

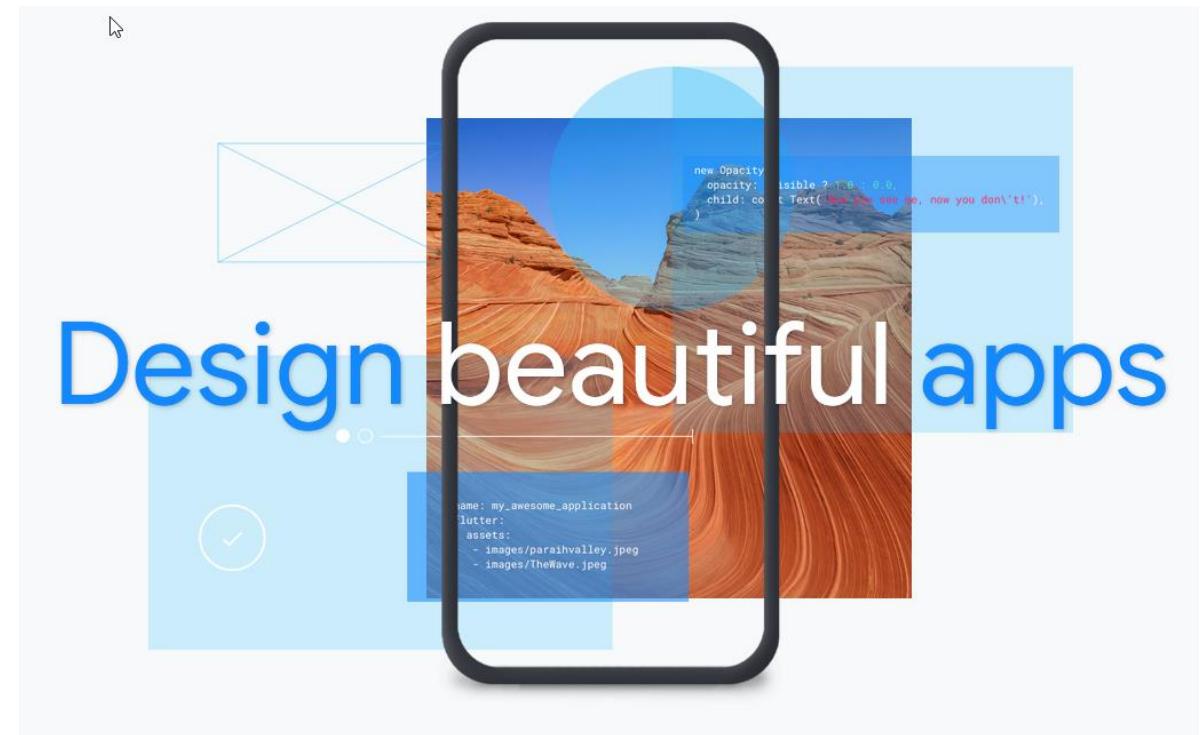


Transformando o futuro das pessoas  
e as pessoas para o futuro.

#Senacfaz75



# Desenvolvimento Mobile: Flutter



## O que é o Flutter?

Flutter é um kit de desenvolvimento de interface de usuário (UI toolkit), de código aberto, criado pelo Google, que possibilita a criação de aplicativos compilados nativamente. Atualmente pode compilar para Android, iOS, Windows, Mac, Linux, Google \*Fuchsia e Web.

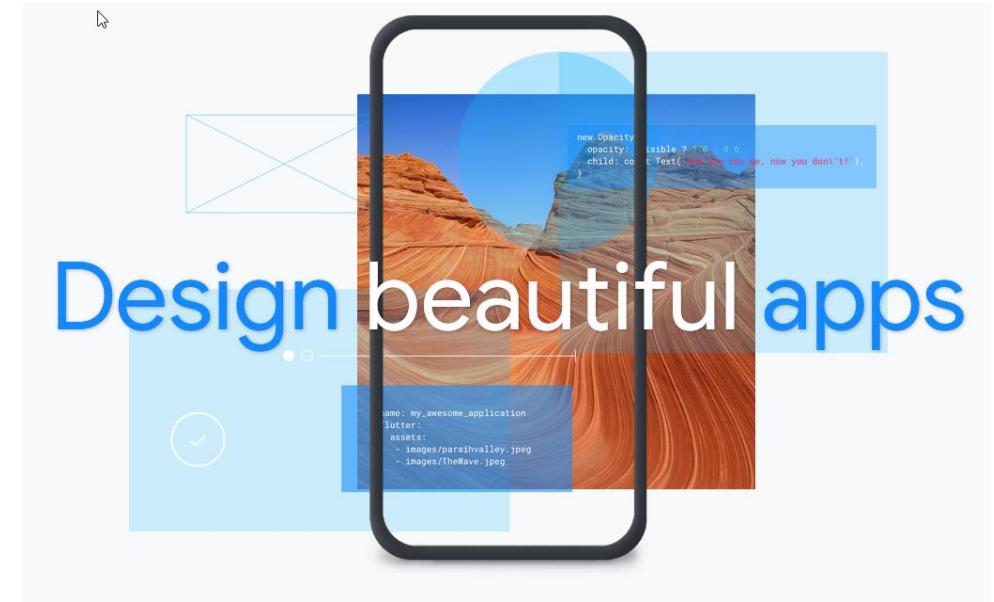
### Características

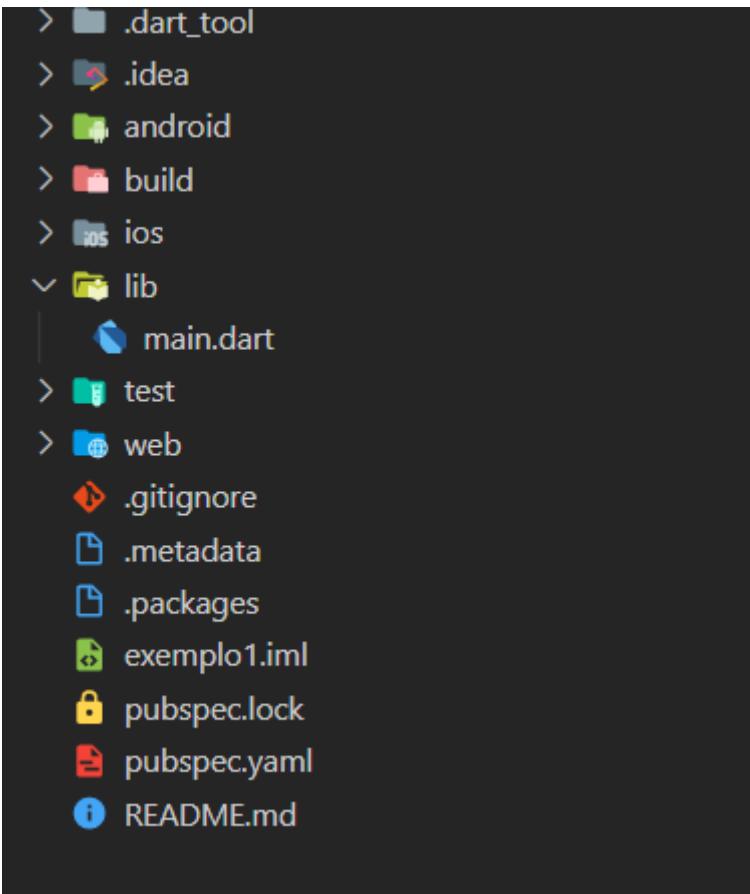
- tem como linguagem base o Dart
- É orientado a objetos
- Misto de Java, Javascript e C#
- Focada em Frontend
- Árvore de Widgets
- A tela inteira é um widget

### Composição

**SDK:** Kit de desenvolvimento para compilar código fonte para código nativo

**Widget:** Componentes UI reutilizáveis desenvolvida do funções, classes e pacotes.





**.dart\_tool:** Criado a partir de um pacote build runner, ficam os arquivos compilados da aplicação. (não mexer) está marcada pelo git como ignorada.

**.idea:** configuração do IntelliJ idea (não mexer).

**Android:** Todos arquivos relacionados a construção do projeto Android.

**build:** componentes de construção do projeto e compilados.

**ios:** pasta de projeto para o XCode.

**lib:** pasta onde programamos nosso projeto, criações de widgets, start do nosso projeto.

**test:** para que for trabalhar com testes, pode-se excluir essa pasta.

**web:** Pasta nova.

**.gitignore:** arquivos que serão ignorados no comit do git.

**.metadatada:** gerenciados pelo Flutter.

**.packages:** rotas de dependência.

**.iml:** criado para IntelliJ.

**pubspec.lock** e **pubspec.yaml:** Criados juntos para configuração do App, exemplo caminhos de assets, dependências.

## Primeiro programa

**Import:** para importar bibliotecas e arquivos externos

**Main():** Classe principal da aplicação Flutter

**runApp():** Disparador do aplicativo

**MaterialApp():** Classe de desenvolvimento visual do Flutter

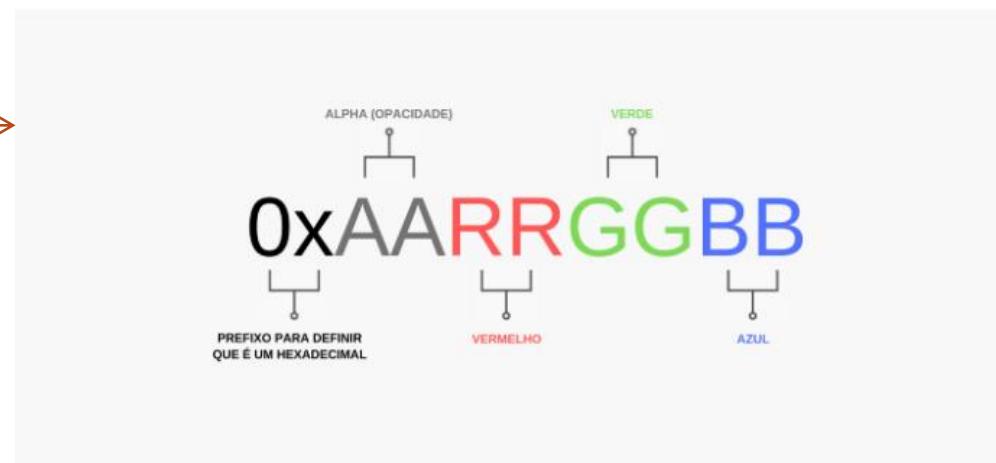
**Title e home:** Parâmetros nomeados de MaterialApp()

**Container():** "Envelope" para um conteúdo da aplicação, lembra uma div.

**Color(minúsculo):** parâmetro nomeado para inserção de Cores

**Colors(maiúsculo):** Construtor para aplicação da cor

```
main01_primeiroPrograma.dart
lib > main01_primeiroProgramma.dart > ...
1 import 'package:flutter/material.dart';
2
3 main() => runApp(MaterialApp(
4   title: 'Sorteio da MegaSena',
5   home: Container(
6     color: Colors.white,
7   ), // Container
8 )); // MaterialApp
```



<https://api.flutter.dev/flutter/dart-ui/Color-class.html>

## Inserindo Colunas

**Import:** para importar bibliotecas e arquivos externos

**Main():** Classe principal da aplicação Flutter

**runApp():** Disparador do aplicativo

**MaterialApp():** Classe de desenvolvimento visual do Flutter

**Title e home:** Parâmetros nomeados de MaterialApp()

**Container():** “Envelope” para um conteúdo da aplicação, lembra uma div.

**Color(minúsculo):** parâmetro nomeado para inserção de Cores

**Colors(maiúsculo):** Construtor para aplicação da cor

**Child:** Parâmetro filho de Container(), só recebe 1 widget

**Column():** Widget para inserir colunas

**Children:** Array filho para vários Widget

**Text():** Widget para inserir textos.

```
lib > main02_colunas.dart > ...
1 //Inserindo um Widget Coluna
2 import 'package:flutter/material.dart';
3
4 Run | Debug | Profile
5 main() => runApp(MaterialApp(
6   title: 'Sorteio da MegaSena',
7   home: Container(
8     color: Colors.grey,
9     child: Column(
10       //Uso um children para colocar vários widgets
11       children: [
12         Text('Texto 1'),
13         Text('Texto 2'),
14         Text('Texto 3')],
15     ), // Column
16   )), // Container // MaterialApp
```

