

# Trabalho de Recuperação

## SCC0265 - Sistemas Interativos Web

### Professora : Renata Pontin M. Fortes

*Trabalho Desenvolvido pelo Aluno*

Ubiratan Soares (5634292)  
São Carlos, 20 de agosto de 2010

#### Introdução

O seguinte enunciado foi proposto para o trabalho de recuperação da disciplina de Sistemas Interativos Web:

*"Implementar um portal de notícias com conteúdo dinâmico, que apresente notícias apenas de esportes. Essas notícias devem ser extraídas dos portais de notícias da Globo, Terra e UOL. É importante destacar que essas notícias precisam ser atualizadas, conforme os portais de notícias são atualizados.*

*Recomendações: Implementar no lado servidor da aplicação a extração das notícias do Globo, Terra e UOL utilizando o RSS desses portais. Também devem ser utilizadas técnicas como XML-DOM ou XSLT."*

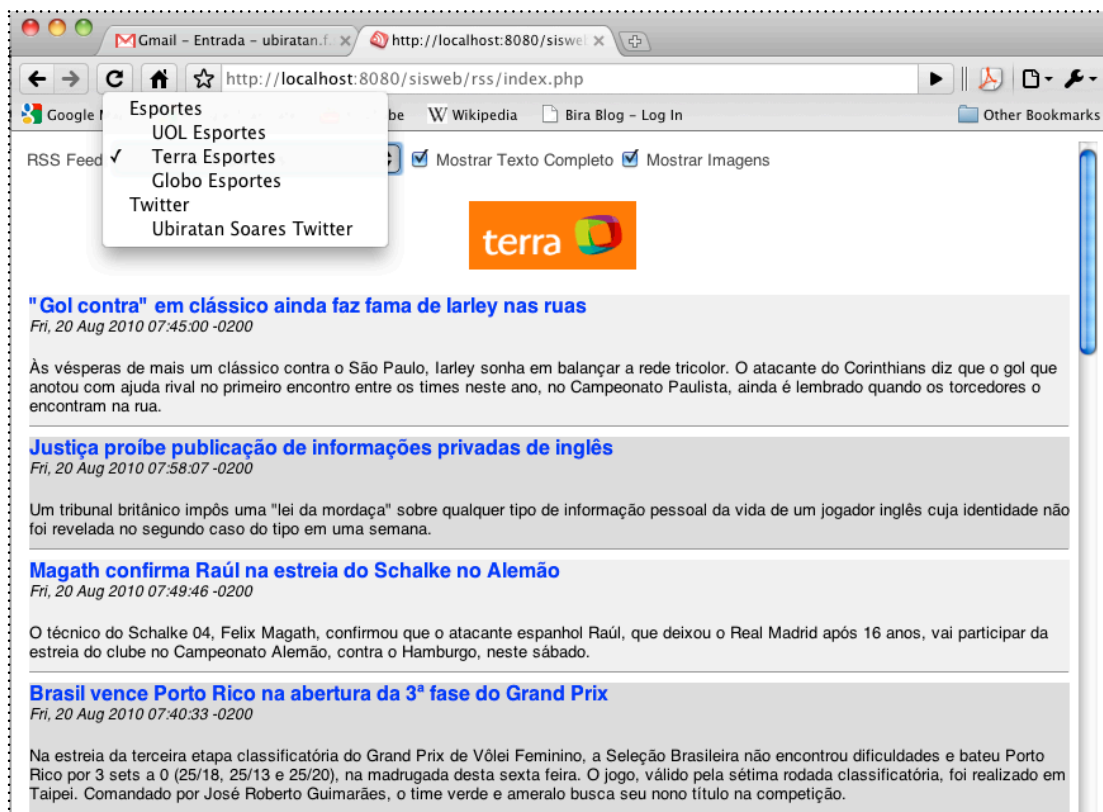
Dessa maneira, apesar do foco em programação no lado do cliente (*client side*) abordado durante toda a disciplina via técnicas de Javascript, XML e correlatos, foi recomendado um mínimo de programação no lado do servidor (*server side*) para que a aplicação sugerida tivesse a funcionalidade desejada.

