instância, objeto ou widget, da classe MaterialApp como parâmetro. Por sua vez, o parâmetro home do widget MaterialApp recebe um objeto da classe Home. A classe Home é uma classe sem estado, porque ela estende a classe StatelessWidget (<https://api.flutter.dev/flutter/widgets/StatelessWidget-class.html>). Isso significa que a classe Home não sofre alteração, seu estado não muda.

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

void main() {
  runApp(
    MaterialApp(
      home: Home(),
    ),
  );
}

class Home extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Text(
      "Olá, Mundo!!!",
    );
  }
}
```

As vírgulas no código geram uma indentação que destaca cada widget e seus parâmetros.  
A remoção das vírgulas gera uma versão mais compacta do código:

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

void main() {
  runApp(MaterialApp(home: Home()));
}

class Home extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Text("Olá, Mundo!!!");
  }
}
```



A função main () também pode ser chamada através da sintaxe de flecha (arrow syntax), como apresentado abaixo. A sintaxe de flecha é usada quando o escopo da função ou método possui apenas uma instrução. Exemplo:

O algoritmo:

```
int func(x) {
  return x * 2;
}

void main() {
  print(func(3));
}
```

Na sintaxe de flecha, poderia ser escrito na forma:

```
int func(x) => x * 2;  
void main() => print(func(5));
```

Um outro exemplo da sintaxe de flecha para fixar:

```
void func() => print(5);  
void main() => func();
```

Então:

```
import 'package:flutter/material.dart';  
  
void main() => runApp(MaterialApp(home: Home()));  
  
class Home extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return Text("Olá, Mundo!!!");  
  }  
}
```

Ou ainda:

```
import 'package:flutter/material.dart';  
  
void main() => runApp(MaterialApp(home: Home()));  
  
class Home extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) => Text("Olá, Mundo!!!");  
}
```



A classe Home sobrescreve, @override, o método build () da classe StatelessWidget. O método build () recebe como parâmetro um objeto context da classe BuildContext. Os objetos da classe BuildContext armazenam uma referência para o widget pai do widget corrente. Dessa forma, os objetos da classe BuildContext possuem uma referência para o contexto em que o widget corrente deve ser criado. Além disso, o método build () retorna um widget. No caso, ele retorna um widget Text:

```
@override
Widget build(BuildContext context) {
  return Text("Olá, Mundo!!!");
}
```

A classe Home é criada pelo desenvolvedor e poderia ter qualquer outro nome, como PaginaInicial, por exemplo:

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

void main() {
  runApp(MaterialApp(home: PaginaInicial()));
}

class PaginaInicial extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Text("Olá, Mundo!!!");
  }
}
```



## Classe Scaffold

A classe Scaffold implementa a estrutura básica de layout visual do material design (<https://api.flutter.dev/flutter/material/Scaffold-class.html>). A classe Scaffold possui 3 parâmetros opcionais de destaque: uma barra (appBar), um corpo (body) e um botão (floatingActionButton).

2) O próximo exemplo apresenta um uso simples da classe Scaffold.

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

void main() => runApp(MaterialApp(home: Home()));

class Home extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold();
  }
}
```

