



# 3<sup>o</sup> Ano - Técnico em Informática para Internet

Dieimes Nunes de Souza  
dieimes.souza@docente.pr.senac.br

# NA AULA DE HOJE...

Foi realizado algumas alterações, veja o código abaixo:

```
class Pagina lista extends StatelessWidget {  
  Widget build(BuildContext context) {  
    return Scaffold(  
      body: Center(  
        child: Padding(  
          padding: const EdgeInsets.all(20),  
          child: TextField(  
            decoration: InputDecoration(  
              labelText: 'Email',  
            ), // InputDecoration  
          ), // TextField  
        ), // Padding  
      ), // Center  
    ); // Scaffold  
  }  
}
```

# CONTROLADOR DE CAMPO DE TEXTO

Um controlador de campo de texto (TextEditingController) é uma classe que ajuda a controlar e manipular o texto inserido ou exibido em um campo de entrada, como um TextField ou TextFormField.

```
1  import 'package:flutter/material.dart';  
2  
3  final TextEditingController email controle = TextEditingController();  
4  
5  class Pagina lista extends StatelessWidget {
```

# CONTROLADOR DE CAMPO DE TEXTO

O **controller: email\_controle** associa o controlador email\_controle ao campo de entrada de texto. Isso permite controlar e acessar o texto inserido ou exibido no campo.

```
child: TextField(  
  controller: email_controle,  
  decoration: InputDecoration(  
    labelText: 'Email',  
  ), // InputDecoration  
) // TextField
```

# FUNÇÃO ENTRAR

```
23 void entrar () {  
24     String texto = email_controle.text;  
25     print(texto);  
26     email_controle.clear();  
27 }
```

- String texto = email\_controle.text; obtém o texto inserido no campo de entrada associado ao controlador email\_controle e armazena-o na variável texto.
- email\_controle.clear(); limpa o campo de entrada, removendo qualquer texto atualmente presente nele.

# COLUNA NO FLUTTER

O comando **child: Column(children: [],)** é usado para criar um layout de coluna no Flutter. Ele permite organizar vários widgets de maneira vertical, um abaixo do outro, dentro de um widget Column.

```
child: Padding(  
  padding: const EdgeInsets.all(20),  
  child: Column(children: [],)
```

Dentro do parâmetro **children** do **Column**, você pode adicionar uma lista de widgets que serão organizados verticalmente. Cada widget na lista será colocado um abaixo do outro, seguindo a ordem em que foram adicionados.

# COLUNA NO FLUTTER

```
body: Center(  
  child: Padding(  
    padding: const EdgeInsets.all(20),  
    child: Column(  
      children: [  
        TextField(  
          controller: email_controle,  
          decoration: InputDecoration(  
            labelText: 'Email',  
          ), // InputDecoration  
        ), // TextField  
      ],  
    ), // Column  
  ), // Padding  
) // Center
```

# CRIAÇÃO DO BOTÃO NO FLUTTER

O comando **ElevatedButton(onPressed: onPressed, child: child)** é usado para criar um botão.

- **onPressed:** Esse parâmetro define a função que será executada quando o botão for pressionado. Geralmente, você atribui uma função a ser chamada quando o botão for clicado.
- **child:** Esse parâmetro define o widget que será exibido como conteúdo do botão. Pode ser um texto, um ícone, uma imagem ou qualquer outro widget desejado. Por exemplo, `child: Text('Clique Aqui')` define um widget de texto "Clique Aqui" como o conteúdo do botão.



# CRIAÇÃO DO BOTÃO NO FLUTTER

```
child: Column(  
  children: [  
    TextField(  
      controller: email_controle,  
      decoration: InputDecoration(  
        labelText: 'Email',  
      ), // InputDecoration  
    ), // TextField  
    ElevatedButton(  
      onPressed: onPressed, child: child),  
  ],
```

# CRIAÇÃO DO BOTÃO NO FLUTTER

```
ElevatedButton(  
  onPressed: entrar,  
  child: Text("Entrar"),  
), // ElevatedButton
```

# CRIAÇÃO DO BOTÃO NO FLUTTER

```
ElevatedButton(  
  onPressed: entrar,  
  child: Text("Entrar"),  
), // ElevatedButton
```

# CENTRALIZANDO A COLUNA

O comando **mainAxisSize: MainAxisSize.min** é usado para definir o tamanho principal (main axis size) de um widget em relação ao eixo principal do layout.

```
child: Column(  
  mainAxisSize: MainAxisSize.min,  
  children: [  
    
```

# Dúvidas e Comentários

Fique à vontade para fazer deste um debate aberto à perguntas e esclarecimentos antes de prosseguirmos.