



Desenvolvimento  
Mobile 1  
Aula 09

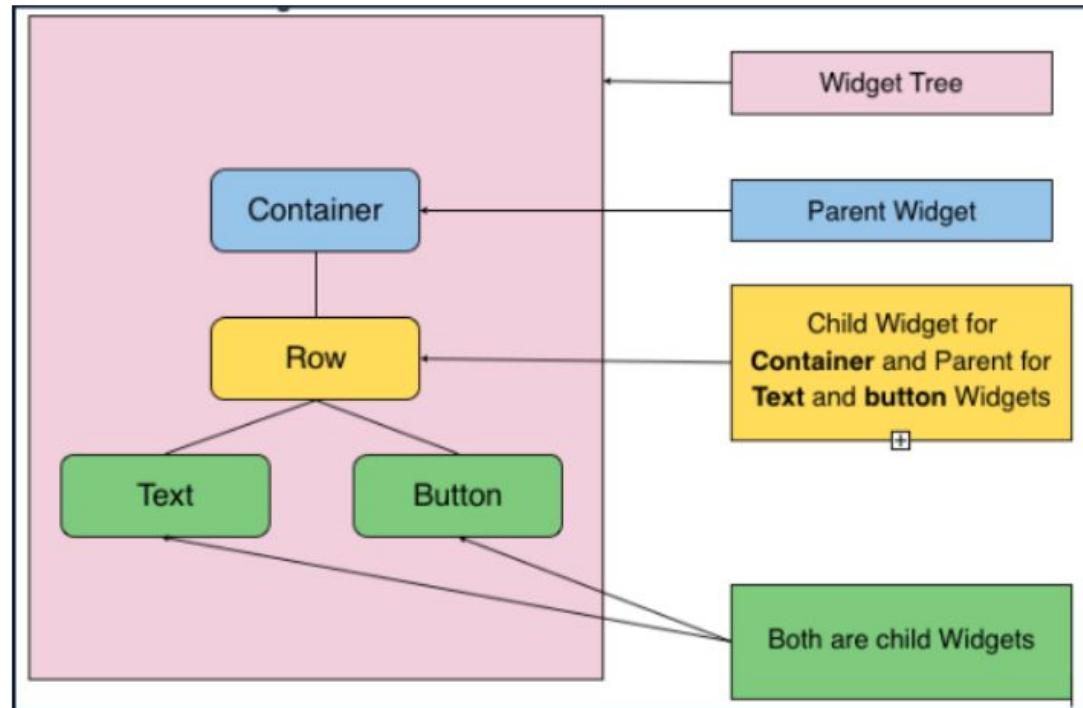
Prof. Me Daniel Vieira

**SENAI**

# Agenda

- 1- Flutter Widgets
- 2- Estrutura de um app
- 3-Material Design
- 4 -Exercícios

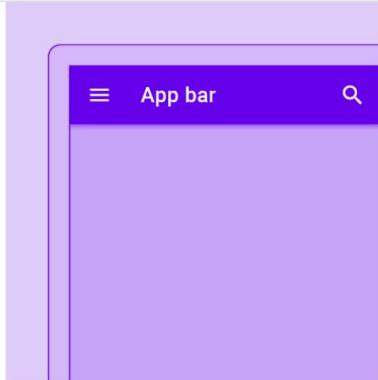
# Flutter Widgets



# Flutter Widgets

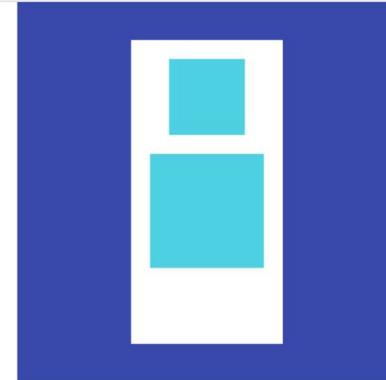
<https://docs.flutter.dev/ui/widgets/basics>

<https://m3.material.io/components/icon-buttons/overview>



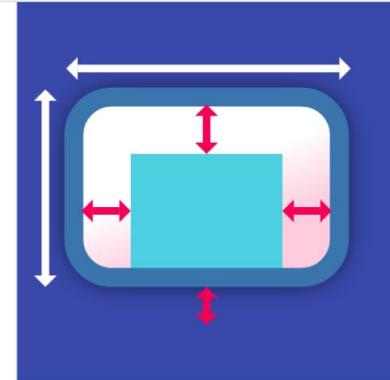
## AppBar

A toolbar that might contain other widgets such as a `TabBar` and a `FlexibleSpaceBar`.



