

**INSTITUTO
FEDERAL**
Paraíba

Campus
Cajazeiras

PROGRAMAÇÃO ORIENTADA A OBJETOS

INTRODUÇÃO

PROF. DIEGO PESSOA

 DIEGO.PESSOA@IFPB.EDU.BR

 [@DIEGOEP](https://github.com/DIEGOEP)



**Técnico em
Informática
Integrado**

APRESENTAÇÃO DA DISCIPLINA

EMENTA

- ▶ Conceitos do paradigma da programação orientada a objetos
- ▶ Abstração
- ▶ Introdução a classes e objetos
- ▶ Agregação e Composição de objetos
- ▶ Encapsulamento, herança, polimorfismo
- ▶ Tratamento de Exceções
- ▶ Projeto orientado a objetos
- ▶ Linguagem de programação orientada a objetos
- ▶ Teste de software

OBJETIVO GERAL

- ▶ Tornar o aluno apto a **compreender os conceitos** de programação orientada a objetos e **projetar, desenvolver e testar** softwares utilizando o paradigma e uma linguagem de programação orientado a objetos.

OBJETIVOS ESPECÍFICOS

- ▶ Compreender os **conceitos básicos** do paradigma de programação orientado a objetos
- ▶ Utilizar **classes, objetos e interação** entre objetos.
- ▶ Entender conceitos como **abstração, encapsulamento, herança e polimorfismo**.
- ▶ Realizar tratamento de **exceções**.
- ▶ **Projetar** software orientado a objetos
- ▶ **Desenvolver** software utilizando uma linguagem de programação orientada a objetos.
- ▶ **Testar** softwares desenvolvidos no paradigma de orientação a objetos.

CONTEÚDO PROGRAMÁTICO

- ▶ Conceitos do paradigma da programação orientada a objetos.
- ▶ Introdução a classes e objetos.
- ▶ Atributos, métodos e interação entre objetos.
- ▶ Sintaxe de linguagem de programação orientada a objetos
- ▶ O que é abstração em orientação a objetos
- ▶ Agregação e Composição de objetos.
- ▶ Encapsulamento
- ▶ Herança e Polimorfismo.
- ▶ Tratamento de Exceções.
- ▶ Análise e Projeto orientado a objetos.
- ▶ Introdução ao teste de software.
- ▶ Teste de software aplicado ao paradigma de orientação a objetos.

APRESENTAÇÃO DA DISCIPLINA



INTRODUÇÃO A POO E CONCEITOS BÁSICOS

O PARADIGMA DE PROGRAMAÇÃO ORIENTADA A OBJETOS

- ▶ Paradigmas são formas de enxergar o mundo (os problemas, a vida, um código de um programa)
- ▶ O paradigma de **Orientação a Objetos** pode ser encarado como uma forma de pensar o seu projeto, desde a arquitetura até a implementação.
- ▶ Outros exemplos de paradigmas de programação são a **programação imperativa** e a **orientada a procedimentos**

REVISANDO O PARADIGMA ORIENTADO A PROCEDIMENTOS

- ▶ Baseado em chamada de funções ou sub-rotinas que operam sobre elas
- ▶ O fluxo de dados concentra todas as variáveis
- ▶ Uma função toma um conjunto de variáveis como argumento e retorna o resultado para o fluxo de dados, para ser usado por outra função ou simplesmente ser exibido para o usuário



PARADIGMA ORIENTADO A OBJETOS

- ▶ Estamos rodeados por objetos: mesa, carro, livro, pessoa, etc; e Os objetos do mundo real têm duas características em comum:
 - ▶ **Estado = propriedades** (nome, peso, altura, cor, etc.);
 - ▶ **Comportamento = ações** (andar, falar, calcular, etc.).

BANCO

PESSOA

CONTA

PARADIGMA ORIENTADO A OBJETOS – DEFINIÇÕES

- ▶ Paradigma para desenvolvimento de software que baseia-se na **utilização de componentes individuais (objetos)** que colaboram para construir sistemas mais complexos.
 - ▶ A colaboração entre os objetos é feita através do envio de mensagens.
 - ▶ Um paradigma é um conjunto de regras que estabelecem fronteiras e descrevem como resolver problemas dentro desta fronteira.

VANTAGENS

- ▶ Facilita a reutilização de código;
- ▶ Os modelos refletem o mundo real de maneira mais aproximada:
 - ▶ Descrevem de maneira mais precisa os dados;
 - ▶ Mais fáceis de entender e manter.
- ▶ Pequenas mudanças nos requisitos não implicam em grandes alterações no sistema em desenvolvimento.

