Estácio	**UNIVERSIDADE ESTACIO DE SÁ** POLO INDAIATUBA – INDAIATUBA/SP DESENVOLVIMENTO FULL STACK - 22.3 Relatório da Missão Prática | Nível 3 | Mundo 3
Aluno:	Saulo Henrique dos Santos
Professor:	Simone Ingrid Monteiro Gama
Repositório:	https://github.com/SauloHenriqueSantos/Mundo03Nivel03

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

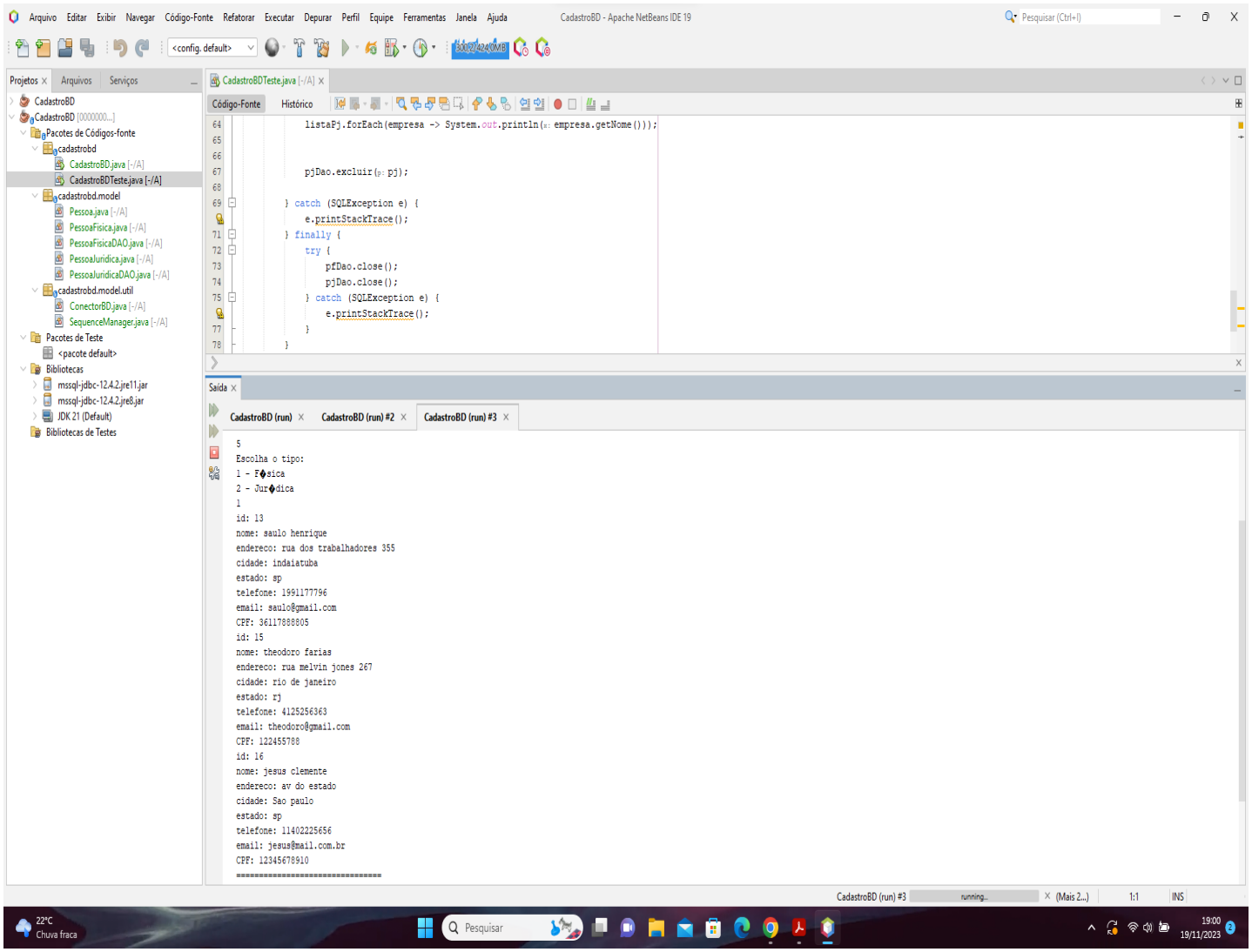
Objetivos da Prática:

- 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:

https://github.com/SauloHenriqueSantos/Mundo03Nivel03

Resultados PT1:



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 MEIO 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, simplificando o desenvolvimento, aumentando a portabilidade das aplicações e promovendo 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.

