

Missão Prática | Nível 3 | Mundo 3

Material de *orientações* para desenvolvimento da *missão prática* do *3º nível de conhecimento*.

RPG0016 - BackEnd sem banco não tem

Criação de aplicativo Java, com acesso ao banco de dados SQL Server através do middleware JDBC.

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.

 **As práticas devem ser feitas individualmente.**

Materiais necessários para a prática

- SQL Server, com o banco de dados gerado em prática anterior (loja).
 - JDK e IDE NetBeans.
 - Navegador para Internet, como o Chrome.
 - Banco de dados SQL Server com o Management Studio.
 - Equipamentos:
 - Computador com acesso à Internet.
 - JDK e IDE NetBeans.
 - Banco de dados SQL Server.
 - Navegador de Internet instalado no computador.
-

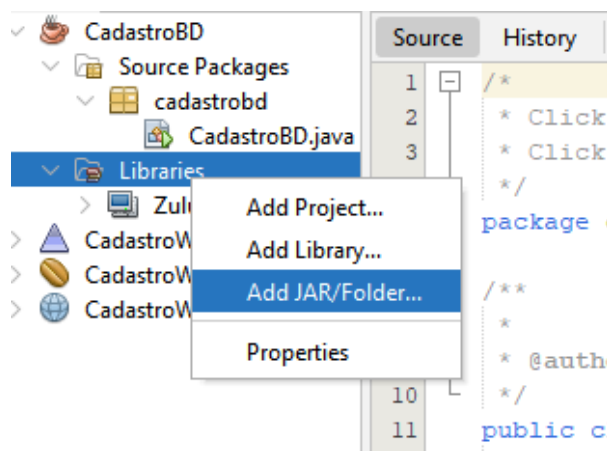
Desenvolvimento da prática

Vamos colocar a **mão na massa**?! Siga as **instruções** abaixo para desenvolvimento desta missão.

👉 1º Procedimento | Mapeamento Objeto-Relacional e DAO

1. Criar o projeto e configurar as bibliotecas necessárias:

- Criar um projeto no NetBeans, utilizando o nome **CadastroBD**, do tipo Aplicativo Java Padrão (modelo Ant).
- Adicionar o driver JDBC para SQL Server ao projeto, com o clique do botão direito sobre **bibliotecas** (libraries) e escolha da opção **jar**.



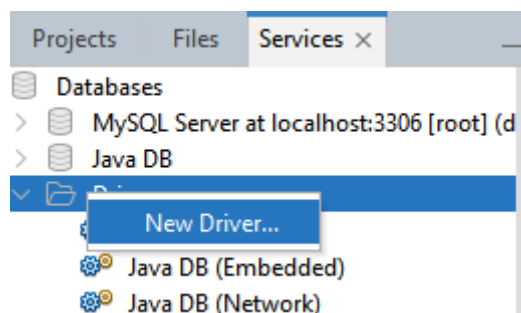
c. Selecionar o arquivo **mssql-jdbc-12.2.0.jre11.jar**, que é parte do arquivo **zip** encontrado no endereço seguinte.

<https://learn.microsoft.com/pt-br/sql/connect/jdbc/download-microsoft-jdbc-driver-for-sql-server?view=sql-server-ver16>

d. Após descompactar o arquivo, copie o arquivo jar necessário para uma pasta de fácil acesso e adicione ao projeto, conforme instrução anterior.

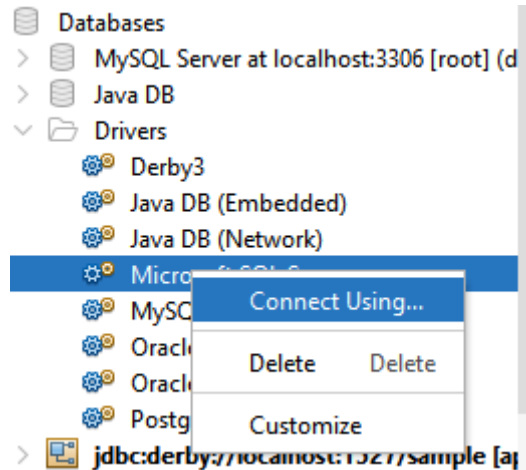
2. Configurar o acesso ao banco pela aba de serviços do NetBeans.

