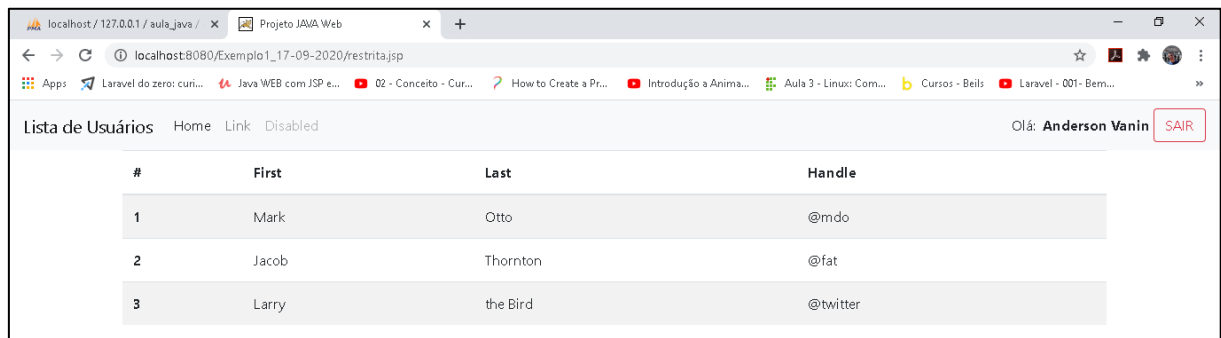


Agora que temos a estrutura de nossa visualização pronta, vamos começar a fazer a programação do arquivo para que sejam listadas as informações da tabela Usuários em nossa página.

Essa é a aparência de nossa tela restrita.jsp:



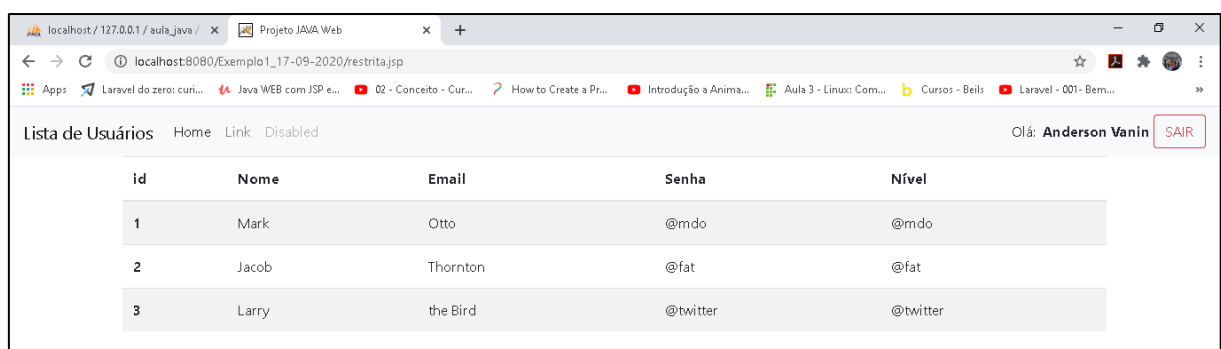
#	First	Last	Handle
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter

Essa é a estrutura de nosso banco de dados com a tabela usuários:



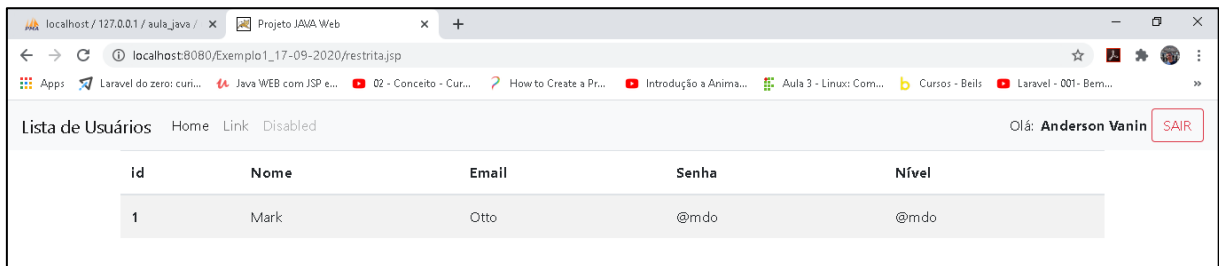
	id	nome	email	senha	nível
<input type="checkbox"/>	1	Anderson Vanin	anderson@email.com	1234	adm
<input type="checkbox"/>	2	Fulano da Silva	fulano@email.com	1234	comum

