

Nesta aula, iremos trabalhar a busca por um nome no banco de dados programando o botão buscar.

No componente **button** ao lado da caixa de input, coloque uma propriedade **name** com o valor **btn\_buscar**.

```
74 | <input class="form-control mr-sm-2" type="search" name="txtbuscar"
75 |     placeholder="Digite um nome" aria-label="Search">
76 | <button class="btn btn-outline-info my-2 my-sm-0" name="btn_buscar" type="submit">Buscar</button>
```

O componente **input** também deve ter um **name** associado à ele com o valor **txtbuscar**.

```
73 | <form class="form-inline my-2 my-lg-0">
74 |     <input class="form-control mr-sm-2" type="search" name="txtbuscar"
75 |         placeholder="Digite um nome" aria-label="Search">
76 |     <button class="btn btn-outline-info my-2 my-sm-0" name="btn_buscar" type="submit">Buscar</button>
77 | </form>
```

Vamos alterar o código no momento em que a tabela é gerada, verificando se há uma requisição do botão buscar, caso contrário ele continua a montagem da tabela normalmente.

```
89 | <tbody>
90 |     <%
91 |         Statement st = null;
92 |         ResultSet rs = null;
93 |         try {
94 |             st = new Conexao().conectar().createStatement();
95 |             if (request.getParameter("btn_buscar") != null) {
96 |                 String busca = '%' + request.getParameter("txtbuscar") + '%';
97 |                 rs = st.executeQuery("SELECT * FROM usuarios WHERE nome LIKE '" + busca + "'");
98 |             }
99 |             else{
100 |                 rs = st.executeQuery("SELECT * FROM usuarios ");
101 |             }

```

```
102 |         while (rs.next()) {
103 |             <%
104 |                 >%
105 |                 <tr>
106 |                     <td><%= rs.getString(1) %></td>
107 |                     <td><%= rs.getString(2) %></td>
108 |                     <td><%= rs.getString(3) %></td>
109 |                     <td><%= rs.getString(4) %></td>
110 |                     <td><%= rs.getString(5) %></td>
111 |                 </tr>

```

Explicação das linhas modificadas:

95: verifica se há uma requisição do botão buscar

96: Cria a *string* de busca. Os termos “%” antes e depois concatenados, significa que vamos buscar termos aproximados, ou seja, que contenham o texto digitado.