## Inserindo Linhas

**Import:** para importar bibliotecas e arquivos externos

**Main():** Classe principal da aplicação Flutter

**runApp():** Disparador do aplicativo

**MaterialApp():** Classe de desenvolvimento visual do Flutter

**Title e home:** Parâmetros nomeados de MaterialApp()

**Container():** “Envelope” para um conteúdo da aplicação, lembra uma div.

**Color(minúsculo):** parâmetro nomeado para inserção de Cores

**Colors(maiúsculo):** Construtor para aplicação da cor

**Child:** Parâmetro filho de Container(), só recebe 1 widget

**Row():** Widget para inserir linhas

**Children:** Array filho para vários Widget

**Text():** Widget para inserir textos.

```
lib > main03_linhas.dart > ...
1 //Inserindo um Widget Linhas
2 import 'package:flutter/material.dart';
3
4 Run | Debug | Profile
5 main() => runApp(MaterialApp(
6   title: 'Sorteio da MegaSena',
7   home: Container(
8     color: Colors.grey,
9     child: Row(
10       //Uso um children para colocar vários widgets
11       children: [
12         Text('Texto 1'),
13         Text('Texto 2'),
14         Text('Texto 3')],
15     ), // Row
16   )); // Container // MaterialApp
```

## Formatando textos

- **style:** Definindo as formatações do Texto
- **fontSize:** Tamanho da fonte
- **fontStyle:** Estilo da fonte
- **fontWeight:** Peso da fonte (Negrito)
- **letterSpacing:** Espaçamento entre os caracteres
- **decoration:** sublinhados e tachados no texto
- **color:** Cor da fonte

```
lib > main04_FormatacaoTexto.dart > ...
1 //Formatação de texto
2 import 'package:flutter/material.dart';
3
4 Run | Debug | Profile
5 main() => runApp(MaterialApp(
6   title: 'Sorteio MegaSena',
7   //container é um envólucro para os conteúdos
8   home: Container(
9     color: Colors.white,
10
11   //Definindo um filho pra o container
12   child: Column(
13     //Definindo um children para o Column
14     children: [
15       Text(
16         'Esse é um exemplo de texto extenso para o widget',
17         style: TextStyle(
18           //Definindo as formatações do Texto
19           fontSize: 40,
20           fontStyle: FontStyle.normal,
21           fontWeight: FontWeight.normal,
22
23           //Espaçamento entre os caracteres
24           letterSpacing: 2,
25
26           //Retirar o sublinhado
27           decoration: TextDecoration.none,
28
29           //Aplicar na Fonte
30           color: Colors.black
31         ), // TextStyle
32       ],
33     ), // Column
34   ), // Container
35 )); // MaterialApp
```

## Espaçamentos internos e externos (Padding e Margin)

### **padding: EdgeInsets**

.all : Espaçamento para todos os lados (valor único)

.fromLTRB: Espaçamento para todos os lados (left, top, right, bottom)

.only: Espaçamento individual (left: 10, por exemplo)

### **margin: EdgeInsets**

.all : Margem para todos os lados (valor único)

.fromLTRB: Margem para todos os lados (left, top, right, bottom)

.only: Espaçamento individual (left: 10, por exemplo)

**BoxDecoration:** Construtor para as bordas do box

**Color:** Aplicação de cores

**Border:** Definindo a borda

**Width:** Espessura

**Color:** Cor



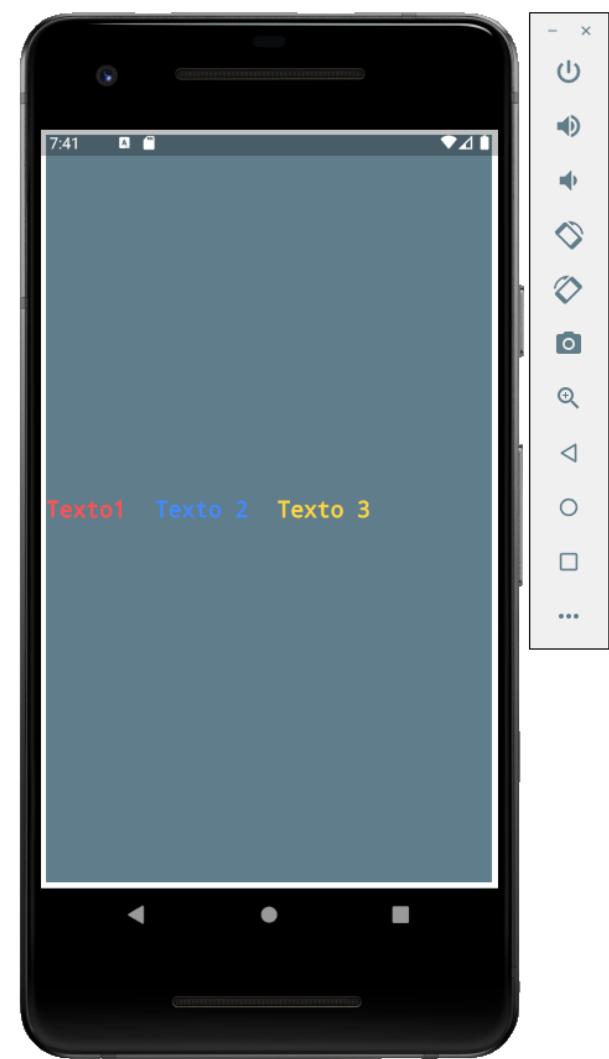
```
lib > main.dart > main
1 import 'package:flutter/material.dart';
Run | Debug | Profile
2 main() => runApp(MaterialApp(
3   //Removendo a faixa de debug
4   debugShowCheckedModeBanner: false,
5   title: 'Sorteio MegaSena',
6   home: Container(
7     //Defindo espaçamento internos
8     // .all .fromLTRB .only
9     //padding: EdgeInsets.all()
10
11    padding: EdgeInsets.fromLTRB(20, 50, 20, 0),
12    //Definindo os espaçamento externos
13    // .all .fromLTRB .only
14    //margin: EdgeInsets.all()
15
16    margin: EdgeInsets.fromLTRB(20, 50, 20, 20),
17    //Borda e cor do Container
18
19    decoration: BoxDecoration(
20      //Adiconando uma cor de background
21      color: Colors.blueGrey,
22
23      //Adicionando uma borda
24      border: Border.all(
25        width: 5,
26        color: Colors.amber
27      ), // Border.all
28    ), // BoxDecoration
29
```

```
30    //Adicionar um texto dentro do Container
31    child: Column(
32      children: [
33        Text(
34          'Texto exemplo para visualizar o preenchimento da tela do App',
35          textAlign: TextAlign.center,
36          style: TextStyle(
37            fontSize: 25,
38            fontWeight: FontWeight.normal,
39            decoration: TextDecoration.none,
40            color: Colors.white
41          ), // TextStyle
42        ) // Text
43      ],
44    ), // Column
45  ), // Container
46 ) // MaterialApp
47 );
```

## Espaçamento no Widget

```
lib > main.dart > main
1 //Padding no Widget
2 import 'package:flutter/material.dart';
3
4 Run | Debug | Profile
5 main() => runApp(MaterialApp(
6     //Removendo a faixa de debug
7     debugShowCheckedModeBanner: false,
8     title: 'Padding no Widget',
9     home: Container(
10         decoration: BoxDecoration(
11             color: Colors.blueGrey,
12             border: Border.all(width: 5, color: Colors.white)), // BoxDecoration
13         child: Row(
14             children: [
15                 Text(
16                     'Texto1',
17                     style: TextStyle(
18                         fontSize: 20,
19                         decoration: TextDecoration.none,
20                         color: Colors.redAccent), // TextStyle
21                 ), // Text
22
23                 //Padding diretamente no widget
24                 Padding(→
25                     padding: EdgeInsets.all(25),
26                     child: Text(
27                         'Texto 2',
28                         style: TextStyle(
29                             fontSize: 20,
30                             decoration: TextDecoration.none,
31                             color: Colors.blueAccent
32
33                         ), // TextStyle
34                         ), // Text
35                 ), // Padding
36
37             ],
38             mainAxisAlignment: MainAxisAlignment.spaceEvenly,
39         ),
40         ),
41     ),
42     ),
43     ],
44     ),
45     ],
46     ),
47 )); // MaterialApp
```

```
36     ],
37     ),
38     ],
39     ),
40     ],
41     ],
42     ],
43     ],
44     ],
45     ],
46     ],
47 )); // MaterialApp
```



O construtor Padding() pode ser usado diretamente em outros widgets.

Há uma forma rápida para colocar um padding no widget, basta teclarmos <CTRL> + <.›. Assim podemos envolver nosso widget em um Padding

# Alinhamentos

## Alinhamento Principal (mainAxisAlignment: MainAxisAlignment)

**mainAxisAlignment:** MainAxisAlignment.start: Alinhamento no início

**mainAxisAlignment:** MainAxisAlignment.end: Alinhamento no Fim

**mainAxisAlignment:** MainAxisAlignment.center: Alinhamento no centro

**mainAxisAlignment:** MainAxisAlignment.spaceEvenly: Alinhamento distribuído

## Rebatendo o alinhamento do Eixo Principal (crossAxisAlignment )

**crossAxisAlignment:** CrossAxisAlignment.start: Alinhamento no início

**crossAxisAlignment:** CrossAxisAlignment.end: Alinhamento no Fim

**crossAxisAlignment:** CrossAxisAlignment.center: Alinhamento no centro



# Alinhamentos

```
lib > main.dart > main
1 //Alinhamentos
2
3 import 'package:flutter/material.dart';
4
5 Run | Debug | Profile
6 main() => runApp(MaterialApp(
7   //removendo a faixa de debug
8   debugShowCheckedModeBanner: false,
9
10  title: 'Alinhamentos',
11  home: Container(
12    //definindo margens
13    margin: EdgeInsets.only(top: 60),
14
15    //defindo borda e cor do container
16    decoration: BoxDecoration(
17      border: Border.all(
18        width: 5,
19        color: Colors.blue
20      ) // Border.all
21    ), // BoxDecoration
22
23    //Criando um child
24    child: Row(
25      children: [
26
27        //Criando textos
28        Text(
29          'T1',
30          style: TextStyle(
31            decoration: TextDecoration.none,
32            fontSize: 30
33          ), // TextStyle
34        ), // Text
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
```

```
//Espaçando internamente o T2
Text(
  'T2',
  style: TextStyle(
    decoration: TextDecoration.none,
    fontSize: 30
  ), // TextStyle
), // Text

Text(
  'T3',
  style: TextStyle(
    decoration: TextDecoration.none,
    fontSize: 30
  ), // TextStyle
) // Text
],
//Alinhamento principal
//mainAxisAlignment: MainAxisAlignment.start .center .end .spaceEvenly
mainAxisAlignment: MainAxisAlignment.start,
//Configurar o eixo que cruza o alinhamento principal
//crossAxisAlignment: CrossAxisAlignment.start .end .center
crossAxisAlignment: CrossAxisAlignment.start,
), // Row
) // Container
)); // MaterialApp

```

# Botões

## TextButton():

Use botões de texto em barras de ferramentas, em diálogos ou em linha com outros conteúdos. Os botões de texto não possuem bordas visíveis e, portanto, devem confiar em sua posição em relação a outros conteúdos para contexto. Evite usar botões de texto onde eles se misturariam com outros conteúdos, por exemplo, no meio de listas.

## ElevatedButton()

Use botões elevados para adicionar dimensão a layouts com botões, por exemplo, em longas listas de conteúdo ocupados ou em espaços amplos. Evite usar botões elevados em conteúdo já elevado, como diálogos ou cards.

## OutlinedButton()

Os OutlinedButton são botões de ênfase média. Eles contêm ações que são importantes, mas não são a ação principal em um aplicativo.

```
lib > main08_buttons.dart > main
1   //Trabalhando com botões
2
3   import 'package:flutter/material.dart';
4
5   Run | Debug | Profile
6   main() => runApp(MaterialApp(
7       title: 'Trabalhando com botões',
8       home: Container(
9           //Margens
10          margin: EdgeInsets.only(top: 60),
11
12          //Espaçamento
13          padding: EdgeInsets.all(20),
14          color: Colors.blueGrey,
15
16          child: Column(
17              //Espaçamento principal
18              mainAxisAlignment: MainAxisAlignment.center,
19
20              children: [
21                  TextButton(
22                      style: TextButton.styleFrom(
23                          //borda arredondada
24                          shape: RoundedRectangleBorder(
25                              //controlar o arredondamento
26                              borderRadius: BorderRadius.circular(10)
27                          ), // RoundedRectangleBorder
28                          //cor de foreground
29                          primary: Colors.pinkAccent,
30
31                          //cor de background
32                          backgroundColor: Colors.white,
33
34                          //Espaçamento dentro do botão
35                          padding:
36                              EdgeInsets.symmetric(
37                                  horizontal: 32,
38                                  vertical: 32) // EdgeInsets.symmetric
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
onPressed: () {
    print('Executando...');
},
child: Text(
    'Meu Botão',
    style: TextStyle(
        fontSize: 20
    ), // TextStyle
) // Text
), // TextButton
//Forçar um espaçamento
const SizedBox(height: 20),
ElevatedButton(
style: ElevatedButton.styleFrom(
//cor Background
primary: Colors.amber,
//cor foreground
onPrimary: Colors.white,
),
onPressed: (){},
child: Text(
    'Botão 2',
    style: TextStyle(
        fontSize: 20
    ), // TextStyle
) // Text
), // ElevatedButton
const SizedBox(height: 20),
```

# Botões

## TextButton():

Use botões de texto em barras de ferramentas, em diálogos ou em linha com outros conteúdos. Os botões de texto não possuem bordas visíveis e, portanto, devem confiar em sua posição em relação a outros conteúdos para contexto. Evite usar botões de texto onde eles se misturariam com outros conteúdos, por exemplo, no meio de listas.