## Column

Layout a list of child widgets in the vertical direction.



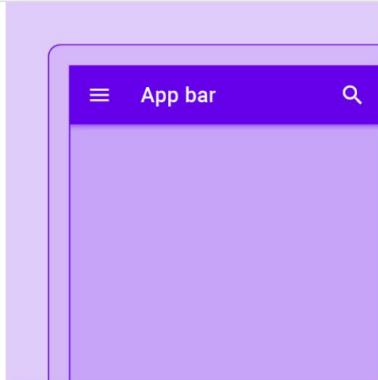
## Container

A convenience widget that combines common painting, positioning, and sizing widgets.

# Flutter Widgets

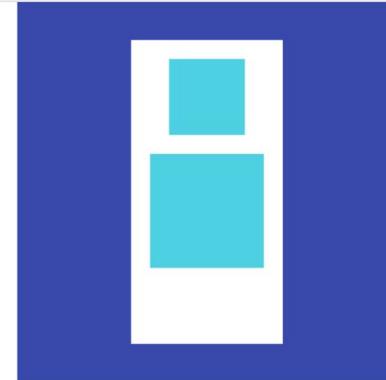
<https://docs.flutter.dev/ui/widgets/basics>

<https://m3.material.io/components/icon-buttons/overview>



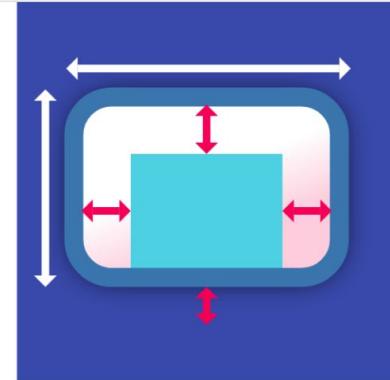
## AppBar

A toolbar that might contain other widgets such as a `TabBar` and a `FlexibleSpaceBar`.



## Column

Layout a list of child widgets in the vertical direction.



## Container

A convenience widget that combines common painting, positioning, and sizing widgets.

# Material ou Cupertino



[Column](#)

Layout a list of child widgets in the vertical direction.



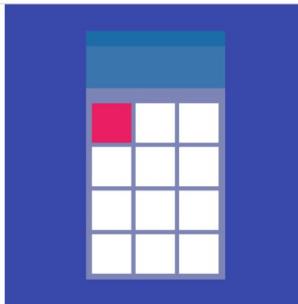
[CustomMultiChildLayout](#)

A widget that uses a delegate to size and position multiple children.



[Flow](#)

A widget that implements the flow layout algorithm.



[GridView](#)



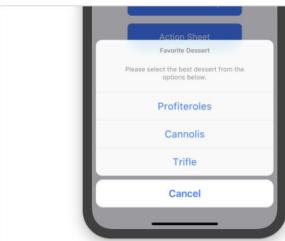
[IndexedStack](#)



[LayoutBuilder](#)

# Flutter Widgets Cupertino

<https://docs.flutter.dev/ui/widgets/cupertino>



[CupertinoActionSheet](#)

An iOS-style modal bottom action sheet to choose an option among many.



[CupertinoActivityIndicator](#)

An iOS-style activity indicator. Displays a circular 'spinner'.

Allow "Maps" to access your location while you use the app?

Your current location will be displayed on the map and used for directions, nearby search results, and estimated travel times.

[Don't Allow](#)   [Allow](#)

[CupertinoAlertDialog](#)

An iOS-style alert dialog.