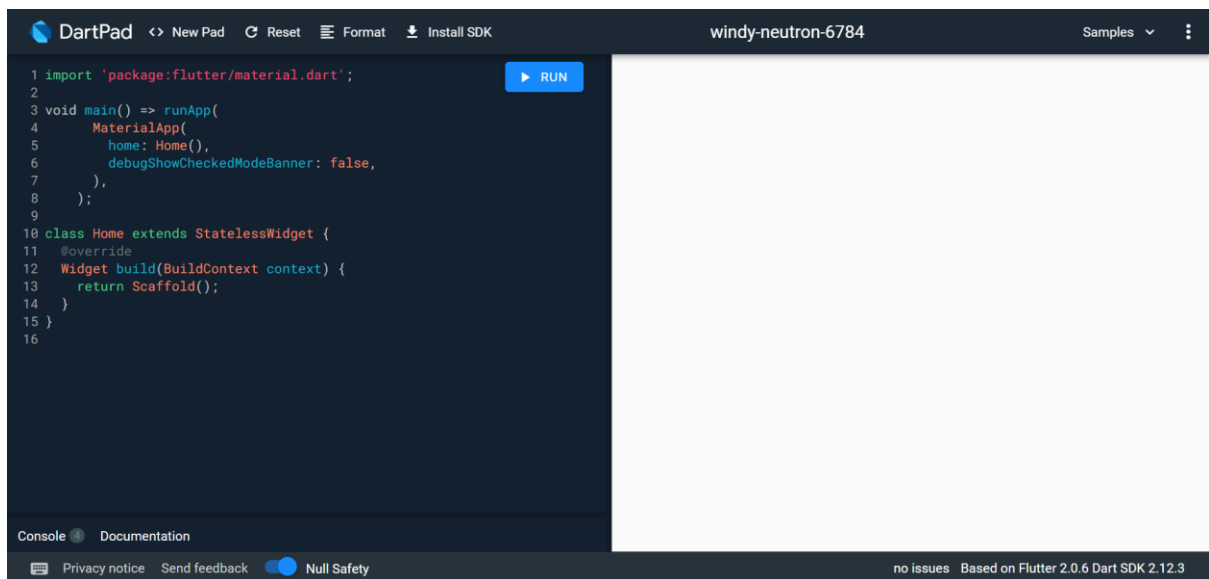


Observe que o exemplo apresenta uma faixa indicativa de debug. Caso você queira ocultar essa faixa, configure o atributo `debugShowCheckedModeBanner` do widget `MaterialApp` como `false`:

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

void main() => runApp(
  MaterialApp(
    home: Home(),
    debugShowCheckedModeBanner: false,
  ),
);

class Home extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold();
  }
}
```



The screenshot shows the DartPad interface with a Flutter app. The code defines a `main` function that runs `runApp` with a `MaterialApp` widget. The `MaterialApp` widget has a `home` property set to `Home()` and `debugShowCheckedModeBanner` set to `false`. The `Home` class extends `StatelessWidget` and overrides the `build` method to return a `Scaffold` widget.

```
1 import 'package:flutter/material.dart';
2
3 void main() => runApp(
4   MaterialApp(
5     home: Home(),
6     debugShowCheckedModeBanner: false,
7   ),
8 );
9
10 class Home extends StatelessWidget {
11   @override
12   Widget build(BuildContext context) {
13     return Scaffold();
14   }
15 }
16
```

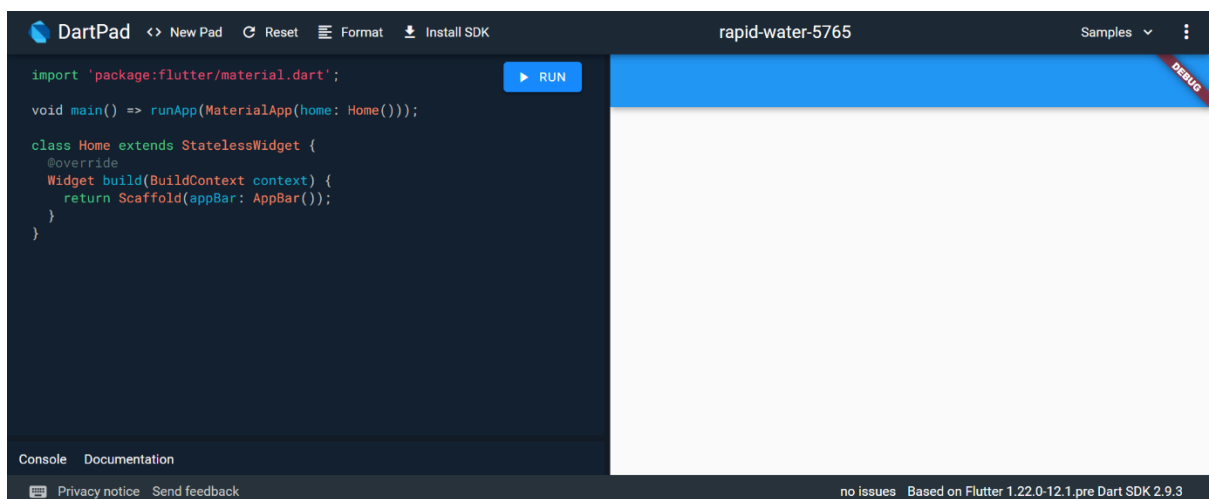
The console shows no issues, and the app is based on Flutter 2.0.6 and Dart SDK 2.12.3.

3) O exemplo seguinte inclui uma barra.

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

void main() => runApp(MaterialApp(home: Home()));

class Home extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(appBar: AppBar());
  }
}
```



The screenshot shows the DartPad interface with a Flutter app. The code defines a `main` function that runs `runApp` with a `MaterialApp` widget. The `MaterialApp` widget has a `home` property set to `Home()`. The `Home` class extends `StatelessWidget` and overrides the `build` method to return a `Scaffold` widget with an `AppBar` widget.