OS QUATRO PILARES

ABSTRAÇÃO

ENCAPSULAMENTO

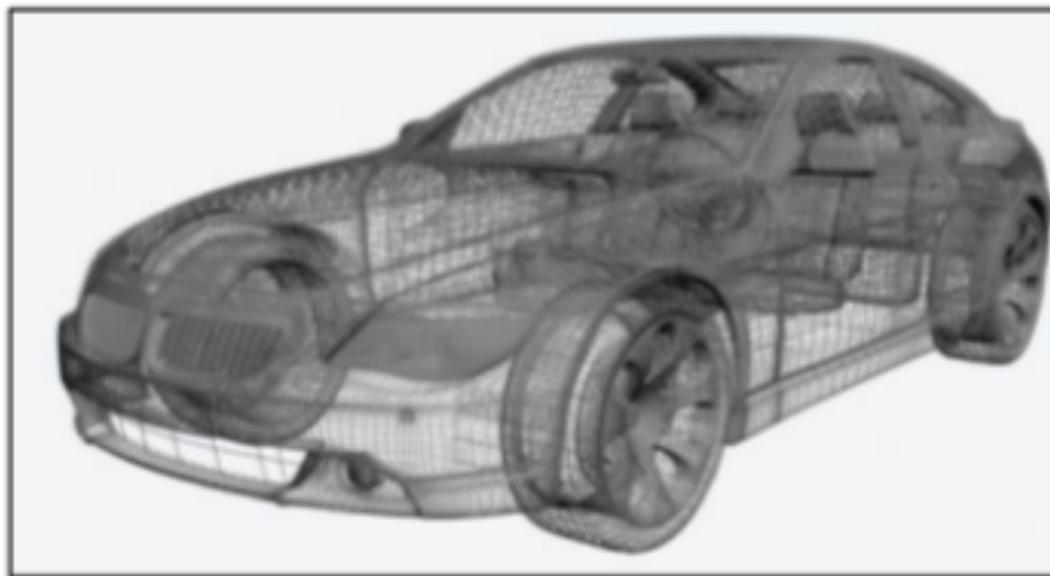
HERANÇA

POLIMORFISMO

ABSTRAÇÃO

CLASSES

- ▶ A estrutura fundamental para definir novos objetos;
- ▶ Uma classe é definida em código-fonte.



Classe



Objeto

CLASSES EM PYTHON

▶ Estrutura:

class nome_da_classe:

atributos

métodos

▶ Exemplo:

```
class Conta:  
    numero = 000000  
    saldo = 0.0
```

INSTÂNCIA

- ▶ Uma instância é um objeto criado com base em uma classe definida;
- ▶ Classe é apenas uma estrutura, que especifica objetos, mas que não pode ser utilizada diretamente;
- ▶ Instância representa o objeto concretizado a partir de uma classe;
- ▶ Uma instância possui um ciclo de vida:



INSTÂNCIA EM PYTHON

▶ Estrutura

variável = **Classe()**

▶ Exemplo

```
if __name__ == '__main__':  
    conta = Conta()  
    conta.saldo = 20  
    conta.numero = "13131-2"  
    print(conta.saldo)  
    print(conta.numero)
```

MÉTODOS

- ▶ Representam os comportamentos de uma classe;
- ▶ Permitem acesso a atributos, tanto para recuperar os valores, como para alterá-los caso necessário;
- ▶ Podem retornar ou não algum valor; e
- ▶ Podem possuir ou não parâmetros.

MÉTODOS

ESTRUTURA:

```
def nome_do_metodo(self, parametros)
```

- ▶ Importante: o parâmetro **self** é obrigatório.

EXEMPLO:

```
class Conta:
    numero = "00000-0"
    saldo = 0.0

    def deposito(self, valor):
        self.saldo += valor

    def saque(self, valor):
        if (self.saldo > 0):
            self.saldo -= valor
        else:
            print("Saldo insuficiente")

if __name__ == '__main__':
    conta = Conta()
    conta.saldo = 20
    conta.numero = "13131-2"
    print(conta.saldo)
    print(conta.numero)
```

MÉTODO CONSTRUTOR

- ▶ Determina que ações devem ser executadas quando da criação de um objeto; e
- ▶ Pode possuir ou não parâmetros.

▶ Estrutura:

Def `__init__(self, parametros)`

▶ Exemplo:

```
class Conta:
    numero = "00000-0"
    saldo = 0.0

    def __init__(self, numero, saldoInicial):
        self.numero = numero
        self.saldo = saldoInicial

conta = Conta("12345-1", 0)
print(conta.numero)
print(conta.saldo)
```

EXERCÍCIO – TAREFA DE CASA

- ▶ Classe Pessoa: Crie uma classe que modele uma pessoa:
 - ▶ Atributos: nome, idade, peso e altura
 - ▶ Métodos: Envelhecer, engordar, emagrecer, crescer.
 - ▶ Obs: Por padrão, a cada ano que nossa pessoa envelhece, sendo a idade dela menor que 21 anos, ela deve crescer 0,5 cm
 - ▶ Pode possuir ou não parâmetros.