



Estácio

Universidade Estácio

Juazeiro da Bahia

Desenvolvimento Full Stack

BackEnd sem banco não tem

Relatório discente de acompanhamento

Nome: Gerson José de Almeida Júnior

Número da Turma: 23.3

Semestre Letivo: 3º Semestre

Github: <https://github.com/gersonjose9713/Missao-Pratica-Nivel-3-Mundo-3/>

Mapeamento Objeto-Relacional e DAO

Códigos Desenvolvidos:

```
1 import cadastrobd.model.PessoaFisica;
2 import cadastrobd.model.PessoaFisicaDAO;
3 import cadastrobd.model.PessoaJuridica;
4 import cadastrobd.model.util.ConectorBD;
5 import cadastrobd.model.PessoaJuridicaDAO;
6 import java.sql.Connection;
7 import java.sql.SQLException;
8 import java.util.List;
9
10 new *
11 public class CadastroBDTeste {
12     new *
13     public static void main(String[] args) {
14         Connection connection = null;
15         try {
16             connection = ConectorBD.getConnection();
17             PessoaFisicaDAO pessoaFisicaDAO = new PessoaFisicaDAO(connection);
18             PessoaJuridicaDAO pessoaJuridicaDAO = new PessoaJuridicaDAO(connection);
19
20             PessoaFisica novaPessoaFisica = new PessoaFisica( idPessoa: 0, nome: "João", logradouro: "Rua Doutor Vieira", cidade: "Manaus", estado: "AM", telefone: "9226801129", email: "erick@123.com.br", cpf: "64373442100");
21             pessoaFisicaDAO.incluir(novaPessoaFisica);
22             System.out.println("Pessoa fisica incluida com sucesso.");
23
24             int idPessoaFisicaAlterar = 4;
25             PessoaFisica pessoaFisicaAlterar = pessoaFisicaDAO.getPessoa(idPessoaFisicaAlterar);
26             if (pessoaFisicaAlterar != null) {
27                 pessoaFisicaAlterar.setNome("Erick");
28                 pessoaFisicaAlterar.setLogradouro("Rua Doutor Vieira");
29                 pessoaFisicaAlterar.setCidade("Manaus");
30                 pessoaFisicaAlterar.setEstado("AM");
31                 pessoaFisicaAlterar.setTelefone("9226801129");
32                 pessoaFisicaAlterar.setEmail("erick@123.com.br");
33                 pessoaFisicaAlterar.setCpf("64373442100");
34                 pessoaFisicaDAO.alterar(pessoaFisicaAlterar);
35                 System.out.println("Pessoa fisica alterada com sucesso.");
36             }
37         } catch (SQLException e) {
38             e.printStackTrace();
39         } finally {
40             if (connection != null) {
41                 connection.close();
42             }
43         }
44     }
45 }
```

IDEA 2023.1.4

Project: CadastroBD

File: CadastroBDTeste.java

```
1 import cadastrobd.model.PessoaFisica;
2 import cadastrobd.model.PessoaFisicaDAO;
3 import cadastrobd.model.PessoaJuridica;
4 import cadastrobd.model.util.ConectorBD;
5 import cadastrobd.model.PessoaJuridicaDAO;
6 import java.sql.Connection;
7 import java.sql.SQLException;
8 import java.util.List;
9
10 new *
11 public class CadastroBDTeste {
12     new *
13     public static void main(String[] args) {
14         Connection connection = null;
15         try {
16             connection = ConectorBD.getConnection();
17             PessoaFisicaDAO pessoaFisicaDAO = new PessoaFisicaDAO(connection);
18             PessoaJuridicaDAO pessoaJuridicaDAO = new PessoaJuridicaDAO(connection);
```

Run: CadastroBDTeste

C:\Java\jdk-20\bin\java.exe -javaagent:C:\JetBrains\IntelliJ IDEA 2023.1.4\lib\idea_rt.jar=55367:C:\JetBrains\IntelliJ IDEA 2023.1.4\bin -Dfile.encoding=UTF-8 -Dsun

Pessoa fisica incluida com sucesso.

