



Desenvolvimento  
Mobile 1  
Aula 08

Prof. Me Daniel Vieira

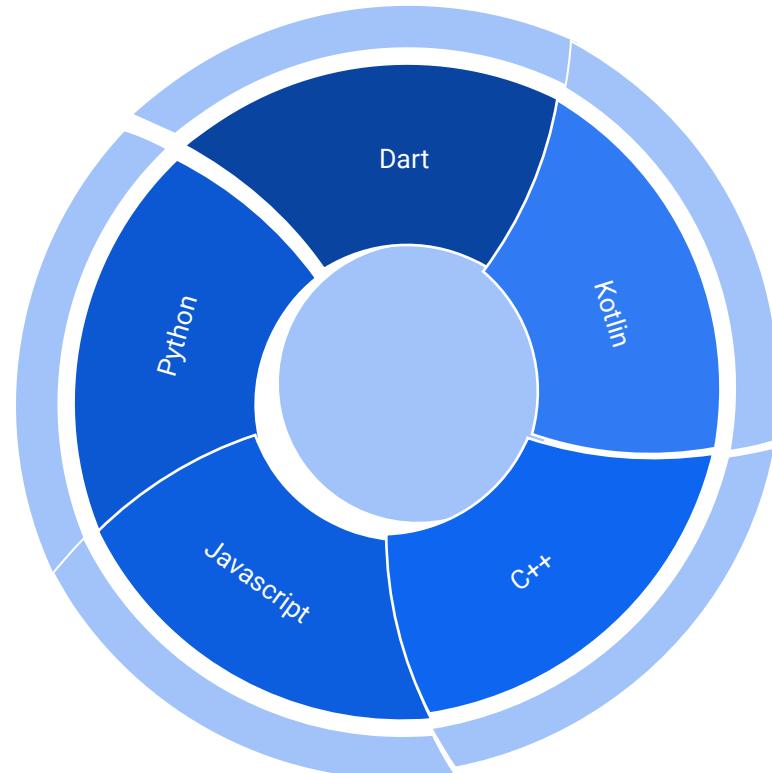
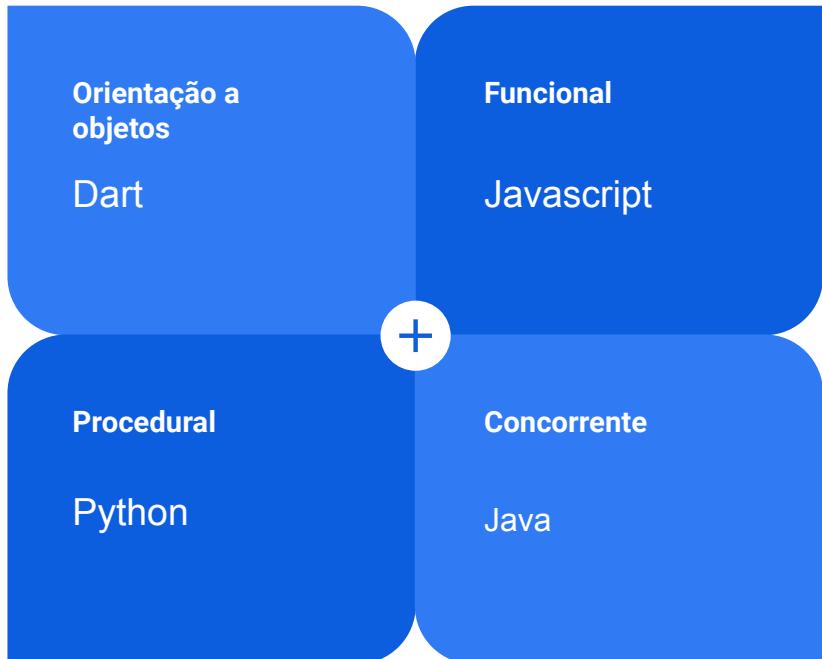
**SENAI**

# Agenda

- 1- Paradigmas de programação
- 2- Orientação a objeto
- 3-Construtor
- 4 -Exercícios

# Paradigmas de programação

O que é um paradigma ?



