

Potenciais de Aplicação da Metodologia AJAX

Bruno Simões

Kleverson Pereira

Marcos Santos

Eduardo Barrere

Associação Educacional Dom Bosco - AEDB

RESUMO

*AJAX (acrônimo em língua inglesa de **Asynchronous Javascript And XML**) é o uso sistemático de Javascript e XML (e derivados) para tornar o navegador mais interativo com o usuário, utilizando-se de solicitações assíncronas de informações. AJAX não é uma tecnologia e sim uma metodologia, são várias tecnologias trabalhando juntas, cada uma fazendo sua parte, oferecendo novas funcionalidades.*

Palavras-Chaves: AJAX. Metodologia. Interatividade. Assíncrona.

1. INTRODUÇÃO

O modelo clássico de aplicação web trabalha assim:

A maioria das ações do usuário na interface dispara uma solicitação HTTP para o servidor web. O servidor processa algo, recuperando dados, realizando cálculos, conversando com vários sistemas legados e então retorna uma página HTML para o cliente.

É um modelo adaptado do uso original da web como um agente de hipertexto, porém o que faz a web boa para hipertexto não necessariamente faz ela boa para aplicações de software.

Esta aproximação possui muito dos sentidos técnicos, mas não faz tudo que um usuário experiente poderia fazer.

Enquanto o servidor está fazendo seu trabalho, o que o usuário estará fazendo? O que é certo, esperando. E a cada etapa em uma tarefa, o usuário aguarda mais uma vez.

Obviamente, se nós estivéssemos projetando a web a partir do zero para aplicações, não faríamos com que os usuários esperassem em vão.

Uma vez que a interface está carregada, por que a interação do usuário deveria parar a cada vez que a aplicação precisasse de algo do servidor?

Na realidade, por que o usuário deveria ver a aplicação ir ao servidor toda vez?

A maior vantagem das aplicações **AJAX** é que elas rodam no próprio navegador web.

Então, para estar hábil a executar aplicações **AJAX**, basta possuir algum dos navegadores modernos, ou seja, lançados após 2001. São eles: Mozilla Firefox, Internet Explorer 5+, Opera, Konqueror e Safari.

2. TECNOLOGIAS INTEGRADAS

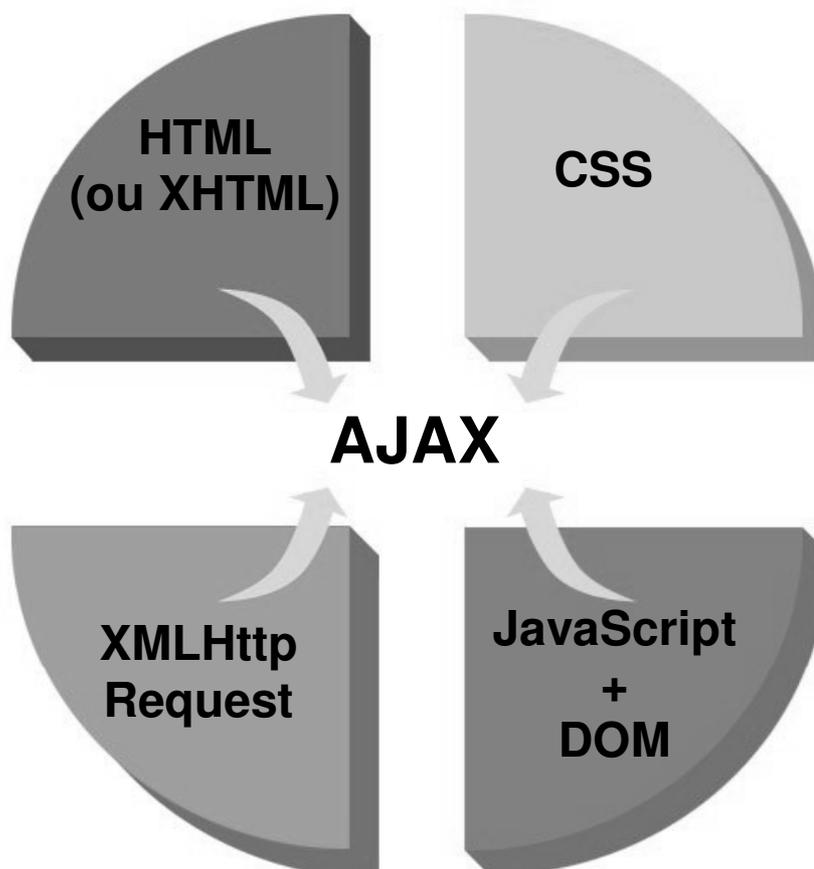


Figura 1. Tecnologias Integradas.

