

**DESENVOLVIMENTO FULL STACK**  
**POLO SETOR O – CEILÂNDIA - DF**  
**PERÍODO 2024.2 FLEX**  
**DISCIPLINA: BACK END SEM BANCO NÃO TEM**  
**MAYCON MOURA**

## **Título da Prática**

Cadastro de Pessoas Físicas e Jurídicas com Persistência em Banco de Dados

## **Códigos**

CadastroBD.java

```
package cadastrobd;

import cadastrobd.model.PessoaFisica;
import cadastrobd.model.PessoaFisicaDAO;
import cadastrobd.model.PessoaJuridica;
import cadastrobd.model.PessoaJuridicaDAO;
import cadastrobd.util.SequenceManager;
import java.sql.SQLException;
import java.util.Scanner;
import java.util.logging.Level;
import java.util.logging.Logger;

public class CadastroBD {
    static String sequencialID = "s_pessoa_id";

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        PessoaFisicaDAO pessoaFisicaDAO = new PessoaFisicaDAO();
        PessoaJuridicaDAO pessoaJuridicaDAO = new PessoaJuridicaDAO();

        while (true) {
            try {
                System.out.println("===== Escolha uma opção =====");
                System.out.println("1 - Incluir");
                System.out.println("2 - Alterar");
                System.out.println("3 - Excluir");
                System.out.println("4 - Exibir pelo ID");
                System.out.println("5 - Exibir todos");
                System.out.println("0 - Sair");
                System.out.println("=====");
                System.out.print("Opção: ");
                int opcao = scanner.nextInt();
                scanner.nextLine();
```

```

        if (opcao == 0) {
            System.out.println("Saindo...");
            break;
        }

        switch (opcao) {
            case 1:
                incluir(scanner, pessoaFisicaDAO, pessoaJuridicaDAO);
                break;
            case 2:
                alterar(scanner, pessoaFisicaDAO, pessoaJuridicaDAO);
                break;
            case 3:
                excluir(scanner, pessoaFisicaDAO, pessoaJuridicaDAO);
                break;
            case 4:
                exibirPorId(scanner, pessoaFisicaDAO, pessoaJuridicaDAO);
                break;
            case 5:
                exibirTodos(scanner, pessoaFisicaDAO, pessoaJuridicaDAO);
                break;
            default:
                System.out.println("Opção inválida!");
                break;
        }
    } catch (SQLException ex) {
        Logger.getLogger(CadastroBD.class.getName()).log(Level.SEVERE, null,
ex);
    }
}

scanner.close();
}

```

```

private static void incluir(Scanner scanner, PessoaFisicaDAO pessoaFisicaDAO,
PessoaJuridicaDAO pessoaJuridicaDAO) throws SQLException {
    System.out.print("Incluir Pessoa Física ou Jurídica? (F/J): ");
    String tipo = scanner.nextLine().toUpperCase();
    switch (tipo) {
        case "F":
            {
                System.out.print("Nome: ");
                String nome = scanner.nextLine();
                System.out.print("Endereço: ");
                String endereco = scanner.nextLine();
                System.out.print("Telefone: ");
                String telefone = scanner.nextLine();
            }
        case "J":
            {
                System.out.print("Nome: ");
                String nome = scanner.nextLine();
                System.out.print("Endereço: ");
                String endereco = scanner.nextLine();
                System.out.print("Telefone: ");
                String telefone = scanner.nextLine();
            }
    }
}

```

```

        System.out.print("Email: ");
        String email = scanner.nextLine();
        System.out.print("CPF: ");
        String cpf = scanner.nextLine();
        PessoaFisica pessoaFisica = new
PessoaFisica(SequenceManager.getValue(sequencialID), nome, endereco, telefone,
email, cpf);
        pessoaFisicaDAO.incluir(pessoaFisica);
        System.out.println("Pessoa Física incluída com sucesso!");
        break;
    }
    case "J":
    {
        System.out.print("Nome: ");
        String nome = scanner.nextLine();
        System.out.print("Endereço: ");
        String endereco = scanner.nextLine();
        System.out.print("Telefone: ");
        String telefone = scanner.nextLine();
        System.out.print("Email: ");
        String email = scanner.nextLine();
        System.out.print("CNPJ: ");
        String cnpj = scanner.nextLine();
        PessoaJuridica pessoaJuridica = new
PessoaJuridica(SequenceManager.getValue(sequencialID), nome, endereco,
telefone, email, cnpj);
        pessoaJuridicaDAO.incluir(pessoaJuridica);
        System.out.println("Pessoa Jurídica incluída com sucesso!");
        break;
    }
    default:
        System.out.println("Tipo inválido!");
        break;
    }
}

