



# Relatório Da Prática

**Campus:** [*POLO SÃO JOÃO*]

**Curso:** [*DESENVOLVIMENTO FULL STACK*]

**Disciplina:** [*Back-end Sem Banco Não Tem*]

**Turma:** [*9001*]

**Semestre Letivo:** [*2024.3*]

**Integrantes da Prática:** [*MATHEUS MACEDO SOUSA*]

**Desenvolvimento de aplicativo Java integrado ao banco de dados SQL Server por meio do middleware JDBC.**

## Objetivo da Prática

O objetivo da prática é implementar persistência utilizando o middleware JDBC, aplicar o padrão DAO para o manuseio de dados, realizar o mapeamento objeto-relacional em sistemas Java e desenvolver um aplicativo cadastral que utilize o SQL Server para persistência de dados.

## Códigos Solicitados

```
package cadastrobd;
```

```
import cadastrobd.model.PessoaFisica;  
import cadastrobd.model.PessoaJuridica;  
import cadastrobd.model.dao.PessoaFisicaDAO;  
import cadastrobd.model.dao.PessoaJuridicaDAO;
```

```

import cadastrobd.model.util.ConectorBD;
import java.sql.SQLException;
import java.util.List;
import java.util.Scanner;

/**
 *
 */
public class CadastroBD {

    static Scanner sc;
    private static ConectorBD bd;
    private static PessoaFisicaDAO pfDao;
    private static PessoaJuridicaDAO pjDao;

    public static void main(String[] args) throws SQLException, ClassNotFoundException {

        bd = new ConectorBD();
        pfDao = new PessoaFisicaDAO(bd);
        pjDao = new PessoaJuridicaDAO(bd);

        sc = new Scanner(System.in);
        int opcao;

        do {
            System.out.println("=====");
            System.out.println("1 - Incluir");
            System.out.println("2 - Alterar");
            System.out.println("3 - Excluir");
            System.out.println("4 - Buscar");
            System.out.println("5 - Mostrar Todos");
            System.out.println("0 - Finalizar");
            System.out.println("=====");

            opcao = sc.nextInt();
            sc.nextLine();

            switch (opcao) {
                case 1 -> {
                    try {
                        Incluir();
                    } catch (ClassNotFoundException | SQLException e) {
                        System.out.println("Erro ao incluir pessoa: " + e.getMessage());
                    }
                    break;
                }
            }
        }
    }
}

```

```

    case 2 -> {
        try {
            alterar();
        } catch (ClassNotFoundException | SQLException e) {
            System.out.println("Erro ao alterar pessoa: " + e.getMessage());
        }
        break;
    }

    case 3 -> {
        try {
            excluir();
        } catch (ClassNotFoundException | SQLException e) {
            System.out.println("Erro ao excluir pessoa: " + e.getMessage());
        }
        break;
    }

    case 4 -> {
        try {
            obter();
        } catch (ClassNotFoundException | SQLException e) {
            System.out.println("Erro ao obter pessoa: " + e.getMessage());
        }
        break;
    }

    case 5 -> {
        try {
            obterTodos();
        } catch (ClassNotFoundException | SQLException e) {
            System.out.println("Erro ao obter todas as pessoas: " + e.getMessage());
        }
        break;
    }

    case 0 ->
        System.out.println("Programa finalizado. ");
    default ->
        System.out.println("Opcao invalida. ");

    }

    } while (opcao != 0);

    sc.close();

}

