

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
CURSO DE ESPECIALIZAÇÃO EM WEB E SISTEMAS DE INFORMAÇÃO

JOSÉ LUIZ SILVEIRA LIMEIRA

## **Utilização de AJAX no desenvolvimento de sistemas Web**

Trabalho de Conclusão de Curso apresentado  
como requisito parcial para a obtenção do grau  
de Especialista

Prof. ME. Henrique Jorge Brodbeck  
Orientador

Prof. Dr. Carlos Alberto Heuser  
Coordenador do Curso

Porto Alegre, novembro de 2006.

## CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Limeira, José Luiz Silveira

Utilização de AJAX no desenvolvimento de sistemas Web / José Luiz Silveira Limeira – Porto Alegre: Curso de Especialização em Web e Sistemas de Informação, 2006.

42 f.:il.

Trabalho de Conclusão de Curso (especialização) – Universidade Federal do Rio Grande do Sul. Curso de Especialização em Web e Sistemas de Informação, Porto Alegre, BR – RS, 2005. Orientador: Henrique Jorge Brodbeck.

1. Desenvolvimento Web. 2.AJAX. 3.XHTML. 4.DOM. 5.CSS. 6.JavaScript. 7.XML. I. Brodbeck, Henrique Jorge. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. José Carlos Ferraz Hennemann

Vice-Reitor: Prof. Pedro Cezar Dutra Fonseca

Pró-Reitora de Pós-Graduação: Profa. Valquiria Linck Bassani

Diretor do Instituto de Informática: Prof. Flávio Rech Wagner

Chefe do Departamento de Informática Aplicada: Prof. Dra. Carla Maria Dal Sasso Freitas

Coordenador do Curso de Especialização em Web e Sistemas de Informação:

Prof. Dr. Carlos Alberto Heuser

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

## **AGRADECIMENTOS**

Inicialmente, agradeço ao Serpro, empresa que trabalho, pelo apoio financeiro para minha participação nesta especialização.

Agradeço, também, minha esposa, Michele, pelo apoio para que eu voltasse a me dedicar aos estudos.

Agradeço aos colegas de turma, que muito contribuíram para o alto nível das aulas e dos trabalhos em grupo.

Por fim, agradeço ao prof. Brodbeck, pelos ensinamentos e pelas orientações passadas, que muito me ajudaram na elaboração desta monografia.

# SUMÁRIO

<b>LISTA DE ABREVIATURAS E SIGLAS</b> .....	<b>5</b>
<b>LISTA DE FIGURAS</b> .....	<b>6</b>
<b>LISTA DE TABELAS</b> .....	<b>7</b>
<b>LISTAGENS</b> .....	<b>8</b>
<b>RESUMO</b> .....	<b>9</b>
<b>ABSTRACT</b> .....	<b>10</b>
<b>1 INTRODUÇÃO</b> .....	<b>11</b>
<b>2 AJAX – TECNOLOGIAS E APLICAÇÕES</b> .....	<b>12</b>
2.1 O que é AJAX.....	12
2.2 Principais Vantagens.....	16
2.3 Tecnologias envolvidas.....	17
2.4 Exemplos de aplicações existentes.....	22
<b>3 FRAMEWORKS PARA AJAX</b> .....	<b>24</b>
<b>4 ESTUDO DE CASO: SITE FINAL SPORTS</b> .....	<b>28</b>
<b>5 FUTURO DO AJAX</b> .....	<b>33</b>
<b>6 CONCLUSÃO</b> .....	<b>34</b>
<b>REFERÊNCIAS</b> .....	<b>36</b>
<b>APÊNDICE: CÓDIGO FONTE DO ESTUDO DE CASO</b> .....	<b>38</b>

## LISTA DE ABREVIATURAS E SIGLAS

AJAX	Asynchronous JavaScript And Xml
API	Application Programming Interfaces
CSS	Cascading Style Sheets
DOM	Document Object Model
HTML	HyperText Markup Language
IDE	Integrated Development Enviroment
SOAP	Simple Object Access Protocol
UDDI	Universal Description Discovery and Integration
Web	World Wide Web, ou teia de alcance mundial
W3C	World Wide Web Consortion
XHTML	Extensible HyperText Markup Language
XML	Extensible Markup Language
XSLT	Extensible Stylesheet Language Transformations

## LISTA DE FIGURAS

Figura 2.1: Estrutura de uma aplicação Web tradicional (SOUSA, 2006) .....	13
Figura 2.2: Estrutura de uma aplicação Web utilizando AJAX (SOUSA, 2006).....	14
Figura 2.3: Comparativo entre Web tradicional e AJAX (GARRET, 2006) .....	15
Figura 2.4: Exemplo de utilização de AJAX.....	16
Figura 2.5: Arquivo sumGet.htm comentando uso de AJAX .....	21
Figura 2.6: Arquivo sumGet.php com script do servidor .....	21
Figura 4.1: Lista com clubes para serem selecionados. ....	29
Figura 4.2: Página mostrando os campeonatos que o clube jogou. ....	30
Figura 4.3: Página mostrando jogos selecionados para o clube no campeonato.....	31
Figura 4.4: Página mostrando as estatísticas do jogo selecionado. ....	32

## LISTA DE TABELAS

Tabela 2.1: Métodos do objeto XMLHttpRequest .....	19
Tabela 2.2: Propriedades do objeto XMLHttpRequest.....	19

## LISTAGENS

Listagem 1 : Descrição do arquivo principal: estatísticas.php. ....	39
Listagem 2 : Descrição do arquivo buscar_campeonatos.php. ....	40
Listagem 3 : Descrição do arquivo mostrar_jogos_campeonato.php. ....	41
Listagem 4 : Descrição do arquivo medias_jogo.php. ....	44
Listagem 5 : Descrição do arquivo ajax.css. ....	44

## RESUMO

Este trabalho tem por objetivo fazer um estudo sobre o desenvolvimento de aplicações Web utilizando AJAX. São abordadas as características das tecnologias envolvidas neste desenvolvimento e a maneira como são integradas. É feita uma apresentação sobre os vários frameworks que facilitam o desenvolvimento, o futuro do AJAX e mostrado um estudo de caso de uma aplicação que utilize recursos do AJAX

**Palavras-Chave:** Desenvolvimento Web, AJAX, XHTML, DOM, CSS, JavaScript e XML.

## **Use of AJAX in Web systems development**

### **ABSTRACT**

This paper aims to study Web applications development using AJAX. The characteristics of the technologies involved in this development are studied along with the manner these are integrated. A number of frameworks that facilitate the development of applications are presented, as well as the future of AJAX and a case study of an application that uses the AJAX resources are showed.

**Keywords:** Web Development, AJAX, XHTML, DOM, CSS, JavaScript e XML.

# 1 INTRODUÇÃO

O objetivo deste trabalho é fazer um estudo sobre o desenvolvimento de aplicações Web, utilizando recursos do AJAX. Ao longo do trabalho, busca-se conceituar as tecnologias envolvidas com AJAX e como elas trabalham de forma integrada para atingir o melhor resultado.

O termo AJAX é uma sigla para “Asynchronous JavaScript And Xml” e é utilizado para especificar um conjunto de tecnologias que ajudam as aplicações Web a ficarem mais dinâmicas, evitando o carregamento de toda página, só atualizando o que for realmente necessário na página.

No capítulo 2 será feito um comparativo entre a forma de desenvolvimento tradicional e o desenvolvimento com AJAX, mostrando as vantagens da utilização de AJAX. Apresentam-se, também, as tecnologias envolvidas, seus principais métodos e propriedades e como integrá-las. Também será dedicado um espaço para comentar os principais sites que utilizam AJAX e os seus principais recursos.

Em seguida, este trabalho, no terceiro capítulo, apresenta alguns frameworks para AJAX, focando nas suas principais características e facilidades. São mostradas as principais bibliotecas de funções que interagem com IDEs para alcançar os melhores resultados no desenvolvimento de aplicações para a Web.

A discussão teórica culmina, no capítulo 4, num estudo de caso, no qual comenta-se passo-a-passo como pode ser adaptado um site de esportes para utilizar os recursos do AJAX. Serão demonstradas as alterações necessárias no código das páginas e como ficará o funcionamento do site depois que os recursos do AJAX forem incluídos.

No capítulo 5 é explorado o futuro do desenvolvimento em AJAX, procurando apresentar o que levará as suas tecnologias serem adotadas para o desenvolvimento Web. Com este trabalho, pretende-se colaborar para apresentar uma nova forma de desenvolvimento de aplicações voltadas para a Web.

## 2 AJAX – TECNOLOGIAS E APLICAÇÕES

Este capítulo irá abordar o significado de AJAX, suas tecnologias envolvidas e mostrar alguns exemplos de aplicações existentes.

### 2.1 O que é AJAX

O AJAX pode ser descrito conforme definição abaixo:

AJAX é um acrônimo consagrado muito recentemente por Jesse James Garret, da Adaptive Path, e significa Asynchronous JavaScript and XML (JavaScript e XML assíncrono), porém o que temos é muito mais mais que a junção de JavaScript com XML, é todo um conceito de navegação e atualização de páginas Web. Algumas partes descritas na definição de AJAX não são novas, as quais muitas vezes foram denominadas de DHTML (HTML Dinâmico) e Script Remoto (SOARES, 2006, p. 17).

O AJAX surgiu para resolver um problema que ocorre desde o surgimento da Internet, que é o de que a interação é feita de forma síncrona, ou seja, exige-se que, para cada solicitação em uma página Web, atualize-se a página inteira no navegador. Não importava se a atualização era só de uma pequena parte da página, toda a página era recebida pelo servidor e a página era toda redesenhada e retornada para o navegador Web.

Antes do surgimento do AJAX, alguns subterfúgios eram utilizados para amenizar esse problema. A utilização de pré-carregamento dos dados, frames e iframes nunca resolveram o problema da maneira mais eficiente. Em alguns casos, apenas davam a falsa impressão de que a página não era toda atualizada, quando, na verdade, os dados já estavam carregados e apenas eram escondidos até serem necessários. Em outros casos (frames ou iframes), eram carregadas partes da página e, geralmente, trafegavam mais informações do que era realmente necessário.

Com o AJAX pode-se trafegar apenas os dados que realmente foram atualizados em uma página Web e assim ganhar em desempenho e interatividade com o usuário. O seu uso é recomendado, principalmente, para páginas que utilizem validações e preenchimento de formulários.