```

```

private static void alterar(Scanner scanner, PessoaFisicaDAO pessoaFisicaDAO,
PessoaJuridicaDAO pessoaJuridicaDAO) throws SQLException {
    System.out.print("Alterar Pessoa Física ou Jurídica? (F/J): ");
    String tipo = scanner.nextLine().toUpperCase();
    System.out.print("ID: ");
    int id = scanner.nextInt();
    scanner.nextLine();
    switch (tipo) {
        case "F":
            PessoaFisica pessoaFisica = pessoaFisicaDAO.getPessoa(id);
            if (pessoaFisica != null) {
                System.out.println("Dados atuais:");
            }
        }
    }
}

```

```

        pessoaFisica.exibir();

        System.out.print("Novo nome: ");
        String nome = scanner.nextLine();
        System.out.print("Novo endereço: ");
        String endereco = scanner.nextLine();

        System.out.print("Novo telefone: ");
        String telefone = scanner.nextLine();
        System.out.print("Novo email: ");
        String email = scanner.nextLine();
        System.out.print("Novo CPF: ");
        String cpf = scanner.nextLine();

        pessoaFisica.setNome(nome);
        pessoaFisica.setEndereco(endereco);
        pessoaFisica.setTelefone(telefone);
        pessoaFisica.setEmail(email);
        pessoaFisica.setCpf(cpf);
        pessoaFisicaDAO.alterar(pessoaFisica);
        System.out.println("Pessoa Física alterada com sucesso!");
    } else {
        System.out.println("Pessoa Física não encontrada!");
    } break;
case "J":
    PessoaJuridica pessoaJuridica = pessoaJuridicaDAO.getPessoa(id);
    if (pessoaJuridica != null) {
        System.out.println("Dados atuais:");
        pessoaJuridica.exibir();

        System.out.print("Novo nome: ");
        String nome = scanner.nextLine();
        System.out.print("Novo endereço: ");
        String endereco = scanner.nextLine();
        System.out.print("Novo telefone: ");
        String telefone = scanner.nextLine();
        System.out.print("Novo email: ");
        String email = scanner.nextLine();
        System.out.print("Novo CNPJ: ");
        String cnpj = scanner.nextLine();

        pessoaJuridica.setNome(nome);
        pessoaJuridica.setEndereco(endereco);
        pessoaJuridica.setTelefone(telefone);
        pessoaJuridica.setEmail(email);
        pessoaJuridica.setCnpj(cnpj);
        pessoaJuridicaDAO.alterar(pessoaJuridica);
        System.out.println("Pessoa Jurídica alterada com sucesso!");
    }
}

```

```

        } else {
            System.out.println("Pessoa Jurídica não encontrada!");
        } break;
    default:
        System.out.println("Tipo inválido!");
        break;
    }
}

```

```

private static void excluir(Scanner scanner, PessoaFisicaDAO pessoaFisicaDAO,
PessoaJuridicaDAO pessoaJuridicaDAO) throws SQLException {
    System.out.print("Excluir Pessoa Física ou Jurídica? (F/J): ");
    String tipo = scanner.nextLine().toUpperCase();
    System.out.print("ID: ");
    int id = scanner.nextInt();
    scanner.nextLine();
    switch (tipo) {
        case "F":
            pessoaFisicaDAO.excluir(id);
            System.out.println("Pessoa Física excluída com sucesso!");
            break;
        case "J":
            pessoaJuridicaDAO.excluir(id);
            System.out.println("Pessoa Jurídica excluída com sucesso!");
            break;
        default:
            System.out.println("Tipo inválido!");
            break;
    }
}

```

```

private static void exibirPorId(Scanner scanner, PessoaFisicaDAO
pessoaFisicaDAO, PessoaJuridicaDAO pessoaJuridicaDAO) throws SQLException {
    System.out.print("Exibir Pessoa Física ou Jurídica? (F/J): ");
    String tipo = scanner.nextLine().toUpperCase();
    System.out.print("ID: ");
    int id = scanner.nextInt();
    scanner.nextLine();
    switch (tipo) {
        case "F":
            PessoaFisica pessoaFisica = pessoaFisicaDAO.getPessoa(id);
            if (pessoaFisica != null) {
                pessoaFisica.exibir();
            } else {
                System.out.println("Pessoa Física não encontrada!");
            } break;
        case "J":
            PessoaJuridica pessoaJuridica = pessoaJuridicaDAO.getPessoa(id);

```

```

        if (pessoaJuridica != null) {
            pessoaJuridica.exibir();
        } else {
            System.out.println("Pessoa Jurídica não encontrada!");
        } break;
    default:
        System.out.println("Tipo inválido!");
        break;
    }
}

private static void exibirTodos(Scanner scanner, PessoaFisicaDAO
pessoaFisicaDAO, PessoaJuridicaDAO pessoaJuridicaDAO) throws SQLException {
    System.out.print("Exibir todas as Pessoas Físicas ou Jurídicas? (F/J): ");
    String tipo = scanner.nextLine().toUpperCase();

    switch (tipo) {
        case "F":
            System.out.println("Exibindo dados de Pessoa Fisica...");
            System.out.println("=====");
            pessoaFisicaDAO.getPessoas().forEach(PessoaFisica::exibir);
            break;
        case "J":
            System.out.println("Exibindo dados de Pessoa Jurídica1...");
            System.out.println("=====");
            pessoaJuridicaDAO.getPessoas().forEach(PessoaJuridica::exibir);
            break;
        default:
            System.out.println("Tipo inválido!");
            break;
    }
}
}
}

