

Aluno: Altair Silva Filho

Matrícula: 202403254409

Unidade: Palhoça – SC

Curso: Desenvolvimento Full Stack

Disciplina: Back-End sem Banco Não Tem

Período: 2025.1

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

### **Back-End Sem Banco Não Tem**

#### **2º Procedimento | Alimentando a Base**

##### **Objetivo**

O projeto visa criar um projeto na plataforma Java, usando a IDE NetBeans, com acesso a Banco de Dados SQL através dos componentes do JDBC.

Neste procedimento, foram feitas alterações no método *main* da classe principal para alimentar o banco, com a implementação do cadastro em modo texto.

##### **Códigos do Projeto**

###### **Classe Principal de Testes**

*ConectorBDTeste.java*

```
import cadastrobd.model.PessoaFisica;
import cadastrobd.model.PessoaFisicaDAO;
import cadastrobd.model.PessoaJuridica;
import cadastrobd.model.PessoaJuridicaDAO;
import java.io.IOException;
import java.util.List;
import java.util.Scanner;

public class CadastroBDTeste {

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

        PessoaFisicaDAO pfDAO = new PessoaFisicaDAO();
        PessoaJuridicaDAO pjDAO = new PessoaJuridicaDAO();

        int opcao;
```

```

while (true) {
    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("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 -> { //Incluir
            System.out.print("F - Física | J - Jurídica\n");
            String tipo = scanner.nextLine().toUpperCase();
            if (tipo.equals("F")) {
                System.out.print("Insira os dados...\n");
                System.out.print("Nome: ");
                String nome = scanner.nextLine();
                System.out.print("Logradouro: ");
                String logradouro = scanner.nextLine();
                System.out.print("Cidade: ");
                String cidade = scanner.nextLine();
                System.out.print("Estado: ");
                String estado = 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();

                pfDAO.incluir(new PessoaFisica(0, nome, logradouro, cidade, estado,
                telefone, email, cpf));
                System.out.println("Pessoa Física incluída com sucesso.");
            }
            else if (tipo.equals("J")) {
                System.out.print("Insira os dados...\n");
                System.out.print("Razão Social: ");
                String nome = scanner.nextLine();
                System.out.print("Logradouro: ");
                String logradouro = scanner.nextLine();
                System.out.print("Cidade: ");
                String cidade = scanner.nextLine();

```

```

        System.out.print("Estado: ");
        String estado = 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();

        pjDAO.incluir(new PessoaJuridica(0, nome, logradouro, cidade, estado,
            telefone, email, cnpj));
        System.out.println("Pessoa Jurídica incluída com sucesso.");
    }
}

```

```

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

    if (tipo.equals("F")) {
        PessoaFisica pf = pfDAO.getPessoa(idAlt);
        if (pf != null) {
            System.out.println("Dados atuais:");
            pf.exibir();

            System.out.print("Altere os dados...\n");
            System.out.print("Novo Nome: ");
            pf.setNome(scanner.nextLine());
            System.out.print("Novo Logradouro: ");
            pf.setLogradouro(scanner.nextLine());
            System.out.print("Nova Cidade: ");
            pf.setCidade(scanner.nextLine());
            System.out.print("Novo Estado: ");
            pf.setEstado(scanner.nextLine());
            System.out.print("Novo Telefone: ");
            pf.setTelefone(scanner.nextLine());
            System.out.print("Novo email: ");
            pf.setEmail(scanner.nextLine());
            System.out.print("Novo CPF: ");
            pf.setCpf(scanner.nextLine());

            pfDAO.alterar(pf);
            System.out.println("Pessoa Física alterada.");
        }
        else {
            System.out.println("Pessoa Física não encontrada.");
        }
    }
    else if (tipo.equals("J")) {
        PessoaJuridica pj = pjDAO.getPessoa(idAlt);

```

```

    if (pj != null) {
        System.out.println("Dados atuais:");
        pj.exibir();

        System.out.print("Altere os dados...\n");
        System.out.print("Nova Razão Social: ");
        pj.setNome(scanner.nextLine());
        System.out.print("Novo Logradouro: ");
        pj.setLogradouro(scanner.nextLine());
        System.out.print("Nova Cidade: ");
        pj.setCidade(scanner.nextLine());
        System.out.print("Novo Estado: ");
        pj.setEstado(scanner.nextLine());
        System.out.print("Novo Telefone: ");
        pj.setTelefone(scanner.nextLine());
        System.out.print("Novo email: ");
        pj.setEmail(scanner.nextLine());
        System.out.print("Novo CNPJ: ");
        pj.setCnpj(scanner.nextLine());

        pjDAO.alterar(pj);
        System.out.println("Pessoa Jurídica alterada.");
    }
    else {
        System.out.println("Pessoa Jurídica não encontrada.");
    }
}
}

```

```

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

    if (tipo.equals("F")) {
        pfDAO.excluir(idExc);
        System.out.println("Pessoa Física excluída.");
    }
    else if (tipo.equals("J")) {
        pjDAO.excluir(idExc);
        System.out.println("Pessoa Jurídica excluída.");
    }
}
}

```

```

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

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

```



```
Output - CadastroBD (run) x
Escolha uma opção: 1
F - Física | J - Jurídica
f
Insira os dados...
Nome: Maria Rocha
Logradouro: Rua 7 de setembro
Cidade: São Paulo
Estado: SP
Telefone: 11998876655
Email: mariarocha@maria.com
CPF: 22233344455
Pessoa Física incluída com sucesso.
=====
1 - Incluir
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo ID
5 - Exibir todos
0 - Finalizar Programa
=====
Escolha uma opção: 1
F - Física | J - Jurídica
j
Insira os dados...
Razão Social: KL LTDA
Logradouro: Rua 12
Cidade: Rio de Janeiro
Estado: RJ
Telefone: 21987654321
Email: contato@kl.com
CNPJ: 0011222000133
Pessoa Jurídica incluída com sucesso.
=====
```