Button

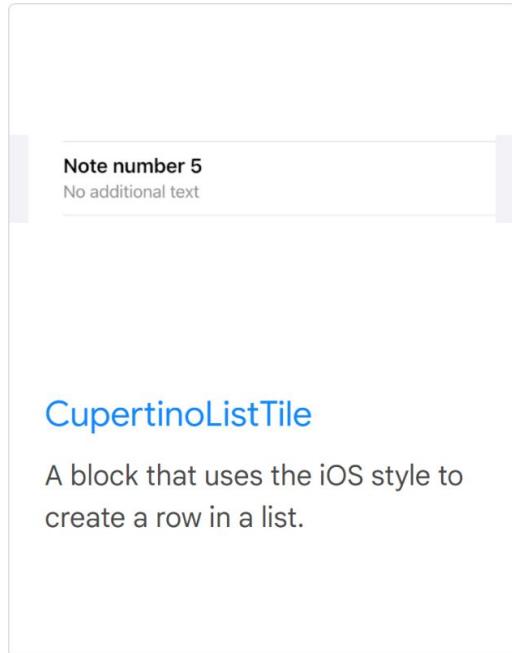


|           |    |       |
|-----------|----|-------|
| Wed Oct 3 | 9  | 31    |
| Thu Oct 4 | 10 | 32    |
| Fri Oct 5 | 11 | 33 AM |
| Sat Oct 6 | 12 | 34 PM |
| Sun Oct 7 | 1  | 35    |



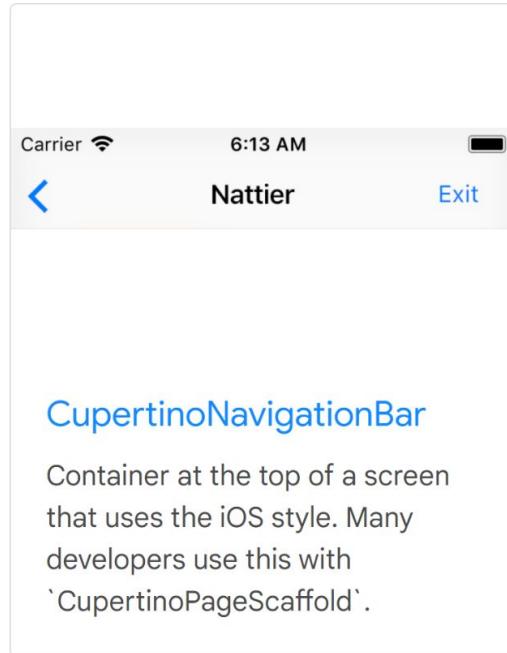
# Flutter Widgets

<https://docs.flutter.dev/ui/widgets/cupertino>



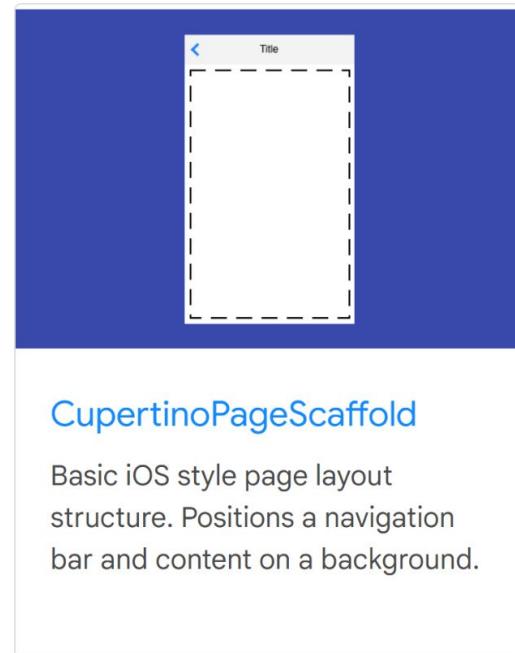
## CupertinoListTile

A block that uses the iOS style to create a row in a list.



## CupertinoNavigationBar

Container at the top of a screen that uses the iOS style. Many developers use this with 'CupertinoPageScaffold'.



## CupertinoPageScaffold

Basic iOS style page layout structure. Positions a navigation bar and content on a background.

# Widgets

O que são Widgets: Os Elementos básicos do Flutter que são Classes em Dart com métodos e parâmetros manipuláveis para poder dar vida a nossa criatividade, como peças de LEGO.

