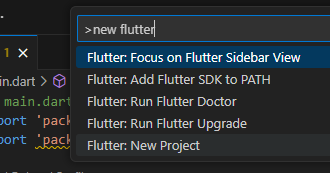
# Widget Stack

* Stack permite sobrepor vários widgets uns aos outros e os ordena de baixo para cima, posiciona seus filhos em relação às bordas de sua caixa.
* Filhos posicionados são aqueles encapsulados em um widget [Positioned](https://api.flutter.dev/flutter/widgets/Positioned-class.html) que tem pelo menos uma propriedade não nula.

Texto preto sobre fundo branco

Descrição gerada automaticamente

* **CTRL + SHIFT + P: para abrir a paleta de comandos**
* **Clique em New Project – Application – selecione a pasta para o projeto**



* A classe Stack tem a propriedade **Alignment**(0.0, 0.0)
* **AlignmentDirectional**: orientação de coordenadas horizontais

Exemplo:

Uma imagem contendo Ícone

Descrição gerada automaticamenteTexto

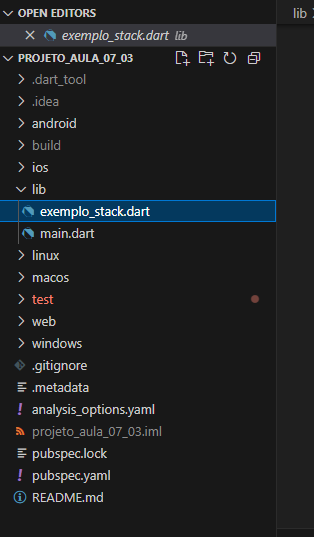
Descrição gerada automaticamente

Texto

Descrição gerada automaticamenteCódigo:

# Criando um projeto com Stack

* Crie um novo projeto
* Na pasta lib crie um arquivo com o nome: widget\_stack.dart
* Criando uma classe WidgetStack que herdará de StatelessWidget



// exemplo\_stack.dart

import 'package:flutter/material.dart';

class ExemploStack extends StatelessWidget {

  const ExemploStack({super.key});

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      //Layout

      appBar: AppBar(

        title: Center(

          child: Text(

            'Trabalhando com Stack',

            style: TextStyle(color: const Color.fromARGB(255, 19, 95, 3)),

          ),

        ),

        backgroundColor: Colors.amber,

      ),

      body: criarStack(),

      backgroundColor: Colors.grey,

    );

  }

}

Stack criarStack() {

  return Stack(

    children: <Widget>[

      Image.network(

        'https://marketplace.canva.com/EAFfbzRyhmQ/2/0/900w/canva-fundo-de-tela-para-celular-aquarela-m%C3%A1rmore-minimalista-aesthetic-rosa-e-branco-jXm\_DWP3Iu4.jpg',

        height: double.infinity,

        width: double.infinity,

        fit: BoxFit.cover,

      ),

      Positioned(bottom: 100, left: 130, child: Text('DS Turma B')),

    ],

  );

}

**1. Importação do Material Design**

import 'package:flutter/material.dart';

* Importa a biblioteca **Material** do Flutter, que contém os widgets básicos para a interface do usuário.

**2. Classe ExemploStack**

class ExemploStack extends StatelessWidget {

const ExemploStack({super.key});

@override

Widget build(BuildContext context) {

return Scaffold(

* Define a **classe ExemploStack**, que estende **StatelessWidget**, ou seja, um widget sem estado.
* O método **build** é responsável por desenhar a interface.

**3. Criando a Estrutura do App (Scaffold)**

appBar: AppBar(

title: Center(

child: Text(

'Trabalhando com Stack',

style: TextStyle(color: const Color.fromARGB(255, 19, 95, 3)),

),

),

backgroundColor: Colors.amber,

),

* **AppBar**: Barra superior do aplicativo.
* **title**: Contém um texto centralizado com o título "Trabalhando com Stack".
* **Estilização**:
  + O texto é verde escuro (Color.fromARGB(255, 19, 95, 3)).
  + O fundo da AppBar é amarelo (Colors.amber).