Pessoa fisica não encontrada para alteração.

Todas as pessoas fisicas:

ID: 17

Nome: João

Logradouro: Rua Ísis

Cidade: Cidade de Deus

Estado: AM

Telefone: 9236062646

Email: João@123

CPF: 71394575653

Pessoa fisica excluida com sucesso.

Pessoa juridica incluida com sucesso.

IDEA 2023.1.4

Project: CadastroBD

File: CadastroBDTeste.java

Database: LojaDB > Loja > dbo > tables > Pessoas

ID	Nome	Logradouro	Cidade	Estado	Telefone	Email
17	João	Rua Ísis	Cidade de Deus	AM	9236062646	João@123
18	Pietra e Andreia Telas ME	Rua Oswaldo de Souza	São Vicente	SP	1328489102	seguranca@pietraeandreiatelasme.co...

IDE interface showing a project named "CadastroBD" with a database connection to "LojaDB". The project structure includes:

- CadastroBD
 - .idea
 - out
 - src
 - cadastro.model.util
 - ConectorBD
 - SequenceManager
 - cadastrobd.model
 - Pessoa
 - PessoaFisica
 - PessoaFisicaDAO
 - PessoaJuridica
 - PessoaJuridicaDAO
 - CadastroBDTeste
 - Main
 - .gitignore
 - CadastroBD.iml
 - mssql-jdbc-12.4.0.jre11.jar lib
 - External Libraries
 - Scratches and Consoles

The database view shows the "PessoaJuridica" table with the following data:

PessoaID	CNPJ
18	87272284000153

Database: LojaDB > Loja > dbo > tables > PessoaJuridica

IDE interface showing the same project "CadastroBD" with a database connection to "LojaDB". The project structure is identical to the previous screenshot.

The database view shows the "PessoaFisica" table with the following data:

PessoaID	CPF
17	71394575653

Database: LojaDB > Loja > dbo > tables > PessoaFisica

Análise e Conclusão:

Qual a importância dos componentes de middleware, como o JDBC?

Os componentes de middleware desempenham um papel crucial na conectividade e na interação entre aplicativos e bancos de dados. O JDBC (Java Database Connectivity) é um middleware que permite que aplicativos Java se comuniquem com bancos de dados relacionais. Sua importância está relacionada a vários fatores:

Conexão com Bancos de Dados: Fornece uma API padrão que permite a conexão com diferentes sistemas de gerenciamento de bancos de dados.

Portabilidade: Torna os aplicativos Java portáveis, uma vez que você pode mudar de banco de dados subjacente sem precisar reescrever todo o código.

Segurança: Pode ser configurado para fornecer recursos de segurança, como autenticação e controle de acesso aos dados.

Desempenho: Oferece controle sobre consultas e transações, permitindo otimizações de desempenho.

Qual a diferença no uso de Statement ou PreparedStatement para a manipulação de dados?

Statement e PreparedStatement são duas maneiras de executar consultas SQL em um banco de dados usando o JDBC:

Statement: As consultas SQL são incorporadas diretamente no código Java. Isso pode ser vulnerável a ataques de injeção de SQL e é menos eficiente, pois a consulta é reanalisada toda vez que é executada.

PreparedStatement: As consultas SQL são pré-compiladas e os parâmetros são vinculados às consultas. Isso melhora a segurança, pois previne ataques de injeção de SQL, e também melhora o desempenho, pois a consulta é compilada apenas uma vez.

Como o padrão DAO melhora a manutenibilidade do software?

O padrão DAO (Data Access Object) é um padrão de projeto que isola a lógica de acesso a dados do restante do código do aplicativo. Isso melhora a manutenibilidade do software de várias maneiras:

Separação de Responsabilidades: Separa a lógica de acesso a dados do restante da aplicação, tornando o código mais organizado e fácil de entender.