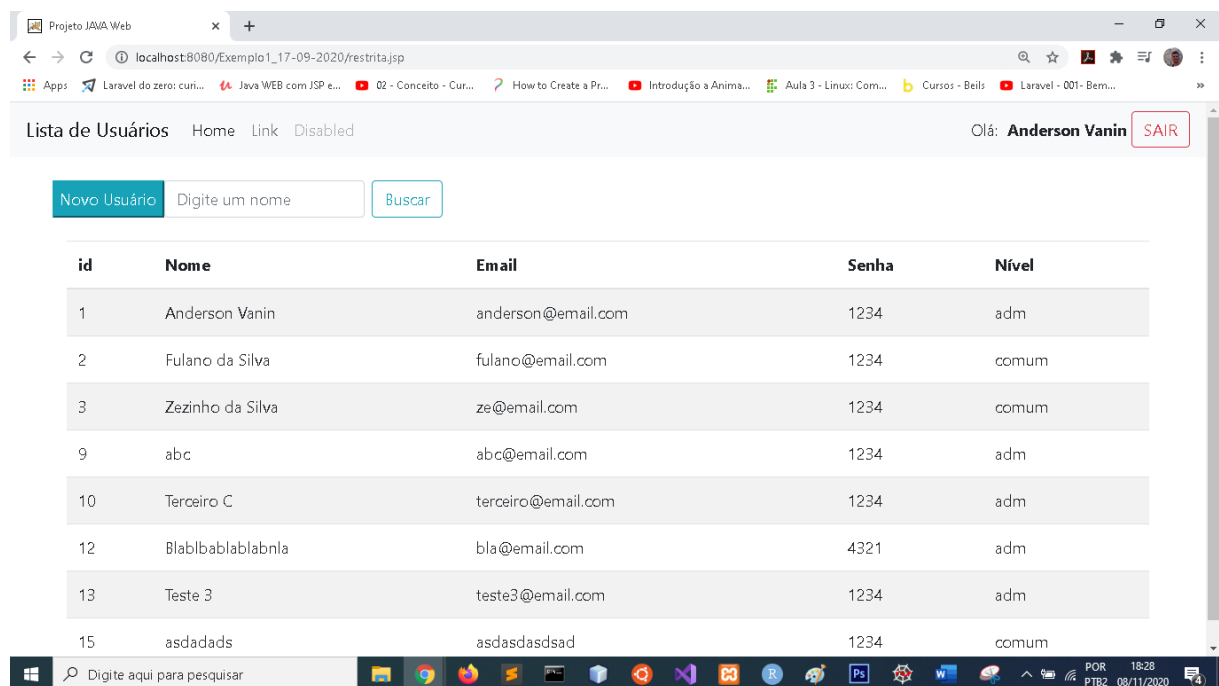
97: resultado da consulta com a busca

100: faz a busca por todos os usuários caso não haja requisição do botão buscar

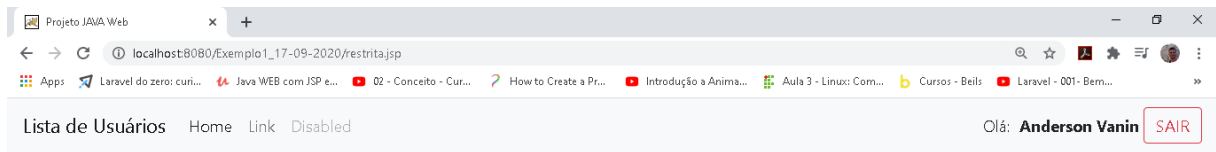
Faça um último ajuste no formulário. Adicione o método = *post*, para que os últimos acessos à esse formulário não fiquem aparecendo na *url* impossibilitando que a página seja novamente exibida com todos os dados originais.

```
73 <form class="form-inline my-2 my-lg-0" method="post">
74   <input class="form-control mr-sm-2" type="search" name="txtbuscar"
75     placeholder="Digite um nome" aria-label="Search">
76   <button class="btn btn-outline-info my-2 my-sm-0" name="btn_buscar" type="submit">Buscar</button>
77 </form>
```

Teste seu projeto e veja que caso não haja nenhuma informação no input buscar, ele irá fazer a listagem normalmente.



id	Nome	Email	Senha	Nível
1	Anderson Vanin	anderson@email.com	1234	adm
2	Fulano da Silva	fulano@email.com	1234	comum
3	Zezinho da Silva	ze@email.com	1234	comum
9	abc	abc@email.com	1234	adm
10	Terceiro C	terceiro@email.com	1234	adm
12	Blablablablabnla	bla@email.com	4321	adm
13	Teste 3	teste3@email.com	1234	adm
15	asdadads	asdasdasdsad	1234	comum



id	Nome	Email	Senha	Nível
2	Fulano da Silva	fulano@email.com	1234	comum



Feito isso, vamos deixar preparado para a próxima aula dois botões: Excluir e Alterar, que devem aparecer em cada uma das linhas dos registros da tabela para que possamos fazer outras modificações nos registros.