# Paradigmas de programação

## Paradigma imperativo

No paradigma imperativo, como o nome já revela, o desenvolvedor cria uma instrução para que a máquina processe as execuções de uma **determinada** maneira.

Dentro dessa categoria existem:

### Orientado ao Objeto

Esse paradigma é um dos mais aplicados por conta das vantagens que ele traz para o processo, como a **modularidade do código** e a função de criar relações entre problemas reais dentro dos termos de **código**.

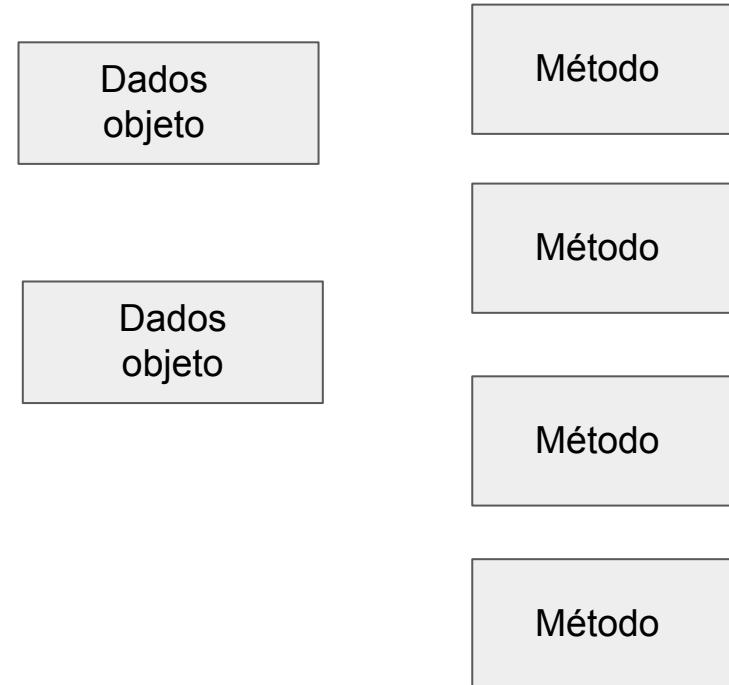
### Procedural

Perfeita para programação geral; consiste em uma lista de instruções para o computador executar as tarefas, uma de cada vez.

A maioria das linguagens de programação que um desenvolvedor aprende na faculdade são procedurais, como C, C++ e Java, por exemplo

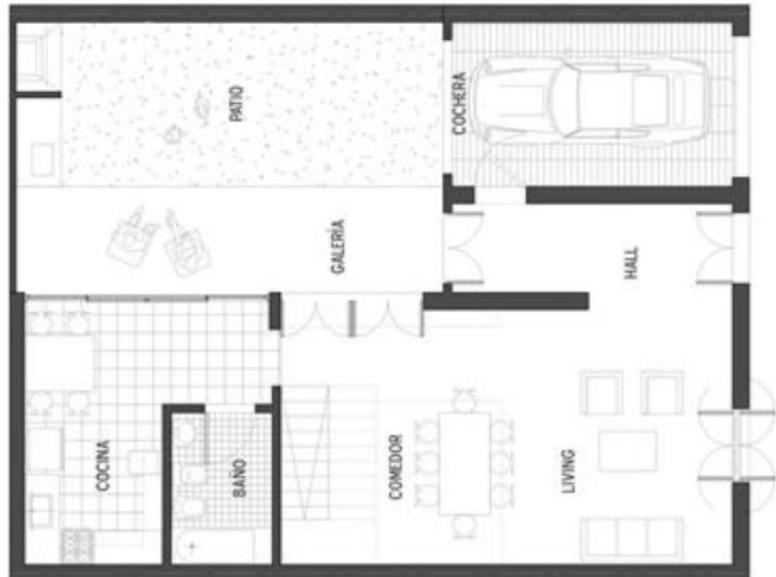
# Paradigmas de programação

## Programação estruturada



# Classe - Orientação a objeto

Classe - Planta



Objeto - Casa



# Classe - Orientação a objeto

## Atributos

Casa (objeto)



Cor

Tipo de janela

Vagas na garagem

...