Observe que precisamos colocar no cabeçalho da tabela da página web, os nomes dos campos que iremos apresentar (id, nome, email, senha e nível). Faça a alteração no código html.



id	Nome	Email	Senha	Nível
1	Mark	Otto	@mdo	@mdo
2	Jacob	Thornton	@fat	@fat
3	Larry	the Bird	@twitter	@twitter

Os dados da tabela, serão buscados através de um laço de repetição, ou seja, cada linha da tabela (tr) será um resultado do laço. Altere a estrutura para que mostre somente uma linha na tabela.

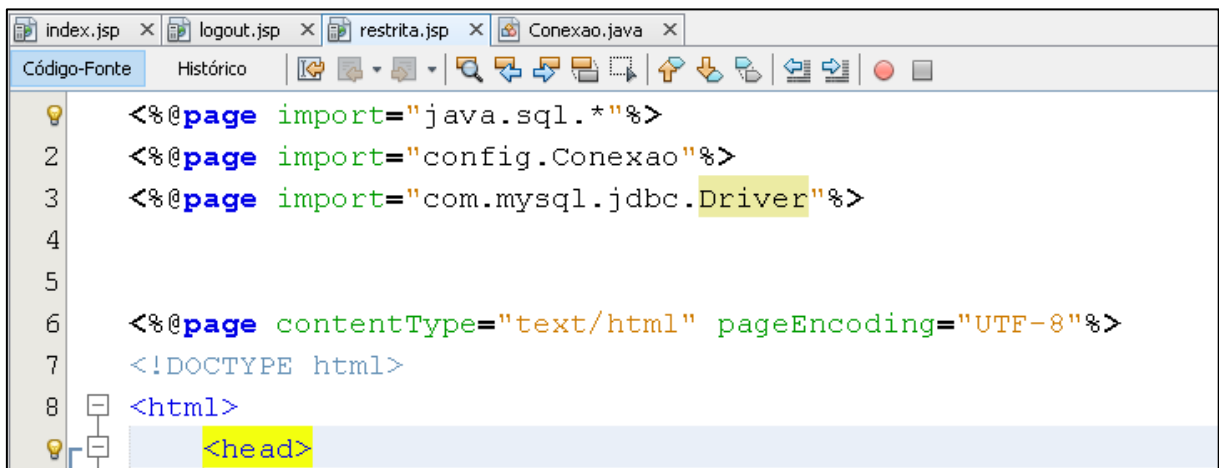


Lista de Usuários Home Link Disabled Olá: Anderson Vanin SAIR

id	Nome	Email	Senha	Nível
1	Mark	Otto	@mdo	@mdo

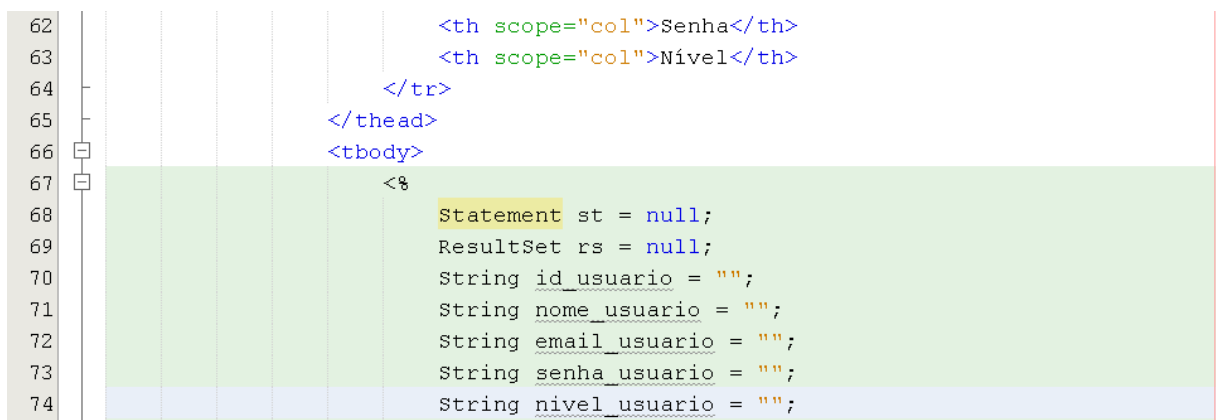
Agora vamos ao código.

O primeiro passo, é fazer a importação das classes que iremos precisar para a conexão com o banco de dados.



```
index.jsp x logout.jsp x restrita.jsp x Conexao.java x
Código-Fonte Histórico
1 <%@page import="java.sql.*"%>
2 <%@page import="config.Conexao"%>
3 <%@page import="com.mysql.jdbc.Driver"%>
4
5
6 <%@page contentType="text/html" pageEncoding="UTF-8"%>
7 <!DOCTYPE html>
8 <html>
9 <head>
```

Agora precisamos inserir no código a chamada para a conexão. Isso deve ser feito antes da linha da tabela que iremos repetir (<tr>).



```
62 <th scope="col">Senha</th>
63 <th scope="col">Nível</th>
64 </tr>
65 </thead>