```

```

private static char opcaoPessoa() {
    System.out.println("F - Pessoa Fisica | J - Pessoa Juridica ");
    char opcaoPessoa = sc.next().charAt(0);
    return opcaoPessoa;
}

private static void Incluir() throws SQLException, ClassNotFoundException {
    char opcaoPessoa = opcaoPessoa();

    if (opcaoPessoa == 'F' || opcaoPessoa == 'f') {
        PessoaFisica pf = new PessoaFisica();
        System.out.println("Inserir dados da Pessoa Fisica: ");

        System.out.println("Id: ");
        int id = sc.nextInt();
        pf.setId(id);
        sc.nextLine();
        System.out.println("Nome: ");
        String nome = sc.nextLine();
        pf.setNome(nome);

        System.out.println("Logradouro: ");
        String logradouro = sc.nextLine();
        pf.setLogradouro(logradouro);

        System.out.println("Cidade: ");
        String cidade = sc.nextLine();
        pf.setCidade(cidade);

        System.out.println("Estado: ");
        String estado = sc.nextLine();
        pf.setEstado(estado);

        System.out.println("Telefone: ");
        String telefone = sc.nextLine();
        pf.setTelefone(telefone);

        System.out.println("Email: ");
        String email = sc.nextLine();
        pf.setEmail(email);

        System.out.println("Cpf: ");
        String cpf = sc.nextLine();
        pf.setCpf(cpf);

        pfDao.incluir(pf);

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

```

```

        System.out.println("Dados de Pessoa Fisica incluido com sucesso!");
        System.out.println("=====");
    }
    if (opcaoPessoa == 'J' || opcaoPessoa == 'j') {
        PessoaJuridica pj = new PessoaJuridica();
        System.out.println("Inserir dados da Pessoa Juridica: ");

        System.out.println("Id: ");
        int id = sc.nextInt();
        pj.setId(id);
        sc.nextLine();
        System.out.println("Nome: ");
        String nome = sc.nextLine();
        pj.setNome(nome);

        System.out.println("Logradouro: ");
        String logradouro = sc.nextLine();
        pj.setLogradouro(logradouro);

        System.out.println("Cidade: ");
        String cidade = sc.nextLine();
        pj.setCidade(cidade);

        System.out.println("Estado: ");
        String estado = sc.nextLine();
        pj.setEstado(estado);

        System.out.println("Telefone: ");
        String telefone = sc.nextLine();
        pj.setTelefone(telefone);

        System.out.println("Email: ");
        String email = sc.nextLine();
        pj.setEmail(email);

        System.out.println("Cnpj: ");
        String cnpj = sc.nextLine();
        pj.setCnpj(cnpj);

        pjDao.incluir(pj);

        System.out.println("=====");
        System.out.println("Dados de Pessoa Juridica incluido com sucesso!");
        System.out.println("=====");
    }
}

private static void alterar() throws SQLException, ClassNotFoundException {

```

```

char opcaoPessoa = opcaoPessoa();

if (opcaoPessoa == 'F' || opcaoPessoa == 'f') {

    System.out.println("Qual o ID do cadastro que deseja alterar: ");
    int id = sc.nextInt();
    sc.nextLine();
    PessoaFisica pf = pfDao.getPessoa(id);
    System.out.println("Exibindo dados do ID selecionado para alteracao: ");
    System.out.println(pf.exibir());

    System.out.println("Nome: ");
    String nome = sc.nextLine();
    pf.setNome(nome);

    System.out.println("Logradouro: ");
    String logradouro = sc.nextLine();
    pf.setLogradouro(logradouro);

    System.out.println("Cidade: ");
    String cidade = sc.nextLine();
    pf.setCidade(cidade);

    System.out.println("Estado: ");
    String estado = sc.nextLine();
    pf.setEstado(estado);

    System.out.println("Telefone: ");
    String telefone = sc.nextLine();
    pf.setTelefone(telefone);

    System.out.println("Email: ");
    String email = sc.nextLine();
    pf.setEmail(email);

    System.out.println("Cpf: ");
    String cpf = sc.nextLine();
    pf.setCpf(cpf);

    pfDao.alterar(pf);

    System.out.println("=====");
    System.out.println("Dados de Pessoa Fisica alterado com sucesso!");
    System.out.println("=====");
}

if (opcaoPessoa == 'J' || opcaoPessoa == 'j') {
    System.out.println("Qual o ID do cadastro que deseja alterar: ");

