

Aluno: Altair Silva Filho

Matrícula: 202403254409

Unidade: Palhoça – SC

Curso: Desenvolvimento Full Stack

Disciplina: Iniciando o caminho pelo Java

Período: 2025.1

## ***Relatório da Missão Prática – Nível 1 – Mundo 3***

### **Iniciando o caminho pelo Java**

#### **2º Procedimento | Criação do Cadastro em Modo Texto**

##### **Objetivo**

Implementação de um cadastro de clientes em modo texto, com persistência em arquivos, baseado na tecnologia Java. Nesse procedimento, houve alteração do método “main” a fim de alcançar o pleno objetivo do projeto

##### **Códigos do Projeto**

###### **Pasta Model**

*Pessoa.java*

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

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

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

```

### *PessoaFisica.java*

```

package model;
/**
 * @author Altair
 */
public class PessoaFisica extends Pessoa {
    private String CPF;
    private int idade;

    public PessoaFisica(int id, String nome, String CPF, int idade) {
        super(id, nome);
        this.CPF = CPF;
        this.idade = idade;
    }

    @Override
    public void exibir() {
        super.exibir();
        System.out.println("CPF: " + CPF);
        System.out.println("Idade: " + idade);
    }

    public String getCpf() {
        return CPF;
    }

    public void setCpf(String CPF) {
        this.CPF = CPF;
    }

    public int getIdade() {
        return idade;
    }
}

```

```

    public void setIdade(int idade) {
        this.idade = idade;
    }
}

```

#### *PessoaJuridica.java*

```

package model;
/**
 * @author Altair
 */
public class PessoaJuridica extends Pessoa {
    private String CNPJ;

    public PessoaJuridica(int id, String nome, String CNPJ) {
        super(id, nome);
        this.CNPJ = CNPJ;
    }

    @Override
    public void exibir() {
        super.exibir();
        System.out.println("CNPJ: " + CNPJ);
    }

    public String getCnpj() {
        return CNPJ;
    }

    public void setCnpj(String CNPJ) {
        this.CNPJ = CNPJ;
    }
}

```

#### *PessoaFisicaRepo.java*

```

package model;

import java.io.*;
import java.util.ArrayList;
import java.util.List;

/**
 * @author Altair
 */
public class PessoaFisicaRepo {
    private ArrayList<PessoaFisica> lista = new ArrayList<>();

    public void inserir(PessoaFisica pessoa) {
        lista.add(pessoa);
    }
}

```

```

public void alterar(PessoaFisica pessoa) {
    lista.stream()
        .filter(p -> p.getID() == pessoa.getID())
        .findFirst()
        .ifPresent(original -> {
            lista.set(lista.indexOf(original), pessoa);
        });
}

public void excluir(int id) {
    lista.removeIf(p -> p.getID() == id);
}

public PessoaFisica obter(int id) {
    return lista.stream()
        .filter(p -> p.getID() == id)
        .findFirst()
        .orElse(null);
}

public List<PessoaFisica> obterTodos() {
    return lista.stream().toList();
}

public void persistir (String nomeArquivo) throws IOException {
    try (ObjectOutputStream out = new ObjectOutputStream(new FileOutputStream(nomeArquivo))) {
        out.writeObject(lista);
    }
}

public void recuperar(String nomeArquivo) throws IOException,
    ClassNotFoundException {
    try (ObjectInputStream in = new ObjectInputStream(new FileInputStream(nomeArquivo))) {
        lista = (ArrayList<PessoaFisica>) in.readObject();
    }
}
}

```

*PessoaJuridicaRepo.java*

```

package model;

import java.util.ArrayList;
import java.io.*;
import java.util.List;

/**
 * @author Altair
 */
public class PessoaJuridicaRepo {
    private ArrayList<PessoaJuridica> lista = new ArrayList<>();
}

```

```

public void inserir(PessoaJuridica pessoa) {
    lista.add(pessoa);
}

public void alterar(PessoaJuridica pessoa) {
    lista.stream()
        .filter(p -> p.getID() == pessoa.getID())
        .findFirst()
        .ifPresent(original -> {
            lista.set(lista.indexOf(original), pessoa);
        });
}

public void excluir(int id) {
    lista.removeIf(p -> p.getID() == id);
}

public PessoaJuridica obter(int id) {
    return lista.stream()
        .filter(p -> p.getID() == id)
        .findFirst()
        .orElse(null);
}

public List<PessoaJuridica> obterTodos() {
    return lista.stream().toList();
}

public void persistir (String nomeArquivo) throws IOException {
    try (ObjectOutputStream out = new ObjectOutputStream(new FileOutputStream(nomeArquivo))) {
        out.writeObject(lista);
    }
}

public void recuperar(String nomeArquivo) throws IOException,
    ClassNotFoundException {
    try (ObjectInputStream in = new ObjectInputStream(new FileInputStream(nomeArquivo))) {
        lista = (ArrayList<PessoaJuridica>) in.readObject();
    }
}
}

```

## Main Class

*CadastroPOO.java*

