Desenvolvimento Web II

Aula 14 - Incorporando o Banco de Dados em Aplicações JSF - Parte 1

Apresentação

Olá, seja bem-vindo(a) à nossa aula! Vamos aprender como incorporar um banco de dados em um sistema de informação web desenvolvido com o framework JSF. Em aulas anteriores criamos um sistema de informação web de transportes urbanos que armazena seus dados em memória. Nesta aula e na próxima iremos melhorar essa aplicação incorporando uma base de dados, ou seja, nosso sistema passará de fato a persistir informações em um banco de dados.

Nesta primeira aula, aprenderemos como utilizar bancos de dados em aplicações JSF e criaremos uma classe Java para gerenciar as conexões com o banco de dados e as tabelas da aplicação. Podemos começar?

Objetivos

- Aprender como utilizar um banco de dados em um sistema de informação web desenvolvido com JSF;
- Aprender como implementar classes Java para conexão com bancos de dados.

Configurando a Utilização de Bancos de Dados em Aplicações JSF

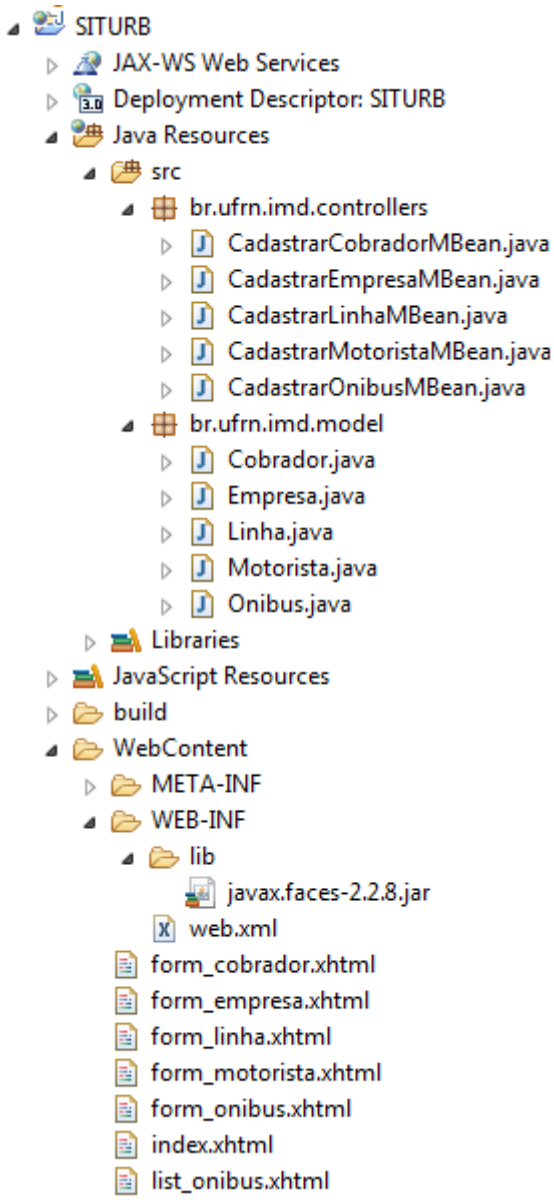
Primeiramente, vamos voltar ao nosso Sistema Integrado de Transportes Urbanos (SITURB) que tem o objetivo de cadastrar as empresas, linhas de ônibus, horários e funcionários (motorista e cobrador). Esse sistema tem sua estrutura apresentada na Figura 1 e é composto pelos seguintes pacotes:

- **br.ufrn.imd.model:** composta pelas classes do modelo Cobrador, Empresa, Linha, Motorista e Onibus;

- **br.ufrn.imd.controllers:** composto pelos managed beans CadastrarCobradorMBean, CadastrarEmpresaMBean, CadastrarLinhaMBean, CadastrarMotoristaMBean e CadastrarOnibusMBean.

