



Desenvolvimento
Mobile 1
Aula 03

Prof. Me Daniel Vieira

SENAI

Agenda

- 1- APP Flutter
- 2-Variáveis Dart
- 3 - Estruturas condicionais
- 4 -Estrutura de repetição
- 5 - Arrays
- 6- Exercícios

App Flutter

The screenshot shows a Flutter development environment with the following components:

- Code Editor:** On the left, the file `main.dart` is open in a dark-themed code editor. The code defines a simple counter application:

```
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       title: 'Counter App',
12       theme: ThemeData(
13         appBarTheme: AppBarTheme(
14           backgroundColor: Colors.red,
15         ), // AppBarTheme
16       ), // ThemeData
17       home: CounterScreen(),
18     );
19 }
```

- Run View:** In the center, a smartphone icon displays the running application. The app has a red header bar with the text "Primeiro aplicativo em Flutter". The main screen shows a counter value of 7, with three buttons below it: "Incrementar" (green), "Reset" (blue), and "Decrementar" (red).
- Sidebar:** On the right, there is a vertical sidebar with various icons for navigation and settings.

App Flutter

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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Counter App',
      theme: ThemeData(
        appBarTheme: AppBarTheme(
          backgroundColor: Colors.red,
        ),
        ),
      home: CounterScreen(),
    );
  }
}
```

App Flutter

```
class CounterScreen extends StatefulWidget {  
    @override  
    _CounterScreenState createState() => _CounterScreenState();  
  
}  
class _CounterScreenState extends State<CounterScreen> {  
    int _counter = 0;  
  
    void _incrementCounter() {  
        setState(() {  
            _counter++;  
        });  
    }  
}
```

App Flutter

```
void _resetCounter() {  
    setState(() {  
        _counter = 0;  
    });  
}  
  
void _decrementCounter() {  
    setState(() {  
        _counter = _counter -1;  
    });  
}
```

App Flutter

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text('Primeiro aplicativo em Flutter'),
    ),
    body: Center(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          Text(
            '_$_counter',
            style: TextStyle(fontSize: 30),
          ),
        ],
      ),
    ),
  );
}
```

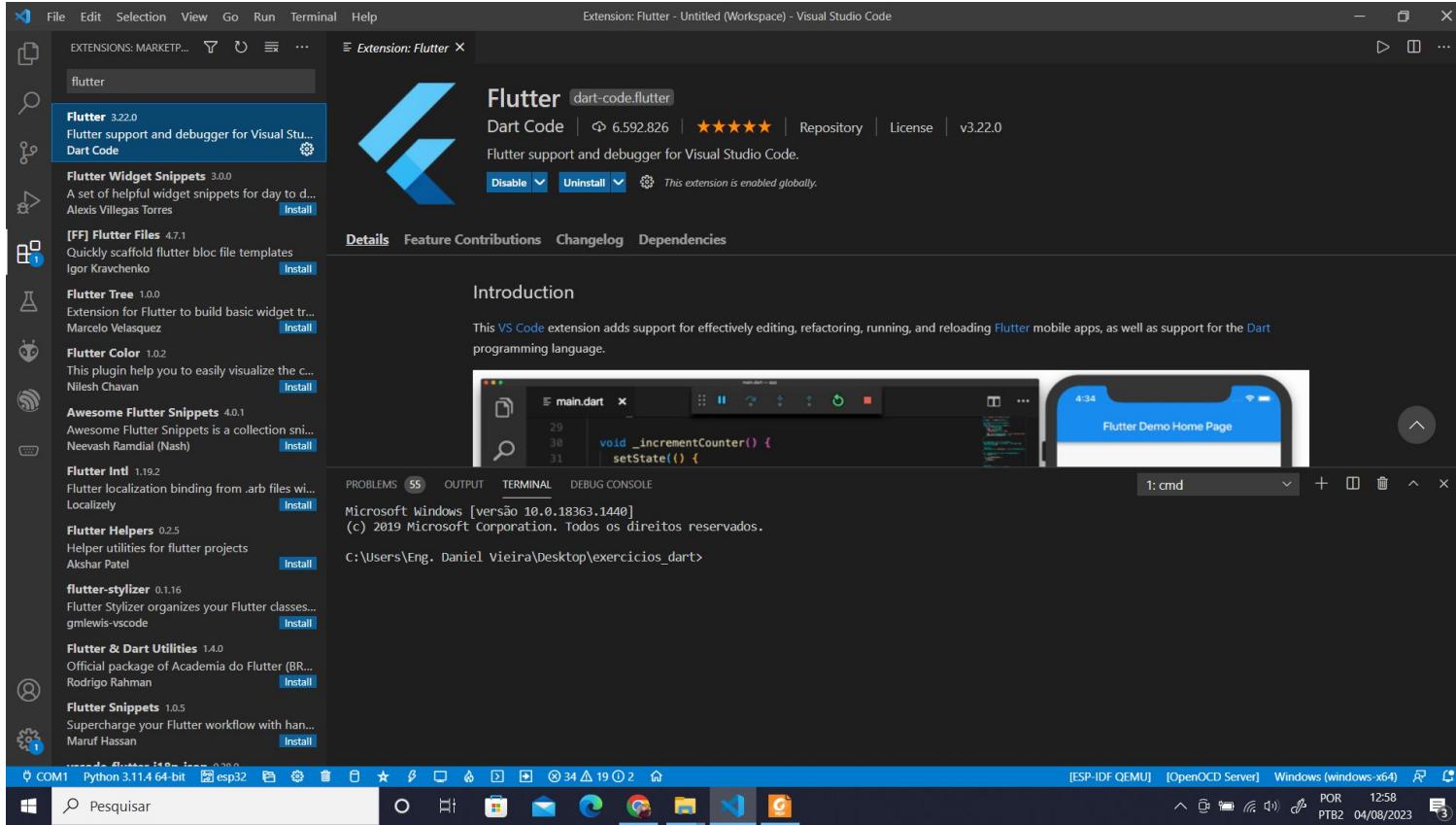
App Flutter

```
SizedBox(height: 30),  
    ElevatedButton(  
        onPressed: _incrementCounter,  
        child: Text('Incrementar'),  
        style: ButtonStyle(  
            backgroundColor: MaterialStateProperty.all<Color>(Colors.green),  
        ),  
    ),  
    SizedBox(height: 30),  
    ElevatedButton(  
        onPressed: _resetCounter,  
        child: Text('Reset'),  
        style: ButtonStyle(  
            backgroundColor: MaterialStateProperty.all<Color>(Colors.blue),  
        ),  
    ),
```