Vamos adicionar uma nova coluna na tabela chamada Ações.

```

84 |      <th scope="col">Email</th>
85 |      <th scope="col">Senha</th>
86 |      <th scope="col">Nível</th>
87 |      <th scope="col">Ações</th>
88 |    </tr>
89 |  </thead>

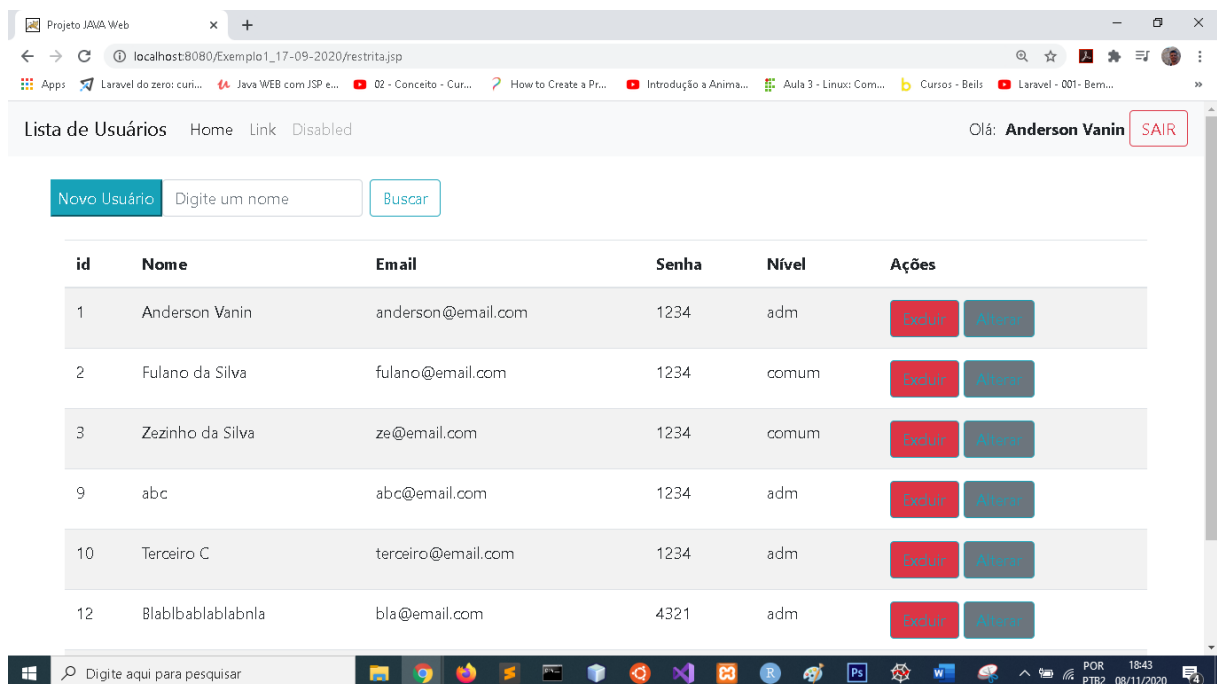
```

Dentro do laço de repetição vamos inserir esses dois botões que irão abrir um modal, cada um, para executar a ação escolhida.

```

106 |    <tr>
107 |      <td><%= rs.getString(1) %></td>
108 |      <td><%= rs.getString(2) %></td>
109 |      <td><%= rs.getString(3) %></td>
110 |      <td><%= rs.getString(4) %></td>
111 |      <td><%= rs.getString(5) %></td>
112 |      <td>
113 |        <a href="#" class="btn btn-outline-info my-2 my-sm-0 btn-danger">Excluir</a>
114 |        <a href="#" class="btn btn-outline-info my-2 my-sm-0 btn-secondary">Alterar</a>
115 |      </td>
116 |    </tr>

```



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/Exemplo1\_17-09-2020/restrita.jsp'. The page title is 'Projeto JAVA Web'. The main content area is titled 'Lista de Usuários' and includes a navigation bar with 'Home', 'Link', and 'Disabled'. A greeting 'Olá: Anderson Vanin' is displayed next to a 'SAIR' button. Below the greeting is a search bar with the placeholder 'Digite um nome' and a 'Buscar' button. A table lists users with columns for 'id', 'Nome', 'Email', 'Senha', 'Nível', and 'Ações'. The table contains 7 rows of user data. The 'Ações' column for each row contains two buttons: 'Excluir' (red) and 'Atualizar' (blue). The Windows taskbar at the bottom shows the search bar and various application icons.

id	Nome	Email	Senha	Nível	Ações
1	Anderson Vanin	anderson@email.com	1234	adm	Excluir Atualizar
2	Fulano da Silva	fulano@email.com	1234	comum	Excluir Atualizar
3	Zezinho da Silva	ze@email.com	1234	comum	Excluir Atualizar
9	abc	abc@email.com	1234	adm	Excluir Atualizar
10	Terceiro C	terceiro@email.com	1234	adm	Excluir Atualizar
12	Blablablablabnla	bla@email.com	4321	adm	Excluir Atualizar

Na próxima aula faremos a codificação desses dois botões.