Estácio	**UNIVERSIDADE ESTACIO DE SÁ** POLO PARANGABA – FORTALEZA/CE DESENVOLVIMENTO FULL STACK - 22.3 Relatório da Missão Prática | Nível 3 | Mundo 3
Aluno:	Jefferson Ponte Pessoa
Professor:	Simone Ingrid Monteiro Gama
Repositório:	https://github.com/jeffersonkako/M3-Nivel3

Título da Prática: 1º Procedimento | Mapeamento Objeto-Relacional e DAO

Objetivos da Prática:

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

Códigos:

Todos os códigos estão no repositório abaixo:

https://github.com/jeffersonkako/M3-Nivel3

Resultados PT1:

```
Run    CadastroBDTeste
/Library/Java/JavaVirtualMachines/openjdk.jdk/Contents/Home/bin/java -javaagent:/Users/jefferson/Applications/IntelliJ
Pessoa Física incluída com o ID: 8
Pedro Cardoso
Fernanda Souza
Teobaldo Capotino
Pessoa Jurídica incluída com o ID: 9
update Pessoa_Juridica  set cnpj = '00333444000122' where Pessoa_Juridica.id_pessoa  = null;
Nenhum registro encontrado para id_pessoa = null
COMPANY
Company X Ltda
Company X Ltda
Nenhum registro encontrado para excluir id_pessoa = null

Process finished with exit code 0
M3-Nivel3 > src > cadastrobd > CadastroBDTeste
```

Análise e Conclusão:

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

   Os componentes de middleware, como o JDBC (Java Database Connectivity), são essenciais porque atuam como intermediários entre diferentes sistemas de software, facilitando a comunicação e o intercâmbio de dados. No caso do JDBC, ele permite que aplicações Java se conectem e executem operações em bancos de dados de maneira padronizada, independente do sistema de gestão de banco de dados (SGBD) utilizado. Isso simplifica o desenvolvimento, aumenta a portabilidade das aplicações e promove a interoperabilidade entre sistemas heterogêneos.

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

A diferença principal entre Statement e PreparedStatement no Java JDBC está na performance e segurança:

1. PreparedStatement: É compilado pela base de dados antecipadamente, o que significa que ao utilizá-lo com parâmetros, ele pode ser reutilizado com diferentes valores, otimizando a performance. Além disso, oferece proteção contra injeção de SQL, pois os valores dos parâmetros são tratados de forma segura pelo driver JDBC.

2. Statement: É adequado para consultas que não necessitam de parâmetros e serão executadas uma única vez. Não é pré-compilado, o que pode ser menos eficiente para execuções repetidas, e não tem a mesma proteção contra injeção de SQL, pois a consulta é montada como uma única String, podendo ser vulnerável se os dados não forem adequadamente sanitizados.

PreparedStatement é geralmente recomendado para operações que incluem dados dinâmicos devido à segurança e performance, enquanto Statement pode ser usado para consultas estáticas simples.

3. Como o padrão DAO melhora a manutenibilidade do software?

O padrão DAO (Data Access Object) melhora a manutenibilidade do software separando a lógica de acesso a dados da lógica de negócio. Isso é feito encapsulando o código de acesso a dados em classes específicas (DAOs). As principais vantagens incluem:

Abstração e Encapsulamento: Ao centralizar o acesso aos dados, o DAO esconde os detalhes específicos do mecanismo de armazenamento ou da fonte de dados utilizada.

Facilidade de Mudança: Mudanças no esquema do banco de dados ou na tecnologia de persistência impactam apenas o DAO, não o restante do código.

Reusabilidade: O mesmo DAO pode ser usado em diferentes partes do sistema, evitando a duplicação de código.

Testabilidade: Com o acesso a dados isolado, é mais fácil mockar essas camadas para testar componentes de negócio de forma independente.

