



Universidade Estácio
Campus Vargem Grande Paulista
Curso: Desenvolvimento Full Stack
Disciplina: Iniciando o Caminho Pelo Java
Turma: 2024.3
3º semestre letivo
Marcia da Silva e Souza

Título da Prática: BackEnd sem banco não tem
Criação de aplicativo Java, com acesso ao banco de dados SQL Server
através do
middleware JDBC

Objetivo 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 SQLServer na persistência de dados.



```
run - CadastroBD (run) X Git - CadastroPOO - master

RUN:

INSERT INTO PessoaFisica success.
Pessoa Fisica incluida com ID: 16

id: 16
nome: Sandra
endereco: Avenida S, 99
cidade: Sao Paulo
estado: SP
telefone: 9999-9999
email: sandra@gmail.com
CPF: 999999999999

Pessoa Fisica alterada.
id: 16
nome: Sandra
endereco: Avenida S, 99
cidade: Rio de Janeiro
estado: RJ
telefone: 9999-9999
email: sandra@gmail.com
CPF: 999999999999
```

Execução da classe CadastroBDTest: Inserir e alterar pessoa física

```
run - CadastroBD (run) X Git - CadastroPOO - master

Pessoas Fisica excluida.

INSERT INTO PessoaJuridica success.
Pessoa Juridica incluida com ID: 19

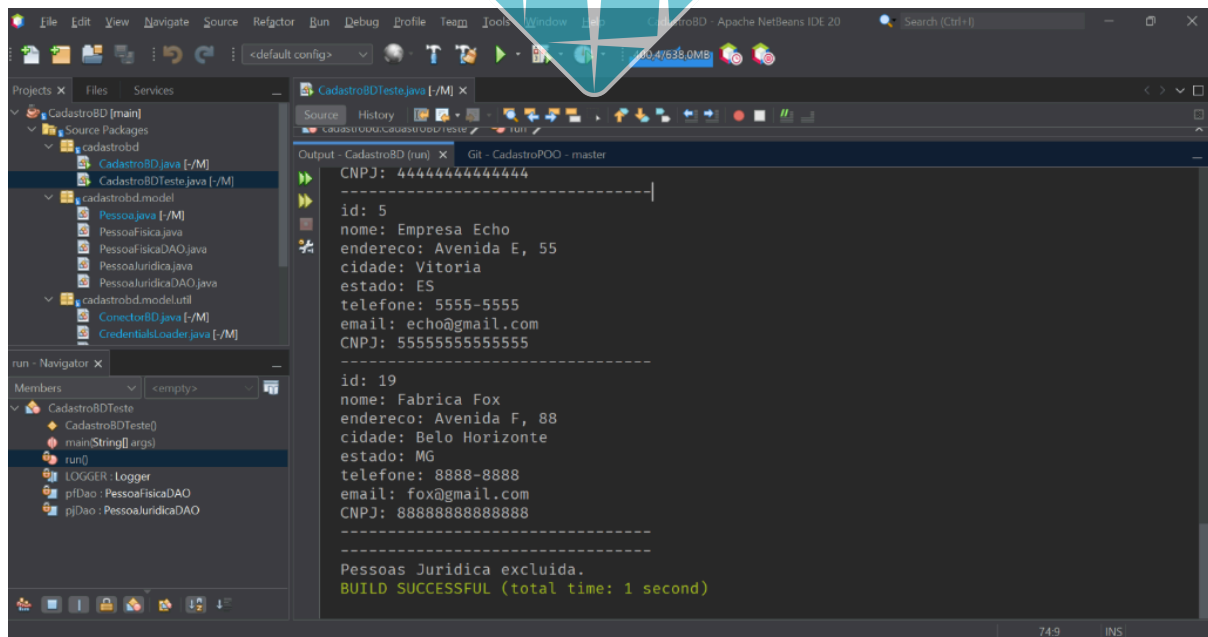
id: 19
nome: Fabrica Fox
endereco: Avenida F, 88
cidade: Goias
estado: GO
telefone: 8888-8888
email: fox@gmail.com
CNPJ: 8888888888888888

Pessoa Juridica alterada.

Exibir todas as pessoas juridicas:

id: 4
nome: Distribuidora Delta
endereco: Avenida D, 44
cidade: Brasilia
estado: DF
```

Execução da classe CadastroBDTest: Exclusão de pessoa física ,inserção e alteração de pessoa juridica



Execução da classe CadastroBDTest: exclusão de pessoa juridica

Álise e conclusão:

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

O middleware, como o JDBC, é fundamental para conectar aplicações com bancos de dados. Ele atua como um intermediário, facilitando a comunicação e permitindo que diferentes partes de uma aplicação trabalhem juntas de forma eficiente. Sem o middleware.

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

Statement: Isso é usado para executar instruções SQL simples e estáticas. Por exemplo, é útil quando você sabe de antemão que sempre usará a mesma consulta SQL. Ele não é muito eficiente para consultas repetidas ou quando há a necessidade de inserir dados dinamicamente



PreparedStatement: Agora, aqui está a magia. PreparedStatement permite preparar uma instrução SQL que pode ser executada múltiplas vezes com parâmetros diferentes. Isso não apenas melhora o desempenho, porque a instrução é compilada uma vez e pode ser executada várias vezes, mas também ajuda na prevenção de injeções SQL, o que torna sua aplicação mais segura.