## ElevatedButton()

Use botões elevados para adicionar dimensão a layouts com botões, por exemplo, em longas listas de conteúdo ocupados ou em espaços amplos. Evite usar botões elevados em conteúdo já elevado, como diálogos ou cards.

## OutlinedButton()

Os OutlinedButton são botões de ênfase média. Eles contêm ações que são importantes, mas não são a ação principal em um aplicativo.

```
69          //Botão outline
70          OutlinedButton(
71              style: OutlinedButton.styleFrom(
72                  //Cor do foreground
73                  primary: Colors.black,
74
75                  //Cor da borda
76                  side: BorderSide(
77                      width: 1,
78                      color: Colors.orange
79                  ) // BorderSide
80
81              ),
82
83              onPressed: (){},
84              child: Text(
85                  'Botão 3',
86                  style: TextStyle(
87                      fontSize: 30
88                  ), // TextStyle
89                  ), // Text
90              ), // OutlinedButton
91
92              const SizedBox(height: 20),
93              //Botão com ícone
94              ElevatedButton.icon(
95                  style: ElevatedButton.styleFrom(
96                      side: BorderSide(
97                          width: 1,
98                          color: Colors.amber
99                      ), // BorderSide
100                     primary: Colors.red,
101                     onPrimary: Colors.amber
102                 ),
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129 ); // MaterialApp
```

```
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129 ); // MaterialApp
```

# Botões

## TextButton():

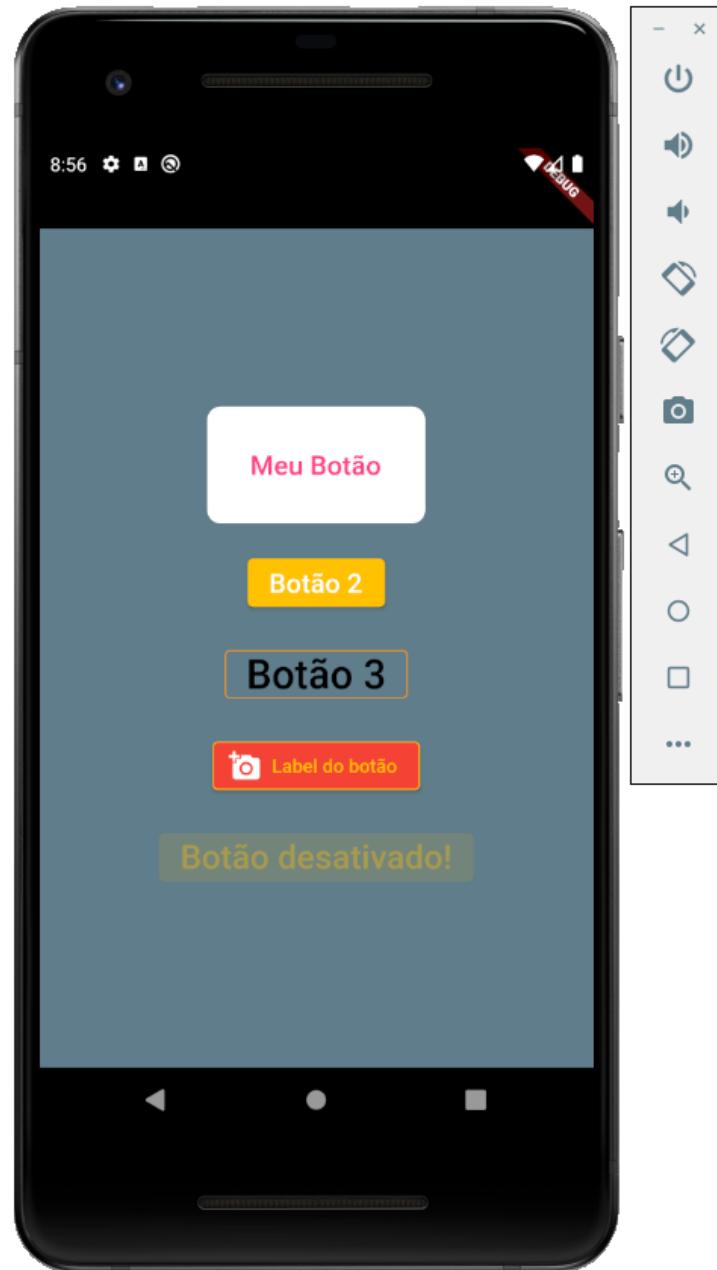
Use botões de texto em barras de ferramentas, em diálogos ou em linha com outros conteúdos. Os botões de texto não possuem bordas visíveis e, portanto, devem confiar em sua posição em relação a outros conteúdos para contexto. Evite usar botões de texto onde eles se misturariam com outros conteúdos, por exemplo, no meio de listas.

## ElevatedButton()

Use botões elevados para adicionar dimensão a layouts com botões, por exemplo, em longas listas de conteúdo ocupados ou em espaços amplos. Evite usar botões elevados em conteúdo já elevado, como diálogos ou cards.

## OutlinedButton()

Os OutlinedButton são botões de ênfase média. Eles contêm ações que são importantes, mas não são a ação principal em um aplicativo.



# Imagens

Os aplicativos Flutter podem incluir código e assets (às vezes chamados de recursos). Um asset é um arquivo que é empacotado e implantado com seu aplicativo e pode ser acessado no tempo de execução. Os tipos comuns de assets incluem dados estáticos (por exemplo, arquivos JSON), arquivos de configuração, ícones e imagens (JPEG, WebP, GIF, WebP / GIF animado, PNG, BMP e WBMP).

## Especificando Assets

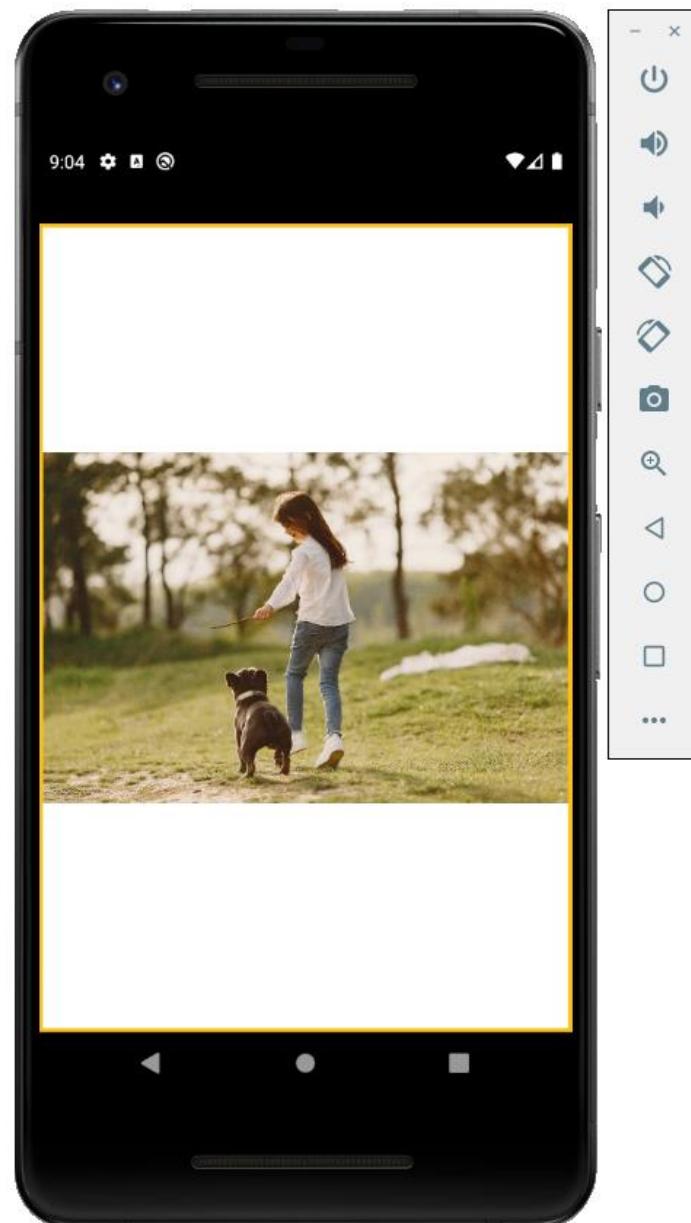
O Flutter usa o arquivo `pubspec.yaml`, localizado na raiz do seu projeto, para identificar os assets exigidos por um aplicativo.

```
! pubspec.yaml
46
47 # To add assets to your application, add an assets section, like this:
48 # assets:
49 #   - images/a_dot_burr.jpeg
50 #   - images/a_dot_ham.jpeg
51 assets:
52   - images/passeio.jpg
53
54 # An image asset can refer to one or more resolution-specific "variants", see
55 # https://flutter.dev/assets-and-images/#resolution-aware.
56
57 # For details regarding adding assets from package dependencies, see
58 # https://flutter.dev/assets-and-images/#from-packages
```

```
lib > main.dart > ...
1 import 'package:flutter/material.dart';
2
3 Run | Debug | Profile
4 main() => runApp(MaterialApp(
5   debugShowCheckedModeBanner: false,
6   title: 'Trabalhando com imagens',
7   home: Container(
8     margin: EdgeInsets.only(top: 60),
9
10    decoration: BoxDecoration(
11      //background
12      color: Colors.white,
13
14      //definir bordas
15      border: Border.all(
16        width: 3,
17        color: Colors.amber
18      ) // Border.all
19    ), // BoxDecoration
20
21    //Um child para o widget Image
22    child: Image.asset(
23      //Caminho da imagem
24      'images/passeio.jpg',
25      //Definição de preenchimento da imagem
26      //cobra todo o espaçamento cortanto a imagem
27      //fit: BoxFit.cover,
28      //fit: BoxFit.contain //valor padrão
29      //fit: BoxFit.fill //Preenche com distorção
30      //fit: BoxFit.fitHeight, //preencher pela altura (Largura na proporção)
31      //fit: BoxFit.fitWidth, //preencher pela largura (altura na proporção)
32      //fit: BoxFit.none //Mantem a imagem original
33      fit: BoxFit.contain //Mantem a imagem original
34    ), // Image.asset
35  ), // Container
36)); // MaterialApp
```

# Imagens

```
lib > main.dart > ...
1 import 'package:flutter/material.dart';
2
3 void main() => runApp(MaterialApp(
4   debugShowCheckedModeBanner: false,
5   title: 'Trabalhando com imagens',
6   home: Container(
7     margin: EdgeInsets.only(top: 60),
8
9     decoration: BoxDecoration(
10       //background
11       color: Colors.white,
12
13       //definir bordas
14       border: Border.all(
15         width: 3,
16         color: Colors.amber
17       ) // Border.all
18     ), // BoxDecoration
19
20     //Um child para o widget Image
21     child: Image.asset(
22       //Caminho da imagem
23       'images/passeio.jpg',
24       //Definição de preenchimento da imagem
25       //cobra todo o espaço cortando a imagem
26       //fit: BoxFit.cover,
27       //fit: BoxFit.contain //valor padrão
28       //fit: BoxFit.fill //Preenche com distorção
29       //fit: BoxFit.fitHeight, //preencher pela altura (Largura na proporção)
30       //fit: BoxFit.fitWidth, //preencher pela largura (altura na proporção)
31       //fit: BoxFit.none //Mantem a imagem original
32       fit: BoxFit.contain //Mantem a imagem original
33     ), // Image.asset
34   ), // Container
35 )); // MaterialApp
```



# Scaffold

Scaffold é uma classe de flutter que fornece muitos widgets ou podemos dizer APIs como Drawer, Snackbar, BottomNavigationBar, FloatingActionButton, AppBar etc. Scaffold irá expandir ou ocupar toda a tela do dispositivo. ... O Scaffold fornecerá uma estrutura para implementar o layout de Material Design básico do aplicativo.

```
lib > main10_Scaffold.dart > ...
1 import 'package:flutter/material.dart';
Run | Debug | Profile
2 main() => runApp(MaterialApp(
3   debugShowCheckedModeBanner: false,
4   home: Scaffold(
5     //Criando uma estrutura
6     //Divisão de 3 áreas: Título, corpo e rodapé
7     appBar: AppBar(
8       //Título
9       title: Text('Minha aplicação'),
10      backgroundColor: Colors.amber,
11    ), // AppBar
12    body: Padding(
13      padding: EdgeInsets.all(50),
14      child: Text(
15        'Conteúdo do App',
16        style: TextStyle(fontSize: 20),
17      ), // Text
18    ), // Padding
19    bottomNavigationBar: BottomAppBar(
20      //Aplicando uma cor
21      color: Colors.lightBlue,
22      child: Padding(
23        padding: EdgeInsets.all(20),
24        child: Row(
25          //Array de widgets
26          children: [
27            Text('Texto 1 bottom'),
28            Padding(
29              padding: const EdgeInsets.all(8.0),
30              child: Text(
31                'Texto 2 bottom',
32                style: TextStyle(fontSize: 20, color: Colors.white),
33              ), // Text
34            ), // Padding
35            Text('Texto 3 bottom')
36          ],
37        ), // Row
38      ), // Padding
39    ), // BottomAppBar
40  ), // Scaffold
41)); // MaterialApp
```



# Entrada de Dados

Um campo de texto permite que o usuário insira texto, com teclado de hardware ou com teclado na tela.

