

# Programação de Dispositivos Móveis

Prof. Rogério B. de Andrade

# Flutter: Entrada de Dados



The image shows a Flutter application window titled "Sample Input" with a blue header bar. A red "DEBUG" banner is in the top right corner. The main content area has a light gray background and contains a centered "Sample Input" title. Below the title is a text input field with a person icon and the placeholder text "Entre com nome". Underneath the input field are two blue buttons: "Enviar" and "Cancelar". At the bottom of the form is the label "Resposta:".

Disponível em:

[https://docs.google.com/document/d/1BKZA0rYerPzHndB-jxJ\\_uAyUzWK1KELfUHjebKyDRIA/edit?usp=sharing](https://docs.google.com/document/d/1BKZA0rYerPzHndB-jxJ_uAyUzWK1KELfUHjebKyDRIA/edit?usp=sharing)

# Flutter: StatelessWidget

É uma classe em Flutter utilizada para criar widgets que não têm estado interno e cuja aparência é baseada somente em suas propriedades de entrada (suas propriedades são definidas uma vez e permanecem durante toda a vida do widget). É útil para criar partes da interface do usuário que não precisam de interatividade ou mudança de estado, como textos estáticos, ícones e imagens.

# Flutter: StatelessWidget

É uma classe em Flutter utilizada para criar widgets que não têm estado interno e cuja aparência é baseada somente em suas propriedades de entrada (suas propriedades são definidas uma vez e permanecem durante toda a vida do widget). É útil para criar partes da interface do usuário que não precisam de interatividade ou mudança de estado, como textos estáticos, ícones e imagens.

# Flutter: StatelessWidget

- › Em Flutter, a classe `State<T>` é usada em conjunto com a classe `StatefulWidget` para definir a interface do usuário e gerenciar o estado interno mutável do widget.
- › A classe `State<T>` é uma classe genérica, onde `T` é o tipo do `StatefulWidget` correspondente. Por exemplo, se o `StatefulWidget` for chamado `MyWidget`, a classe `State` correspondente será chamada `MyWidgetState`. A classe `State` estende a classe `StatefulWidget` e, portanto, é responsável por gerenciar o estado do widget.
- › Ao criar um `StatefulWidget` em Flutter, é necessário criar uma nova instância da classe `State` associada ao `StatefulWidget`, usando o método `createState()`. Em seguida, a instância da classe `State` é vinculada ao `StatefulWidget` usando a palavra-chave `of` e o método `BuildContext`, conforme mostrado abaixo:

# Flutter: StatefulWidget

Em Flutter, a criação de um StatefulWidget envolve duas classes: a classe StatefulWidget propriamente dita e a classe State associada. O método createState() é responsável por criar a instância da classe State.

Quando um StatefulWidget é criado, ele cria uma nova instância da classe State, usando o método createState(). Esse método é chamado apenas uma vez durante a vida do StatefulWidget e é responsável por criar e retornar uma nova instância da classe State associada. A classe State contém o estado interno mutável do widget e é responsável por atualizar a interface do usuário do widget sempre que o estado é alterado.

A assinatura do método createState() é a seguinte:

```
@override  
_MyWidgetState createState() => _MyWidgetState();
```

# Flutter: Character underscore

O caractere `_` antes de uma variável em Dart é usado para indicar que a variável é privada e só pode ser acessada dentro da classe em que foi definida. Essa convenção é conhecida como "underscore" em inglês e é comumente usada para indicar que a variável não deve ser acessada ou modificada diretamente fora da classe.

# Flutter: Scaffold

Em Flutter, Scaffold é um widget que fornece a estrutura básica de uma tela, incluindo uma barra de aplicativo, corpo e um fundo. Ele é amplamente usado para construir telas em aplicativos móveis e é um dos widgets mais comuns em Flutter.

O widget Scaffold possui vários parâmetros, como appBar, body, floatingActionButton, drawer, bottomNavigationBar, entre outros. Esses parâmetros podem ser usados para personalizar a aparência e o comportamento da tela.

O widget Scaffold é uma maneira conveniente e eficiente de criar telas em aplicativos móveis usando Flutter. Ele fornece a estrutura básica da tela, permitindo que os desenvolvedores se concentrem em adicionar conteúdo e recursos específicos do aplicativo.



# Flutter: Child e Children

Em Flutter, child é um dos parâmetros mais comuns em vários widgets e é usado para definir o conteúdo ou um único filho do widget. O parâmetro child espera receber um widget filho que será colocado dentro do widget pai.

Por exemplo,

- O widget Container possui um parâmetro child que é usado para definir o conteúdo do container.

- O widget Text possui um parâmetro child que é usado para definir o texto exibido no widget.

- O widget Icon possui um parâmetro child que é usado para definir o ícone exibido no widget.

Esses são apenas alguns exemplos de como o parâmetro child é usado em diferentes widgets em Flutter.

O child é usado para adicionar um único widget dentro de outro widget. Além disso, é possível usar outros parâmetros como children, que aceitam uma lista de widgets, para adicionar vários widgets dentro de outro widget.

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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Meu aplicativo',
      home: Scaffold(
        appBar: AppBar(
          title: Text('Minha barra de aplicativo'),
        ),
        body: Center(
          child: Text('Meu corpo'),
        ),
      ),
    );
  }
}
```

## Situação problema:

Com base no exemplo visto, modifique o exemplo dados para que o formulário contenha as seguintes informações:

- Nome
- Endereço
- Número
- Complemento
- UF
- CEP

\* Todos deve ser campos abertos, posteriormente iremos substituir por widgets específicos de acordo com o tipo de entrada. Por enquanto não é necessário aplicar máscara. Analisar se todas a entrada solicitas foram informadas, neste caso enviar uma mensagem indicando o envio, do contrário, mostrar qual das informações não foram informadas (destacar a entrada em vermelho neste caso).