

# Desenvolvimento Web II

## Aula 12 - Prática: Adotando o jQuery no Sistema de Locadora

# Apresentação

---

Em aulas anteriores, você desenvolveu um sistema de Livraria web utilizando as tecnologias Servlets, JSP e AJAX. A partir desta aula, você irá melhorar sua aplicação, incorporando a tecnologia jQuery que você estudou nas últimas aulas. Você verá que, além de tornar suas aplicações mais ricas e interativas, será capaz de desenvolver código web com maior produtividade e gerar código com melhores aspectos de qualidade.

Tenha uma boa leitura!



**Vídeo 01** - Apresentação

## Objetivos

- Adicionar o recurso jQuery no sistema da Livraria.
- Utilizar um diálogo (componente dialog) do jQuery UI para exibir o carrinho de compras (ao invés do popup do navegador).

## Incluindo as referências do jQuery (padrão e UI)

---

O primeiro passo para se utilizar jQuery é incluir as suas bibliotecas. Para isso, podemos fazer uso do arquivo `cabecalho.jspf`, pois ele já é incluído, automaticamente, em praticamente todos as páginas do sistema.

A seguir, veremos o código do arquivo `cabecalho.jsp`, que agora está incluindo as bibliotecas do jQuery (linhas 11 a 13):

```
1 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
2 <%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>
3 <html>
4   <head>
5     <title>
6       Minha Livraria Virtual
7     </title>
8     <script type="text/javascript" src="http://code.jquery.com/jquery-2.1.3.js"></script>
9     <script src="https://code.jquery.com/ui/1.12.0/jquery-ui.js"></script>
10    <link rel="stylesheet" href="https://code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css"/>
11  </head>
12
13  <body bgcolor="#FFFFFF">
14    <center>
15      <hr>
16      <br>
17      <h1>
18        
19        <font size="+3">Minha Livraria Virtual</font>
20        
21      </h1>
22    </center>
23
24    <br>
25    <hr>
```

## Atividade 01

---

1. Faça o ajuste na implementação do sistema da livraria virtual, como mostrado.



**Vídeo 02** - Animate

## Substituindo o código JavaScript comum por funções da jQuery

---

Com intuito de demonstrar o quanto jQuery é mais produtivo e menos propenso a erros em relação a JavaScript puro (devido a menor quantidade de código e a menor complexidade do mesmo), nesta seção, iremos apresentar as funções Javascript que iremos no fim da aula utilizar para alterar os arquivos JSP tanto com o código JavaScript puro (comentado, não será executado) como o código jQuery (não comentado e que efetivamente serão executados) para que seja possível se fazer uma comparação.

Vamos apresentar várias funções Javascript/jQuery novas que serão utilizadas no projeto e mais a frente o código final modificado, então por enquanto somente mantenha a atenção nas diferenças entre utilizar ou não JQuery.

Começemos pela função `verDetalhes()`, do arquivo `catalogo.jsp`, mostrada a seguir. No código dessa função, você pode ver tanto o código antigo, comentado com `//`, como o novo, bem mais simples e enxuto.

```

1 function verDetalhes(url){
2
3     // Sem jQuery
4     //var catalogo = document.getElementById("catalogo").attributes;
5     //catalogo.getElementsByTagName("style").nodeValue="visibility: hidden; height: 0px;";
6
7     // Com jQuery
8     $("#catalogo").hide();
9     carregarDetalhes(url);
10
11 }

```

Vamos ver agora a nova função `verCatalogo()`. Repare que se o código fosse escrito em Javascript puro, sem o uso do jQuery, você precisaria manipular valores de estilo CSS diretamente enquanto com jQuery basta se utilizar os métodos `show()` e `hide()` para obter o mesmo resultado (mostra/esconder um elemento). O arquivo que irá futuramente conter essa função é o `catalogo.jsp`. Veja a função `verCatalogo()` abaixo:

```

1 function verCatalogo(){
2
3     // Sem jQuery
4     //var catalogo = document.getElementById("catalogo").attributes;
5     //catalogo.getElementsByTagName("style").nodeValue="";
6     //
7     //var detalhes = document.getElementById("detalhes").attributes;
8     //detalhes.getElementsByTagName("style").nodeValue="visibility: hidden; height: 0px;";
9
10    // Com jQuery
11    $("#catalogo").show();
12    $("#detalhes").hide();
13 }

