



**FDV – Faculdade de Viçosa**  
**Departamento de Ciência de Tecnologia**  
**Sistemas de Informação**

# **WEB STANDARDS**

## **Desenvolvendo Sistemas Arrasadores usando AJAX**

**Professor:** Sérgio dos Santos

**FDV:** 7º Período [ Sistemas de Informação ] FORMANDO dez/2007

**E-mail:** sergio.serginhosant@gmail.com

**Cel:** 31 8411 2320

**Colunista no iMasters desde dez/2006**

<http://www.imasters.com.br/>

**Delicious:** [del.icio.us/serginhosant/webstandards](http://del.icio.us/serginhosant/webstandards)

**Blog:** [www.gigro.com/blog](http://www.gigro.com/blog)

Viçosa – Março - 2007



# ÍNDICE

<b><u>1. INTRODUÇÃO.....</u></b>	<b><u>5</u></b>
<b><u>2. WEB STANDARDS.....</u></b>	<b><u>6</u></b>
<b><u>3. TABLESS.....</u></b>	<b><u>6</u></b>
<b><u>3.1. O QUE É TABLESS E COMO FUNCIONA ESSA METODOLOGIA?.....</u></b>	<b><u>7</u></b>
<b><u>3.2. DESENVOLVIMENTO MAIS ÁGIL.....</u></b>	<b><u>7</u></b>
<b><u>3.3. TECNOLOGIA ACESSÍVEL.....</u></b>	<b><u>8</u></b>
<b><u>3.4. FACILIDADE PARA CRIAR VERSÕES DE LAYOUT.....</u></b>	<b><u>8</u></b>
<b><u>4. SEMÂNTICA WEB.....</u></b>	<b><u>8</u></b>
<b><u>4.1. ACESSIBILIDADE.....</u></b>	<b><u>10</u></b>
<b><u>5. AJAX.....</u></b>	<b><u>10</u></b>
<b><u>5.1. O SERVIDOR FORNECE DADOS, E NÃO CONTEÚDO.....</u></b>	<b><u>11</u></b>
<b><u>5.2. VANTAGENS.....</u></b>	<b><u>12</u></b>
i) MAIOR INTERATIVIDADE NAS APLICAÇÕES.....	12
ii) REDUÇÃO DO CONSUMO DE BANDA.....	12
iii) REDUÇÃO DE PROCESSAMENTO NO SERVIDOR.....	12
iv) NÃO É PROPRIETÁRIO.....	13
v) PORTABILIDADE.....	13
<b><u>5.3. DESVANTAGENS.....</u></b>	<b><u>13</u></b>
vi) CAPACIDADES LIMITADAS.....	13
vii) PERFORMANCE DO CLIENTE.....	13
viii) COMPORTAMENTO DOS BOTÕES BACK E FORWARD.....	14
ix) BOOKMARKING (UNIQUE URL).....	14
x) LATÊNCIA DA REDE.....	14
xi) REQUER CONECTIVIDADE PERMANENTE.....	15
<b><u>5.4. DOM.....</u></b>	<b><u>15</u></b>
<b><u>6. BIBLIOTECAS.....</u></b>	<b><u>16</u></b>

<b>6.1. INICIANDO COM A JQUERY.....</b>	<b>16</b>
<b>6.2. A SINTAXE DA JQUERY.....</b>	<b>16</b>
<b>7. CONCLUSÃO.....</b>	<b>17</b>
<b>8. REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>18</b>

## 1. Introdução

Segundo REIS (2005), comparando as aplicações Desktop com as aplicações Web, constata-se que as Desktop dão ao utilizador uma maior riqueza de utilização e melhores tempos de resposta que as aplicações Web tradicionais. A mesma simplicidade que fez proliferar as páginas Web criou um abismo entre a experiência de utilização que se consegue oferecer numa aplicação Desktop e uma página Web Standard. No entanto essa distância está cada vez mais curta com a chegada de novas tecnologias e novas técnicas, entre elas o AJAX.

Como construir sistemas arrasadores usando Web Standards e AJAX? Este é o objetivo deste curso. Construir uma aplicação AJAX muito simples, que te faça entender os conceitos básicos por trás disso tudo. A primeira coisa importante a ser dita aqui é a respeito de acessibilidade. É possível, e bastante desejável, construir aplicações AJAX, simples, completamente funcional em navegadores não compatíveis. Para isso, trabalhamos o AJAX como última camada, a ser aplicada ao final de todo o restante. Nessa abordagem, seu código é construído de acordo com as seguintes camadas:

- Conteúdo HTML;
- Programação Server-Side (fazer funcionar);
- CSS (esta etapa e a anterior podem mudar de ordem); e
- Javascript e AJAX.

