

Programação WEB

Guia para uso de Banco de Dados no projeto Java web no Netbeans para fazer acesso de Login

Osvaldo Cesar Pinheiro de Almeida
(osvaldo.almeida@fatec.sp.gov.br)

Botucatu - SP
Setembro - 2022

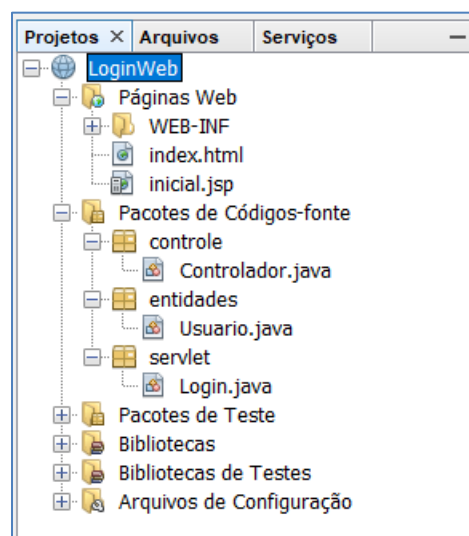
Inicialização

Requisitos iniciais:

- Seguir previamente o Guia de Criação de Projeto web de Login;
- Netbeans completo e Java (jdk) instalado - Versão recomendada do Netbeans 8.2 ou maior;
- Observe as imagens com destaque em vermelho.

Usando o Projeto

Vamos utilizar o projeto criado anteriormente no "Guia de Criação de Projeto web de Login", como ilustrado na figura abaixo:



Usando um SGBD

Iremos utilizar nesse guia o sistema de gerenciador de banco de dados (SGBD) MySQL. Caso você já possua o MySQL instalado utilize-o, senão instale o MySQL em seu micro. Pode ser uma versão da Internet, retirada diretamente do site oficial do MySQL (Lembrando que o SGBD é gratuito para ensino). Ou utilize a versão que está na pasta Z (pasta de disciplinas da faculdade).

Criação da Base de Dados de Exemplo

Para o nosso exemplo criaremos uma base de dados, chamada "aulalweb" e criaremos também uma tabela chamada Usuário. Para isso usaremos os seguintes comandos:

#Criação da base de dados

```
create database pweb;
```

#Definir o uso da base

```
use pweb;
```

#Criação da tabela Usuario

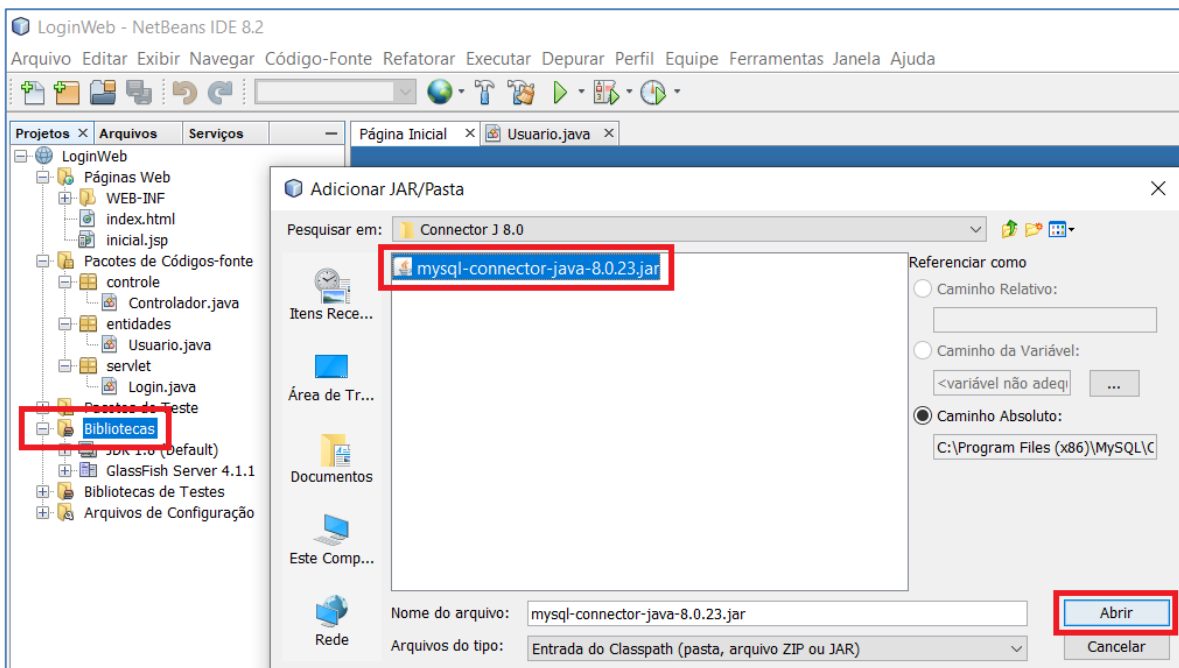
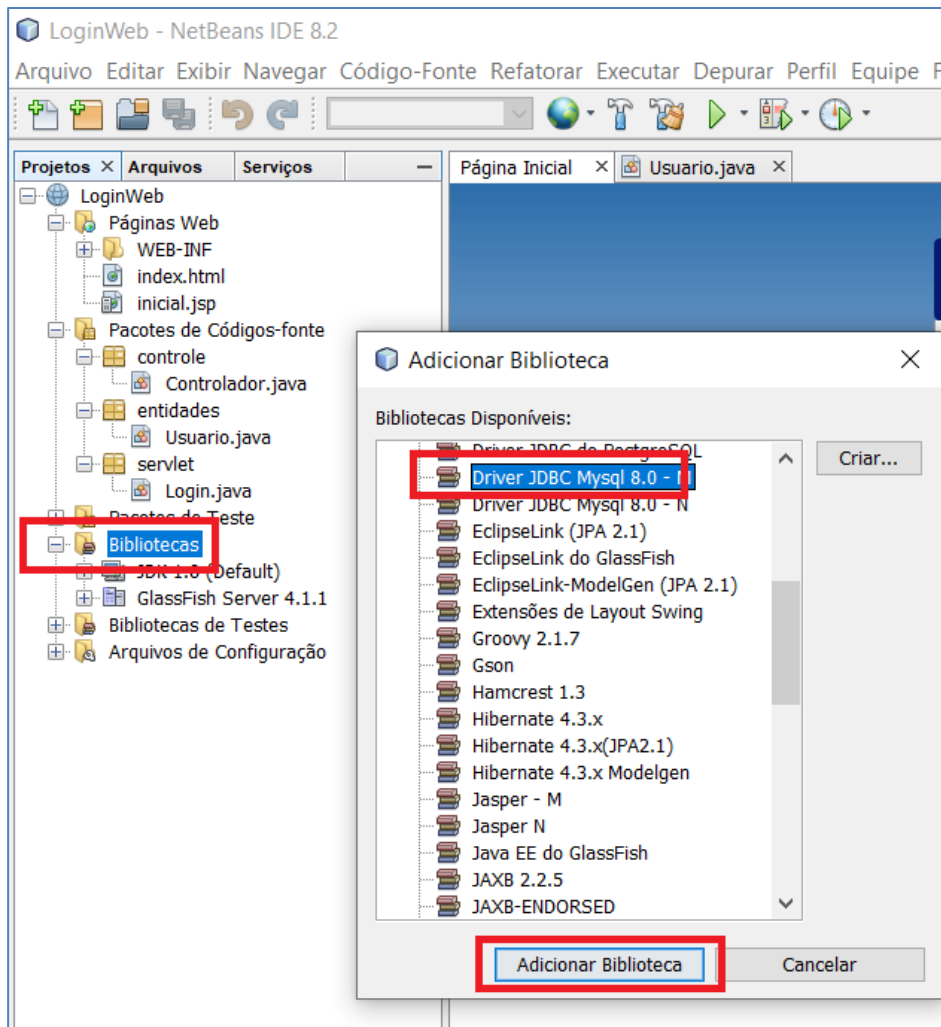
```
create table usuario(  
cod integer primary key auto_increment,  
nome varchar(50),  
login varchar(20),  
senha varchar(32));
```

Configurando o Projeto

Clicando com o botão direito em cima do nome "bibliotecas" (libraries), aparecerá uma opção para adicionar uma nova biblioteca (library). Adicione a biblioteca do Mysql (verificar a versão do MySQL – 5 ou 8).

