



## Atividade 6: JavaScript e CSS

### 1. Listagem de contas - Edição (Sistema Bancário)

Vamos utilizar o JavaScript para melhorar a listagem de contas e incluir a operação de alteração de contas nesta mesma página.

Inicialmente, vamos modificar a interface para termos algo como a seguir:

ID	Número	Agência	Descricao	Editar
7	1111-1	333	Conta corrente BB	<input type="radio"/>
8	1111-2	333	Conta corrente BB	<input type="radio"/>
9	1111-3	333	Conta corrente BB	<input type="radio"/>
17	111	123	CC	<input type="radio"/>
16	498	1234	CC	<input type="radio"/>

  

ID:   
Número:   
Agência:   
Descrição:

Na tabela, vamos incluir uma coluna com um botão radio para selecionar uma conta para alteração dos dados:

```
<table border="1">
  <thead>
    <tr>
      <th>ID</th>
      <th><b>Número</b></th>
      <th><b>Agência</b></th>
      <th><b>Descricao</b></th>
      <th><b>Editar</b></th>
    </tr>
  </thead>
  <tbody>
    <c:forEach items="${ccs}" var="cc">
      <tr id="row_${cc.id}">
        <td align="center">${cc.id}</td>
        <td align="center">${cc.numero}</td>
        <td align="center">${cc.agencia}</td>
        <td align="center">${cc.descricao}</td>
```

```
 <input type="radio" name="alterar" value="{cc.id}" onClick="copiaDados(value)"></td> </tr> </c:forEach> </tbody> </table> |
```

Abaixo da tabela inclua os campos para os dados da conta dentro de um *form*, semelhante ao formulário da página de inserção. E acrescente um campo para o ID (somente leitura, não editável), conforme abaixo:

```

<form action="/save" method="post">
    ID: <input type="text" id="id" name="id" readonly style="color: #AAAAAA"
/><br>
    Número: <input type="text" id="numero" name="numero" /><br>
    Agência: <input type="text" id="agencia" name="agencia" /><br>
    Descrição: <input type="text" id="descricao" name="descricao" /><br>
    <br /> <input type="submit" name="action" value="Salvar" />
</form>

```

A ideia é que quando o usuário selecionar uma conta (clicando em um botão radio), os dados da conta sejam copiados para os campos abaixo da tabela, isso permitirá que o usuário edite os campos. Isso é realizado através de uma função JavaScript que é chamada quando o usuário seleciona uma conta para edição. Veja na coluna do botão de radio o evento `onClick` (`onClick="copiaDados(value)"`).

Vamos criar uma função JavaScript para fazer a transferência de dados de uma linha da tabela para os campos de texto:

```

<script type="application/javascript">
1     function copiaDados(id){
2         document.getElementById('id').value = id;
3         var linhaTabela = document.getElementById('row_'+id);
4         var celulas = linhaTabela.getElementsByTagName("td");
5         document.getElementById('numero').value = celulas[1].textContent;
6         document.getElementById('agencia').value = celulas[2].textContent;
7         document.getElementById('descricao').value = celulas[3].textContent;
8     }
</script>

```

Esta função recebe como parâmetro o ID da conta e preenche os campos de texto com os dados de uma linha (células da tabela) correspondente ao ID. Na linha 3 do código acima, a linha da tabela é obtida e armazenada na variável **linhaTabela** (observe que cada linha da tabela deve ter um id: `<tr id="row_{conta.id}">`). Em seguida, na linha 4, são obtidas as células da tabela (e são armazenadas em um array – variável **celulas**).

Copie essas linhas logo abaixo da tag `<body>` (remova os números das linhas).

Implemente o método para salvar a conta no *controller* (ou utilize a implementação dos exercícios anteriores) e teste a sua aplicação. Compile e execute no gradle (`gradle build` e `gradle bootRun` e acesse a página: `localhost:8080`).

## 2. Exercícios - Javascript

Inclua uma nova coluna na tabela anterior para remoção de conta, conforme a figura abaixo. Esta coluna é semelhante à coluna Editar, mantenha a chamada para a função que copia dados nos campos da conta.

ID	Número	Agência	Descricao	Editar	Remover
7	1111-1	333	Conta corrente BB	<input type="radio"/>	<input type="radio"/>
8	1111-2	333	Conta corrente BB	<input type="radio"/>	<input type="radio"/>
9	1111-3	333	Conta corrente BB	<input type="radio"/>	<input type="radio"/>
17	111	123	CC	<input type="radio"/>	<input type="radio"/>
16	498	1234	CC	<input type="radio"/>	<input type="radio"/>

  

ID:   
Número:   
Agência:   
Descrição:

Porém, agora é preciso que o botão “Salvar” do *form* seja modificado de modo que faça a remoção da conta em vez de alteração. Para isso, precisamos alterar a ação do formulário (*action* do *form*) para invocar o método no *controller* que faz a remoção e mudar o texto do botão (para “Remover”).

Para isso, vamos modificar a função *copiarDados* da seguinte forma:

1) Acrescente um parâmetro na função *copiarDados*:

```
function copiarDados(id, opcao) {
```

2) Na chamada desta função (no botão de radio das colunas Editar e Remover da tabela) passe por parâmetro uma string para identificar se a opção escolhida é editar ou remover.

3) Na função *copiarDados*, verifique a opção escolhida e modifique o *form* de acordo (por exemplo, se a opção escolhida for remover, mude a *action* do *form* de “/save” para “/delete”, por exemplo). E também modifique o texto do botão, de “Salvar” para “Remover”.

O resultado, será como na figura abaixo:

ID	Número	Agência	Descricao	Editar	Remover
7	1111-1	333	Conta corrente BB	<input type="radio"/>	<input checked="" type="radio"/>
8	1111-2	333	Conta corrente BB	<input type="radio"/>	<input type="radio"/>
9	1111-3	333	Conta corrente BB	<input type="radio"/>	<input type="radio"/>
17	111	123	CC	<input type="radio"/>	<input type="radio"/>
16	498	1234	CC	<input type="radio"/>	<input type="radio"/>

  

ID:

Número:

Agência:

Descrição:

4) Quando clicar em Remover, mostre uma mensagem de confirmação para o usuário, por exemplo, como:

localhost:8080 diz

Confirma a remoção da Conta: 1111-2 agência: 333?

Caso a resposta seja verdadeira (OK), prosseguir com a remoção da conta, caso contrário, não há nenhuma ação.

Para isso, o botão deve ter tratar o evento **onclick="return nome\_da\_funcao()"** e a função deve ser codificada de modo que mostre a mensagem somente no caso de remoção. O **return** na chamada da função faz com que o formulário não seja submetido (a requisição não seja realizada) se a função retornar falso.

5) Na alteração da conta, faça validações dos campos com JavaScript para garantir que os campos não estão vazios antes de enviar a requisição para o servidor.

### 3. Exercícios - CSS

Modifique as interfaces de seu projeto Sistema Bancário, usando tags de estilo do CSS. Você pode criar um *layout* comum para as páginas com o cabeçalho, menu (barra de navegação) para os links, modificar a aparência dos formulários (campos de texto, labels, botões, etc.), tabelas, etc. Crie um arquivo de estilos (ex. estilo.css) com as definições de estilos padrões de sua aplicação.