Fazendo assim, ao terminar a camada 2, você tem uma aplicação completamente funcional, em qualquer navegador, até no Lynx. A camada 3 melhora (muito) a experiência para quem tem navegadores mais atuais.

AJAX é uma camada, acima do CSS, que melhora ainda mais a experiência de quem tem navegadores capazes de vê-la. Isso tem vantagens muito semelhantes ao uso de CSS para construção de layouts tableless, você separa seu conteúdo do comportamento, e pode alterar um sem precisar mexer no outro. E, é claro, você pode escolher não trabalhar assim, e se tiver uma aplicação realmente fantástica, como o Google Maps, as pessoas vão agradecê-lo. Depende do que você vai desenvolver, e como isso vai ser usado.

## 2. Web Standards

Segundo a Wikipedia (2007), Web Standards pode ser traduzido como normas para Web, e têm por objetivo a criação de uma Web universal e define-se: *Web Standards é um conjunto de normas, diretrizes, recomendações, notas, artigos, tutoriais e afins de caráter técnico, produzidos pelo W3C e destinados a orientar fabricantes, desenvolvedores e projetistas para o uso de práticas que possibilitem a criação de uma Web acessível a todos, ajudados por diversos recursos que fazem uma Web mais agradável de usar.*

No nosso dia-a-dia percebemos a importância dos padrões. Já imaginou se ao comprar um eletrodoméstico você tivesse que adaptar a tomada da sua casa? Na Web, isso também vale, ou seja, também é necessário haver padrões. O W3C criou linguagens básicas de publicação de conteúdo para Web. É essas linguagens que são chamadas de Web Standards (Padrões Web) HTML, XHTML, CSS, SVG, XML, XSLT, entre vários outros. [1]

Deve-se perceber as incríveis vantagens que o desenvolvimento com os Padrões oferece, não só para a execução do trabalho, mas para a estruturação da Web em si e contribuindo para a Web do futuro, onde ninguém terá que garimpar em buscadores para conseguir a informação que precisa, mas a informação estará conosco onde estivermos, teremos acesso a ela sem barreiras, na hora que quisermos, onde quisermos, e usando o dispositivo que for. [1]

## 3. Tableless

Tableless é uma forma de desenvolvimento de sites que não utiliza tabelas para disposição de conteúdo na página, pois defende que os códigos HTML deveriam ser usados para o propósito que foram criados, sendo que tabelas foram criadas para exibir dados tabulares. O termo ganhou destaque no Brasil sendo pouco utilizado no mundo. [3]

Dentre as possíveis vantagens da metodologia, estão a diminuição do peso da página (na realidade, o conteúdo da página é mostrado mais rápido, o que não significa que o peso da página seja menor - isso, é claro, sempre dependendo de como o HTML é feito) e melhora da acessibilidade - podem ser vistos em dispositivos móveis sem necessidade de versão especial e por

deficientes visuais, embora não são muitos os sites tableless que carregam atualmente esta segunda característica. [3]

### **3.1. O que é tableless e como funciona essa metodologia?**

Primeiramente, precisamos entender que a Web foi criada para que pudéssemos trocar informações, e que estas informações pudessem ser acessadas de qualquer dispositivo. [1] Com isso, ocorreu uma briga ferrenha entre os principais browsers: Internet Explorer e Netscape para ganharem usuários quando os padrões ainda eram rascunhos, conseqüentemente cada fabricante de browser começou a criar seus próprios padrões. Dessa forma, um site desenvolvido para um browser não era corretamente visualizado pelo outro, obrigando os desenvolvedores escolherem e estudarem as "particularidades" do navegador mais popular, no caso o Internet Explorer. Esta falta de padrão, que tornava inviável fazer um site acessível em múltiplos browsers e sistemas, gerou movimentos como WaSP (Web Standards Project) que contribuíram bastante para mudar essa situação. Ao mesmo tempo induziu a grande maioria dos profissionais de desenvolvimento Web à trabalharem de forma errada durante anos e anos e o que é pior, achando que estavam corretos. [1]

O termo Tableless é uma conotação mais publicitária para se referir a sites que seguem os Padrões. Nesse metodologia se usa o XHTML para as informações e o CSS para a formatação e as tabelas não são mais usadas para formatar as páginas. [1]

### **3.2. Desenvolvimento mais ágil**

O retrabalho de design para que o HTML fique com aparência profissional é praticamente reduzido a zero com o desenvolvimento Web standards. [1]

### **3.3. Tecnologia acessível**

Um designer treinado em Web Standards não depende de um programador para saber o que pode ou não ser implementado em um projeto de web; e vice-versa. [1]

