

Desenvolvimento
Mobile 3
Aula 02

Prof. Me Daniel Vieira



Agenda

- 1- Responsividade
- 2 - Exemplos
- 3- Aplicativo
- 4 - Definição do projeto livre

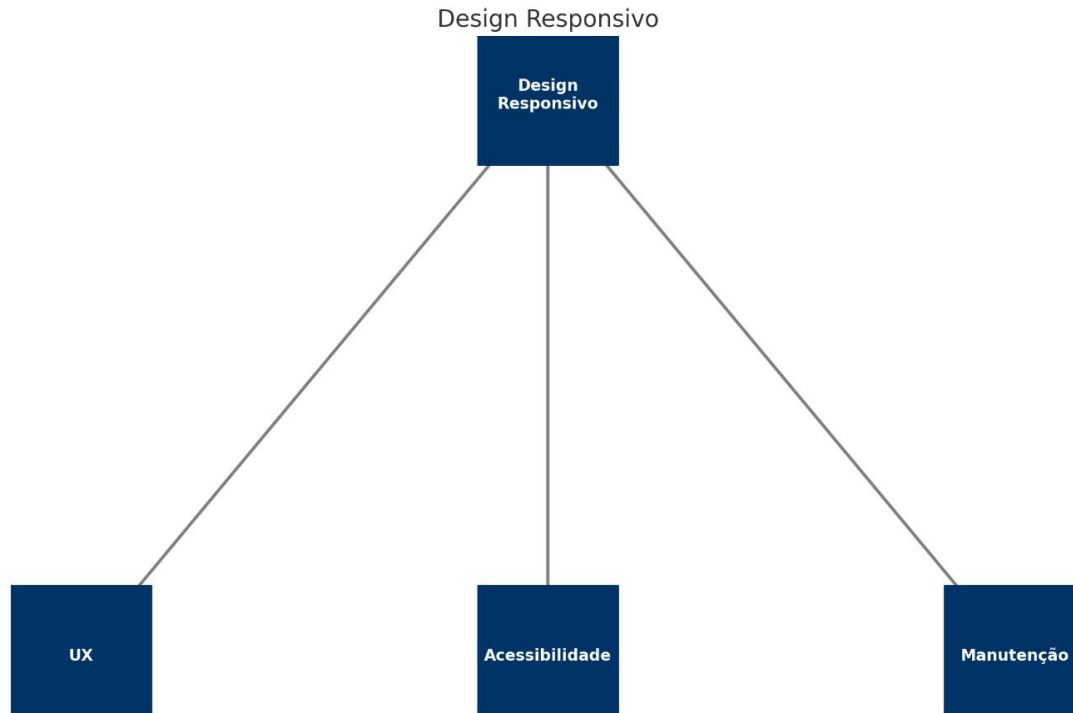
Responsividade

O que é responsividade ?

O design responsivo é uma abordagem de design que visa criar interfaces que se adaptam a diferentes tamanhos de tela e dispositivos. No Flutter, isso significa garantir que seu aplicativo funcione bem em celulares, tablets e desktops, ajustando-se de maneira flexível às mudanças na resolução e orientação.

Responsividade

Porque criar um app responsivo ?



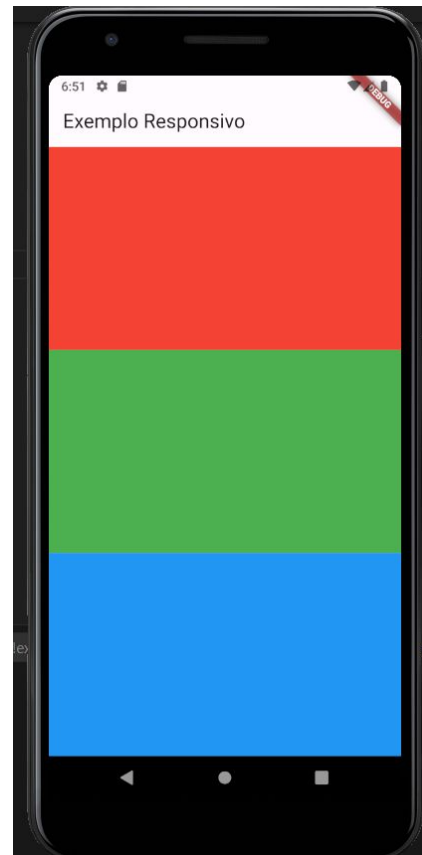
Estruturas e Widgets Básicos

Flutter oferece vários widgets que ajudam a criar layouts flexíveis. Vamos ver alguns dos mais importantes:

- **Container**: Um widget básico que pode ser usado para adicionar padding, margin e outras propriedades de layout.
- **Row e Column**: Widgets que organizam outros widgets em uma linha ou coluna.
- **Expanded**: Um widget que expande seu filho para preencher o espaço disponível dentro de um **Row** ou **Column**.
- **Flexible**: Similar ao **Expanded**, mas permite definir uma proporção do espaço disponível.

Plano de Ensino

```
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   @override
9   Widget build(BuildContext context) {
10     return MaterialApp(
11       home: Scaffold(
12         appBar: AppBar(title: Text('Exemplo Responsivo')),
13         body: Column(
14           children: [
15             Expanded(
16               child: Container(color: Colors.red),
17             ),
18             Expanded(
19               child: Container(color: Colors.green),
20             ),
21             Expanded(
22               child: Container(color: Colors.blue),
23             ),
24           ],
25         ),
26       ),
27     );
28   }
29 }
```



Exemplo com MediaQuery e LayoutBuilder

2.2. MediaQuery e LayoutBuilder

- **MediaQuery:** Permite acessar as dimensões da tela e a orientação do dispositivo.
- **LayoutBuilder:** Fornece a capacidade de construir widgets com base no tamanho disponível.

Exemplo com MediaQuery e LayoutBuilder

2.2. MediaQuery e LayoutBuilder

- **MediaQuery:** Permite acessar as dimensões da tela e a orientação do dispositivo.
- **LayoutBuilder:** Fornece a capacidade de construir widgets com base no tamanho disponível.

Exemplo com MediaQuery e LayoutBuilder

2.2. MediaQuery e LayoutBuilder

- **MediaQuery:** Permite acessar as dimensões da tela e a orientação do dispositivo.
- **LayoutBuilder:** Fornece a capacidade de construir widgets com base no tamanho disponível.

App responsivo

```
import 'package:flutter/material.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Responsive Demo',
      home: ResponsiveHome(),
    );
  }
}
```

App responsivo

```
class ResponsiveHome extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    // Obter as informações de MediaQuery  
    final mediaQueryData = MediaQuery.of(context);  
    final screenWidth = mediaQueryData.size.width;  
    final screenHeight = mediaQueryData.size.height;  
  
    // Definir os breakpoints  
    final isMobile = screenWidth < 600;  
    final isTablet = screenWidth >= 600 && screenWidth < 1024;  
    final isDesktop = screenWidth >= 1024;
```

App responsivo

```
return Scaffold(  
  appBar: AppBar(  
    title: Text('Flutter Responsive Demo'),  
  ),  
  drawer: isMobile  
    ? Drawer(  
      child: ListView(  
        children: [  
          DrawerHeader(  
            child: Text('Menu'),  
          ),  
          ListTile(  
            title: Text('Item 1'),  
          ),  
          ListTile(  
            title: Text('Item 2'),  
          ),  
        ],  
      ),  
    )  
)
```

Obrigado!

Prof. Me Daniel Vieira

Email: danielvieira2006@gmail.com

Linkedin: Daniel Vieira

Instagram: Prof daniel.vieira95

