

ESTÁCIO DE SÁ

Campus: Polo Centro - Araraquara/SP

Curso: Desenvolvimento Full Stack

Disciplina: Iniciando o caminho pelo Java

Turma: 9002 - **Semestre Letivo:** 1.2025

Alexandra Karem de Almeida Boiago

Matricula: 20230202545467

Repositório GIT:

<https://github.com/Alexandra-Almeida-DV/CadastroPOO>



1. SISTEMA DE CADASTRO EM MODO TEXTO COM PERSISTÊNCIA DE DADOS EM ARQUIVOS BINÁRIOS

2. OBJETIVO DA PRÁTICA

Desenvolver um sistema de cadastro em Java, utilizando conceitos de Programação Orientada a Objetos (POO), herança, serialização, e manipulação de arquivos binários. O sistema permite incluir, alterar, excluir, buscar e persistir dados de pessoas físicas e jurídicas por meio de um menu interativo em modo texto.

3. CÓDIGOS DESENVOLVIDOS

3.1 Entidades

- Pessoa.java (Classe base)
 - PessoaFisica.java
 - PessoaJuridica.java

3.2 Gerenciadores

- PessoaFisicaRepo.java

- PessoaJuridicaRepo.java

3.3 Classe Principal

- CadastroPOO.java

4. RESULTADOS DA EXECUÇÃO

```
run:
Dados de pessoas físicas persistidos com sucesso!
Dados de pessoas físicas recuperados com sucesso!

Pessoas Físicas Cadastradas:
ID: 1
Nome: João Silva
CPF: 111.111.111-11
Idade: 30
-----
ID: 2
Nome: Maria Oliveira
CPF: 222.222.222-22
Idade: 25
-----

Dados de pessoas jurídicas persistidos com sucesso!
Dados de pessoas jurídicas recuperados com sucesso!

Pessoas Jurídicas Cadastradas:
ID: 3
Nome: Empresa ABC Ltda
CNPJ: 11.111.111/0001-11
-----
ID: 4
Nome: Comércio XYZ S/A
CNPJ: 22.222.222/0001-22
-----
BUILD SUCCESSFUL (total time: 0 seconds)
|
```

5. ANÁLISE E CONCLUSÃO

5.1 Quais as vantagens e desvantagens do uso de herança?

Vantagens:

Reutilização de código, evitando duplicidade.

Facilita a manutenção e evolução do software.

Melhora a organização e estruturação do projeto.

Desvantagens:

Pode dificultar o entendimento se houver muitas camadas.

O acoplamento entre as classes pode aumentar.

5.2 POR QUE A INTERFACE SERIALIZABLE É NECESSÁRIA AO EFETUAR PERSISTÊNCIA EM ARQUIVOS BINÁRIOS?

A interface `Serializable` é fundamental para permitir que os objetos Java sejam convertidos em uma sequência de bytes. Isso é necessário para gravar os dados em arquivos binários e, posteriormente, poder recriar os objetos originais ao ler esses dados.

5.3 COMO O PARADIGMA FUNCIONAL É UTILIZADO PELA API STREAM NO JAVA?

A API Stream do Java adota o paradigma funcional ao permitir que coleções sejam manipuladas de forma declarativa, com uso de funções puras, lambda expressions, métodos encadeados e imutable data. Isso facilita operações como filtragem, mapeamento e redução de dados com código mais conciso e legível.

5.4 QUANDO TRABALHAMOS COM JAVA, QUAL PADRÃO DE DESENVOLVIMENTO É ADOTADO NA PERSISTÊNCIA DE DADOS EM ARQUIVOS?

Geralmente, é adotado o padrão DAO (Data Access Object) ou repositório, que encapsula a lógica de persistência. Isso separa a lógica de negócio da lógica de acesso a dados, seguindo o princípio da responsabilidade única e facilitando a manutenção e testes.

6. Considerações Finais

O projeto demonstrou:

Organização com pacotes e classes bem definidas.

Persistência eficiente usando serialização.

Habilidades em POO (herança, polimorfismo, encapsulamento).

