**Universidade Estácio de Sá**

**Polo Estácio EAD - Batatais / SP**

**Curso:** Desenvolvimento Full Stack  
**Disciplina:** BackEnd sem banco não tem

**Missão Prática -** Nível 3  
**Turma:** RPG0016  
**Semestre Letivo:** 2024.3  
 **Integrante:** Daiana Maira de Oliveira Lascala

**Título**

Criação de aplicativo Java, com acesso ao banco de dados SQL Server através do

middleware JDBC.

**Objetivo da Prática**

O objetivo desta prática foi implementar um sistema de cadastro de pessoas físicas e jurídicas em Java, utilizando um banco de dados SQL Server para persistência dos dados. O sistema permite realizar operações de inclusão, alteração, exclusão, consulta por ID e listagem completa de registros, com tratamento de exceções para garantir a robustez do programa. O projeto incluiu a implementação do padrão DAO para separação de responsabilidades e manipulação de dados, além da utilização do JDBC para conexão com o banco de dados.

**Descrição dos Códigos:**

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template

\*/

package cadastrobd;

import cadastrobd.model.PessoaFisica;

import cadastrobd.model.PessoaJuridica;

import cadastrobd.model.PessoaFisicaDAO;

import cadastrobd.model.PessoaJuridicaDAO;

import java.util.Scanner;

/\*\*

\*

\* @author Lascala

\*/

public class CadastroBDTeste {

public static void main(String[] args) {

try (Scanner scanner = new Scanner(System.in)) {

PessoaFisicaDAO pessoaFisicaDAO = new PessoaFisicaDAO();

PessoaJuridicaDAO pessoaJuridicaDAO = new PessoaJuridicaDAO();

int opcao;

do {

System.out.println("Selecione 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");

opcao = scanner.nextInt();

scanner.nextLine();

switch (opcao) {

case 1 -> {

System.out.println("Selecione o tipo:");

System.out.println("1 - Pessoa Física");

System.out.println("2 - Pessoa Jurídica");

int tipo = scanner.nextInt();

scanner.nextLine();

if (tipo == 1) {

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();

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

boolean sucessoInclusao = pessoaFisicaDAO.incluir(pessoaFisica);

System.out.println("Inclusão de Pessoa Física: " + (sucessoInclusao ? "Sucesso" : "Falha"));

} else if (tipo == 2) {

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("CNPJ: ");

String cnpj = scanner.nextLine();

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

boolean sucessoInclusaoPJ = pessoaJuridicaDAO.incluir(pessoaJuridica);

System.out.println("Inclusão de Pessoa Jurídica: " + (sucessoInclusaoPJ ? "Sucesso" : "Falha"));

}

}

case 2 -> {

System.out.println("Selecione o tipo:");

System.out.println("1 - Pessoa Física");

System.out.println("2 - Pessoa Jurídica");

int tipo = scanner.nextInt();

scanner.nextLine();

System.out.print("ID da pessoa a ser alterada: ");

int id = scanner.nextInt();

scanner.nextLine();

if (tipo == 1) {

PessoaFisica pessoaFisicaAtual = pessoaFisicaDAO.getPessoa(id);

if (pessoaFisicaAtual != null) {

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

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

String novoTelefone = scanner.nextLine();

pessoaFisicaAtual.setTelefone(novoTelefone);

boolean sucessoAlteracao = pessoaFisicaDAO.alterar(pessoaFisicaAtual);

System.out.println("Alteração de Pessoa Física: " + (sucessoAlteracao ? "Sucesso" : "Falha"));

} else {

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

}

} else if (tipo == 2) {

PessoaJuridica pessoaJuridicaAtual = pessoaJuridicaDAO.getPessoa(id);

if (pessoaJuridicaAtual != null) {

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

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

String novoTelefone = scanner.nextLine();

pessoaJuridicaAtual.setTelefone(novoTelefone);

boolean sucessoAlteracaoPJ = pessoaJuridicaDAO.alterar(pessoaJuridicaAtual);