XHTML:

É a sigla em inglês para **EXtensible HyperText Markup Language** que em português resulta em Linguagem Extensível para Marcação de Hipertexto, uma aplicação XML, escrita para substituir o HTML e nada mais é do que um HTML "puro, claro e limpo".

Todas as linguagens de marcação da web são baseadas em SGML, uma metalinguagem complexa, projetada para máquinas com a finalidade de servir de base para criação de outras linguagens.

O XML foi criado a partir do SGML. Com XML você cria suas próprias tags e atributos para escrever seu documento web. Isto significa que é você quem cria sua linguagem de marcação. XHTML foi criado dentro deste conceito e por isso é uma aplicação XML.

As tags e atributos do XHTML foram criados ("inventados") aproveitando-se as nossas conhecidas tags e atributos do HTML 4.01 e suas regras.

Conclui-se que ao usar XHTML, estamos escrevendo um código XML, onde as tags e atributos já estão definidos e isto proporciona todos os benefícios de XML sem as complicações e complexidade do SGML.

CSS:

Cascading Style Sheets, ou simplesmente CSS, é uma linguagem de estilo utilizada para definir a apresentação de documentos escritos em uma linguagem de marcação, como HTML ou XML.

Seu principal benefício é prover a separação entre o formato e o conteúdo de um documento. Ao invés de colocar a formatação dentro do documento, o desenvolvedor cria um link para uma página que contém os estilos, procedendo de forma idêntica para todas as páginas de um portal. Quando quiser alterar a aparência do portal basta, portanto modificar apenas um arquivo.

DOM:

O Document Object Model é uma API para documentos HTML e XML. Providencia uma representação estruturada do documento, permitindo alterar o conteúdo e apresentação visual. Essencialmente, liga páginas web a scripts ou linguagens de programação

XMLHTTPREQUEST:

O coração do AJAX está no objeto XMLHttpRequest, é ele o responsável por tornar possível esta interação entre o servidor e uma página.

Ele foi implementado originalmente no Internet Explorer 5 como um componente ActiveX, porém logo ele foi implementado nos demais navegadores, e para alívio de todos os desenvolvedores web atualmente todos implementam o comportamento do XMLHttpRequest da mesma forma. Veremos agora alguns métodos e propriedades do objeto XMLHttpRequest.

Métodos:

abort() - Interrompe a solicitação atual.

open("method", "url") - Define a que página deverá ser conectada. Pode-se usar o método GET, POST.

send(content) - Envia a solicitação ao servidor, podendo passar algum parâmetro ao servidor caso a solicitação for feita com o método POST.

Propriedades:

Onreadystatechange - Manipula eventos que é acionado a cada mudança do estado, normalmente uma chamada a uma função JavaScript.

ReadyState - Determina a situação atual da solicitação. Varia de 0 a 4, sendo 0 não-iniciada; 1 carregando, 2 carregada, 3 em processamento, 4 concluída.

ResponseXML - Retorna a resposta do servidor, porém no formato XML. Pode ser visto como um objeto DOM.

Status - Especifica o código enviado pelo servidor, 200 para OK e 404 para a famosa "página não pode ser encontrada".

JAVASCRIPT:

JavaScript é uma linguagem de programação criada pela Netscape em 1995, que a princípio se chamava LiveScript, para atender, principalmente, as seguintes necessidades:

- Validação de formulários no lado cliente (programa navegador);
- Interação com a página. Assim, foi feita como uma linguagem de script.