- Na aba de **Serviços**, divisão **Banco de Dados**, clique com o botão direito em **Drivers** e escolha **Novo Driver**.



b. Na janela que se abrirá, clicar em **Add** (Adicionar), escolher o arquivo **jar** utilizado no passo anterior e finalizar com **Ok**.

c. O reconhecimento será automático, e podemos definir uma conexão com o clique do botão direito sobre o driver e escolha de **Conectar Utilizando**.



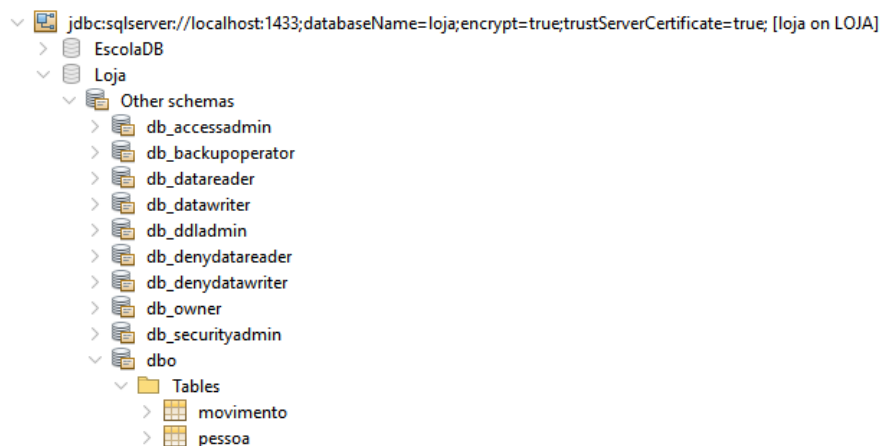
d. Para os campos **database**, **user** e **password**, utilizar o valor **loja**, de acordo com os elementos criados em exercício anterior sobre a criação do banco de dados de exemplo, marcando também a opção **Lembrar Senha**.

e. Para o campo JDBC URL deve ser utilizada a seguinte expressão:

jdbc:sqlserver://localhost:1433;databaseName=loja;encrypt=true;trustServerCertificate=true;

f. Clicar em **Testar Conexão** e, estando tudo certo, **Finalizar**.

g. Ao clicar duas vezes na nova conexão, os objetos do banco estarão todos disponíveis na árvore de navegação.



h. Utilizar o clique do botão direito sobre as tabelas, e escolher **Visualizar Dados** (View Data), para consultar os dados atualmente no banco.

3. Voltando ao projeto, criar o pacote **cadastrobd.model**, e nele criar as classes apresentadas a seguir:

- Classe **Pessoa**, com os campos **id**, **nome**, **logradouro**, **cidade**, **estado**, **telefone** e **email**, construtor padrão e completo, além de método **exibir**, para impressão dos dados no console
- Classe **PessoaFisica**, herdando de **Pessoa**, com acréscimo do campo **cpf**, além da reescrita dos construtores e uso de polimorfismo em **exibir**
- Classe **PessoaJuridica**, herdando de **Pessoa**, com acréscimo do campo **cnpj**, além da reescrita dos construtores e uso de polimorfismo em **exibir**

4. Criar o pacote **cadastro.model.util**, para inclusão das classes utilitárias que são apresentadas a seguir:

- Classe **ConectorBD**, com os métodos **getConnection**, para retornar uma conexão com o banco de dados, **getPrepared**, para retornar um objeto do tipo **PreparedStatement** a partir de um SQL fornecido com parâmetro, e **getSelect**, para retornar o **ResultSet** relacionado a uma consulta.
- Ainda na classe **ConectorBD**, adicionar métodos **close** sobrecarregados para **Statement**, **ResultSet** e **Connection**, visando garantir o fechamento, ou encerramento, de todos os objetos de acesso ao banco gerados.
- Classe **SequenceManager**, que terá o método **getValue**, recebendo o nome da sequência como parâmetro e retornando o próximo valor.