A adição dessa biblioteca vai permitir que o sistema faça acesso à base de dados Mysql.

Caso a biblioteca do MySQL 8 não esteja presente é possível fazer a criação da biblioteca, para uso futuro, ou incluir diretamente o “.jar” da biblioteca.



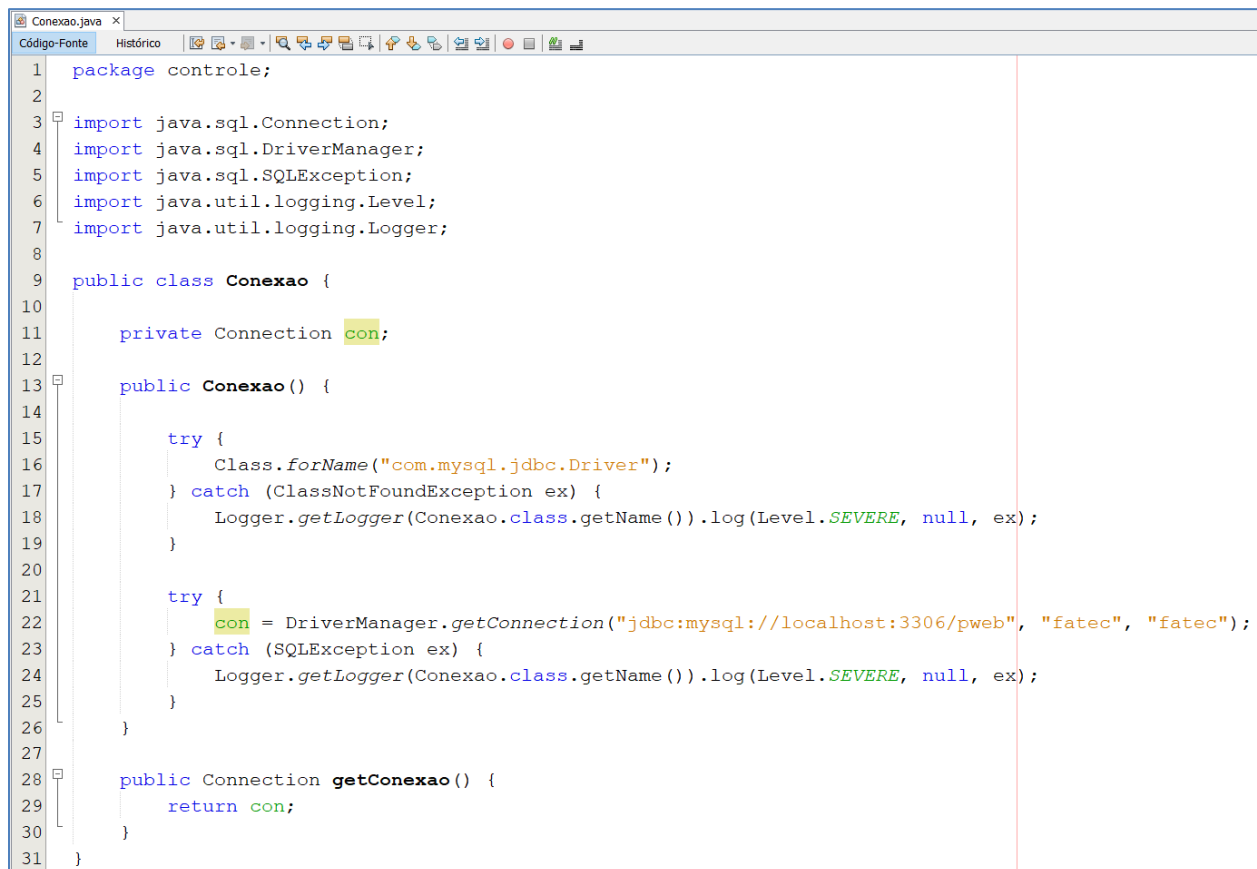
Criação de Classes

Vamos fazer a criação de duas novas classes:

- Conexao;
- Cadastro.

