

Desenvolvimento para Dispositivos Móveis

Navigator

Profº: Joseph Donald

Contatos:

☎(83) 98228-8607

📷@josephdonald

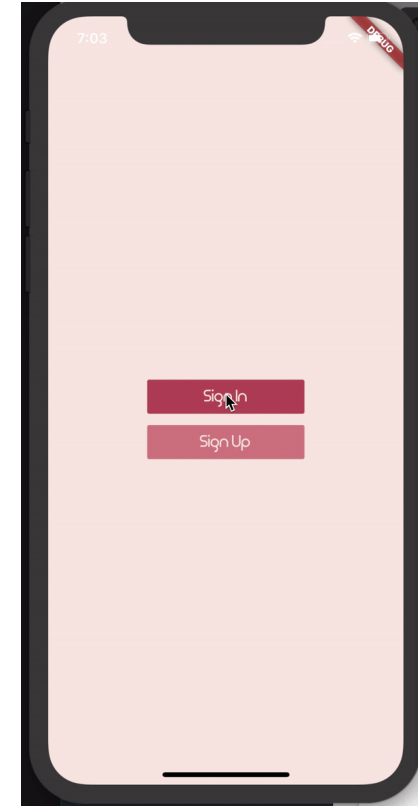
✉030106382@prof.uninassau.edu.br

“Se você tem uma maçã e eu tenho outra; e nós trocamos as maçãs, então cada um terá sua maçã. Mas se você tem uma ideia e eu tenho outra, e nós as trocamos; então cada um terá duas ideias.”

George Bernard Shaw

A transição de telas em um aplicativo desempenha um papel crucial na experiência do usuário. Algumas razões pelas quais a transição de telas é importante, são:

- **Feedback visual:** A transição de telas fornece um feedback visual para o usuário, indicando que uma ação foi concluída com sucesso. Isso ajuda a transmitir ao usuário que o aplicativo está respondendo às suas interações.
- **Orientação espacial:** A transição de telas pode ajudar o usuário a manter a orientação espacial dentro do aplicativo. Quando uma tela desaparece e uma nova tela é exibida com uma transição animada, o usuário tem uma melhor compreensão da hierarquia das telas.
- **Delimitação de contexto:** Cada tela em um aplicativo tem um contexto e uma funcionalidade específicos. A transição de telas ajuda a delimitar claramente esses contextos e separar as diferentes partes do aplicativo.
- **Melhoria da usabilidade:** Uma transição de tela bem projetada pode melhorar a usabilidade do aplicativo. Uma navegação suave e rápida pode ajudar a reduzir a sensação de espera do usuário, tornando a experiência do aplicativo mais ágil e responsiva.
- **Efeitos emocionais:** A transição de telas também pode ter um efeito emocional no usuário. Uma transição bem executada pode criar uma sensação de satisfação do usuário e qualidade no aplicativo, o que pode gerar uma impressão positiva e aumentar a satisfação geral dos usuários.