# Classe - Orientação a objeto

## Métodos

Casa (objeto)



Abrir portão  
Abrir janelas  
Ligar luzes  
...

# Classe - Orientação a objeto

```
class Casa
{
    // Atributos definem características
    // Métodos definem o que os objetos podem fazer
    String? cor;

}
void main()
{
    String nome = "Daniel";
    Casa minhaCasa = new Casa(); // instancia a classe em um objeto new é opcional no dart
    minhaCasa.cor= "Blue";
    print(minhaCasa.cor);
}
```

# Classe - Orientação a objeto

```
class Casa
{
  // Atributos definem características
  // Métodos definem o que os objetos podem fazer
  String? cor;
  // Método
  void abrirJanela(int qtdeJanelas)
  {
    print("Abrir Janela, qtde janelas $qtdeJanelas");
  }
  void abrirPorta()
  {
    print("Abrir porta da casa $cor");
  }
  void abrirCasa()
  {
    this.abrirJanela(2);
    this.abrirPorta();
  }
}
```

```
void main()
{
  String nome = "Daniel";
  Casa minhaCasa = new Casa(); // instancia a classe em um objeto new é opcional no dart
  Casa minhaCasa2 = new Casa();
  minhaCasa.cor= "Blue";
  minhaCasa2.cor = "Vermelho";
  //minhaCasa.abrirJanela(2);
  // print(minhaCasa.cor);
  // minhaCasa2.abrirPorta();
  minhaCasa2.abrirCasa();
  minhaCasa.abrirCasa();
}
```

# Classe - Orientação a objeto

Exemplo :

Criar uma classe usuario com dois atributos:  
email e senha e criar a autenticação do usuário

```
class Usuario
{
    String ? usuario;
    String? senha;
    void autentica()
    {
        var usuario = "Senai";
        var senha = "senai@2023";
        if(this.usuario == usuario && this.senha == senha)
        {
            print("Login correto");
        }
        else{
            print("Erro, tente novamente");
        }
    }
}
```

```
void main()
{
    Usuario usuario= Usuario();
    usuario.usuario= "Daniel";
    usuario.senha= "senai@2023";
    usuario.autentica();
}
```

# Exercícios

- 1) Criar uma função que receba as informações de um usuário digitado pelo teclado: Nome, Curso, Idade
- 2) Criar uma função que calcule a área de um triângulo a partir de dados digitados pelo usuário.  $A = (b * h)/2$  e retorne esse valor
- 3) Criar uma função que calcule o salário do usuário a partir dos valores digitados pelo teclado considerando um desconto de 10 % de impostos e bonificação de 20% em cima do salário
- 4) Criar um programa de transações bancárias
  - 1 - Saque
  - 2- Pix
  - 3- Empréstimos
  - 4 - TransferênciasPara cada opção do Menu perguntar para o usuário o valor para realizar a transação e passar o valor por função

# Exercícios

5 - Criar um programa para realizar a conversão de moedas conforme o valor digitado pelo usuário em R\$ e a escolha para qual moeda o usuário quer converter o valor: Euro, dólar, francos suíços

6- Criar uma classe chamada carrinho de compras com os seguintes atributos:  
Itens, Quantidades

Métodos: adicionar itens - Adiciona um item ao carrinho

Remover item - remove um item do carrinho

Calcular o total() - Retorna o valor total do carrinho

7 - Criar uma classe chamada carro com os seguintes atributos: marca, modelo, ano, motor ligado

Métodos da classe:

ligar\_motor(): Um método que liga o motor do carro e atualiza o atributo motor\_ligado para True.

desligar\_motor(): Um método que desliga o motor do carro e atualiza o atributo motor\_ligado para False.

status\_motor(): Um método que retorna uma mensagem indicando se o motor está ligado ou desligado. Teste sua classe criando um objeto Carro, ligando e desligando o motor, e verificando o status do motor.

# Exercícios

Link forms

<https://docs.google.com/forms/d/1IICWTjrrz28Ax1dE30OWfgyauxxOXpzYGPjs7sPDm8k/edit>

# Obrigado!

Prof. Me Daniel Vieira

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

Linkedin: Daniel Vieira

Instagram: Prof daniel.vieira95