5. Codificar as classes no padrão DAO, no pacote **cadastro.model**.

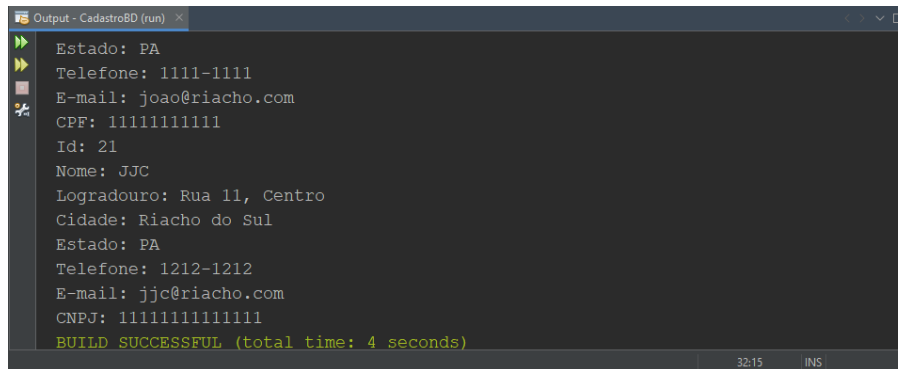
- Classe **PessoaFisicaDAO**, com os métodos **getPessoa**, retornando uma pessoa física a partir do seu **id**, **getPessoas**, para retorno de todas as pessoas físicas do banco de dados, **incluir**, para inclusão de uma pessoa física, fornecida como **parâmetro**, nas tabelas **Pessoa** e **PessoaFisica**, **alterar**, para alteração dos dados de uma pessoa física, e **excluir**, para remoção da pessoa do banco em ambas as tabelas.
- Classe **PessoaJuridicaDAO**, com os métodos **getPessoa**, retornando uma pessoa jurídica a partir do seu **id**, **getPessoas**, para retorno de todas as pessoas jurídicas do banco de dados, **incluir**, para inclusão de uma pessoa jurídica, fornecida como **parâmetro**, nas tabelas **Pessoa** e **PessoaJuridica**, **alterar**, para alteração dos dados de uma pessoa jurídica, e **excluir**, para remoção da pessoa do banco em ambas as tabelas.
- Utilizar nas classes objetos dos tipos **ConectorBD** e **SequenceManager**.

6. Criar uma classe principal de testes com o nome **CadastroBDTeste**, efetuando as operações seguintes no método **main**:

- Instanciar uma pessoa física e persistir no banco de dados.
- Alterar os dados da pessoa física no banco.
- Consultar todas as pessoas físicas do banco de dados e listar no console.
- Excluir a pessoa física criada anteriormente no banco.
- Instanciar uma pessoa jurídica e persistir no banco de dados.

- Alterar os dados da pessoa jurídica no banco.
- Consultar todas as pessoas jurídicas do banco e listar no console.
- Excluir a pessoa jurídica criada anteriormente no banco.

7. A saída do sistema deverá ser semelhante à que é apresentada a seguir:



```

Estado: PA
Telefone: 1111-1111
E-mail: joao@riacho.com
CPF: 11111111111
Id: 21
Nome: JJC
Logradouro: Rua 11, Centro
Cidade: Riacho do Sul
Estado: PA
Telefone: 1212-1212
E-mail: jjc@riacho.com
CNPJ: 11111111111111
BUILD SUCCESSFUL (total time: 4 seconds)
  
```

8. Verificar os resultados obtidos através do console de saída do NetBeans.

Resultados esperados

1. É importante que o código seja organizado.
2. Outro ponto importante é explorar as funcionalidades oferecidas pelo NetBeans para melhoria da produtividade.
3. Nesse exercício, é esperado que o estudante demonstre as habilidades básicas para a construção de projetos na plataforma Java, com acesso a banco de dados através dos componentes do JDBC.

