

Tutorial Flutter: Fundamentos do Dart

- **Recursos necessários:**

- Visual Studio Code (Extensões Flutter)

- **Sinopse:**

Neste tutorial iremos explicar sobre os fundamentos da linguagem Dart e exemplos de suas utilizações.

- **1. O que é Dart ?**

Dart é uma linguagem de programação desenvolvida pelo Google, focada em eficiência, produtividade e escalabilidade. É muito usada para desenvolvimento web, mobile (especialmente com Flutter) e até mesmo aplicações de back-end. Também, fácil de aprender para quem já tem experiência com linguagens orientadas a objetos, como Java ou JavaScript, e oferece recursos modernos que tornam o desenvolvimento mais rápido e produtivo.

- ❖ **Váriaveis e tipos de dados:**

- Em Dart, você pode declarar variáveis com tipos explícitos ou permitir que o compilador deduza o tipo usando var.

- **Principais Tipos de Dados:**

- **int**: Números inteiros.
- **double**: Números de ponto flutuante.
- **String**: Cadeia de caracteres (texto).
- **bool**: Valores booleanos (true ou false).

EXEMPLO:

```
// Declaração de variáveis com tipo explícito
int idade = 25;
double altura = 1.75;
String nome = "João";
bool estalogado = true;

// Inferência de tipo com 'var'
var cidade = "São Paulo"; // Dart infere que é uma String
var temperatura = 26.5; // Dart infere que é um double
```

- ❖ **2. Controle de fluxo:**

- **2.1 – Estruturas Condicionais (if-else):**

As estruturas if-else permitem executar diferentes blocos de código com base em condições específicas;

EXEMPLO:

```
// Exemplo IF-ELSE
int nota = 85;

if (nota >= 90) {
    print("Excelente");
}
```

```

} else if (nota >= 70) {
    print("Bom");
} else {
    print("Precisa melhorar");
}

```

▪ 2.2 – Estruturas de Repetição:

▪ for e for-in:

Usado para repetir um bloco de código por um número fixo de vezes ou iterar sobre coleções.

EXEMPLO

// Exemplo de iteração FOR

```
List<String> frutas = ["Maçã", "Banana", "Laranja", "Manga"];
```

```

for (int i = 0; i < frutas.length; i++) {
    print("Fruta: ${frutas[i]}");
}

```

// Exemplo de iteração FOR-IN

```

List<String> cores = ["Vermelho", "Verde", "Azul"];
for (var cor in cores) {
    print(cor);
}

```

▪ While:

O loop while repete enquanto a condição é verdadeira

EXEMPLO

// Exemplo de iteração WHILE

```

int contador = 0;

while (contador < 3) {
    print("Contador: $contador");
    contador++;
}

```

❖ 3. Funções e Métodos:

- Funções são blocos de código reutilizáveis que podem realizar uma tarefa específica. Elas podem retornar um valor ou não.

EXEMPLO

```

int soma(int a, int b) {
    return a + b;
}

void saudacao(String nome, int? idade) {
    if (idade == null) {

```

```
idade = 18;
}
print("Olá, $nome! Você tem $idade anos.");
}
```

❖ 4. Classes e Objetos:

- Uma classe é um "molde" para criar objetos que encapsulam atributos (dados) e métodos (comportamentos).

EXEMPLO

```
class Pessoa {
  String nome;
  int idade;

  // Construtor
  Pessoa(this.nome, this.idade);

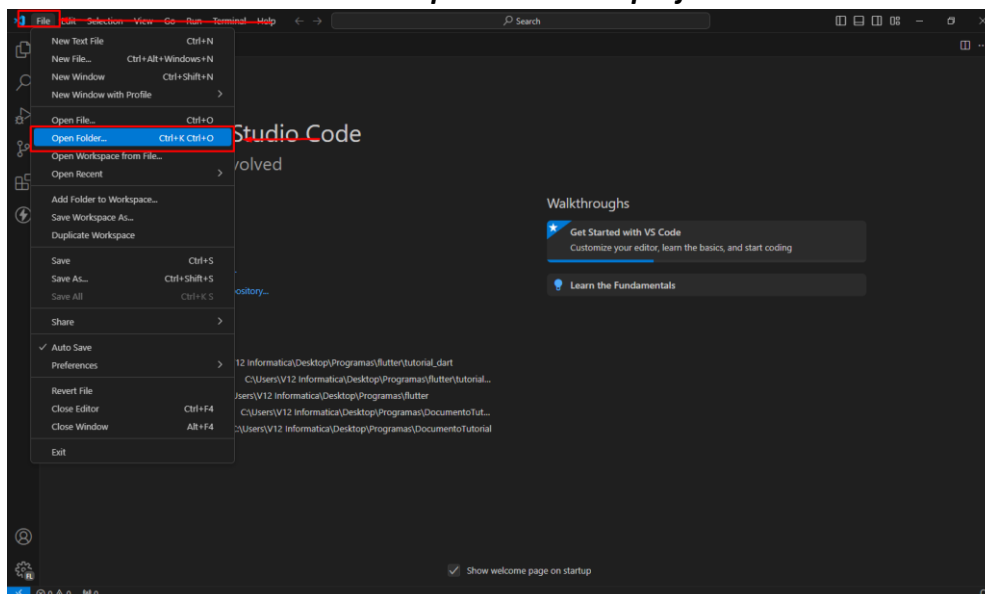
  // Método
  void apresentar() {
    print("Olá, meu nome é $nome e eu tenho $idade anos.");
  }
}
```