Ainda nesse projeto, temos as views: form_cobrador.xhtml, form_empresa.xhtml, form_linha.xhtml, form_motorista.xhtml, form_onibus.xhtml, list_onibus.xhtml e index.html. Dessa forma, vamos iniciar incluindo o conector para o banco de dados MySQL, mysql-connector-java-8.0.11.jar clicando com o botão direto no projeto, escolhendo a opção “Properties, clicando a opção “Java Build Path” na esquerda, selecionando a aba Libraties, depois clique em Add External jar e escolha o arquivo mysql-connector-java-8.0.11.jar que você salvou no computador. Esse conector tem o objetivo de possibilitar a conexão da nossa aplicação com o banco de dados MySQL. Adicione também na pasta WEB-INF/lib o arquivo javax.faces-2.2.8.jar.

Figura 01 - Estrutura do projeto SITURB.



Incluído o conector, vamos criar um pacote denominado br.ufrn.imd.dao e definir uma classe para realizar a conexão com nossa base de dados. Essa classe denominada GerenciadorConexao é apresentada na Listagem 1 e tem como objetivo fornecer uma conexão para que possamos realizar as alterações no banco de dados.

```
1  package br.ufrn.imd.dao;
2
3  import java.sql.Connection;
4  import java.sql.DriverManager;
5
6  /**
7  * Classe responsável por fornecedor conexão com o banco de dados para a
8  * aplicação.
9  *
10 * @author itamir.filho
11 */
12 public class GerenciadorConexao {
13
14    private static Connection conexao;
15
16    /**
17       * Método estático para obtenção de conexão.
18       *
19       * @return
20       */
21    public static Connection getConexao() {
22
23       if (conexao == null) {
24         String username = "root";
25         String password = "123456";
26         // Informa a URL do banco (siturb) e o timezone do servidor
27         String url = "jdbc:mysql://localhost/siturb?serverTimezone=UTC";
28         try {
29              Class.forName("com.mysql.cj.jdbc.Driver");
30              conexao = DriverManager.getConnection(url, username, password);
31         } catch (Exception e) {
32              e.printStackTrace();
33         }
34       }
35    return conexao;
36    }
37 }
```

Listagem 1 - Classe para obtenção da conexão com o banco de dados.

Observe que estamos utilizando o Driver de conexão com o MySQL que está na nossa máquina (localhost) e o banco de dados com o nome “siturb”, especificados na variável URL presente na Listagem 1. Além disso, atente-se que só teremos uma conexão com o banco de dados, que será inicializada uma única vez quando a mesma for nula. Isso é importante, pois otimizamos a quantidade de conexões da nossa aplicação com o banco de dados.

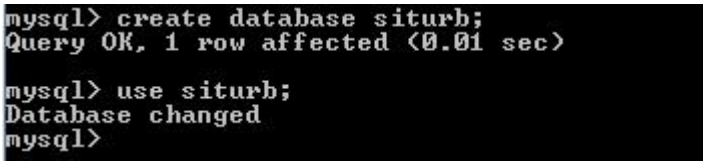
Criando as Tabelas no Banco de Dados para Armazenamento das Informações

Como já definimos nossa classe para gerenciar a conexão com o nosso banco de dados, presente na Listagem 1, vamos agora criar nosso banco de dados e as tabelas referentes ao modelo da nossa aplicação JSF. Assim, teremos as seguintes tabelas: cobrador, empresa, linha, motorista e onibus. Começaremos com o comando para conectar no servidor do banco de dados MySQL, apresentado na Figura 2. Observe que os parâmetros “user” e “password” devem ser preenchidos com o usuário e senha que você definiu na instalação do MySQL.

Figura 02 - Comando para se conectar ao servidor de banco de dados MySQL.

Conectado no servidor MySQL, execute os comandos para criação e utilização do banco de dados “siturb” que conterá as tabelas referentes ao modelo da nossa aplicação. Esses comandos são apresentados na Figura 3.

Figura 03 - Comandos para criar e utilizar o banco de dados siturb.



Definido o banco de dados, criaremos as tabelas. Os scripts para criação das tabelas são: cobrador, empresa, linha, motorista e onibus, apresentados respectivamente nas Figuras 4, 5, 6, 7 e 8.

Figura 04 - Script para criação da tabela cobrador.

Figura 05 - Script para criação da tabela empresa.

Figura 06 - Script para criação da tabela linha.

