



# Botões e entrada de dados

Apresentado por: Eliane Dantas e Natalia Costa



# AGENDA

Introdução

Criação de botões

Entrada de dados

# Introdução

- A criação de botões e a entrada de dados são elementos fundamentais para tornar os aplicativos mais dinâmicos e interativos.
- Dessa forma, os botões são widgets responsáveis por acionar ações quando pressionados, enquanto a entrada de dados pode ser realizada por meio de campos de texto, seleção de itens em listas, entre outros.

# Criando botões

- Na classe `_MyHomePageState` criaremos mais um método chamado `_decrementCounter`.
- Este método ficará responsável por decrementar o atributo `_counter` da nossa aplicação e por executar o método `setState` disponível em todo Widget do tipo `Stateful`:

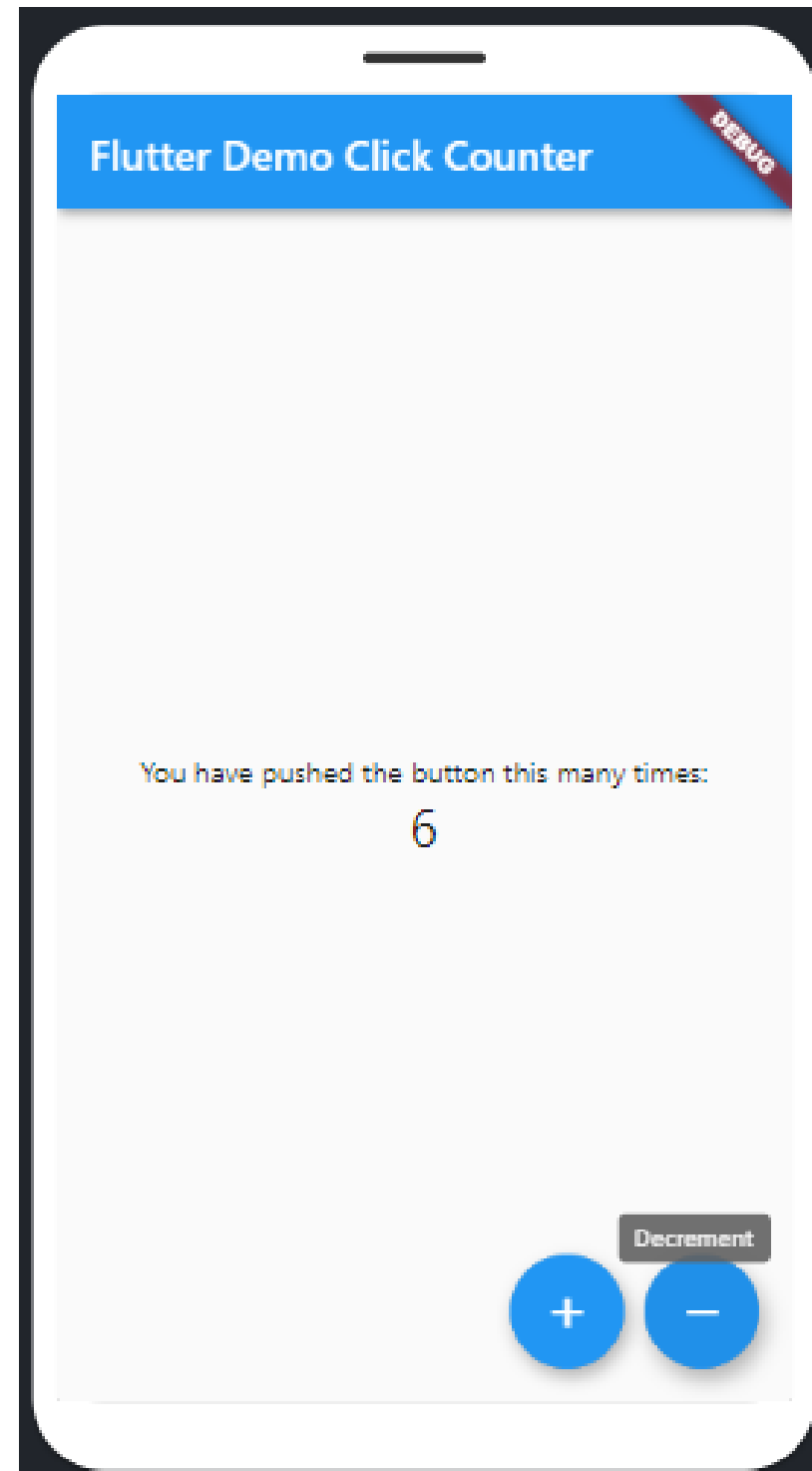
```
35 void decrementCounter() {  
36     setState(() {  
37         _counter--;  
38     });  
39 }
```

