Layouts no Flutter

Apresentado por: Eliane Dantas e Natalia Costa

AGENDA

- Introdução
- Center
- Column
- Row
- Container

Introdução

- No geral, o layout de uma tela é composto por Widgets visíveis, como barras de menu, painéis, imagens etc., e também por Widgets invisíveis, como linhas, colunas e grades.
- Esses Widgets invisíveis usamos para organizar a tela, alinhando os Widgets visíveis e delimitando o espaço que eles ocupam.

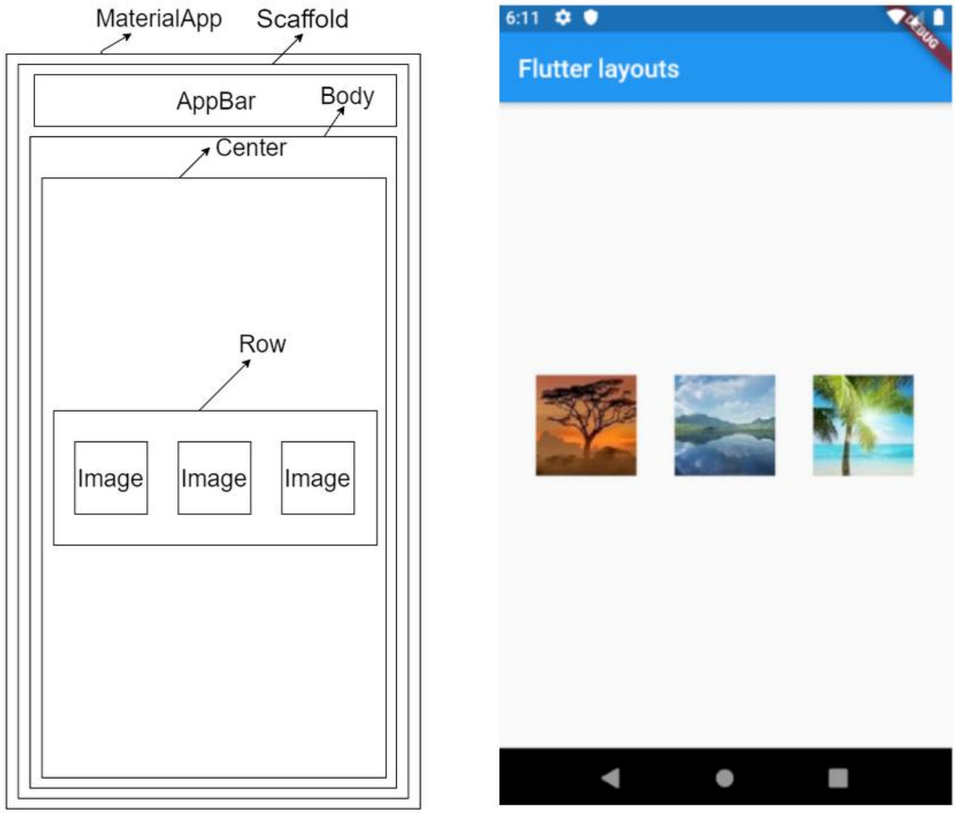

Figura 01 - Layouts
Fonte: <https://www.devmedia.com.br/flutter-criando-layouts-com-center-column-e-row/40743>

Center

- Centraliza todos os seus Widgets filhos. E para customizar a forma como essa centralização ocorre, podemos usar propriedades que determinam as dimensões de Center.
  - heightFactor: Se o valor não for nulo, define a altura de Center pela altura do filho multiplicado por esse valor.
  - widthFactor: Se o valor não for nulo, define a largura de Center pela largura do filho multiplicado por esse valor.

Figura 02 - Layout center
Fonte: <https://www.devmedia.com.br/flutter-criando-layouts-com-center-column-e-row/40743>

Center

```dart
Center(
  heightFactor: 2,
  child: Text('Olá Mundo!',
      textDirection: TextDirection.ltr,
      style: TextStyle(
        fontSize: 32,
        color: Colors.black,
      )), // TextStyle // Text
), // Center
```

Código completo disponível em layoutCenter.dart

- Neste exemplo, a propriedade heightFactor define a altura de Center multiplicando seu valor (2) pela altura do Text filho.

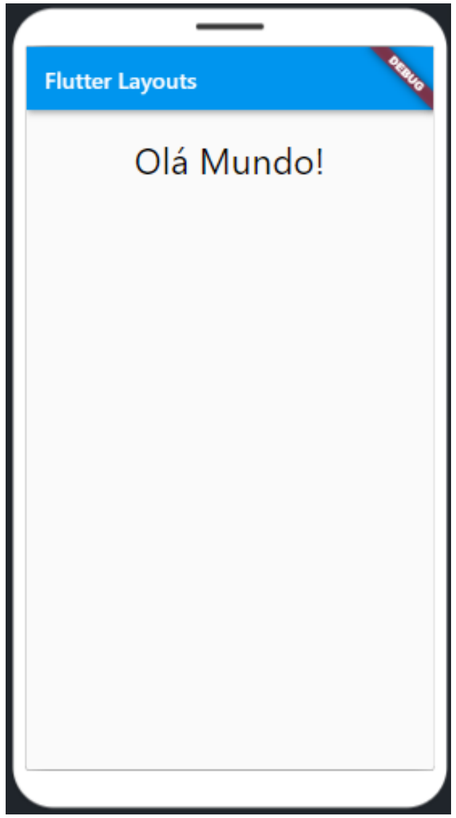

Figura 03 - Execução center

Center

Podemos ver na Figura 4 que a altura de Row é o dobro da altura de Text, conforme foi definido pela propriedade heightFactor.

Figura 04 - HeightFactor
Fonte: <https://www.devmedia.com.br/flutter-criando-layouts-com-center-column-e-row/40743>

Column

Com esse Widget alinhamos os Widgets na tela do app no sentido vertical, como se fizessem parte de uma coluna.

Para customizar esse alinhamento utilizamos as seguintes propriedades:

- mainAxisAlignment: que alinha os filhos no eixo principal.
- crossAxisAlignment: que alinha os filhos no eixo transversal

Em um Widget Column o eixo principal é vertical e o eixo transversal é horizontal.

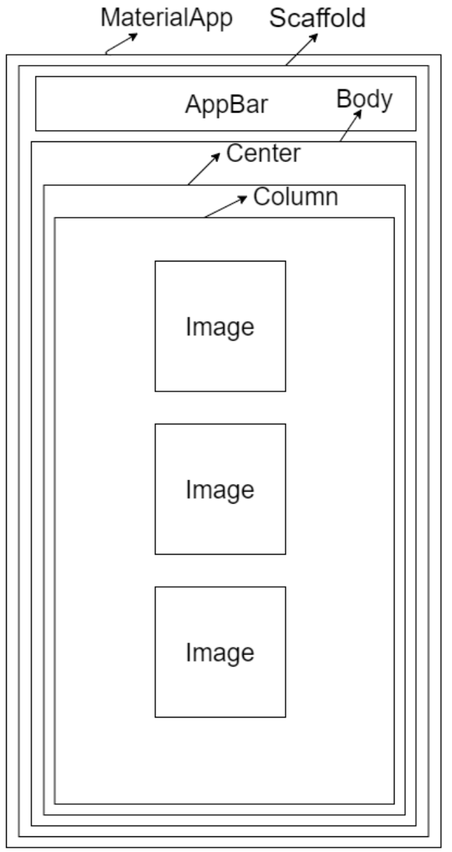

Figura 05 - Layout column
Fonte: <https://www.devmedia.com.br/flutter-criando-layouts-com-center-column-e-row/40743>

Column

Criação de layout com Column e três widgets images, disponível em layoutColumn.dart

- Para utilizarmos imagens em projetos Flutter, conforme o exemplo acima, temos que criar um diretório chamado images na raiz do projeto e copiar para ele as **imagens** que queremos usar, para esse projeto utilize a pasta images. Também precisamos alterar o arquivo **pubspec.yaml** no diretório raiz.

Para isso, abra o arquivo e localize o seguinte trecho de código dentro dele:

```
# The following section is specific to Flutter.
flutter:
  assets:
```

Altere para o seguinte:

```
# The following section is specific to Flutter.
flutter:
  assets:
     - images/lands_01.jpg
     - images/lands_02.jpg
     - images/lands_03.jpg
```

Column

Após executar o app, o resultado deve se parecer com isso:

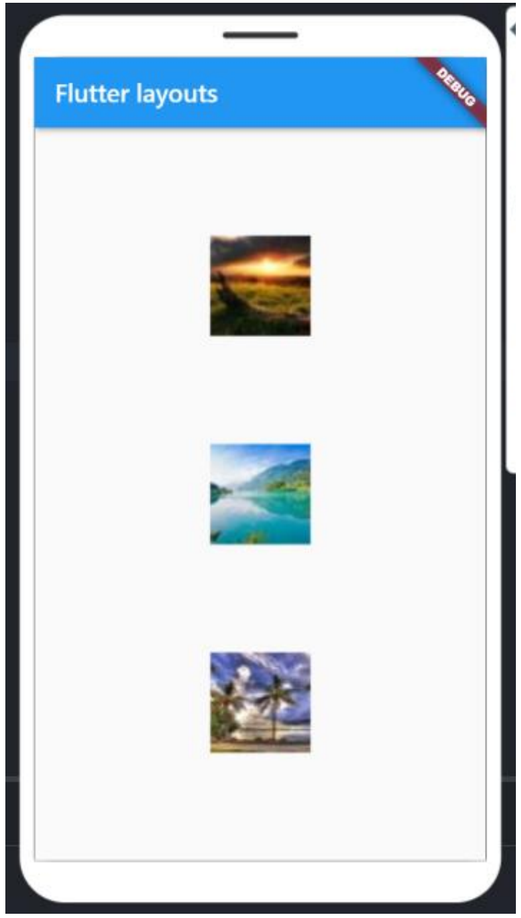

Figura 06 - Execução column

Row

Usamos esse Widget para alinhamento horizontal, utilizando as seguintes propriedades:

- mainAxisAlignment: alinha os filhos no eixo principal.
- crossAxisAlignment: alinha os filhos no eixo transversal.

Em um Widget Row o eixo principal é horizontal e o eixo transversal é vertical.

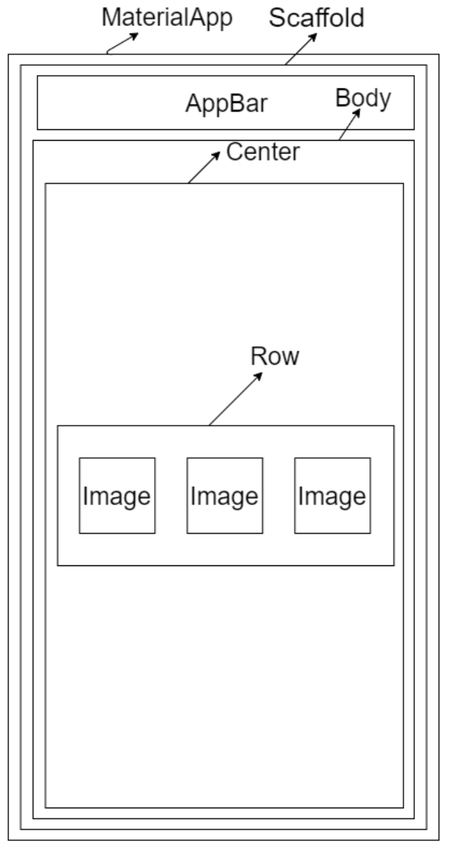

Figura 07 - Layout row
Fonte: <https://www.devmedia.com.br/flutter-criando-layouts-com-center-column-e-row/40743>

Row

No exemplo disponível em layoutRow.dart, usamos Row para exibir três imagens em uma tela do app, uma ao lado da outra.

Figura 08 - Execução row

Container

- A classe Container é um widget de conveniência que combina pintura, posicionamento e dimensionamento comuns de widgets.
- Uma classe Container pode ser usada para armazenar um ou mais widgets e posicioná-los na tela de acordo com nossa conveniência.
- Basicamente, um contêiner é como uma caixa para armazenar o conteúdo.

Figura 09 - Container
Fonte: <https://www.darttutorial.org/flutter-tutorial/flutter-container/>

Referências

DIAS, R. **Flutter: Criando layouts com Center, Column e Row.** Disponível em: <https://www.devmedia.com.br/flutter-criando-layouts-com-center-column-e-row/40743>. Acesso em: 7 mar. 2024.

Classe de contêiner em Flutter – Acervo Lima. Disponível em: <https://acervolima.com/classe-de-conteiner-em-flutter/>. Acesso em: 7 mar. 2024.

Building user interfaces with Flutter. Disponível em: <https://docs.flutter.dev/ui>. Acesso em: 7 mar. 2024.

Flutter Container. Disponível em: <https://www.darttutorial.org/flutter-tutorial/flutter-container/>. Acesso em: 8 mar. 2024.