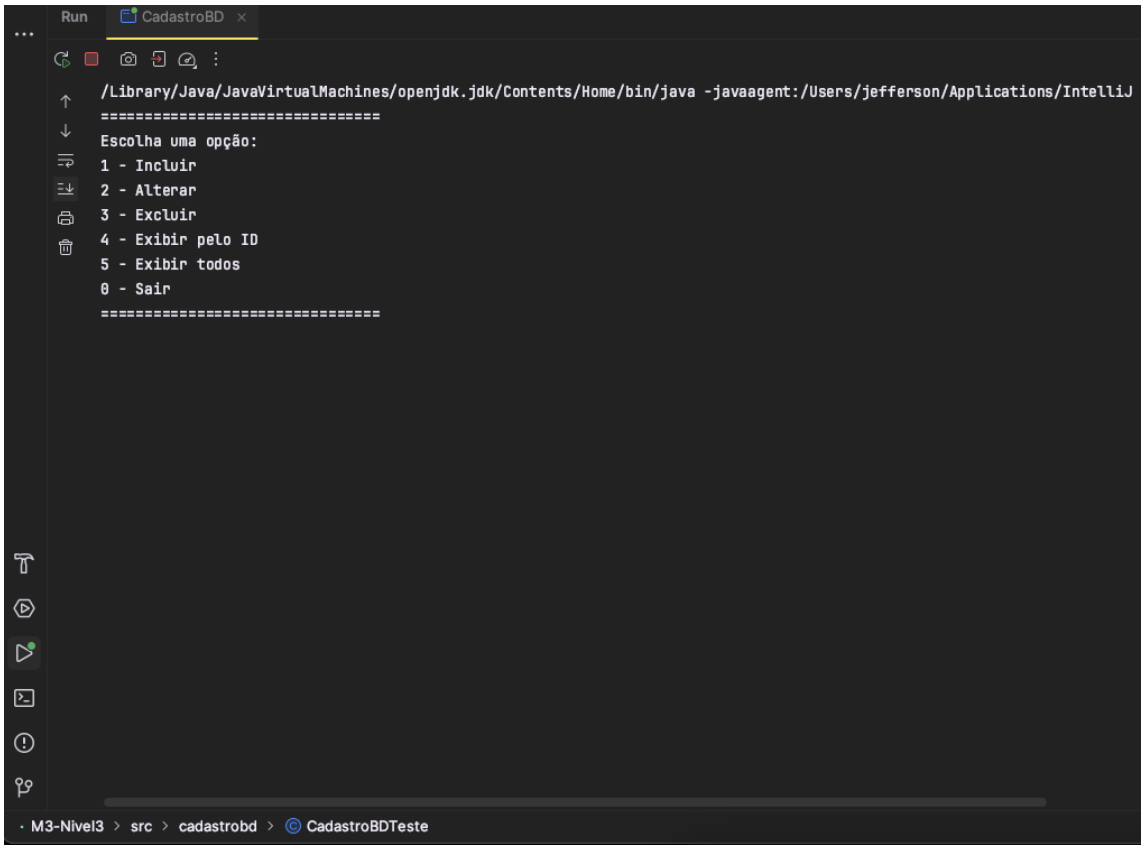
Organização: O código fica mais organizado com responsabilidades bem definidas, facilitando o entendimento e a manutenção do software.

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

No banco de dados relacional, a herança é emulada, pois não há suporte nativo como em linguagens de programação orientadas a objetos. Isso é geralmente feito usando três técnicas: a primeira, Tabela Única, onde uma única tabela contém as colunas para todas as propriedades de todas as classes na hierarquia; a segunda, Tabela por Classe, onde cada classe tem sua própria tabela contendo todas as suas propriedades, incluindo as herdadas; e a terceira, Tabela por Subclasse, onde a superclasse tem uma tabela e cada subclasse tem sua própria tabela contendo apenas as propriedades que não estão na superclasse. A escolha entre essas técnicas depende do equilíbrio desejado entre redundância de dados, performance de consultas e complexidade na manutenção do esquema do banco.

Título da Prática: 2º Procedimento | Alimentando a Base

Codigos:



```
Run    CadastroBD
================================
1
Escolha o tipo:
1 - Física
2 - Jurídica
1
Digite o nome:
jefferson teste
Digite o CPF:
03744455599
Digite o endereço:
Rua teste
Digite a cidade:
Fortaleza
Digite o estado:
CE
Digite o telefone:
85988991111
Digite o email:
jefferson@gmail.com
================================
Escolha uma opção:
1 - Incluir
2 - Alterar
3 - Excluir
4 - Exibir pelo ID
5 - Exibir todos
0 - Sair
================================
M3-Nivel3 > src > cadastrobd > CadastroBDTeste
```

DBeaver 23.2.4 - pessoa

master | dbo@master | *<master> Script-8 | *<master> Script-9 | pessoa | pessoa_fisica | pessoa_juridica | produto | usuario | *<master> Script-10

Propriedades | Dados | ER Diagrama

Insira uma expressão SQL para filtrar os resultados (use Ctrl+Espaço)

	id_pessoa	nome	endereco	cidade	estado	telefone	email
1	1	Pedro Cardoso	Avenida Brasil	Rio de Janeiro	RJ	21988887777	pedro.cardoso@email.com
2	2	Fernanda Souza	Travessa Joaquim	Salvador	BA	71977776666	fernanda.souza@email.com
3	3	COMPANY	Rua das Laranjeiras	Curitiba	PR	41966665555	company@email.com
4	5	Company X Ltda	Av Santos Dumont, 100	Fortaleza	CE	8532215200	empresa@empresa.com
5	9	Company X Ltda	Av Santos Dumont, 100	Fortaleza	CE	8532215200	empresa@empresa.com
6	11	jefferson teste	Rua teste	Fortaleza	CE	85988991111	jefferson@gmail.com

