



# Navegação entre Telas

Apresentado por: Eliane Dantas e Natalia Costa



# AGENDA

Introdução

Uso do Navigator

Uso de Rotas Nomeadas

# Introdução

- O Flutter fornece um sistema completo para navegar entre telas e lidar com links diretos;
- Aplicativos pequenos, sem links diretos complexos, podem usar o Navigator;
- Além disso, há o uso das rotas nomeadas.

# O uso do Navigator

- O widget Navigator exibe telas como uma pilha, usando animações de transições corretas para a plataforma de destino;
- O Navigator é responsável por empilhar e desempilhar telas dentro do aplicativo;
- O Navigator utiliza dois métodos: o `Navigator.push` e o `Navigator.pop`.

# O uso do Navigator.push

- O método Navigator.push é utilizado para navegar para uma nova tela;
- Este método utiliza dois argumentos: o contexto atual e a rota para a nova tela;



Figura 01 – Uso do Navigator.push

# O uso do Navigator.pop

- Para voltar para a tela anterior, deve-se usar o método ‘Navigator.pop’;
- Este método remove o widget atual da pilha e volta para o widget anterior;
- Um exemplo do seu uso é quando o usuário pressiona o botão de voltar.



The screenshot shows a Flutter application running in an IDE. The code in main.dart is as follows:

```
main.dart M x
39 | @override
40 | Widget build(BuildContext context) {
41 |   return Scaffold(
42 |     appBar: AppBar(
43 |       title: const Text('Segunda Tela'),
44 |     ), // AppBar
45 |     body: Center(
46 |       child: ElevatedButton(
47 |         onPressed: () {
48 |           Navigator.pop(context); ←
49 |         },
50 |         child: const Text('Voltar'),
51 |       ), // ElevatedButton
52 |     ), // Center
53 |   ); // Scaffold
54 |
55 }
```

The application displays a blue header bar with the text "Segunda Tela". Below it is a center-aligned button labeled "Voltar". A red arrow points to the line of code where `Navigator.pop(context)` is called, indicating its location within the button's `onPressed` event handler.

Figura 02 – Uso do Navigator.pop

O exemplo com o código-fonte utilizando o navigator básico encontra-se no arquivo **navigatorBasico.dart**

# O uso de Rotas Nomeadas

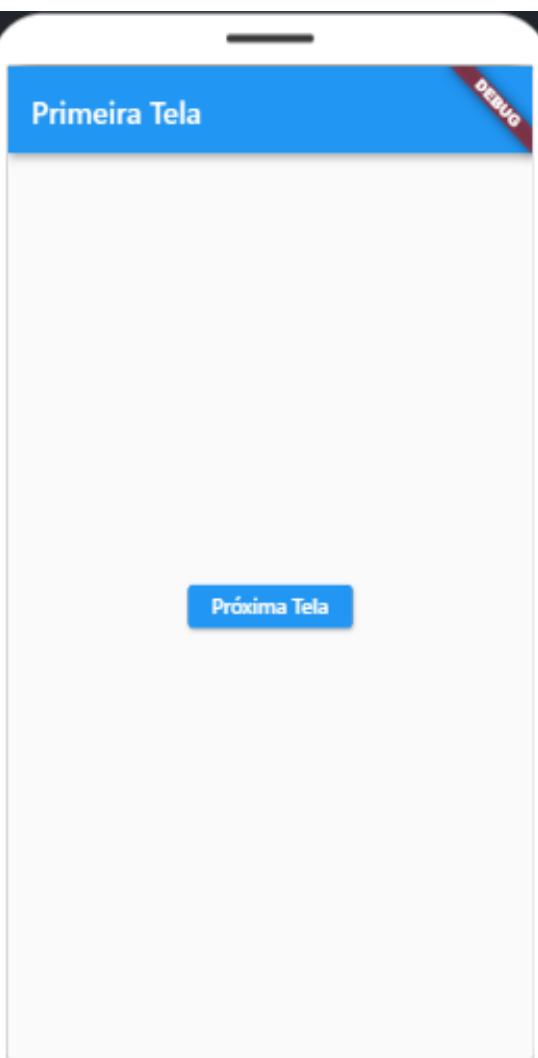
- No roteamento nomeado, é definido uma tabela de rotas, com os nomes de rotas e seus respectivos widgets;
- Então, é possível navegar para um widget usando o seu nome de rota;
- É útil quando há muitas telas no aplicativo e a navegação se torna complexa;
- A seguir será apresentado um exemplo do seu uso.