**4. Corpo do Aplicativo (body: criarStack())**

body: criarStack(),

backgroundColor: Colors.grey,

* O **corpo da tela** chama a função criarStack(), que retorna um **Stack**.
* O **fundo da tela** (backgroundColor) é definido como **cinza**.

**5. Criando a Stack**

Stack criarStack() {

return Stack(

children: <Widget>[

* O **Stack** permite sobrepor widgets.
* A propriedade children: <Widget>[] contém os elementos da pilha.

**6. Imagem de Fundo**

Image.network(

'https://marketplace.canva.com/EAFfbzRyhmQ/2/0/900w/canva-fundo-de-tela-para-celular-aquarela-m%C3%A1rmore-minimalista-aesthetic-rosa-e-branco-jXm\_DWP3Iu4.jpg',

height: double.infinity,

width: double.infinity,

fit: BoxFit.cover,

),

* **Imagem de fundo** carregada da internet (Image.network()).
* **Ajustes da imagem**:
  + height: double.infinity: Ocupa toda a altura da tela.
  + width: double.infinity: Ocupa toda a largura da tela.
  + fit: BoxFit.cover: Ajusta a imagem para cobrir todo o espaço disponível.

**7. Texto Posicionado sobre a Imagem**

Positioned(bottom: 100, left: 130, child: Text('DS Turma B')),

* **Positioned**: Posiciona o texto dentro do Stack.
  + bottom: 100: Coloca o texto **100 pixels acima da borda inferior**.
  + left: 130: Desloca o texto **130 pixels a partir da esquerda**.
* **Text('DS Turma B')**: Exibe o texto **"DS Turma B"** sobre a imagem.

No arquivo main.dart

// main.dart

import 'package:flutter/material.dart';

import 'package:projeto\_aula\_07\_03/exemplo\_stack.dart';

void main() {

  runApp(Aplicativo());

}

class Aplicativo extends StatelessWidget {

  const Aplicativo({super.key});

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      debugShowCheckedModeBanner: false,

      title: 'TesteExemplo',

      theme: ThemeData(primarySwatch: Colors.grey),

      home: ExemploStack(),

    );

  }

}

**1. Importação de Bibliotecas**

import 'package:flutter/material.dart';

import 'package:projeto\_aula\_07\_03/exemplo\_stack.dart';

* Importa **Material Design**, que contém os widgets do Flutter.
* Importa **exemplo\_stack.dart**, que contém a tela ExemploStack.

**2. Função main()**

void main() {

runApp(Aplicativo());

}

* **main()**: Ponto de entrada do app.
* **runApp(Aplicativo())**: Inicia o app chamando a classe Aplicativo, que é um **StatelessWidget**.

**3. Classe Aplicativo**

class Aplicativo extends StatelessWidget {

const Aplicativo({super.key});

@override

Widget build(BuildContext context) {

return MaterialApp(

* Define **Aplicativo** como um **StatelessWidget** (não tem estado).
* O método build() retorna um MaterialApp, que gerencia a estrutura do app.

**4. Configuração do MaterialApp**

debugShowCheckedModeBanner: false,

title: 'TesteExemplo',

theme: ThemeData(primarySwatch: Colors.grey),

home: ExemploStack(),

* **debugShowCheckedModeBanner: false** → Esconde a faixa "DEBUG" no canto superior direito.
* **title: 'TesteExemplo'** → Define o título do app.
* **theme: ThemeData(primarySwatch: Colors.grey)** → Define um tema cinza para o app.
* **home: ExemploStack()** → Define a tela ExemploStack como a tela inicial.

Resultado:

Mapa

O conteúdo gerado por IA pode estar incorreto.