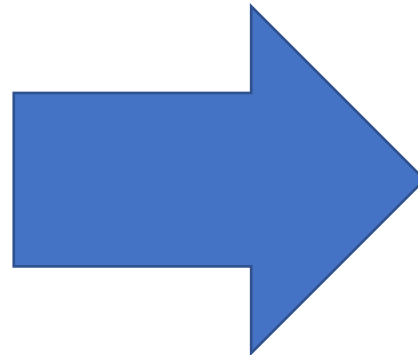


Navegação entre Telas

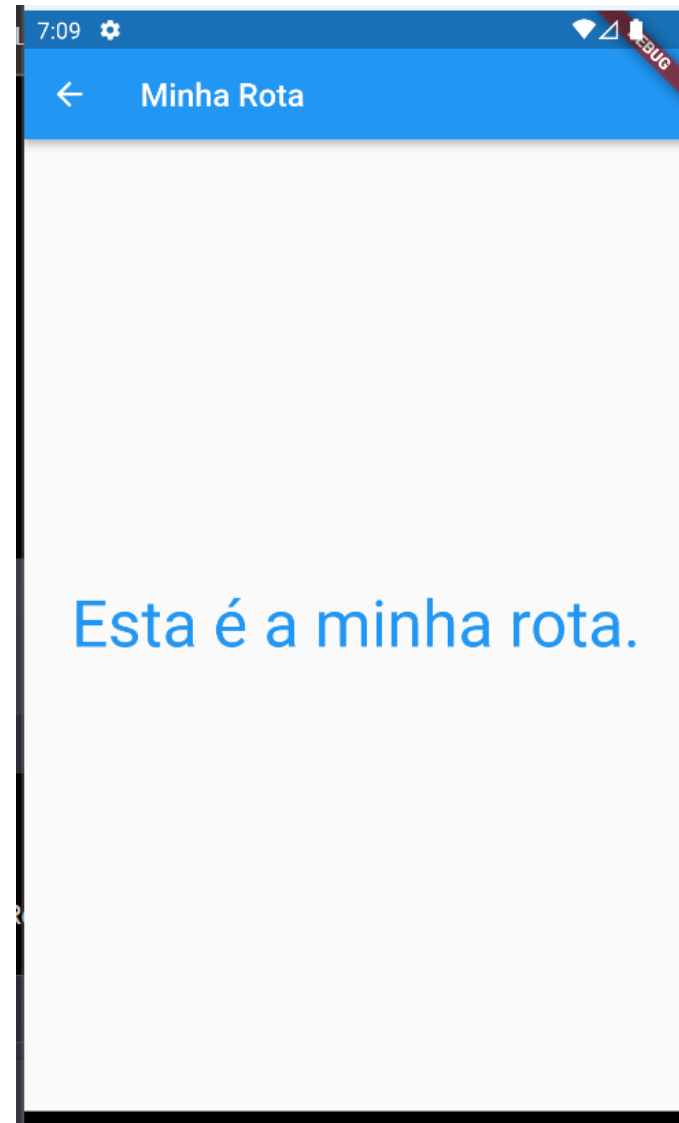
- Os Widgets de navegação permitem que o usuário mova-se pelo aplicativo ou que este leve a diferentes partes de suas funcionalidades.
- A classe utilizada para estas transições chama-se **Navigator**. A **Navigator** fornece vários métodos de transição, entre os mais utilizados temos o **push()** e o **pop()** para a inclusão ou remoção de rotas.
- A utilização do Navigator acontece através de widgets clicáveis como: *TextButton*, *ElevatedButton*, *GestureDetector*, *BottomNavigationBar* e etc.
- Além disso, ele facilita a passagem de dados entre as telas.



Tela 1



Tela 2



Tela 1

```
13 class _HomeState extends State<Home> {
14   @override
15   Widget build(BuildContext context) {
16     return Scaffold(
17       body: Center(
18         child: TextButton(
19           style: ButtonStyle(
20             backgroundColor: MaterialStateProperty.all(Colors.amber), // ButtonStyle
21             onPressed: () {
22               Navigator.push(context,
23                 MaterialPageRoute(builder: (context) => MinhaRota()));
24             },
25             child: const Text(
26               "MUDAR DE ROTA",
27               style: TextStyle(fontSize: 45),
28             )), // Text // TextButton
29       ), // Center
30     ); // Scaffold
31   }
32 }
33
```