Javascript tem sintaxe semelhante a do Java, mas é totalmente diferente no conceito e no uso. Suas principais características são:

- Oferece tipagem dinâmica - tipos de variáveis não são definidos;
- É interpretada, ao invés de compilada;
- Possui ótimas ferramentas padrão para listagens (como as linguagens de script, de modo geral);
- Oferece bom suporte a expressões regulares (característica também comum a linguagens de script).

3. COMO A METODOLOGIA AJAX FUNCIONA

- Sem o uso de AJAX(modos clássico)

Em uma aplicação web clássica baseada em páginas, o navegador é efetivamente um terminal burro. Ele não sabe nada sobre o que o usuário está realmente realizando em suas ações conseqüentes.

Todas essas informações são retidas no servidor web, tipicamente na sessão do usuário. Sessões de usuários no lado servidor são comuns atualmente. Se você está trabalhando em PHP, Java, .NET ou outra linguagem usada para aplicações web, a sessão no lado servidor faz parte da API padrão, assim como controle de solicitações, respostas, e tipos de conteúdo.

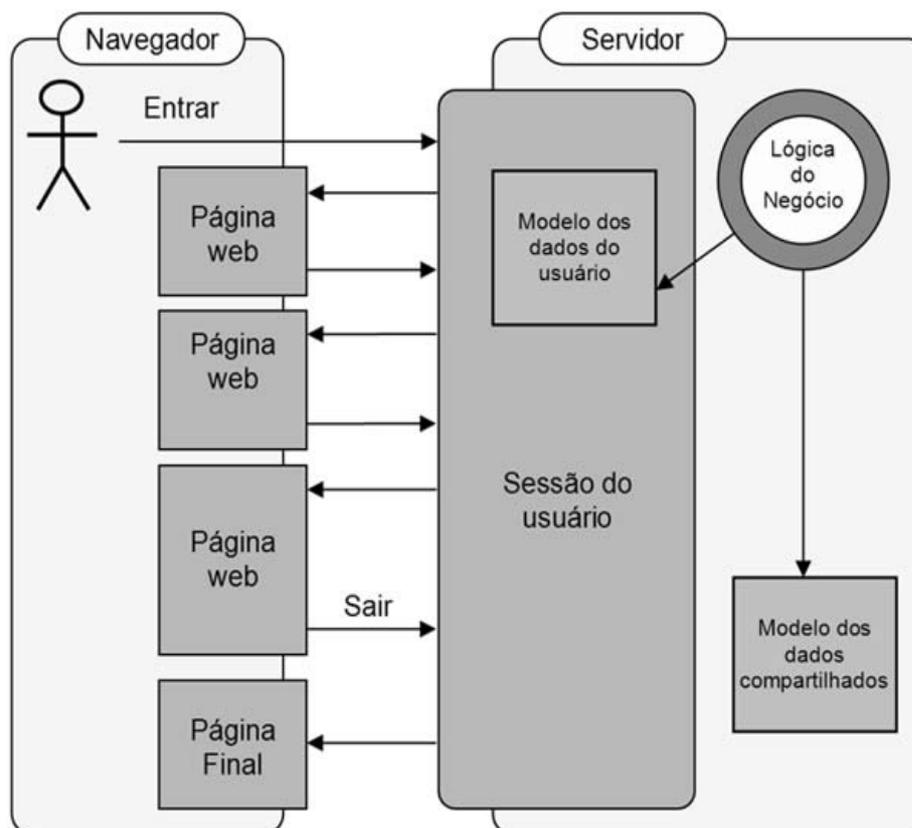


Figura 2. Navegação Método Clássico.

Quando o usuário entra em uma sessão, são criados vários objetos no servidor e ao mesmo tempo é enviado ao navegador uma mistura de apresentação padrão e dados do usuário.

Toda vez que o usuário interage com o site, um outro documento é enviado para o navegador, contendo a mesma mistura de cabeçalhos e dados. O navegador retira o documento anterior e exibe o novo, porque ele não sabe que o outro documento produz um resultado muito semelhante.

Quando o usuário efetua a saída ou fecha o navegador, a aplicação sai e a sessão é destruída. Qualquer informação que o usuário necessite ver na próxima vez que ele entrar terá que ser passada para a camada de persistência de dados em cada visita.

