

Image

O Image é um dos Widgets mais utilizados em aplicações Flutter. Esse Widget é utilizado para a inclusão de imagens em nossas aplicações.

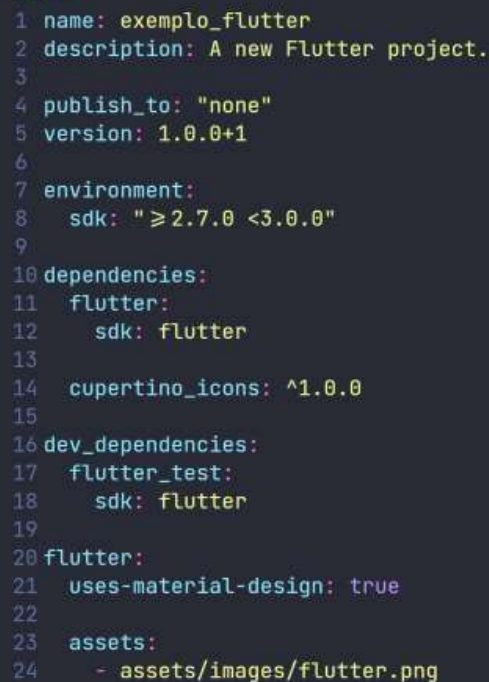
Um aspecto importante quanto ao uso de imagens em nossas aplicações, é o fato de todas as imagens de uma aplicação serem consideradas assets do projeto. Sendo assim, para utilizarmos imagens em nossas aplicações devemos optar por duas formas distintas de adicioná-las às nossas telas:

- Usando imagens diretamente nos assets do projeto, declarando-os no pubspec.yaml;
- Carregando-a diretamente da internet, utilizando a função network do próprio Widget.

Carregando imagens locais

Para carregarmos imagens locais, ou seja, que tenham sido inseridas como assets do projeto, utilizamos o método asset da classe Image.

Pubspec.yaml



```
1 name: exemplo_flutter
2 description: A new Flutter project.
3
4 publish_to: "none"
5 version: 1.0.0+1
6
7 environment:
8   sdk: "≥ 2.7.0 <3.0.0"
9
10 dependencies:
11   flutter:
12     sdk: flutter
13
14   cupertino_icons: ^1.0.0
15
16 dev_dependencies:
17   flutter_test:
18     sdk: flutter
19
20 flutter:
21   uses-material-design: true
22
23   assets:
24     - assets/images/flutter.png
```

main.dart

```
1 Scaffold(  
2   appBar: AppBar(  
3     title: Text("Aplicação Exemplo"),  
4   ),  
5   body: Image.asset("images/flutter.png"),  
6 )
```

Carregando imagens através da internet

Para carregarmos imagens diretamente da internet, utilizamos o método `network` da classe `Image`.

```
1 Scaffold(  
2   appBar: AppBar(  
3     title: Text("Aplicação Exemplo"),  
4   ),  
5   body: Image.network(  
6     "https://www.kindpng.com/picc/m/355-3  
557482_flutter-logo-png-transparent-png.png"),  
7 )
```

Propriedade width e height

As propriedades `width` e `height`, como o próprio nome já sugere, definem, respectivamente, a largura e a altura da imagem. Sendo que essas propriedades recebem um valor `double`.

```
1 Scaffold(  
2   appBar: AppBar(  
3     title: Text("Aplicação Exemplo"),  
4   ),  
5   body: Image.asset("assets/images/flutter.png",  
6     width: 100.0,  
7     height: 100.0,  
8   ),  
9 )
```

Propriedade fit

A propriedade `fit`, como o próprio nome já sugere, define como a imagem deverá se comportar em relação ao seu pai. Essa propriedade recebe um objeto `BoxFit`. Para tal utilizamos uma das constantes presentes nesta mesma classe. Dentre as possibilidades encontramos:

- **Contains:** O maior possível, mas ainda contendo a fonte inteiramente dentro da caixa de destino;
- **Cover:** O menor possível, mas ainda cobrindo toda a caixa de destino;
- **Fill:** Preenche a caixa de destino distorcendo a proporção da fonte;
- **fitWidth:** Certifica-se de que toda a largura da fonte seja mostrada, independentemente de isso significar que a fonte ultrapassa a caixa de destino verticalmente;
- **fitHeight:** Certifica-se de que a altura total da fonte seja mostrada, independentemente de isso significar que a fonte ultrapassa a caixa de destino horizontalmente;
- **scaleDown:** Alinhe a fonte dentro da caixa de destino (por padrão, centralizando) e, se necessário, reduza a fonte para garantir que a fonte caiba dentro da caixa;



```
1 Scaffold(  
2   appBar: AppBar(  
3     title: Text("Aplicação Exemplo"),  
4   ),  
5   body: Container(  
6     color: Colors.amber,  
7     width: double.infinity,  
8     height: double.infinity,  
9     child: Image.asset("assets/images/flutter.png",  
10    fit: BoxFit.cover,  
11  ),  
12 ),  
13 )
```