Tela 2

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

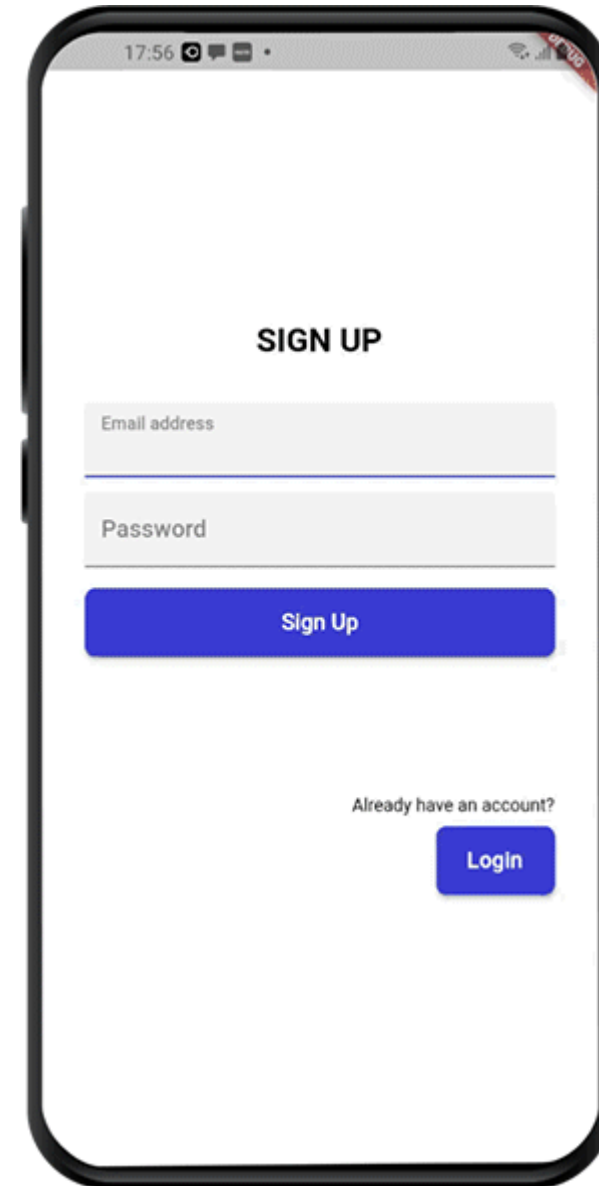
class MinhaRota extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Minha Rota'),
      ), // AppBar
      body: const Center(
        child: Text(
          'Esta é a minha rota.',
          style: TextStyle(fontSize: 40, color: Colors.blue),
        ), // Text
      ), // Center
    ); // Scaffold
  }
}
```

Transportar dados entre as telas de um aplicativo é de extrema importância, pois permite a comunicação e o compartilhamento de informações entre os diferentes componentes do aplicativo. Algumas razões pelas quais o transporte de dados entre telas é essencial, são:

1. **Compartilhamento de informações:** Dados transportados entre as telas permitem que informações sejam passadas de uma tela para outra. Isso pode incluir dados de autenticação do usuário, preferências do usuário, resultados de pesquisas, informações de produtos selecionados, detalhes de pedidos e muito mais.
2. **Personalização da experiência do usuário:** Permite personalizar a experiência do usuário com base nas informações recebidas. Por exemplo, os dados transportados podem ser utilizados para exibir conteúdo personalizado, adaptar a interface do usuário, ajustar configurações e fornecer recomendações relevantes.
3. **Mantendo o estado do aplicativo:** O transporte de dados também é fundamental para manter o estado do aplicativo durante a navegação entre telas. Ao passar dados entre as telas, é possível preservar o estado atual do aplicativo, como seleções de filtros, itens em um carrinho de compras, progresso de um formulário e outras informações temporárias.
4. **Separação de responsabilidades:** Ao transportar dados entre as telas, você pode garantir que cada tela seja responsável apenas pelas informações necessárias para sua funcionalidade específica. Isso promove uma arquitetura modular e de fácil manutenção, em que cada tela tem um escopo limitado e recebe apenas os dados relevantes para sua exibição e funcionalidade.
5. **Integração com serviços externos:** O transporte de dados entre as telas é útil para integrar o aplicativo com serviços externos, como APIs. Os dados podem ser enviados como parâmetros para solicitações de API, permitindo a recuperação e atualização de informações em tempo real.

Navegação com transporte de dados

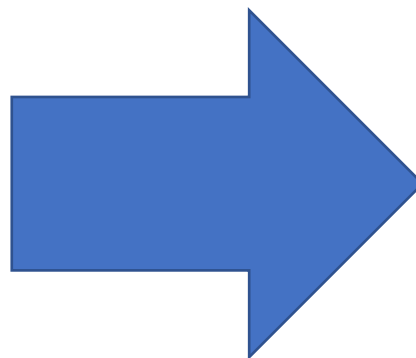
- Para transportar dados usando o **MaterialPageRoute** no Flutter, você pode utilizar a funcionalidade *arguments* através da classe **RouteSettings**.
- Os argumentos podem ser qualquer tipo de dado, como objetos, listas, mapas, etc.
- Para receber os dados enviados, utiliza-se a classe **ModalRoute**, que captura os argumentos enviados ao novo contexto e pode atribuí-lo a uma variável, objeto ou lista na tela de destino.



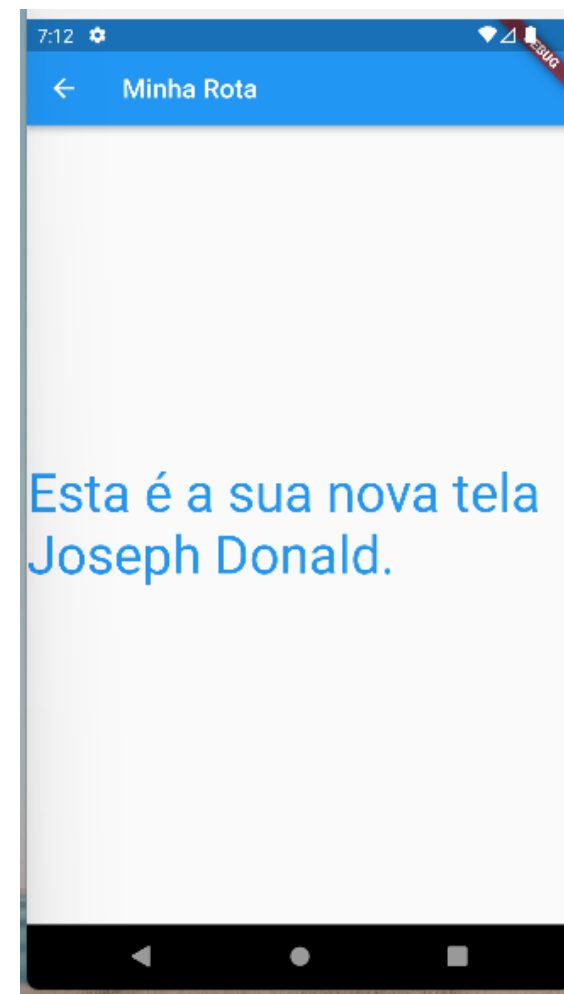


Navegação entre Telas com transporte de dados (Exemplo)

Tela 1



Tela 2



Navegação entre Telas com transporte de dados (Código)

- Na tela que **envia** a informações você terá por exemplo:

```
13 class _HomeState extends State<Home> {  
14   @override  
15   Widget build(BuildContext context) {  
16     return Scaffold(  
17       body: Center(  
18         child: TextButton(  
19           style: ButtonStyle(  
20             backgroundColor: MaterialStateProperty.all(Colors.amber),  
21             onPressed: () {  
22               Navigator.push(  
23                 context,  
24                 MaterialPageRoute(  
25                   builder: (context) => MinhaRota(),  
26                   settings: RouteSettings(arguments: "Joseph Donald")));  
27             },  
28             child: const Text(  
29               "MUDAR DE ROTA",  
30               style: TextStyle(fontSize: 45),  
31             )), // Text // TextButton  
32         ), // Center  
33       ); // Scaffold  
34   }  
35 }
```



Navegação entre Telas com transporte de dados (Código)

- Na tela que **recebe** a informações você terá por exemplo:

```
3  import 'package:flutter/material.dart';
4
5  class MinhaRota extends StatelessWidget {
6    @override
7    Widget build(BuildContext context) {
8      String nomeRecebido = ModalRoute.of(context)?.settings.arguments as String;
9
10     return Scaffold(
11       appBar: AppBar(
12         title: Text('Minha Rota'),
13       ), // AppBar
14       body: Center(
15         child: Text(
16           'Esta é a sua nova tela $nomeRecebido.',
17           style: TextStyle(fontSize: 40, color: Colors.blue),
18         ), // Text
19       ), // Center
20     ); // Scaffold
21   }
22 }
23
```