

# Programação Para Dispositivos Móveis

## Aula 5

Paulo Tomé

Escola Superior de Tecnologia e Gestão de Viseu  
*ptome@estgv.ipv.pt*

Ano letivo 2024-2025

# Sumário

- 1 Route
- 2 Widget Navigator
- 3 Widget Router
- 4 Widget Drawer
- 5 Widget NavigationBar

- Um ecrã de uma aplicação em Flutter é uma *Route*.
- Uma aplicação pode ter várias *routes*.
- Para se navegar entre ecrãs, as aplicações podem utilizar:
  - *Navigator*,
  - *Router*.
- A navegação entre *routes* faz-se utilizando-se os princípios subjacentes a uma *stack*. Adiciona-se mais um elemento à *stack* quando se passa de uma *route* para outra. Vem-se para o ecrã anterior através de um *pop*.

- Este *widget* permite gerir um conjunto de "sub" (filhos) *widgets* de acordo tendo por bases princípios de uma *stack*.
- Este *widget* tem dois métodos: *push* e *pop*.
- O método *push* faz o carregamento.
- O método *pop* retira da "stack".

- Considere-se o extrato de código:

```
class MyHomePage extends StatefulWidget {  
  const MyHomePage({super.key, required this.title});  
  final String title;  
  @override  
  State<MyHomePage> createState() => _MyHomePageState();  
}  
  
class _MyHomePageState extends State<MyHomePage> {  
  void carrega()  
  {  
    Navigator.push(  
      context,  
      MaterialPageRoute(builder: (context) =>  
        const Segunda(title: 'Segunda')),  
    );  
  }  
  
  @override  
  Widget build(BuildContext context) {  
  
    ...  
  }  
}
```

- O método `carrega` faz um `push`:

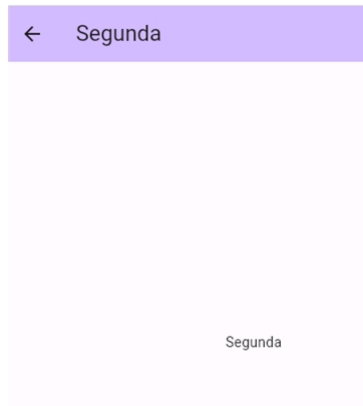
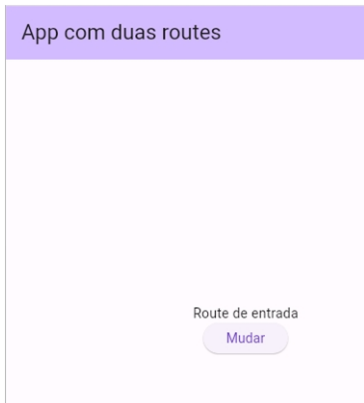
```
void carrega()
{
  Navigator.push(
    context,
    MaterialPageRoute(builder: (context) =>
      const Segunda(title: 'Segunda')),
  );
}
```

**Segunda** é uma classe que implementa outra route.

- Código de *route* **Segunda**:

```
import 'package:flutter/material.dart';
class Segunda extends StatefulWidget {
  const Segunda({super.key, required this.title});
  final String title;
  @override
  State<Segunda> createState() => _Segunda();
}
class _Segunda extends State<Segunda> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Theme.of(context).colorScheme.
        inversePrimary,
        title: Text(widget.title),),
      body: Center(child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: <Widget>[
          const Text('Segunda'),
          ...
        ],
      )),
    );
  }
}
```

- Ecrãs da aplicação:





- O método *pop* permite voltar à *route* anterior:

```
Navigator.pop(context);
```

- A navegação pode ocorrer através das designadas *named routes*. Este método está atualmente a ser desaconselhado.
- No *MaterialApp* são definidas as rotas:

```
MaterialApp(  
  title: '?????',  
  routes: {  
  
    '/': (context) => const Primeiroecr(),  
    '/segunda': (context) => const Segunda(),  
  },  
  initialRoute: '/',  
)
```

- A transição é feita da forma seguinte:

```
Navigator.pushNamed(context, '/segunda');
```

# Widget Router

- A navegação pode ocorrer utilizando-se o package *Go\_router*.
- Este método terá de ser adicionado nas dependências:

```
dependencies:  
  go_router:
```

- Deve ser definido na aplicação um ficheiro com as diversas rotas:

```
import 'package:go_router/go_router.dart';  
import 'main.dart';  
import 'segunda.dart';  
final rotas = GoRouter(  
  initialLocation: '/',  
  routes: [  
    GoRoute(  
      name: 'home',  
      path: '/',  
      builder: (context, state) =>const MyHomePage(title: 'Apl. Teste'),  
    ),  
    GoRoute(  
      name: 'segunda',  
      path: '/segunda',  
      builder: (context, sate) => const Segunda(title: 'Apl. Teste'),  
    ),  
  ],  
);
```