```

```

        int id = sc.nextInt();
        sc.nextLine();
        PessoaJuridica pj = pjDao.getPessoa(id);
        System.out.println("Exibindo dados do ID selecionado para alteracao: ");
        System.out.println(pj.exibir());

        System.out.println("Nome: ");
        String nome = sc.nextLine();
        pj.setNome(nome);

        System.out.println("Logradouro: ");
        String logradouro = sc.nextLine();
        pj.setLogradouro(logradouro);

        System.out.println("Cidade: ");
        String cidade = sc.nextLine();
        pj.setCidade(cidade);

        System.out.println("Estado: ");
        String estado = sc.nextLine();
        pj.setEstado(estado);

        System.out.println("Telefone: ");
        String telefone = sc.nextLine();
        pj.setTelefone(telefone);

        System.out.println("Email: ");
        String email = sc.nextLine();
        pj.setEmail(email);

        System.out.println("Cnpj: ");
        String cnpj = sc.nextLine();
        pj.setCnpj(cnpj);

        pjDao.alterar(pj);

        System.out.println("=====");
        System.out.println("Dados de Pessoa Juridica alterado com sucesso!");
        System.out.println("=====");
    }
}

private static void excluir() throws SQLException, ClassNotFoundException {
    char opcaoPessoa = opcaoPessoa();

    if (opcaoPessoa == 'F' || opcaoPessoa == 'f') {
        System.out.println("Qual o ID do cadastro que deseja excluir: ");
        int id = sc.nextInt();

```

```

        sc.nextLine();
        PessoaFisica pessoa = pfDao.getPessoa(id);
        pfDao.excluir(pessoa);

        System.out.println("=====");
        System.out.println("Dados de Pessoa Fisica excluido com sucesso!");
        System.out.println("=====");
    }

    if (opcaoPessoa == 'J' || opcaoPessoa == 'j') {
        System.out.println("Qual o ID do cadastro que deseja excluir: ");
        int id = sc.nextInt();
        sc.nextLine();
        PessoaJuridica pessoa = pjDao.getPessoa(id);
        pjDao.excluir(pessoa);

        System.out.println("=====");
        System.out.println("Dados de Pessoa Juridica excluido com sucesso!");
        System.out.println("=====");
    }
}

private static void obter() throws ClassNotFoundException, SQLException {
    char opcaoPessoa = opcaoPessoa();

    if (opcaoPessoa == 'F' || opcaoPessoa == 'f') {

        System.out.println("Qual o ID do cadastro que deseja obter: ");
        int id = sc.nextInt();
        sc.nextLine();
        PessoaFisica pf = pfDao.getPessoa(id);
        System.out.println("=====");
        System.out.println("Exibindo dados de Pessoa Fisica");
        System.out.println("=====");

        System.out.println(pf.exibir());
    }

    if (opcaoPessoa == 'J' || opcaoPessoa == 'j') {
        System.out.println("Qual o ID do cadastro que deseja obter: ");
        int id = sc.nextInt();
        sc.nextLine();
        PessoaJuridica pj = pjDao.getPessoa(id);

        System.out.println("=====");
        System.out.println("Exibindo dados de Pessoa Juridica");
        System.out.println("=====");
    }
}

```



```

        System.out.println(pj.exibir());
    }
}

private static void obterTodos() throws ClassNotFoundException, SQLException {
    char opcaoPessoa = opcaoPessoa();

    if (opcaoPessoa == 'F' || opcaoPessoa == 'f') {
        List<PessoaFisica> pfs = pfDao.getPessoas();

        System.out.println("=====");
        System.out.println("Exibindo dados de todas as Pessoas Juridicas");
        System.out.println("=====");

        pfs.forEach(pf -> System.out.println(pf.exibir()));
    }

    if (opcaoPessoa == 'J' || opcaoPessoa == 'j') {
        List<PessoaJuridica> pjs = pjDao.getPessoas();

        System.out.println("=====");
        System.out.println("Exibindo dados de todas as Pessoas Juridicas");
        System.out.println("=====");

        pjs.forEach(pj -> System.out.println(pj.exibir()));
    }
}
}

