

# **Desenvolvimento Full Stack**

**Nível 1: Iniciando o Caminho Pelo Java**

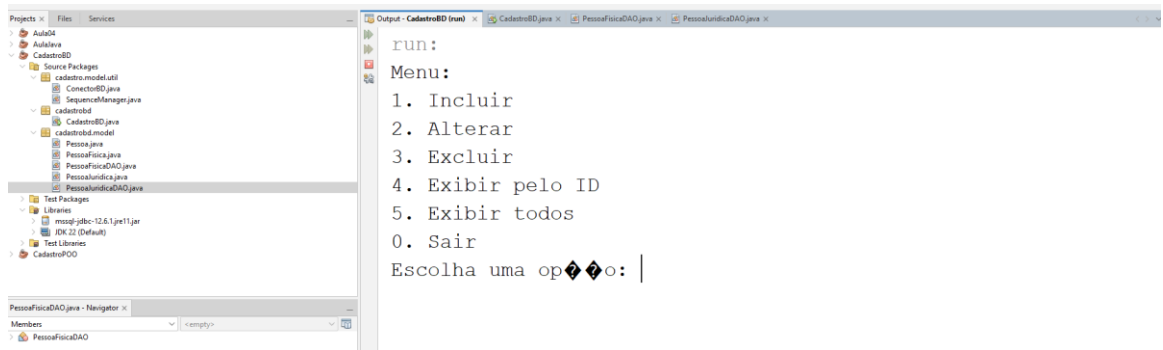
**2023.1**

**Mundo 3 Período 2024.1**

**Breno Félix de Souza**

## 2º Procedimento | Alimentando a Base

- 1) Alterar o método main da classe principal do projeto, para implementação do cadastro em modo texto:
  - a. Apresentar as opções do programa para o usuário, sendo 1 para incluir, 2 para alterar, 3 para excluir, 4 para exibir pelo id, 5 para exibir todos e 0 para finalizar a execução.



- b. Selecionada a opção incluir, escolher o tipo (Física ou Jurídica), receber os dados a partir do teclado e adicionar no banco de dados através da classe DAO correta.**

```
private static void incluir(Scanner scanner, PessoaFisicaDAO pfDao,
PessoaJuridicaDAO pjDao) throws SQLException {

    System.out.print("Tipo (1. Física / 2. Jurídica): ");

    int tipo = Integer.parseInt(scanner.nextLine());

    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();

    if (tipo == 1) {

        System.out.print("CPF: ");

        String cpf = scanner.nextLine();

        PessoaFisica pf = new PessoaFisica(0, nome, logradouro, cidade, estado,
telefone, email, cpf);

        pfDao.incluir(pf);

    } else if (tipo == 2) {

        System.out.print("CNPJ: ");

        String cnpj = scanner.nextLine();

        PessoaJuridica pj = new PessoaJuridica(0, nome, logradouro, cidade, estado,
telefone, email, cnpj);

        pjDao.incluir(pj);

    } else {

        System.out.println("Tipo inválido.");

    }

}
```

- c. Selecionada a opção alterar, escolher o tipo (Física ou Jurídica), receber o id a partir do teclado, apresentar os dados atuais, solicitar os novos dados e alterar no banco de dados através do DAO.**

```
private static void alterar(Scanner scanner, PessoaFisicaDAO pfDao, PessoaJuridicaDAO pjDao) throws
SQLException {

    System.out.print("Tipo (1. Física / 2. Jurídica): ");

    int tipo = Integer.parseInt(scanner.nextLine());

    System.out.print("ID: ");

    int id = Integer.parseInt(scanner.nextLine());

    if (tipo == 1) {

        PessoaFisica pf = pfDao.getPessoa(id);

        if (pf != null) {

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

            pf.exibir();

            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);

    } else {

        System.out.println("Pessoa física não encontrada.");

    }

} else if (tipo == 2) {

    PessoaJuridica pj = pjDao.getPessoa(id);

    if (pj != null) {

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

        pj.exibir();

        System.out.print("Novo Nome: ");
        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);

    } else {

        System.out.println("Pessoa jurídica não encontrada.");

    }

} else {

    System.out.println("Tipo inválido.");

}

}
```

- d. **Selecione a opção excluir, escolher o tipo (Física ou Jurídica), receber o id a partir do teclado e remover do banco de dados através do DAO.**

```
private static void excluir(Scanner scanner, PessoaFisicaDAO pfDao,
PessoaJuridicaDAO pjDao) throws SQLException {

    System.out.print("Tipo (1. Física / 2. Jurídica): ");

    int tipo = Integer.parseInt(scanner.nextLine());

    System.out.print("ID: ");

    int id = Integer.parseInt(scanner.nextLine());

    if (tipo == 1) {
        pfDao.excluir(id);
    } else if (tipo == 2) {
        pjDao.excluir(id);
    } else {
        System.out.println("Tipo inválido.");
    }
}
```