```

Ainda no mesmo arquivo, iremos ter a função `carregarDetalhes()`. O código Javascript puro está comentado com `/*` e `*/`. É interessante ver que, mesmo com o uso do do Ajax no Javascript puro, o código jQuery é muito mais simples e fácil de entender.

```
1 function carregarDetalhes(url){
2
3     // Sem jQuery
4     /*
5     var ajaxReq = new XMLHttpRequest();
6     ajaxReq.onreadystatechange = function() {
7         if (ajaxReq.readyState == 4 && ajaxReq.status == 200) {
8             var resp = ajaxReq.responseText;
9             var detalhes = document.getElementById("detalhes");
10             detalhes.innerHTML=resp;
11             detalhes.attributes.getNamedItem("style").nodeValue="";
12         }
13     };
14
15     ajaxReq.open("GET", url, true);
16     ajaxReq.send();
17     */
18
19     // Com jQuery
20     $("#detalhes").load(url);
21     $("#detalhes").show();
22
23 }
```

jQuery realmente ajuda a simplificar o código não é mesmo? Imagina ter que fazer um sistema completo somente utilizando Javascript puro... seria muito mais trabalhoso. Veja também como ficará a nova função adicionarLivro():

```

1 function adicionarLivro(url){
2
3     // Sem jQuery
4     /*
5     var ajaxReq = new XMLHttpRequest();
6     ajaxReq.onreadystatechange = function() {
7         if (ajaxReq.readyState == 4 && ajaxReq.status == 200) {
8             var resp = ajaxReq.responseText;
9             var livroAdicionado = document.getElementById("livroAdicionado");
10            livroAdicionado.innerHTML="Você adicionou o livro <em>"+ resp +"</em> ao seu carrinho d
11        }
12    };
13
14    ajaxReq.open("GET", url, true);
15    ajaxReq.send();
16    verCatalogo();
17    */
18
19    // Com jQuery
20    $("#carrinhoVazioMsg").hide();
21    $.get(url, function(dados){
22        $("#livroAdicionado").html("Você adicionou um livro ao seu carrinho de compras.");
23    });
24    $('#carrinhoVazioMsg').hide();
25    $('#carrinhoLinks').show();
26    verCatalogo();
27    verCarrinho();
28 }

```

Após comparar o JavaScript puro (comentado nos trechos acima) com o código jQuery que desempenha o mesmo propósito, é notável que é o código jQuery é menor, mais simples e mais legível.

## Atividade 02

---

1. Faça o ajuste na implementação do sistema da livraria virtual, como mostrado.

# Exibindo o carrinho de compras utilizando o componente dialog do jQuery UI

No trecho de código a seguir, damos início à apresentação das novas funções necessárias para exibir o carrinho de compras utilizando o componente Dialog da biblioteca jQuery UI ao invés dele ser carregado em uma nova página. Nesse código, podemos ver a função `verCarrinho()`, do arquivo `catalogo.jsp`, com o componente jQuery e também com o código comentado de como seria a implementação em Javascript puro para efeitos de comparação.



## Vídeo 03 - Tabs

```
1 function verCarrinho(limpar,remover){
2
3     //var url = "mostrarCarrinho?limpar="+limpar+"&remover="+remover;
4     //var options =
5         "height=500,width=700,location=no,menubar=no,resizable=no,scrollbars=no,title
6         bar=no,status=no,toolbar=no";
7     //
8     //window.open(url,"carrinho",options,true);
9
10    var url="mostrarCarrinho?limpar="+limpar+"&remover="+remover;
11    $("#carrinho").load(url);
12    $("#carrinho").dialog({
13        height:600,
14        width:700,
15        modal:true,
16        title:"Carrinho de Compras"
17    });
18 }
```

A função `verCarrinho()` exibida acima simplesmente utiliza o jQuery para acessar o URL do carrinho, obtendo seu HTML e o colocando dentro da div “carrinho” (div criada para atender a esse propósito) e a exibe utilizando o método `dialog()`. Observe que o parâmetro `modal` com valor `true` faz com que a página em segundo plano não



fique acessível enquanto o dialogo estiver aberto. O código comentado em Javascript puro tentaria simular o diálogo do jQuery simplesmente abrindo uma nova janela com o carrinho, o que não é uma boa prática.