O termo Web 2.0 veio junto com o surgimento do AJAX. Tim O'Reilly, no artigo intitulado "What is Web 2.0" (REILLY, 2005), escreveu a melhor definição sobre a Web 2.0. Ele destaca a revolução causada na forma de apresentação das informações e interação com o usuário. O AJAX aparece como parte essencial nessa revolução de

conceitos. Mas coube a Jesse James Garrett da Adaptive Path usar pela primeira vez o termo AJAX, em fevereiro de 2005, em seu artigo intitulado “AJAX: A New Approach to Web Applications” (GARRET, 2006).

Praticamente todos os navegadores, em suas últimas versões, são compatíveis com as técnicas utilizadas no AJAX, viabilizando o seu uso em qualquer aplicação Web.

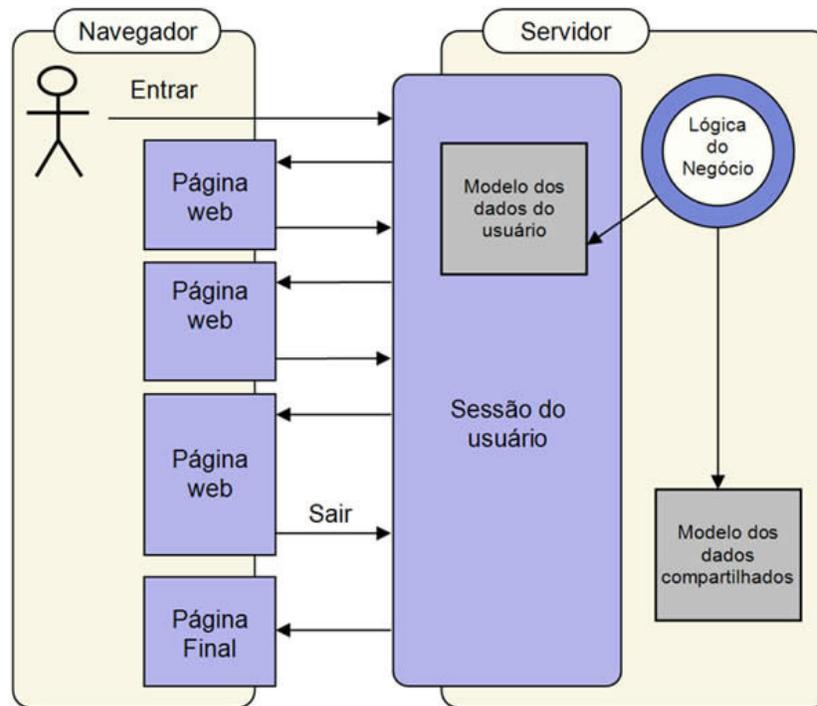


Figura 2.1: Estrutura de uma aplicação Web tradicional (SOUSA, 2006)

Na figura 2.1, pode-se visualizar a estrutura tradicional da Web, com o fluxo das informações, sendo transferido toda a página Web entre o servidor e o navegador cada vez que é solicitada qualquer informação da página.

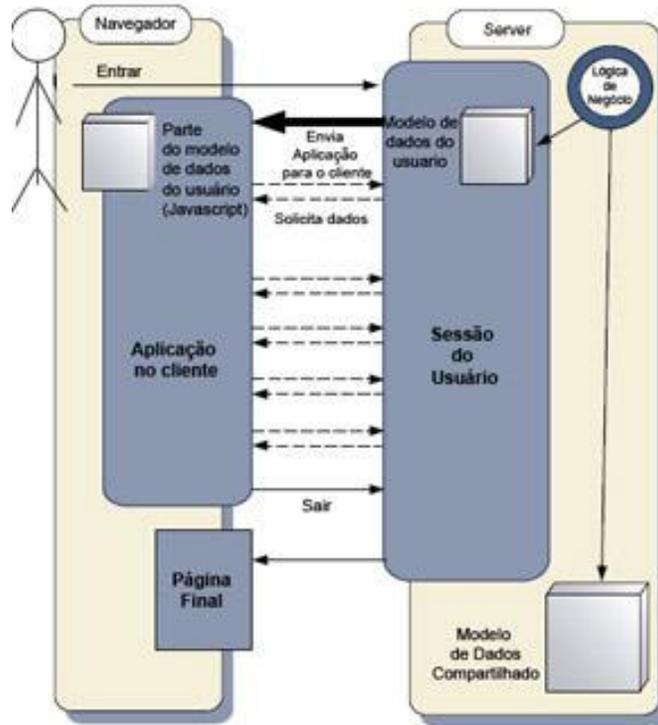


Figura 2.2: Estrutura de uma aplicação Web utilizando AJAX (SOUSA, 2006)

Na figura 2.2, observa-se um fluxo de informações utilizando AJAX. Neste caso, o navegador interage com o servidor, passando apenas as informações solicitadas e apenas nos momentos que realmente são necessárias, evitando, assim, o tráfego de toda página a cada requisição.

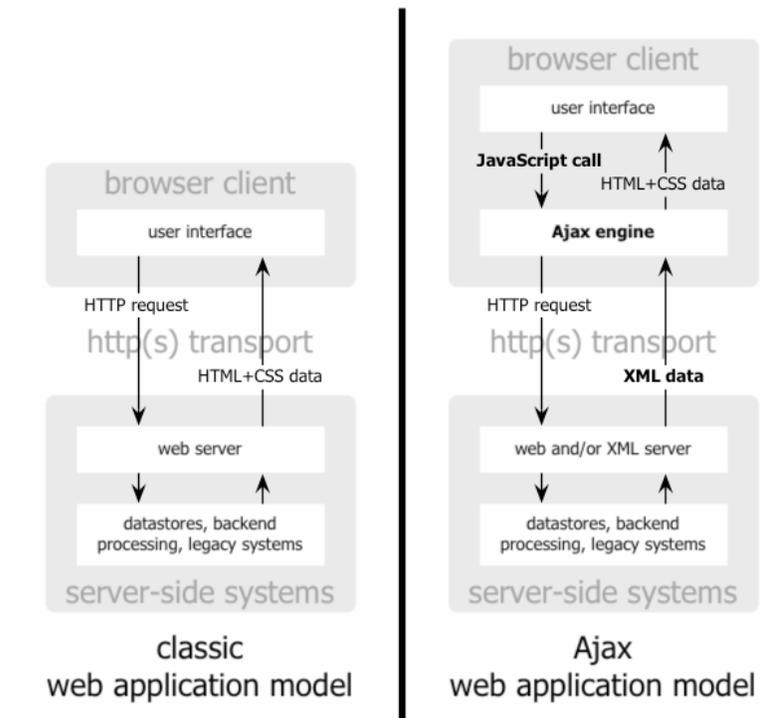


Figura 2.3: Comparativo entre Web tradicional e AJAX (GARRET, 2006)

Na figura 2.3, percebe-se bem a diferença entre os dois modelos e os módulos envolvidos. À esquerda, no modelo tradicional, tem-se a interface do usuário interagindo diretamente com o servidor Web para enviar a página inteira para o servidor e depois receber uma nova página inteira para ser mostrada ao usuário que fica aguardando o retorno sem conseguir fazer qualquer outra operação na página. À direita, no modelo AJAX, a interface do usuário interage com um mecanismo AJAX que intermedia as solicitações entre o cliente e servidor, só passando as informações necessárias de ambos os lados, permitindo que o usuário, quando solicite algo da página Web, possa continuar trabalhando na página enquanto o resultado ainda não é mostrado.

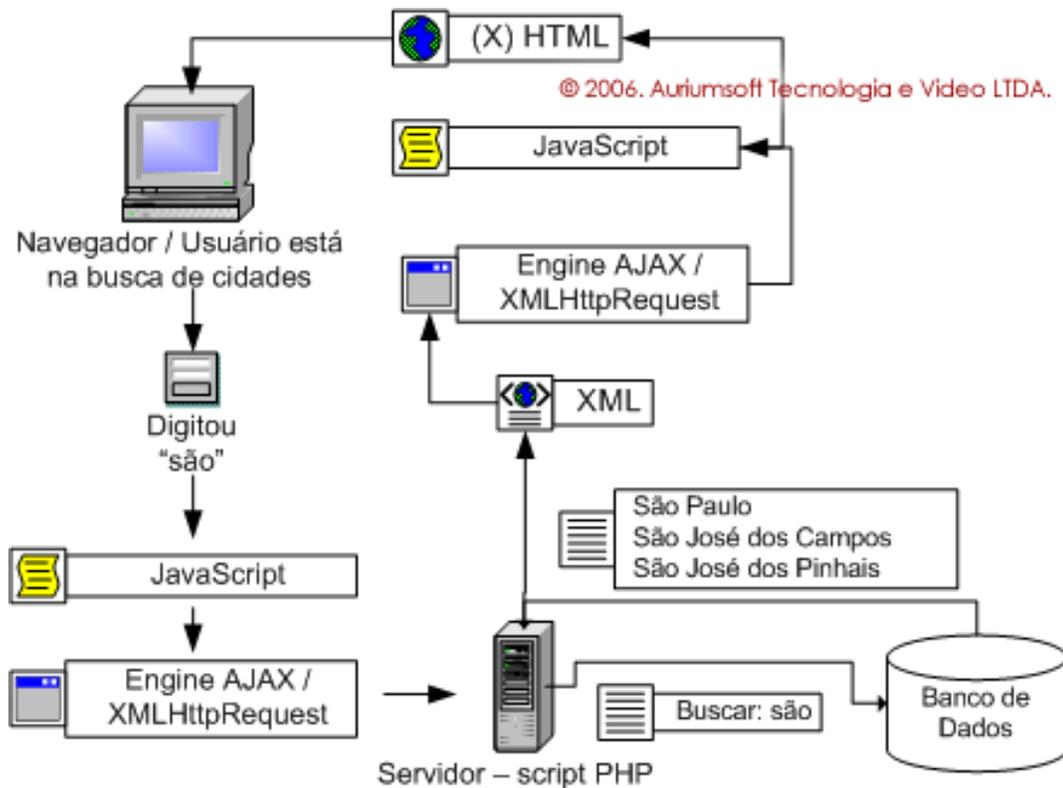


Figura 2.4: Exemplo de utilização de AJAX (BAUDISCH, 2006)

Na figura 2.4, é demonstrado um exemplo do processo de atualização de página usando AJAX. Neste exemplo, o usuário digita a palavra “são” e o javascript dispara um evento que chama o mecanismo AJAX. Este invoca o script apropriado no servidor para buscar no banco de dados as cidades que iniciam com a palavra “são”. Monta o retorno em XML e devolve para o mecanismo AJAX. Este se encarrega de interagir com a página HTML apenas atualizando o objeto apropriado, sem precisar atualizar todas as páginas, mostrando as cidades selecionadas no navegador do usuário.

