

Relatório discente de acompanhamento – 2º Procedimento



Campus: Cabo Frio - EAD

Curso: Desenvolvimento Full Stack

Disciplina: Nível 1: Iniciando o Caminho Pelo Java

Turma: 9001

Semestre Letivo: 2025.1

Integrantes:

- Wellen de Mello Souza

Repositório GIT: https://github.com/Kirah-Dev/mundo_3_Pratica_1_Java_Start.git

Relatório de Prática

Título: Criação do cadastro em modo texto

1. Objetivo da Prática

O objetivo desta prática é aprimorar o sistema de cadastro de pessoas (físicas e jurídicas) desenvolvido anteriormente, adicionando uma interface textual para interação com o usuário. Isso inclui:

- Implementar um menu interativo para as operações de CRUD (Criar, Ler, Atualizar, Deletar).
- Permitir que o usuário insira dados através do teclado.
- Continuar utilizando persistência de dados em arquivos binários para salvar e recuperar as informações.
- Reforçar o uso de tratamento de exceções para garantir a robustez do sistema.

2. Códigos Implementados

Main.java – Atualizado

```
import model.PessoaFisica;
import model.PessoaJuridica;

import java.io.IOException;
import java.util.List;
```

```

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        PessoaFisicaRepo pessoaFisicaRepo = new PessoaFisicaRepo();
        PessoaJuridicaRepo pessoaJuridicaRepo = new
PessoaJuridicaRepo();

        int opcao;
        do {
            System.out.println("=====");
            System.out.println("1 - Incluir Pessoa");
            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("Digite a opção: ");
            opcao = scanner.nextInt();
            scanner.nextLine(); // Consume newline

            switch (opcao) {
                case 1:
                    incluirPessoa(scanner, pessoaFisicaRepo,
pessoaJuridicaRepo);
                    break;
                case 2:
                    alterarPessoa(scanner, pessoaFisicaRepo,
pessoaJuridicaRepo);
                    break;
                case 3:
                    excluirPessoa(scanner, pessoaFisicaRepo,
pessoaJuridicaRepo);
                    break;
                case 4:
                    buscarPeloId(scanner, pessoaFisicaRepo,
pessoaJuridicaRepo);
                    break;
                case 5:
                    exibirTodos(scanner, pessoaFisicaRepo,
pessoaJuridicaRepo);
                    break;
                case 6:
                    persistirDados(scanner, pessoaFisicaRepo,
pessoaJuridicaRepo);
                    break;
                case 7:
                    recuperarDados(scanner, pessoaFisicaRepo,
pessoaJuridicaRepo);
                    break;
                case 0:
                    System.out.println("Finalizando o programa...");
                    break;
                default:

```

```

        System.out.println("Opção inválida. Tente
novamente.");
    }

    } while (opcao != 0);

    scanner.close();
}

private static void incluirPessoa(Scanner scanner,
PessoaFisicaRepo pessoaFisicaRepo, PessoaJuridicaRepo
pessoaJuridicaRepo) {
    System.out.println("F - Pessoa Fisica | J - Pessoa Juridica");
    System.out.print("Digite o tipo de pessoa (F/J): ");
    String tipo = scanner.nextLine().toUpperCase();

    if (tipo.equals("F")) {
        System.out.print("Digite o id da pessoa: ");
        int id = scanner.nextInt();
        scanner.nextLine(); // Consume newline

        System.out.println("Insira os dados...");
        System.out.print("Nome: ");
        String nome = scanner.nextLine();
        System.out.print("CPF: ");
        String cpf = scanner.nextLine();
        System.out.print("Idade: ");
        int idade = scanner.nextInt();
        scanner.nextLine(); // Consume newline

        PessoaFisica pessoaFisica = new PessoaFisica(id, nome,
cpf, idade);
        pessoaFisicaRepo.inserir(pessoaFisica);
        System.out.println("Pessoa Física adicionada com
sucesso!");

    } else if (tipo.equals("J")) {
        System.out.print("Digite o id da pessoa: ");
        int id = scanner.nextInt();
        scanner.nextLine(); // Consume newline

        System.out.println("Insira os dados...");
        System.out.print("Nome: ");
        String nome = scanner.nextLine();
        System.out.print("CNPJ: ");
        String cnpj = scanner.nextLine();

        PessoaJuridica pessoaJuridica = new PessoaJuridica(id,
nome, cnpj);
        pessoaJuridicaRepo.inserir(pessoaJuridica);
        System.out.println("Pessoa Jurídica adicionada com
sucesso!");
    } else {
        System.out.println("Tipo de pessoa inválido.");
    }
}

private static void alterarPessoa(Scanner scanner,
PessoaFisicaRepo pessoaFisicaRepo, PessoaJuridicaRepo
pessoaJuridicaRepo) {
    System.out.println("F - Pessoa Fisica | J - Pessoa Juridica");