O campo de texto chama o retorno de chamada `onChanged` sempre que o usuário altera o texto no campo. Se o usuário indicar que acabou de digitar no campo (por exemplo, pressionando um botão no teclado virtual ), o campo de texto chama o retorno de chamada `onSubmitted` .

Para controlar o texto que é exibido no campo de texto, use o controlador. Por exemplo, para definir o valor inicial do campo de texto, use um controlador que já contenha algum texto. O controlador também pode controlar a seleção e a região de composição (e observar as mudanças no texto, seleção e região de composição).

```
lib > main.dart > ...
1 //Entrada de dados
2
3 import 'package:flutter/material.dart';
4 import 'package:flutter/services.dart';
5
6 Run | Debug | Profile
7 main() => runApp(MaterialApp(
8   home: Entrada(),
9 ));
10 //Criando um estado
11 class Entrada extends StatefulWidget {
12   @override
13   _EntradaState createState() => _EntradaState();
14 }
15
16 class _EntradaState extends State<Entrada> {
17   //Iniciar um controlador
18   TextEditingController _textEditingController = TextEditingController();
19
20   //Uma variável de saída na tela
21   String _resposta = 'Resultado';
22
23   @override
24   Widget build(BuildContext context) {
25     return Scaffold(
26       appBar: AppBar(
27         title: Text('Entrada de dados'),
28         backgroundColor: Colors.amber,
29       ), // AppBar
30       body: Column(
31         children: [
32           Padding(
33             padding: EdgeInsets.all(20),
34           ),
35         ],
36       ),
37     );
38   }
39
40   void _atualizar() {
41     setState(() {
42       _resposta = _textEditingController.text;
43     });
44   }
45
46   void _limpar() {
47     _textEditingController.clear();
48   }
49
50   void _deletar() {
51     _textEditingController.selection =
52       TextSelection.fromPosition(TextPosition(
53         offset: 0,
54       ));
55     _textEditingController.removeSelection(TextSelectionRange(
56       start: 0,
57       end: 1,
58     ));
59   }
60
61   void _inverter() {
62     _textEditingController.text =
63       _textEditingController.text.reversed.toString();
64   }
65
66   void _formatar() {
67     setState(() {
68       _resposta = _textEditingController.text
69         .toString()
70         .split(' ')
71         .map((String item) {
72           return item[0].toUpperCase() +
73             item.substring(1).toLowerCase();
74         })
75         .join(' ');
76     });
77   }
78
79   void _mostrar() {
80     setState(() {
81       _resposta = _textEditingController.text;
82     });
83   }
84
85   void _esconder() {
86     setState(() {
87       _resposta = '';
88     });
89   }
90
91   void _capturar() {
92     setState(() {
93       _resposta = _textEditingController.value.text;
94     });
95   }
96
97   void _limparTodos() {
98     setState(() {
99       _resposta = '';
100      _textEditingController.clear();
101    });
102  }
103
104  void _atualizarTodos() {
105    setState(() {
106      _resposta = _textEditingController.text;
107      _textEditingController.clear();
108    });
109  }
110
111  void _deletarTodos() {
112    setState(() {
113      _resposta = '';
114      _textEditingController.selection =
115        TextSelection.fromPosition(TextPosition(
116          offset: 0,
117        ));
118    });
119  }
120
121  void _inverterTodos() {
122    setState(() {
123      _resposta = _textEditingController.text
124        .toString()
125        .split(' ')
126        .map((String item) {
127          return item[0].toUpperCase() +
128            item.substring(1).toLowerCase();
129        })
130        .join(' ');
131    });
132  }
133
134  void _formatarTodos() {
135    setState(() {
136      _resposta = _textEditingController.text
137        .toString()
138        .split(' ')
139        .map((String item) {
140          return item[0].toUpperCase() +
141            item.substring(1).toLowerCase();
142        })
143        .join(' ');
144    });
145  }
146
147  void _mostrarTodos() {
148    setState(() {
149      _resposta = _textEditingController.text;
150    });
151  }
152
153  void _esconderTodos() {
154    setState(() {
155      _resposta = '';
156    });
157  }
158
159  void _capturarTodos() {
160    setState(() {
161      _resposta = _textEditingController.value.text;
162    });
163  }
164
165  void _limparTodosTodos() {
166    setState(() {
167      _resposta = '';
168      _textEditingController.clear();
169    });
170  }
171
172  void _atualizarTodosTodos() {
173    setState(() {
174      _resposta = _textEditingController.text;
175      _textEditingController.clear();
176    });
177  }
178
179  void _deletarTodosTodos() {
180    setState(() {
181      _resposta = '';
182      _textEditingController.selection =
183        TextSelection.fromPosition(TextPosition(
184          offset: 0,
185        ));
186    });
187  }
188
189  void _inverterTodosTodos() {
190    setState(() {
191      _resposta = _textEditingController.text
192        .toString()
193        .split(' ')
194        .map((String item) {
195          return item[0].toUpperCase() +
196            item.substring(1).toLowerCase();
197        })
198        .join(' ');
199    });
200  }
201
202  void _formatarTodosTodos() {
203    setState(() {
204      _resposta = _textEditingController.text
205        .toString()
206        .split(' ')
207        .map((String item) {
208          return item[0].toUpperCase() +
209            item.substring(1).toLowerCase();
210        })
211        .join(' ');
212    });
213  }
214
215  void _mostrarTodosTodos() {
216    setState(() {
217      _resposta = _textEditingController.text;
218    });
219  }
220
221  void _esconderTodosTodos() {
222    setState(() {
223      _resposta = '';
224    });
225  }
226
227  void _capturarTodosTodos() {
228    setState(() {
229      _resposta = _textEditingController.value.text;
230    });
231  }
232
233  void _limparTodosTodosTodos() {
234    setState(() {
235      _resposta = '';
236      _textEditingController.clear();
237    });
238  }
239
240  void _atualizarTodosTodosTodos() {
241    setState(() {
242      _resposta = _textEditingController.text;
243      _textEditingController.clear();
244    });
245  }
246
247  void _deletarTodosTodosTodos() {
248    setState(() {
249      _resposta = '';
250      _textEditingController.selection =
251        TextSelection.fromPosition(TextPosition(
252          offset: 0,
253        ));
254    });
255  }
256
257  void _inverterTodosTodosTodos() {
258    setState(() {
259      _resposta = _textEditingController.text
260        .toString()
261        .split(' ')
262        .map((String item) {
263          return item[0].toUpperCase() +
264            item.substring(1).toLowerCase();
265        })
266        .join(' ');
267    });
268  }
269
270  void _formatarTodosTodosTodos() {
271    setState(() {
272      _resposta = _textEditingController.text
273        .toString()
274        .split(' ')
275        .map((String item) {
276          return item[0].toUpperCase() +
277            item.substring(1).toLowerCase();
278        })
279        .join(' ');
280    });
281  }
282
283  void _mostrarTodosTodosTodos() {
284    setState(() {
285      _resposta = _textEditingController.text;
286    });
287  }
288
289  void _esconderTodosTodosTodos() {
290    setState(() {
291      _resposta = '';
292    });
293  }
294
295  void _capturarTodosTodosTodos() {
296    setState(() {
297      _resposta = _textEditingController.value.text;
298    });
299  }
299
300  void _atualizarTodosTodosTodosTodos() {
301    setState(() {
302      _resposta = _textEditingController.text;
303      _textEditingController.clear();
304    });
305  }
306
307  void _deletarTodosTodosTodosTodos() {
308    setState(() {
309      _resposta = '';
310      _textEditingController.selection =
311        TextSelection.fromPosition(TextPosition(
312          offset: 0,
313        ));
314    });
315  }
316
317  void _inverterTodosTodosTodosTodos() {
318    setState(() {
319      _resposta = _textEditingController.text
320        .toString()
321        .split(' ')
322        .map((String item) {
323          return item[0].toUpperCase() +
324            item.substring(1).toLowerCase();
325        })
326        .join(' ');
327    });
328  }
329
330  void _formatarTodosTodosTodosTodos() {
331    setState(() {
332      _resposta = _textEditingController.text
333        .toString()
334        .split(' ')
335        .map((String item) {
336          return item[0].toUpperCase() +
337            item.substring(1).toLowerCase();
338        })
339        .join(' ');
340    });
341  }
342
343  void _mostrarTodosTodosTodosTodos() {
344    setState(() {
345      _resposta = _textEditingController.text;
346    });
347  }
348
349  void _esconderTodosTodosTodosTodos() {
350    setState(() {
351      _resposta = '';
352    });
353  }
354
355  void _capturarTodosTodosTodosTodos() {
356    setState(() {
357      _resposta = _textEditingController.value.text;
358    });
359  }
360
361  void _atualizarTodosTodosTodosTodosTodos() {
362    setState(() {
363      _resposta = _textEditingController.text;
364      _textEditingController.clear();
365    });
366  }
367
368  void _deletarTodosTodosTodosTodosTodos() {
369    setState(() {
370      _resposta = '';
371      _textEditingController.selection =
372        TextSelection.fromPosition(TextPosition(
373          offset: 0,
374        ));
375    });
376  }
377
378  void _inverterTodosTodosTodosTodosTodos() {
379    setState(() {
380      _resposta = _textEditingController.text
381        .toString()
382        .split(' ')
383        .map((String item) {
384          return item[0].toUpperCase() +
385            item.substring(1).toLowerCase();
386        })
387        .join(' ');
388    });
389  }
390
391  void _formatarTodosTodosTodosTodosTodos() {
392    setState(() {
393      _resposta = _textEditingController.text
394        .toString()
395        .split(' ')
396        .map((String item) {
397          return item[0].toUpperCase() +
398            item.substring(1).toLowerCase();
399        })
400        .join(' ');
401    });
402  }
403
404  void _mostrarTodosTodosTodosTodosTodos() {
405    setState(() {
406      _resposta = _textEditingController.text;
407    });
408  }
409
410  void _esconderTodosTodosTodosTodosTodos() {
411    setState(() {
412      _resposta = '';
413    });
414  }
415
416  void _capturarTodosTodosTodosTodosTodos() {
417    setState(() {
418      _resposta = _textEditingController.value.text;
419    });
420  }
421
422  void _atualizarTodosTodosTodosTodosTodosTodos() {
423    setState(() {
424      _resposta = _textEditingController.text;
425      _textEditingController.clear();
426    });
427  }
428
429  void _deletarTodosTodosTodosTodosTodosTodos() {
430    setState(() {
431      _resposta = '';
432      _textEditingController.selection =
433        TextSelection.fromPosition(TextPosition(
434          offset: 0,
435        ));
436    });
437  }
438
439  void _inverterTodosTodosTodosTodosTodosTodos() {
440    setState(() {
441      _resposta = _textEditingController.text
442        .toString()
443        .split(' ')
444        .map((String item) {
445          return item[0].toUpperCase() +
446            item.substring(1).toLowerCase();
447        })
448        .join(' ');
449    });
450  }
451
452  void _formatarTodosTodosTodosTodosTodosTodos() {
453    setState(() {
454      _resposta = _textEditingController.text
455        .toString()
456        .split(' ')
457        .map((String item) {
458          return item[0].toUpperCase() +
459            item.substring(1).toLowerCase();
460        })
461        .join(' ');
462    });
463  }
464
465  void _mostrarTodosTodosTodosTodosTodosTodos() {
466    setState(() {
467      _resposta = _textEditingController.text;
468    });
469  }
470
471  void _esconderTodosTodosTodosTodosTodosTodos() {
472    setState(() {
473      _resposta = '';
474    });
475  }
476
477  void _capturarTodosTodosTodosTodosTodosTodos() {
478    setState(() {
479      _resposta = _textEditingController.value.text;
480    });
481  }
482
483  void _atualizarTodosTodosTodosTodosTodosTodosTodos() {
484    setState(() {
485      _resposta = _textEditingController.text;
486      _textEditingController.clear();
487    });
488  }
489
490  void _deletarTodosTodosTodosTodosTodosTodosTodos() {
491    setState(() {
492      _resposta = '';
493      _textEditingController.selection =
494        TextSelection.fromPosition(TextPosition(
495          offset: 0,
496        ));
497    });
498  }
499
500  void _inverterTodosTodosTodosTodosTodosTodosTodos() {
501    setState(() {
502      _resposta = _textEditingController.text
503        .toString()
504        .split(' ')
505        .map((String item) {
506          return item[0].toUpperCase() +
507            item.substring(1).toLowerCase();
508        })
509        .join(' ');
510    });
511  }
512
513  void _formatarTodosTodosTodosTodosTodosTodosTodos() {
514    setState(() {
515      _resposta = _textEditingController.text
516        .toString()
517        .split(' ')
518        .map((String item) {
519          return item[0].toUpperCase() +
520            item.substring(1).toLowerCase();
521        })
522        .join(' ');
523    });
524  }
525
526  void _mostrarTodosTodosTodosTodosTodosTodosTodos() {
527    setState(() {
528      _resposta = _textEditingController.text;
529    });
530  }
531
532  void _esconderTodosTodosTodosTodosTodosTodosTodos() {
533    setState(() {
534      _resposta = '';
535    });
536  }
537
538  void _capturarTodosTodosTodosTodosTodosTodosTodos() {
539    setState(() {
540      _resposta = _textEditingController.value.text;
541    });
542  }
543
544  void _atualizarTodosTodosTodosTodosTodosTodosTodosTodos() {
545    setState(() {
546      _resposta = _textEditingController.text;
547      _textEditingController.clear();
548    });
549  }
550
551  void _deletarTodosTodosTodosTodosTodosTodosTodosTodos() {
552    setState(() {
553      _resposta = '';
554      _textEditingController.selection =
555        TextSelection.fromPosition(TextPosition(
556          offset: 0,
557        ));
558    });
559  }
560
561  void _inverterTodosTodosTodosTodosTodosTodosTodosTodos() {
562    setState(() {
563      _resposta = _textEditingController.text
564        .toString()
565        .split(' ')
566        .map((String item) {
567          return item[0].toUpperCase() +
568            item.substring(1).toLowerCase();
569        })
570        .join(' ');
571    });
572  }
573
574  void _formatarTodosTodosTodosTodosTodosTodosTodosTodos() {
575    setState(() {
576      _resposta = _textEditingController.text
577        .toString()
578        .split(' ')
579        .map((String item) {
580          return item[0].toUpperCase() +
581            item.substring(1).toLowerCase();
582        })
583        .join(' ');
584    });
585  }
586
587  void _mostrarTodosTodosTodosTodosTodosTodosTodosTodos() {
588    setState(() {
589      _resposta = _textEditingController.text;
590    });
591  }
592
593  void _esconderTodosTodosTodosTodosTodosTodosTodosTodos() {
594    setState(() {
595      _resposta = '';
596    });
597  }
598
599  void _capturarTodosTodosTodosTodosTodosTodosTodosTodos() {
600    setState(() {
601      _resposta = _textEditingController.value.text;
602    });
603  }
604
605  void _atualizarTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
606    setState(() {
607      _resposta = _textEditingController.text;
608      _textEditingController.clear();
609    });
610  }
611
612  void _deletarTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
613    setState(() {
614      _resposta = '';
615      _textEditingController.selection =
616        TextSelection.fromPosition(TextPosition(
617          offset: 0,
618        ));
619    });
620  }
621
622  void _inverterTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
623    setState(() {
624      _resposta = _textEditingController.text
625        .toString()
626        .split(' ')
627        .map((String item) {
628          return item[0].toUpperCase() +
629            item.substring(1).toLowerCase();
630        })
631        .join(' ');
632    });
633  }
634
635  void _formatarTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
636    setState(() {