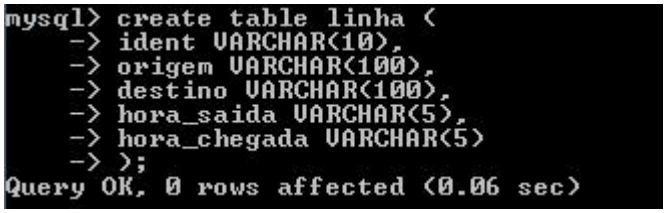
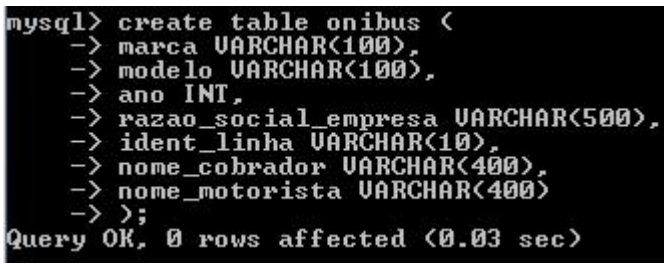


Figura 07 - Script para criação da tabela motorista.

Figura 08 - Script para criação da tabela onibus.



Pronto, já temos nossas tabelas criadas, o que pode ser comprovado pelo comando apresentado na Figura 9.

Figura 09 - Execução de comando para apresentar tabelas criadas no banco de dados da aplicação SITURB.

Vamos incluir informações no nosso banco de dados? Começaremos com a inserção de um cobrador com o seguinte comando:

- insert into cobrador (nome, cpf, matricula, endereco) values ('Peter Parker', '111.111.111-11', '111011-1', 'Rua das Casas, NY.');

Um motorista, com o seguinte comando:

- insert into motorista (nome, cpf, matricula, endereco, registro_cnh, categoria_cnh) values ('Tony Stark', '222.222.222-22', '121121-2', 'Malibu', '11123', 'AE');

Uma empresa, com o seguinte comando:

- insert into empresa (razao_social, cnpj) values ('Empresas Stark', '26.118.781/0001-07');

Uma linha, com o seguinte comando:

- insert into linha (ident, origem, destino, hora_saida, hora_chegada) values ('49', 'Rodoviária', 'Midway', '11:00', '12:30');

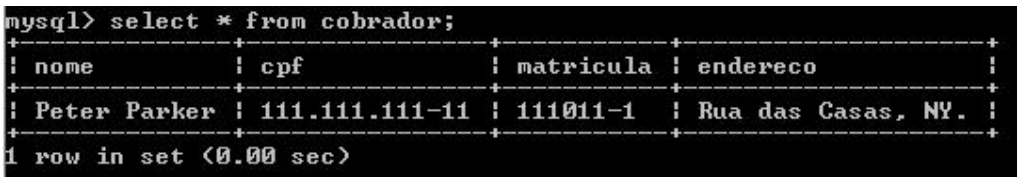
Um ônibus, com o seguinte comando:

- insert into onibus (marca, modelo, ano, razao_social_empresa, ident_linha, nome_cobrador, nome_motorista) values ('0371', 'Mercedes', 1999, 'Empresas Stark', '49', 'Peter Parker', 'Tony Stark');

Além desses dados sugeridos, use sua criatividade para criar mais empresas, linhas, cobradores, motoristas e ônibus. Para consultar os dados inseridos, vamos realizar comandos de seleção. Para selecionar todos os cobradores inseridos: “select

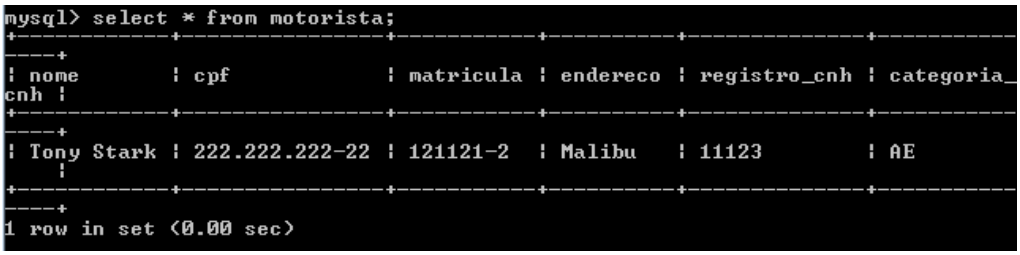
* from cobrador;". Esse comando e sua execução é apresentado na Figura 10.