Como o padrão DAO melhora a manutenibilidade do software?

O Dao tem a função de capturar e persistir os dados de um objeto(entidade) que representa um registro em uma base de dados (uma linha em uma tabela, por exemplo).

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

A herança em um banco de dados relacional pode ser mapeada de várias formas, dependendo de vários fatores, como a quantidade de dados, o tamanho da hierarquia, os tipos de consultas, entre outros

2º procedimento:

Resultados dos códigos:



```
run:|

=====
1 - Incluir
2 - Alterar
3 - Excluir
4 - Buscar pelo ID
5 - Exibir todos
0 - Sair
=====
ESCOLHA: 1
F - Pessoa Fisica | J - Pessoa Juridica
TIPO DE PESSOA: F
Informe o nome: Hugo
Informe o CPF: 88888888888
Informe o endereco: Habitat Hectare
Informe a cidade: Higienopolis
Informe o estado: SP
Informe o telefone: 8888-8888
Informe o email: hugo@gmail.com
INSERT INTO PessoaFisica success.

=====
1 - Incluir
```

Execução da classe CadastroBD: 1incluir

```
endereco: Avenida C, 33
cidade: Fortaleza
estado: CE
telefone: 3333-3333
email: carlos@gmail.com
CPF: 33333333333

-----
id: 20
nome: Hugo
endereco: Habitat Hectare
cidade: Higienopolis
estado: SP
telefone: 8888-8888
email: hugo@gmail.com
CPF: 88888888888

-----
id: 4
nome: Distribuidora Delta
endereco: Avenida D, 44
cidade: Brasilia
estado: DF
telefone: 4444-4444
email: delta@gmail.com
CNPJ: 444444444444444
```

Execução da classe CadastroBD: Pessoa incluída com sucesso.



```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config> 286.2/728.0MB
Projects X Files Services
  CadastroBD [main]
    Source Packages
      cadastrobd
        CadastroBD.java [-/M]
        CadastroBDTeste.java [-/M]
        cadastrobd.model
          Pessoa.java [-/M]
          PessoaFisica.java
          PessoaFisicaDAO.java
          PessoaJuridica.java
          PessoaJuridicaDAO.java
        cadastrobd.model.util
          ConectorBD.java [-/M]
          CredentialsLoader.java [-/M]
CadastroBD.java - Navigator X
Members <empty>
  CadastroBD
    CadastroBD()
    intAnswerQuestion(String question) : Integer
    main(String[] args)
    printMenu()
    run()
    strAnswerQuestion(String question) : String
    LOGGER : Logger
    in : Scanner
    pfDao : PessoaFisicaDAO
    pjDao : PessoaJuridicaDAO
Output - CadastroBD (run) X Git - CadastroPOO - master
  1 - Incluir
  2 - Alterar
  3 - Excluir
  4 - Buscar pelo ID
  5 - Exibir todos
  0 - Sair
  =====
  ESCOLHA: 2
  F - Pessoa Fisica | J - Pessoa Juridica
  TIPO DE PESSOA: F
  Informe o ID da Pessoa Fisica: 20
  Informe o nome: Gisele
  Informe o CPF: 88888888888
  Informe o endereco: Habitat Hectare
  Informe a cidade: Governador Valadares
  Informe o estado: MG
  Informe o telefone: 8888-8888
  Informe o email: gisele@hotmail.com
  =====
  1 - Incluir
  2 - Alterar
```

Execução da classe CadatroBD: Alterar

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config> 286.2/728.0MB
Projects X Files Services
  CadastroBD [main]
    Source Packages
      cadastrobd
        CadastroBD.java [-/M]
        CadastroBDTeste.java [-/M]
        cadastrobd.model
          Pessoa.java [-/M]
          PessoaFisica.java
          PessoaFisicaDAO.java
          PessoaJuridica.java
          PessoaJuridicaDAO.java
        cadastrobd.model.util
          ConectorBD.java [-/M]
          CredentialsLoader.java [-/M]
CadastroBD.java - Navigator X
Members <empty>
  CadastroBD
    CadastroBD()
    intAnswerQuestion(String question) : Integer
    main(String[] args)
    printMenu()
    run()
    strAnswerQuestion(String question) : String
    LOGGER : Logger
    in : Scanner
    pfDao : PessoaFisicaDAO
    pjDao : PessoaJuridicaDAO
Output - CadastroBD (run) X Git - CadastroPOO - master
  nome: Carlos
  endereco: Avenida C, 33
  cidade: Fortaleza
  estado: CE
  telefone: 3333-3333
  email: carlos@gmail.com
  CPF: 33333333333
  -----
  id: 20
  nome: Gisele
  endereco: Habitat Hectare
  cidade: Governador Valadares
  estado: MG
  telefone: 8888-8888
  email: gisele@hotmail.com
  CPF: 88888888888
  -----
  id: 4
  nome: Distribuidora Delta
  endereco: Avenida D, 44
  cidade: Brasilia
  estado: DF
  telefone: 4444-4444
  email: delta@gmail.com
```

Execução da classe CadastroBD: Pessoa alterada com sucesso