App Flutter

```
SizedBox(height: 30),  
    ElevatedButton(  
        onPressed: _decrementCounter,  
        child: Text('Decrementar'),  
        style: ButtonStyle(  
            backgroundColor: MaterialStateProperty.all<Color>(Colors.red),  
        ),  
    ),  
,  
],  
,  
,  
);  
}  
}
```

Configuração VS CODE



Tipos de variáveis Dart

```
import "dart:io";
void main()
{
    // Variável que armazena números inteiros
    int idade = 28;
    print("Idade: $idade");
    // Variável que armazena números decimais
    double raio = 10.25;
    print("Raio: $raio");
    // Variável que armazena caracteres e textos
    String nome = "Daniel";
    print("Olá $nome, seja bem vindo");
    //Variavel que armazena verdadeiro ou false
    bool ligado = true;
```

Tipos de variáveis Dart

// Variável que guarda uma lista genérica

```
List numerosgenericos = [10,"Daniel", true,20];  
print(numerosgenericos);
```

// Variável que guarda uma lista de números inteiros

```
List<int> numerosinteiros = [10,20,30,40];  
print(numerosinteiros);
```

// Variável que guarda um dicionário com chave e valor em formato texto

```
Map<String, String> nome_sobrenome = {"Daniel": "Vieira", "Senai": "Roberto Mange"};
```

// Variável sem tipo pré definido, seu tipo é igual ao tipo do primeiro valor que recebe

```
var sobrenome = "Vieira";  
print(sobrenome);
```

Exemplo código Dart

```
import 'dart:io';
void main()
{
    print("Digite seu nome");
    var nome = stdin.readLineSync();
    print("Seu nome é $nome");
}
```

```
import 'dart:io';

void main() {
    // Solicitando ao usuário que digite seu nome
    print("Digite seu nome:");
    String nome = stdin.readLineSync()!;
    // Solicitando ao usuário que digite sua idade
    print("Digite sua idade:");
    String idadeString = stdin.readLineSync()!;
    int idade = int.parse(idadeString);

    // Exibindo as informações digitadas pelo
    // usuário
    print("Seu nome é: $nome");
    print("Sua idade é: $idade");
}
```

Comandos Dart

void main(){}

- void -> Vazio, não retorna nada;
- () → recebe argumentos de fora;
- {} -> Nosso código fica aqui dentro
- print();
- função que imprime informações no console;
- () → recebe a informação;
- ; → necessário para terminar a frase.

Comandos Dart

Para acessar informações de uma variável utilizamos o \$variável
Necessário estar entre “ \$variavel”

Para receber uma informação digitada pelo usuário utilizamos a função
print("Digite seu nome");

```
var nome = stdin.readLineSync();
```

Para utilizar a função stdin.readLineSync() é necessário importar a biblioteca
dart:io

Essa biblioteca permite utilizar o comando acima

```
import 'dart:io';
```

Obrigado!

Prof. Me Daniel Vieira

Email: danielvieira2006@gmail.com

Linkedin: Daniel Vieira

Instagram: Prof daniel.vieira95

