

	<p align="center"><b>Universidade Estácio</b></p> <p align="center"><b>Polo São Lourenço da Mata</b></p> <p align="center"><b>Desenvolvimento Full Stack</b></p> <p align="center"><b>Semestre 2024.1</b></p>	<p>Disciplina: <b>Back-end Sem Banco Não Tem.</b></p> <p>Aluno: <b>Manoel José</b></p> <p>Matrícula: 202301361117</p> <p>Turma: 2023.1</p>
---	---	--

**Criação do aplicativo Java, com acesso ao banco de dados SQL Server através do middleware JDBC.**

**Objetivos da Prática:**

1. Implementar persistência com base no middleware JDBC.
2. Utilizar o padrão DAO (Data Access Object) no manuseio de dados.
3. Implementar o mapeamento objeto-relacional em sistemas Java.
4. Criar sistemas cadastrais com persistência em banco relacional.
5. No final do exercício, o aluno terá criado um aplicativo cadastral com uso do SQL Server na persistência de dados.

**2º Procedimento | Alimentando a Base**

**1- Alterar o método main da classe principal do projeto, para implementação do cadastro em modo texto:**

**Classe CadastroBD (Main):**

```
package cadastrobd;

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

import java.util.List;
import java.util.Scanner;

public class CadastroBD {
```

```

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

    int opcao;
    do {
        exibirMenu();
        System.out.print("Opção: ");
        opcao = scanner.nextInt();
        scanner.nextLine(); // Limpar o buffer

        switch (opcao) {
            case 1:
                incluirOpcao();
                break;
            case 2:
                alterarOpcao();
                break;
            case 3:
                excluirOpcao();
                break;
            case 4:
                exibirPorIdOpcao();
                break;
            case 5:
                exibirTodosOpcao();
                break;
            case 0:
                System.out.println("Saindo do programa...");
                break;
            default:
                System.out.println("Opção inválida. Tente novamente.");
                break;
        }
    } while (opcao != 0);

    scanner.close();
}

private static void exibirMenu() {
    System.out.println("\n===== MENU =====");
    System.out.println("1 - Incluir");
    System.out.println("2 - Alterar");
    System.out.println("3 - Excluir");
    System.out.println("4 - Exibir por ID");
}

```

```

        System.out.println("5 - Exibir todos");
        System.out.println("0 - Sair");
        System.out.println("=====\n");
    }

    private static void incluirOpcao() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Selecione o tipo de pessoa:");
        System.out.println("1 - Pessoa Física");
        System.out.println("2 - Pessoa Jurídica");
        System.out.print("Opção: ");
        int tipo = scanner.nextInt();
        scanner.nextLine(); // Limpar o buffer

        if (tipo == 1) {
            // Incluir Pessoa Física
            incluirPessoaFisica(scanner);
        } else if (tipo == 2) {
            // Incluir Pessoa Jurídica
            incluirPessoaJuridica(scanner);
        } else {
            System.out.println("Opção inválida.");
        }
    }

    private static void incluirPessoaFisica(Scanner scanner) {
        // Implementar a lógica de inclusão de Pessoa Física
    }

    private static void incluirPessoaJuridica(Scanner scanner) {
        // Implementar a lógica de inclusão de Pessoa Jurídica
    }

    private static void alterarOpcao() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Selecione o tipo de pessoa:");
        System.out.println("1 - Pessoa Física");
        System.out.println("2 - Pessoa Jurídica");
        System.out.print("Opção: ");
        int tipo = scanner.nextInt();
        scanner.nextLine(); // Limpar o buffer

        if (tipo == 1) {
            // Alterar Pessoa Física

```

```

        alterarPessoaFisica(scanner);
    } else if (tipo == 2) {
        // Alterar Pessoa Jurídica
        alterarPessoaJuridica(scanner);
    } else {
        System.out.println("Opção inválida.");
    }
}

```

```

private static void alterarPessoaFisica(Scanner scanner) {
    // Implementar a lógica de alteração de Pessoa Física
}

```

```

private static void alterarPessoaJuridica(Scanner scanner) {
    // Implementar a lógica de alteração de Pessoa Jurídica
}

```

```

private static void excluirOpcao() {
    Scanner scanner = new Scanner(System.in);
    System.out.println("Selecione o tipo de pessoa:");
    System.out.println("1 - Pessoa Física");
    System.out.println("2 - Pessoa Jurídica");
    System.out.print("Opção: ");
    int tipo = scanner.nextInt();
    System.out.print("Informe o ID da pessoa: ");
    int id = scanner.nextInt();

    if (tipo == 1) {
        // Excluir Pessoa Física
        PessoaFisicaDAO pessoaFisicaDAO = new PessoaFisicaDAO();
        pessoaFisicaDAO.excluir(id);
    } else if (tipo == 2) {
        // Excluir Pessoa Jurídica
        PessoaJuridicaDAO pessoaJuridicaDAO = new PessoaJuridicaDAO();
        pessoaJuridicaDAO.excluir(id);
    } else {
        System.out.println("Opção inválida.");
    }
}
}

```

```

private static void exibirPorIdOpcao() {
    Scanner scanner = new Scanner(System.in);
    System.out.println("Selecione o tipo de pessoa:");
    System.out.println("1 - Pessoa Física");
}