### **3.4. Facilidade para criar versões de layout**

Uma vez que conteúdo, programação e design estão em camadas separadas, é incomparavelmente mais rápido desenvolver versões de layout para a mesma tela. Com a separação entre conteúdo e layout promovida pelos Web Standards, designers e programadores podem trabalhar simultaneamente e de maneira independente. [1]

Perfeita integração com .NET, Java, PHP, ASP etc; Web Standards e Tableless têm total integração com as principais tecnologias de mercado. Pequenas mudanças precisam ser feitas em uma ou outra plataforma. Lembrando sempre, que cada caso é um caso. Por exemplo: se você tem um grande portal, provavelmente a maior vantagem que você vai obter, será a economia de banda. [1]

Com a metodologia Web Standards, evita-se que apenas um membro da equipe tenha domínio exclusivo sobre o desenvolvimento, assegurando que qualquer desenvolvedor poderá dar continuidade ou realizar futuras manutenções. Optar pelos Web Standards significa valer-se da estrutura semântica simples e coerente do HTML, fator primordial para ter um melhor posicionamento nas ferramentas de busca. O código HTML se torna muito mais compacto ao se separar conteúdo, design e programação, conforme os Web Standards. Além disso, a tecnologia Tableless permite que o navegador interprete as informações de layout (em um arquivo CSS) de 30% a 70% mais rapidamente. [1]

## **4. Semântica Web**

Semântica é uma palavra de origem grega (semantiké) que traduzida seria "a arte da significação". É o estudo do significado das palavras e de sua relação de significação nos signos lingüísticos. Signo é todo objeto ou símbolo

que tem um significado por si mesmo, como a cruz que representa o cristianismo, a cor vermelha como um sinal de pare, uma palavra qualquer como "cadeira" que nos remete a um objeto no mundo real. Tudo isto é signo e as "tags" são os signos que serão objetos dos nossos estudos direcionados a Web. Pra nós desenvolvedores, o tema semântica é o estudo desses signos chamados de tags, sua relação uns com os outros e sua aplicabilidade. [2]

Escrever algo semanticamente correto, nada mais é do que utilizar-se desses símbolos (tags) considerando o significado real pelo qual foram criados. Por exemplo, se você no conteúdo do seu HTML for falar sobre a ONU, você não precisa ficar escrevendo Organização das Nações Unidas toda vez que for fazer referências sobre a organização. Até mesmo porque provavelmente o seu público e todo mundo sabe o que ONU significa. Por estilo, talvez apenas na primeira referência que você fizer você pode escrever o nome por extenso, mas depois, você pode apenas escrever "ONU" e no seu código XHTML, você escreve da seguinte maneira: [2]

```
<acronym title="Organização das Nações Unidas">ONU</acronym>
```

O que isto significa? Os sistemas de indexação como os mecanismos de busca do Google, vão armazenar as duas palavras do seu documento para uma eventual busca. Se você não utilizar a tag `<acronym>` e somente usar a palavra ONU no seu documento, quando alguém fizer uma busca no Google por extenso da frase "Organização das Nações Unidas", seu documento não será referenciado e nem encontrado pelos buscadores por não possuir nenhuma referência a esta frase, mesmo que todo mundo saiba que "ONU" signifique "Organização das Nações Unidas". Se a busca tivesse sido feita por ONU, a historia seria outra, mas como não foi, seu documento não foi encontrado. Você precisa dizer para os mecanismos de busca que ONU é um acrônimo e dizer o seu real significado.

Escrever algo semântico então, é utilizar uma tag no contexto certo para a qual ela foi criada. Veja alguns erros comuns e algumas tags que poderiam ser utilizadas mas que muitas vezes são esquecidas:

- A tag `<table>` foi criada para exibir dados tabulares e não pra estruturar sites;
- Usar a tag `<br />` para simular parágrafos, onde a tag `<p>` poderia ser usada;

- Usar qualquer outra tag que não seja os headings (títulos) como h1, h2, h3, etc, para declarar títulos;
- Quando usar uma citação em um texto coloque entre a tag <cite>;
- Tudo que for uma lista, como menus, itens, tópicos etc, utilize as tags <ul> (lista sem ordem) ou <ol> (lista ordenada) juntamente com a tag <li> (lista); e
- Quando fizer um formulário de inserção de dados utilize a tag <label> (rótulo) para declarar o nome do campo.