```

import java.io.IOException;
import model.PessoaFisica;
import model.PessoaFisicaRepo;
import model.PessoaJuridica;

```

```

import model.PessoaJuridicaRepo;
import java.util.Scanner;

/**
 * @author Altair
 */
public class CadastroPOO {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        PessoaFisicaRepo repoPF = new PessoaFisicaRepo();
        PessoaJuridicaRepo repoPJ = new PessoaJuridicaRepo();

        int opcao;

        do {
            System.out.println("=====");
            System.out.println("1 - Incluir");
            System.out.println("2 - Alterar Pessoa");
            System.out.println("3 - Excluir Pessoa");
            System.out.println("4 - Buscar pelo ID");
            System.out.println("5 - Exibir todos");
            System.out.println("6 - Persistir dados");
            System.out.println("7 - Recuperar dados");
            System.out.println("0 - Finalizar Programa");
            System.out.println("=====");
            System.out.print("Escolha uma opção: ");

            try {
                opcao = Integer.parseInt(scanner.nextLine());
            }
            catch (NumberFormatException e){
                System.out.println("Entrada inválida! Digite um número.");
                opcao = -1;
            }

            switch(opcao) {
                case 1 -> {
                    System.out.print("F - Física | J - Jurídica\n");
                    String tipo = scanner.nextLine().toUpperCase();
                    if (tipo.equals("F")) {
                        System.out.print("Digite o ID da pessoa: ");
                        int id = Integer.parseInt(scanner.nextLine());
                        System.out.print("Insira os dados...\n");
                        System.out.print("Nome: ");
                        String nome = scanner.nextLine();
                        System.out.print("CPF: ");
                        String cpf = scanner.nextLine();
                        System.out.print("Idade: ");
                        int idade = Integer.parseInt(scanner.nextLine());

                        repoPF.inserir(new PessoaFisica(id, nome, cpf, idade));
                    }
                }
            }
        } while (opcao != 0);
    }
}

```

```

        System.out.println("Pessoa Física adicionada.");
    }
    else if (tipo.equals("J")) {
        System.out.print("Digite o ID: ");
        int id = Integer.parseInt(scanner.nextLine());
        System.out.print("Insira os dados...\n");
        System.out.print("Nome: ");
        String nome = scanner.nextLine();
        System.out.print("CNPJ: ");
        String cnpj = scanner.nextLine();

        repoPJ.inserir(new PessoaJuridica(id, nome, cnpj));
        System.out.println("Pessoa Jurídica adicionada.");
    }
}

case 2 -> {
    System.out.print("F - Física | J - Jurídica\n");
    String tipo = scanner.nextLine().toUpperCase();
    System.out.print("Digite o ID da pessoa: ");
    int id = Integer.parseInt(scanner.nextLine());

    if (tipo.equals("F")) {
        PessoaFisica pf = repoPF.obter(id);
        if (pf != null) {
            System.out.println("Dados atuais:");
            pf.exibir();

            System.out.print("Altere os dados...\n");
            System.out.print("Nome: ");
            String nome = scanner.nextLine();
            System.out.print("CPF: ");
            String cpf = scanner.nextLine();
            System.out.print("idade: ");
            int idade = Integer.parseInt(scanner.nextLine());

            repoPF.alterar(new PessoaFisica(id, nome, cpf, idade));
            System.out.println("Pessoa Física alterada.");
        }
        else {
            System.out.println("Pessoa Física não encontrada.");
        }
    }
    else if (tipo.equals("J")) {
        PessoaJuridica pj = repoPJ.obter(id);
        if (pj != null) {
            System.out.println("Dados atuais:");
            pj.exibir();

            System.out.print("Altere os dados...\n");
            System.out.print("Nome: ");
            String nome = scanner.nextLine();

```

```

        System.out.print("CNPJ: ");
        String cnpj = scanner.nextLine();

        repoPJ.alterar(new PessoaJuridica(id, nome, cnpj));
        System.out.println("Pessoa Jurídica alterada.");
    }
    else {
        System.out.println("Pessoa Jurídica não encontrada.");
    }
}
}

```