Criando Novas telas – Tela de Login e tela de Cadastro

// tela\_cadastro.dart

import 'package:flutter/material.dart';

class TelaCadastro extends StatelessWidget {

  const TelaCadastro({super.key});

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        title: const Text("Cadastro"),

        backgroundColor: Colors.amber,

      ),

      body: Padding(

        padding: const EdgeInsets.all(16.0),

        child: Column(

          mainAxisAlignment: MainAxisAlignment.center,

          children: [

            TextField(

              decoration: InputDecoration(

                labelText: "Nome",

                border: OutlineInputBorder(),

              ),

            ),

            const SizedBox(height: 16),

            TextField(

              decoration: InputDecoration(

                labelText: "Email",

                border: OutlineInputBorder(),

              ),

            ),

            const SizedBox(height: 16),

            TextField(

              obscureText: true,

              decoration: InputDecoration(

                labelText: "Senha",

                border: OutlineInputBorder(),

              ),

            ),

            const SizedBox(height: 16),

            ElevatedButton(

              onPressed: () {

                // Aqui pode adicionar a lógica de cadastro

              },

              child: const Text("Cadastrar"),

            ),

          ],

        ),

      ),

    );

  }

}

Este código define uma **tela de cadastro** no Flutter, onde o usuário pode inserir seu nome, e-mail e senha.

**📌 1. Importação do Material Design**

import 'package:flutter/material.dart';

* Importa os **widgets** do Flutter para criar a interface gráfica.

**📌 2. Classe TelaCadastro**

class TelaCadastro extends StatelessWidget {

const TelaCadastro({super.key});

* Define uma **tela de cadastro** como um StatelessWidget, pois não precisa gerenciar estados.

**📌 3. Método build(BuildContext context)**

@override

Widget build(BuildContext context) {

* O método build **constrói** a interface da tela.

**📌 4. Scaffold (estrutura da tela)**

return Scaffold(

appBar: AppBar(

title: const Text("Cadastro"),

backgroundColor: Colors.amber,

),

* **Scaffold**: Estrutura básica de uma tela no Flutter.
* **AppBar**: Barra superior com o título "Cadastro", de cor **amarela**.

**📌 5. Corpo da tela (body)**

body: Padding(

padding: const EdgeInsets.all(16.0),

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: [

* **Padding**: Adiciona um espaçamento de **16 pixels** ao redor do conteúdo.
* **Column**: Organiza os elementos **verticalmente**.
* **mainAxisAlignment: MainAxisAlignment.center**: Centraliza os elementos na tela.

**📌 6. Campos de Entrada (TextField)**

Cada TextField representa um campo de entrada do formulário.

**📝 Campo Nome**

TextField(

decoration: InputDecoration(

labelText: "Nome",

border: OutlineInputBorder(),

),

),

* O usuário pode inserir o **nome**.
* labelText: "Nome" → Rótulo dentro do campo.
* border: OutlineInputBorder() → Adiciona uma borda ao redor do campo.

**📧 Campo E-mail**

const SizedBox(height: 16),

TextField(

decoration: InputDecoration(

labelText: "Email",

border: OutlineInputBorder(),

),

),

* O usuário pode inserir o **e-mail**.
* SizedBox(height: 16) adiciona um espaço **entre os campos**.

**🔒 Campo Senha**

const SizedBox(height: 16),

TextField(

obscureText: true,

decoration: InputDecoration(

labelText: "Senha",

border: OutlineInputBorder(),

),

),

* obscureText: true → Oculta o texto digitado (para senhas).

**📌 7. Botão de Cadastro**

const SizedBox(height: 16),

ElevatedButton(

onPressed: () {

// Aqui pode adicionar a lógica de cadastro

},

child: const Text("Cadastrar"),

),

* **ElevatedButton**: Botão elevado para enviar o formulário.
* **onPressed**: Quando pressionado, poderá executar uma lógica de cadastro.