66 <tbody>
67 <%
68     Statement st = null;
69     ResultSet rs = null;
70     String id_usuario = "";
71     String nome_usuario = "";
72     String email_usuario = "";
73     String senha_usuario = "";
74     String nivel_usuario = "";
```

```

74 String nivel_usuario = "";
75
76 try {
77
78     st = new Conexao().conectar().createStatement();
79     rs = st.executeQuery("SELECT * FROM usuarios ");
80     while (rs.next()) {
81         //AQUI COLOCAREMOS A LINHA DA TABELA
82         //QUE IRÁ RECEBER O RESULTADO DO LAÇO
83         //DE REPETIÇÃO!!
84     }
85 } catch (Exception e) {
86     out.println(e);
87 }
88 %>
89
90 <tr>
91     <th scope="row">1</th>
92     <td>Mark</td>
93     <td>Otto</td>
94     <td>@mdo</td>
95     <td>@mdo</td>
96 </tr>
97 </tbody>
98 </table>
99 </div>
100 </body>
101 </html>

```

Para inserir o html que faz a linha, precisamos parar o script, inserir o html e depois abrir o script novamente para que ele consiga finalizar o laço de repetição.

```

while (rs.next()) {
    %>
    <tr>
        <td>1</td>
        <td>Mark</td>
        <td>Otto</td>
        <td>@mdo</td>
        <td>@mdo</td>
    </tr>
    <%
}

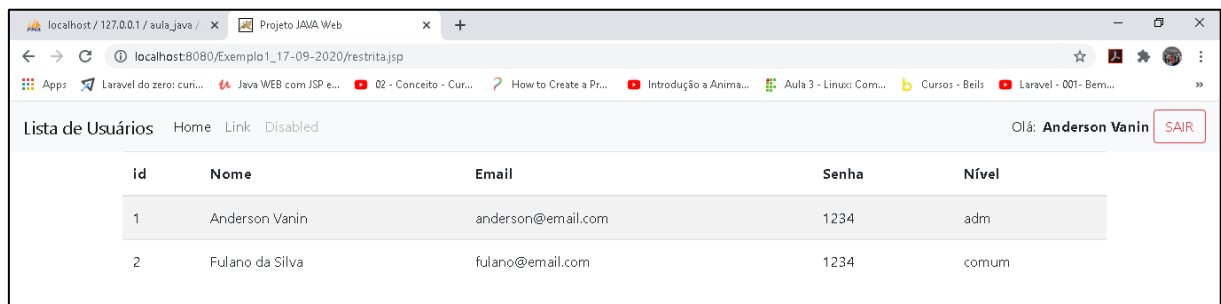
```

Para cada coluna da tabela (<td>) é necessário fazer um getString para retornar o valor correspondente no banco de dados daquela coluna.

