



## **UNIVERSIDADE ESTÁCIO DE SÁ**

1855 - Polo Gleba A - Santana de Parnaíba - SP  
Desenvolvimento Full Stack  
Iniciando o Caminho Pelo Java  
2004.3

### **Relatório Discente de Acompanhamento**

#### **Integrantes da Prática:**

. Maurício Buglia Santos

[https://github.com/Kabal1989/Missao\\_Pratica\\_N1](https://github.com/Kabal1989/Missao_Pratica_N1)

Link Repositório no GitHub

### **Desenvolvimento e Persistência de Dados em um Sistema de Cadastro de Pessoas em Java**

O objetivo desta prática é aplicar conceitos de Programação Orientada a Objetos (POO) no desenvolvimento de um sistema de cadastro que permite a persistência de dados utilizando a linguagem Java. O projeto envolve a criação de classes, uso de herança, implementação de repositórios para manipulação de dados e armazenamento em arquivos binários.

## CÓDIGO FONTE:

```
package model;

import java.io.Serializable;
/**
 *
 * @author kabal
 */
public class Pessoa implements Serializable {
    private int id;
    private String nome;

    public Pessoa() {
    }

    public Pessoa(int id, String nome) {
        this.id = id;
        this.nome = nome;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public void exibir() {
        System.out.println("ID: " + id + ", Nome: " + nome);
    }
}
```

## Resultados da execução:

Dados de Pessoa Física Armazenados.  
Dados de Pessoa Física Recuperados.  
ID: 1, Nome: Ana  
CPF: 111.111.111-11, Idade: 25  
ID: 2, Nome: Carlos  
CPF: 222.222.222-22, Idade: 52

Dados de Pessoa Jurídica Armazenados.  
Dados de Pessoa Jurídica Recuperados.  
ID: 3, Nome: XPTO Sales  
CNPJ: 33.333.333/3333-33  
ID: 4, Nome: XPTO Solutions  
CNPJ: 44.444.444/4444-44

## **Análise e Conclusão:**

### **1. Quais as vantagens e desvantagens do uso de herança?**

Vantagens:

- **Reutilização de código:** A herança permite que classes compartilhem funcionalidades, evitando duplicação de código.
- **Organização:** Facilita a organização de hierarquias de classes, tornando o código mais compreensível e estruturado.

Desvantagens:

- **Acoplamento:** A herança cria uma relação de dependência entre classes, o que pode tornar a manutenção mais difícil se houver mudanças nas classes base.
- **Fragilidade:** Mudanças na classe base podem ter efeitos colaterais em todas as subclasses, causando problemas inesperados.

### **2. Por que a interface `Serializable` é necessária ao efetuar persistência em arquivos binários**

A interface `Serializable` é necessária para que os objetos possam ser convertidos em um formato de byte stream, permitindo que sejam salvos em arquivos binários. Sem essa interface, a JVM não saberia como transformar o estado de um objeto em uma sequência de bytes e como reconstruí-lo posteriormente.

### **3. Como o paradigma funcional é utilizado pela API stream no Java?**

O paradigma funcional é utilizado pela API Stream do Java através de operações como `map`, `filter` e `reduce`, que permitem transformar e processar coleções de dados de maneira declarativa. Essas operações são aplicadas de forma imutável e geralmente utilizam expressões lambda ou referências de métodos, características do estilo de programação funcional.

### **4. Quando trabalhamos com Java, qual padrão de desenvolvimento é adotado na persistência de dados em arquivos?**

O padrão de desenvolvimento mais comum na persistência de dados em arquivos no Java é o uso de Data Access Objects (DAO). O DAO separa a lógica de persistência de dados do restante da aplicação, permitindo que as operações de armazenamento e recuperação de dados sejam encapsuladas em classes específicas, como repositórios.

## Cadastro de Pessoas Físicas e Jurídicas em Java em Modo Texto

O objetivo foi desenvolver aplicação em Java que permita o cadastro, consulta, alteração e exclusão de pessoas físicas e jurídicas, utilizando classes de repositório para armazenar e recuperar os dados, e implementando um menu interativo em modo texto.

### Análise e Conclusão

#### a) O que são elementos estáticos e qual o motivo para o método main adotar esse modificador?

- Elementos estáticos são aqueles que pertencem à classe e não a instâncias individuais da classe. Isso significa que todos os objetos da classe compartilham o mesmo valor da variável ou podem acessar o mesmo método.
- O método main é declarado como static porque ele é o ponto de entrada do programa e precisa ser acessível sem a necessidade de criar uma instância da classe. Ele precisa ser executado pela JVM diretamente.

#### b) Para que serve a classe Scanner?

- A classe Scanner é utilizada para ler dados de entrada do usuário. No contexto do seu projeto, ela é usada para capturar as escolhas do usuário no menu interativo e os dados que serão armazenados no sistema, como nome, CPF, CNPJ, etc.

#### c) Como o uso de classes de repositório impactou na organização do código?

- Organização: O uso de classes de repositório (como PessoaFisicaRepo e PessoaJuridicaRepo) permitiu separar a lógica de manipulação de dados da lógica de interação com o usuário. Isso melhora a manutenibilidade do código, tornando-o mais modular e fácil de entender.
- Reutilização: Essas classes de repositório podem ser reutilizadas em diferentes partes do programa ou mesmo em outros projetos que precisem de funcionalidade similar.
- Isolamento de responsabilidades: Cada classe tem uma responsabilidade clara e definida, o que segue o princípio de Responsabilidade Única (SRP) do SOLID.

### 6. Conclusão Geral

- Exemplo: "A prática desenvolvida mostrou a importância de uma boa organização de código e da utilização de padrões de projeto como a separação entre lógica de negócios e persistência de dados. Através do uso de classes de repositório, o código ficou mais modular, o que facilita a manutenção e expansão da aplicação."

[https://github.com/Kabal1989/Missao\\_Pratica\\_N1](https://github.com/Kabal1989/Missao_Pratica_N1)

Link Repositório no GITHUB