```

```

System.out.print("Digite o tipo de pessoa (F/J): ");
String tipo = scanner.nextLine().toUpperCase();

if (tipo.equals("F")) {
    System.out.print("Digite o id da pessoa a ser alterada:");

    int id = scanner.nextInt();
    scanner.nextLine(); // Consume newline

    PessoaFisica pessoaFisica = pessoaFisicaRepo.obter(id);
    if (pessoaFisica == null) {
        System.out.println("Pessoa Física não encontrada.");
        return;
    }

    System.out.println("Dados atuais: ");
    pessoaFisica.exibir();

    System.out.println("Insira os novos dados...");
    System.out.print("Nome: ");
    String nome = scanner.nextLine();
    System.out.print("CPF: ");
    String cpf = scanner.nextLine();
    System.out.print("Idade: ");
    int idade = scanner.nextInt();
    scanner.nextLine(); // Consume newline

    pessoaFisica.setNome(nome);
    pessoaFisica.setCpf(cpf);
    pessoaFisica.setIdade(idade);

    pessoaFisicaRepo.alterar(pessoaFisica);
    System.out.println("Pessoa Física alterada com sucesso!");
} else if (tipo.equals("J")) {
    System.out.print("Digite o id da pessoa a ser alterada:");

    int id = scanner.nextInt();
    scanner.nextLine(); // Consume newline

    PessoaJuridica pessoaJuridica =
pessoaJuridicaRepo.obter(id);
    if (pessoaJuridica == null) {
        System.out.println("Pessoa Jurídica não encontrada.");
        return;
    }

    System.out.println("Dados atuais: ");
    pessoaJuridica.exibir();

    System.out.println("Insira os novos dados...");
    System.out.print("Nome: ");
    String nome = scanner.nextLine();
    System.out.print("CNPJ: ");
    String cnpj = scanner.nextLine();

    pessoaJuridica.setNome(nome);
    pessoaJuridica.setCnpj(cnpj);

    pessoaJuridicaRepo.alterar(pessoaJuridica);
    System.out.println("Pessoa Jurídica alterada com
sucesso!");
}

```

```

        } else {
            System.out.println("Tipo de pessoa inválido.");
        }
    }

    private static void excluirPessoa(Scanner scanner,
        PessoaFisicaRepo pessoaFisicaRepo, PessoaJuridicaRepo
        pessoaJuridicaRepo) {
        System.out.println("F - Pessoa Fisica | J - Pessoa Juridica");
        System.out.print("Digite o tipo de pessoa (F/J): ");
        String tipo = scanner.nextLine().toUpperCase();

        if (tipo.equals("F")) {
            System.out.print("Digite o id da pessoa a ser excluída:
");
            int id = scanner.nextInt();
            scanner.nextLine(); // Consume newline

            pessoaFisicaRepo.excluir(id);
            System.out.println("Pessoa Física excluída com sucesso!");
        } else if (tipo.equals("J")) {
            System.out.print("Digite o id da pessoa a ser excluída:
");
            int id = scanner.nextInt();
            scanner.nextLine(); // Consume newline

            pessoaJuridicaRepo.excluir(id);
            System.out.println("Pessoa Jurídica excluída com
sucesso!");
        } else {
            System.out.println("Tipo de pessoa inválido.");
        }
    }

    private static void buscarPeloId(Scanner scanner, PessoaFisicaRepo
        pessoaFisicaRepo, PessoaJuridicaRepo pessoaJuridicaRepo) {
        System.out.println("F - Pessoa Fisica | J - Pessoa Juridica");
        System.out.print("Digite o tipo de pessoa (F/J): ");
        String tipo = scanner.nextLine().toUpperCase();

        if (tipo.equals("F")) {
            System.out.print("Digite o id da pessoa a ser buscada: ");
            int id = scanner.nextInt();
            scanner.nextLine(); // Consume newline

            PessoaFisica pessoaFisica = pessoaFisicaRepo.obter(id);
            if (pessoaFisica == null) {
                System.out.println("Pessoa Física não encontrada.");
            } else {
                pessoaFisica.exibir();
            }
        } else if (tipo.equals("J")) {
            System.out.print("Digite o id da pessoa a ser buscada: ");
            int id = scanner.nextInt();
            scanner.nextLine(); // Consume newline

            PessoaJuridica pessoaJuridica =
            pessoaJuridicaRepo.obter(id);
            if (pessoaJuridica == null) {
                System.out.println("Pessoa Jurídica não encontrada.");
            } else {