#### A Implementação

Para a implementação do trabalho, foi escolhida como plataforma do servidor de aplicação o tradicional combo Apache e PHP, que é Open Source e amplamente divulgado e utilizado em aplicações pela Web. Assim, o produto final do trabalho realizado consiste em três arquivos contendo os códigos-fonte que implementam a aplicação sugerida:

- **index.php** - Arquivo que será a "página principal" do portal de notícias. Contém as implementações de chamada assíncrona usando o objeto XMLHttpRequest do Javascript, além das funções em PHP que estabelecem funcionalidades dependentes do lado do servidor.
- **ajax.php** - Arquivo que contém a combinação de AJAX e PHP para implementar a interpretação do conteúdo XML lido do feed e sua respectiva visualização pelo usuário no browser. O parser do feed XML sendo exibido é realizado aqui, utilizando o XPath.
- **style.css** - Folha de estilo simples para trazer uma qualidade visual aceitável para a implementação.
- **feeds.xml** - Arquivo XML que contém os endereços de cada feed que compõe o canal de notícias.

O resultado final da implementação pode ser conferido nos *screenshots* a seguir:

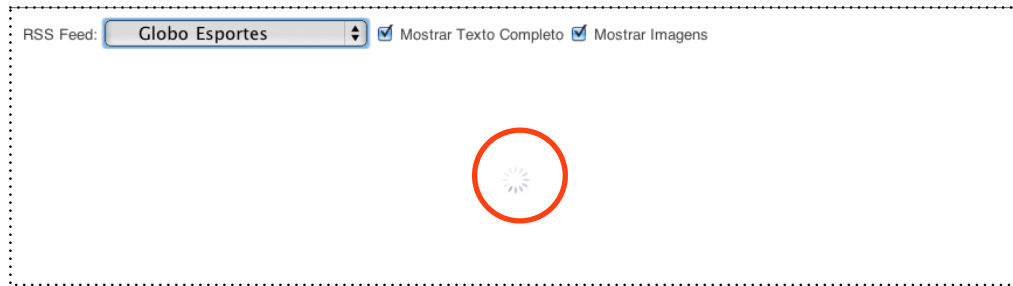


A página é bastante simples e oferece ao usuário uma caixa de seleção, na qual se fez a escolha do feed RSS a ser exibido. Além disso, são oferecidos dois *checkbox* nos quais o usuário pode decidir entre mostrar ou não o texto completo dos feeds e as imagens dos mesmos.



As entradas de cada feed estão destacados em *divs* html diferenciados pela cor. No topo de cada div, encontra-se o link para o site na Web que contém a matéria completa. Além disso, a imagem associada ao arquivo XML importado pelo servidor do respectivo canal de notícias é mostrada tanto no topo quando no fundo da página, para evidenciar o portal de onde as notícias em exibição foram extraídas.

Foi implementado um sistema de *timing* conforme o exigido, que atualiza o canal de notícias sendo exibido em um tempo facilmente definido pela admistrador da aplicação. Esse mecanismo foi desenvolvido em Javascript, e por padrão, remonta o feed XML sendo exibido a cada 10 segundos.



É importante destacar que o mecanismo de atualização automática mantém o *refresh* do feed sendo exibido em intervalos de tempo constante. Isso pode facilmente ser conseguido através da implementação de uma variável do tipo *timerRunning*, que controla o mecanismo e impede que mudanças nos *checkboxes* alterem o *timeout* ajustado pelo Javascript.

Por fim, algumas observações de natureza prática. O feed do “UOL Esportes” apenas envia os links das notícias, e não parte dos textos das matérias. Além disso, dos feeds escolhidos, somente o “Globo Esportes” envia imagens representativas para cada notícia.

As funcionalidades oferecidas pelos checkboxes foram testadas em outros feeds da Web. Finalmente, no sentido de oferecer uma melhor demonstração do mecanismo de timing durante a apresentação desse trabalho, foi disponibilizada uma entrada em *feeds.xml* para o feed RSS da página do Twitter do autor desse trabalho, de maneira que uma atualização simples pode ser realizada e exibida de maneira controlada durante a exposição.

## Listagem dos Códigos-Fonte

Segue a listagem dos códigos-fonte desenvolvidos.