4. Utilizando AJAX

Quando o usuário entra, um documento mais complexo é entregue ao navegador, em maior proporção, que na maioria das vezes é código JavaScript. Este documento permanecerá com o usuário por toda a sessão, ainda que ele resolva provavelmente alterar sua aparência consideravelmente, enquanto o usuário está interagindo com ele.

Ele sabe como responder às informações inseridas pelo usuário e é capaz de decidir se manipula a entrada do usuário ele mesmo ou se passa uma solicitação para o servidor web (o qual tem acesso ao banco de dados do sistema e outros recursos), ou ainda, se faz uma combinação de ambos. Quando é solicitado algo ao servidor web apenas uma parte pertinente a essa solicitação é carregada.

Ele também pode armazenar o estado, porque o documento continua persistindo sobre toda a sessão do usuário e ao sair ele é salvo no servidor diminuindo a quantidade de processos no servidor.

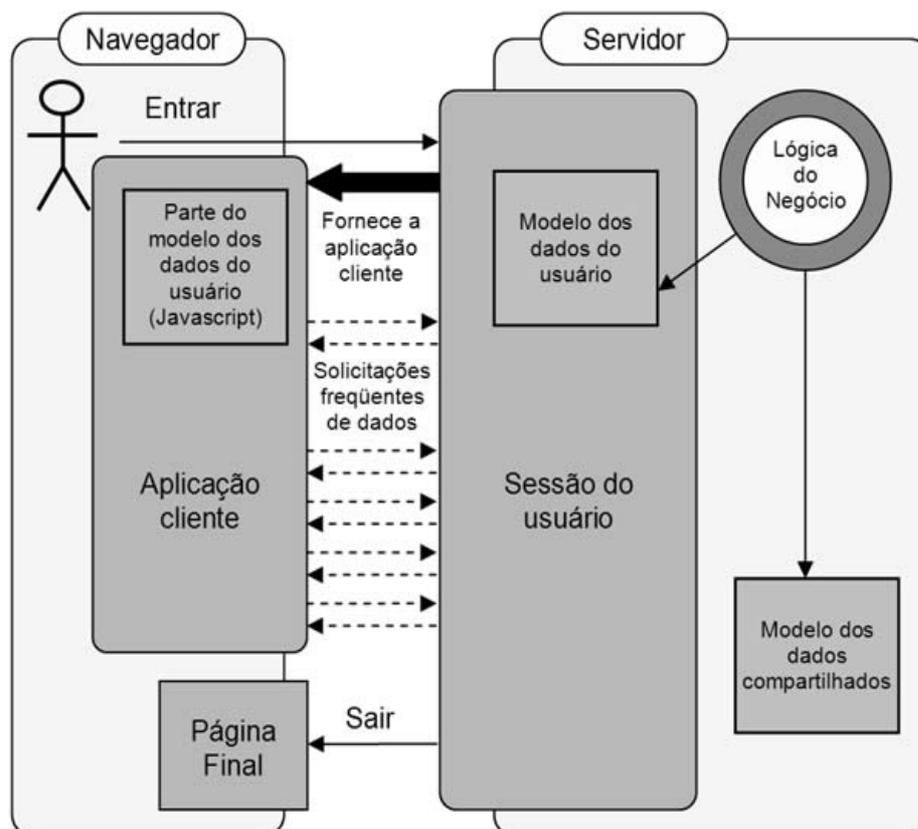


Figura 3. Navegação Metodologia AJAX.

5. Aplicabilidade

BENEFÍCIOS:

O AJAX proporciona uma melhor navegabilidade para o usuário do site ou sistema, deixando a aplicação Web como se fosse aplicativo de máquina local, ou seja, um programa executável.

Com ele, uma página Web fica mais rápida, pois não é necessário o carregamento de todas as imagens novamente e todo o conteúdo da página.

Se o AJAX for usado em páginas com poucos acessos, não se perceberá muita diferença no tráfego que está sendo gerado pela navegação dos usuários, mas utilizando-se em sites com um volume considerável, por exemplo o site do Terra (www.terra.com.br) ou globo.com (www.globo.com), é possível ter uma considerável redução de tráfego de informações, simplesmente pelo motivo de não retornar a página toda em HTML.