Valor: 11

```
Run    CadastroBD ×
2 - Alterar
3 - Excluir
4 - Exibir pelo ID
5 - Exibir todos
0 - Sair
================================
2
Escolha o tipo:
1 - Física
2 - Jurídica
1
Digite o ID da pessoa:
11
Digite o novo nome:
Jefferson Pessoa
Digite o novo CPF:
11122233344
Digite o novo endereço:
Rua teste 2
Digite a nova cidade:
Sao paulo
Digite o novo estado:
SP
Digite o novo telefone:
1188998811
Digite o novo email:
jefferson2@gmail.com
================================
Escolha uma opção:
1 - Incluir
2 - Alterar
3 - Excluir
4 - Exibir pelo ID
5 - Exibir todos
0 - Sair
================================
-Nivel3 > src > cadastrobd > CadastroBDTeste
```

master | dbo@master

*<master> Script-8 | *<master> Script-9 | pessoa | pessoa_fisica | pessoa_juridica | produto | usuario | *<master> Script-10

Propriedades | Dados | ER Diagrama | master | Banco de dados

pessoa | Insira uma expressão SQL para filtrar os resultados (use Ctrl+Espaço)

Grade | Texto

	id_pessoa	nome	endereco	cidade	estado	telefone	email
1	1	Pedro Cardoso	Avenida Brasil	Rio de Janeiro	RJ	21988887777	pedro.cardoso@email.com
2	2	Fernanda Souza	Travessa Joaquim	Salvador	BA	71977776666	fernanda.souza@email.com
3	3	COMPANY	Rua das Laranjeiras	Curitiba	PR	41966665555	company@email.com
4	5	Company X Ltda	Av Santos Dumont, 100	Fortaleza	CE	8532215200	empresa@empresa.com
5	9	Company X Ltda	Av Santos Dumont, 100	Fortaleza	CE	8532215200	empresa@empresa.com
6	11	Jefferson Pessoa	Rua teste 2	Sao paulo	SP	1188998811	jefferson2@gmail.com

Valor
11

Run | CadastroBD

```
2 - Alterar
3 - Excluir
4 - Exibir pelo ID
5 - Exibir todos
0 - Sair
================================
1
Escolha o tipo:
1 - Física
2 - Jurídica
2
Digite o nome:
empresa teste
Digite o CNPJ:
99229999992299
Digite o endereço:
rua empresa
Digite a cidade:
SP
Digite o estado:
SP
Digite o telefone:
11888888888
Digite o email:
company@gmail.com
================================
Escolha uma opção:
1 - Incluir
2 - Alterar
3 - Excluir
4 - Exibir pelo ID
5 - Exibir todos
0 - Sair
================================
```

M3-Nivel3 > src > cadastrobd > CadastroBDTeste

master | dbo@master

*<master> Script-8 | *<master> Script-9 | pessoa | pessoa_fisica | pessoa_juridica | produto | usuario | *<master> Script-10

Propriedades | Dados | ER Diagrama | master | Banco de dados

pessoa | Insira uma expressão SQL para filtrar os resultados (use Ctrl+Espaço)

Grade | Texto

	id_pessoa	nome	endereco	cidade	estado	telefone	email
1	1	Pedro Cardoso	Avenida Brasil	Rio de Janeiro	RJ	21988887777	pedro.cardoso@email.com
2	2	Fernanda Souza	Travessa Joaquim	Salvador	BA	71977776666	fernanda.souza@email.com
3	3	COMPANY	Rua das Laranjeiras	Curitiba	PR	41966665555	company@email.com
4	5	Company X Ltda	Av Santos Dumont, 100	Fortaleza	CE	8532215200	empresa@empresa.com
5	9	Company X Ltda	Av Santos Dumont, 100	Fortaleza	CE	8532215200	empresa@empresa.com
6	11	Jefferson Pessoa	Rua teste 2	Sao paulo	SP	1188998811	jefferson2@gmail.com
7	12	empresa teste	rua empresa	SP	SP	11888888888	company@gmail.com

Valor
11

*<master> Script-8 | *<master> Script-9 | pessoa | pessoa_fisica | pessoa_juridica | produto | usuario | *<master> Script-10

Propriedades | Dados | ER Diagrama

pessoa_juridica | Insira uma expressão SQL para filtrar os resultados (use Ctrl+Espaço)

	id_pessoa	cnpj
1	3	11223344000155
2	5	00333444000122
3	9	00333444000122
4	12	99229999992299