```

```

System.out.println("2 - Pessoa Jurídica");
System.out.print("Opção: ");
int tipo = scanner.nextInt();
System.out.print("Informe o ID da pessoa: ");
int id = scanner.nextInt();

switch (tipo) {
    case 1:
    {
        // Exibir Pessoa Física por ID
        PessoaFisicaDAO pessoaFisicaDAO = new PessoaFisicaDAO();
        PessoaFisica pessoa = pessoaFisicaDAO.getPessoa(id);
        if (pessoa != null) {
            pessoa.exibir();
        } else {
            System.out.println("Pessoa física com o ID " + id + " não encontrada.");
        } break;
    }
    case 2:
    {
        // Exibir Pessoa Jurídica por ID
        PessoaJuridicaDAO pessoaJuridicaDAO = new PessoaJuridicaDAO();
        PessoaJuridica pessoa = pessoaJuridicaDAO.getPessoa(id);
        if (pessoa != null) {
            pessoa.exibir();
        } else {
            System.out.println("Pessoa jurídica com o ID " + id + " não encontrada.");
        } break;
    }
    default:
        System.out.println("Opção inválida.");
        break;
}
}

```

```

private static void exibirTodosOpcao() {
    Scanner scanner = new Scanner(System.in);
    System.out.println("Selecione o tipo de pessoa:");
    System.out.println("1 - Pessoa Física");
    System.out.println("2 - Pessoa Jurídica");
    System.out.print("Opção: ");
    int tipo = scanner.nextInt();

    if (tipo == 1) {

```

```

        // Exibir todas as Pessoas Físicas
        PessoaFisicaDAO pessoaFisicaDAO = new PessoaFisicaDAO();
        List<PessoaFisica> pessoasFisicas = pessoaFisicaDAO.getPessoas();
        for (PessoaFisica pessoa : pessoasFisicas) {
            pessoa.exibir();
        }
    } else if (tipo == 2) {
        // Exibir todas as Pessoas Jurídicas
        PessoaJuridicaDAO pessoaJuridicaDAO = new PessoaJuridicaDAO();
        List<PessoaJuridica> pessoasJuridicas = pessoaJuridicaDAO.getPessoas();
        for (PessoaJuridica pessoa : pessoasJuridicas) {
            pessoa.exibir();
        }
    } else {
        System.out.println("Opção inválida.");
    }
}
}

```

## 2- Testar as funcionalidades do sistema:

## Output - CadastroBD (run)

```
run:

===== MENU =====
1 - Incluir
2 - Alterar
3 - Excluir
4 - Exibir por ID
5 - Exibir todos
0 - Sair
=====

Opção: 4
Selecione o tipo de pessoa:
1 - Pessoa Física
2 - Pessoa Jurídica
Opção: 1
Informe o ID da pessoa: 7
ID: 7
Nome: Joao
Logradouro: Rua 12, casa 3, Quitanda
Cidade: Riacho do Sul
Estado: PA
Telefone: 0
Email: joao@riacho.com
CPF: 111111111111

===== MENU =====
1 - Incluir
2 - Alterar
3 - Excluir
4 - Exibir por ID
5 - Exibir todos
0 - Sair
=====

Opção: 0
Saindo do programa...
BUILD SUCCESSFUL (total time: 17 seconds)
|
```

run:

===== MENU =====

- 1 - Incluir
- 2 - Alterar
- 3 - Excluir
- 4 - Exibir por ID
- 5 - Exibir todos
- 0 - Sair

=====

Opção: 4

Selecione o tipo de pessoa:

- 1 - Pessoa Física
- 2 - Pessoa Jurídica

Opção: 1

Informe o ID da pessoa: 7

ID: 7

Nome: Joao

Logradouro: Rua 12, casa 3, Quitanda

Cidade: Riacho do Sul

Estado: PA

Telefone: 0

Email: joao@riacho.com

CPF: 111111111111

===== MENU =====

- 1 - Incluir
- 2 - Alterar
- 3 - Excluir
- 4 - Exibir por ID
- 5 - Exibir todos
- 0 - Sair

=====

Opção: 0

Saindo do programa...

BUILD SUCCESSFUL (total time: 17 seconds)

### Análise e Conclusão:

- A. Quais as diferenças entre a persistência em arquivo e a persistência em banco de dados?



**R-** A persistência em arquivo armazena dados em arquivos no sistema de arquivos do computador, enquanto a persistência em banco de dados utiliza um sistema de gerenciamento de banco de dados (SGBD) para armazenar e gerenciar os dados de forma organizada e acessível por meio de consultas estruturadas. Os bancos de dados oferecem recursos como consultas complexas, indexação e segurança, que os arquivos simples não possuem.

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

**R-** O uso de operadores lambda no Java simplificou a impressão dos valores contidos nas entidades ao permitir uma sintaxe mais concisa e expressiva para funções anônimas. Por exemplo, ao utilizar métodos como `forEach` em coleções, em vez de escrever uma classe anônima, agora você pode usar uma expressão lambda para definir o comportamento a ser executado em cada elemento da coleção. Isso reduz significativamente a quantidade de código necessário e torna o código mais legível e fácil de entender.

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

**R-** Métodos acionados diretamente pelo método `main`, sem o uso de um objeto, precisam ser marcados como `static` porque o método `main` é estático. Isso significa que ele pertence à classe em vez de pertencer a instâncias individuais da classe. Para que o método `main` possa chamar outros métodos diretamente, esses métodos também precisam ser estáticos, pois métodos estáticos só podem chamar outros métodos estáticos diretamente. Isso permite que o método `main` seja chamado sem a necessidade de criar uma instância da classe.