## 2.2 Principais Vantagens

### Menor uso de banda

Utilizando AJAX, menos informações irão trafegar na rede, diminuindo sensivelmente o uso de banda e tornando a navegação muito mais rápida.

### Respostas mais rápidas

Com AJAX, os usuários não precisam esperar que toda a página seja processada a cada vez que fazem uma solicitação. O usuário pode interagir com uma parte da página e enquanto aguarda o retorno, pode continuar interagindo com as demais seções da página, tornando a navegação mais rápida e eficiente.

### **Mais interatividade**

Com AJAX, sem precisar usar recursos pesados de programação ou necessidade de softwares proprietários, é possível disponibilizar aplicações com alta capacidade de interatividade e usabilidade. As interfaces são mais amigáveis e divertidas para o usuário final.

## **2.3 Tecnologias envolvidas**

AJAX não é uma tecnologia por si só, na verdade, é um termo utilizado para a reunião de várias tecnologias. AJAX incorpora e utiliza as tecnologias abaixo descritas:

- Padrões de apresentação utilizando **XHTML** e **CSS**;
- Apresentações dinâmicas e interativas utilizando **Document Object Model**;
- Manipulação de elementos de uma página utilizando **XML** e **XSLT**;
- Recuperação de dados assíncrona usando **XMLHttpRequest**;
- e **JavaScript**, que junta tudo isso. (GARRET, 2005)

Descrever-se-á, agora, em detalhes, cada uma dessas tecnologias e outras que podem ser utilizadas para apresentarem resultados dentro dos conceitos do AJAX:

**XHTML** é a linguagem HTML baseada em XML. Com ela pode-se combinar marcações HTML com as regras XML. A idéia é ter um documento padronizado e acessível para ser apresentado em várias plataformas e dispositivos, seja Computador Desktop, PDA, Telefone Celular, Televisão, etc.

**CSS** (Cascading Style Sheets), ou folhas de estilos em cascata, é outra tecnologia importante. Com CSS é possível definir padrões de aparência e comportamento dos elementos de uma página. As cores, parágrafos, margens, espaçamentos, espessura ou qualquer outro elemento da página, pode ser definido em apenas um arquivo e, assim, a qualquer momento, pode-se mudar totalmente a forma de visualização de todas as páginas de um site, apenas alterando esse arquivo que contém as definições de CSS.

Esse recurso torna possível que, em conjunto com AJAX, se possa mudar dinamicamente a aparência do site, para se adaptar ao gosto de cada usuário.

**DOM** é uma interface de programação de aplicativos (API) para documentos HTML e XML. É a definição da estrutura lógica dos documentos e o meio pelo qual um documento é acessado e manipulado.

Dessa forma, o XML apresenta as informações como documentos e o DOM pode ser utilizado para o gerenciamento dessas informações.

Com a API DOM é possível criar documentos, navegar pela estrutura de um documento qualquer, incluir, alterar e apagar nós do documento (o próprio nó ou seu conteúdo). Isso é válido tanto para páginas HTML quanto para documentos XML. Ou seja, na grande maioria das vezes utilizar-se-á a API DOM para manipular uma página Web pelo AJAX. (SOARES, 2006, p.68)

DOM é uma especificação do W3C, portanto, é um padrão para acessar e alterar a estrutura de documentos HTML e XML.

Em uma página Web típica (bem formada), os elementos são estruturados em árvore, com o elemento HTML sendo a raiz, seguido pelos elementos HEAD e BODY, os quais representam o cabeçalho e o corpo do documento respectivamente. Dentro do corpo do documento (body) encontram-se os demais elementos, tais como tabelas, parágrafos, listas, divisores, hiperlinks e muitos outros.

**XML** ou eXtensible Markup Language é um dos pilares do AJAX (o X mais precisamente), pois é através dele que se pode manipular os elementos de uma página (como elementos DOM) ou processar as respostas do servidor.

**XSLT** é a linguagem para transformar documentos XML. Com ela, documentos XML podem ser transformados em um outro documento, mas com nós incluídos, alterados ou até removidos. XSLT é uma linguagem declarativa, na qual se pode especificar regras que irão transformar um arquivo XML no formato desejado. Combinando XSLT com JavaScript, pode-se alterar um documento dinamicamente, buscar informações do servidor, construir o formato de apresentação e mostrar o resultado final em uma página HTML, utilizando assim os recursos do AJAX.

**XMLHttpRequest** é o objeto que faz a conexão assíncrona entre a página e o servidor de aplicações Web, ou seja, é a tecnologia principal do AJAX, sem ele, o AJAX não existiria. Trata-se de um objeto JavaScript que pode ser usado para fazer requisições ao servidor Web, em segundo plano, sem congelar o navegador ou recarregar a página atual. O objeto XMLHttpRequest é hoje parte da especificação do DOM, nível 3. Ou seja, qualquer navegador que queira oferecer suporte aos padrões precisa implementar o objeto XMLHttpRequest. (SOARES, 2006, p.83)

XMLHttpRequest troca informações com o servidor Web, geralmente utilizando XML, embora possa utilizar qualquer formato de texto, incluindo HTML, plain text e JSON (Java Script Object Notation).

O XMLHttpRequest foi criado inicialmente no navegador Internet Explorer 5 como um componente do ActiveX. Depois disso, outros navegadores passaram a implementá-lo, mas o grande problema é que o XMLHttpRequest não é um padrão W3C e por isso cada navegador pode implementar de maneiras diferentes. Atualmente, o Mozilla Firefox, Konqueror, Safari e Opera implementam o XMLHttpRequest da mesma forma, ou seja, como um objeto JavaScript nativo. Já o Internet Explorer prefere implementar esse mesmo recurso utilizando outro objeto, o ActiveX.

Exemplo da criação do objeto XMLHttpRequest no Internet Explorer:

```
var xmlhttp=new ActiveXObject("Microsoft.XMLHttp")
```

Exemplo da criação do objeto XMLHttpRequest nos demais navegadores:

```
var xmlhttp=new XMLHttpRequest()
```

O XMLHttpRequest possui métodos e propriedades, conforme tabelas a seguir:

Tabela 2.1: Métodos do objeto XMLHttpRequest

<b>Métodos</b>	<b>Descrição</b>
Open(método, url, Assíncrona, usuário, senha)	Esse método relaciona o objeto à página Web que se deseja conectar. O argumento de método pode ser GET, POST ou PUT. O endereço da url pode ser relativo ou absoluto. Os 3 últimos parâmetros são opcionais.
Send(content)	Envia a solicitação para o Servidor. Caso a conexão tenha sido aberta com o parâmetro Assinc igual a false (indicando que a conexão não é assíncrona), esse método aguarda a resposta do servidor; caso contrário, não há espera (o que deve ser o padrão para aplicações AJAX).
setRequestHeader	Configura o cabeçalho http especificado com o valor fornecido.
getResponseHeader	Retorna o valor da string do cabeçalho especificado.
getAllResponseHeaders	Retorna uma string com todos os cabeçalhos http especificados.
Abort	Interrompe o processamento atual do objeto XMLHttpRequest.

Fonte: ASLESON, 2006, p.25.

Tabela 2.2: Propriedades do objeto XMLHttpRequest

<b>Propriedades</b>	<b>Descrição</b>
Status	Contém o código de status enviado pelo servidor Web. (isto é, 200 para OK, 404 para Não Encontrado e assim por diante)
statusText	A versão em texto do código de status http. (isto é, OK ou Not Found e assim por diante)
readyState	O estado da solicitação. Os cinco valores possíveis

	são 0- não inicializada, 1- carregando, 2- carregada, 3- interativa e 4- concluída.
responseText	A resposta do servidor na forma de uma string.
responseXML	A resposta do servidor em formato XML.
Onreadystatechange	O manipulador de eventos que é acionado a cada mudança de estado, normalmente, uma chamada a uma função JavaScript.

Fonte: SOARES, 2006, p.88-92.

Na figura 2.5, tem-se um código exemplo comentado e que mostra bem a utilização dos métodos e propriedades do AJAX. O exemplo solicita a entrada de dois valores e ao clicar no botão é feita a soma dos valores e apresentado o resultado. A figura 2.5 mostra o arquivo cliente, com as chamadas para as rotinas JavaScript que irão instanciar o objeto XMLHttpRequest. Na figura 2.6 é mostrado o script do servidor invocado para fazer o cálculo e envio das informações atualizadas para a página.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>AJAX (xmlhttprequest+form+get)</title>
<script language="javascript" type="text/javascript">
function createXMLHttpRequest() {
/*
Interface genérica para acesso ao conector remoto.
*/
    try{ return new ActiveXObject("Msxml2.XMLHTTP"); }catch(e){}
    try{ return new ActiveXObject("Microsoft.XMLHTTP"); }catch(e){}
    try{ return new XMLHttpRequest(); }catch(e){}
    alert("XMLHttpRequest not supported");
    return null;
}
// contem objeto para solicitar recursos remotos.
var xhReq = createXMLHttpRequest();
var valor1 = document.getElementById("valor1");
// primeiro campo da soma
var valor2 = document.getElementById("valor2");

function xhSend(url){
/*
envia a requisicao para o servidor, e tudo o que eh
retornado eh passado a funcao do_readyStateChange
*/

```

```

    xhrReq.open("GET",url,true);
    xhrReq.onreadystatechange = do_readyStateChange;
    xhrReq.send(null);
}
function do_readyStateChange(){
  /*
  se a requisicao estiver completa entao exibir o
  valor no campo de formulario com id total.
  */
  if (xhrReq.readyState != 4) { return; }
  var total = document.getElementById("total");
  total.value = xhrReq.responseText ;
}
//alert(serverResponse);
</script>
</head>

