



# Flutter

Wesley Dias Maciel

2021/02

# Prática 06

## Stack

Documentação: <https://api.flutter.dev/flutter/widgets/Stack-class.html>

**Objetivo:** apresentar o widget básico Stack. O widget Stack gera uma caixa e posiciona seus filhos em relação às bordas dessa caixa. Esta classe é útil se você deseja sobrepor vários filhos. Exemplo: sobrepor um texto e uma imagem com um botão anexado na parte inferior.

Ao invés de ser orientado horizontal ou verticalmente, um widget Stack permite que você coloque widgets filhos uns sobre os outros. Os objetos filhos de Stack são posicionados no eixo z do plano cartesiano. Um widget Stack também permite que você use o widget **Positioned**, para determinar o posicionamento dos widgets filhos de Stack. Com o widget **Positioned**, você pode posicionar os filhos de Stack em relação à borda superior, direita, inferior ou esquerda de Stack. Objetos da classe Stack são baseados no modelo de layout de posicionamento **absolute** da Web.

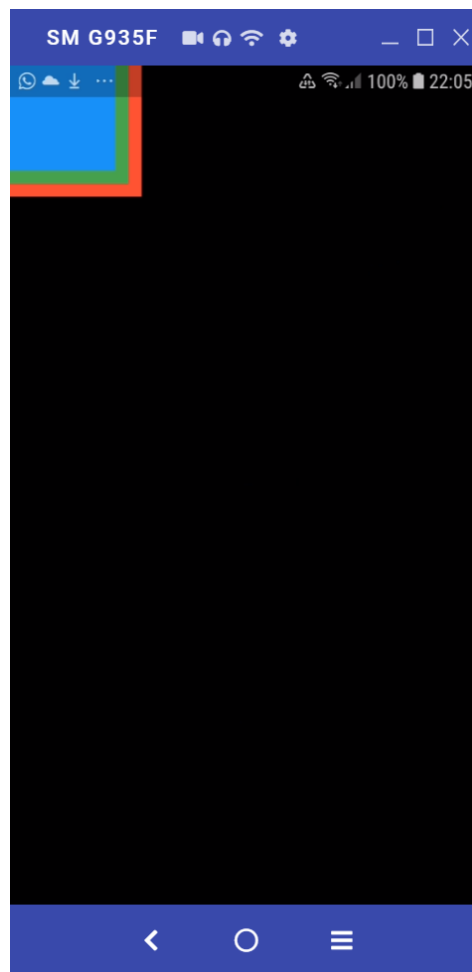
O widget Stack posiciona seus filhos com base na ordem em que eles são declarados, com o primeiro filho posicionado na parte inferior. Se você quiser alterar a ordem em que os filhos são posicionados, você terá que declarar os filhos na nova ordem.

- 1) No Visual Studio Code, crie um novo projeto Flutter. Clique em View > Command Palette ou pressione CTRL + SHIFT + P. Na caixa de entrada, informe:  
**>Flutter: New Project**
- 2) Na nova janela aberta, selecione um diretório para seu projeto. **OBS:** o caminho do diretório deve ter apenas caracteres ASCII. Além disso, o caminho do diretório não pode ter espaços em branco. Dessa forma, não use: acentos, símbolos, espaços, cedilha, etc. Exemplo: C:\projeto\flutter.
- 3) Na nova caixa de entrada, informe um nome para seu projeto. Exemplo: pratica06. Em seguida, pressione a tecla ENTER. **OBS:** o nome do projeto deve ter apenas caracteres ASCII (<https://pt.wikipedia.org/wiki/ASCII>). Além disso, o nome do projeto não pode ter espaços em branco. Dessa forma, não use: acentos, símbolos, espaços, cedilha, etc.
- 4) Aguarde a finalização do processo de criação de seu projeto. Após a criação de seu projeto, observe a estrutura de diretórios criada e o código inicial gerado como exemplo em lib\main.dart. Apague o código inicial gerado como exemplo em lib\main.dart.

- 5) Um aplicativo Flutter mínimo simplesmente chama a função `runApp()` com um widget. No exemplo abaixo, a função `runApp()` é chamada com um widget para empilhamento (Stack). O widget Stack empilha 3 widgets Container. O primeiro contêiner declarado é o vermelho. Por isso, ele é a base da pilha. O segundo contêiner declarado é o verde, segundo da pilha. Por fim, o último contêiner declarado é o azul. Por isso, ele é o topo da pilha.