Outra função que será criada é a `continuar()`. Essa função simplesmente fecha o diálogo do jQuery que exibe o carrinho de compras. O código comentado em Javascript puro na verdade não trata de um diálogo como o do jQuery, mas sim fecharia uma janela popup com o carrinho de compras, o que é uma abordagem muito menos elegante.

```
1 function continuar(url){  
2     //window.opener.open(url,"_parent");  
3     //window.close();  
4  
5     $("#carrinho").dialog("close");  
6 }
```

A função `fecharVerCatalogo()`, que será adicionada no arquivo `mostrarCarrinho.jsp`, fecha o diálogo jQuery em exibição, além de tornar visível o contêiner que contém o catálogo de livros:

```
1 function fecharVerCatalogo() {  
2  
3     $("#carrinho").dialog("close");  
4     verCatalogo();  
5  
6 }
```

Ainda no arquivo `mostrarCarrinho.jsp`, a função `acessarUrlNoCarrinho()` faz uma solicitação AJAX utilizando o método `load()` da API do jQuery padrão, porém exibe o resultado no dialog do carrinho. Isso é útil para várias ações que são realizadas dentro do carrinho como esvaziar, incrementar/decrementar quantidade de livros, etc.

```
1 function acessarUrlNoCarrinho(url) {  
2     $("#carrinho").load(url);  
3 }
```

Também no arquivo `mostrarCarrinho.jsp`, a função `verDetalhesDoCarrinho(url)` é utilizada para duas ações: na página principal detalhar um livro e em seguida fechar o carrinho (para que o usuário possa ver o detalhamento exibido). Segue o código:

```
1 function verDetalhesDoCarrinho(url) {  
2     verDetalhes(url);  
3     $('#carrinho').dialog('close');  
4 }
```

## Atividade 03

---

1. Faça o ajuste na implementação do sistema da livreria virtual, como mostrado.

## Arquivos fontes modificados

---

A seguir, você pode ver os arquivos modificados em suas versões completas, já com as novas funções Javascript utilizando o jQueryo jQuery. O código comentado em cada função seria sua versão usando Javascript puro, sem o jQuery para se obter um efeito similar. Começamos pelo arquivo catalogo.jsp, da pasta livros:



**Vídeo 04** - Animate CSS

```
1 <%@ include file="/cabecalho.jsp" %>
2 <script type="text/javascript">
3
4 function verCarrinho(limpar,remover){
5
6     //var url = "mostrarCarrinho?limpar="+limpar+"&remover="+remover;
7     //var options = "height=500,width=700,location=no,menubar=no,resizable=no,scrollbars=no,title
8     //
9     //window.open(url,"carrinho",options,true);
10
11     var url = "mostrarCarrinho?limpar="+limpar+"&remover="+remover;
12     $("#carrinho").load(url);
13     $("#carrinho").dialog({
14         height:600,
15         width:700,
16         modal:true,
17         title:"Carrinho de Compras"
18     });
19 }
20
21 function verDetalhes(url){
22
23     // Sem jQuery
24     //var catalogo = document.getElementById("catalogo").attributes;
25     //catalogo.getItem("style").nodeValue="visibility: hidden; height: 0px;";
26
27     // Com jQuery
28     $("#catalogo").hide();
29     carregarDetalhes(url);
30
31 }
32
33 function verCatalogo(){
34
35     // Sem jQuery
36     //var catalogo = document.getElementById("catalogo").attributes;
37     //catalogo.getItem("style").nodeValue="";
38     //
39     //var detalhes = document.getElementById("detalhes").attributes;
40     //detalhes.getItem("style").nodeValue="visibility: hidden; height: 0px;";
41
42     // Com jQuery
43     $("#catalogo").show();
44     $("#detalhes").hide();
45
46 }
47
48 function carregarDetalhes(url){
49
50     // Sem jQuery
51     /*
```

```

52 var ajaxReq = new XMLHttpRequest();
53 ajaxReq.onreadystatechange = function() {
54     if (ajaxReq.readyState == 4 && ajaxReq.status == 200) {
55         var resp = ajaxReq.responseText;
56         var detalhes = document.getElementById("detalhes");
57         detalhes.innerHTML=resp;
58         detalhes.attributes.getNamedItem("style").nodeValue="";
59     }
60 };
61
62 ajaxReq.open("GET", url, true);
63 ajaxReq.send();
64 */
65
66 // Com jQuery
67 $("#detalhes").load(url);
68 $("#detalhes").show();
69
70 }
71
72 function adicionarLivro(url){
73
74     // Sem jQuery
75     /*
76     var ajaxReq = new XMLHttpRequest();
77     ajaxReq.onreadystatechange = function() {
78         if (ajaxReq.readyState == 4 && ajaxReq.status == 200) {
79             var resp = ajaxReq.responseText;
80             var livroAdicionado = document.getElementById("livroAdicionado");
81             livroAdicionado.innerHTML="Você adicionou o livro <em>" + resp + "</em> ao seu carrinho
82         }
83     };
84
85     ajaxReq.open("GET", url, true);
86     ajaxReq.send();
87     verCatalogo();
88     */
89
90     // Com jQuery
91     $("#carrinhoVazioMsg").hide();
92     $.get(url, function(dados){
93         $("#livroAdicionado").html("Você adicionou um livro ao seu carrinho de compras.");
94     });
95     $("#carrinhoVazioMsg").hide();
96     $("#carrinhoLinks").show();
97     verCatalogo();
98     verCarrinho();
99 }
100 </script>
101 <jsp:useBean id="livrariaBean" class="livraria.negocio.LivrariaBean" scope="page" >
102     <jsp:setProperty name="livrariaBean" property="sistema" value="${sistemaLivraria}" />