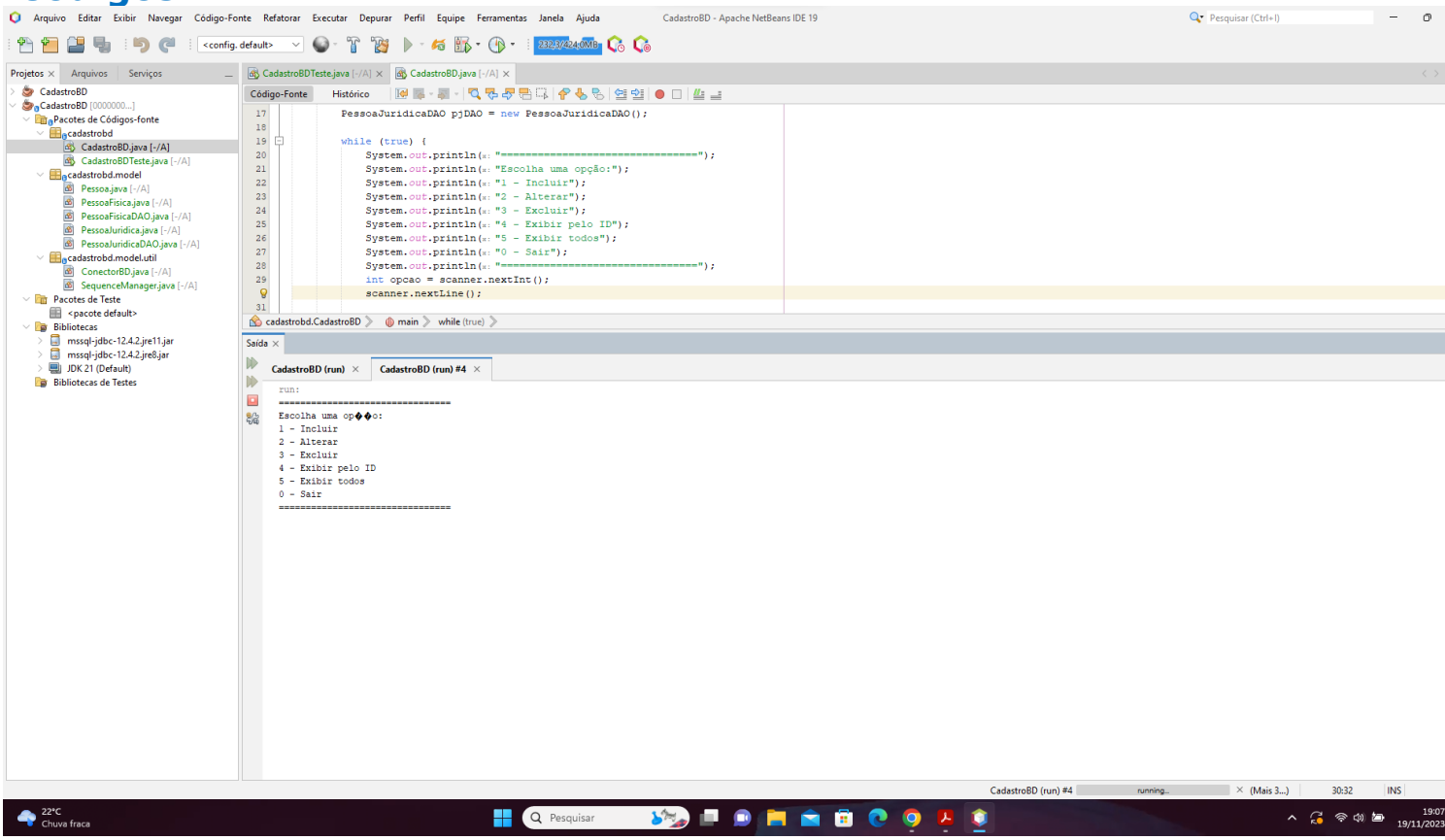
Organização: O código fica mais organizado com responsabilidades bem definidas, facilitando o entendimento e a manutenção do software.