```

```

        pessoaJuridica.exibir();
    }
} else {
    System.out.println("Tipo de pessoa inválido.");
}
}

private static void exibirTodos(Scanner scanner, PessoaFisicaRepo
pessoaFisicaRepo, PessoaJuridicaRepo pessoaJuridicaRepo) {
    System.out.println("F - Pessoa Fisica | J - Pessoa Juridica");
    System.out.print("Digite o tipo de pessoa (F/J): ");
    String tipo = scanner.nextLine().toUpperCase();

    if (tipo.equals("F")) {
        List<PessoaFisica> pessoasFisicas =
pessoaFisicaRepo.obterTodos();
        if (pessoasFisicas.isEmpty()) {
            System.out.println("Nenhuma Pessoa Física
cadastrada.");
        } else {
            for (PessoaFisica pessoaFisica : pessoasFisicas) {
                pessoaFisica.exibir();
                System.out.println("-----");
            }
        }
    } else if (tipo.equals("J")) {
        List<PessoaJuridica> pessoasJuridicas =
pessoaJuridicaRepo.obterTodos();
        if (pessoasJuridicas.isEmpty()) {
            System.out.println("Nenhuma Pessoa Jurídica
cadastrada.");
        } else {
            for (PessoaJuridica pessoaJuridica : pessoasJuridicas)
{
                pessoaJuridica.exibir();
                System.out.println("-----");
            }
        }
    } else {
        System.out.println("Tipo de pessoa inválido.");
    }
}

private static void persistirDados(Scanner scanner,
PessoaFisicaRepo pessoaFisicaRepo, PessoaJuridicaRepo
pessoaJuridicaRepo) {
    System.out.print("Digite o prefixo dos arquivos: ");
    String prefixo = scanner.nextLine();

    try {
        pessoaFisicaRepo.persistir(prefixo + ".fisica.bin");
        pessoaJuridicaRepo.persistir(prefixo + ".juridica.bin");
        System.out.println("Dados persistidos com sucesso nos
arquivos " + prefixo + ".fisica.bin e " + prefixo + ".juridica.bin");
    } catch (IOException e) {
        System.err.println("Erro ao persistir dados: " +
e.getMessage());
    }
}
}

```

```

        private static void recuperarDados(Scanner scanner,
        PessoaFisicaRepo pessoaFisicaRepo, PessoaJuridicaRepo
        pessoaJuridicaRepo) {
            System.out.print("Digite o prefixo dos arquivos: ");
            String prefixo = scanner.nextLine();

            try {
                pessoaFisicaRepo.recuperar(prefixo + ".fisica.bin");
                pessoaJuridicaRepo.recuperar(prefixo + ".juridica.bin");
                System.out.println("Dados recuperados com sucesso dos
arquivos " + prefixo + ".fisica.bin e " + prefixo + ".juridica.bin");
            } catch (IOException | ClassNotFoundException e) {
                System.err.println("Erro ao recuperar dados: " +
e.getMessage());
            }
        }
    }
}

```

3. Resultados da Execução

Saída como na imagem

```

=====
1 - Incluir Pessoa
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo Id
5 - Exibir Todos
6 - Persistir Dados
7 - Recuperar Dados
0 - Finalizar Programa
=====
Digite a opção: 1
F - Pessoa Fisica | J - Pessoa Juridica
Digite o tipo de pessoa (F/J): F
Digite o id da pessoa: 120
Insira os dados...
Nome:

```

Outras saídas - testes

```

=====
1 - Incluir Pessoa
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo Id
5 - Exibir Todos
6 - Persistir Dados
7 - Recuperar Dados
0 - Finalizar Programa
=====
Digite a opção: 1
F - Pessoa Fisica | J - Pessoa Juridica
Digite o tipo de pessoa (F/J): F
Digite o id da pessoa: 123
Insira os dados...
Nome: João Silva
CPF: 12345678900
Idade: 30

```

Pessoa Física adicionada com sucesso!

```
=====
1 - Incluir Pessoa
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo Id
5 - Exibir Todos
6 - Persistir Dados
7 - Recuperar Dados
0 - Finalizar Programa
=====
Digite a opção: 5
F - Pessoa Fisica | J - Pessoa Juridica
Digite o tipo de pessoa (F/J): F
Id: 123
Nome: João Silva
CPF: 12345678900
Idade: 30
-----
```

```
=====
1 - Incluir Pessoa
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo Id
5 - Exibir Todos
6 - Persistir Dados
7 - Recuperar Dados
0 - Finalizar Programa
=====
Digite a opção: 6
Digite o prefixo dos arquivos: cadastro
Dados persistidos com sucesso nos arquivos cadastro.fisica.bin e
cadastro.juridica.bin
```

5. Análise e Conclusão

- **O que são elementos estáticos e qual o motivo para o método main adotar esse modificador?**

Elementos estáticos (variáveis e métodos) pertencem à classe em si, e não a instâncias específicas da classe. Eles são compartilhados por todas as instâncias. O método main é estático porque ele precisa ser acessível sem a necessidade de criar um objeto da classe Main. A JVM (Java Virtual Machine) precisa ser capaz de encontrar e executar o método main sem instanciar a classe.

- **Para que serve a classe Scanner?**

A classe Scanner é utilizada para ler dados de entrada do usuário (ou de arquivos) de forma formatada. Ela permite que o programa receba informações inseridas pelo usuário no console, como números, textos, etc. No nosso projeto, ela é essencial para receber as opções do menu e os dados das pessoas a serem cadastradas.

- **Como o uso de classes de repositório impactou na organização do código?**

O uso de classes de repositório (PessoaFisicaRepo e PessoaJuridicaRepo) permitiu separar a lógica de acesso e manipulação dos dados (CRUD) da lógica principal do programa (a interface textual no Main). Isso tornou o código mais modular, organizado e fácil de manter. As classes de repositório encapsulam a complexidade da persistência, permitindo que a classe Main se concentre na interação com o usuário.