```

```

103 </jsp:useBean>
104
105 <p><h3><font id="livroAdicionado" color="red" size="+2"> </font></h3></p>
106
107 <div id="carrinhoLinks" style="display: none;">
108 <c:url var="url" value="javascript:verCarrinho(0,0)">
109 </c:url>
110 <p>
111 <a href="{url}">Ver carrinho de compras</a>
112 <c:url var="url" value="/livros/comprar" />
113 <a href="{url}">Finalizar compras</a></strong>
114 </p>
115 </div>
116 <div id="carrinhoVazioMsg" style="display: none;">Seu carrinho de compras está vazio.</div>
117 <c:choose>
118 <c:when test="{sessionScope.cart.numeroltens > 0}">
119 <script type="text/javascript">
120 <$( '#carrinhoVazioMsg' ).hide();
121 <$( '#carrinhoLinks' ).show();
122 </script>
123 </c:when>
124 <c:otherwise>
125 <script type="text/javascript">
126 <$( '#carrinhoVazioMsg' ).show();
127 <$( '#carrinhoLinks' ).hide();
128 </script>
129 </c:otherwise>
130 </c:choose>
131 <br>
132 <br>
133 <div id="carrinho">
134 </div>
135
136 <div id="detalhes">
137 </div>
138
139 <div id="catalogo">
140 <h3>Livros disponíveis para compra:</h3>
141 <%
142 // variável buscarValue mantém o termo de busca no formulário ou colocar uma string vazia.
143 String buscarValue = request.getParameter("buscar");
144 if (buscarValue == null) {
145     buscarValue = "";
146 }
147 %>
148 <form method="GET">
149 <input type="hidden" name="Add" value="" />
150 Busca: <input type="text" name="buscar" value="<%= buscarValue %>" />
151 <input type="submit" value="Buscar" />
152 </form>
153 <table summary="layout">

```

```

154 <c:forEach var="livro" begin="0" items="${requestScope.livros}">
155 <tr>
156 <c:set var="idLivro" value="${livro.idLivro}" />
157 <td bgcolor="#ffffaa">
158 <c:url var="url" value="/livros/detalhesLivro" >
159 <c:param name="idLivro" value="${idLivro}"/>
160 </c:url>
161 <strong><a href="javascript:verDetalhes('${url}');"><strong>${livro.titulo}</strong></a>
162 </td>
163 <td bgcolor="#ffffaa" rowspan=2>
164 <fmt:formatNumber value="${livro.preco}" type="currency"/>
165 </td>
166 <td bgcolor="#ffffaa" rowspan=2>
167 <c:url var="url" value="/livros/catalogo" >
168 <c:param name="Add" value="${idLivro}"/>
169 </c:url>
170 <a href="javascript:adicionarLivro('${url}')"> Adicionar livro ao carrinho </a>
171 </td>
172 </tr>
173 <tr>
174 <td bgcolor="#ffffff">
175 <em>${livro.autores} </em>
176 </td>
177 </tr>
178 </c:forEach>
179 </table>
180 </div>
181
182 <%@ include file="/rodape.jsp" %>

