

Nesta aula, iremos trabalhar a validação do campo email, para que não seja possível cadastrar um novo usuário com um email já cadastrado na base de dados.

Como precisaremos fazer um SELECT no banco para verificar se o valor do campo email existe, voltamos a necessitar da função executeQuery.

Na página restrita.jsp, posicione o local de inserção da consulta:

```
171 try {
172     st = new Conexao().conectar().createStatement();
173     //insira aqui o executeQuery para pesquisar um valor no BD
174
175     //fim da consulta
176     st.executeUpdate("INSERT INTO usuarios (nome, email, senha, nivel) "
177         + "VALUES ('" + nome + "','" + email + "','" + senha + "','" + nivel + "'
178     response.sendRedirect("restrita.jsp");
```

Insira o seguinte trecho de código:

```
172 //insira aqui o executeQuery para pesquisar um valor no BD
173 rs = st.executeQuery("SELECT * FROM usuarios WHERE email = '" + email + "' ");
174 while (rs.next()) {
175     rs.getRow(); //recupera quantos registros achou com o email
176     if (rs.getRow() > 0){
177         out.print("Usuário já cadastrado!");
178         return; //mostra a mensagem e sai do método
179     }
180 }
181 //fim da consulta
182 st.executeUpdate("INSERT INTO usuarios (nome, email, senha, nivel) "
183     + "VALUES ('" + nome + "','" + email + "','" + senha + "','" + nivel + "'
184 response.sendRedirect("restrita.jsp");
```

Explicando o código:

173 ➔ é feita uma consulta no banco de dados para procurar no campo email o valor digitado no formulário de cadastro.

174 ➔ O resultado da consulta armazenado em rs, é colocado em um laço de repetição para que toda a tabela seja percorrida até o final.

175 ➔ rs.getRow() armazena o número de linhas que foram encontradas atendendo a instrução SQL da linha 173. É um número inteiro.

176 ➔ Verificamos se o valor retornado é maior que 0. Se for 0 significa que não foram encontrados registros com esse email, mas se o valor for diferente de 0, ou seja, maior, deve ser enviada uma mensagem informando que esse email já existe na base.

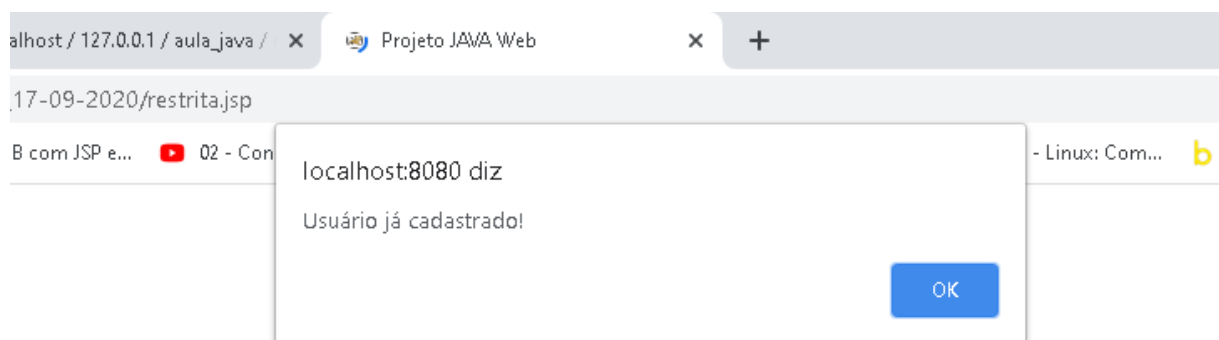
177 ➔ Mensagem para o usuário informando que já existe um registro cadastrado.

178 ➔ o comando return, faz com que a execução encerre e sai do método. Se essa função não é colocada, ao terminar o if, o programa faz as próximas instruções, ou seja, irá cadastrar mesmo encontrando registros iguais. Portanto, o uso desta função é importante para encerrar os trechos seguintes.

Podemos melhorar um pouco mais a visualização de Usuário Já Cadastrado utilizando, por exemplo, um alert do Java Script. Veja:

```
out.print("<script>alert('Usuário já cadastrado!');</script>");
```

Tome cuidado no uso das aspas.



Chegamos na metade do nosso projeto. Como atividade prática, vocês devem enviar o projeto feito até aqui através do link fornecido em aula pelo professor.

Na próxima aula veremos como codificar o botão para buscar um usuário no banco de dados.