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

void main() {
  runApp(
    Stack(
      textDirection: TextDirection.ltr,
      children: <Widget>[
        Container(
          width: 100,
          height: 100,
          color: Colors.red,
        ),
        Container(
          width: 90,
          height: 90,
          color: Colors.green,
        ),
        Container(
          width: 80,
          height: 80,
          color: Colors.blue,
        ),
      ],
    ),
  );
}
```

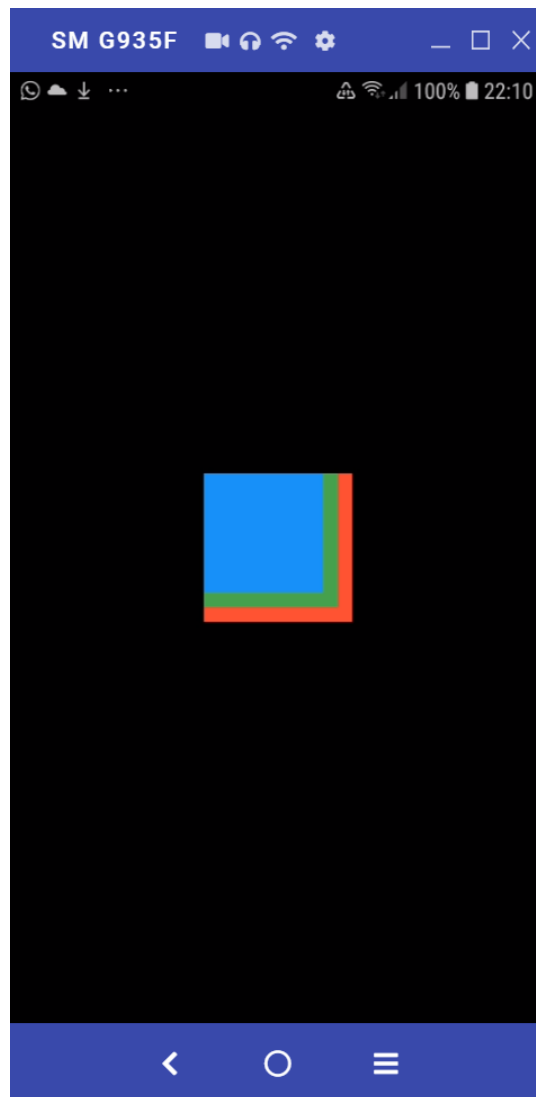


6) O próximo exemplo apresenta a pilha de widgets centralizada.

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

void main() {
  runApp(
    Center(
      child: Stack(
        textDirection: TextDirection.ltr,
        children: <Widget>[
          Container(
            width: 100,
            height: 100,
            color: Colors.red,
          ),
          Container(
            width: 90,
            height: 90,
            color: Colors.green,
          ),
          Container(
```

```
width: 80,  
height: 80,  
color: Colors.blue,  
,  
],  
,  
,  
);  
}
```



- 7) O exemplo abaixo apresenta um widget Center que centraliza uma caixa (SizedBox) de 250 pixels de largura (width) e 250 pixels de altura (height). Dentro dessa caixa, há uma pilha (Stack). A pilha possui 2 filhos: 2 contêineres (Container). O primeiro contêiner possui fundo branco, 250 pixels de largura e 250 pixels de altura. O segundo contêiner possui espaçamento entre borda e conteúdo de 5 pixels (padding: EdgeInsets.all(5.0)). O

conteúdo desse segundo contêiner é centralizado na parte de baixo do contêiner (alignment: Alignment.bottomCenter). Esse segundo contêiner é decorado com um gradiente na cor preta. Por fim, o segundo contêiner também possui um filho: um texto (Text). O texto é configurado com cor branca e fonte de tamanho 20 pixels.

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

void main() {
  runApp(
    Center(
      child: SizedBox(
        width: 250,
        height: 250,
        child: Stack(
          textDirection: TextDirection.ltr,
          children: <Widget>[
            Container(
              width: 250,
              height: 250,
              color: Colors.white,
            ),
            Container(
              padding: EdgeInsets.all(5.0),
              alignment: Alignment.bottomCenter,
              decoration: BoxDecoration(
                gradient: LinearGradient(
                  begin: Alignment.topCenter,
                  end: Alignment.bottomCenter,
                  colors: <Color>[
                    Colors.black.withAlpha(0),
                    Colors.black12,
                    Colors.black45
                  ],
                ),
              ),
            ),
            child: Text(
              "Google Flutter",
              textDirection: TextDirection.ltr,
              style: TextStyle(color: Colors.white, fontSize: 20.0),
            ),
          ),
        ],
      ),
    ),
  );
};
```

}



## Exercício

- 1) Altere o exemplo desta prática, para que ele apresente uma tela semelhante à do exemplo abaixo.

Link para o exercício: <https://media.giphy.com/media/xT0xezQGU5xCDJuCPe/giphy.gif>