Classe Conexao

Deve ser criada uma classe Java chamada Conexao. Essa classe deve ser implementada de maneira que o código fique como no exemplo abaixo:



```
1 package controle;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.SQLException;
6 import java.util.logging.Level;
7 import java.util.logging.Logger;
8
9 public class Conexao {
10
11     private Connection con;
12
13     public Conexao() {
14
15         try {
16             Class.forName("com.mysql.jdbc.Driver");
17         } catch (ClassNotFoundException ex) {
18             Logger.getLogger(Conexao.class.getName()).log(Level.SEVERE, null, ex);
19         }
20
21         try {
22             con = DriverManager.getConnection("jdbc:mysql://localhost:3306/pweb", "fatec", "fatec");
23         } catch (SQLException ex) {
24             Logger.getLogger(Conexao.class.getName()).log(Level.SEVERE, null, ex);
25         }
26     }
27
28     public Connection getConexao() {
29         return con;
30     }
31 }
```

Na linha 16, onde aparece o código: `Class.forName("com.mysql.jdbc.Driver");`. O nome "`com.mysql.jdbc.Driver`" representa o nome do driver de conexão para o Mysql.

Na linha 22, onde aparece o código:

`con = DriverManager.getConnection("jdbc:mysql://localhost:3306/pweb", "fatec", "fatec");`. No primeiro parâmetro estão definidos o SGBD usado ("mysql"), o endereço do servidor ("localhost:3306"), e a base que será usada ("pweb"). O segundo parâmetro define o usuário que irá acessar a base. Enquanto o último parâmetro define a senha.

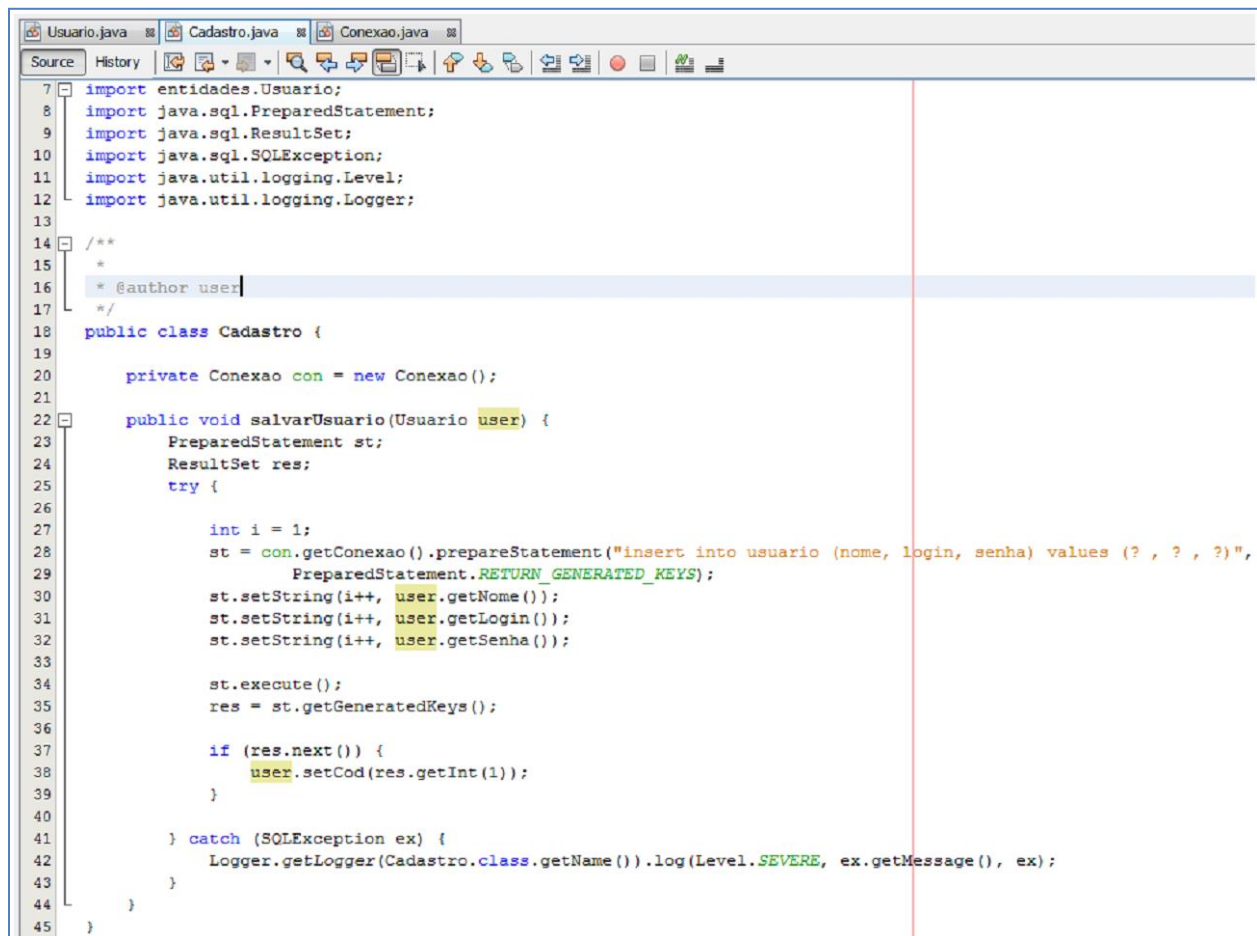
Essa classe Conexao servirá como meio de criação de conexão com o banco de dados Mysql.