- e. **Selecionada a opção obter, escolher o tipo (Física ou Jurídica), receber o id a partir do teclado e apresentar os dados atuais, recuperados do banco através do DAO.**

```
private static void exibirPeloid(Scanner scanner, PessoaFisicaDAO pfDao, PessoaJuridicaDAO
pjDao) throws SQLException {

    System.out.print("Tipo (1. Física / 2. Jurídica): ");

    int tipo = Integer.parseInt(scanner.nextLine());

    System.out.print("ID: ");

    int id = Integer.parseInt(scanner.nextLine());

    if (tipo == 1) {

        PessoaFisica pf = pfDao.getPessoa(id);

        if (pf != null) {

            pf.exibir();

        } else {

            System.out.println("Pessoa física não encontrada.");

        }

    } else if (tipo == 2) {

        PessoaJuridica pj = pjDao.getPessoa(id);

        if (pj != null) {

            pj.exibir();

        } else {

            System.out.println("Pessoa jurídica não encontrada.");

        }

    } else {

        System.out.println("Tipo inválido.");

    }

}
```

- f. **Selecione a opção obterTodos, escolher o tipo (Física ou Jurídica) e apresentar os dados de todas as entidades presentes no banco de dados por intermédio do DAO.**

```
private static void exibirTodos(Scanner scanner, PessoaFisicaDAO pfDao,
PessoaJuridicaDAO pjDao) throws SQLException {

    System.out.print("Tipo (1. Física / 2. Jurídica): ");

    int tipo = Integer.parseInt(scanner.nextLine());

    if (tipo == 1) {

        List<PessoaFisica> pessoas = pfDao.getPessoas();

        for (PessoaFisica pf : pessoas) {

            pf.exibir();

        }

    } else if (tipo == 2) {

        List<PessoaJuridica> pessoas = pjDao.getPessoas();

        for (PessoaJuridica pj : pessoas) {

            pj.exibir();

        }

    } else {

        System.out.println("Tipo inválido.");

    }

}

}
```



**g. Qualquer exceção que possa ocorrer durante a execução do sistema deverá ser tratada.**

**h. Selecionada a opção sair, finalizar a execução do sistema.**

```
int opcao;

do {

    System.out.println("Menu:");

    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.print("Escolha uma opção: ");

    opcao = Integer.parseInt(scanner.nextLine());

    switch (opcao) {

        case 1:

            incluir(scanner, pfDao, pjDao);

            break;

        case 2:

            alterar(scanner, pfDao, pjDao);

            break;

        case 3:

            excluir(scanner, pfDao, pjDao);

            break;

        case 4:

            exibirPeloid(scanner, pfDao, pjDao);

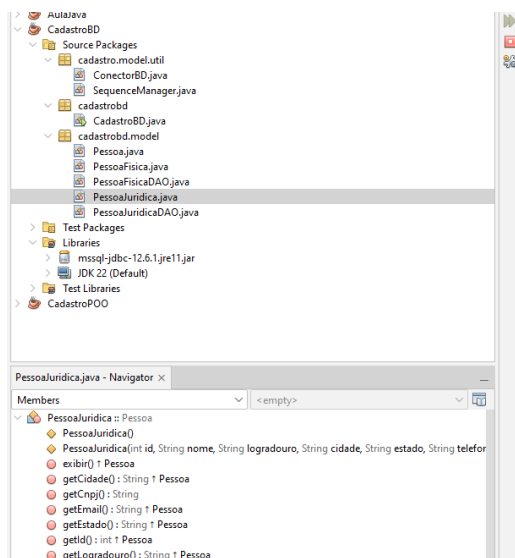
            break;
```

```
        case 5:
            exibirTodos(scanner, pfDao, pjDao);
            break;
        case 0:
            System.out.println("Saindo...");
            break;
        default:
            System.out.println("Opção inválida.");
            break;
    }
} while (opcao != 0);

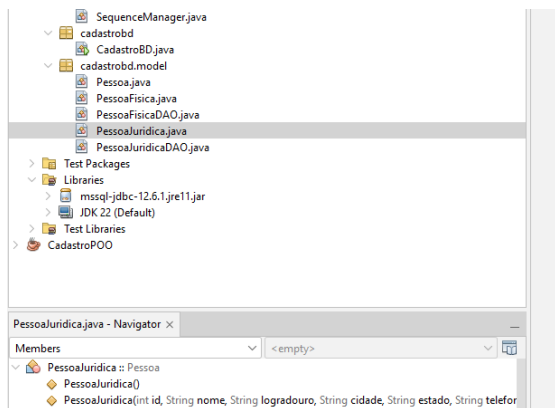
scanner.close();
}
```