<body>
<h1>XML HTTP REQUEST via GET</h1>
<form id="a">
  <input id="valor1" type="text" size="5"/><input id="valor2" type="text" size="5" />
  =<input id="total" type="text" size="5" />
  <input type="button" value="send"
onClick="javascript:xhSend('sumGet.php?v1='+valor1.value+'&v2='+valor2.value+'');">
</form>
<p> <font size="2">
  <i>Envia a soma dos 2 campos e retorna o resultado usando xml http request.</i>
</font></p>
<p><a href="sumGet.phps">
  <i><font size="2">Clique aqui para exibir o código fonte do script no
servidor.</font></i>
</a></p>
</body>
</html>

```

Figura 2.5: Arquivo sumGet.htm comentando uso de AJAX (PEREIRA, 2005)

```

<?
# faz a soma entre os dois valores passados via GET.
echo(( (int)($_GET['v1'])) + ((int)($_GET['v2'])));
?>

```

Figura 2.6: Arquivo sumGet.phps com script do servidor (PEREIRA, 2005)

**JavaScript** é uma linguagem de scripts voltada para executar operações no lado do cliente. Com ela, é possível fazer validações de campos, abertura de janelas, controle da utilização de botões, mensagens de alertas, confirmações e principalmente para criar

uma interatividade maior do usuário com a página utilizada. Com JavaScript também é possível alterar o estilo de uma página de forma dinâmica.

JavaScript não é uma linguagem compilada, ela é interpretada e é inserida junto com o código fonte HTML das páginas. Apesar de não possuir alguns conceitos importantes de orientação a objeto (definição de classe e herança explícita), ainda assim encontram-se alguns conceitos de orientação a objetos no JavaScript, como o conceito de objeto, herança implícita, encapsulamento e o polimorfismo.

JavaScript é considerado um dos fundamentos do AJAX, pois é ele que permite que o conceito do AJAX de atualizar partes de uma página Web possa ser implementado.

**Web Services**, ou apenas serviços da Web, podem ser usados em conjunto com o AJAX. Para isso, é necessário a utilização de um protocolo de comunicação, sendo mais conhecido o **SOAP**, Simple Object Access Protocol, um protocolo que permite a comunicação entre aplicações, independente da linguagem e plataforma utilizadas por cada uma. Com SOAP, é possível fazer chamadas a procedimentos remotos e ter o retorno em XML. O Yahoo, Google e outros sites disponibilizam Web Services para acesso a algumas de suas aplicações. Através do WSDL, são fornecidos pelos sites, detalhes das funções e parâmetros de cada Web Service no formato XML para que seja fácil a sua utilização por outras aplicações. Os UDDI são as chamadas listas de Web Services que existem no mundo. A UDDI é um serviço que permite que sejam publicados pelos sites todos os Web Services que são disponibilizados para uso externo por outros usuários.

## 2.4 Exemplos de aplicações existentes

Já existem muitas aplicações que utilizam AJAX. Algumas aplicações exploram todos os recursos oferecidos pela tecnologia e outras são mais simples, colocando apenas alguns conceitos de AJAX em seus conteúdos.

A Google, com certeza, é a empresa que mais tem desenvolvido aplicações utilizando recursos do AJAX. Entre as aplicações da Google que utilizam AJAX, se destacam o Gmail, o Google Calendar, o Google Maps, o Google Suggest e o Orkut.

No Google Suggest (GOOGLE, 2006), à medida que se vai digitando um texto, vão aparecendo sugestões de textos associados e a quantidade de páginas encontradas com cada texto.

Já no Google Maps (MAPS, 2006), é possível visualizar mapas de localidades de forma fluída, com opção de zoom e diversas outras informações.

No Gmail (GMAIL, 2006), visualizam-se mensagens específicas sem precisar fazer o *refresh* de toda página. Além disso, os arquivos anexos, contatos e outras informações, vão aparecendo na tela à medida que seu carregamento é feito, com recursos de AJAX, possibilitando que o usuário possa ver algumas informações na tela enquanto outras têm o seu carregamento efetuado, agilizando a navegação do usuário no site.

No Gmail e Orkut (ORKUT, 2006), o acesso dos usuários é feito através de recursos do AJAX, para agilizar a verificação das informações e redirecionamento das páginas a

serem mostradas. No Orkut, várias páginas usam recursos do AJAX, uma delas é a página inicial de consulta de um usuário, aonde os perfis Social, Profissional e Pessoal podem ser consultados sem precisar atualizar toda a página.

No Yahoo, a seção de e-mail também oferece recursos de AJAX, o que possibilita que a interface do Yahoo Mail (YAHOO, 2006) seja mais rápida para leitura das mensagens. Além disso, é possível usar recursos de *drag-and-drop* (arrastar e soltar) para enviar as mensagens rapidamente para outra pasta.

Já a Amazon, disponibiliza em seu site uma ferramenta de pesquisa de diamantes (AMAZON, 2006). São mostrados aos usuários controles de preço, cor, forma e outras características dos diamantes e, através de controles deslizantes, o usuário restringe sua pesquisa. À medida que os controles são ajustados, o site mostra a quantidade de diamantes encontrados. Para isso, recursos de AJAX são empregados, criando uma interatividade bastante interessante durante a seleção das opções.

Backbase (BACKBASE, 2006) é provavelmente a empresa que tem o sistema mais orientado a AJAX. Seu site apresenta vários recursos AJAX, sendo que seu mecanismo é desenvolvida toda em JavaScript puro. No site, existem várias demonstrações de aplicações AJAX disponíveis para download.

O Flickr (FLICKR, 2006), site para armazenar imagens, utiliza vários recursos de AJAX, entre eles, o de fazer buscas por palavras chaves ou características e os resultados das imagens encontradas são mostradas sem fazer nenhum refresh nos demais itens da página.

### 3 FRAMEWORKS PARA AJAX

Para facilitar o desenvolvimento de aplicações Web, pode-se utilizar um Framework, que irá agilizar, descomplicar e padronizar as aplicações Web. A grande vantagem dos frameworks é que eles geram o código pronto de muitas operações básicas e com isso os desenvolvedores de aplicações Web ficam com mais tempo disponível para implementar as tarefas mais difíceis dos sites, pois das mais comuns, os frameworks já se encarregam.

Entre as tarefas que os frameworks implementam estão os recursos em AJAX, como a utilização do objeto XMLHttpRequest. Com isso, é possível que boa parte da implementação necessária para que os módulos interajam com o envio de solicitações entre o cliente e o servidor possa ser implementada e gerenciada pelos frameworks.

Existem dezenas de frameworks para AJAX. Abaixo estão selecionados alguns dos Frameworks Web que implementam os recursos do AJAX:

#### **AJAXLIB**

É uma classe escrita em JavaScript e que pode ser utilizada em conjunto com várias linguagens, como PHP, PERL e JSP. É uma ferramenta simples e indicada para aplicações que não requerem muitos recursos.

Esse produto possui código aberto. (AJAXLIB, 2006)

#### **ATLAS**

É um framework da Microsoft que vem integrado ao ASP.NET. Com um biblioteca que contém vários componentes prontos, ele permite facilmente incorporar no site funções de *drag-and-drop*, menus dinâmicos, *popup* com informações temporárias (Exemplo: calendário), controle de painéis na página, blocos sempre visíveis mesmo com rolagem da página, controle de campos *dropdown* em cascata (alterando um campo atualiza os demais), controle de animações, etc. No site do produto, é possível ver todos os componentes disponíveis para o Atlas e exemplos interessantes para sua utilização. (ATLAS, 2006)

## **DOJO**

O Dojo é um toolkit DHTML escrito em Javascript e independente de plataforma. Seu foco está no desempenho, por isso é recomendado se o uso de JavaScript for intenso. Possui uma API complexa e sua documentação não é muito completa. É um dos frameworks mais antigos, seu desenvolvimento foi iniciado em setembro de 2004 e hoje é considerado um dos produtos mais maduros.

Um recurso que diferencia o Dojo é seu suporte aos botões Voltar e Avançar, possibilitando registrar um método de retorno de chamada que será acionado se o usuário clicar no botão Voltar ou Avançar. (ASLESON, 2006, p.254)

Esse produto possui código aberto. (DOJO, 2006)

## **DWR**

O DWR (Direct Web Remoting) é um framework Java para rodar aplicações AJAX. DWR é um dos mais conceituados frameworks AJAX disponíveis para a plataforma Java. Tudo funciona de maneira simples e de fácil aprendizado. A estrutura do código fonte Java fica acessível ao cliente via JavaScript, não existindo uma distinção entre lado cliente e servidor do ponto de vista do desenvolvedor. Tudo é feito de forma transparente, pois o DWR cuida de todo processo de criação e manipulação do XMLHttpRequest.

Esse produto possui código aberto. (DWR, 2006)

## **ECHO2**

Echo2 é um framework desenvolvido em Java que cria a interface baseada em templates HTML, não necessitando que o desenvolvedor saiba HTML ou JavaScript, apenas tenha conhecimento em Java. O desenvolvimento fica muito parecido com o feito utilizando o Swing.

Esse produto possui código aberto. (ECHO2, 2006)

## **GWT**

Com o GWT (Google Web Toolkit), é possível transformar uma aplicação feita em Java em aplicações Web com AJAX integrado ao HTML, CSS e JavaScript. A aplicação já fica pronta para rodar em qualquer navegador da maneira mais leve possível.