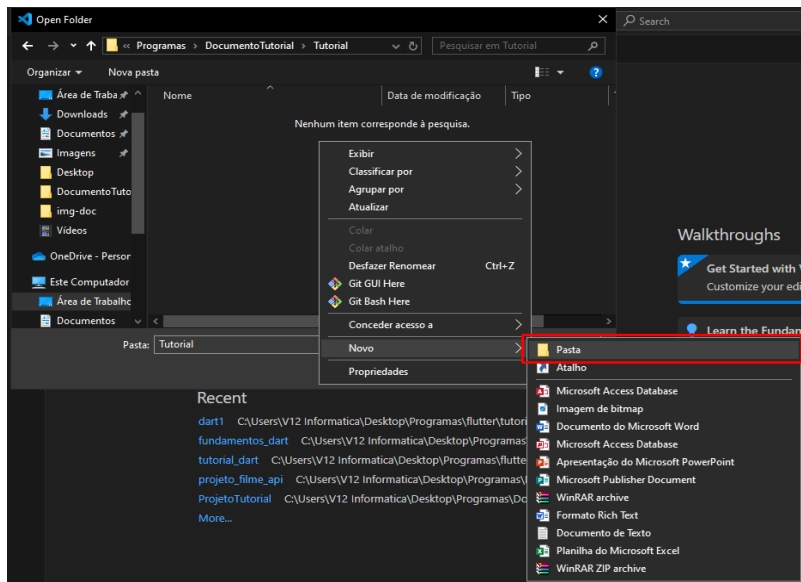
```
// Exemplo de P00
Pessoa pessoa1 = Pessoa("Ana", 30); // OBJETO 1 da classe modelo PESSOA
pessoa1.apresentar();

Pessoa pessoa2 = Pessoa("Carlos", 25); // OBJETO 2 da classe modelo PESSOA
pessoa2.apresentar();
```

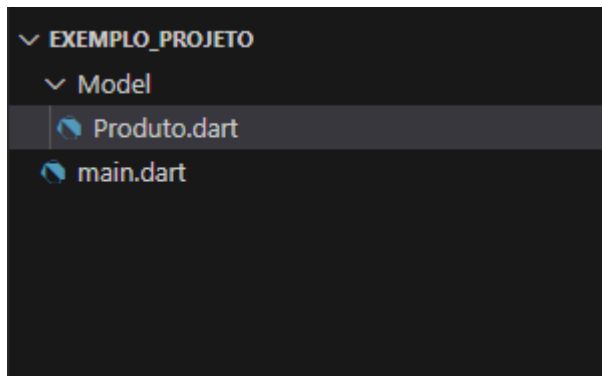
Agora vamos realizar um projeto completo utilizando os principais fundamentos de Dart:

1. Crie um novo diretório para realizar o projeto:





Após criar um novo diretório, abra-o e crie os seguintes arquivos:



Aqui temos um arquivo `main.dart`, na raiz do projeto, que é responsável por executar a aplicação, e um diretório `Model` com um arquivo `Produto.dart`, esse arquivo será no classe `Modelo`.

Abra o arquivo Produto.dart e insira o código:

```
class Produto {
  String nome;
  double preco;

  Produto(this.nome, this.preco);

  void mostrarProduto() {
    print("Produto: $nome, Preço: R\$ $preco");
  }
}
```

Depois siga até o main.dart e coloque o código:

```
import 'dart:io'; // Necessário para leitura de entrada no console

import 'Model/produto.dart';

void main() {
  List<Produto> produtos = [];
```

```

bool continuar = true;

// Laço para cadastrar produtos
while (continuar) {
    print("Digite o nome do produto:");
    String nomeProduto = stdin.readLineSync()!;

    print("Digite o preço do produto:");
    double precoProduto = double.parse(stdin.readLineSync()!);

    Produto novoProduto = Produto(nomeProduto, precoProduto);
    produtos.add(novoProduto);

    print("Deseja adicionar outro produto? (s/n)");
    String resposta = stdin.readLineSync()!;

    if (resposta.toLowerCase() != 's') {
        continuar = false;
    }
}

// Exibindo todos os produtos cadastrados
print("\nProdutos cadastrados:");
for (var produto in produtos) {
    produto.mostrarProduto();
}

// Pesquisar um produto pelo nome
print("\nDigite o nome do produto que deseja buscar:");
String nomeBusca = stdin.readLineSync()!;

bool produtoEncontrado = false;
for (var produto in produtos) {
    if (produto.nome.toLowerCase() == nomeBusca.toLowerCase()) {
        print("Produto encontrado:");
        produto.mostrarProduto();
        produtoEncontrado = true;
        break;
    }
}

if (!produtoEncontrado) {
    print("Produto não encontrado.");
}
}

```

Dessa forma criamos uma simples aplicação para cadastrar, listar e buscar produtos em uma lista utilizando fundamentos básicos e principais do DART.s