### Arquivo: feeds.xml

```
1. <?xml version="1.0" encoding="UTF-8"?>
2. <rss>
3.     <options images="true" fulltext="true" />
4.     <feeds>
5.         <category name="Esportes">
6.             <feed name="UOL Esportes" url="http://rss.esporte.uol.com.br/ultimas/index.xml" />
7.             <feed name="Terra Esportes" url="http://rss.terra.com.br/0,,EI1137,00.xml" />
8.             <feed name="Globo Esportes" url="http://globoesporte.globo.com/dynamo/rss2.xml" />
9.         </category>
10.
11.         <category name="Twitter">
12.             <feed name="Ubiratan Soares Twitter" url="http://twitter.com/statuses/user_timeline/
13.             27679912.rss" />
14.         </category>
15.     </feeds>
16. </rss>
```

## Arquivo : index.php

```
1.    <link rel="icon" type="image/png" href="include/feed-icon-red.png">
2.
3.    <style language='text/css'>
4.
5.        @import url('include/style.css');
6.
7.    </style>
8.
9.    <script language="javascript">
10.
11.        var XMLHttpRequestObject = false;
12.
13.        var img = new Image();
14.        img.src = "include/loading.gif";
15.
16.        if ( window.XMLHttpRequest ) XMLHttpRequestObject = new XMLHttpRequest();
17.
18.        else if (window.ActiveXObject) XMLHttpRequestObject = new ActiveXObject("Microsoft.XMLHTTP");
19.
20.        function changeOptions(){
21.            LoadFeed();
22.        }
23.
24.
25.        function LoadFeed(value, initial){
26.
27.            if ( XMLHttpRequestObject ){
28.
29.                var dsource = "include/ajax.php?loadfeed";
30.
31.                if (value) dsource += "=" + value;
32.
33.                else{
34.
35.                    var val = document.getElementById("rss_feed").value;
36.
37.                    if ( val == "" ) document.getElementById("rss_feed").selectedIndex += 1;
38.
39.                    dsource += "=" + document.getElementById("rss_feed").value;
40.                }
41.
42.                dsource += "&fulltext=" + document.getElementById("showtext").checked;
43.
44.                dsource += "&showimages=" + document.getElementById("showimages").checked;
45.
46.                XMLHttpRequestObject.open("GET",dsource);
47.
48.                XMLHttpRequestObject.onreadystatechange = function(){
49.
50.                    if ( XMLHttpRequestObject.readyState != 4 ){
51.                        document.getElementById("rss_div").innerHTML = "<center><img class=\"loading
52.                        \" src=\"\" + img.src + \"\"></center>";
53.                    }
54.
55.                    if ( XMLHttpRequestObject.readyState == 4 && XMLHttpRequestObject.status == 200 ){
56.                        document.getElementById("rss_div").innerHTML =
57.                        XMLHttpRequestObject.responseText;
58.                    }
59.                }
60.            }
61.            //end function
62.
63.            XMLHttpRequestObject.send(null);
64.
65.            //listenForNews();
66.
67.        }
```

```

63.             InitializeTimer();
64.
65.         }//end if
66.
67.     }//end LoadFeed
68.
69.
70.     // Funções para temporização. Atualizam o feed na página a cada 30s. Desincronização após várias chamadas.
71.
72.     /*
73.     function listenForNews() {
74.
75.         setTimeout('updateNews();', 30 * 1000);
76.         return;
77.     }
78.
79.     function updateNews() {
80.         LoadFeed();
81.         setTimeout('updateNews();', 60 * 1000);
82.         return;
83.     }
84.     */
85.
86.     var secs
87.     var timerID = null
88.     var timerRunning = false
89.     var delay = 1000
90.
91.     function InitializeTimer(){
92.
93.         // Ajusta o tempo em segundos entre atualizações
94.         secs = 10;
95.         StopTheClock();
96.         StartTheTimer();
97.     }
98.
99.     function StopTheClock(){
100.
101.         if(timerRunning) clearTimeout(timerID);
102.         timerRunning = false;
103.     }
104.
105.     function StartTheTimer(){
106.
107.         if (secs==0){
108.             StopTheClock();
109.             LoadFeed();
110.         }
111.         else{
112.             self.status = secs;
113.             secs = secs - 1;
114.             timerRunning = true;
115.             timerID = self.setTimeout("StartTheTimer()", delay);
116.         }
117.     }
118.
119. </script>
120.
121. <?php
122.
123. $feeds_xml="feeds.xml";
124.
125.
126.
127.
128.