Além disso, o GWT oferece bibliotecas prontas para que qualquer um possa colocar os recursos de AJAX facilmente em seu site sem precisar maiores conhecimentos de programação.

O GWT foi utilizado pela Google para desenvolver os sites Google Maps e Gmail.

Pode ser integrado com qualquer IDE Java.

Existem versões do GWT para Linux e Windows. (GWT, 2006)

## **MOCHIKIT**

O Mochikit extrai algumas funcionalidades do Python para sua API. A biblioteca também é distribuída com a versão do framework do Python para Web, chamada de TurboGears.

O código do Mochikit é muito limpo e fácil de utilizar. Sua estrutura de log é muito boa. (MOCHIKIT, 2006)

## **PROTOTYPE**

O Prototype Implementa um conjunto inteiro de efeitos visuais, assim como o *drag-and-drop* e alguns componentes de interface com o usuário.

O suporte ao AJAX é razoavelmente direto e bem desenvolvido, uma vez que foi desenvolvido para Ruby on Rails.

Segundo pesquisa realizada em setembro de 2006 pelo site Ajaxian (AJAXIAN, 2006) o Prototype é o framework mais utilizado. O que contribui para isso é a facilidade de uso de suas funções e sua documentação bem detalhada. Seu código é todo escrito em JavaScript.

Esse produto possui código aberto. (PROTOTYPE, 2006)

## **SAJAX**

O SAJAX (Simple AJAX Toolkit) é um framework para AJAX com implementações do lado do servidor. Possui implementações para PHP, ASP, ColdFusion, Perl, Python, Ruby entre outros.

Esse produto possui código aberto. (SAJAX, 2006)

## **SYMFONY**

É um Framework desenvolvido em PHP e traz como características a geração de código tableless e opção de internacionalização. Utiliza o modelo MVC para separação do código gerado.

Esse produto possui código aberto. (SYMFONY, 2006)

## **XAJAX**

XAJAX é uma biblioteca de funções em PHP para facilitar o uso de AJAX. São bibliotecas desenvolvidas como software livre.

Os Frameworks SAJAX e XAJAX possuem uma diferença básica, enquanto no SAJAX o retorno é tratado em uma função JavaScript, no XAJAX o retorno é feito em uma função do PHP, assim como os comandos e eventos adicionados são feitos no PHP, resguardando o código JavaScript. O XAJAX também tem a vantagem de funcionar muito bem nos três principais navegadores: Internet Explorer, Firefox e Opera. (XAJAX, 2006)

## **XOAD**

É um framework desenvolvido em PHP e baseado nos recursos do AJAX. Tem como característica principal ser orientado a objetos. Utiliza JSON e objetos PHP para fazer a comunicação entre cliente e servidor. Possui uma documentação bem completa, descrevendo todas as suas classes, métodos e variáveis. No site oficial, podem ser encontrados vários exemplos de utilizam deste framework.

Esse produto possui código aberto. (XOAD, 2006)

## **Yahoo! UI**

A biblioteca Yahoo! UI (YUI) foi desenvolvida pelo Yahoo! para desenvolvimento dos seus sites e teve seu código aberto junto com uma gama de recursos excelentes para desenvolvedores.

O YUI é mais uma coleção distinta de "utilidades" do que uma biblioteca (como as que foram tratadas aqui), com cinco scripts principais que tratam de: animação, AJAX, manipulação DOM, *drag-and-drop*, tratamento de eventos. A implementação do controle de eventos é um dos atrativos do YUI. (FEITOSA, 2006)

Existem seis controles, chamados de: Calendar, Slider, Menu, AutoComplete, Tree View, Container classes (com o qual você pode executar janelas no estilo de widgets) . Todos esses controles possuem exemplos no site do Yahoo para serem baixados.

É um framework voltado para aplicações menores, pois não possui muitos recursos. (DEVELOPER, 2006)

## 4 ESTUDO DE CASO: SITE FINAL SPORTS

O estudo de caso foi feito analisando o site Final Sports (FINAL, 2006). Esse site é voltado para esportes em geral, focando, principalmente, o futebol gaúcho. A proposta era remodelar a forma como algumas informações eram acessadas e apresentadas para os usuários do site. A seção escolhida foi o Camarote Final, uma seção que apresenta informações sobre os jogos de futebol, notas de jogadores, bolão de apostas e classificação dos campeonatos. A forma como são apresentadas as informações sobre os jogos e as notas dos jogadores exigia muitos cliques dos usuários e várias atualizações de páginas até chegar a mostrar o que era o objetivo das pesquisas. Foi pensando em otimizar essas consultas que propõe-se esse estudo de caso.

Para obter-se as notas dos jogadores em um determinado jogo, é necessário que seja escolhido, em uma primeira página, o clube. Depois, mostram-se os jogos desse clube em todos os campeonatos disputados, sem nenhum filtro. Ao clicar em um jogo, é mostrada uma terceira página com as notas do jogo selecionado e demais estatísticas. Se o usuário quiser consultar estatísticas de outro clube, tem que voltar até a primeira página para iniciar sua consulta selecionando outro clube.

Na nova proposta, será utilizada apenas uma página para mostrar todas essas informações. Inicialmente, é apresentada uma lista com os clubes a serem escolhidos. Após escolher o clube, é apresentada, logo abaixo, uma segunda lista com os campeonatos desse clube, já utilizando recursos do AJAX para não atualizar toda a página, atualizando apenas a lista de campeonatos. Em seguida, solicita-se a seleção do campeonato. Após selecionar o campeonato, é carregado o bloco abaixo, com recursos AJAX, para mostrar todos os jogos do time selecionado, no campeonato selecionado. A partir disso, pode-se clicar em qualquer jogo que serão mostrados, ao lado, os jogadores que atuaram nesse jogo, suas médias de notas, gols marcados e assistências feitas. A cada clique em um jogo, apenas o bloco com as estatísticas do jogo é atualizado.

Só serão atualizados os jogos se a lista de campeonatos for alterada para selecionar outro campeonato. O mesmo ocorre com a lista de campeonatos, que só sofrerá atualização quando um outro clube for selecionado na lista de clubes.

No Apêndice são mostradas as listagens de todos os códigos fontes utilizados para esse estudo de caso. O framework Prototype foi o escolhido para ajudar no uso do AJAX para o desenvolvimento do caso de uso.

Na Listagem 1, é apresentado o arquivo *estatística.php*, que é o arquivo principal desse estudo de caso. É nele que são criadas as funções JavaScript para atualizar os quatro blocos de informações. A classe `Ajax.Updater`, apresentada na função `buscarConteudo`, é a única classe utilizada, de forma explícita, do framework prototype. É essa classe que instancia o objeto `XMLHttpRequest`, faz as chamadas aos

procedimentos remotos e retorna as informações para a localização correta na página, sem precisar fazer atualização de toda página, apenas do espaço delimitado pela tag DIV especificada na sua chamada.

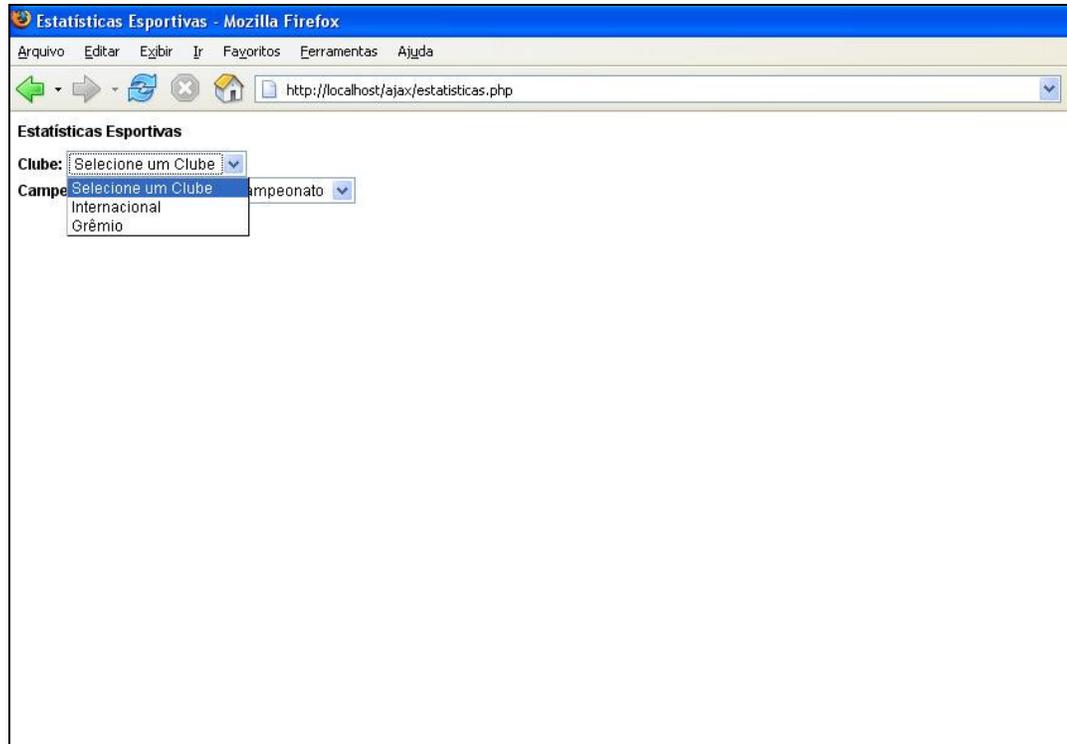


Figura 4.1: Lista com clubes para serem selecionados.

No primeiro bloco, caracterizado pela DIV “clubes” e apresentado na figura 4.1, são carregados, na entrada da página, os clubes que possuem jogos a serem mostrados.

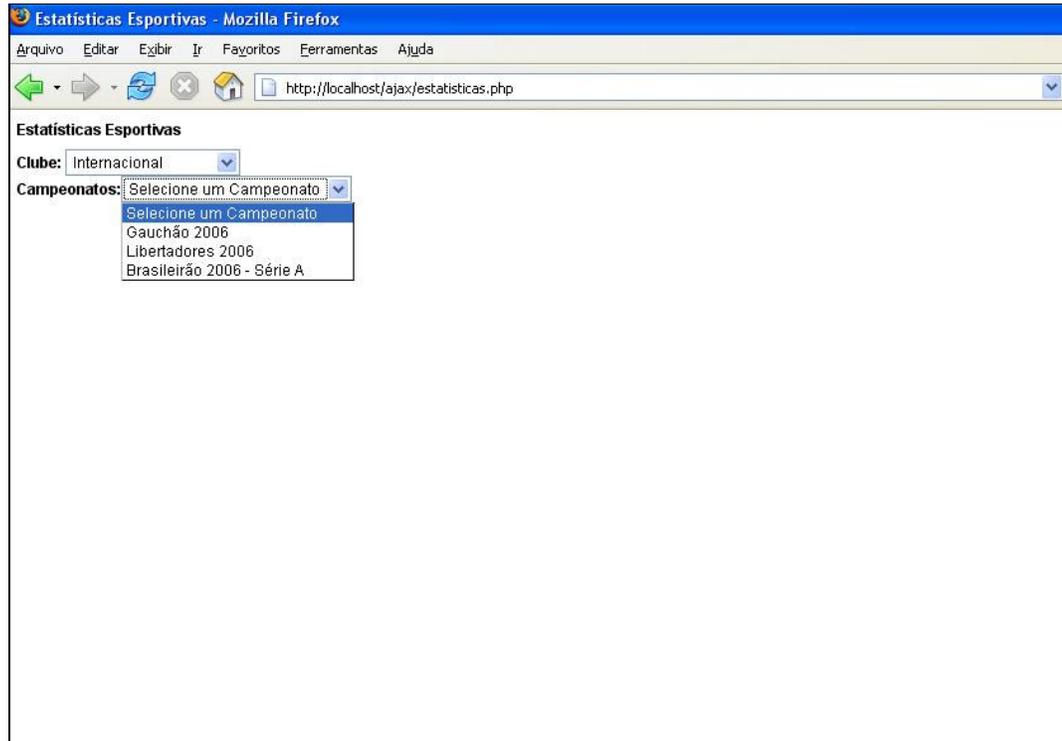
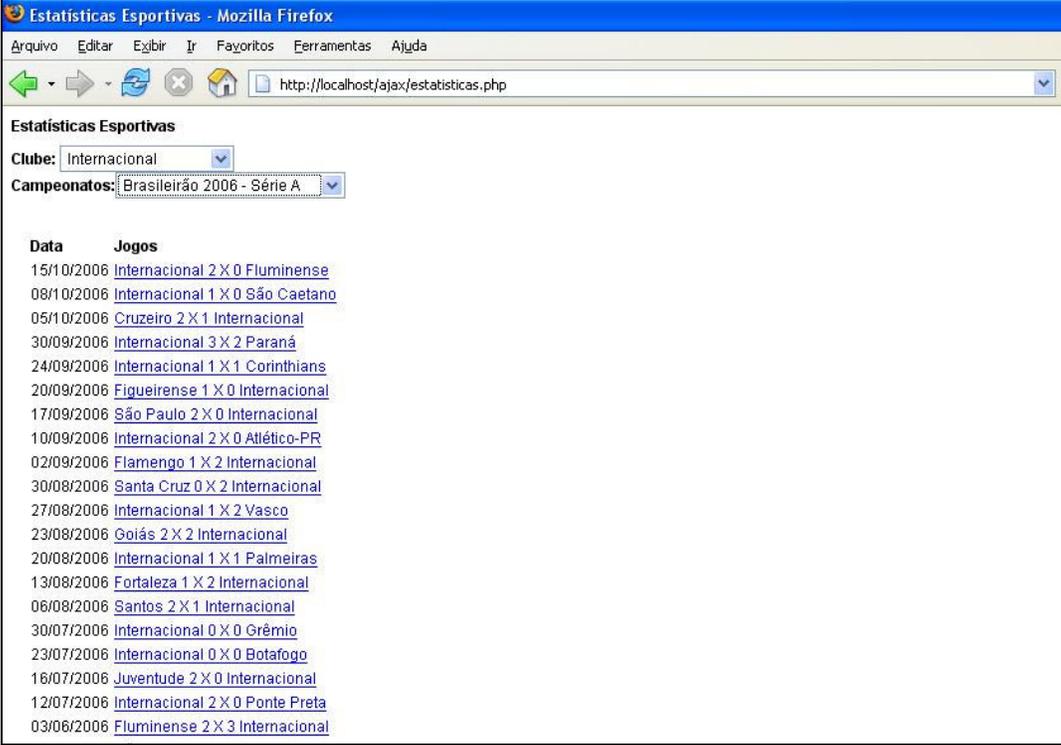


Figura 4.2: Página mostrando os campeonatos que o clube jogou.

No segundo bloco, caracterizado pela DIV “campeonatos” e apresentado na figura 4.2, são mostrados os campeonatos dos quais o clube participou ou está participando. Esse conteúdo é carregado dinamicamente cada vez que um novo clube é selecionado, atualizando apenas a lista de campeonatos, sem atualizar toda página.



**Estatísticas Esportivas**

Clube:

Campeonatos:

Data	Jogos
15/10/2006	<a href="#">Internacional 2 X 0 Fluminense</a>
08/10/2006	<a href="#">Internacional 1 X 0 São Caetano</a>
05/10/2006	<a href="#">Cruzeiro 2 X 1 Internacional</a>
30/09/2006	<a href="#">Internacional 3 X 2 Paraná</a>
24/09/2006	<a href="#">Internacional 1 X 1 Corinthians</a>
20/09/2006	<a href="#">Figueirense 1 X 0 Internacional</a>
17/09/2006	<a href="#">São Paulo 2 X 0 Internacional</a>
10/09/2006	<a href="#">Internacional 2 X 0 Atlético-PR</a>
02/09/2006	<a href="#">Flamengo 1 X 2 Internacional</a>
30/08/2006	<a href="#">Santa Cruz 0 X 2 Internacional</a>
27/08/2006	<a href="#">Internacional 1 X 2 Vasco</a>
23/08/2006	<a href="#">Goiás 2 X 2 Internacional</a>
20/08/2006	<a href="#">Internacional 1 X 1 Palmeiras</a>
13/08/2006	<a href="#">Fortaleza 1 X 2 Internacional</a>
06/08/2006	<a href="#">Santos 2 X 1 Internacional</a>
30/07/2006	<a href="#">Internacional 0 X 0 Grêmio</a>
23/07/2006	<a href="#">Internacional 0 X 0 Botafogo</a>
16/07/2006	<a href="#">Juventude 2 X 0 Internacional</a>
12/07/2006	<a href="#">Internacional 2 X 0 Ponte Preta</a>
03/06/2006	<a href="#">Fluminense 2 X 3 Internacional</a>

Figura 4.3: Página mostrando jogos selecionados para o clube no campeonato.

No terceiro bloco, caracterizado pela DIV “jogos” e apresentado na figura 4.3, são mostrados os jogos do campeonato selecionado. Esses jogos são atualizados cada vez que um novo campeonato é selecionado, sem atualizar toda página, apenas este bloco.

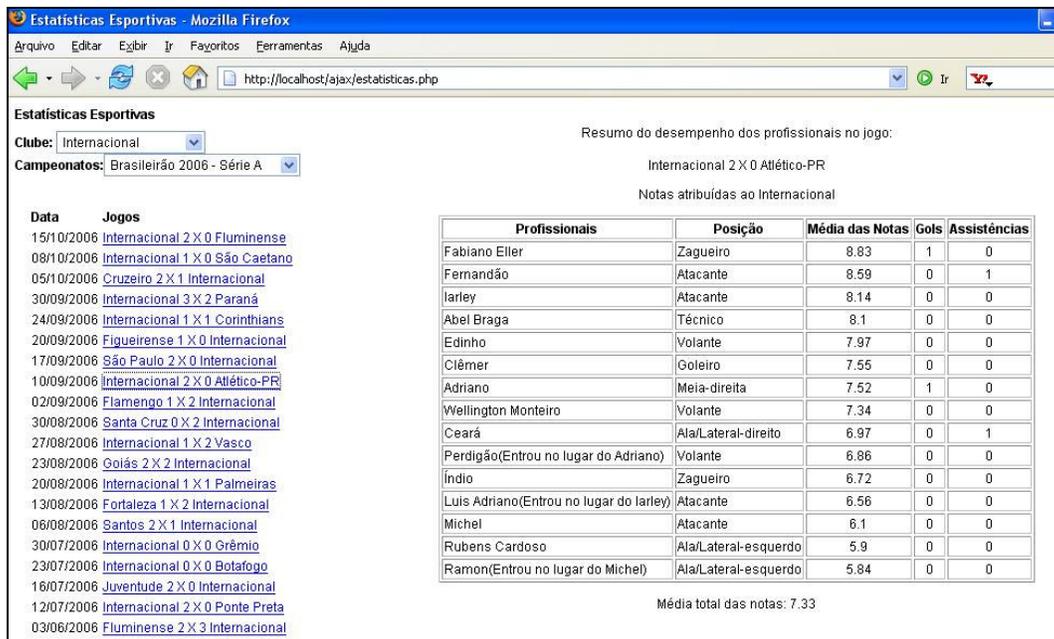


Figura 4.4: Página mostrando as estatísticas do jogo selecionado.