- Primeiramente, são definidas as rotas;
- A propriedade “initialRoute” define com qual rota o aplicativo deve começar;
- A propriedade “routes” define as rotas nomeadas disponíveis e os widgets que serão construídos ao navegar para essas rotas.

```
void main() {  
  runApp(  
    MaterialApp(  
      title: 'Rotas Nomeadas',  
      initialRoute: '/',  
      routes: {  
        '/': (context) => const PrimeiraTela(),  
        '/segunda': (context) => const SegundaTela(),  
      },  
    ), // MaterialApp  
);  
}
```

Figura 03 – Definição das rotas

- Serão criadas duas telas;
- A primeira tela contém um botão que navega para a segunda tela;
- A função “Navigator.pushNamed” aciona a navegação, indicando qual será o caminho.



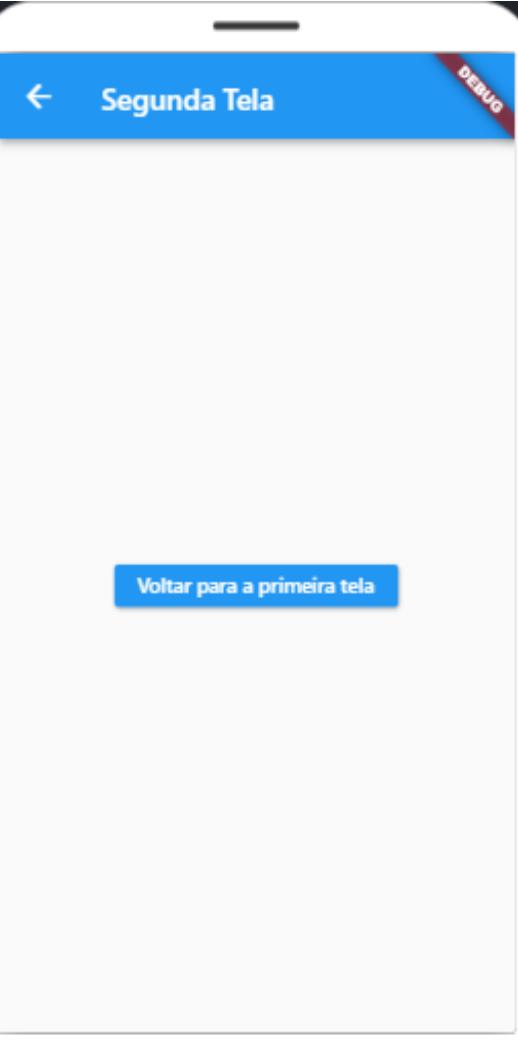
```

main.dart M x
16 class PrimeiraTela extends StatelessWidget {
17   const PrimeiraTela({super.key});
18
19   @override
20   Widget build(BuildContext context) {
21     return Scaffold(
22       appBar: AppBar(
23         title: const Text('Primeira Tela'),
24       ), // AppBar
25       body: Center(
26         child: ElevatedButton(
27           onPressed: () {
28             Navigator.pushNamed(context, '/segunda');
29           },
30           child: const Text('Próxima Tela'),
31         ), // ElevatedButton
32       ), // Center
33     ); // Scaffold
34   }
35 }

```

Figura 04 – Primeira Tela

- A segunda tela contém um botão que retornar para a primeira tela;
- A função “Navigator.pop” permite retornar para a tela anterior.



```

main.dart M ×
36
37 class SegundaTela extends StatelessWidget {
38   const SegundaTela({super.key});
39
40   @override
41   Widget build(BuildContext context) {
42     return Scaffold(
43       appBar: AppBar(
44         title: const Text('Segunda Tela'),
45       ), // AppBar
46       body: Center(
47         child: ElevatedButton(
48           onPressed: () {
49             Navigator.pop(context);
50           },
51           child: const Text('Voltar para a primeira tela'),
52         ), // ElevatedButton
53       ), // Center
54     ); // Scaffold
55   }
56 }

```

Figura 05 – Segunda Tela

O exemplo com o código-fonte utilizando rotas nomeadas encontra-se no arquivo **rotasNomeadas.dart**

# Referências

7.2. Navegação e roteamento em Flutter: Navegação entre telas. Disponível em: <https://cursa.app/en/page/navegacao-e-roteamento-em-flutter-navegacao-entre-telas>. Acesso em: 4 mar. 2024.

Navigate to a new screen and back. Disponível em: <https://docs.flutter.dev/cookbook/navigation/navigation-basics>. Acesso em: 4 mar. 2024.