```

```

129. function LoadFile($xml_file)
130. {
131.     $xml = null;
132.     if ( !file_exists($xml_file) || !( $xml = simplexml_load_file($xml_file)) )
133.     {
134.         echo '</select>';
135.         echo "Error opening XML feeds file: $xml_file<br>";
136.         return;
137.     }
138.
139.     return $xml;
140. }
141.
142. function ShowFeedOptions($xml_file)
143. {
144.     echo '<select id="rss_feed" onChange="LoadFeed(this.value)" title="Bira RSS Reader">';
145.
146.     $xml = LoadFile($xml_file);
147.
148.     if ( $xml == null ) return;
149.
150.     $url = array();
151.
152.     foreach($xml->feeds->category as $category){
153.
154.         echo "<option value=\"\">" . $category['name'] . "</option>";
155.
156.         foreach($category->feed as $feed){
157.
158.             if ( $feed['name'] != "" ){
159.
160.                 // Montando a caixa de escolha
161.                 $name = "&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~";
162.                 $url = str_replace("&", "%26", $feed['url']);
163.                 $url = str_replace("http://", "", htmlspecialchars($url));
164.                 echo "<option value=\"$url\">" . $name . "</option>";
165.             }
166.         }
167.     }
168.
169.     echo '</select>';
170.
171.     $options = array();
172.
173.     $rss = $xml->options[0];
174.
175.     // Escolhas do usuário para imagem e conteúdo
176.
177.     if ( strtolower($rss['fulltext'])=="true" ) array_push($options, "true");
178.     else array_push($options, "false");
179.
180.     if ( strtolower($rss['images'])=="true" ) array_push($options, "true");
181.     else array_push($options, "false");
182.
183.     return $options;
184. }
185.
186. function SetOptions($options)
187. {
188.     if ( sizeof($options) == 2 )
189.     {
190.         echo "<script>document.getElementById(\"showtext\").checked = " . $options[0] . ";</script>";
191.         echo "<script>document.getElementById(\"showimages\").checked = " . $options[1] . ";</script>";
192.     }
193. }

```

```

195.     if ( isset($_GET['links']) ){
196.
197.         $xml = LoadFile($xml_file);
198.         if ( $xml == null )
199.             return;
200.         return;
201.     }
202.
203.     ?>
204.
205.     <!--Montando a formatação principal da página -->
206.
207.     <div class="container" id="container_div">
208.
209.         <a name="top"></a>
210.
211.         <div class="header_div">
212.
213.             <div style="display:run-in;float:left;">
214.
215.                 <font class="graytext">RSS Feed:</font>
216.
217.                 <?php $options = ShowFeedOptions($feeds_xml); ?>
218.
219.                 <input type="checkbox" id="showtext" onChange="changeOptions(this)" checked>
220.
221.                 <font class="graytext"> Mostrar Texto Completo</font>
222.
223.                 <input type="checkbox" id="showimages" onChange="changeOptions(this)" checked>
224.
225.                 <font class="graytext">Mostrar Imagens</font>
226.
227.             </div>
228.
229.             <?php SetOptions($options); ?>
230.
231.         </div>
232.
233.         <div class="rss" id="rss_div">
234.             <script> LoadFeed(document.getElementById('rss_feed').value, true); </script>
235.         </div>

```

## Arquivo : ajax.php

```

1.     <?php
2.
3.     header('Content-type: text/xml; charset=utf-8');
4.
5.
6.     if ( isset($_GET['loadfeed']) ){
7.
8.         if ( $_GET['loadfeed'] != "" ){
9.
10.             // Obtendo dados da Internet
11.             $feed = "http://" . $_GET['loadfeed'];
12.
13.             // Checando o Encoding
14.             $data = trim( file_get_contents(htmlspecialchars_decode($feed)) );
15.
16.             // Carregando o feed
17.             $xml = simplexml_load_string($data);
18.
19.             // Em caso de falha
20.             if ( !$xml ){
21.                 echo "Error Loading Feed: ".htmlentities($_GET['loadfeed'])."<br>";