System.out.println("Alteração de Pessoa Jurídica: " + (sucessoAlteracaoPJ ? "Sucesso" : "Falha"));

} else {

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

}

}

}

case 3 -> {

System.out.println("Selecione o tipo:");

System.out.println("1 - Pessoa Física");

System.out.println("2 - Pessoa Jurídica");

int tipo = scanner.nextInt();

scanner.nextLine();

System.out.print("ID da pessoa a ser excluída: ");

int id = scanner.nextInt();

scanner.nextLine();

boolean sucessoExclusao = false;

if (tipo == 1) {

sucessoExclusao = pessoaFisicaDAO.excluir(id);

} else if (tipo == 2) {

sucessoExclusao = pessoaJuridicaDAO.excluir(id);

}

System.out.println("Exclusão: " + (sucessoExclusao ? "Sucesso" : "Falha"));

}

case 4 -> {

System.out.println("Selecione o tipo:");

System.out.println("1 - Pessoa Física");

System.out.println("2 - Pessoa Jurídica");

int tipo = scanner.nextInt();

scanner.nextLine();

System.out.print("ID da pessoa a ser exibida: ");

int id = scanner.nextInt();

scanner.nextLine();

if (tipo == 1) {

PessoaFisica pessoaFisicaExibida = pessoaFisicaDAO.getPessoa(id);

System.out.println("Pessoa Física: " + (pessoaFisicaExibida != null ? pessoaFisicaExibida : "Não encontrada."));

} else if (tipo == 2) {

PessoaJuridica pessoaJuridicaExibida = pessoaJuridicaDAO.getPessoa(id);

System.out.println("Pessoa Jurídica: " + (pessoaJuridicaExibida != null ? pessoaJuridicaExibida : "Não encontrada."));

}

}

case 5 -> {

System.out.println("Selecione o tipo:");

System.out.println("1 - Pessoa Física");

System.out.println("2 - Pessoa Jurídica");

int tipo = scanner.nextInt();

scanner.nextLine();

if (tipo == 1) {

System.out.println("Lista de Pessoas Físicas:");

for (PessoaFisica pf : pessoaFisicaDAO.getPessoas()) {

System.out.println(pf);

}

} else if (tipo == 2) {

System.out.println("Lista de Pessoas Jurídicas:");

for (PessoaJuridica pj : pessoaJuridicaDAO.getPessoas()) {

System.out.println(pj);

}

}

}

case 0 -> System.out.println("Saindo...");

default -> System.out.println("Opção inválida. Tente novamente.");

}

} while (opcao != 0);

}

}

}

**Os resultados da execução dos códigos**

Selecione uma opção:

1 - Incluir

2 - Alterar

3 - Excluir

4 - Exibir pelo ID

5 - Exibir todos

0 - Sair

1

Selecione o tipo:

1 - Pessoa Física

2 - Pessoa Jurídica

1

Nome: João da Silva

Logradouro: Rua A, 123

Cidade: São Paulo

Estado: SP

Telefone: 99999-9999

Email: joao@example.com

CPF: 123.456.789-00

Inclusão de Pessoa Física: Sucesso

**Análise e Conclusão:**

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

A persistência em arquivo envolve o armazenamento de dados em formatos como texto ou binário, o que pode dificultar operações complexas e consultas eficientes. Já a persistência em banco de dados oferece suporte a operações avançadas, como transações e consultas otimizadas, além de possibilitar a manipulação de grandes volumes de dados de forma estruturada.

**Uso de Operador Lambda para Impressão**

Nas versões mais recentes do Java, o uso de expressões lambda simplifica a execução de operações sobre coleções. Por exemplo, ao usar um forEach com lambda para imprimir os dados de uma lista, o código se torna mais conciso e legível.

**Métodos static no main**

Os métodos acionados diretamente pelo método main precisam ser marcados como static porque o main é um método estático. Métodos estáticos não podem acessar membros de instância diretamente, então quaisquer métodos chamados a partir de main também devem ser estáticos para serem acessíveis.

**Repositório Git:**

https://github.com/DaianaLascala/Missao-3---BackEnd-sem-banco-n-o-tem