Relatório discente de acompanhamento

Os Relatórios de Práticas deverão ser confeccionados em arquivo no formato PDF, com a Logo da Universidade, nome do Campus, nome do Curso, nome da Disciplina, número da Turma, semestre letivo, nome dos integrantes da Prática. Além disso, o projeto deve ser armazenado em um repositório no GIT e o respectivo endereço deve constar na documentação. A documentação do projeto deve conter:

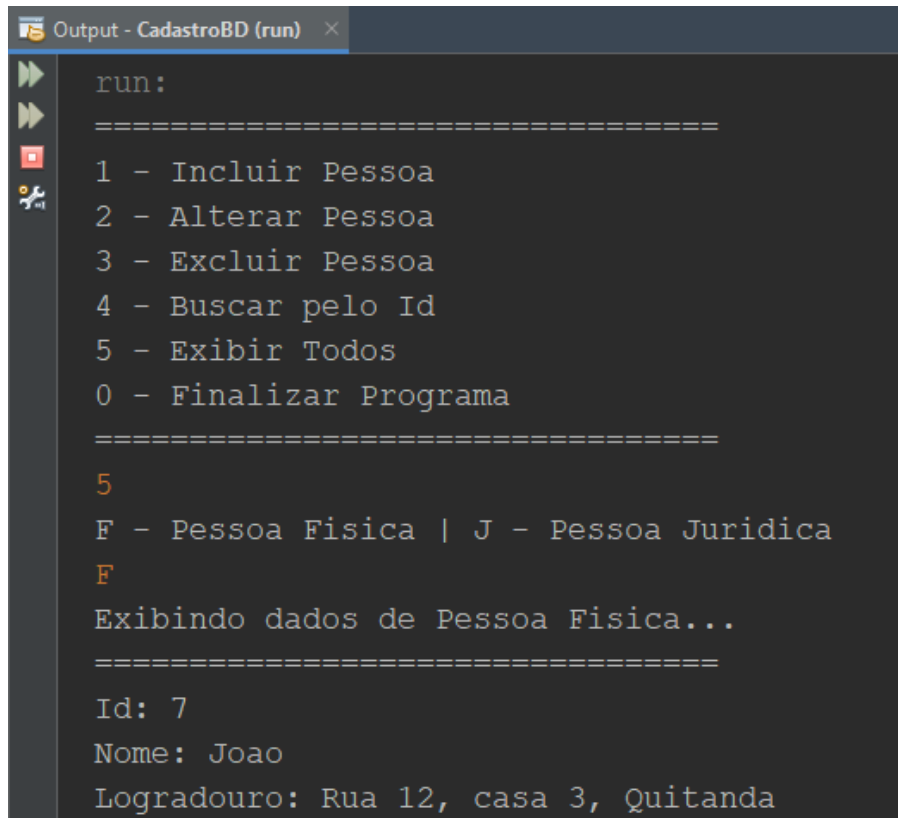
1. Título da Prática;
2. Objetivo da Prática;
3. Todos os códigos solicitados neste roteiro de aula;

4. Os resultados da execução dos códigos também devem ser apresentados;
5. Análise e Conclusão:
 - Qual a importância dos componentes de middleware, como o JDBC?
 - Qual a diferença no uso de Statement ou PreparedStatement para a manipulação de dados?
 - Como o padrão DAO melhora a manutenibilidade do software?
 - Como a herança é refletida no banco de dados, quando lidamos com um modelo estritamente relacional?

2º Procedimento | Alimentando a Base

1. Alterar o método **main** da classe principal do projeto, para implementação do cadastro em modo texto:
 - Apresentar as opções do programa para o usuário, sendo 1 para incluir, 2 para alterar, 3 para excluir, 4 para exibir pelo id, 5 para exibir todos e 0 para finalizar a execução.
 - Selecionada a opção **incluir**, escolher o tipo (Física ou Jurídica), receber os dados a partir do teclado e adicionar no banco de dados através da classe DAO correta.
 - Selecionada a opção **alterar**, escolher o tipo (Física ou Jurídica), receber o **id** a partir do teclado, apresentar os dados atuais, solicitar os novos dados e alterar no banco de dados através do DAO.
 - Selecionada a opção **excluir**, escolher o tipo (Física ou Jurídica), receber o **id** a partir do teclado e remover do banco de dados através do DAO.
 - Selecionada a opção **obter**, escolher o tipo (Física ou Jurídica), receber o **id** a partir do teclado e apresentar os dados atuais, recuperados do banco através do DAO.
 - Selecionada a opção **obterTodos**, escolher o tipo (Física ou Jurídica) e apresentar os dados de todas as entidades presentes no banco de dados por intermédio do DAO.
 - Qualquer exceção que possa ocorrer durante a execução do sistema deverá ser tratada.
 - Selecionada a opção **sair**, finalizar a execução do sistema.
2. Testar as funcionalidades do sistema:
 - Efetuar as diversas operações disponibilizadas, tanto para pessoa jurídica quanto para pessoa física.
 - Feitas as operações, verificar os dados no SQL Server, com a utilização da aba Services, divisão Databases, do NetBeans, ou através do SQL Server Management Studio.