Ele também reduz drasticamente o tempo de transferência e apresentação das páginas, pois como ele traz apenas os dados e não toda a página a cada link acessado, a quantidade de dados necessários para serem apresentados são bem menores.

Pelo AJAX trabalhar de forma assíncrona, os benefícios estão basicamente sobre esta plataforma, pois ela traz uma interatividade muito grande ao site ou sistema desenvolvido, ou seja, qualquer ação do usuário pode ser refletida no navegador de forma quase instantânea e, assim, agilizando os recursos existentes na página.

Os conceitos mais sofisticados de interface com o usuário, assim como “arrastar e soltar”, se tornam praticáveis, trazendo as experiências dessas interfaces em pé de igualdade com os controles de aplicações desktop.

DESVANTAGENS:

Um dos problemas no uso de AJAX que pode ser facilmente verificado é a questão de a navegação do usuário não ficar no histórico do navegador, com isso, se o mesmo desejar usar o botão voltar do navegador, poderá não ter o resultado esperado.

Outra desvantagem encontrada é de não ser modificada a barra de endereço, desta forma, fica mais complicada a tarefa de colocar um botão do tipo, “Recomende esta notícia”, ou ainda, salvar nos Favoritos um conteúdo do site visitado.

É importante não se exceder com o AJAX. Lembre-se que o código JavaScript está sendo executado no navegador de seu cliente e ter milhares de linhas de JavaScript pode tornar muito lenta as experiências de seus usuários. Scripts mal-codificados podem sair rapidamente de controle, principalmente conforme o volume aumentar.

Existe ainda a questão de o usuário não estar preparado ou familiarizado com AJAX, ou seja, estar esperando por uma reação do navegador enquanto acontece outra. No caso de uma mudança de página, o usuário pode estar esperando que toda a página faça o refresh e, com o AJAX, somente o conteúdo necessário altera. Desta forma, pode acontecer de o usuário não perceber que o conteúdo já alterou e ficar aguardando muito para continuar sua navegação. Neste caso, é necessário fazer alguma ação para que o usuário perceba que algo mudou, como por exemplo, no caso do Gmail do Google, onde para cada mudança de página, ele tem uma div que apresenta a mensagem

“Carregando...”. Isto faz com que o usuário saiba quando a página está alterada em seu navegador.

6. Exemplo prático de aplicação da metodologia AJAX

Para finalizar este artigo, propomos uma breve descrição de uma aplicação pratica na utilização da metodologia AJAX, neste caso temos uma combo-box com a lista de estados e a outra combo-box que será carregada com a lista das cidades em função do estado escolhido.