```

case 3 -> {
    System.out.print("F - Física | J - Jurídica\n");
    String tipo = scanner.nextLine().toUpperCase();
    System.out.print("Digite o ID da pessoa: ");
    int id = Integer.parseInt(scanner.nextLine());

    if (tipo.equals("F")) {
        repoPF.excluir(id);
        System.out.println("Pessoa Física excluída.");
    }
    else if (tipo.equals("J")) {
        repoPJ.excluir(id);
        System.out.println("Pessoa Jurídica excluída.");
    }
}
}

```

```

case 4 -> {
    System.out.print("F - Física | J - Jurídica\n");
    String tipo = scanner.nextLine().toUpperCase();
    System.out.print("Digite o ID da pessoa: ");
    int id = Integer.parseInt(scanner.nextLine());

    if (tipo.equals("F")) {
        PessoaFisica pf = repoPF.obter(id);
        if (pf != null) {
            pf.exibir();
        }
        else {
            System.out.println("Pessoa Física não encontrada.");
        }
    }
    else if (tipo.equals("J")) {
        PessoaJuridica pj = repoPJ.obter(id);
        if (pj != null) {
            pj.exibir();
        }
        else {
            System.out.println("Pessoa Jurídica não encontrada.");
        }
    }
}
}

```



```
}
```

```
case 5 -> {
```

```
    System.out.print("F - Física | J - Jurídica\n");
```

```
    String tipo = scanner.nextLine().toUpperCase();
```

```
    if (tipo.equals("F")) {
```

```
        if (repoPF.obterTodos().isEmpty()) {
```

```
            System.out.println("Nenhuma Pessoa Física Cadastrada");
```

```
        }
```

```
        else {
```

```
            for (PessoaFisica pf : repoPF.obterTodos()) {
```

```
                pf.exibir();
```

```
                System.out.println("-----");
```

```
            }
```

```
        }
```

```
    }
```

```
    else if (tipo.equals("J")) {
```

```
        if (repoPJ.obterTodos().isEmpty()) {
```

```
            System.out.println("Nenhuma Pessoa Jurídica Cadastrada");
```

```
        }
```

```
        else {
```

```
            for (PessoaJuridica pj : repoPJ.obterTodos()) {
```

```
                pj.exibir();
```

```
                System.out.println("-----");
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

```
case 6 -> {
```

```
    System.out.print("Digite o prefixo dos arquivos: ");
```

```
    String prefixo = scanner.nextLine();
```

```
    try {
```

```
        repoPF.persistir(prefixo + ".fisica.bin");
```

```
        repoPJ.persistir(prefixo + ".juridica.bin");
```

```
        System.out.println("Dados salvos com sucesso.");
```

```
    } catch (IOException e) {
```

```
        System.out.println("Erro ao salvar dados: " + e.getMessage());
```

```
    }
```

```
}
```

```
case 7 -> {
```

```
    System.out.print("Digite o prefixo dos arquivos: ");
```

```
    String prefixo = scanner.nextLine();
```

```
    try {
```

```
        repoPF.recuperar(prefixo + ".fisica.bin");
```

```
        repoPJ.recuperar(prefixo + ".juridica.bin");
```

```
        System.out.println("Dados recuperados com sucesso.");
```

```
    } catch (IOException | ClassNotFoundException e) {
```

```
        System.out.println("Erro ao recuperar dados: " + e.getMessage());
```

```
    }
```

```

    }

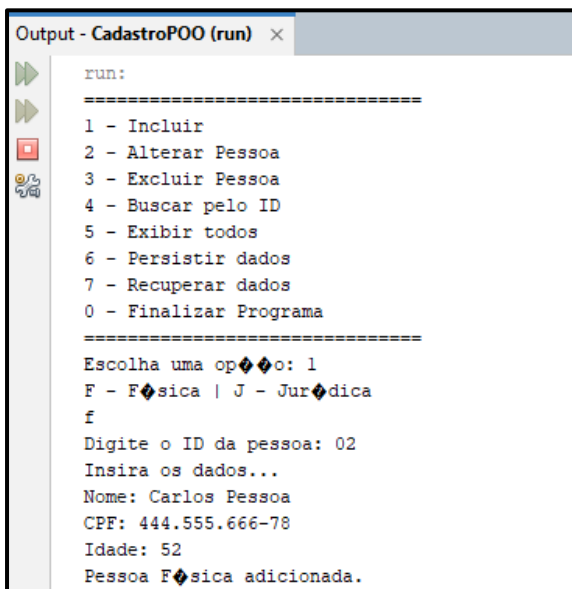
    case 0 -> {
        System.out.println("Finalizando o programa...");
    }

    default -> System.out.println("Opção inválida!");
}
}
while (opcao != 0);
scanner.close();
}
}

```

## Resultados da execução dos códigos

As janelas dos resultados de alguns testes são mostradas na sequência de figuras abaixo.