No quarto bloco, caracterizado pela DIV “notas” e apresentado na figura 4.4, são mostradas as estatísticas dos jogadores para um jogo específico, como média de notas, gols e assistências, conforme seleção do usuário feita no terceiro bloco. Como os blocos anteriores, esse bloco é atualizado sem atualizar toda a página.

Demonstram-se, por este estudo de caso, as vantagens de utilizar os recursos de AJAX no desenvolvimento de aplicações Web, tornando as aplicações mais dinâmicas, rápidas e intuitivas. A utilização do framework Prototype facilita bastante o uso dos recursos do AJAX, pois já implementa várias funções importantes para o uso correto das tecnologias envolvidas.

## 5 FUTURO DO AJAX

Apesar do termo AJAX ser novo, muitas aplicações já utilizam seus recursos e princípios. A cada dia surgem mais frameworks, bibliotecas de funções para facilitar a sua utilização e assim disseminar o seu uso pelos desenvolvedores.

Os maiores problemas na utilização do AJAX passam pela complexidade do código utilizado em JavaScript, a dificuldade de depuração do código e a incompatibilidade do código JavaScript entre diferentes navegadores.

Felizmente, várias alternativas estão sendo criadas para amenizar essas dificuldades, como a utilização de frameworks, bibliotecas de funções, e utilização de navegadores que seguem aos padrões do W3C de implementação para JavaScript e de depuradores (debuggers) de código JavaScript e exploradores do DOM integrados ao navegador.

Outro problema na adoção do AJAX, por ser um termo novo, é não possuir muita documentação a seu respeito, mas isso já começa a ser resolvido, pois muitos sites populares já iniciaram a utilizá-lo e assim cada vez mais artigos e documentações aparecem sobre AJAX.

A tendência é que cada vez mais as aplicações tornem-se mais interativas e utilizem os conceitos de AJAX. Os IDEs para desenvolvimento Web preocupam-se em oferecer um ambiente integrado e por isso estão incorporando frameworks e bibliotecas AJAX para facilitar e agilizar o desenvolvimento de aplicações.

Os usuários já não aceitam mais sites que não tenham vida, interatividade. Já é até uma imposição dos usuários que as aplicações sejam mais agradáveis e ágeis, adaptando-se ao modo de vida dos usuários, sempre ávidos por informações de forma rápida.

Com todo esse cenário favorável, é natural dizer que o conjunto de tecnologias que cercam o termo AJAX, cada vez mais se consolida como uma filosofia para implementação de aplicações, até se tornar um padrão para todo desenvolvimento Web que queira produzir resultados com cada vez mais usabilidade, interatividade e agilidade.

## 6 CONCLUSÃO

Essa monografia teve o objetivo de apresentar os recursos do AJAX, suas tecnologias associadas e a maneira como integrá-las para que sejam desenvolvidas aplicações Web mais interativas e eficientes.

A importância desse trabalho está na apresentação de novos conceitos para o desenvolvimento Web, abrindo possibilidades para aplicações, que antes não eram viáveis para sua utilização na Web, passem a ser disponibilizadas. A chamada Web 2.0 chegou para revolucionar a Internet e proporcionar recursos que dão mais vida às aplicações. O AJAX é o principal responsável por essas mudanças de conceitos, como se procurou evidenciar nesse trabalho.

Ao longo da monografia, conceituou-se o que significa o termo AJAX e quais são os ganhos que ele pode oferecer para o desenvolvimento Web que adote seus recursos. Os problemas, que antes eram enfrentados com páginas lentas e muito tráfego de informações desnecessárias, podem ser resolvidos através do uso do AJAX. Ele veio para dar uma solução a questões desse tipo.

As várias tecnologias envolvidas com o AJAX foram caracterizadas e foram apresentadas suas finalidades. Os padrões de desenvolvimento e a maneira como essas tecnologias devem interagir é algo muito importante no sucesso de aplicações que pretendem tirar o máximo proveito dos recursos do AJAX.

Foram apresentados vários sites que utilizam AJAX e os recursos que eles adotaram. A grande maioria desses sites utiliza um ou mais frameworks para AJAX. A importância na adoção de um framework é explanada nesse trabalho, pois a sua utilização facilita muito o desenvolvimento de aplicações AJAX. Dos frameworks citados, os que mais se destacam são o Prototype, Dojo e Mochikit. Esses três são os mais completos frameworks analisados. O Prototype, além de possuir uma infinidade de recursos e de ter uma documentação muito detalhada sobre as suas bibliotecas, é o framework mais utilizado mundialmente. Por isso, optou-se, no estudo de caso, pela utilização do framework Prototype.

A partir do estudo de caso apresentado, é possível verificar que uma aplicação Web, que antes apresentava alguns problemas de navegação, apresentação e recuperação de informações, pode ser transformada para uma aplicação AJAX. Os benefícios de dinamicidade, fácil acesso às informações e tempo de resposta mais rápido, foram incorporados de uma maneira tranquila e sem precisar grandes mudanças nos programas

antes escritos sem esses recursos. O framework Prototype foi utilizado para implementar o estudo de caso, o que facilitou bastante a adaptação do site para AJAX.

Os problemas e futuras necessidades do AJAX foram apresentados. A conclusão é que é muito viável e recomendada a sua utilização em praticamente todos os tipos de aplicações possíveis para serem utilizadas na Web. A cada dia, novos padrões e novas bibliotecas de funções surgem para cada vez mais facilitar a utilização de seus recursos. Com isso, as possibilidades estão abertas, cabe as pessoas perceberem as vantagens que podem ter com os recursos aqui apresentados e, assim, procurarem aproveitar o máximo todos os benefícios que o AJAX pode proporcionar.

## REFERÊNCIAS

- AJAXIAN. 2006. **Ajaxian.com 2006 Survey Results**. Set. 2006. Disponível em: <<http://ajaxian.com/archives/ajaxiancom-2006-survey-results>>. Acesso em: out. 2006.
- AJAXLIB. 2006. Disponível em: <<http://AJAXlib.sourceforge.net>>. Acesso em: out. 2006.
- AMAZON **Diamond Search**. 2006. Disponível em: <[http://www.amazon.com/gp/gsl/search/finder?\\_encoding=UTF8&productGroupID=lose\\_diamonds](http://www.amazon.com/gp/gsl/search/finder?_encoding=UTF8&productGroupID=lose_diamonds)>. Acesso em: out. 2006.
- ASLESON, R.; SCHUTTA, N. **Fundamentos do AJAX**. Rio de Janeiro: Alta Books, 2006.
- ATLAS. 2006. Disponível em: <<http://atlas.asp.net/atlastoolkit>>. Acesso em: out. 2006.
- BACKBASE. 2006. Disponível em: <<http://www.backbase.com>>. Acesso em: out. 2006.
- BAUDISCH, A. **O que é AJAX e como aplicá-lo com PHP, parte 1**. set. 2006. Disponível em: <<http://www.plugmasters.com.br/sys/materias/251/1/O-que--AJAX-e-como-aplic-lo-com-PHP,-parte-1>>. Acesso em: out. 2006.
- DEVELOPER, Yahoo. 2006. Disponível em: <<http://developer.yahoo.com>>. Acesso em: out. 2006.
- DOJO. 2006. Disponível em: <<http://dojotoolkit.org>>. Acesso em: out. 2006.
- DWR. 2006. Disponível em: <<http://getahead.ltd.uk>>. Acesso em: out. 2006.
- ECHO2. 2006. Disponível em: <<http://www.nextapp.com/products/echo2>>. Acesso em: out. 2006.
- FEITOSA, C. **Bibliotecas JavaScripts**. Disponível em: <<http://cirofeitosa.com.br/post/bibliotecas-javascript-2>>. Acesso em: out. 2006.
- FINAL. 2006. Disponível em: <<http://www.final.com.br>>. Acesso em: out. 2006.
- FLICKR. 2006. Disponível em: <<http://www.flickr.com>>. Acesso em: out. 2006.
- GARRET, J. **Ajax: a New Approach to Web Applications**. fev. 2005. Disponível em: <<http://www.adaptivepath.com/publications/essays/archives/000385.php>>. Acesso em: out. 2006.

- GMAIL. 2006. Disponível em: <<http://mail.google.com>>. Acesso em: out. 2006.
- GOOGLE Suggest. 2006. Disponível em: <<http://www.google.com/webhp?hl=en>>. Acesso em: out. 2006.
- GWT. 2006. Disponível em: <<http://code.google.com>>. Acesso em: out. 2006.
- MAPS, Google. 2006. Disponível em: <<http://maps.google.com>>. Acesso em: out. 2006.
- MOCHIKIT. 2006. Disponível em: <<http://mochikit.com>>. Acesso em: out. 2006.
- ORKUT. 2006. Disponível em: <<http://www.orkut.com>>. Acesso em: out. 2006.
- PEREIRA, W. **XML HTTP REQUEST via GET**. 2005. Disponível em: <<http://www.dotpix.com.br/~wendel/projetos/ajax/getXMLHttpRequest/sumGet.html>>. Acesso em: out. 2006.
- PROTOTYPE. 2006. Disponível em: <<http://prototype.conio.net>>. Acesso em: out. 2006.
- REILLY, T. **What Is Web 2.0**. set. 2005. Disponível em: <<http://tinyurl.com/743r5>>. Acesso em: out. 2006.
- SAJAX. 2006. Disponível em: <<http://absinth.modernmethod.com/sajax>>. Acesso em: out. 2006.
- SOARES, W. **AJAX (Asynchronous JavaScript And XML): guia prático para Windows**. São Paulo: Érica, 2006.
- SOUSA, M. **Unindo JavaServer Faces a Ajax: melhorando o processo de desenvolvimento Web**. 2006. Disponível em: <<http://www.devmedia.com.br/visualizaComponente.aspx?comp=3199&site=6>>. Acesso em: out. 2006.
- SYMFONY. 2006. Disponível em: <<http://www.symfony-project.com>>. Acesso em: out. 2006.
- XAJAX. 2006. Disponível em: <<http://www.xAJAXproject.org>>. Acesso em: out. 2006.
- XOAD. 2006. Disponível em: <<http://www.xoad.org>>. Acesso em: out. 2006.
- YAHOO Mail. 2006. Disponível em: <<http://mail.yahoo.com.br>>. Acesso em: out. 2006.

## APÊNDICE: CÓDIGO FONTE DO ESTUDO DE CASO