Facilita a Troca de Fontes de Dados: É mais fácil trocar a fonte de dados (por exemplo, de um banco de dados relacional para um serviço web) sem afetar o restante do código.

Testabilidade: Permite a criação de testes unitários mais facilmente, pois a lógica de acesso a dados pode ser substituída por implementações de teste.

Reutilização de Código: Promove a reutilização de código, uma vez que a lógica de acesso a dados pode ser compartilhada entre diferentes partes do aplicativo.

Como a herança é refletida no banco de dados, quando lidamos com um modelo estritamente relacional?

Em um modelo de banco de dados estritamente relacional, a herança não é diretamente suportada como em linguagens de programação orientadas a objetos. No entanto, existem várias abordagens para representar a herança em bancos de dados relacionais:

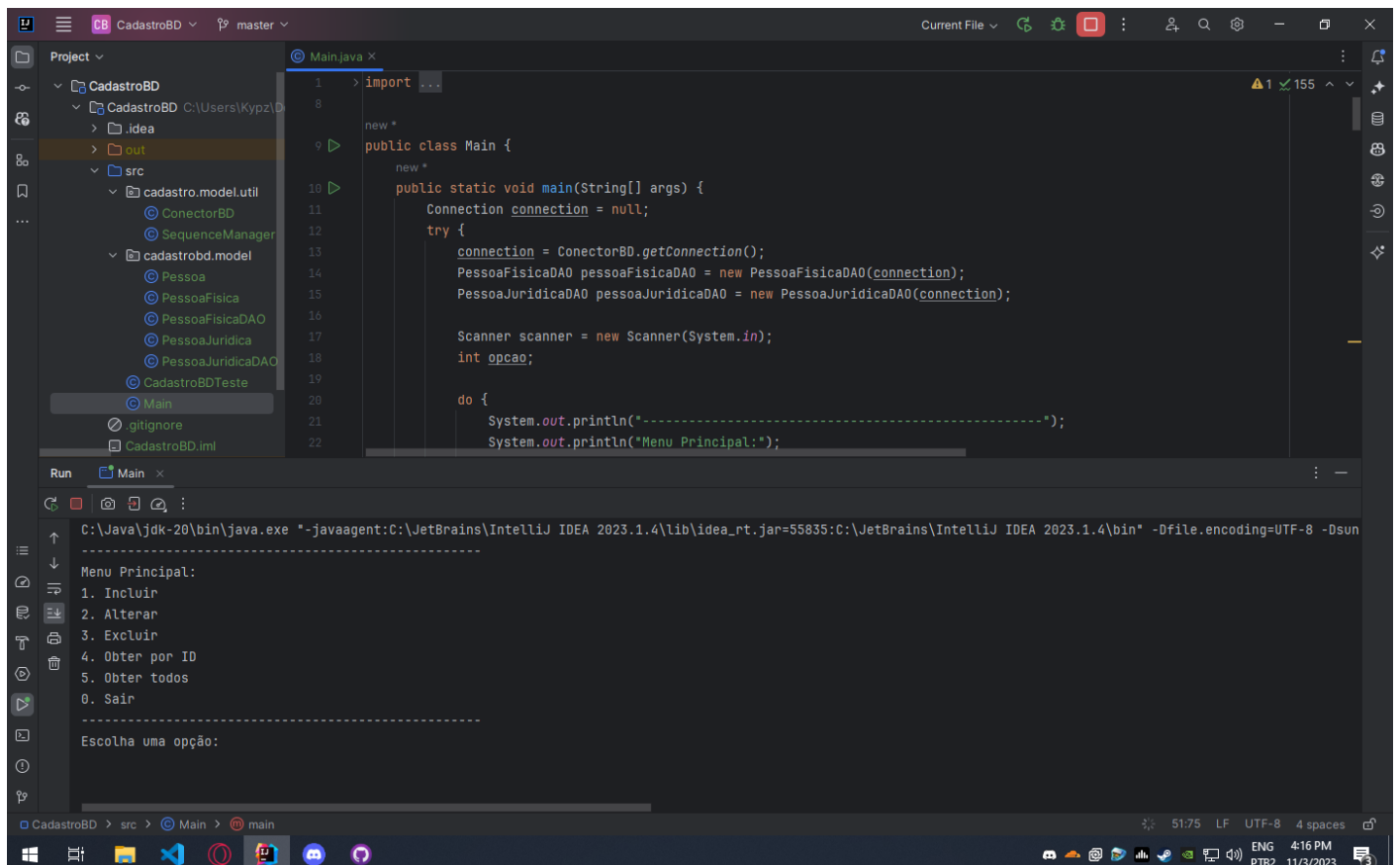
Tabela por Classe (Single Table Inheritance): Uma abordagem comum é usar uma única tabela que armazena dados de todas as subclasses. Um campo discriminador (geralmente uma coluna) é usado para identificar a classe de cada registro.