## 2) Testar as funcionalidades do sistema:

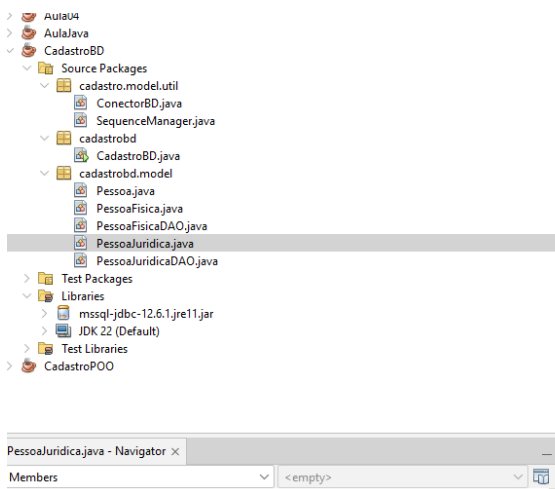
- a. Efetuar as diversas operações disponibilizadas, tanto para pessoa jurídica quanto para pessoa física.



```
4. Exibir pelo ID
5. Exibir todos
0. Sair
Escolha uma opção: 4
Tipo (1. Física / 2. Jurídica): 1
ID: 4
ID: 4
Nome: Breno
Logradouro: Rua rua
Cidade: Brumadinho
Estado: MG
Telefone: 319874562
Email: teste@teste.com
CPF: 12345678910
```



```
Escolha uma opção: 1
Tipo (1. Física / 2. Jurídica): 2
Nome: Junin techpar
Logradouro: rua tão
Cidade: Belo Horizonte
Estado: MG
Telefone: 31985476552
Email: junintechpar@teste.com
CNPJ: 1245800019654
```



```
run:
Menu:
1. Incluir
2. Alterar
3. Excluir
4. Exibir pelo ID
5. Exibir todos
0. Sair
Escolha uma opção: 3
Tipo (1. Física / 2. Jurídica): 1
ID: 4
```

- b. Feitas as operações, verificar os dados no SQL Server, com a utilização da aba Services, divisão Databases, do NetBeans, ou através do SQL Server Management Studio.

The screenshot shows the NetBeans IDE with the 'Pesquisador de Objetos' (Object Explorer) on the left and a SQL query window on the right. The query window displays the following SQL query:

```
SELECT * FROM pessoa
```

The query results are shown in a table with the following columns: idpessoa, nome, logradouro, cidade, estado, telefone, and email. The results are as follows:

idpessoa	nome	logradouro	cidade	estado	telefone	email
1	Joao	Rua 12, casa 3, Quitanda	Riacho do Sul	PA	1111-1111	joao@riacho.com
2	JJC	Rua 11, Centro	Riacho do Norte	PA	1212-1212	jjc@riacho.com
3	Junin techpar	rua táos	Belo Horizonte	MG	31985476552	junintechpar@teste.com

### 3) Ajustar as características para obter uma execução como a seguinte:

The screenshot shows the NetBeans IDE with the 'Output - CadastroBD (run)' window open. The output displays the following information:

```
Email: teste@teste.com
CPF: 12345678910
Menu:
1. Incluir
2. Alterar
3. Excluir
4. Exibir pelo ID
5. Exibir todos
0. Sair
Escolha uma opção: 5
Tipo (1. Física / 2. Jurídica): 1
ID: 1
Nome: Joao
Logradouro: Rua 12, casa 3, Quitanda
Cidade: Riacho do Sul
Estado: PA
Telefone: 1111-1111
Email: joao@riacho.com
CPF: 11111111111
ID: 4
```

## **Relatório discente de acompanhamento**

### **1) Título da Prática;**

2º Procedimento | Alimentando a Base

### **2) Objetivo da Prática;**

Criar sistemas cadastrais com persistência em banco relacional.  
Server na persistência de dados.

### **4) Todos os códigos solicitados neste roteiro de aula;**

### **5) Os resultados da execução dos códigos também devem ser apresentados;**

### **6) Análise e Conclusão:**

- a. Quais as diferenças entre a persistência em arquivo e a persistência em banco de dados?
- b. Como o uso de operador lambda simplificou a impressão dos valores contidos nas entidades, nas versões mais recentes do Java?
- c. Por que métodos acionados diretamente pelo método main, sem o uso de um objeto, precisam ser marcados como static?

## **Endereço do projeto no GITHUB**

[https://github.com/BrenoSouza2023/Missao-Pratica\\_N-vel-3-Mundo-3.git](https://github.com/BrenoSouza2023/Missao-Pratica_N-vel-3-Mundo-3.git)