Em cada td faça uma quebra do html e uma chamada de script do java para retornar o getString relativo a cada um dos campos desejados (id = 1, nome = 2, email = 3, senha = 4, nível = 5).

```
while (rs.next()) {  
    %>  
    <tr>  
        <td><%= rs.getString(1) %></td>  
        <td><%= rs.getString(2) %></td>  
        <td><%= rs.getString(3) %></td>  
        <td><%= rs.getString(4) %></td>  
        <td><%= rs.getString(5) %></td>  
    </tr>  
    <%  
}
```

Salve o arquivo e teste novamente no navegador.

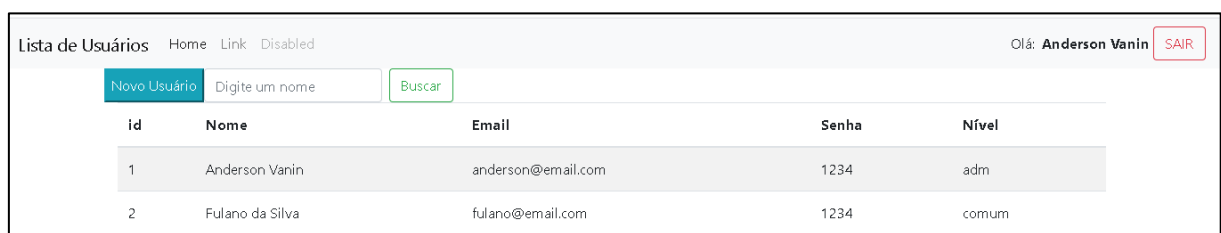


The screenshot shows a web browser window with the URL `localhost/127.0.0.1/aula_java/`. The page displays a table titled "Lista de Usuários" with the following data:

id	Nome	Email	Senha	Nível
1	Anderson Vanin	anderson@email.com	1234	adm
2	Fulano da Silva	fulano@email.com	1234	comum

At the top right, there is a greeting "Olá: Anderson Vanin" and a "SAIR" button.

Para dar continuidade ao CRUD, vamos inserir um botão para fazer o cadastro de novos usuários e um caixa de texto para buscar um determinado usuário.



The screenshot shows the same web application with additional UI elements. At the top left, there is a "Novo Usuário" button. Below it, there is a search form with a text input labeled "Digite um nome" and a "Buscar" button. The user list table remains the same.

id	Nome	Email	Senha	Nível
1	Anderson Vanin	anderson@email.com	1234	adm
2	Fulano da Silva	fulano@email.com	1234	comum

Segue o código do botão e caixa de input utilizados neste projeto:

```
<div class="row mt-4 mb-4">

    <button class="btn-info">Novo Usuário</button>

    <form class="form-inline my-2 my-lg-0">

        <input class="form-control mr-sm-2" type="search"
name="txtbuscar" placeholder="Digite um nome" aria-label="Search">

        <button class="btn btn-outline-info my-2 my-sm-0"
type="submit">Buscar</button>

    </form>

</div>
```

Desenvolvedores: fiquem a vontade para aplicarem a estilização dos componentes html da forma que acharem melhor. Pode ser criado folhas de estilos personalizadas para cada página. Neste projeto, iremos utilizar vários componentes do Bootstrap para acelerar o desenvolvimento, porém o conteúdo teórico sobre a estilização não será abordada, nos concentrando na programação do Java.

Na próxima aula vamos fazer a codificação do botão para inserir novos usuários na tabela do banco de dados.