```

Veja, agora, o código fonte completo do arquivo mostrarCarrinho.jsp, também armazenado na pasta livros:

```

1 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
2 <%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>
3 <script>
4
5 function continuar(url) {
6
7     //window.opener.open(url, "_parent");
8     //window.close();
9
10    $("#carrinho").dialog("close");
11 }
12
13 function fecharVerCatalogo() {
14     $("#carrinho").dialog("close");
15     verCatalogo();
16 }
17
18
19 function acessarUrlNoCarrinho(url) {
20     $("#carrinho").load(url);
21 }
22
23 function verDetalhesDoCarrinho(url) {
24     verDetalhes(url);
25     $('#carrinho').dialog('close');
26 }
27
28 </script>
29
30 <jsp:useBean id="livrariaBean" class="livraria.negocio.LivrariaBean" scope="page" >
31     <jsp:setProperty name="livrariaBean" property="sistema" value="${sistemaLivraria}" />
32 </jsp:useBean>
33 <c:if test="${param.limpar == 'limpar'}">
34     <font color="red" size="+2">
35         <strong>O carrinho de compras foi esvaziado!</strong><br> <br>
36     </font>
37 </c:if>
38 <c:if test="${param.remover != '0' && param.remover != null}">
39     <c:set var="id" value="${param.remover}" />
40     <jsp:setProperty name="livrariaBean" property="idLivro" value="${id}" />
41     <c:set var="livroRemovido" value="${livrariaBean.livro}" />
42     <font color="red" size="+2">O seguinte livro foi removido do carrinho:
43         <em>${livroRemovido.titulo}</em>.
44         <br> <br>
45     </font>
46 </c:if>
47 <c:if test="${sessionScope.cart.numeroItens > 0}">
48     <font size="+2">Quantidade de itens do carrinho: ${sessionScope.cart.numeroItens}
49     <c:if test="${sessionScope.cart.numeroItens == 1}">
50         livro.
51     </c:if>

```

```

52 <c:if test="${sessionScope.cart.numeroltens > 1}">
53     livros.
54 </c:if>
55 </font><br>
56 <table summary="layout">
57     <tr>
58         <th align=left colspan="3">Quantidade</th>
59         <th align=left>Título</th>
60         <th align=left>Preço</th>
61     </tr>
62     <c:forEach var="itemCompra" items="${sessionScope.cart.itens}">
63         <c:set var="livro" value="${itemCompra.item}" />
64         <tr>
65             <td bgcolor="#ffffff">
66                 <c:url var="url" value="/livros/mostrarCarrinho" >
67                     <c:param name="alterar" value="${livro.idLivro}" />
68                     <c:param name="quantidade" value="1" />
69                     <c:param name="remover" value="0" />
70                 </c:url>
71                 <a href="javascript:acessarUrlNoCarrinho('${url}');" style="text-decoration: none;">[+]</a>
72             </td>
73             <td bgcolor="#ffffff">
74                 <c:url var="url" value="/livros/mostrarCarrinho" >
75                     <c:param name="alterar" value="${livro.idLivro}" />
76                     <c:param name="quantidade" value="-1" />
77                     <c:param name="remover" value="0" />
78                 </c:url>
79                 <a href="javascript:acessarUrlNoCarrinho('${url}');" style="text-decoration: none;">[-]</a>
80             </td>
81             <td align="left" bgcolor="#ffffff">${itemCompra.quantidade}</td>
82             <td bgcolor="#ffffaa">
83                 <c:url var="url" value="/livros/detalhesLivro" >
84                     <c:param name="idLivro" value="${livro.idLivro}" />
85                     <c:param name="Clear" value="0" /></c:url>
86                 <strong><a href="javascript:verDetalhesDoCarrinho('${url}');">${livro.titulo}</a></strong>
87             </td>
88             <td bgcolor="#ffffaa" align="right">
89                 <fmt:formatNumber value="${livro.preco}" type="currency"/>
90             </td>
91             <td bgcolor="#ffffaa">
92                 <c:url var="url" value="/livros/mostrarCarrinho" >
93                     <c:param name="remover" value="${livro.idLivro}" /></c:url>
94                 <strong><a href="javascript:acessarUrlNoCarrinho('${url}');">Remover</a></strong>
95             </td>
96         </tr>
97     </c:forEach>
98     <tr>
99         <td colspan="5" bgcolor="#ffffff"><br></td>
100    </tr>
101    <tr>
102        <td colspan="2" align="right" bgcolor="#ffffff">Subtotal</td>