```

### **Pessoa Física:**

Ao incluir uma pessoa física, o console exibe:

Inserir dados da Pessoa Física:

Id: 1

Nome: João Silva

Logradouro: Rua A

Cidade: São Paulo

Estado: SP

Telefone: (11) 98765-4321

Email: joao@email.com

CPF: 123.456.789-00

- Dados de Pessoa Física incluídos com sucesso!

## **Pessoa Jurídica:**

Ao incluir uma pessoa jurídica, o console exibe:

Inserir dados da Pessoa Jurídica:

Id: 2

Nome: Empresa X

Logradouro: Avenida B

Cidade: Rio de Janeiro

Estado: RJ

Telefone: (21) 99876-5432

Email: contato@empresax.com

CNPJ: 12.345.678/0001-90

- Dados de Pessoa Jurídica incluídos com sucesso!

- **Alteração de Pessoa Física:**

Após alterar os dados, o console exibe:

csharp

Exibindo dados do ID selecionado para alteração:

[Dados antigos...]

Nome: João Pereira

- Dados de Pessoa Física alterados com sucesso!

## **Exclusão de Dados**

- **Exclusão de Pessoa Física:**

Após a exclusão, o console exibe:

Dados de Pessoa Física excluídos com sucesso!

## Resultados da Execução dos Códigos

Os resultados demonstram que o aplicativo funciona conforme esperado, permitindo a inclusão, alteração, exclusão e consulta de registros de forma interativa e amigável. A interação via console é eficiente, embora futuras implementações possam considerar interfaces gráficas para facilitar a visualização e manipulação dos dados.

```
Output
CadastroBD (run) ×  CadastroBD (run) #2 ×
run:
=====
1 - Incluir
2 - Alterar
3 - Excluir
4 - Buscar
5 - Mostrar Todos
0 - Finalizar
=====
1
F - Pessoa Fisica | J - Pessoa Juridica
f
Inserir dados da Pessoa Fisica:
Id:
12
Nome:
mauro venicius
Logradouro:
porto velho, 3579
Cidade:
teresina
Estado:
piaui
Telefone:
8695896542
Email:
mv_mil@gamil.com
Cpf:
12336545687
Erro ao incluir pessoa: com.microsoft.sqlserver.jdbc.SQLServerDriver
=====
1 - Incluir
2 - Alterar
```

# **Análise e Conclusão**

## **Diferenças entre Persistência em Arquivo e Persistência em Banco de Dados**

A persistência em arquivo armazena dados em arquivos simples, sendo fácil de implementar e adequada para aplicações pequenas. No entanto, carece de funcionalidades como consultas complexas e integridade referencial. Em contraste, a persistência em banco de dados utiliza sistemas gerenciadores que oferecem robustez, segurança e suporte a transações, permitindo a manipulação eficiente de grandes volumes de dados.

## **Uso de Operador Lambda em Java**

Os operadores lambda nas versões mais recentes do Java simplificam a impressão de valores em entidades, proporcionando uma sintaxe concisa e legível. Eles permitem o uso de expressões funcionais para iterar coleções, como o `forEach`, tornando o código mais limpo e fácil de manter, sem a necessidade de implementar interfaces como `Runnable` ou `Consumer`.

## **Métodos Static e a Chamada Direta pelo Método Main**

Métodos chamados diretamente pelo método `main` devem ser marcados como `static` porque pertencem à classe, e não a uma instância específica. Essa marcação permite que o método `main`, como ponto de entrada da aplicação, seja chamado sem criar um objeto da classe, facilitando a execução do programa.