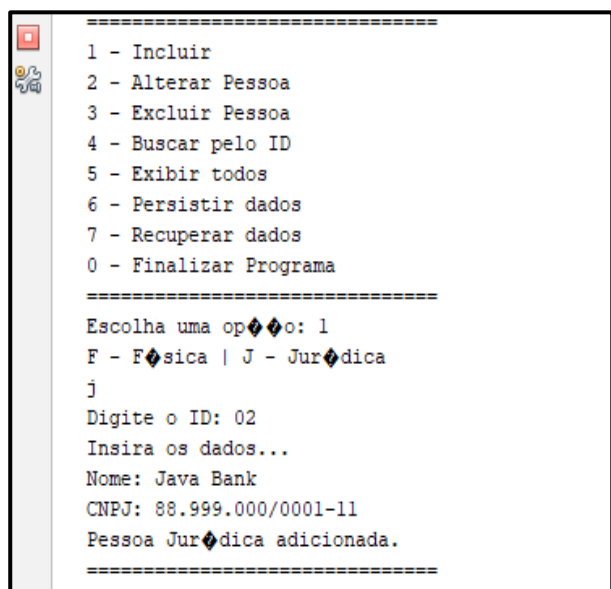


```

run:
=====
1 - Incluir
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo ID
5 - Exibir todos
6 - Persistir dados
7 - Recuperar dados
0 - Finalizar Programa
=====
Escolha uma opção: 1
F - Física | J - Jurídica
f
Digite o ID da pessoa: 02
Insira os dados...
Nome: Carlos Pessoa
CPF: 444.555.666-78
Idade: 52
Pessoa Física adicionada.

```

*Inclusão de pessoa física*

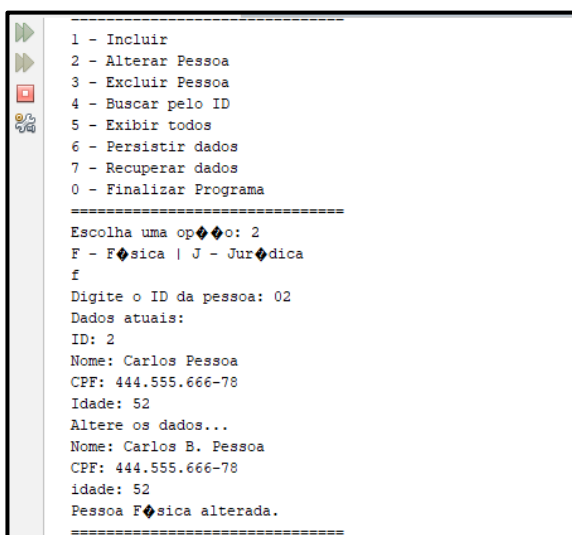


```

=====
1 - Incluir
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo ID
5 - Exibir todos
6 - Persistir dados
7 - Recuperar dados
0 - Finalizar Programa
=====
Escolha uma opção: 1
F - Física | J - Jurídica
J
Digite o ID: 02
Insira os dados...
Nome: Java Bank
CNPJ: 88.999.000/0001-11
Pessoa Jurídica adicionada.
=====

