



Missão Prática | Nível 3 | Mundo 3

Claudio Daniel Pereira Carvalho Leite | 202401506958

Polo Cremação - Belém - PA

BackEnd sem banco não tem – 2025.1 – 3º Semestre

Objetivo da Prática

A atividade prática teve como objetivo principal consolidar conhecimentos em persistência de dados utilizando a stack Java, com foco na implementação de soluções baseadas no middleware JDBC e no padrão de projeto DAO (Data Access Object). Foi realizada a aplicação de técnicas de mapeamento objeto-relacional para garantir a integração entre as entidades do sistema e um banco de dados relacional. Como produto final, desenvolveu-se um sistema cadastral funcional com persistência de dados implementada via SQL Server.

1º Procedimento | Mapeamento Objeto-Relacional e DAO

Códigos:

Pacote com.loja :

1. Classe

https://github.com/claudio-py/loja_java_crud/blob/main/loja-app/src/main/java/com/loja/CadastroBDTest.java

CadastroBDTest:

Pacote loja.model :

2. Classe

https://github.com/claudio-py/loja_java_crud/blob/main/loja-app/src/main/java/com/loja/model/Pessoa.java

Pessoa:

3. Classe PessoaFisica:

https://github.com/claudio-py/loja_java_crud/blob/main/loja-app/src/main/java/com/loja/model/PessoaFisica.java

4. Classe PessoaJuridica

https://github.com/claudio-py/loja_java_crud/blob/main/loja-app/src/main/java/com/loja/model/PessoaJuridica.java

5. Classe PessoaFisicaDAO

https://github.com/claudio-py/loja_java_crud/blob/main/loja-app/src/main/java/com/loja/model/PessoaFisicaDAO.java

6. Classe PessoaJuridicaDAO

https://github.com/claudio-py/loja_java_crud/blob/main/loja-app/src/main/java/com/loja/model/PessoaJuridicaDAO.java

Pacote loja.model.util :

7. Class conectorBD:

https://github.com/claudio-py/loja_java_crud/blob/main/loja-app/src/main/java/com/loja/model/util/ConectorBD.java

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

Os componentes de middleware como o JDBC são fundamentais para abstrair a comunicação entre a aplicação Java e o banco de dados relacional. Eles permitem o envio de comandos SQL, o gerenciamento de conexões e o tratamento de resultados de forma padronizada, independente do SGBD utilizado, promovendo portabilidade, escalabilidade e desacoplamento entre as camadas da aplicação.

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

Statement é utilizado para executar comandos SQL estáticos, enquanto PreparedStatement é indicado para comandos parametrizados. PreparedStatement oferece maior segurança contra injeção de SQL, melhor desempenho em execuções repetidas e facilita a legibilidade e manutenção do código. Em ambientes corporativos, o uso de PreparedStatement é considerado boa prática.

c) Como o padrão DAO melhora a manutenibilidade do software?

O padrão DAO (Data Access Object) isola a lógica de acesso a dados da lógica de negócios, promovendo separação de responsabilidades. Isso facilita a manutenção, testes unitários, reaproveitamento de código e a troca de tecnologias de persistência sem impacto direto nas demais camadas da aplicação.

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

Em modelos relacionais, a herança é refletida por meio de estratégias de mapeamento, como:

Tabela única com coluna discriminadora (Table-per-Hierarchy);

Uma tabela por classe concreta (Table-per-Concrete-Class);

Tabelas separadas com relacionamento por chave estrangeira (Table-per-Subclass).

Cada abordagem possui trade-offs em termos de desempenho, normalização e complexidade nas consultas.

2º Procedimento | Alimentando a Base

Codigo:

Pacote com.loja :

8. Classe

Main

https://github.com/claudio-py/loja_java_crud/blob/main/loja-app/src/main/java/com/loja/Main.java

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

A persistência em arquivo é simples, de baixo custo e adequada para pequenos volumes de dados, porém não oferece suporte nativo a transações, concorrência ou consultas otimizadas. Já a persistência em banco de dados é mais robusta, permitindo integridade referencial, transações ACID, escalabilidade e consultas complexas via SQL, sendo mais adequada para aplicações corporativas.

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

O operador lambda permitiu a implementação de expressões funcionais de forma concisa, eliminando a necessidade de classes anônimas para percorrer coleções.

Na prática, isso simplifica a leitura e escrita de código ao iterar sobre listas de entidades e aplicar operações como `forEach`, reduzindo boilerplate e aumentando a produtividade.

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

Porque o método `main` é estático e pertence à classe, não a uma instância. Métodos chamados diretamente a partir do `main` também devem ser estáticos, pois não há objeto instanciado disponível naquele contexto. O modificador `static` permite que esses métodos sejam acessados diretamente pela classe, garantindo consistência na execução do programa.

Conclusão

A atividade prática consolidou conceitos fundamentais de persistência de dados em aplicações Java, utilizando JDBC como middleware e o padrão DAO para garantir uma arquitetura modular e de fácil manutenção. Foram aplicadas técnicas de mapeamento objeto-relacional em um contexto real, com a construção de um sistema cadastral integrado ao SQL Server. A utilização de recursos modernos da linguagem, como expressões lambda, aliada à aplicação correta dos princípios de orientação a objetos e boas práticas de codificação, reforçou a importância de soluções escaláveis, seguras e aderentes ao padrão corporativo.