Classe Cadastro

Deve ser criada uma classe Java chamada Cadastro. Nessa classe serão implementados os métodos de manipulação do banco de dados. Para isso, serão criados como exemplos dois métodos nessa classe:

- consultarUsuario;
- salvarUsuario.

A classe Cadastro com o primeiro método para salvar um objeto do tipo Usuário no banco deve ficar como no exemplo abaixo:



```
1 import entidades.Usuario;
2 import java.sql.PreparedStatement;
3 import java.sql.ResultSet;
4 import java.sql.SQLException;
5 import java.util.logging.Level;
6 import java.util.logging.Logger;
7
8 /**
9  * @author user
10 */
11 public class Cadastro {
12
13     private Conexao con = new Conexao();
14
15     public void salvarUsuario(Usuario user) {
16         PreparedStatement st;
17         ResultSet res;
18         try {
19
20             int i = 1;
21             st = con.getConnection().prepareStatement("insert into usuario (nome, login, senha) values (?, ?, ?)",
22                 PreparedStatement.RETURN_GENERATED_KEYS);
23             st.setString(i++, user.getNome());
24             st.setString(i++, user.getLogin());
25             st.setString(i++, user.getSenha());
26
27             st.execute();
28             res = st.getGeneratedKeys();
29
30             if (res.next()) {
31                 user.setCod(res.getInt(1));
32             }
33
34         } catch (SQLException ex) {
35             Logger.getLogger(Cadastro.class.getName()).log(Level.SEVERE, ex.getMessage(), ex);
36         }
37     }
38 }
```