3. Ajustar as características para obter uma execução como a seguinte:



```
run:
=====
1 - Incluir Pessoa
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo Id
5 - Exibir Todos
0 - Finalizar Programa
=====
5
F - Pessoa Fisica | J - Pessoa Juridica
F
Exibindo dados de Pessoa Fisica...
=====
Id: 7
Nome: Joao
Logradouro: Rua 12, casa 3, Quitanda
```

Resultados esperados

1. É importante que o código seja organizado.
2. Outro ponto importante é explorar as funcionalidades oferecidas pelo NetBeans para melhoria da produtividade.
3. Nesse exercício, é esperado que o estudante demonstre habilidade para a criação de sistemas na plataforma Java, com acesso a banco de dados relacional.

Relatório discente de acompanhamento

Os Relatórios de Práticas deverão ser confeccionados em arquivo no formato PDF, com a Logo da Universidade, nome do Campus, nome do Curso, nome da Disciplina, número da Turma, semestre letivo, nome dos integrantes da Prática. Além disso, o projeto deve ser armazenado em um repositório no GIT e o respectivo endereço deve constar na documentação. A documentação do projeto deve conter:

1. Título da Prática;

2. Objetivo da Prática;
3. Todos os códigos solicitados neste roteiro de aula;
4. Os resultados da execução dos códigos também devem ser apresentados;
5. Análise e Conclusão:
 - Quais as diferenças entre a persistência em arquivo e a persistência em banco de dados?
 - Como o uso de operador lambda simplificou a impressão dos valores contidos nas entidades, nas versões mais recentes do Java?
 - Por que métodos acionados diretamente pelo método main, sem o uso de um objeto, precisam ser marcados como static?

Observações

Pré-requisitos:

1. Os estudantes precisam instalar o JDK e o NetBeans;
2. Também é necessário instalar o SQL Server e criar o banco de dados que foi solicitado na **Prática 2 – Vamos manter as informações.**

Referência Bibliográfica:

1. <https://stecine.azureedge.net/repositorio/02034/index.html>
2. <https://stecine.azureedge.net/repositorio/01425/index.html>
3. <https://stensineme.blob.core.windows.net/hmlgrehpoh/00212ti/01678/index.html>
4. Introdução ao middleware JDBC pela Dev Media. Disponível no endereço <https://www.devmedia.com.br/jdbc-tutorial/6638>. Acessado em 01/03/2023.
5. Padrões de Projeto. Disponível em <https://refactoring.guru/pt-br/design-patterns>. Acessado em 01/03/2023.
6. Os 4 pilares da Programação Orientada a Objetos. Disponível no endereço <https://www.devmedia.com.br/os-4-pilares-da-programacao-orientada-a-objetos/9264>. Acessado em 01/03/2023.

Entrega da prática

Chegou a hora, gamer!

✍️ Armazene o projeto em um repositório no GIT.

✍️ Anexar a documentação do projeto (PDF) no GIT.

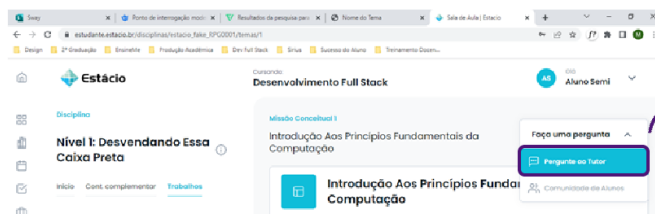
✍️ Compartilhe o link do repositório do GIT com o seu tutor para correção da prática, por meio da **Sala de Aula Virtual**, na aba "**Trabalhos**" do respectivo nível de conhecimento.

✍️ Ei, não se esqueça de entregar este trabalho na data estipulada no calendário acadêmico!



Dúvidas?

Não se preocupe! Entre em contato com o tutor do nível de conhecimento pela SAVA e tire suas dúvidas com ele!



Clique neste botão para falar com o tutor!