```
<html>
<title>Estatísticas Esportivas</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<link href="ajax.css" rel="stylesheet" type="text/css">
<script language="javascript" type="text/javascript"
src="prototype.js"></script>
<script>

function buscarConteudo(divdestino, url, pars)
{
    var myAjax = new Ajax.Updater(divdestino, url, {method: 'post',
parameters: pars, evalScripts: true});
}

function atualizarClube() {
    var v_clube      = document.getElementById("clube").value;
    var v_campeonato = document.getElementById("campeonato").value;

    if(v_clube == "") {
        /* Usa uma página vazia para limpar conteúdo de uma área */
        buscarConteudo('campeonatos', 'vazio_camp.htm', '');
        limparResultados();
        return true;
    }

    limparResultados();
    buscarConteudo('campeonatos',
'http://localhost/ajax/buscar_campeonatos.php?p_cod_clube=' +
v_clube, '');
}

function atualizarCampeonato() {
    var v_clube      = document.getElementById("clube").value;
    var v_campeonato = document.getElementById("campeonato").value;

    if(v_clube == "") {
        /* Usa uma página vazia para limpar conteúdo de uma área */
        buscarConteudo('campeonatos', 'vazio_camp.htm', '');
        limparResultados();
        return true;
    }

    if(v_campeonato == "") {
        limparResultados();
        return true;
    }
}
```

```

    }

    buscarConteudo('notas', 'vazio.htm', '');
    buscarConteudo('jogos',
'http://localhost/ajax/mostrar_jogos_campeonato.php?p_cod_campeonato='
+ v_campeonato + '&p_cod_clube=' + v_clube, '');
    return true;
}

function limparResultados() {
    buscarConteudo('jogos', 'vazio.htm', '');
    buscarConteudo('notas', 'vazio.htm', '');
    return true;
}

</script>
</head>

<body onload="javascript:buscarConteudo('campeonatos',
'vazio_camp.htm', '');">
<h1>Estatísticas Esportivas</h1>
<div id="clubes">

    <span style="font-weight:bold;">Clube:</span>
    <select id="clube" onchange="atualizarClube();">
        <option selected value="">Selecione um Clube</option>
        <option value="1">Internacional</option>
        <option value="25">Grêmio</option>
    </select>

</div>

<div id="campeonatos">
</div>

<div id="jogos">
</div>

<div id="notas">
</div>

</body>
</html>

```

Listagem 1 : Descrição do arquivo principal: estatísticas.php.