Os códigos das linhas 25 à 38 representam:

```
// Define qual será o comando SQL que será executado
28     st = con.getConnection().prepareStatement("insert into usuario (nome, login, senha) values
        (?, ?, ?)", PreparedStatement.RETURN_GENERATED_KEYS);

//As três linhas definem os parâmetros que substituirão as três interrogações ("?", "?", "?") presentes
//no comando sql "(?, ?, ?)"
30     st.setString(i++, user.getNome());
31     st.setString(i++, user.getLogin());
32     st.setString(i++, user.getSenha());

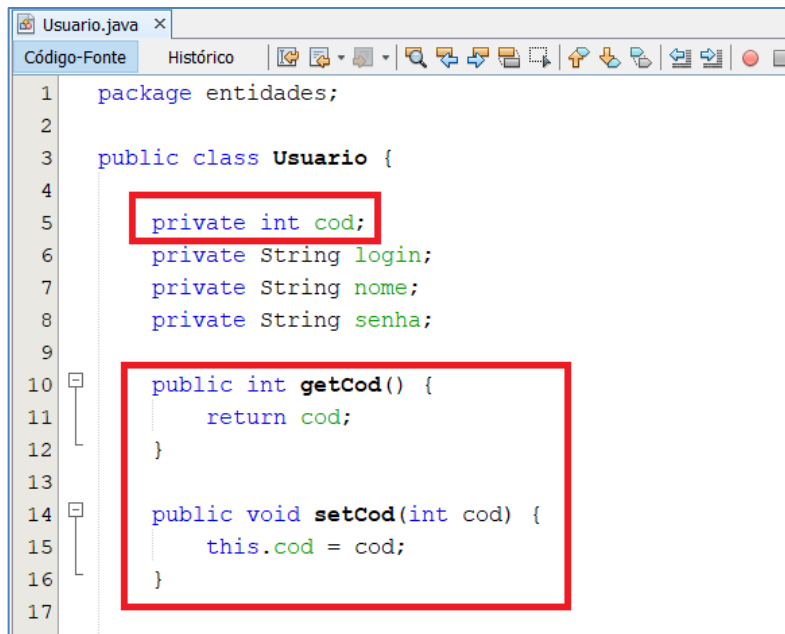
//Execução do comando no banco de dados
34     st.execute();

//Recuperação do identificador (cod) único para cada Usuário
35     res = st.getGeneratedKeys();

//Verifica se o identificador foi recuperado
37     if (res.next()) {

//Recupera o identificador (cod)
38         pes.setCod(res.getInt(1));
39     }
```

Como verificado no código acima, há uma menção à um identificador único por Usuário (cod). Esse identificador será controlado automaticamente pelo banco de dados. Contudo, é necessário criar o atributo para referenciá-lo na classe Java Usuário. Dessa maneira, devemos criar o atributo *cod*, além de seus respectivos métodos *get* e *set*, como no exemplo abaixo:



```
1 package entidades;
2
3 public class Usuario {
4     private int cod;
5     private String login;
6     private String nome;
7     private String senha;
8
9
10    public int getCod() {
11        return cod;
12    }
13
14    public void setCod(int cod) {
15        this.cod = cod;
16    }
17 }
```

Após essa modificação, podemos completar os demais métodos da classe Cadastro.

Método para consultar Usuário:



```
49 public Usuario consultarUsuario(String login) {
50     Usuario user = null;
51     PreparedStatement st;
52     ResultSet res;
53     int i = 1;
54
55     try {
56         st = con.getConnection().prepareStatement("select cod, nome, login, senha "
57             + "from usuario where login = ? ");
58         st.setString(i, login);
59
60         res = st.executeQuery();
61
62         if(res.next()){
63             user = new Usuario();
64             user.setCod(res.getInt("cod"));
65             user.setNome(res.getString("nome"));
66             user.setLogin(res.getString("login"));
67             user.setSenha(res.getString("senha"));
68         }
69     }
70
71     catch (SQLException ex) {
72         Logger.getLogger(Cadastro.class.getName()).log(Level.SEVERE, null, ex);
73     }
74
75     return user;
76 }
77 }
```


O método acima faz uma consulta no banco de dados selecionando o usuário baseado no login passado como parâmetro. Caso o usuário exista no banco, todos os seus dados serão retornados.

Atualizando o Projeto

Após as alterações nas duas classes é necessário fazer algumas alterações na classe Controlador para que passamos executar os métodos do banco de dados. Assim, serão acrescentados os códigos marcados nos exemplos abaixo.

Acrescentando o atributo da classe Cadastro e consultando o Usuário no processo de Login:

```
10 public class Controlador {
11
12     private Cadastro cad = new Cadastro();
13
14     public Usuario validaUsuario(Usuario user) {
15
16         Usuario novo = cad.consultarUsuario(user.getLogin());
17
18         if (novo != null && user.getLogin().equals(novo.getLogin()) &&
19             user.getSenha().equals(novo.getSenha())) {
20             return novo;
21         }
22         return null;
23     }
24 }
```

Após a implementação das alterações acrescente um usuário na base de dados e faça um teste com o programa.

Bom trabalho!!!