```

## Execução

```

run:
Picked up JAVA_TOOL_OPTIONS: -Dfile.encoding=UTF-8 -
Dsun.jnu.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -
Dsun.stderr.encoding=UTF-8 -Dconsole.encoding=UTF-8
===== Escolha uma opção =====
1 - Incluir
2 - Alterar
3 - Excluir
4 - Exibir pelo ID
5 - Exibir todos

```

0 - Sair

=====

Opção: 1

Incluir Pessoa Física ou Jurídica? (F/J): F

Nome: Arthur

Endereço: Endereço X

Telefone: 55 556688899

Email: arthur@mail.com

CPF: 02365489787

Pessoa Física incluída com sucesso!

===== Escolha uma opção =====

1 - Incluir

2 - Alterar

3 - Excluir

4 - Exibir pelo ID

5 - Exibir todos

0 - Sair

=====

Opção: 5

Exibir todas as Pessoas Físicas ou Jurídicas? (F/J): F

Exibindo dados de Pessoa Física...

=====

ID: 36

Nome: Julia

Endereço: fff

Telefone: 5555

Email: julia

CPF: 01232156898

-----

ID: 0

Nome: Maria

Endereço: dddd

Telefone: 2222

Email: ssss

CPF: 02332223212

-----

ID: 38

Nome: Arthur

Endereço: Endereço X

Telefone: 55 556688899

Email: arthur@mail.com

CPF: 02365489787

-----

===== Escolha uma opção =====

1 - Incluir

2 - Alterar

3 - Excluir

4 - Exibir pelo ID

5 - Exibir todos

0 - Sair

=====

Opção: 4

Exibir Pessoa Física ou Jurídica? (F/J): F

ID: 38

ID: 38

Nome: Arthur

Endereço: Endereço X

Telefone: 55 556688899

Email: arthur@mail.com

CPF: 02365489787

-----

===== Escolha uma opção =====

1 - Incluir

2 - Alterar

3 - Excluir

4 - Exibir pelo ID

5 - Exibir todos

0 - Sair

=====

Opção: 3

Excluir Pessoa Física ou Jurídica? (F/J): F

ID: 38

Pessoa Física excluída com sucesso!

===== Escolha uma opção =====

1 - Incluir

2 - Alterar

3 - Excluir

4 - Exibir pelo ID

5 - Exibir todos

0 - Sair

=====

Opção: 5

Exibir todas as Pessoas Físicas ou Jurídicas? (F/J): F

Exibindo dados de Pessoa Fisica...

=====

ID: 36

Nome: Julia

Endereço: fff

Telefone: 5555

Email: julia

CPF: 01232156898

-----

ID: 0

Nome: Maria

Endereço: dddd

Telefone: 2222



Email: ssss  
CPF: 02332223212

-----  
===== Escolha uma opção =====  
1 - Incluir  
2 - Alterar  
3 - Excluir  
4 - Exibir pelo ID  
5 - Exibir todos  
0 - Sair  
=====

Opção:

## Objetivo da Prática

O objetivo desta prática é desenvolver um sistema de cadastro que permita a inclusão, alteração, exclusão e consulta de pessoas físicas e jurídicas em um banco de dados, utilizando os conceitos de Data Access Object (DAO), JDBC para conexão com o banco, e explorar técnicas avançadas do Java, como o uso de operadores lambda.

## Análise e Conclusão

### Diferenças entre a persistência em arquivo e a persistência em banco de dados

Persistir dados em arquivos é uma abordagem simples, útil para pequenas aplicações, mas que se torna limitada e ineficiente à medida que a aplicação cresce. A persistência em banco de dados oferece melhor desempenho, suporte a transações, e recursos avançados como consultas complexas, integridade referencial e segurança. Além disso, bancos de dados são mais escaláveis e facilitam o acesso concorrente aos dados.

### Como o uso de operador lambda simplificou a impressão dos valores contidos nas entidades, nas versões mais recentes do Java

Os operadores lambda, introduzidos no Java 8, permitem escrever código mais conciso e legível. Em vez de criar classes anônimas ou métodos convencionais para iterar e imprimir dados, os lambdas permitem executar operações como `forEach` diretamente sobre coleções, simplificando a lógica e tornando o código mais expressivo.

### Por que métodos acionados diretamente pelo método main, sem o uso de um objeto, precisam ser marcados como static?

Métodos marcados como `static` pertencem à classe, não a uma instância específica dela. Como o método `main` é estático e serve como ponto de entrada do programa, ele só pode invocar outros métodos ou variáveis que também sejam estáticos. Isso garante que o método `main` possa ser executado sem a necessidade de criar um objeto da classe, permitindo a inicialização direta da aplicação.