```

<?php

$conexao = mysql_pconnect("localhost","limeira","limeira") or
die("Erro na conexão: " . mysql_error());
mysql_select_db("monografia");

$v_cod_clube = $_GET['p_cod_clube'];
$v_ano       = date("Y");

$sql = " SELECT distinct c.cod_campeonato, c.nome nome_campeonato ";
$sql .= " FROM jogo j, campeonato c ";
$sql .= " where (j.cod_clube_local      = $v_cod_clube ";
$sql .= "        or j.cod_clube_visitante = $v_cod_clube)";
$sql .= " and   c.cod_campeonato      = j.cod_campeonato ";
$sql .= " and   c.ano_campeonato      = $v_ano ";

```

```

$sql .= " ORDER BY c.cod_campeonato ";

$resultado = mysql_query($sql)
or die ("

```

Listagem 2 : Descrição do arquivo buscar\_campeonatos.php.

```

<?php

$conexao = mysql_pconnect("localhost","limeira","limeira") or
die("Erro na conexão: " . mysql_error());
mysql_select_db("monografia");

$v_cod_campeonato = $_GET['p_cod_campeonato'];
$v_cod_clube_padrao = $_GET['p_cod_clube'];

$sql = " SELECT j.cod_jogo, j.cod_clube_local, j.cod_clube_visitante,
";
$sql .= " j.placar_time_local, j.placar_time_visitante, ";
$sql .= " c1.nome as time_local, c2.nome as time_visitante, ";
$sql .= " DATE_FORMAT(j.data_jogo, '%d/%m/%Y') as data_jogo,
j.situacao_jogo ";
$sql .= " FROM jogo j, clube c1, clube c2, campeonato c ";
$sql .= " where c.cod_campeonato = $v_cod_campeonato ";
$sql .= " and c.cod_campeonato = j.cod_campeonato ";
$sql .= " and (j.cod_clube_local = $v_cod_clube_padrao ";
$sql .= " or j.cod_clube_visitante = $v_cod_clube_padrao)";
$sql .= " and c1.cod_clube = j.cod_clube_local ";
$sql .= " and c2.cod_clube = j.cod_clube_visitante ";
$sql .= " and j.situacao_jogo = 'A' ";
$sql .= " ORDER BY j.data_jogo desc ";

//print($sql . "<br>");

print("<center>");
$resultado = mysql_query($sql)

```

```

or die ("  
Não foi possível realizar a consulta ao banco de dados.
Contacte o Administrador do Sistema." . mysql_error());

$v_prim = 0;
while ($linha=mysql_fetch_array($resultado))
{
    if ($v_prim == 0 )
    {
        $v_prim = 1;
        print("<table border=0><tr>");
        print("<th align=left>Data</th>");
        print("<th align=left>Jogos</th>");
        print("</tr>");
    }
    $v_cod_jogo = $linha["cod_jogo"];
    $v_nome_jogo = $linha['time_local'] . " " .
$linha['placar_time_local'] . " X " .
                $linha['placar_time_visitante'] . " " .
$linha['time_visitante'];

    print("<tr>");
    print("<td align=left>" . $linha['data_jogo'] . "</td>");

    print("<td width=300 align=left><a href=javascript:" .
"buscarConteudo('notas', 'http://localhost/ajax/medias_jogo.php?p_cod_c
ampeonato=" .
        $v_cod_campeonato . "&p_cod_jogo=" . $v_cod_jogo . "', ')" .
">" . $v_nome_jogo . "</a></td>");
    print("</tr>");
}
print ("</table>");
?>

```

Listagem 3 : Descrição do arquivo mostrar\_jogos\_campeonato.php.

```

<?php

$conexao = mysql_pconnect("localhost","limeira","limeira") or
die("Erro na conexão: " . mysql_error());
mysql_select_db("monografia");

$v_cod_campeonato = $_GET['p_cod_campeonato'];
$v_cod_jogo = $_GET['p_cod_jogo'];

$sql = " SELECT cod_jogo, cod_fase, cod_clube_local,
cod_clube_visitante, ";
$sql .= " placar_time_local, placar_time_visitante, cod_campeonato,
cod_rodada, ";
$sql .= " c1.nome as time_local, c2.nome as time_visitante ";
$sql .= " FROM jogo, clube c1, clube c2 ";
$sql .= " where cod_campeonato = $v_cod_campeonato ";
$sql .= " and cod_jogo = $v_cod_jogo ";
$sql .= " and c1.cod_clube = cod_clube_local ";
$sql .= " and c2.cod_clube = cod_clube_visitante ";
$sql .= " ORDER BY cod_campeonato, cod_fase, cod_rodada ";

```

```

$resultado = mysql_query($sql)
or die ("Não foi possível realizar a consulta ao banco de dados.
Contacte o Administrador do Sistema." . mysql_error());

$linha=mysql_fetch_array($resultado);
$v_nome_jogo = $linha['time_local'] . " " .
$linha['placar_time_local'] . " X " .
                $linha['placar_time_visitante'] . " " .
$linha['time_visitante'];

print("<br><center>");
print("Resumo do desempenho dos profissionais no jogo:<br><br>" .
$v_nome_jogo . "<br>");

$sql = " SELECT c.nome, c.cod_clube ";
$sql .= " FROM   jogo j, clube c ";
$sql .= " WHERE  j.cod_jogo = $v_cod_jogo ";
$sql .= " and    (c.cod_clube = j.cod_clube_local ";
$sql .= "        or c.cod_clube = j.cod_clube_visitante) ";
$sql .= " and    c.avaliar_notas = 'S' ";

$resultado_times = mysql_query($sql)
or die ("Não foi possível realizar a consulta ao banco de dados.
Contacte o Administrador do Sistema." . mysql_error());

while ($linha_times=mysql_fetch_array($resultado_times))
{

$v_cod_time_avaliado = $linha_times['cod_clube'];
$v_nome_time_avaliado = $linha_times['nome'];

$sql = " SELECT ej.cod_profissional, po.nome nome_posicao, p.nome as
nome_profissional, ";
$sql .= " ej.observacao observacao, ";
$sql .= " sum(nj.nota) total_notas, count(nj.nota) total_aval,
count(*) total_jogos, ";
$sql .= " sum( nj.nota ) / count( nj.nota ) media ";
$sql .= " FROM escalacao_jogo ej LEFT JOIN nota_jogo nj ";
$sql .= " ON nj.cod_jogo = ej.cod_jogo and nj.cod_profissional =
ej.cod_profissional ";
$sql .= " and nj.nota <> 99 ";
$sql .= " INNER JOIN profissional p ON p.cod_profissional =
ej.cod_profissional ";
$sql .= " INNER JOIN posicao po ON po.cod_posicao = p.cod_posicao ";
$sql .= " INNER JOIN jogo jo ON jo.cod_jogo = ej.cod_jogo ";
$sql .= " WHERE jo.cod_campeonato = " . $v_cod_campeonato;
$sql .= " and   ej.cod_jogo = " . $v_cod_jogo;
$sql .= " and   p.cod_clube = " . $v_cod_time_avaliado;
$sql .= " GROUP BY ej.cod_profissional, po.nome, p.nome, ej.observacao
";
$sql .= " ORDER BY media desc, p.cod_posicao ";

$resultado = mysql_query($sql)
or die ("Não foi possível realizar a consulta ao banco de dados.
Contacte o Administrador do Sistema." . mysql_error());

print("<p align=center>Notas atribuídas ao " . $v_nome_time_avaliado .
"</p>");
print("<table border=1><tr>");
print("<th>Profissionais</th>");

```

```

print("<th>Posição</th>");
print("<th>Média das Notas</th>");
print("<th>Gols</th>");
print("<th>Assistências</th></tr>");
$v_total_jogos = 0;
$v_total_notas = 0;
while ($linha=mysql_fetch_array($resultado))
{
    print("<tr>");
    if ($linha['observacao'] == "")
    {
        print("<td>" . $linha['nome_profissional'] . "</td>");
    }
    else
    {
        print("<td>" . $linha['nome_profissional'] . "(" .
        $linha['observacao'] . ")</td>");
    }
    print("<td>" . $linha['nome_posicao'] . "</td>");

    if ($linha['total_jogos'] > 0)
    {
        $v_media = $linha['total_notas'] / $linha['total_jogos'];
        if ($linha['total_aval'] > 0)
        {
            $v_total_notas = $v_total_notas + $linha['total_notas'];
            $v_total_jogos = $v_total_jogos + $linha['total_jogos'];
        }
    }
    else
    {
        $v_media = 0;
    }

    if ($v_media == 0 && $linha['total_aval'] == 0)
    {
        print("<td align=center>sem nota</td>");
    }
    else
    {
        print("<td align=center>" . round($v_media,2) . "</td>");
    }

    $sql = " SELECT count(*) total_jogos, ";
    $sql .= " sum(ej.gols_marcados) total_gols, sum(ej.nro_assistencias)
total_assistencias";
    $sql .= " FROM escalacao_jogo ej, jogo jo ";
    $sql .= " WHERE jo.cod_campeonato = $v_cod_campeonato";
    $sql .= " and jo.cod_jogo = $v_cod_jogo";
    $sql .= " and ej.cod_profissional = " .
$linha['cod_profissional'];
    $sql .= " and jo.cod_jogo = ej.cod_jogo ";
    $resultado2 = mysql_query($sql)
    or die ("Não foi possível realizar a consulta ao banco de dados.
Contacte o Administrador do Sistema.");
    $linha2=mysql_fetch_array($resultado2);

    print("<td align=center>" . $linha2['total_gols'] . "</td>");
    print("<td align=center>" . $linha2['total_assistencias'] .
"</td></tr>");

```

```

}
print("</table>");
if ($v_total_jogos > 0)
{
    $v_media_total = $v_total_notas / $v_total_jogos;
}
else
{
    $v_media_total = 0;
}
print("<p align=center>Média total das notas: " .
round($v_media_total,2) . "</p>");
}
?>

```

Listagem 4 : Descrição do arquivo medias\_jogo.php.

```

* { font-family: Arial, Helvetica, sans-serif; font-size: 12px; }
#jogos
{
    align: left;
    border: 0px solid white;
    font-size: 12px;
    left: 20px;
    position: absolute;
    text-align: left;
    top: 100px;
    width: 350;
}
#notas
{
    align: left;
    border: 0px solid white;
    font-size: 12px;
    left: 400px;
    position: absolute;
    text-align: left;
    top: 10px;
}

```

Listagem 5 : Descrição do arquivo ajax.css.