Figura 10 - Select de todos os cobradores do banco de dados.



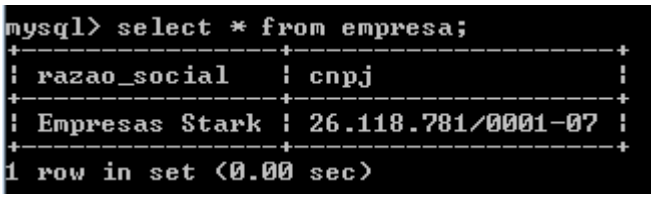
Para selecionar todos os motoristas inseridos: "select * from motorista;". Esse comando e sua execução é apresentado na Figura 11.

Figura 11 - Select de todos os motoristas do banco de dados.



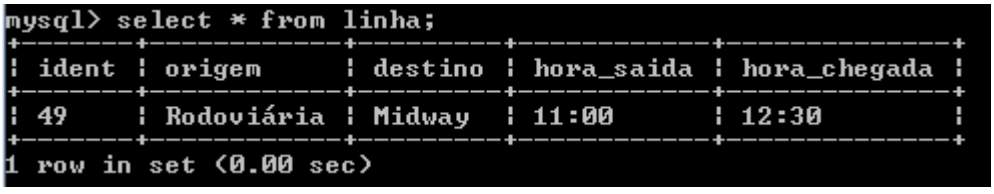
Para selecionar todas as empresas inseridas: "select * from empresa;". Esse comando e sua execução é apresentado na Figura 12.

Figura 12 - Select de todas as empresas do banco de dados.



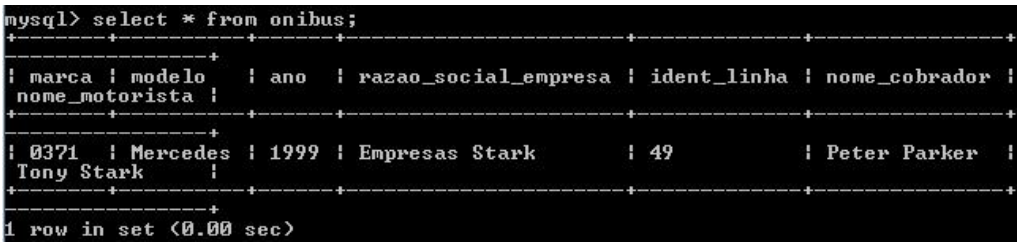
Para selecionar todas as linhas inseridas: "select * from linha;". Esse comando e sua execução é apresentado na Figura 13.

Figura 13 - Select de todas as linhas do banco de dados.



Para selecionar todas os ônibus inseridos: "select * from onibus;". Esse comando e sua execução é apresentado na Figura 14.

Figura 14 - Select de todas os ônibus do banco de dados.



Pronto! Agora já estamos com a classe gerenciadora de conexões com nosso banco de dados e tabelas criados. Na próxima aula, continuaremos esse projeto desenvolvendo as classes para realização de consultas e inserções pela aplicação JSF. Interessante, não é? Estamos apenas começando. Até a próxima aula e bons estudos!

Atividade 02

---

1. Realize a inserção de 4 motoristas, 4 cobradores, 6 linhas de ônibus, 3 empresas e 10 ônibus no banco de dados **siturb**.
2. Verifique os dados inseridos através de comandos de seleção (**select**).

Resumo

Nessa aula retomamos o desenvolvimento da nossa aplicação JSF implementada na aula 19, denominada Sistema Integrado de Transportes Urbanos (SITURB) e ainda realizamos o desenvolvimento de uma classe Java gerenciadora de conexões com o banco de dados. Além disso, criamos um banco de dados denominado siturb e suas tabelas para armazenamento das informações dessa aplicação. Por fim, inserimos informações nessas tabelas e as consultamos para exercitar os comandos de inserção e seleção em bancos de dados.

Referências

<http://docs.oracle.com/javase/8/docs/api/java/sql/PreparedStatement.html>.

<http://dev.mysql.com/downloads/connector/j/>.

<http://dev.mysql.com/doc/refman/5.6/en/index.html>.

<http://dev.mysql.com/doc/refman/5.6/en/sql-syntax.html>.