

Widgets

Computação para Dispositivos Móveis

Prof. Gustavo Custodio
gustavo.custodio@anhembib.br



Widgets

AppBar

AppBar

- O AppBar é a barra que fica no topo dos aplicativos.

AppBar

- O AppBar é a barra que fica no topo dos aplicativos.
- É comum que os apps possuam ícones de menu, de edição ou de busca nessas barras.

AppBar

- O AppBar é a barra que fica no topo dos aplicativos.
- É comum que os apps possuam ícones de menu, de edição ou de busca nessas barras.

AppBar

- O menu à esquerda é definido pelo atributo `leading` do AppBar.
- O símbolo de busca é parte do atributo `actions`.
 - Um atributo que define quais ações o usuário pode realizar em um app.
 - Recebe uma lista de ações possíveis.

AppBar

- Exemplo de código.

AppBar

- Exemplo de código.

```
1
2  class MyApp extends StatelessWidget {
3    const MyApp({Key? key}) : super(key: key);
4
5    // This widget is the root of your application.
6    @override
7    Widget build(BuildContext context) {
8      return MaterialApp(
9        home: BarraApp(),
10     );
11   }
12 }
```


AppBar

- Exemplo de código.

AppBar

- Exemplo de código.

```
1 class BarraApp extends StatelessWidget {
2   Widget build(BuildContext context) {
3     return Scaffold(
4       appBar: AppBar(
5         backgroundColor: Colors.indigo,
6         title: Text('Aplicativo com Barra'),
7         leading: Padding(
8           padding: const EdgeInsets.all(10.0),
9           child: Icon(Icons.menu),
10        ),
11        actions: <Widget>[
12          Padding(
13            padding: const EdgeInsets.all(15.0),
14            child: Icon(Icons.search),
15          ),
16        ],
17      ),
18    );
19  }
20 }
```

AppBar

- `Icons.menu` representa o ícone com 3 linhas horizontais;

AppBar

- `Icons.menu` representa o ícone com 3 linhas horizontais;
- `Icons.search` representa o ícone de busca.

AppBar

- `Icons.menu` representa o ícone com 3 linhas horizontais;
- `Icons.search` representa o ícone de busca.
- Vamos adicionar um ícone de editar ao lado da busca:
 - `Icons.edit`.

AppBar

- O menu no canto superior esquerdo é um lugar comum para adicionar a navegação dentro do aplicativo.

AppBar

- O menu no canto superior esquerdo é um lugar comum para adicionar a navegação dentro do aplicativo.
 - Na próxima aula veremos como adicionar um comportamento nesse menu.
 - E, conseqüentemente, como navegar por telas diferentes.



Widgets

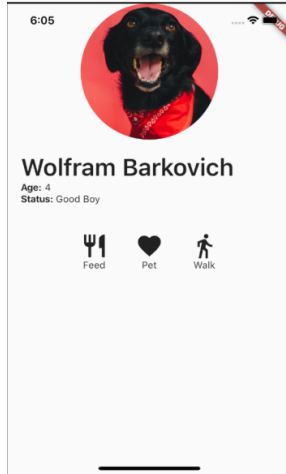
Prática

Prática

- Vamos criar uma tela de perfil para um cachorro.

Prática

- Vamos criar uma tela de perfil para um cachorro.



Prática

- Primeiro vamos começar com um código de uma tela vazia.

Prática

- Primeiro vamos começar com um código de uma tela vazia.
- Teremos três funções para construir a tela:

Prática

- Primeiro vamos começar com um código de uma tela vazia.
- Teremos três funções para construir a tela:
 - `_buildProfileImage(BuildContext context);`
 - `_buildProfileDetails(BuildContext context);`
 - `_buildActions(BuildContext context).`

Prática

- O `BuildContext` é utilizado para localizar cada widget na árvore de widgets.

Prática

- O `BuildContext` é utilizado para localizar cada widget na árvore de widgets.
- Dessa forma é necessário passá-lo como parâmetro sempre que retornamos um widget que ajudará a construir essa árvore.

Prática

- Crie um arquivo chamado `profile_screen.dart`.

Prática

- Crie um arquivo chamado `profile_screen.dart`.
- No `main.dart` adicione o seguinte código:

Prática

- Crie um arquivo chamado `profile_screen.dart`.
- No `main.dart` adicione o seguinte código:

```
1 import 'package:flutter/material.dart';
2 import './profile_screen.dart';
3
4 void main() {
5   runApp(MyApp());
6 }
7
8 class MyApp extends StatelessWidget {
9   @override
10  Widget build(BuildContext context) {
11    return MaterialApp(
12      home: ProfileScreen(),
13    );
14  }
15 }
```

Prática

- Agora adicione o código no arquivo `profile_screen.dart`:

Prática

- Agora adicione o código no arquivo `profile_screen.dart`:

```
1 import 'package:flutter/material.dart';
2
3 class ProfileScreen extends StatelessWidget {
4   @override
5   Widget build(BuildContext context) {
6     return Scaffold(
7       body: Column(
8         children: <Widget>[
9           _buildProfileImage(context),
10          _buildProfileDetails(context),
11          _buildActions(context),
12        ],
13      ),
14    );
15  }
16
17  Widget _buildProfileImage(BuildContext context) {
18    return Container();
19  }
20  Widget _buildProfileDetails(BuildContext context) {
21    return Container();
22  }
23  Widget _buildActions(BuildContext context) {
24    return Container();
25  }
26 }
```

Prática

- Por enquanto o app não terá nenhuma funcionalidade.

Prática

- Por enquanto o app não terá nenhuma funcionalidade.
- Cada método corresponde a uma parte diferente do app.

Prática

- Por enquanto o app não terá nenhuma funcionalidade.
- Cada método corresponde a uma parte diferente do app.
- Vamos começar com o `_buildProfileImage(BuildContext context);`

Prática

```
1 Widget _buildProfileImage(BuildContext context) {  
2   return Container(  
3     width: 200, // largura  
4     height: 200, //altura  
5     child: CircleAvatar(  
6       backgroundImage: NetworkImage(  
7         'https://www.collinsdictionary.com/images/full/dog_230497594.jpg'),  
8     ),  
9   );  
10 }
```


Prática

```
1 Widget _buildProfileImage(BuildContext context) {  
2   return Container(  
3     width: 200, // largura  
4     height: 200, //altura  
5     child: CircleAvatar(  
6       backgroundImage: NetworkImage(  
7         'https://www.collinsdictionary.com/images/full/dog_230497594.jpg'),  
8     ),  
9   );  
10 }
```

- Isso resulta na foto do cachorro aparecendo na tela.

Prática

- Depois disso, adicionamos informações sobre o cachorro.

Prática

- Depois disso, adicionamos informações sobre o cachorro.

```
1 Widget _buildProfileDetails(BuildContext context) {  
2   return Padding(  
3     padding: const EdgeInsets.all(20.0),  
4     child: Column(  
5       crossAxisAlignment: CrossAxisAlignment.start,  
6       children: <Widget>[  
7         Text(  
8           'Wolfram Barkovich',  
9           style: TextStyle(fontSize: 35, fontWeight: FontWeight.w600),  
10        ),  
11        _buildDetailsRow('Age', '4'),  
12        _buildDetailsRow('Status', 'Good Boy'),  
13      ],  
14    ),  
15  );  
16 }
```

Prática

- O texto é adicionado em um widget separado.

Prática

- O texto é adicionado em um widget separado.

```
1 Widget _buildDetailsRow(String heading, String value) {  
2   return Row(  
3     children: <Widget>[  
4       Text(  
5         '$heading: ',  
6         style: TextStyle(fontWeight: FontWeight.bold),  
7       ),  
8       Text(value),  
9     ],  
10  );  
11 }
```

Prática

- Depois disso, adicionamos alguns botões falsos que simulam a interação com o cachorro.

Prática

- Depois disso, adicionamos alguns botões falsos que simulam a interação com o cachorro.

```
1 Widget _buildActions(BuildContext context) {  
2   return Row(  
3     mainAxisAlignment: MainAxisAlignment.center,  
4     children: <Widget>[  
5       _buildIcon(Icons.restaurant, 'Feed'),  
6       _buildIcon(Icons.favorite, 'Pet'),  
7       _buildIcon(Icons.directions_walk, 'Walk'),  
8     ],  
9   );  
10 }
```

Prática

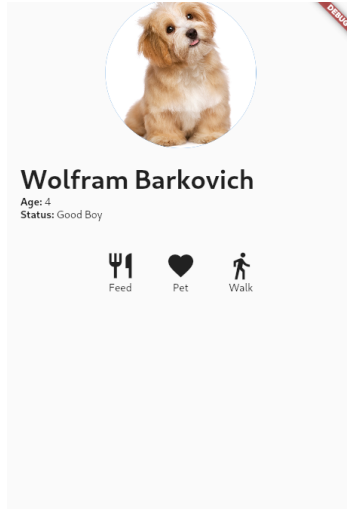
```
1
2 Widget _buildIcon(IconData icon, String text) {
3     return Padding(
4         padding: const EdgeInsets.all(20.0),
5         child: Column(
6             children: <Widget>[Icon(icon, size: 40), Text(text)],
7         ),
8     );
9 }
```


Prática

- Com isso, obtemos o seguinte resultado:

Prática

- Com isso, obtemos o seguinte resultado:



Exercício

- Coloque uma imagem de fundo atrás da foto do cachorro.
- Use o widget Stack para isso.

Exercício

- Crie o layout de um aplicativo de perguntas e respostas utilizando os widgets do flutter.

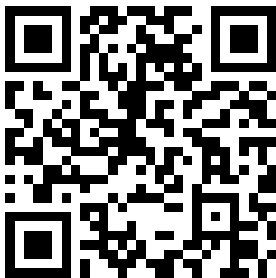
Referências

 Simone Alessandria and Brian Kayfitz.

Flutter Cookbook: Over 100 proven techniques and solutions for app development with Flutter 2.2 and Dart.

Packt Publishing Ltd, 2021.

Conteúdo



<https://gustavotcustodio.github.io/dispomoveis.html>

Obrigado

gustavo.custodio@anhembi.br