```

*Inclusão de pessoa jurídica*

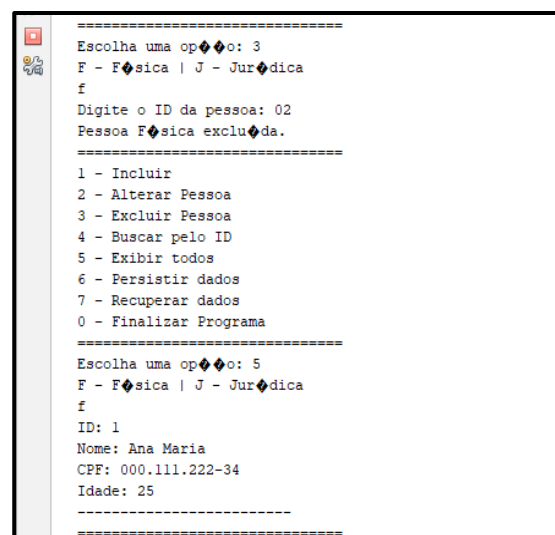


```

=====
1 - Incluir
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo ID
5 - Exibir todos
6 - Persistir dados
7 - Recuperar dados
0 - Finalizar Programa
=====
Escolha uma opção: 2
F - Física | J - Jurídica
f
Digite o ID da pessoa: 02
Dados atuais:
ID: 2
Nome: Carlos Pessoa
CPF: 444.555.666-78
Idade: 52
Altere os dados...
Nome: Carlos B. Pessoa
CPF: 444.555.666-78
idade: 52
Pessoa Física alterada.
=====

```

*Alteração de pessoa física*



```

=====
Escolha uma opção: 3
F - Física | J - Jurídica
f
Digite o ID da pessoa: 02
Pessoa Física excluída.
=====
1 - Incluir
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo ID
5 - Exibir todos
6 - Persistir dados
7 - Recuperar dados
0 - Finalizar Programa
=====
Escolha uma opção: 5
F - Física | J - Jurídica
f
ID: 1
Nome: Ana Maria
CPF: 000.111.222-34
Idade: 25
=====

```

*Exclusão de pessoa física*

```
=====
Pessoa Física alterada.
=====
1 - Incluir
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo ID
5 - Exibir todos
6 - Persistir dados
7 - Recuperar dados
0 - Finalizar Programa
=====
Escolha uma opção: 5
F - Física | J - Jurídica
f
ID: 1
Nome: Ana Maria
CPF: 000.111.222-34
Idade: 25
=====
ID: 2
Nome: Carlos B. Pessoa
CPF: 444.555.666-78
Idade: 52
=====
```

Exibição dos dados

```
=====
1 - Incluir
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo ID
5 - Exibir todos
6 - Persistir dados
7 - Recuperar dados
0 - Finalizar Programa
=====
Escolha uma opção: 4
F - Física | J - Jurídica
j
Digite o ID da pessoa: 2
ID: 2
Nome: XSOL Empreendimentos
CNPJ: 00.111.222/0001-22
=====
```

Busca pelo ID

```
=====
Escolha uma opção: 6
Digite o prefixo dos arquivos: 01
Dados salvos com sucesso.
=====
1 - Incluir
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo ID
5 - Exibir todos
6 - Persistir dados
7 - Recuperar dados
0 - Finalizar Programa
=====
Escolha uma opção: 7
Digite o prefixo dos arquivos: 01
Dados recuperados com sucesso.
=====
1 - Incluir
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo ID
5 - Exibir todos
6 - Persistir dados
7 - Recuperar dados
0 - Finalizar Programa
=====
```

Persistência e recuperação dos dados

```
=====
Escolha uma opção: 7
Digite o prefixo dos arquivos: 01
Dados recuperados com sucesso.
=====
1 - Incluir
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo ID
5 - Exibir todos
6 - Persistir dados
7 - Recuperar dados
0 - Finalizar Programa
=====
Escolha uma opção: 5
F - Física | J - Jurídica
j
ID: 1
Nome: Java Bank
CNPJ: 44.555.666/0001-77
=====
ID: 2
Nome: XSOL Empreendimentos
CNPJ: 00.111.222/0001-22
=====
```

Resultado da recuperação dos dados

## Análise e Conclusão

A missão mostrou o uso dos elementos estáticos, visto no método *main* do código CadastroPOO.java. Elementos estáticos pertencem à classe, podendo ser chamados sem a necessidade de se criar uma instância de classe. Isto permite que o método seja o ponto de entrada do programa.

Outra classe, a *Scanner*, também foi demonstrada no projeto. Ela serve para ler as entradas digitadas no teclado ou os dados de um arquivo.

Por fim, podemos ver como o uso de classes de repositório teve um efeito positivo na organização do código. As funcionalidades de inclusão, alteração, exclusão, obtenção, persistência e

recuperação de dados foram tratadas pelas classes “PessoaFisicaRepo” e “PessoaJuridicaRepo”, deixando a classe principal com a responsabilidade de interação com o usuário. Isto também tem um efeito na clareza do projeto, já que se pode ver a diferença entre as entidades, pessoas físicas e jurídicas, e as classes para controle de dados.

Armazenado no Github: <https://github.com/altairsf/CadastroPOO.git>