Missão Prática | Nível 3 | Mundo 3

Desenvolvimento Full Stack

RPG0016 - BackEnd sem banco não tem

Aluno: Miguel Burali Artioli Firmino

Objetivos da prática:

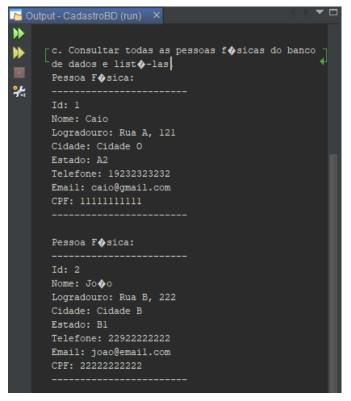
- 1. Implementar persistência com base no middleware JDBC.
- 2. Utilizar o padrão DAO (Data Access Object) no manuseio de dados.
- 3. Implementar o mapeamento objeto-relacional em sistemas Java.
- 4. Criar sistemas cadastrais com persistência em banco relacional.
- 5. No final do exercício, o aluno terá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

Resultados:

Seguem abaixo os resultados do procedimento 1:

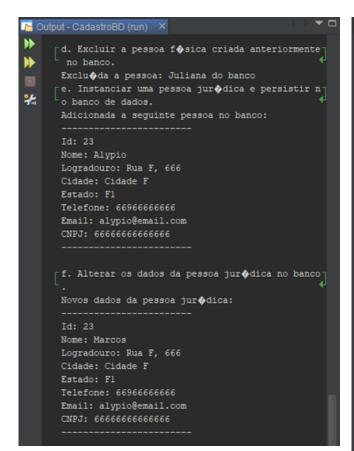
```
Noutput - CadastroBD (run)
*
     a. Instanciar uma pessoa f�sica e persistir no
      Adicionada a seguinte pessoa no banco:
<u>پ</u>ږ
      Nome: Margaret
       Logradouro: Rua E, 555
      Cidade: Cidade E
      Estado: El
       Email: margaret@email.com
      CPF: 5555555555
      b. Alterar os dados da pessoa fosica no banco.
      Novos dados da pessoa f∳sica:
       Id: 22
      Nome: Juliana
       Logradouro: Rua E, 555
      Cidade: Cidade E
      Estado: El
       Telefone: 5595555555
       Email: margaret@email.com
       CPF: 5555555555
```

Objetivos A e B

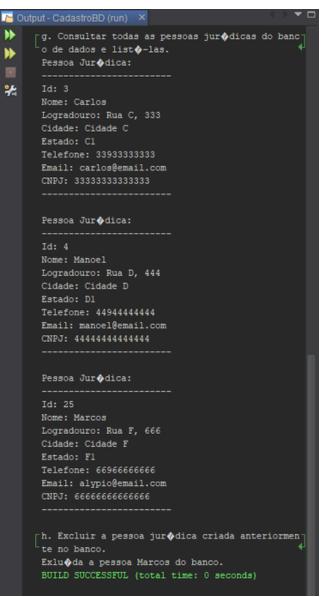


Objetivo C (há mais dados ocultos)





Objetivos D, E e F



Objetivos G e H



Missão Prática | Nível 3 | Mundo 3 Análise e Conclusão 1

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

Componentes como o JDBC são fundamentais, porque atuam como ponte entre o Java e o banco de dados. Com o JDBC, é possivel executar comandos SQL em qualquer banco, de forma padronizada, o que aumenta portabilidade e eficiência da aplicação.

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

O Statement é usado para executar instruções SQL simples, mas não é seguro contra injeção de SQL e não é otimizado para comandos repetitivos. Já o PreparedStatement permite preparar um comando SQL com parâmetros, tornando a execução mais segura (contra injeção de SQL) e eficiente, pois o SQL pode ser pré-compilado pelo banco. Portanto, na maioria dos casos, PreparedStatement é preferido.

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

O padrão DAO (Data Access Object) é baseado na criação de um objeto para representar as funcionalidades pertinentes à lógica da aplicação, e outro para lidar com o banco de dados. O padrão permite trabalhar com mais eficiência, pois, com a separação de responsabilidades, tem-se uma estrutura mais organizada, na qual uma mudança em uma parte do código não afeta as outras.

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

Na prática que foi proposta nesse trabalho, utilizei a estratégia de tabelas com junções, onde cria-se uma tabela para a classe principal: "pessoa", e outras para as sub-classes: "pessoa_fisica" e "pessoa_juridica", que são unidas por meio da chave estrangeira: "id" por meio de junções, quando é necessário obter os dados completos. Achei interessante o uso dessa estratégia, pois com ela não é necessário duplicidade de dados.



Resultados:

Seguem abaixo os resultados do procedimento 2:

```
Љ Output - CadastroBD (run) ×
      1 - Incluir Pessoa
%
      4 - Buscar pelo Id
      5 - Exibir Todos
      0 - Finalizar Programa
      Escolha uma Op��o:
      F - Pessoa F♦sica | J - Pessoa Jur♦dica
      Insira Id da pessoa:
      Buscando pessoa...
      Dados de Caio:
      Nome: Caio
      Logradouro: Rua A, 121
      Cidade: Cidade O
      Estado: A2
      Email: caio@gmail.com
```

Buscando uma pessoa pelo id (opção 4)

Excluindo uma pessoa pelo id (opção 3)



Missão Prática | Nível 3 | Mundo 3 Análise e Conclusão 2

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

A persistência em arquivo trata-se de salvar os dados nos armazenamento do sistema, geralmente em formato .txt, .csv ou .json. É uma solução mais fácil de programar, porém menos organizada e escalável. A persistência em banco de dados, por outro lado, significa armazenar os dados em banco de dados de forma mais estruturada, eficiente e segura, por meio de sistemas como SQL Server, MySQL, etc.

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 escrita de código mais curto e prático, sem a necesside de criar classes para implementar as funcionalidades necessárias. Isso facilitou a leitura e manipulação de objetos. É possivel reduzir drasticamente a escrita de código como no exemplo abaixo:

```
lista.forEach(e -> System.out.println(e));
// ao invés de

for (int i; i < lista.size(); i++) {
        System.out.println(lista.get(i));
}</pre>
```

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

Pois o método main é estático e pertence à classe, e não a uma instância (objeto). Métodos estáticos só podem acessar diretamente outros métodos estáticos. Como não há um objeto criado ainda, é necessário que os métodos chamados também sejam estáticos para que possam ser acessados diretamente pelo Java durante a execução do programa.