- Por fim, no atributo `floatingActionButton` do `Widget Scaffold`, adicione mais um botão ao lado do botão de incremento.
- Faremos isso agrupando ambos os botões no `Widget Row`, também disponibilizado pelo Flutter.

```
61 floatingActionButton: Row(  
62   mainAxisAlignment: MainAxisAlignment.end,  
63   children: <Widget>[  
64     FloatingActionButton(  
65       onPressed: _incrementCounter,  
66       tooltip: 'Increment',  
67       child: const Icon(Icons.add),  
68     ), // FloatingActionButton  
69     const SizedBox( width: 10.0, ),  
70     FloatingActionButton(  
71       onPressed: _decrementCounter,  
72       tooltip: 'Decrement',  
73       child: const Icon(Icons.remove),  
74     ), // FloatingActionButton  
75   ], // <Widget>[]  
76 ), // Row
```

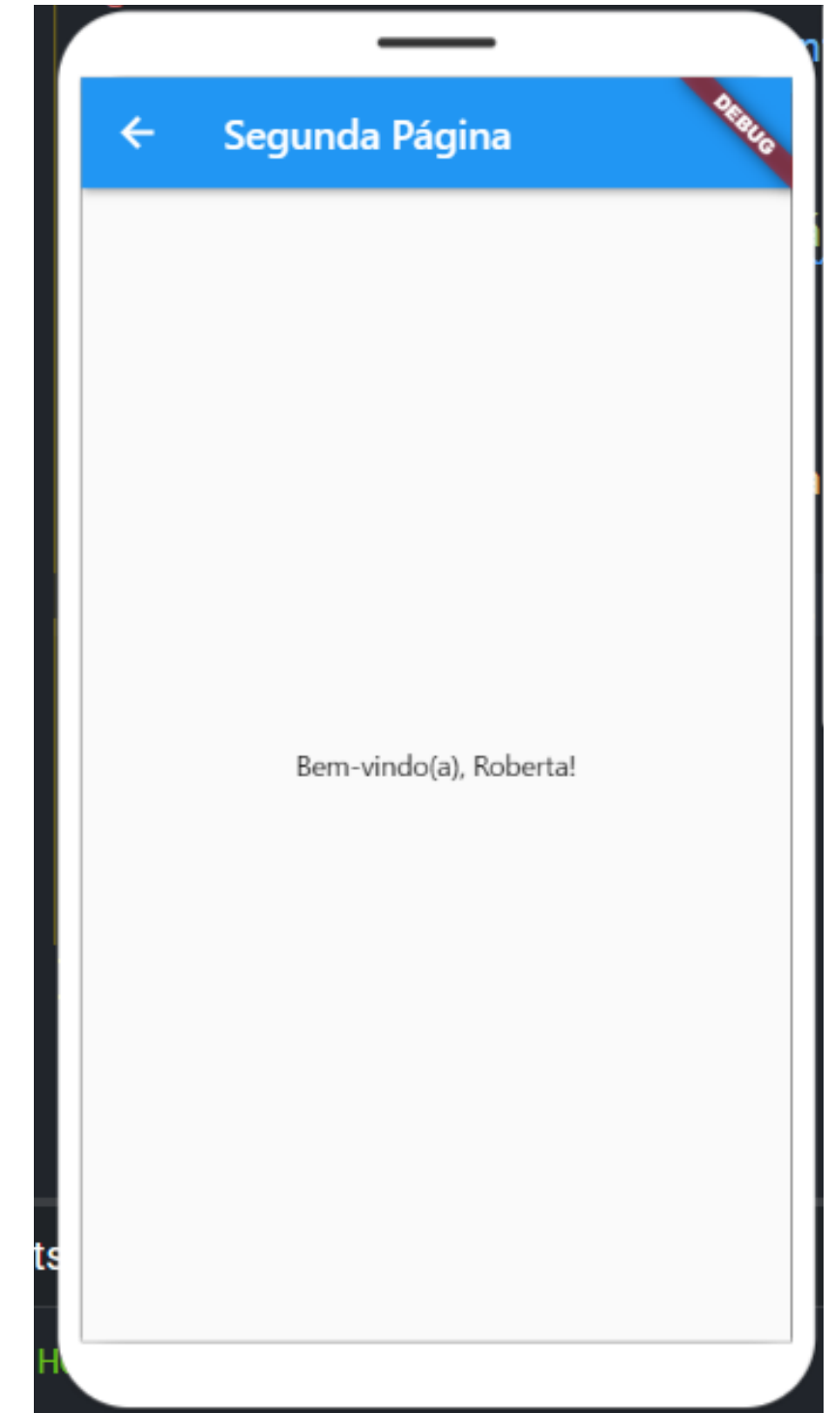
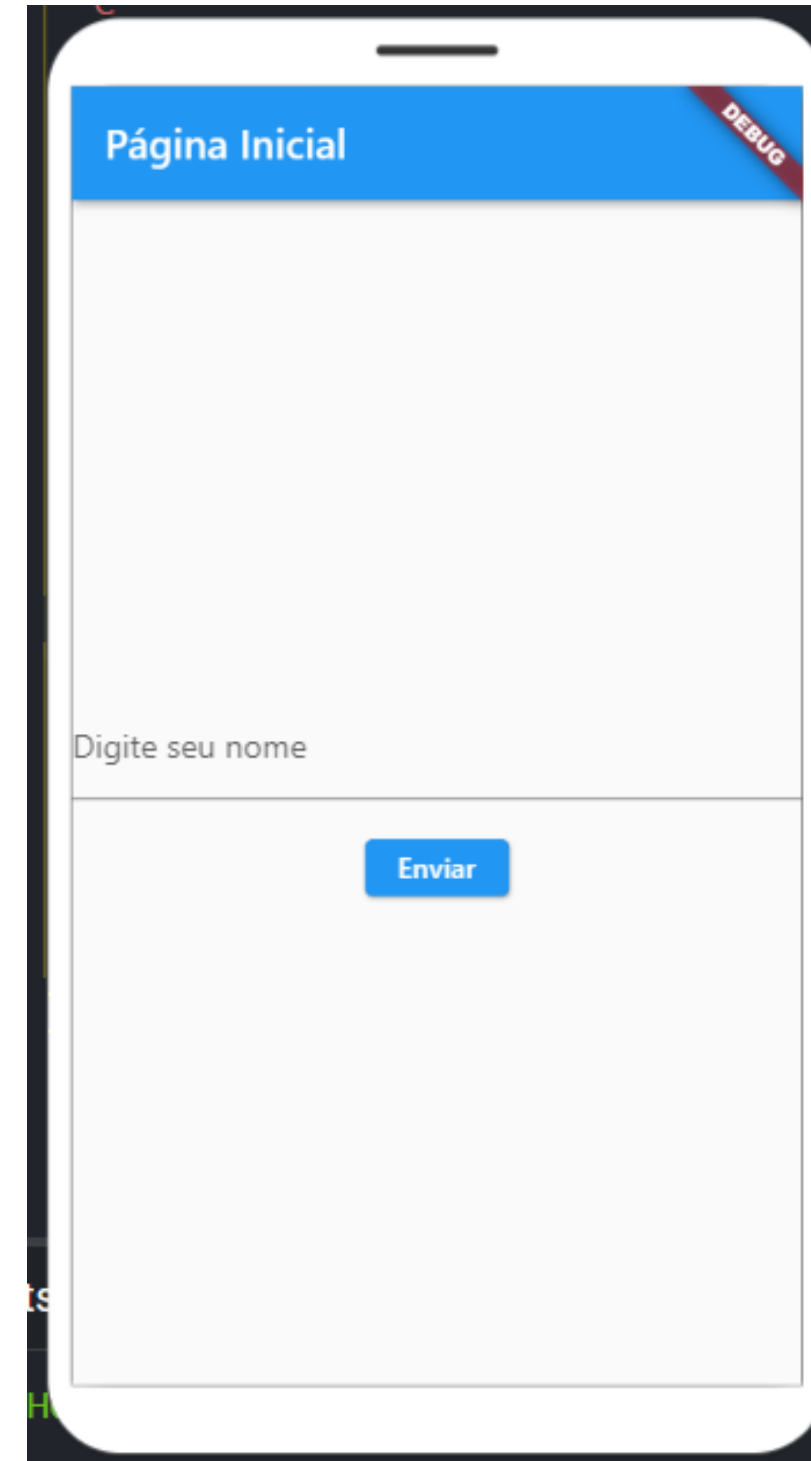
## Resultado esperado:

- Agora temos dois botões disponíveis na tela, um para incrementar o contador e outro para decrementá-lo.
- Acesse o código completo no arquivo criandoBotoes.dart



# Entrada de dados


- Agora substitua o código existente por esse: `entradaDados.dart`
- Neste exemplo, solicitamos que o usuário digite seu nome e exibimos uma mensagem de boas-vindas.
- Para isso, passamos o valor digitado no `TextField` da `HomePage` para a `SecondPage`, quando o botão "Enviar" é clicado.



# Entrada de dados

- HomePage: Esta é a classe responsável por definir a tela inicial da aplicação. Ela estende StatelessWidget, o que significa que não tem estado interno mutável.
  - No método build, retorna um Scaffold, que é uma estrutura de página típica em aplicativos Flutter. Ele contém um AppBar com o título 'Página Inicial' e um corpo que consiste em um TextField para o usuário digitar seu nome e um ElevatedButton que, quando clicado, navega para a SecondPage enviando o nome digitado.
- SecondPage: Esta classe define a segunda página da aplicação. Também estende StatelessWidget. Ela recebe o nome do usuário como parâmetro e o exibe em um Text com uma mensagem de boas-vindas.





**No próximo módulo,  
entenderemos melhor  
como funciona a  
navegação entre telas.**

# Referências

**Building user interfaces with Flutter.** Disponível em:  
<<https://docs.flutter.dev/ui>>. Acesso em: 4 mar. 2024.

**Como capturar de input de dados no Flutter.** Disponível em:  
<<https://www.devmedia.com.br/exemplo/flutter-tela-com-captura-do-widget-textfield/109>>. Acesso em: 5 mar. 2024.

**Hello world com Flutter.** Disponível em:  
<<https://www.devmedia.com.br/hello-world-com-flutter/40321>>. Acesso em:  
5 mar. 2024.