637      _resposta = _textEditingController.text
638        .toString()
639        .split(' ')
640        .map((String item) {
641          return item[0].toUpperCase() +
642            item.substring(1).toLowerCase();
643        })
644        .join(' ');
645    });
646  }
647
648  void _mostrarTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
649    setState(() {
650      _resposta = _textEditingController.text;
651    });
652  }
653
654  void _esconderTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
655    setState(() {
656      _resposta = '';
657    });
658  }
659
660  void _capturarTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
661    setState(() {
662      _resposta = _textEditingController.value.text;
663    });
664  }
665
666  void _atualizarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
667    setState(() {
668      _resposta = _textEditingController.text;
669      _textEditingController.clear();
670    });
671  }
672
673  void _deletarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
674    setState(() {
675      _resposta = '';
676      _textEditingController.selection =
677        TextSelection.fromPosition(TextPosition(
678          offset: 0,
679        ));
680    });
681  }
682
683  void _inverterTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
684    setState(() {
685      _resposta = _textEditingController.text
686        .toString()
687        .split(' ')
688        .map((String item) {
689          return item[0].toUpperCase() +
690            item.substring(1).toLowerCase();
691        })
692        .join(' ');
693    });
694  }
695
696  void _formatarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
697    setState(() {
698      _resposta = _textEditingController.text
699        .toString()
700        .split(' ')
701        .map((String item) {
702          return item[0].toUpperCase() +
703            item.substring(1).toLowerCase();
704        })
705        .join(' ');
706    });
707  }
708
709  void _mostrarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
710    setState(() {
711      _resposta = _textEditingController.text;
712    });
713  }
714
715  void _esconderTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
716    setState(() {
717      _resposta = '';
718    });
719  }
720
721  void _capturarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
722    setState(() {
723      _resposta = _textEditingController.value.text;
724    });
725  }
726
727  void _atualizarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
728    setState(() {
729      _resposta = _textEditingController.text;
730      _textEditingController.clear();
731    });
732  }
733
734  void _deletarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
735    setState(() {
736      _resposta = '';
737      _textEditingController.selection =
738        TextSelection.fromPosition(TextPosition(
739          offset: 0,
740        ));
741    });
742  }
743
744  void _inverterTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
745    setState(() {
746      _resposta = _textEditingController.text
747        .toString()
748        .split(' ')
749        .map((String item) {
750          return item[0].toUpperCase() +
751            item.substring(1).toLowerCase();
752        })
753        .join(' ');
754    });
755  }
756
757  void _formatarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
758    setState(() {
759      _resposta = _textEditingController.text
760        .toString()
761        .split(' ')
762        .map((String item) {
763          return item[0].toUpperCase() +
764            item.substring(1).toLowerCase();
765        })
766        .join(' ');
767    });
768  }
769
770  void _mostrarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
771    setState(() {
772      _resposta = _textEditingController.text;
773    });
774  }
775
776  void _esconderTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
777    setState(() {
778      _resposta = '';
779    });
780  }
781
782  void _capturarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
783    setState(() {
784      _resposta = _textEditingController.value.text;
785    });
786  }
787
788  void _atualizarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
789    setState(() {
790      _resposta = _textEditingController.text;
791      _textEditingController.clear();
792    });
793  }
794
795  void _deletarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
796    setState(() {
797      _resposta = '';
798      _textEditingController.selection =
799        TextSelection.fromPosition(TextPosition(
800          offset: 0,
801        ));
802    });
803  }
804
805  void _inverterTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
806    setState(() {
807      _resposta = _textEditingController.text
808        .toString()
809        .split(' ')
810        .map((String item) {
811          return item[0].toUpperCase() +
812            item.substring(1).toLowerCase();
813        })
814        .join(' ');
815    });
816  }
817
818  void _formatarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
819    setState(() {
820      _resposta = _textEditingController.text
821        .toString()
822        .split(' ')
823        .map((String item) {
824          return item[0].toUpperCase() +
825            item.substring(1).toLowerCase();
826        })
827        .join(' ');
828    });
829  }
830
831  void _mostrarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
832    setState(() {
833      _resposta = _textEditingController.text;
834    });
835  }
836
837  void _esconderTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
838    setState(() {
839      _resposta = '';
840    });
841  }
842
843  void _capturarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
844    setState(() {
845      _resposta = _textEditingController.value.text;
846    });
847  }
848
849  void _atualizarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
850    setState(() {
851      _resposta = _textEditingController.text;
852      _textEditingController.clear();
853    });
854  }
855
856  void _deletarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
857    setState(() {
858      _resposta = '';
859      _textEditingController.selection =
860        TextSelection.fromPosition(TextPosition(
861          offset: 0,
862        ));
863    });
864  }
865
866  void _inverterTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
867    setState(() {
868      _resposta = _textEditingController.text
869        .toString()
870        .split(' ')
871        .map((String item) {
872          return item[0].toUpperCase() +
873            item.substring(1).toLowerCase();
874        })
875        .join(' ');
876    });
877  }
878
879  void _formatarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
880    setState(() {
881      _resposta = _textEditingController.text
882        .toString()
883        .split(' ')
884        .map((String item) {
885          return item[0].toUpperCase() +
886            item.substring(1).toLowerCase();
887        })
888        .join(' ');
889    });
890  }
891
892  void _mostrarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
893    setState(() {
894      _resposta = _textEditingController.text;
895    });
896  }
897
898  void _esconderTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
899    setState(() {
900      _resposta = '';
901    });
902  }
903
904  void _capturarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
905    setState(() {
906      _resposta = _textEditingController.value.text;
907    });
908  }
909
910  void _atualizarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
911    setState(() {
912      _resposta = _textEditingController.text;
913      _textEditingController.clear();
914    });
915  }
916
917  void _deletarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
918    setState(() {
919      _resposta = '';
920      _textEditingController.selection =
921        TextSelection.fromPosition(TextPosition(
922          offset: 0,
923        ));
924    });
925  }
926
927  void _inverterTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
928    setState(() {
929      _resposta = _textEditingController.text
930        .toString()
931        .split(' ')
932        .map((String item) {
933          return item[0].toUpperCase() +
934            item.substring(1).toLowerCase();
935        })
936        .join(' ');
937    });
938  }
939
940  void _formatarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
941    setState(() {
942      _resposta = _textEditingController.text
943        .toString()
944        .split(' ')
945        .map((String item) {
946          return item[0].toUpperCase() +
947            item.substring(1).toLowerCase();
948        })
949        .join(' ');
950    });
951  }
952
953  void _mostrarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
954    setState(() {
955      _resposta = _textEditingController.text;
956    });
957  }
958
959  void _esconderTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
960    setState(() {
961      _resposta = '';
962    });
963  }
964
965  void _capturarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
966    setState(() {
967      _resposta = _textEditingController.value.text;
968    });
969  }
970
971  void _atualizarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
972    setState(() {
973      _resposta = _textEditingController.text;
974      _textEditingController.clear();
975    });
976  }
977
978  void _deletarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
979    setState(() {
980      _resposta = '';
981      _textEditingController.selection =
982        TextSelection.fromPosition(TextPosition(
983          offset: 0,
984        ));
985    });
986  }
987
988  void _inverterTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
989    setState(() {
990      _resposta = _textEditingController.text
991        .toString()
992        .split(' ')
993        .map((String item) {
994          return item[0].toUpperCase() +
995            item.substring(1).toLowerCase();
996        })
997        .join(' ');
998    });
999  }
999
1000  void _formatarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
1001    setState(() {
1002      _resposta = _textEditingController.text
1003        .toString()
1004        .split(' ')
1005        .map((String item) {
1006          return item[0].toUpperCase() +
1007            item.substring(1).toLowerCase();
1008        })
1009        .join(' ');
1010    });
1011  }
1012
1013  void _mostrarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
1014    setState(() {
1015      _resposta = _textEditingController.text;
1016    });
1017  }
1018
1019  void _esconderTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
1020    setState(() {
1021      _resposta = '';
1022    });
1023  }
1024
1025  void _capturarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
1026    setState(() {
1027      _resposta = _textEditingController.value.text;
1028    });
1029  }
1030
1031  void _atualizarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
1032    setState(() {
1033      _resposta = _textEditingController.text;
1034      _textEditingController.clear();
1035    });
1036  }
1037
1038  void _deletarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
1039    setState(() {
1040      _resposta = '';
1041      _textEditingController.selection =
1042        TextSelection.fromPosition(TextPosition(
1043          offset: 0,
1044        ));
1045    });
1046  }
1047
1048  void _inverterTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
1049    setState(() {
1050      _resposta = _textEditingController.text
1051        .toString()
1052        .split(' ')
1053        .map((String item) {
1054          return item[0].toUpperCase() +
1055            item.substring(1).toLowerCase();
1056        })
1057        .join(' ');
1058    });
1059  }
1060
1061  void _formatarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
1062    setState(() {
1063      _resposta = _textEditingController.text
1064        .toString()
1065        .split(' ')
1066        .map((String item) {
1067          return item[0].toUpperCase() +
1068            item.substring(1).toLowerCase();
1069        })
1070        .join(' ');
1071    });
1072  }
1073
1074  void _mostrarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
1075    setState(() {
1076      _resposta = _textEditingController.text;
1077    });
1078  }
1079
1080  void _esconderTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
1081    setState(() {
1082      _resposta = '';
1083    });
1084  }
1085
1086  void _capturarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
1087    setState(() {
1088      _resposta = _textEditingController.value.text;
1089    });
1090  }
1091
1092  void _atualizarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
1093    setState(() {
1094      _resposta = _textEditingController.text;
1095      _textEditingController.clear();
1096    });
1097  }
1098
1099  void _deletarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
1100    setState(() {
1101      _resposta = '';
1102      _textEditingController.selection =
1103        TextSelection.fromPosition(TextPosition(
1104          offset: 0,
1105        ));
1106    });
1107  }
1108
1109  void _inverterTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
1110    setState(() {
1111      _resposta = _textEditingController.text
1112        .toString()
1113        .split(' ')
1114        .map((String item) {
1115          return item[0].toUpperCase() +
1116            item.substring(1).toLowerCase();
1117        })
1118        .join(' ');
1119    });
1120  }
1121
1122  void _formatarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
1123    setState(() {
1124      _resposta = _textEditingController.text
1125        .toString()
1126        .split(' ')
1127        .map((String item) {
1128          return item[0].toUpperCase() +
1129            item.substring(1).toLowerCase();
1130        })
1131        .join(' ');
1132    });
1133  }
1134
1135  void _mostrarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
1136    setState(() {
1137      _resposta = _textEditingController.text;
1138    });
1139  }
1140
1141  void _esconderTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
1142    setState(() {
1143      _resposta = '';
1144    });
1145  }
1146
1147  void _capturarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
1148    setState(() {
1149      _resposta = _textEditingController.value.text;
1150    });
1151  }
1152
1153  void _atualizarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
1154    setState(() {
1155      _resposta = _textEditingController.text;
1156      _textEditingController.clear();
1157    });
1158  }
1159
1160  void _deletarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
1161    setState(() {
1162      _resposta = '';
1163      _textEditingController.selection =
1164        TextSelection.fromPosition(TextPosition(
1165          offset: 0,
1166        ));
1167    });
1168  }
1169
1170  void _inverterTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
1171    setState(() {
1172      _resposta = _textEditingController.text
1173        .toString()
1174        .split(' ')
1175        .map((String item) {
1176          return item[0].toUpperCase() +
1177            item.substring(1).toLowerCase();
1178        })
1179        .join(' ');
1180    });
1181  }
1182
1183  void _formatarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
1184    setState(() {
1185      _resposta = _textEditingController.text
1186        .toString()
1187        .split(' ')
1188        .map((String item) {
1189          return item[0].toUpperCase() +
1190            item.substring(1).toLowerCase();
1191        })
1192        .join(' ');
1193    });
1194  }
1195
1196  void _mostrarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
1197    setState(() {
1198      _resposta = _textEditingController.text;
1199    });
1200  }
1201
1202  void _esconderTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
1203    setState(() {
1204      _resposta = '';
1205    });
1206  }
1207
1208  void _capturarTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodosTodos() {
1209    setState(() {
1210      _resposta = _textEditingController.value.text;
1211    });
1212  }
1213
121
```