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

void main() => runApp(MaterialApp(home: Home()));

class Home extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(appBar: AppBar());
  }
}
```

The console shows no issues, and the app is based on Flutter 1.22.0-12.1.pre and Dart SDK 2.9.3.

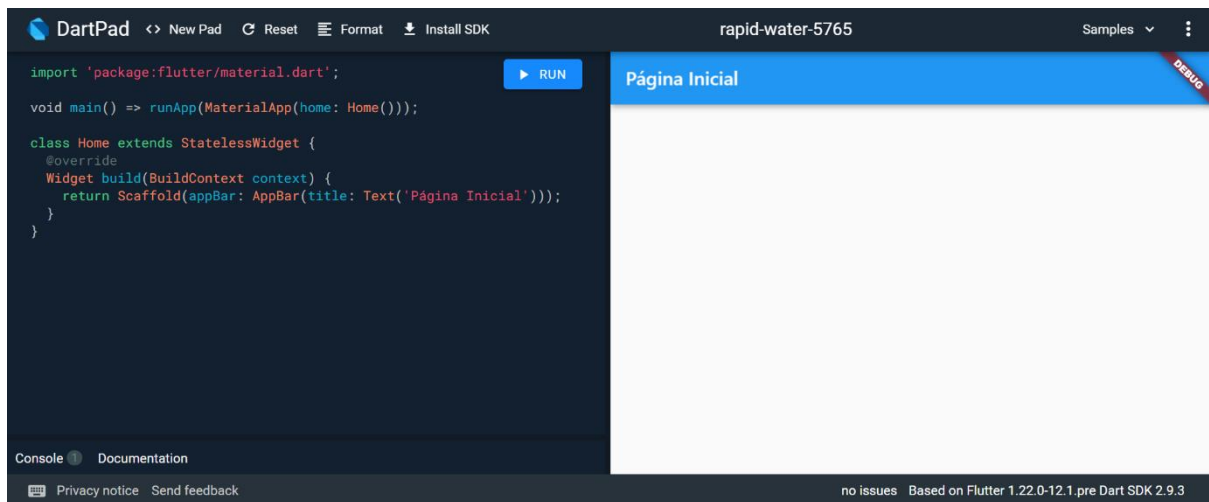
4) O exemplo abaixo inclui um título na barra.



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

void main() => runApp(MaterialApp(home: Home()));

class Home extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(appBar: AppBar(title: Text('Página Inicial')));
  }
}
```



5) Também podemos incluir um ícone (leading) na barra.

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

void main() => runApp(MaterialApp(home: Home()));

class Home extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        leading: Icon(Icons.home),
        title: Text('Página Inicial'),
      ),
    );
  }
}
```



6) O exemplo a seguir inclui um corpo, body, com um texto.

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

void main() => runApp(MaterialApp(home: Home()));

class Home extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        leading: Icon(Icons.home),
        title: Text('Página Inicial'),
      ),
      body: Text('Olá, Mundo!!!'),
    );
  }
}
```