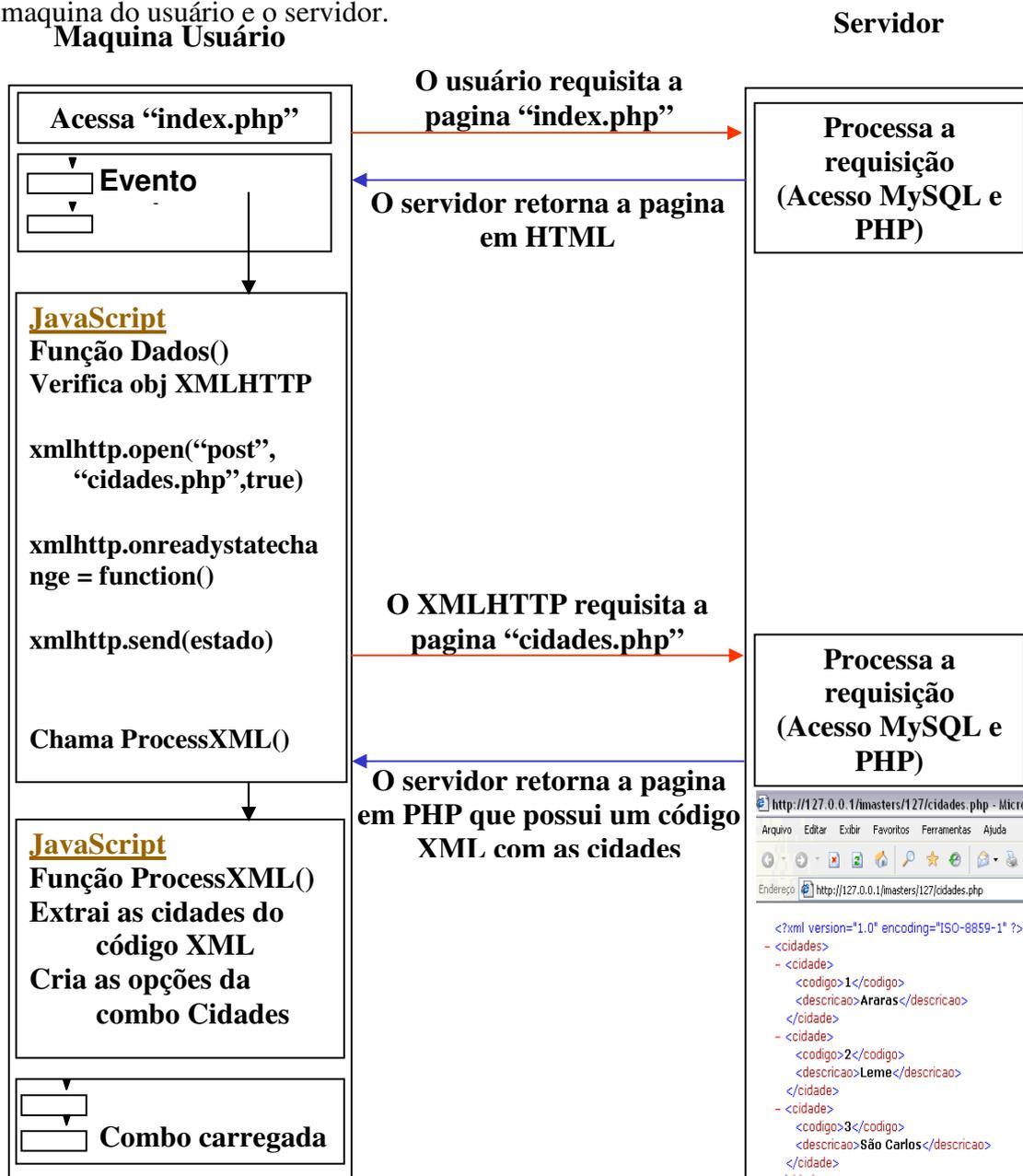
Nosso exemplo possui 3 paginas:

Conecta.php - Responsável pela conexão ao banco de dados. Também seleciona base de dados a ser utilizado.

Index.php - Responsável pela visualização da pagina e, é onde se encontra a chamada para o AJAX.

Cidades.php - Monta um código XML para listar todas as cidades do estado selecionado.

Abaixo uma representação grafica sobre as interações que ocorrem entre a maquina do usuário e o servidor.



7. Conclusão

Podemos notar os inumeros ganhos a aplicação web pode ter utilizando o AJAX mas seu uso deve ser moderado e ser aplicado somente onde for necessário para que o resultado nao saia ao contrário do esperado.

Outra observação que alguns fundamentos da web são constantemente esquecidos pelos utilizadores do AJAX, como a alteração do endereço na barra do navegador, o bookmark, o botão voltar e avançar.

Esta metodologia veio para possibilitar uma aplicação web ser tranformada em aplicação desktop.

8. Referencias

Terra. Disponível em:

< <http://tecnologia.terra.com.br/interna/0,,OI696304-EI4802,00.html> >

Acesso em: 10 abr. 2007.

Elcio Ferreira. Disponível em:

< <http://www.tableless.com.br/artigos/ajaxdemo/> >

Acesso em: 15 abr. 2007.

Maujor. Disponível em:

< <http://www.maujor.com/tutorial/xhtml.php> >

Acesso em: 15 abr. 2007.

Mozilla. Disponível em:

< <http://developer.mozilla.org/pt/docs/DOM> >

Acesso em: 15 abr. 2007.