Entendemos que os Widgets podem ser relacionados/conectados (como Mães e Filhas) através de parâmetros child/children.

Container:

Um Widget que permite a criação de um espaço retangular com tamanho vertical (height) e tamanho horizontal (width), cores (color), filha(o) única(o) (child), além de outros parâmetros que podemos verificar no catálogo de Widgets da documentação.

Stack:

Um Widget com o poder de desenhar seus filhos um por cima do outro, com parâmetros como filhos múltiplos (children), alinhamento (alignment), além de outros parâmetros que podemos verificar no catálogo de Widgets da

# Widgets

Um Widget com o poder de desenhar seus filhos um ao lado do outro verticalmente, com parâmetros como filhos múltiplos (children), alinhamento vertical (mainAxisAlignment), alinhamento horizontal (crossAxisAlignment), além de outros parâmetros que podemos verificar no catálogo de Widgets da documentação.

Row:

Um Widget com o poder de desenhar seus filhos um ao lado do outro horizontalmente, com parâmetros como filhos múltiplos (children), alinhamento horizontal (mainAxisAlignment), alinhamento vertical (crossAxisAlignment), além de outros parâmetros que podemos verificar no catálogo de Widgets da documentação.

ElevatedButton:

O Primeiro Botão que criamos! Tem poder de ação com o parâmetro onPressed, além do parâmetro childque permite adicionar um novo Widget dentro do ElevatedButton, além de outros parâmetros que podemos verificar no catálogo de Widgets da documentação.

# Widgets

Text:

O Widget responsável por usar uma String para mostrar na tela informação escrita, no formato que desejamos! Com os parâmetros de style temos o poder de alterar o tamanho, a cor e muitas outras características do nosso texto.

Overflow:

Quando não temos tanta noção de espaço dentro da nossa tela, nossos Widgets podem ser maiores do que o espaço visível, nessas ocasiões as informações dos nossos Widgets podem não estar completamente visíveis para o usuário do aplicativo, perdendo informação importante por falta de espaço. Esse defeito é denominado Overflow.

Árvore de Widgets: Para que fique mais claro a relação de Widgets pais e Widgets filhos precisamos entender claramente o conceito da Árvore de Widgets. Usamos algumas ferramentas do Android Studio para facilitar nossa compreensão da árvore. Com o Flutter Inspector e o Layout Explorer podemos entender como os Widgets estão dispostos na tela, inclusive visualizar seus parâmetros e seus limites!

# App Bar

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

/// Flutter code sample for [AppBar].

void main() => runApp(const AppBarApp());

class AppBarApp extends StatelessWidget {
  const AppBarApp({super.key});

  @override
  Widget build(BuildContext context) {
    return const MaterialApp(
      home: AppBarExample(),
    );
  }
}
```

# App Bar

```
class AppBarExample extends StatelessWidget {  
  const AppBarExample({super.key});  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: const Text('AppBar Demo'),  
        actions: <Widget>[  
          IconButton(  
            icon: const Icon(Icons.add_alert),  
            tooltip: 'Show Snackbar',  
            onPressed: () {  
              ScaffoldMessenger.of(context).showSnackBar(  
                const SnackBar(content: Text('This is a snackbar')));  
            },  
        ),  
      ),  
    );  
  }  
}
```

# AppBar

```
IconButton(  
  icon: const Icon(Icons.navigate_next),  
  tooltip: 'Go to the next page',  
  onPressed: () {  
    Navigator.push(context, MaterialPageRoute<void>(  
      builder: (BuildContext context) {  
        return Scaffold(  
          appBar: AppBar(  
            title: const Text('Next page'),  
          ),  
          body: const Center(  
            child: Text(  
              'This is the next page',  
              style: TextStyle(fontSize: 24),  
            ),  
            ),  
          );  
        },  
      );  
    }  
  );  
}
```

# Obrigado!

Prof. Me Daniel Vieira

Email: [danielvieira2006@gmail.com](mailto:danielvieira2006@gmail.com)

Linkedin: Daniel Vieira

Instagram: Prof daniel.vieira95

