

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:

```
Output - CadastroBD (run) x
run:
a. Instanciar uma pessoa física e persistir no
banco de dados.
Adicionada a seguinte pessoa no banco:
-----
Id: 22
Nome: Margaret
Logradouro: Rua E, 555
Cidade: Cidade E
Estado: E1
Telefone: 55955555555
Email: margaret@email.com
CPF: 55555555555
-----

b. Alterar os dados da pessoa física no banco.
Novos dados da pessoa física:
-----
Id: 22
Nome: Juliana
Logradouro: Rua E, 555
Cidade: Cidade E
Estado: E1
Telefone: 55955555555
Email: margaret@email.com
CPF: 55555555555
-----
```

Objetivos A e B

```
Output - CadastroBD (run) x
c. Consultar todas as pessoas físicas do banco
de dados e listá-las
Pessoa Física:
-----
Id: 1
Nome: Caio
Logradouro: Rua A, 121
Cidade: Cidade O
Estado: A2
Telefone: 19232323232
Email: caio@gmail.com
CPF: 11111111111
-----

Pessoa Física:
-----
Id: 2
Nome: João
Logradouro: Rua B, 222
Cidade: Cidade B
Estado: B1
Telefone: 22922222222
Email: joao@email.com
CPF: 22222222222
-----
```

Objetivo C (há mais dados ocultos)

```
Output - CadastroBD (run) x
[d. Excluir a pessoa física criada anteriormente]
no banco.
Excluída a pessoa: Juliana do banco
[e. Instanciar uma pessoa jurídica e persistir n]
o banco de dados.
Adicionada a seguinte pessoa no banco:
-----
Id: 23
Nome: Aypio
Logradouro: Rua F, 666
Cidade: Cidade F
Estado: Fl
Telefone: 669666666666
Email: aypio@email.com
CNPJ: 66666666666666
-----

[f. Alterar os dados da pessoa jurídica no banco]
.
Novos dados da pessoa jurídica:
-----
Id: 23
Nome: Marcos
Logradouro: Rua F, 666
Cidade: Cidade F
Estado: Fl
Telefone: 669666666666
Email: aypio@email.com
CNPJ: 66666666666666
-----
```

Objetivos D, E e F

```
Output - CadastroBD (run) x
[g. Consultar todas as pessoas jurídicas do banc]
o de dados e listá-las.
Pessoa Jurídica:
-----
Id: 3
Nome: Carlos
Logradouro: Rua C, 333
Cidade: Cidade C
Estado: Cl
Telefone: 339333333333
Email: carlos@email.com
CNPJ: 33333333333333
-----

Pessoa Jurídica:
-----
Id: 4
Nome: Manoel
Logradouro: Rua D, 444
Cidade: Cidade D
Estado: Dl
Telefone: 449444444444
Email: manoel@email.com
CNPJ: 44444444444444
-----

Pessoa Jurídica:
-----
Id: 25
Nome: Marcos
Logradouro: Rua F, 666
Cidade: Cidade F
Estado: Fl
Telefone: 669666666666
Email: aypio@email.com
CNPJ: 66666666666666
-----

[h. Excluir a pessoa jurídica criada anteriormen]
te no banco.
Excluída a pessoa Marcos do banco.
BUILD SUCCESSFUL (total time: 0 seconds)
```

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 possível 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:

```
Output - CadastroBD (run) x
run:
=====
1 - Incluir Pessoa
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo Id
5 - Exibir Todos
0 - Finalizar Programa
=====
Escolha uma Opção:

4
Escolha o tipo de pessoa:
F - Pessoa Física | J - Pessoa Jurídica

F
Insira Id da pessoa:

1
Buscando pessoa...
Dados de Caio:
-----
Id: 1
Nome: Caio
Logradouro: Rua A, 121
Cidade: Cidade 0
Estado: A2
Telefone: 19232323232
Email: caio@gmail.com
CPF: 11111111111
-----
```

Buscando uma pessoa pelo id
(opção 4)

```
=====
1 - Incluir Pessoa
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo Id
5 - Exibir Todos
0 - Finalizar Programa
=====
Escolha uma Opção:

3
Escolha o tipo de pessoa:
F - Pessoa Física | J - Pessoa Jurídica

J
Insira Id da pessoa:

1
Exluindo pessoa jurídica...
Pessoa excluída com sucesso!
```

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 no 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 necessidade de criar classes para implementar as funcionalidades necessárias. Isso facilitou a leitura e manipulação de objetos. É possível 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));  
}
```

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.