Tabela por Subclasse (Class Table Inheritance): Cada classe é mapeada para uma tabela separada no banco de dados. Isso mantém uma estrutura mais semelhante à herança em linguagens orientadas a objetos, mas pode resultar em um esquema de banco de dados mais complexo.

Tabela por Associação (Concrete Table Inheritance): Cada tabela corresponde a uma classe concreta (não abstrata). É uma abordagem direta, mas pode resultar em duplicação de dados e complexidade.

Alimentando a Base

Códigos Desenvolvidos:



```
1  > import ...
2
3  new *
4
5  public class Main {
6
7      new *
8
9      public static void main(String[] args) {
10         Connection connection = null;
11         try {
12             connection = ConectorBD.getConnection();
13             PessoaFisicaDAO pessoaFisicaDAO = new PessoaFisicaDAO(connection);
14             PessoaJuridicaDAO pessoaJuridicaDAO = new PessoaJuridicaDAO(connection);
15
16             Scanner scanner = new Scanner(System.in);
17             int opcao;
18
19             do {
20                 System.out.println("-----");
21                 System.out.println("Menu Principal:");
22                 System.out.println("1. Incluir");
23                 System.out.println("2. Alterar");
24                 System.out.println("3. Excluir");
25                 System.out.println("4. Obter por ID");
26                 System.out.println("5. Obter todos");
27                 System.out.println("0. Sair");
28                 System.out.println("-----");
29                 System.out.println("Escolha uma opção:");
30                 opcao = scanner.nextInt();
31             } while (opcao != 0);
32         } catch (SQLException e) {
33             e.printStackTrace();
34         }
35     }
36 }
```

Run Main

```
C:\Java\jdk-20\bin\java.exe -javaagent:C:\JetBrains\IntelliJ IDEA 2023.1.4\lib\idea_rt.jar=55835:C:\JetBrains\IntelliJ IDEA 2023.1.4\bin* -Dfile.encoding=UTF-8 -Dsun
-----
Menu Principal:
1. Incluir
2. Alterar
3. Excluir
4. Obter por ID
5. Obter todos
0. Sair
-----
Escolha uma opção:
```

Project

CadastroBD

CadastroBD

.idea

out

src

cadastro.model.util

ConectorBD

SequenceManager

cadastrobd.model

Pessoa

PessoaFisica

PessoaFisicaDAO

Main.java

```
1 > import ...
8
9 new *
10 public class Main {
11     new *
12     public static void main(String[] args) {
13         Connection connection = null;
14         try {
15             connection = ConectorBD.getConnection();
16             PessoaFisicaDAO pessoaFisicaDAO = new PessoaFisicaDAO(connection);
17             PessoaJuridicaDAO pessoaJuridicaDAO = new PessoaJuridicaDAO(connection);
18         }
19     }
20 }
```

Run

Main

Escolha uma opção: 1

Escolha:

1. Pessoa Fisica

2. Pessoa Juridica

1

Digite o nome: João

Digite o logradouro: Rua 3

Digite a cidade: São Paulo

Digite o estado: SP

Digite o telefone: 123654789

Digite o email: João@123

Digite o CPF: 12345678125

Pessoa fisica incluida com sucesso.