Valor

Edit value:

3

```
================================
Escolha uma opção:
1 - Incluir
2 - Alterar
3 - Excluir
4 - Exibir pelo ID
5 - Exibir todos
0 - Sair
================================
3
Escolha o tipo:
1 - Física
2 - Jurídica
1
Digite o ID da pessoa:
11
================================
Escolha uma opção:
1 - Incluir
2 - Alterar
3 - Excluir
4 - Exibir pelo ID
5 - Exibir todos
0 - Sair
================================
```

M3-Nivel3 > src > cadastrobd > CadastroBDTeste

*<master> Script-8 | *<master> Script-9 | pessoa | pessoa_fisica | pessoa_juridica | produto | usuario | *<master> Script-10

Propriedades | Dados | ER Diagrama

pessoa | Insira uma expressão SQL para filtrar os resultados (use Ctrl+Espaço)

	id_pessoa	nome	endereco	cidade	estado	telefone	email
1	1	Pedro Cardoso	Avenida Brasil	Rio de Janeiro	RJ	21988887777	pedro.cardoso@email.com
2	2	Fernanda Souza	Travessa Joaquim	Salvador	BA	71977776666	fernanda.souza@email.com
3	3	COMPANY	Rua das Laranjeiras	Curitiba	PR	41966665555	company@email.com
4	5	Company X Ltda	Av Santos Dumont, 100	Fortaleza	CE	8532215200	empresa@empresa.com
5	9	Company X Ltda	Av Santos Dumont, 100	Fortaleza	CE	8532215200	empresa@empresa.com
6	12	empresa teste	rua empresa	SP	SP	11888888888	company@gmail.com

Valor

12

```
================================
Escolha uma opção:
1 - Incluir
2 - Alterar
3 - Excluir
4 - Exibir pelo ID
5 - Exibir todos
0 - Sair
================================
4
Escolha o tipo:
1 - Física
2 - Jurídica
1
Digite o ID da pessoa:
1
id: 1
nome: Pedro Cardoso
endereco: Avenida Brasil
cidade: Rio de Janeiro
estado: RJ
telefone: 21988887777
email: pedro.cardoso@email.com
CPF: 11122233344
================================
```

```
Run    CadastroBD ×

4 - Exibir pelo ID
5 - Exibir todos
0 - Sair
================================
5
Escolha o tipo:
1 - Física
2 - Jurídica
1
id: 1
nome: Pedro Cardoso
endereco: Avenida Brasil
cidade: Rio de Janeiro
estado: RJ
telefone: 21988887777
email: pedro.cardoso@email.com
CPF: 11122233344
id: 2
nome: Fernanda Souza
endereco: Travessa Joaquim
cidade: Salvador
estado: BA
telefone: 71977776666
email: fernanda.souza@email.com
CPF: 44433322211
```

Análise e Conclusão:

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

   A persistência em arquivo envolve o armazenamento de dados em um sistema de arquivos em formatos como texto, binário, JSON, XML, etc. É uma forma simples e direta de salvar dados, mas não é tão escalável, segura ou eficiente para consultas complexas e manipulação de dados quando comparada à persistência em banco de dados.

   A persistência em banco de dados usa sistemas de gerenciamento de banco de dados (DBMS) para armazenar, recuperar e gerenciar dados de maneira estruturada, utilizando tabelas, índices, e suportando operações complexas e transações. Oferece recursos avançados como consistência, atomicidade, isolamento, durabilidade (ACID), segurança, backup, e otimizações para acessos simultâneos.

2. **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 no Java simplificou a impressão dos valores contidos nas entidades ao permitir que os desenvolvedores escrevam código mais conciso e legível, sem a necessidade de criar classes anônimas para operações simples. Com lambdas, é possível passar comportamento de forma direta e declarativa. Por exemplo, para imprimir os valores de uma lista, antes do Java 8, seria necessário criar um loop explícito. Com lambdas, isso pode ser feito de maneira simplificada usando métodos de stream e operações como forEach, assim:

   ```
   lista.forEach(elemento -> System.out.println(elemento));
   ```

   Esse código substitui várias linhas de um loop for ou for-each tradicional e torna o código mais expressivo e focado no que se deseja realizar (a ação de imprimir), e não no como fazer o loop (o controle de fluxo).

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

   Métodos acionados diretamente pelo método main precisam ser marcados como static porque o main é um método estático e, portanto, pertence à classe e não a uma instância específica da classe. Métodos estáticos só podem chamar diretamente outros métodos estáticos, já que eles são acessados sem criar um objeto da classe, ou seja, antes de qualquer instância da classe existir. Assim, para que o método main possa invocar outros métodos sem a necessidade de criar um objeto, esses outros métodos também devem ser estáticos.