```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config> 507.2/728.0MB

Projects X Files Services
CadastroBD [main]
  Source Packages
    cadastrobd
      CadastroBD.java [-/M]
      CadastroBDTeste.java [-/M]
      cadastrobd.model
        Pessoa.java [-/M]
        PessoaFisica.java
        PessoaFisicaDAO.java
        PessoaJuridica.java
        PessoaJuridicaDAO.java
      cadastrobd.model.util
        ConectorBD.java [-/M]
        CredentialsLoader.java [-/M]

CadastroBD.java - Navigator X
Members
  CadastroBD
    CadastroBD()
    intAnswerQuestion(String question) : Integer
    main(String[] args)
    printMenu()
    run()
    strAnswerQuestion(String question) : String
    LOGGER : Logger
    in : Scanner
    pfDao : PessoaFisicaDAO
    pjDao : PessoaJuridicaDAO

Output - CadastroBD (run) X Git - CadastroPOO - master
email: echo@gmail.com
CNPJ: 5555555555555555
=====
1 - Incluir
2 - Alterar
3 - Excluir
4 - Buscar pelo ID
5 - Exibir todos
0 - Sair
=====
ESCOLHA: 3
F - Pessoa Fisica | J - Pessoa Juridica
TIPO DE PESSOA: f
Informe o ID da Pessoa Fisica: 20
Excluido com sucesso.
=====
1 - Incluir
2 - Alterar
3 - Excluir
4 - Buscar pelo ID
5 - Exibir todos
```

Execução da classe CadastroBD: Excluir

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config> 507.3/728.0MB

Projects X Files Services
CadastroBD [main]
  Source Packages
    cadastrobd
      CadastroBD.java [-/M]
      CadastroBDTeste.java [-/M]
      cadastrobd.model
        Pessoa.java [-/M]
        PessoaFisica.java
        PessoaFisicaDAO.java
        PessoaJuridica.java
        PessoaJuridicaDAO.java
      cadastrobd.model.util
        ConectorBD.java [-/M]
        CredentialsLoader.java [-/M]

CadastroBDTeste.java - Navigator X
Members
  CadastroBDTeste
    CadastroBDTeste()
    intAnswerQuestion(String question) : Integer
    main(String[] args)
    printMenu()
    run()
    strAnswerQuestion(String question) : String
    LOGGER : Logger
    in : Scanner
    pfDao : PessoaFisicaDAO
    pjDao : PessoaJuridicaDAO

Output - CadastroBD (run) X Git - CadastroPOO - master
=====
1 - Incluir
2 - Alterar
3 - Excluir
4 - Buscar pelo ID
5 - Exibir todos
0 - Sair
=====
ESCOLHA: 4
F - Pessoa Fisica | J - Pessoa Juridica
TIPO DE PESSOA: j
Informe o ID da Pessoa Juridica: 5
id: 5
nome: Empresa Echo
endereco: Avenida E, 55
cidade: Vitoria
estado: ES
telefone: 5555-5555
email: echo@gmail.com
CNPJ: 5555555555555555
=====
1 - Incluir
```

Execução da classe CadastroBD: Buscar pelo ID



```
1 - Incluir
2 - Alterar
3 - Excluir
4 - Buscar pelo ID
5 - Exibir todos
0 - Sair
=====
ESCOLHA: 5
=====
id: 1
nome: Andrea
endereco: Avenida A, 11
cidade: Rio Branco
estado: AC
telefone: 1111-1111
email: andre@gmail.com
CPF: 11111111111
-----
id: 2
nome: Bruna
endereco: Avenida B, 22
cidade: Salvador
```

Execução da classe CadastroBD: exibir todos

Análise e conclusão:

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

Cada método tem seu uso, mas a persistência em arquivo e a em banco de dados têm diferenças bem claras.

Persistência em arquivo: Ideal para dados simples e menos estruturados. Fácil de implementar, mas pode se tornar caótica à medida que os dados crescem. Dificulta buscas complexas e manipulação de dados. Segurança e integridade também são limitadas.

Persistência em banco de dados: Perfeita para dados estruturados e grandes volumes de informações. Oferece robustez, segurança, integridade e ferramentas avançadas para buscas e manipulações complexas. É escalável e facilita a manutenção. Claro, requer mais configuração inicial e gerenciamento contínuo.



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

Os operadores lambda realmente transformaram a forma como manipulamos coleções e iteramos sobre elementos no Java. Antes, a impressão dos valores das entidades exigia a criação de um loop explícito. Agora, com lambdas, esse processo é muito mais conciso e legível.

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

Métodos precisam ser marcados como static porque o método main é estático. E métodos estáticos pertencem à classe, não a uma instância. Isso significa que podem ser invocados sem criar um objeto da classe. Para acessar um método a partir de um método estático (como main), o método chamado também precisa ser estático.

Basicamente, o main é a porta de entrada e precisa ter tudo pronto sem depender de criar instâncias.