Menu Principal:

CadastroBD

CadastroBD

.idea

out

src

cadastro.model.util

ConectorBD

SequenceManager

cadastrobd.model

Pessoa

PessoaFisica

PessoaFisicaDAO

PessoaJuridica

PessoaJuridicaDAO

CadastroBDTeste

Main

.gitignore

CadastroBD.iml

Main.java

Pessoas

PessoaFisica

3 rows

Tx: Auto

DDL

WHERE

ORDER BY

ID	Nome	Logradouro	Cidade	Estado	Telefone	Email
17	João	Rua Ísis	Cidade de Deus	AM	9236062646	João@123
18	Pietra e Andreia Telas ME	Rua Oswaldo de Souza	São Vicente	SP	1328489102	seguranca@pietraeandreiataelasmae.co...
19	João	Rua 3	São Paulo	SP	123654789	João@123

Database

LojaDB

Loja

dbo

tables

Pessoas

SUM: Not enough values

IDE interface showing a project named "CadastroBD" with a table view of "Pessoas". The table displays columns: ID, Nome, Logradouro, Cidade, Estado, Telefone, and Email. The data is filtered by "WHERE" and "ORDER BY".

ID	Nome	Logradouro	Cidade	Estado	Telefone	Email
17	João	Rua Isis	Cidade de Deus	AM	9236062646	João@123
18	Pietra e Andreia Telas ME	Rua Oswaldo de Souza	São Vicente	SP	1328489102	seguranca@pietraeandreiataelasme.co...
19	Pedro	Rua 4	Juazeiro	BA	654897215	Pedro@123

Database: LojaDB > Loja > dbo > tables > Pessoas

IDE interface showing a project named "CadastroBD" with a table view of "PessoaFisica". The table displays columns: PessoaID and CPF. The data is filtered by "WHERE" and "ORDER BY".

PessoaID	CPF
19	64581589721
17	71394575653

Database: LojaDB > Loja > dbo > tables > PessoaFisica

IDEA

Project: CadastroBD

Files:

- CadastroBD
- idea
- out
- src
 - cadastro.model.util
 - ConectorBD
 - SequenceManager
 - cadastrobd.model
 - Pessoa
 - PessoaFisica
 - PessoaFisicaDAO

Main.java

```
1 > import ...
8
9 new *
10 public class Main {
11     new *
12     public static void main(String[] args) {
13         Connection connection = null;
14         try {
15             connection = ConectorBD.getConnection();
16             PessoaFisicaDAO pessoaFisicaDAO = new PessoaFisicaDAO(connection);
17             PessoaJuridicaDAO pessoaJuridicaDAO = new PessoaJuridicaDAO(connection);
18         }
19     }
20 }
```

Run: Main

Output:

```
-----
Menu Principal:
1. Incluir
2. Alterar
3. Excluir
4. Obter por ID
5. Obter todos
0. Sair
-----
Escolha uma opção: 3
-----
Escolha:
1. Pessoa Fisica
2. Pessoa Juridica
1
Digite o ID da pessoa a ser excluida: 19
-----
Pessoa fisica excluida com sucesso.
-----
Menu Principal:
```

51:75 LF UTF-8 4 spaces

ENG 4:23 PM PTB2 11/3/2023

IDEA

Project: CadastroBD

Files:

- CadastroBD
- idea
- out
- src
 - cadastro.model.util
 - ConectorBD
 - SequenceManager
 - cadastrobd.model
 - Pessoa
 - PessoaFisica
 - PessoaFisicaDAO
 - PessoaJuridica
 - PessoaJuridicaDAO
 - CadastroBDTeste
 - Main

Database: LojaDB > Loja > dbo > tables > Pessoas

ID	Nome	Logradouro	Cidade	Estado	Telefone	Email
17	João	Rua Ísis	Cidade de Deus	AM	9236062646	João@123
18	Pietra e Andreia Telas ME	Rua Oswaldo de Souza	São Vicente	SP	1328489102	seguranca@pietraeandreiabelasme.co...