Figura 1 – Resultado da inserção de Pessoas – Física e Jurídica

```
Output - CadastroBD (run) x
E-mail: joao@riacho.com
CPF: 11100022233
ID: 57
Nome: Maria Rocha
Logradouro: Rua 7 de setembro
Cidade: São Paulo
Estado: SP
Telefone: 11998876655
E-mail: mariarocha@maria.com
CPF: 22233344455
=====
1 - Incluir
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo ID
5 - Exibir todos
0 - Finalizar Programa
=====
Escolha uma opção: 4
F - Física | J - Jurídica
j
Digite o ID da pessoa: 58
ID: 58
Nome: KL LTDA
Logradouro: Rua 12
Cidade: Rio de Janeiro
Estado: RJ
Telefone: 21987654321
E-mail: contato@kl.com
CNPJ: 0011222000133
=====
```

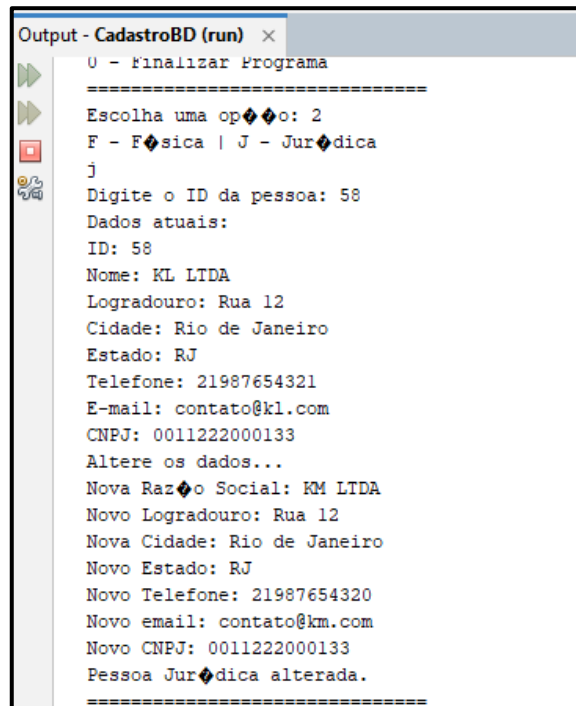
Figura 2 – Resultado da opção 4 – Buscar pelo ID

```
=====
1 - Incluir
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo ID
5 - Exibir todos
0 - Finalizar Programa
=====
Escolha uma opção: 3
F - Física | J - Jurídica
f
Digite o ID da pessoa: 57
Pessoa Física excluída.
=====
1 - Incluir
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo ID
5 - Exibir todos
0 - Finalizar Programa
=====
```

Figura 3 – Resultado da opção 3 - Exclusão

```
Output - CadastroBD (run) #3 x
run:
=====
1 - Incluir
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo ID
5 - Exibir todos
0 - Finalizar Programa
=====
Escolha uma opção: 0
Finalizando o programa...
Sistema finalizado
BUILD SUCCESSFUL (total time: 3 seconds)
```

Figura 2 – Resultado da opção 0 - Finalizar



```
U - Finalizar Programa
=====
Escolha uma opção: 2
F - Física | J - Jurídica
j
Digite o ID da pessoa: 58
Dados atuais:
ID: 58
Nome: KL LTDA
Logradouro: Rua 12
Cidade: Rio de Janeiro
Estado: RJ
Telefone: 21987654321
E-mail: contato@kl.com
CNPJ: 0011222000133
Altere os dados...
Nova Razão Social: KM LTDA
Novo Logradouro: Rua 12
Nova Cidade: Rio de Janeiro
Novo Estado: RJ
Novo Telefone: 21987654320
Novo email: contato@km.com
Novo CNPJ: 0011222000133
Pessoa Jurídica alterada.
=====
```

Figura 5 – Resultado da opção 2 - Alteração

## Análise e Conclusão

O segundo procedimento da missão destacou como o uso dos componentes de middleware, como o JDBC, é importante para a criação de aplicações, pois permitem a integração de forma transparente entre o back-end e o front-end com pouca ou nenhuma alteração do código para os diversos produtos dos fornecedores. No caso, o JDBC serve como driver de conexão no ambiente JAVA que permite a conexão com um banco de dados SQL.

A propósito, esta, a permanência em banco, é uma das estratégias usadas para se preservar informações, o estado das aplicações e dados de forma geral. Já tínhamos estudado um pouco sobre isso no Nível 1 - Mundo 3, quando vimos como salvar as informações em arquivos. Este método, embora seja mais simples de implementar, é menos flexível e eficiente para a manipulação de grandes volumes de dados ou de suportar operações mais complexas.

Por outro lado, a permanência em banco de dados organiza de modo mais eficiente e estruturadas das informações, permitindo consultas e transações mais complexas, bem como a manipulação de grandes volumes de informação.

Uma outra característica dos programas em Java mostrada nesta missão, foi como o uso de operadores “lambda” simplificam a leitura, escrita e manipulação de coleções. Isso é muito útil, por exemplo, ao imprimir valores contidos em uma lista de entidades. Foi usado este operador neste procedimento no arquivo *CadastroBDTeste.java* na opção de exibição de todas as pessoas do sistema.

Por fim, verificou-se como os métodos static são usados. Estes estão relacionados com como o Java gerencia a execução de programas sem instanciar objetos previamente, como na classe de inicialização.