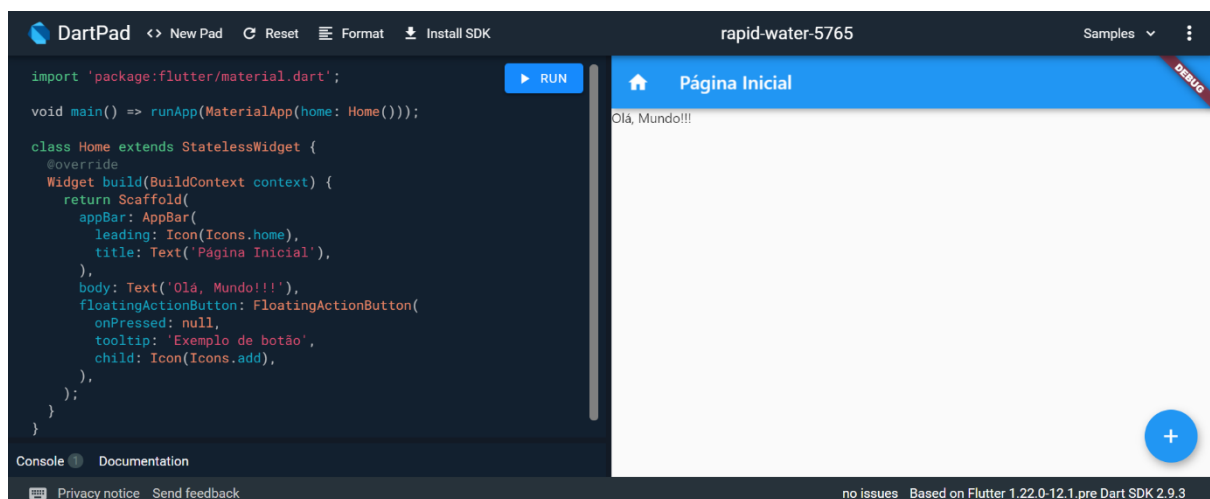


- 7) O próximo exemplo inclui um botão. Neste exemplo, não há uma funcionalidade associada ao botão, porque o parâmetro `onPressed` do botão recebe o valor `null`. Ao manter o botão pressionado, ele apresenta a mensagem (tooltip) “Exemplo de botão”. O botão possui um ícone, o sinal de adição (+).

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

void main() => runApp(MaterialApp(home: Home()));

class Home extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        leading: Icon(Icons.home),
        title: Text('Página Inicial'),
      ),
      body: Text('Olá, Mundo!!!'),
      floatingActionButton: FloatingActionButton(
        onPressed: null,
        tooltip: 'Exemplo de botão',
        child: Icon(Icons.add),
      ),
    );
  }
}
```

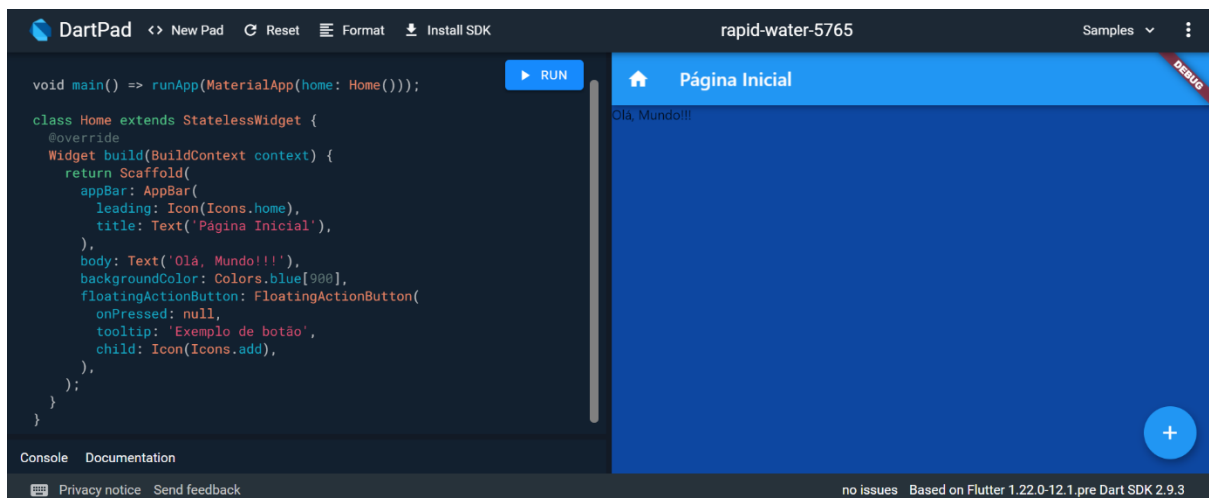


- 8) O próximo exemplo aplica cor azul ao fundo do corpo (body).

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

```
void main() => runApp(MaterialApp(home: Home()));

class Home extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        leading: Icon(Icons.home),
        title: Text('Página Inicial'),
      ),
      body: Text('Olá, Mundo!!!'),
      backgroundColor: Colors.blue[900],
      floatingActionButton: FloatingActionButton(
        onPressed: null,
        tooltip: 'Exemplo de botão',
        child: Icon(Icons.add),
      ),
    );
  }
}
```



9) O exemplo abaixo aplica estilo ao texto escrito no corpo (body):

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

void main() => runApp(MaterialApp(home: Home()));

class Home extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
```

```
appBar: AppBar(  
  leading: Icon(Icons.home),  
  title: Text('Página Inicial'),  
)  
,  
body: Text(  
  'Olá, Mundo!!!',  
  style: TextStyle(  
    color: Colors.white,  
    fontSize: 30,  
    fontWeight: FontWeight.bold,  
  ),  
)  
,  
backgroundColor: Colors.blue[900],  
floatingActionButton: FloatingActionButton(  
  onPressed: null,  
  tooltip: 'Exemplo de botão',  
  child: Icon(Icons.add),  
)  
,  
)  
);  
}
```