## Entrada de Dados

Um campo de texto permite que o usuário insira texto, com teclado de hardware ou com teclado na tela.

O campo de texto chama o retorno de chamada `onChanged` sempre que o usuário altera o texto no campo. Se o usuário indicar que acabou de digitar no campo (por exemplo, pressionando um botão no teclado virtual), o campo de texto chama o retorno de chamada `onSubmitted`.

Para controlar o texto que é exibido no campo de texto, use o controlador. Por exemplo, para definir o valor inicial do campo de texto, use um controlador que já contenha algum texto. O controlador também pode controlar a seleção e a região de composição (e observar as mudanças no texto, seleção e região de composição).

```
73         child: Text('Salvar')
74     ), // ElevatedButton
75     Padding(
76         padding: const EdgeInsets.only(top: 20),
77         child: Text(_resposta, style:
78             TextStyle(
79                 decoration: TextDecoration.none
80             ) // TextStyle
81         ), // Text
82         ) // Padding
83     ],
84     ), // Column
85 ); // Scaffold
86 }
87 }
```



## Stateless

Widgets sem estado não requerem estado mutável, ou seja, é imutável . Em palavras simples, Widgets sem estado não podem mudar seu estado durante o tempo de execução do aplicativo, o que significa que os widgets não podem ser redesenhados enquanto o aplicativo está em ação.

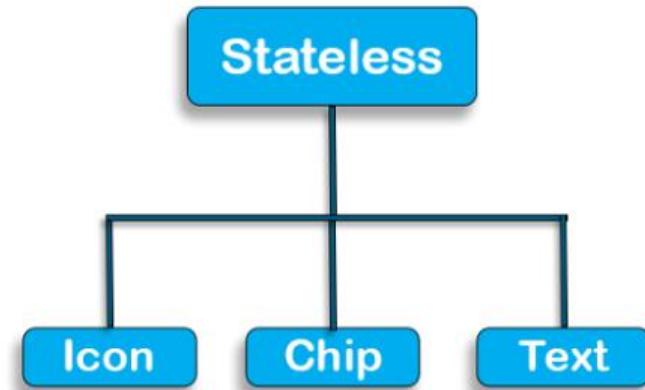
No widget sem estado, o método “ build ” pode ser chamado apenas UMA VEZ enquanto o aplicativo está em ação, que é responsável por desenhar os widgets na tela do dispositivo.

## Stateful

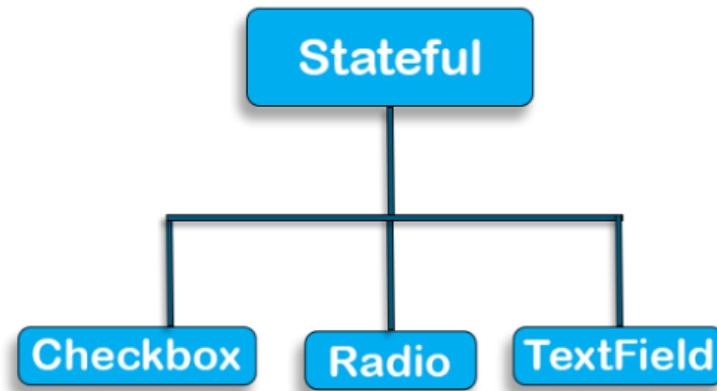
Widgets com estado têm um estado mutável, ou seja, são mutáveis e podem ser desenhados várias vezes durante seu tempo de vida.

Eles são os widgets que podem mudar de estado várias vezes e podem ser redesenhados na tela qualquer número de vezes enquanto o aplicativo está em ação.

Exemplos de widgets sem estado são os seguintes:



Exemplos de widgets com estado são os seguintes:

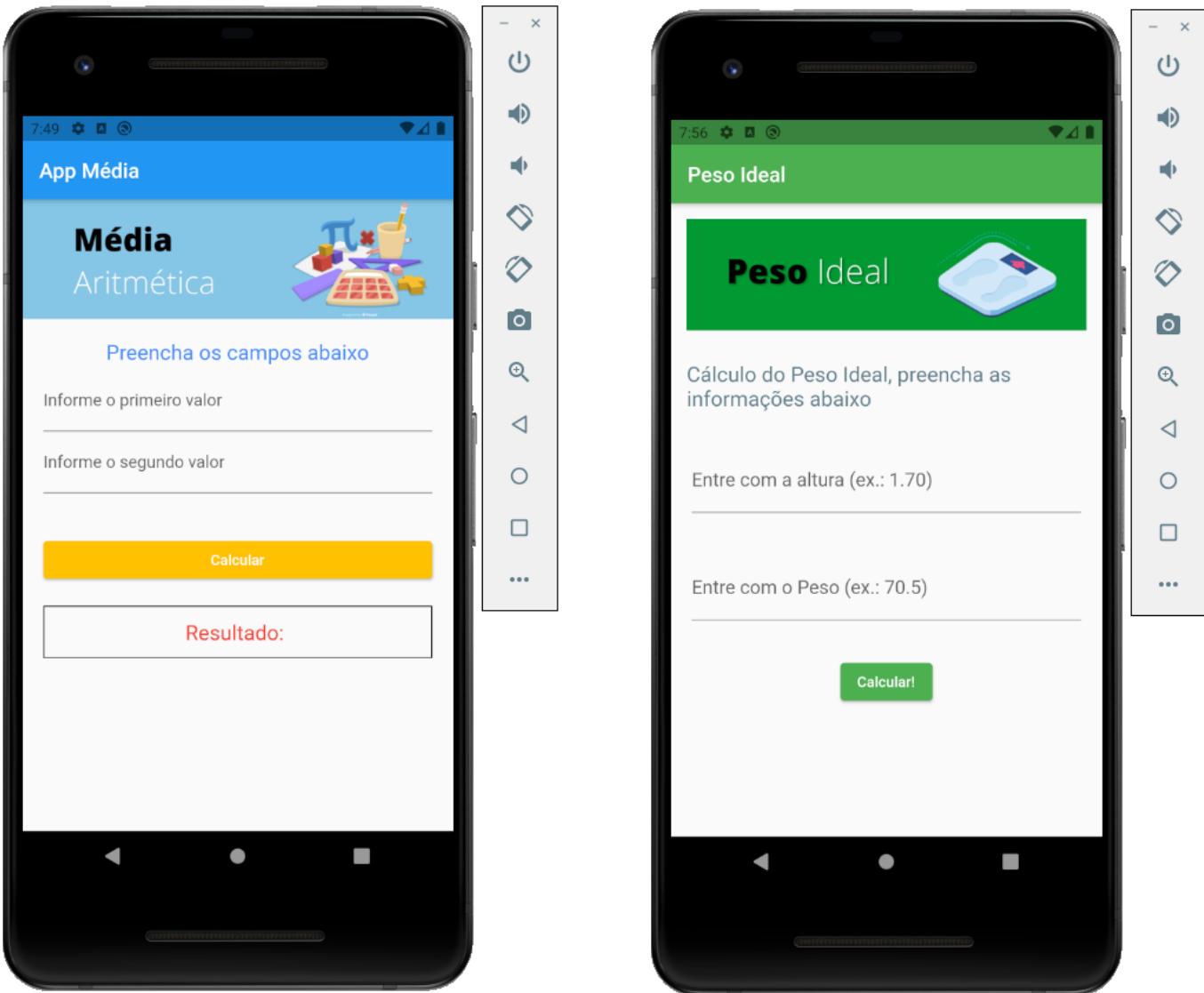


# Atividade 1 Flutter

Partindo do pressuposto de que vocês receberam um job de um cliente e baseando-se no design ao lado, fornecido por ele, desenvolva as aplicações :

## Observações:

Enviar a pasta lib, a pasta images o arquivo .yaml compactados para o email do professor.



## Gesture Detector()

Se você precisar detectar que tipo de gesto está acontecendo em um widget, a maneira mais fácil é envolver o widget como filho de GestureDetector.

Ele pode ter várias funções de retorno de chamada, cada uma para diferentes tipos de gestos. Ele também suporta vários tipos de gestos, como toque, toque secundário, toque longo, arrasto vertical / horizontal, panorâmica e escala.