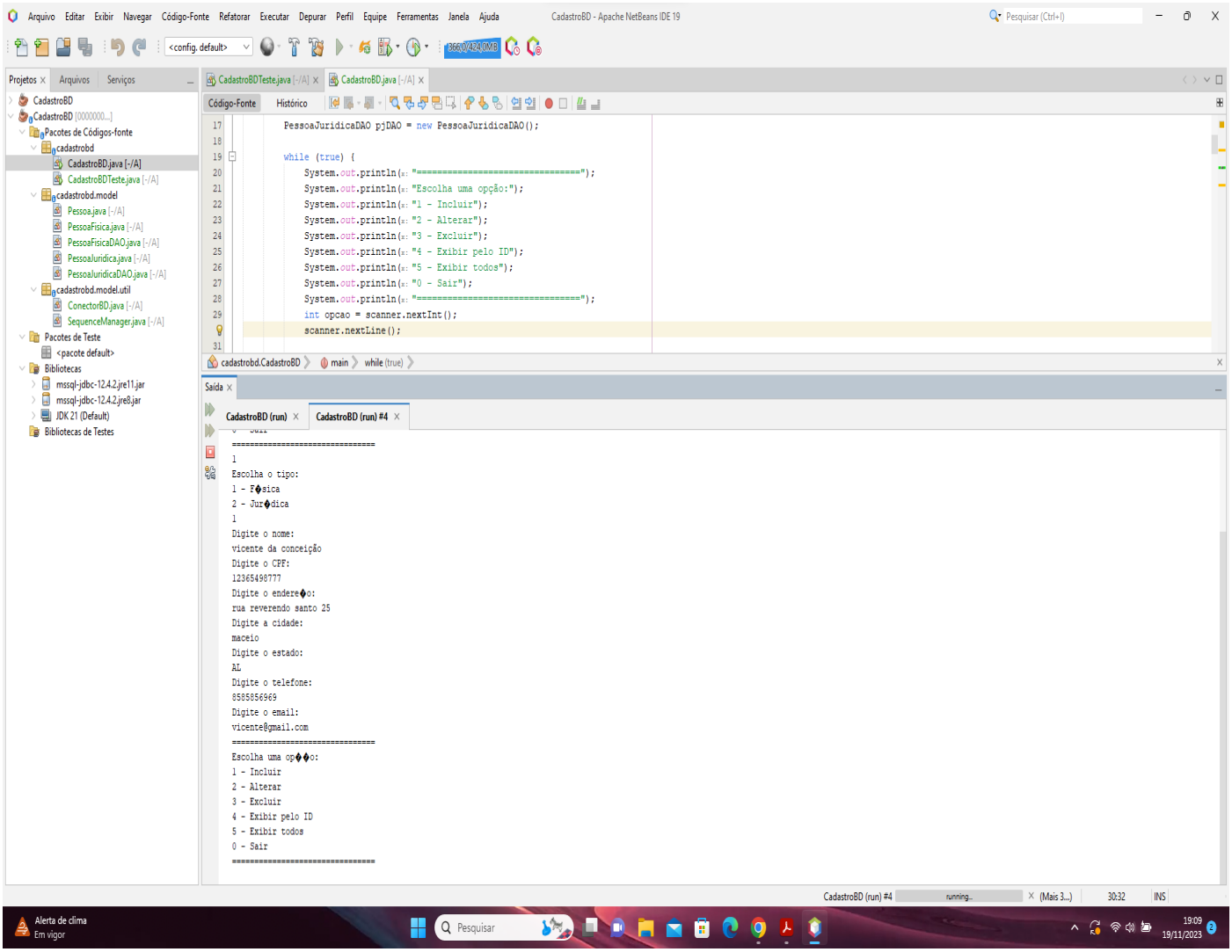
4. Como a herança é refletida no banco de dados,quando lidamos com um modelo estritamente relacional?

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:





Resultados | Mensagens

	id_pessoa	nome	endereco	cidade	estado	telefone	email
1	4	Pedro Cardoso	Avenida Brasil	Rio de Janeiro	RJ	21988887777	pedro.cardoso@email.com
2	5	Fernanda Souza	Travessa Joaquim	Salvador	BA	71977776666	fernanda.souza@email.com
3	6	COMPANY	Rua das Laranjeiras	Curitiba	PR	41966665555	company@email.com
4	7	Saulo Henrique	Rua John Keneddy	Salvador	BA	38344332	saulo.henrique@email.com
5	8	Lais Farias	Av. Nicolau	Fortaleza	CE	991177796	laisfaris@email.com
6	9	cacambas	Rua Itu	Sao Paulo	SP	38852121	cacambas@email.com
7	10	Jhon Kennedy	Rua Itu	Indaiatuba	SP	38344332	saulo@gmail.com
8	11	Rafael Paixao	Av da Saude	Salto	SP	989867543	rafael@gmail.com
9	12	Rafael Paixao	Av da Saude	Salto	SP	989867543	rafael@gmail.com
10	13	saulo henrique	rua dos trabalhadores 355	indaiatuba	sp	1991177796	saulo@gmail.com
11	14	copos ltda	rua tupinamba 15	salto	sp	1140256365	copos@gmail.com
12	15	theodoro farias	rua melvin jones 267	rio de janeiro	rj	4125256363	theodoro@gmail.com
13	16	jesus clemente	av do estado	Sao paulo	sp	11402225656	jesus@mail.com.br
14	17	vicente da conceição	rua reverendo santo 25	maceio	AL	8585856969	vicente@gmail.com

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.

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.