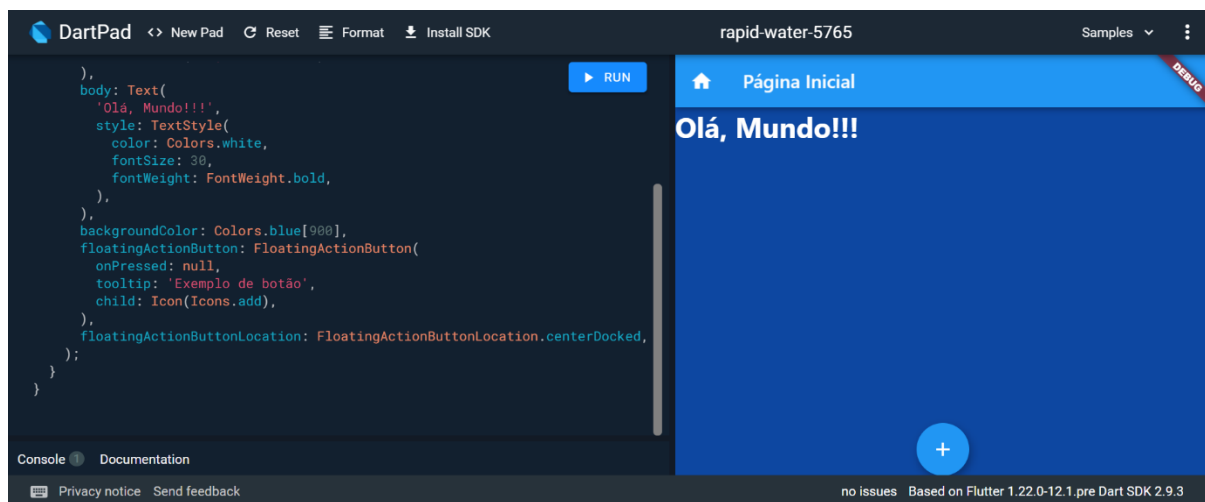


10) O próximo exemplo centraliza o botão:

```
import 'package:flutter/material.dart';  
  
void main() => runApp(MaterialApp(home: Home()));  
  
class Home extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  

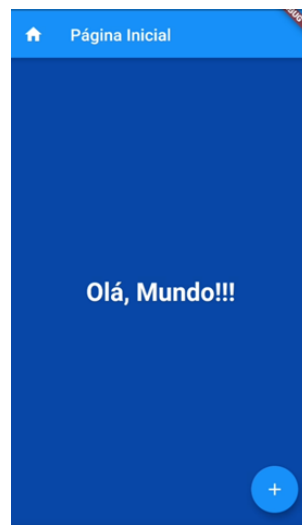
```

```
appBar: AppBar(  
  leading: Icon(Icons.home),  
  title: Text('Página Inicial'),  
)  
,  
body: Text(  
  'Olá, Mundo!!!',  
  style: TextStyle(  
    color: Colors.white,  
    fontSize: 30,  
    fontWeight: FontWeight.bold,  
  ),  
)  
,  
backgroundColor: Colors.blue[900],  
floatingActionButton: FloatingActionButton(  
  onPressed: null,  
  tooltip: 'Exemplo de botão',  
  child: Icon(Icons.add),  
)  
,  
floatingActionButtonLocation: FloatingActionButtonLocation.centerDocked,  
);  
}  
}
```

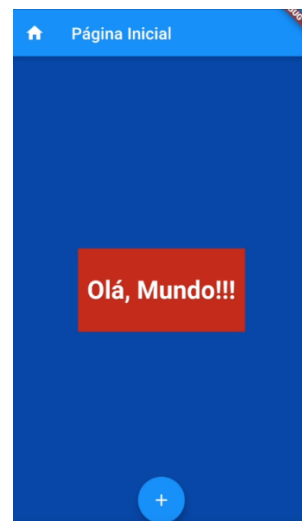


## Exercício

- 1) Altere o exemplo desta prática, para que ele apresente uma tela semelhante à apresentada abaixo.



- 2) Altere o exemplo desta prática, para que ele apresente uma tela semelhante à apresentada abaixo.



- 3) Altere o exemplo desta prática, para que ele apresente uma tela semelhante à apresentada abaixo.

Link para o exercício: <https://media.giphy.com/media/pt0EKLDJmVvIS/giphy.gif>