Usaremos o parâmetro onTap(), mas fique a vontade para aprender as outras funcionalidades dessa classe,



```
lib > main.dart > _GestureState > build
1 //Gesture Detector
2 //Detectando toque na tela
3
4 import 'package:flutter/material.dart';
5
6 Run | Debug | Profile
7 main() => runApp(MaterialApp(
8   home: Gesture(),
9 )); // MaterialApp
10
11 class Gesture extends StatefulWidget {
12   @override
13   _GestureState createState() => _GestureState();
14 }
15
16 class _GestureState extends State<Gesture> {
17
18   String _personagem = '';
19   var _imagem = AssetImage('images/imagem.png');
20
21   personagemEscolhido(String personagem) {
22     if (personagem == 'joao') {
23       _personagem = 'Você escolheu o João!';
24       _imagem = AssetImage('images/jo.png');
25     } else if (personagem == 'juliana') {
26       _personagem = 'Você escolheu a Juliana!';
27       _imagem = AssetImage('images/ju.png');
28     } else {
29       _personagem = 'Você escolheu o José!';
30       _imagem = AssetImage('images/ze.png');
31     }
32   }
33
34   @override
35   Widget build(BuildContext context) {
36     return Scaffold(
37       appBar: AppBar(
38         title: Text('Gesture App'),
39         backgroundColor: Color(0xffcd853f),
40       ), // AppBar
41       body: Padding(
42         padding: EdgeInsets.all(20),
43         //Primeiro uma column
44         child: Column(
45           children: [
46             //Agora uma Row
```

## Gesture Detector()

Se você precisar detectar que tipo de gesto está acontecendo em um widget, a maneira mais fácil é envolver o widget como filho de GestureDetector.

Ele pode ter várias funções de retorno de chamada, cada uma para diferentes tipos de gestos. Ele também suporta vários tipos de gestos, como toque, toque secundário, toque longo, arrasto vertical / horizontal, panorâmica e escala.

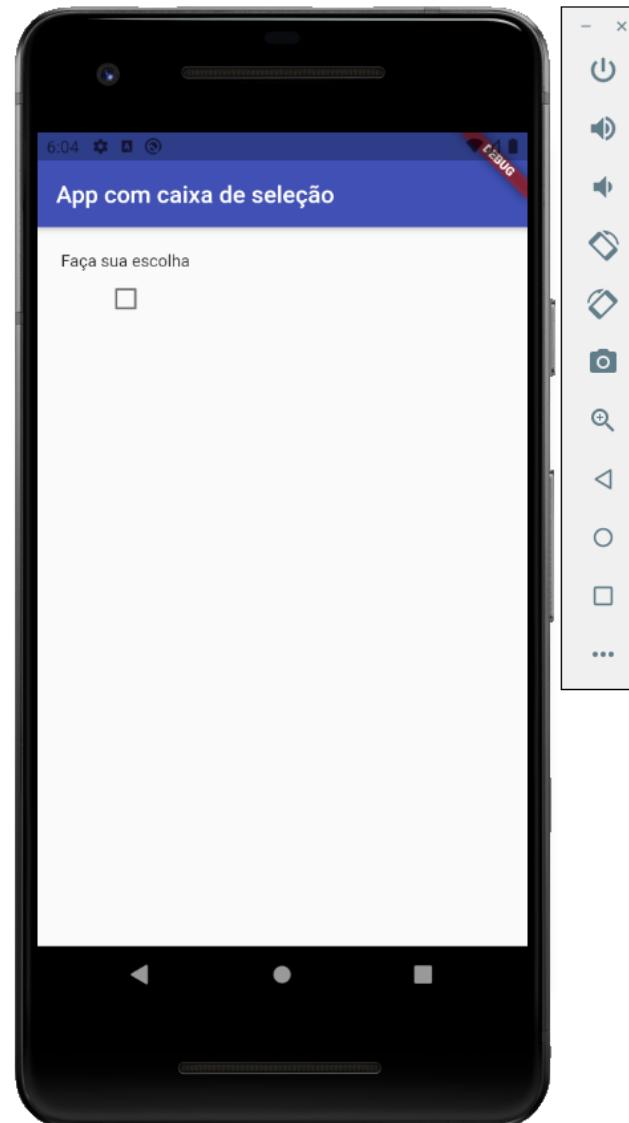
Usaremos o parâmetro onTap(), mas fique a vontade para aprender as outras funcionalidades dessa classe,

```
45   Row(
46     mainAxisSize: MainAxisSize.spaceEvenly,
47     children: [
48       //Detectando o toque na imagem
49       GestureDetector(
50         //Evento de 1 toque na tela
51         onTap: () {
52           setState(() {
53             personagemEscolhido('joao');
54           });
55         },
56         //Inserção da imagem
57         child: Image.asset(
58           'images/Joao.png',
59           width: 100,
60         ), // Image.asset
61       ), // GestureDetector
62       GestureDetector(
63         onTap: () {
64           setState(() {
65             personagemEscolhido('juliana');
66           });
67         },
68         child: Image.asset(
69           'images/Juliana.png',
70           width: 100,
71         ), // Image.asset
72       ), // GestureDetector
73       GestureDetector(
74         onTap: () {
75           setState(() {
76             personagemEscolhido('jose');
77           });
78         },
79         child: Image.asset(
80           'images/Jose.png',
81           width: 100,
82         ), // Image.asset
83       ), // GestureDetector
84     ],
85   ), // Row
86   //mais uma Column
87   Padding(
88     padding: const EdgeInsets.only(
89       top: 20,
90     ), // EdgeInsets.only
91     child: Column(
92       children: [
93         Text(
94           'Escolha um personagem',
95           style: TextStyle(
96             fontSize: 20,
97           ), // TextStyle
98         ), // Text
99         Padding(
100          padding: EdgeInsets.only(top: 20, bottom: 20),
101        ), // EdgeInsets.only
102        child: Column(
103          children: [
104            Image(
105              image: _imagem,
106              width: 100,
107            ), // Image
108          ],
109        ), // Column
110      ), // Padding
111      Padding(
112        padding: const EdgeInsets.only(top: 20),
113      ), // EdgeInsets.only
114      child: Text(
115        _personagem,
116        style: TextStyle(
117          fontSize: 30,
118          color: Colors.orange,
119        ), // TextStyle
120      ), // Text
121    ), // Padding
122  ],
123  ), // Column
124  ), // Padding
125  ],
126  ), // Column
127  ), // Padding
128 ); // Scaffold
129 }
130 }
```

## Checkbox()

A caixa de seleção não mantém nenhum estado. Em vez disso, quando o estado da caixa de seleção muda, o widget chama o retorno de chamada onChanged.

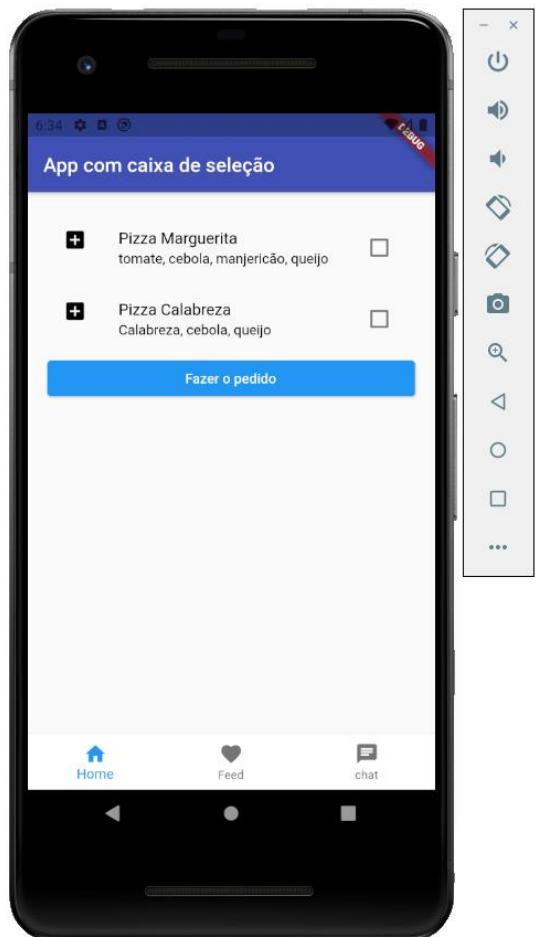
A maioria dos widgets que usam uma caixa de seleção ouvirá o retorno de chamada onChanged e reconstruirá a caixa de seleção com um novo valor para atualizar sua aparência.



```
lib > main14_checkbox_1.dart > ...
1 import 'package:flutter/material.dart';
Run | Debug | Profile
2 main() => runApp(MaterialApp(
3   |   home: MeuCheckbox(),
4   |));
// MaterialApp
5
6 class MeuCheckbox extends StatefulWidget {
7   @override
8   _MeuCheckboxState createState() => _MeuCheckboxState();
9 }
10
11 class _MeuCheckboxState extends State<MeuCheckbox> {
12   bool _selecaoCheckbox = false;
13
14   @override
15   Widget build(BuildContext context) {
16     return Scaffold(
17       appBar: AppBar(
18         title: Text('App com caixa de seleção'),
19         backgroundColor: Colors.indigo,
20       ),
21       body: Padding(
22         padding: const EdgeInsets.all(20.0),
23         child: Column(
24           children: [
25             Text('Faça sua escolha'),
26             Checkbox(
27               //valor inicial do checkbox
28               value: _selecaoCheckbox, //desmarcado
29               onChanged: (bool? valor) {
30                 setState(() {
31                   _selecaoCheckbox = valor!;
32                 });
33               },
34             ],
35           ), // Column
36         ), // Padding
37       ); // Scaffold
38     }
39 }
```

# CheckboxListTile()

CheckboxListTile é um widget que combina uma caixa de seleção e um bloco de lista. Ele permite que você crie uma caixa de seleção junto com o título, subtítulo e ícone sem a necessidade de criar um widget separado para cada parte.

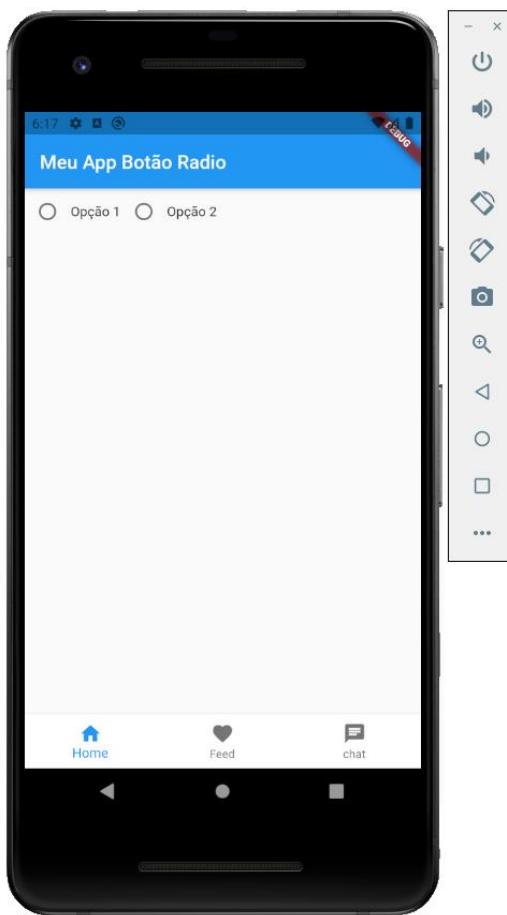


```
1 import 'package:flutter/material.dart';
Run | Debug | Profile
2 main() => runApp(MaterialApp(
3   home: MeuCheckbox(),
4 )); // MaterialApp
5
6 class MeuCheckbox extends StatefulWidget {
7   @override
8   _MeuCheckboxState createState() => _MeuCheckboxState();
9 }
10
11 class _MeuCheckboxState extends State<MeuCheckbox> {
12   bool _pizzaMarguerita = false;
13   bool _pizzaCalabresa = false;
14
15   @override
16   Widget build(BuildContext context) {
17     return Scaffold(
18       appBar: AppBar(
19         title: Text('App com caixa de seleção'),
20         backgroundColor: Colors.indigo,
21       ), // AppBar
22       body: Padding(
23         padding: const EdgeInsets.all(20.0),
24         child: Column(
25           //Para esticar o conteúdo
26           crossAxisAlignment: CrossAxisAlignment.stretch,
27           children: [
28
29             CheckboxListTile(
30               title: Text('Pizza Marguerita'),
31               subtitle: Text('tomate, cebola, manjericão, queijo'),
32               //Cor do checkbox
33               activeColor: Colors.black,
34               //cor do texto
35               selected: true,
36               //definindo um ícone
37               secondary: Icon(Icons.add_box),
38               value: _pizzaMarguerita,
39               onChanged: (bool? valor) {
40                 setState(() {
41                   _pizzaMarguerita = valor!;
42                 });
43               }, // CheckboxListTile
44
45             CheckboxListTile(
46               title: Text('Pizza Calabresa'),
47               subtitle: Text('Calabresa, cebola, queijo'),
48               //Cor do checkbox
49               activeColor: Colors.black,
50               //cor do texto
51               selected: true,
52               //definindo um ícone
53               secondary: Icon(Icons.add_box),
54               value: _pizzaCalabresa,
55               onChanged: (bool? valor) {
56                 setState(() {
57                   _pizzaCalabresa = valor!;
58                 });
59               }, // CheckboxListTile
60               ElevatedButton(
61                 onPressed: () {
62                   print('Pedido:');
63                   print('Pizza Marguerita: ${_pizzaMarguerita}');
64                   print('PizzaCalabresa: ${_pizzaCalabresa}');
65                 },
66                 child: Text('Fazer o pedido'),
67               ), // ElevatedButton
68             ), // Column
69           ), // Padding
70           bottomNavigationBar: BottomNavigationBar(
71             backgroundColor: Colors.white,
72             currentIndex: 0,
73             items: [
74               BottomNavigationBarItem(
75                 icon: Icon(Icons.home),
76                 label: 'Home',
77               ), // BottomNavigationBarItem
78               BottomNavigationBarItem(
79                 icon: Icon(Icons.favorite),
80                 label: 'Feed',
81               ), // BottomNavigationBarItem
82               BottomNavigationBarItem(
83                 icon: Icon(Icons.chat),
84                 label: 'chat',
85               ), // BottomNavigationBarItem
86             ], // BottomNavigationBar
87           ); // Scaffold
88         }
89       } // Scaffold