Database: LojaDB > Loja > dbo > tables > Pessoas

ENG 4:23 PM PTB2 11/3/2023

IDEA

Project: CadastroBD

Database: LojaDB > Loja > dbo > tables > PessoaFisica

	PessoaID	CPF
1	17	71394575653

Database: LojaDB > Loja > dbo > tables > PessoaFisica

IDEA

Project: CadastroBD

Database: LojaDB > Loja > dbo > tables > PessoaFisica

```
System.out.println("-----");
System.out.println("Pessoa fisica incluida com sucesso.");
} else if (incluir == 2) {
    PessoaJuridica novaPessoaJuridica = lerPessoaJuridica(scanner);
    pessoaJuridicaDAO.incluir(novaPessoaJuridica);
    System.out.println("-----");
    System.out.println("Pessoa juridica incluida com sucesso.");
} else {
    System.out.println("-----");
    System.out.println("Opção inválida. Tente novamente.");
}
break;
```

Run: Main

0. Sair

Escolha uma opção: 4

Escolha:

1. Pessoa Fisica

2. Pessoa Juridica

1

Digite o ID da pessoa a ser obtida: 17

ID: 17

Nome: João

Logradouro: Rua Ísis

Cidade: Cidade de Deus

Estado: AM

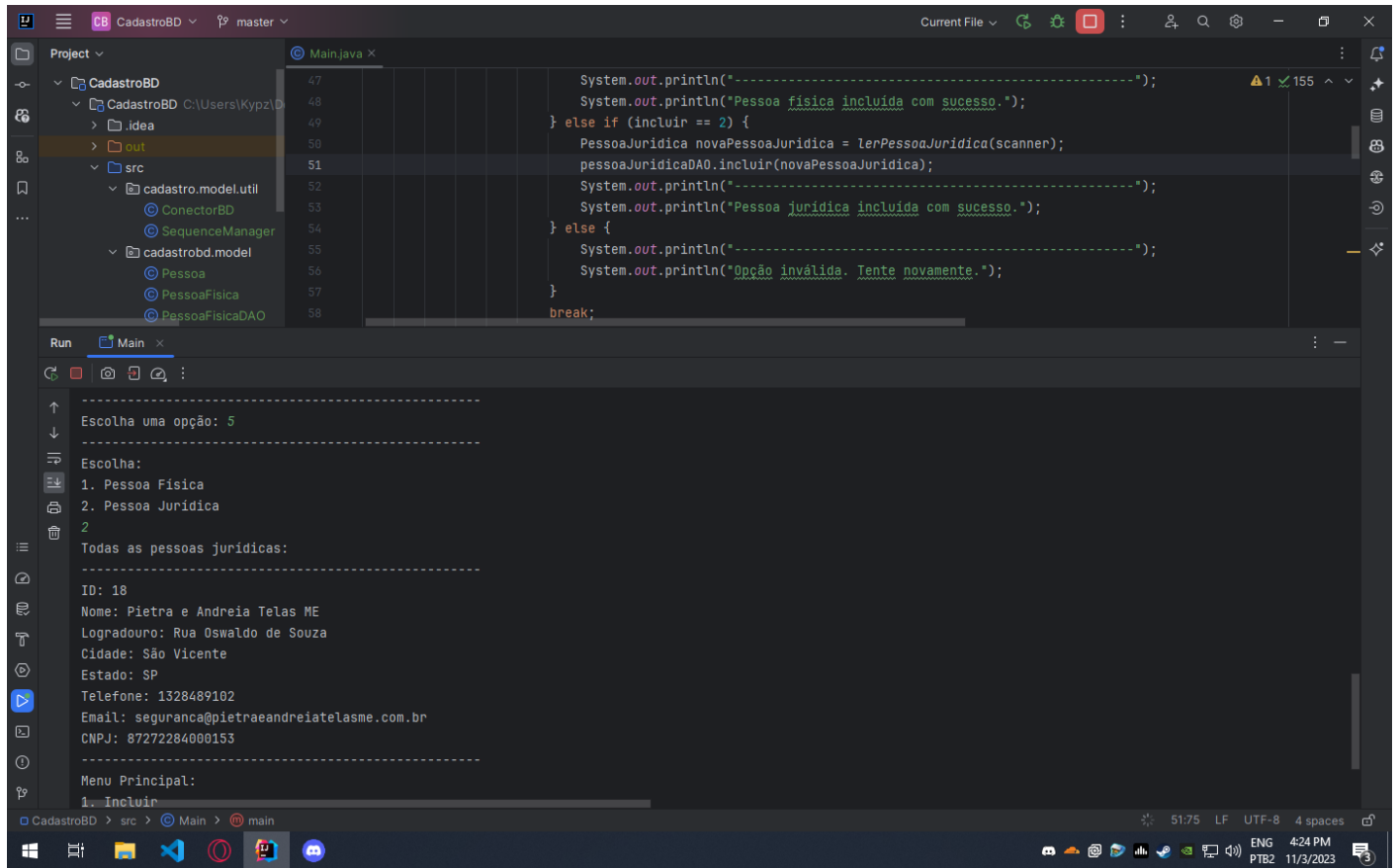
Telefone: 9236062646

Email: João@123

CPF: 71394575653

Menu Principal:

1. Incluir



```
47 System.out.println("-----");
48 System.out.println("Pessoa fisica incluida com sucesso.");
49 } else if (incluir == 2) {
50 PessoaJuridica novaPessoaJuridica = lerPessoaJuridica(scanner);
51 pessoaJuridicaDAO.incluir(novaPessoaJuridica);
52 System.out.println("-----");
53 System.out.println("Pessoa juridica incluida com sucesso.");
54 } else {
55 System.out.println("-----");
56 System.out.println("Opção inválida. Tente novamente.");
57 }
58 break;
```

Run Main x

Escolha uma opção: 5

Escolha:

1. Pessoa Fisica
2. Pessoa Juridica

2

Todas as pessoas juridicas:

ID: 18

Nome: Pietra e Andreia Telas ME

Logradouro: Rua Oswaldo de Souza