```



```
103     <td bgcolor="#ffffaa" align="right">
104         <fmt:formatNumber value="${sessionScope.cart.total}" type="currency"/>
105     </td>
106     <td><br></td>
107 </tr>
108 </table>
109 <p> <p>
110 <c:url var="url" value="/livros/catalogo" >
111     <c:param name="Add" value="" />
112 </c:url>
113 <strong>
114     <a href="javascript:continuar('${url}')">Continuar comprando</a>
115     <c:url var="url" value="/livros/comprar" />
116         <a href="${url}">Finalizar compra</a>
117         <c:url var="url" value="/livros/mostrarCarrinho">
118             <c:param name="limpar" value="limpar"/>
119             <c:param name="remover" value="0"/>
120         </c:url>
121         <a href="javascript:acessarUrlNoCarrinho('${url}')">Esvaziar carrinho</a></strong>
122 </strong>
123 </c:if>
124 <c:if test="${sessionScope.cart.numeroItens <= 0}">
125 <font size="+2">Carrinho vazio</font>
126 <br> <br>
127 <c:url var="url" value="/livros/catalogo" >
128     <c:param name="Add" value="" />
129 </c:url>
130 <strong><a href="javascript:continuar('${url}')">Fechar</a></strong>
131 </c:if>
```

Por fim, temos o código do arquivo detalhesLivro.jsp

```

1 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
2 <%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>
3 <jsp:useBean id="livrariaBean" class="livraria.negocio.LivrariaBean" scope="page" >
4     <jsp:setProperty name="livrariaBean" property="sistema" value="{sistemaLivraria}" />
5 </jsp:useBean>
6 <c:if test="{!empty param.idLivro}" >
7     <c:set var="id" value="{param.idLivro}" />
8     <jsp:setProperty name="livrariaBean" property="idLivro" value="{id}" />
9     <c:set var="livro" value="{livrariaBean.livro}" />
10    <h2>{livro.titulo}</h2>
11    Autoria de <em> {livro.autores}</em>
12    ({livro.ano})<br> <br>
13    <h4>Descrição</h4>
14    <blockquote>{livro.descricao}</blockquote>
15    <h4>Preço: <fmt:formatNumber value="{livro.preco}" type="currency" /></h4>
16    <c:url var="url" value="/livros/catalogo" >
17        <c:param name="Add" value="{id}" />
18    </c:url>
19    <a href="javascript:adicionarLivro('{url}')">Adicionar ao carrinho </a>
20 </c:if>
21 <c:url var="url" value="/livros/catalogo" >
22     <c:param name="Add" value="" />
23 </c:url>
24 <a href="javascript:verCatalogo()">Continuar comprando</a></strong></p>
25

```

Repare que basicamente os links de “Adicionar ao carrinho” e “Continuar comprando” foram alterados para executar as funções Javascript que contém os comandos JQuery necessários para que essas operações sejam executadas com a usabilidade correta na nossa aplicação modificada.

## Atividade 04

1. Implemente todos os recursos vistos nessa aula no projeto da Livraria.
2. Teste todo o sistema e se certifique que a usabilidade ainda está adequada, ou seja, se o uso do JQuery fez alguma URL não responder de forma desejada faça as correções necessárias.

## Resumo

---

Nesta aula, você praticou o uso de jQuery (padrão e UI), diminuindo drasticamente a quantidade de linhas de código. O código final também ficou mais simples e fácil de ler e manter. Também podemos observar que ao utilizar jQuery aumentamos muito a produtividade e a qualidade do desenvolvimento.

## Autoavaliação

---

1. Descreva um exemplo do que você fez para substituir o código JavaScript puro por um código de mais alto nível oferecido pela jQuery padrão.
2. Descreva o que você precisou fazer para conseguir usar um diálogo (componente dialog) do jQuery UI para exibir o carrinho de compras (ao invés do popup do navegador).

## Referências

---

JQUERY. Disponível em: <<http://jquery.com/>>. Acesso em: 1 ago. 2012.

W3c. Disponível em: <<http://www.w3.org/>>. Acesso em: 1 ago. 2012.

W3school. Disponível em: <<http://www.w3schools.com/>>. Acesso em: 1 ago. 2012.

\_\_\_\_\_. **Ajax**. Disponível em: <<http://www.w3schools.com/Ajax/>>. Acesso em 1 ago. 2012.