```

# Radio()

Usado para selecionar entre vários valores mutuamente exclusivos. Quando um botão de opção em um grupo é selecionado, os outros botões de opção no grupo deixam de ser selecionados. Os valores são do tipo True/False.

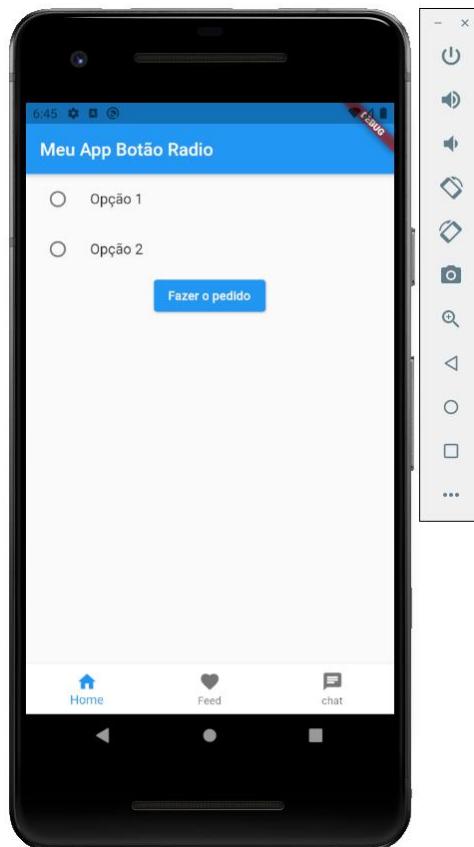


```
1 import 'package:flutter/material.dart';
2 Run | Debug | Profile
3 void main() => runApp(MaterialApp(
4   home: BotaoRadio(),
5 )); // MaterialApp
6
7 class BotaoRadio extends StatefulWidget {
8   @override
9   _BotaoRadioState createState() => _BotaoRadioState();
10
11 class _BotaoRadioState extends State<BotaoRadio> {
12   int? _opcao;
13
14   @override
15   Widget build(BuildContext context) {
16     return Scaffold(
17       appBar: AppBar(
18         title: Text('Meu App Botão Radio'),
19         backgroundColor: Colors.blue,
20     ), // AppBar
21     body: Container(
22       child: Row(
23         children: [
24           Radio(
25             value: 1,
26             groupValue: _opcao,
27             onChanged: (int? selecao) {
28               print('Seleção: $selecao');
29               setState(() {
30                 _opcao = selecao;
31               });
32             }, // Radio
33             Text('Opção 1'),
34             Radio(
35               value: 2,
36               groupValue: _opcao,
37               onChanged: (int? selecao) {
38                 print('Seleção: $selecao');
39                 setState(() {
40                   _opcao = selecao;
41                 });
42               }, // Radio
43               Text('Opção 2')
44             ],
45           ), // Row
46         ], // Container
47       bottomNavigationBar: BottomNavigationBar(
48         backgroundColor: Colors.white,
49         currentIndex: 0,
50         items: [
51           BottomNavigationBarItem(
52             icon: Icon(Icons.home),
53             label: 'Home',
54         ), // BottomNavigationBarItem
55         BottomNavigationBarItem(
56             icon: Icon(Icons.favorite),
57             label: 'Feed',
58         ), // BottomNavigationBarItem
59         BottomNavigationBarItem(
60             icon: Icon(Icons.chat),
61             label: 'chat',
62         ), // BottomNavigationBarItem
63     ], // BottomNavigationBar
64   ); // Scaffold
65 }
66 }
```

## RadioListTile()

O RadioListTile nada mais é que um ListTile contendo um botão de rádio por padrão. Todo o bloco da lista é interativo: tocar em qualquer lugar do bloco seleciona o botão de rádio.

As propriedades value, groupValue, onChanged e activeColor deste widget são idênticas às propriedades com nomes semelhantes no widget Radio.



```
1 import 'package:flutter/material.dart';
Run | Debug | Profile
2 void main() => runApp(MaterialApp(
3   home: BotaoRadio(),
4 )); // MaterialApp
5
6 class BotaoRadio extends StatefulWidget {
7   @override
8   _BotaoRadioState createState() => _BotaoRadioState();
9 }
10
11 class _BotaoRadioState extends State<BotaoRadio> {
12   //Variável para capturar o valor do botão
13   int? _opcao;
14
15   @override
16   Widget build(BuildContext context) {
17     return Scaffold(
18       appBar: AppBar(
19         title: Text('Meu App Botão Radio'),
20         backgroundColor: Colors.blue,
21       ), // AppBar
22       body: Container(
23         child: Column(
24           children: [
25             RadioListTile(
26               title: Text('Opção 1'),
27               value: 1,
28               groupValue: _opcao,
29               onChanged: (int? selecao) {
30                 setState(() {
31                   _opcao = selecao;
32                 });
33               }, // RadioListTile
34             RadioListTile(
35               title: Text('Opção 2'),
36               value: 2,
37               groupValue: _opcao,
38               onChanged: (int? selecao) {
39                 setState(() {
40                   _opcao = selecao;
41                 });
42               }, // RadioListTile
43             ],
44           ],
45         ),
46       ), // Column
47     ), // Container
48     bottomNavigationBar: BottomNavigationBar(
49       backgroundColor: Colors.white,
50       currentIndex: 0,
51       items: [
52         BottomNavigationBarItem(
53           icon: Icon(Icons.home),
54           label: 'Home',
55         ), // BottomNavigationBarItem
56         BottomNavigationBarItem(
57           icon: Icon(Icons.favorite),
58           label: 'Feed',
59         ), // BottomNavigationBarItem
60         BottomNavigationBarItem(
61           icon: Icon(Icons.chat),
62           label: 'chat',
63         ), // BottomNavigationBarItem
64       ],
65     ), // BottomNavigationBar
66   ); // Scaffold
67 }
```

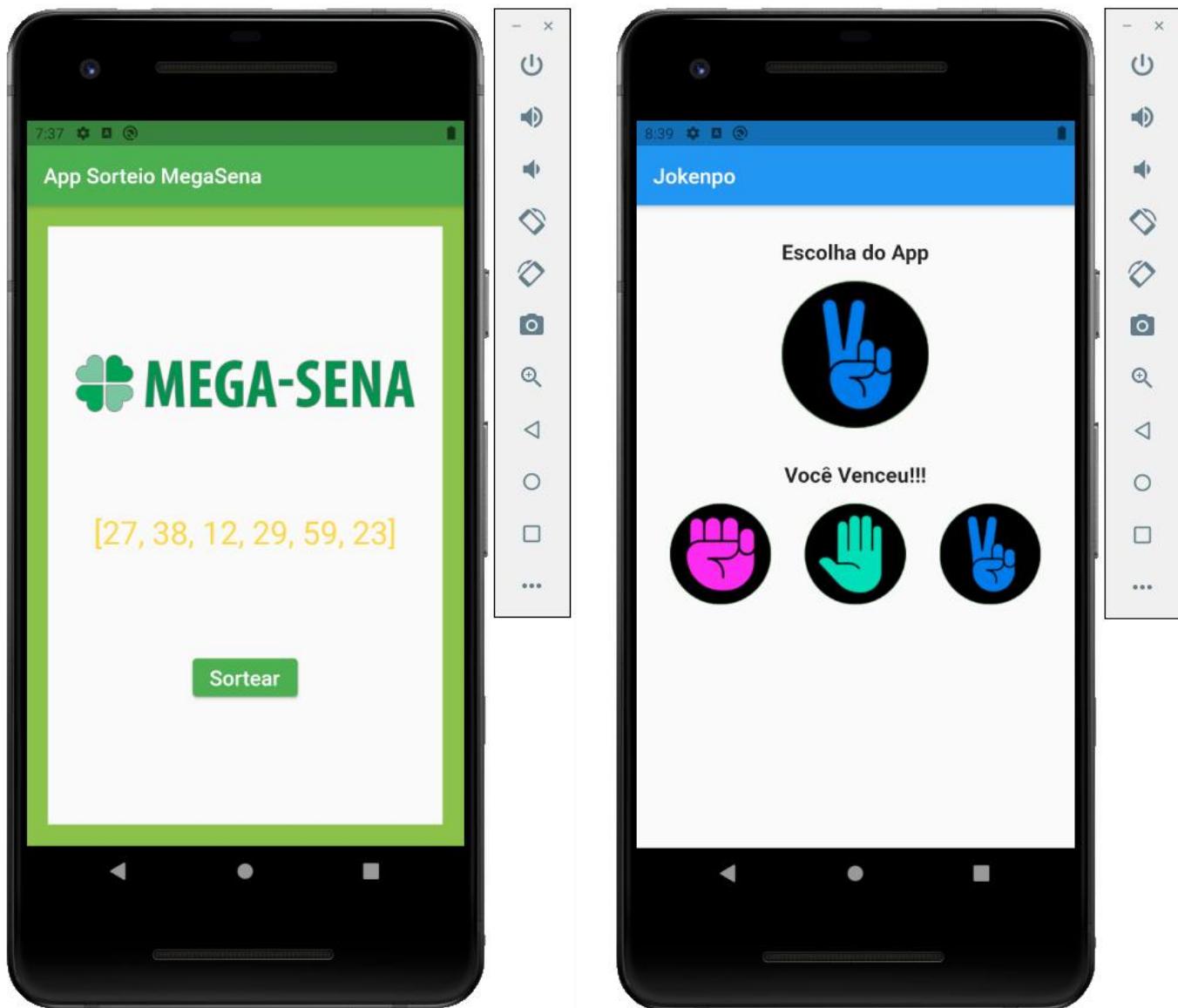
```
44
45   ElevatedButton(
46     onPressed: () {
47       print('Seleção:');
48       print('Opção escolhida: $_opcao');
49     },
50     child: Text('Fazer o pedido')
51   ) // ElevatedButton
52 ], // Column
53 ), // Container
54 bottomNavigationBar: BottomNavigationBar(
55   backgroundColor: Colors.white,
56   currentIndex: 0,
57   items: [
58     BottomNavigationBarItem(
59       icon: Icon(Icons.home),
60       label: 'Home',
61     ), // BottomNavigationBarItem
62     BottomNavigationBarItem(
63       icon: Icon(Icons.favorite),
64       label: 'Feed',
65     ), // BottomNavigationBarItem
66     BottomNavigationBarItem(
67       icon: Icon(Icons.chat),
68       label: 'chat',
69     ), // BottomNavigationBarItem
70   ],
71 ), // BottomNavigationBar
72 } // Scaffold
73 }
```

## Atividade 2 Flutter

Partindo do pressuposto de que vocês receberam um job de um cliente e baseando-se no design ao lado, fornecido por ele, desenvolva as aplicações :

### Observações:

- Os números da mega não pode se repetir
- Enviar a pasta lib, a pasta images
- o arquivo .yaml compactados para o email do professor.





Siga o Senac em Minas nas Redes Sociais:

