

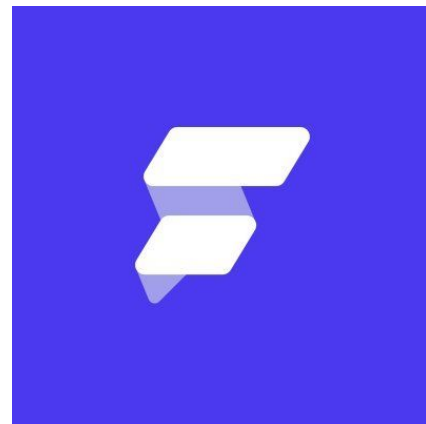


Flutter Avançado - Aula 1

Exercícios 1 a 4



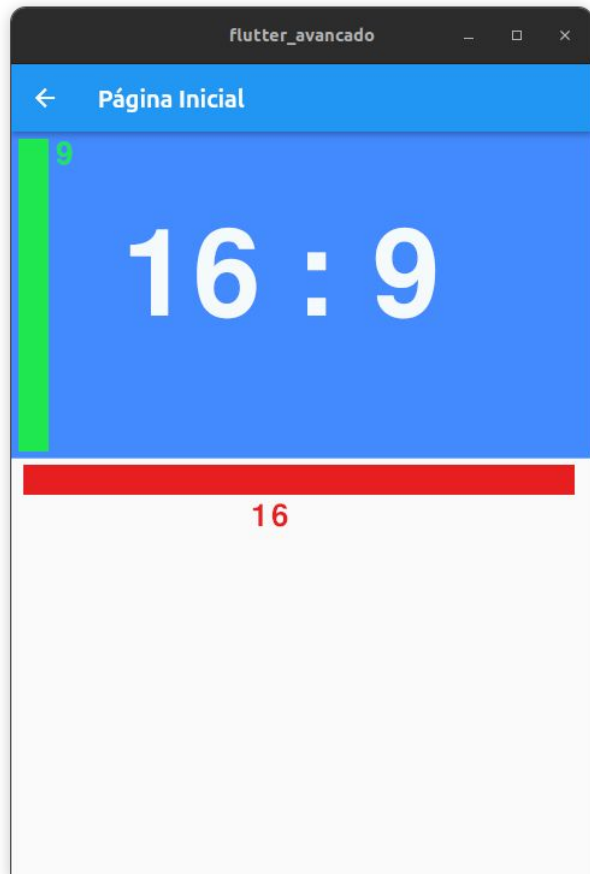
- FlutterFlow [Ver mais](#)





- Widget que se redimensionará de acordo com o tamanho da tela
- Você deverá informar uma proporção de tela (*aspectRatio*)
- O widget se redimensionará de acordo com a proporção passada

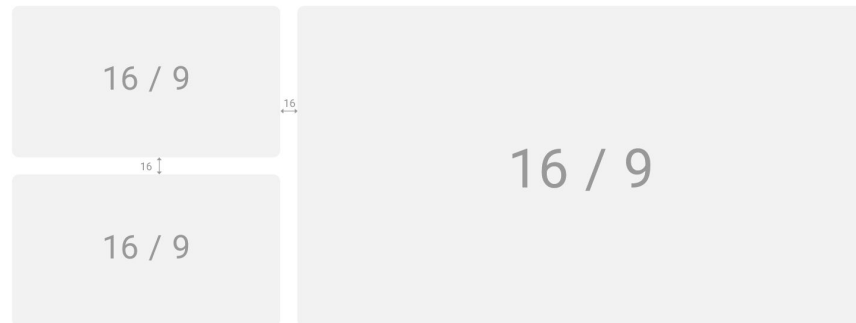
```
body: AspectRatio(  
  aspectRatio: 16 / 9,  
  child: Container(  
    color: Colors.blueAccent,  
  ), // Container  
) // AspectRatio
```





O comportamento do *AspectRatio* pode ser afetado em duas situações:

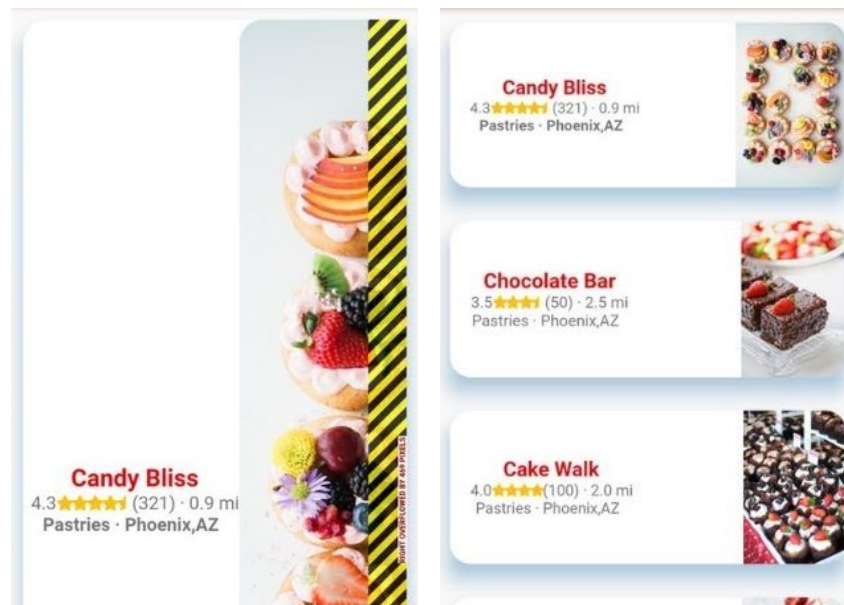
- Caso o *AspectRatio* esteja dentro de um *expanded*, nesse caso, seu tamanho será determinado pelo próprio *Expanded*
- Caso o componente pai do *AspectRatio* tenha um tamanho fixo



- Vamos criar uma lista de imagens
- Cada imagem deve ser exibida com um *aspectRatio* de 3 / 1
- No cabeçalho, adicionar dois botões para aumentar o *aspectRatio*
- Um botão para aumentar o *width* e outro para aumentar o *height*

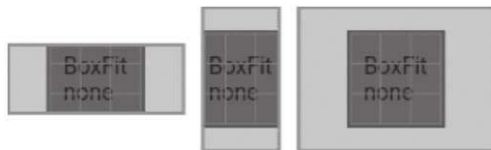


- O *FittedBox* serve para adequar o tamanho de um *widget* filho ao tamanho do seu *widget* pai
- Possui as propriedades *align* e *fit*
- *Align* determinará o alinhamento do item filho dentro do item pai
- *Fit* determinará o comportamento do filho ao preencher o componente pai

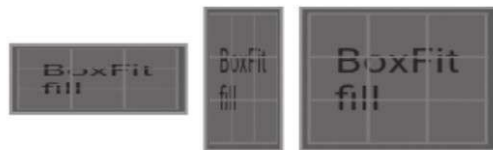


- O BoxFit determina como uma caixa será desenhada dentro de outra caixa

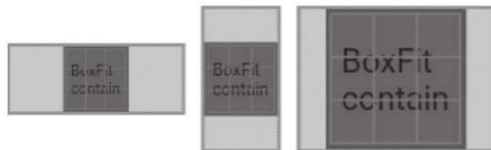
```
color: Colors.tealAccent,
child: FittedBox(
  alignment: Alignment.center,
  fit: BoxFit.none,
  child: Container(
    height: 50,
    width: 50,
    color: Colors.blue,
  ), // Container
), // FittedBox
```



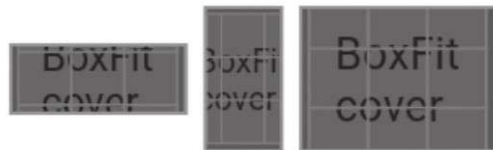
BoxFit.none



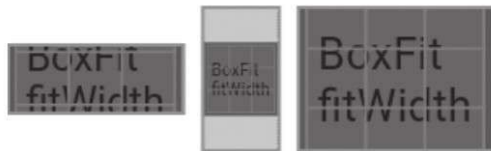
BoxFit.fill



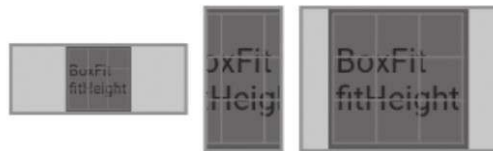
BoxFit.contain



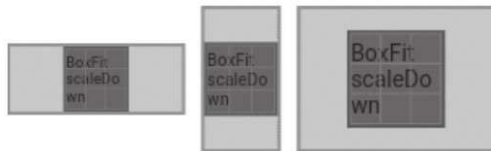
BoxFit.cover



BoxFit.fitWidth



BoxFit.fitHeight



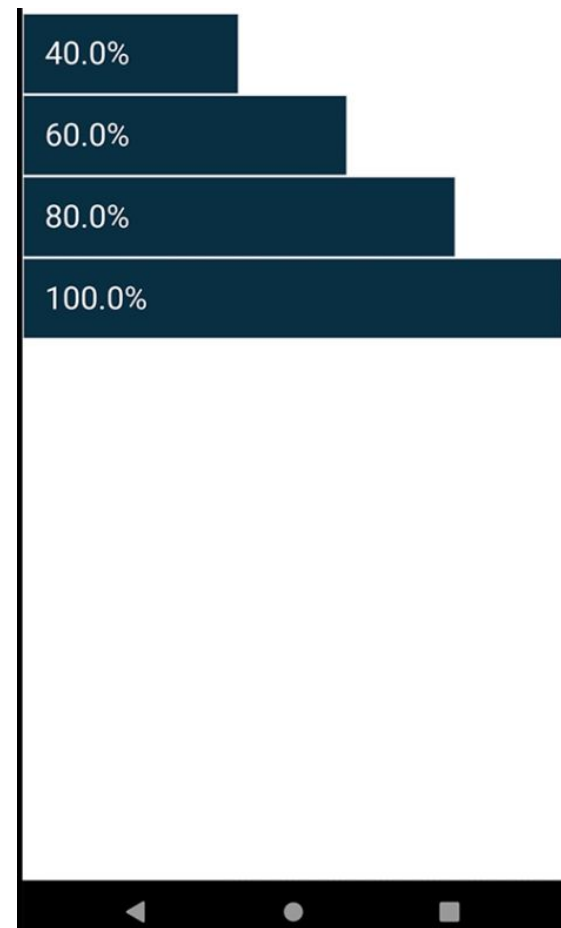
BoxFit.scaleDown

- Vamos criar um app de viagens
- Liste alguns destinos possíveis
- No exemplo ao lado, utilizei uma *ListView*
- Ao invés de utilizar uma *ListTile*, criei uma widget próprio para listar os meus destinos
- Para cada item, utilizei um *FittedBox* para tornar os itens responsivos





- *Widget* que se redimensiona de acordo com percentual definido
- É possível definir tamanho para altura e largura
- Para *Columns* e *Rows*, é necessário além de utilizar o *Flexible*



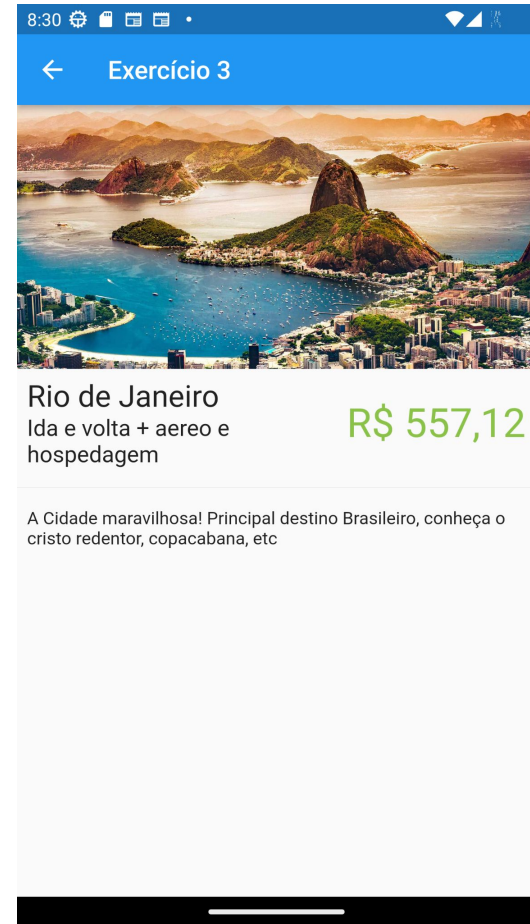


- Um widget que controla como um filho de uma *Row* ou *Column* se comporta
- A propriedade flex (inteiro) determinará quantas porções de espaço o *Flexible* ocupará
- A propriedade Fit determinará o comportamento do *Flexible*
- *FlexFit.loose*: O widget filho pode ser menor que o tamanho do pai
- *FlexFit.tight*: O widget filho ocupará todo o espaço disponível



```
body: Column(  
  children: [  
    Flexible(  
      flex: 1,  
      fit: FlexFit.tight,  
      child: FractionallySizedBox(  
        heightFactor: 0.5,  
        child: Container(  
          color: Colors.yellow,  
        ),  
      ),  
    ),  
    Flexible(  
      flex: 1,  
      fit: FlexFit.tight,  
      child: FractionallySizedBox(  
        heightFactor: 0.5,  
        child: Container(  
          color: Colors.red,  
        ),  
      ),  
    ),  
  ],  
)
```

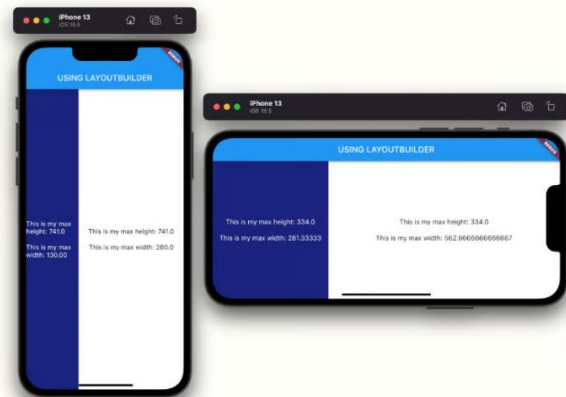
- Continuando o aplicativo de viagens
- Ao clicar em qualquer um dos destinos, deve mostrar os detalhes
- A imagem do app deve ocupar cerca de $\frac{1}{3}$ da tela
- O descritivo deve ocupar o resto





- Similar ao método build, sendo chamado sempre que o *widget* é redimensionado
- Ao criar, você deve implementar um método
- Nesse método, deverá ser retornado um widget

```
Widget build(BuildContext context) {  
  return Scaffold(  
    body: LayoutBuilder(  
      builder: (BuildContext p0, BoxConstraints p1) {  
        return Column(  
          children: [  
            Text("Height: ${p1.minHeight} - ${p1.maxHeight}"),  
            Text("Width: ${p1.minWidth} - ${p1.maxWidth}"),  
          ],  
        ); // Column  
      }  
    ), // LayoutBuilder  
  ); // Scaffold  
}
```





- Vamos criar um app que liste três ícones
- De acordo com o tamanho da tela, os ícones devem ser exibidos um embaixo do outro ou um do lado do outro