Cidade: São Vicente

Estado: SP

Telefone: 1328489102

Email: seguranca@pietraeandreiaelasm.com.br

CNPJ: 87272284000153

Menu Principal:

1. Incluir

Análise e Conclusão:

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

Persistência em Arquivo:

A persistência em arquivo envolve o armazenamento de dados em arquivos no sistema de arquivos do computador.

É mais adequada para armazenar dados simples, como configurações, logs e pequenas quantidades de informações.

Não oferece recursos de consulta avançados e relacionamentos entre dados.

Geralmente é menos escalável do que um banco de dados, pois o desempenho pode diminuir à medida que os arquivos crescem.

Persistência em Banco de Dados:

A persistência em banco de dados envolve o armazenamento de dados em sistemas de gerenciamento de banco de dados (DBMS), como MySQL, PostgreSQL, Oracle, etc.

É apropriada para armazenar grandes volumes de dados e suporta consultas complexas, transações e relacionamentos entre tabelas.

Oferece recursos de segurança, controle de concorrência e escalabilidade.

É mais apropriada para aplicativos que exigem manipulação de dados sofisticada e alta disponibilidade.

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

O uso de operadores lambda e a API de streams tornam a iteração e a impressão de valores mais concisas e legíveis, reduzindo a necessidade de escrever loops explícitos

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

A palavra-chave static em um método indica que ele pertence à classe em vez de a instâncias individuais da classe. Como o método main é chamado pelo sistema antes de criar qualquer instância do programa, ele deve ser static para garantir que possa ser invocado diretamente pela classe, sem a necessidade de criar um objeto.