- Método build da classe MyApp:

```
@override
Widget build(BuildContext context) {
  return MaterialApp.router(
    routerConfig: rotas,
    ...
  );
}
```

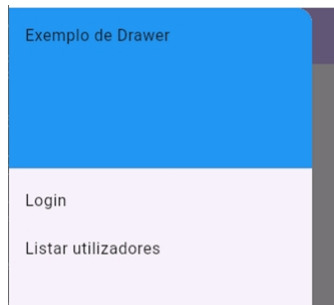
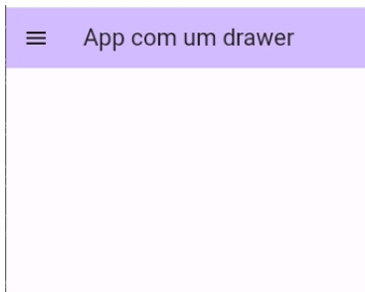
- A transição é feita da forma seguinte:

```
context.go('/segunda');
```

- O *widget Scaffold* tem a característica *drawer*
- Esta característica permite definir um menu do tipo *hamburger*.
- A definição do menu faz-se através do *widget Drawer*:

```
Drawer(  
  child: ListView(padding: EdgeInsets.zero,  
    children: [  
      const DrawerHeader(  
        decoration: BoxDecoration(color: Colors.blue),  
        child: Text('Exemplo de Drawer'), // título do drawer  
      ),  
      ListTile( // primeira opção  
        title: const Text('Login'),  
        onTap: () {},  
      ),  
    ],  
  ),  
  ...  
)
```

- Ecrãs da aplicação:



- O *drawer* é definido através de uma *Listview*.
- O *widget DrawerHeader* define o título do *drawer*.
- Cada opção do *drawer* é definida através de um *ListTile*.
- No *ListTile* define o texto do item e a ação a executar.

- O *widget Scaffold* tem a característica *bottomNavigationBar*
- Esta característica definir na parte de baixo da aplicação um conjunto de opções de navegação (vulgarmente de 3 a 5).
- A definição das opções faz-se através do *widget NavigationBar*:



# Widget NavigationBar

```
NavigationBar(  
  onDestinationSelected: (int index) {  
    setState(() {  
      currentPageIndex = index;  
    });  
  },  
  indicatorColor: Colors.amber,  
  selectedIndex: currentPageIndex,  
  destinations: const <Widget>[  
    NavigationDestination(  
      selectedIcon: Icon(Icons.home),  
      icon: Icon(Icons.home_outlined),  
      label: 'Início',  
    ),  
    NavigationDestination( //<ion-icon name="arrow-redo-outline"></ion-icon>  
      icon: Icon(Ionicons.arrow_forward),  
      label: 'Ecran 2',  
    ),  
    NavigationDestination(  
      icon:  
        Icon(Ionicons.stats_chart),  
  
      label: 'Ecran 3',  
    ),  
  ],  
),
```

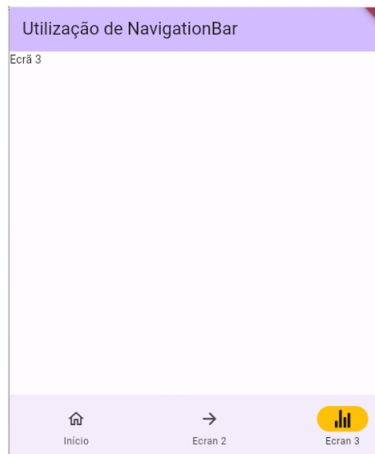
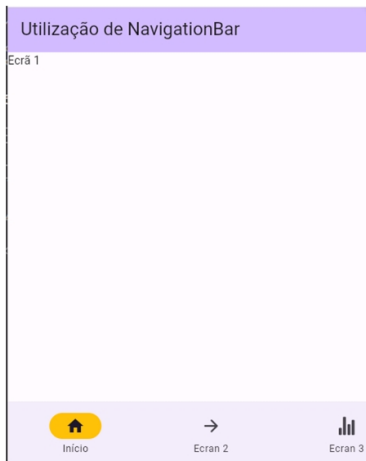
# Widget NavigationBar

- A característica *onDestinationSelected* permite determinar qual foi o item selecionado.
- O *widget NavigationDestination* permite definir os vários elemento da barra de navegação.
- O corpo do *widget Scaffold* é:

```
body: <Widget>[  
  
    Text('Ecrã 1'),  
    Text('Ecrã 2'),  
    Text('Ecrã 3'), // chartData: chartDatav  
  
    ][currentPageIndex],
```

- Ao se pressionar a primeira opção, aparece: "Ecrã 1"

- Ecrãs da aplicação:



- Desenvolver uma aplicação que tenha o layout gráfico apresentado no diapositivo anterior.

- <https://api.flutter.dev/flutter/widgets/Navigator-class.html>
- <https://docs.flutter.dev/cookbook/design/drawer>
- <https://api.flutter.dev/flutter/material/BottomNavigationBar-class.html>