```

```

22.         return;
23.     }
24.
25.
26.     $i = 1;
27.
28.     // Mostrando a imagem que encabeça o feed
29.
30.     if ( $xml->channel->image->url != "" ){
31.
32.         $url = $xml->channel->image->url;
33.         $title = $xml->channel->image->title;
34.         echo "<img src=\"\$url\" alt=\"\$title\" style=\"margin-bottom: 20px; margin-top: 20px;
display:block; margin-left: auto;                margin-right: auto; \">>";
35.     }
36.     else echo "<br><br><br>";
37.
38.     // Mostrando os feeds
39.
40.     foreach($xml->xpath('//item') as $item){
41.
42.
43.         //echo "<div>";
44.         //echo "<div class=\"rss_light\">";
45.
46.         // Decidindo se o feed é par ou ímpar para efeitos de layout
47.
48.         if($i%2) {echo "<div class=\"rss_light\">";}
49.
50.         else {echo "<div class=\"rss_dark\">";}
51.
52.         $i++;
53.
54.         // Mostrando o link
55.         echo "<a class=\"rss_link\" href=\"\$item->link\" target=\"_blank\">";
56.
57.         // Mostrando o título
58.         echo "<font class=\"rss_title\">$item->title</font>";
59.         echo "</a>"; // end the link
60.
61.         // Link para o topo
62.         //echo "&nbsp;&nbsp;&nbsp;<a class=\"top\" href=\"#top\">Ir para o topo</a>";
63.
64.         echo "<font class=\"rss_date\">$item->pubDate</font>";
65.
66.         // Mostrando a descrição segundo decisão do usuário
67.         if ( isset($_GET['fulltext']) && $_GET['fulltext'] == "true" ){
68.
69.             if ( isset($_GET['showimages']) && $_GET['showimages'] == "false" ){
70.
71.                 $description = strip_tags($item->description, '<a>');
72.                 echo "<font class=\"rss_description\">$description</font><br>";
73.             }
74.             else echo "<br><font class=\"rss_description\">$item->description</font>";
75.         }
76.
77.         echo "<hr size=\"1\" style=\"color:gray\"></div>";
78.
79.     }//end foreach
80.
81.     // A imagem do feed também vai embaixo da página
82.
83.     if ( $xml->channel->image->url != "" ){
84.
85.         $url = $xml->channel->image->url;
86.         $title = $xml->channel->image->title;

```



```

87.             echo "<img src=\"\$url\" alt=\"\$title\" style=\"margin-bottom: 20px; margin-top: 20px;
display:block; margin-left: auto;
88.             margin-right: auto; \">";
89.             }
90.         }
91.     }
92.
93.     ?>

```

## Arquivo : style.css

```

1.  body{
2.      overflow: auto;
3.      font-family: Helvetica;}
4.
5.  div.header_div{
6.      position: relative;
7.      /*background-color: #EFEFEF;*/
8.      width: 100%;
9.      overflow: auto;
10.     margin: auto;}
11.
12.  div.container{
13.      /*border: 1px solid black;
14.      background-color: #EFEFEF;*/
15.      position: relative;
16.      margin: auto;
17.      display: block;
18.      overflow: auto;
19.      /* SPECIFY HEIGHT & WIDTH AS DESIRED */
20.      height: 100%;
21.      width: 99%;}
22.
23.  div.rss_light{
24.      width: 99%;
25.      margin: 2px;
26.      background-color: #F0F0F0 ;
27.      position: relative;
28.      /*border: 1px thin gray;*/
29.      padding-top: 0px;
30.      padding-bottom:0px;}
31.
32.  div.rss_dark{
33.      width: 99%;
34.      margin: 2px;
35.      background-color: #E0E0E0 ;
36.      position: relative;
37.      /*border: 1px thin gray;*/
38.      padding-top: 0px;
39.      padding-bottom:0px;}
40.
41.  img.loading{
42.      position: relative;
43.      margin-top: 100px;}
44.
45.  a.rss_link,a.rss_link:active{
46.      display: inline;
47.      color: blue;
48.      text-decoration:none;}
49.
50.  a.rss_link:hover{color: #BBBBFF;}
51.
52.  a.rss_link:visited{color: gray;}
53.
54.  a.top,a.top:active{
55.      text-decoration: none;

```

```
56.         font-size: 10pt;
57.         display: inline;
58.         color: gray;}
59.
60.     a.top:hover {
61.         /*text-decoration: underline;*/
62.         color: black;
63.         display: inline;}
64.
65.     font.graytext{
66.         font-size: 10pt;
67.         color: #444444;}
68.
69.     font.rss_title{
70.         font-size: 12pt;
71.         font-weight: bold;
72.         text-decoration: none;}
73.
74.     font.rss_date{
75.         font-size: 9pt;
76.         font-weight: normal;
77.         color: black;
78.         display: block;
79.         font-style:italic;}
80.
81.     font.rss_description{
82.         font-size: 10pt;
83.         color: black;}
84.
85.     select{
86.         color: black;
87.         font-size: 11pt;
88.         background-color: #EAEAEA;}
```