#### **4.1. Acessibilidade**

O uso de Web Standards facilita muito a aplicação de normas de acessibilidade. Isso garante o acesso ao website, não importando a que combinação de navegador e plataforma o usuário possua. Cabe agora aos desenvolvedores atualizarem seus conhecimentos e revolucionar a Internet, pois internet mudou a vida das pessoas; agora, são as pessoas que vão mudar a internet.

#### **5. Ajax**

AJAX (acrônimo em língua inglesa de Asynchronous Javascript And XML) é o uso sistemático de Javascript e XML para tornar o navegador mais interativo com o usuário, utilizando-se de solicitações assíncronas de informações. AJAX não é somente um novo modelo, é também uma iniciativa na construção de aplicações web mais dinâmicas e criativas. AJAX não é uma tecnologia, são realmente várias tecnologias trabalhando juntas, cada uma fazendo sua parte, oferecendo novas funcionalidades. AJAX incorpora em seu modelo: **[3]**

- Apresentação baseada em padrões, usando XHTML e CSS;
- Exposição e interação dinâmica usando o DOM;
- Intercâmbio e manipulação de dados usando XML e XSLT;
- Recuperação assíncrona de dados usando o objeto XMLHttpRequest; e
- JavaScript unindo todas elas em conjunto.

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 — 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. [3]

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. [3]

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. [3]

### **5.1. O servidor fornece dados, e não conteúdo**

Uma aplicação web clássica oferece a mesma mistura de alegorias, conteúdos e dados em todos os passos. Quando nosso usuário adiciona um item na cesta de compras, tudo que precisamos realmente é responder com o valor atualizado da cesta ou informar se alguma coisa deu errado. Como ilustrado na figura logo abaixo, isto será uma parte muito pequena de todo o documento vivo.

## **5.2. Vantagens**

Segundo REIS (2005), as aplicações que sigam o modelo AJAX dão um grande passo quando se tenta comparar com a complexidade e riqueza das aplicações Desktop tradicionais.

### **i) Maior Interatividade nas aplicações**

É uma das principais vantagens e uma das principais razões para utilizar o AJAX em aplicações Web. Como maior parte da aplicação é executada no lado do cliente isto permite que uma aplicação AJAX se comporte como uma aplicação Desktop e não seja tão limitada pela rede em termos de comunicação com servidor, possibilitando uma interação de ações do usuário de modo contínuo. As aplicações AJAX são mais reativas e permitam uma maior riqueza em termos gráficos (efeitos visuais que indicam mudanças de estado, um envio de dados, ou poder arrastar e colar elementos, tal como possível em uma aplicação Desktop).

### **ii) Redução do consumo de banda**

Na maioria das aplicações Web que utilizam o modelo AJAX, devido á sua característica de atualizações granulares, consegue-se reduzir a quantidade de informação trocada com o servidor (não é necessário reenviar totalmente a página da interface, são enviados para o Web Browser do cliente apenas dados referentes a aplicação ).

### **iii) Redução de processamento no servidor**

Em casos de aplicações que envolvam realizar muitos cálculos no lado do servidor; através de aplicações AJAX é possível em alguns casos trazer parte desses processamentos para o cliente (por exemplo, a ordenação de tabelas de produtos), permitindo desta forma reduzir o processamento necessário por parte do servidor. O que é de interface se processa na interface, o que são dados de negócio se processa no servidor.

#### **iv) Não é proprietário**

AJAX não é um nome de nenhuma marca ou produto, apenas uma designação de um método de como projetar aplicações Web, utilizando um conjunto de tecnologias Web padrões existentes.

#### **v) Portabilidade**

Como o AJAX é constituído por tecnologias que são utilizadas pela maioria dos Web Browsers existentes no mercado, torna-se uma técnica que não se restringe a um Web Browser, nem a uma plataforma. Além disso, não requer a instalação de qualquer plugin no Web Browser ou software no cliente.

### **5.3. Desvantagens**

O AJAX possui alguns pontos que devem ser observados, grande parte devido ao comportamento dos Web Browsers.

#### **vi) Capacidades limitadas**

Atualmente algumas aplicações AJAX conseguem fazer coisas inimagináveis, para algum tempo atrás na Web, mas existem muitas restrições nesta plataforma. Por consequência, o AJAX ao se basear em tecnologias existentes, herda as suas limitações. Alguns exemplos de limitações são: capacidade multimídia (streaming de video), armazenamento local no cliente, gráficos em tempo real, interação com hardware (impressoras, webcams). Algumas destas limitações têm sido solucionadas nos Web Browsers mais recentes ou através de plugins específicos (como por exemplo, o Flash), mas existem algumas que serão muito difíceis de superar, devido as características não operacional da Web.

#### **vii) Performance do cliente**

Numa aplicação AJAX transfere-se muito do processamento do servidor para o cliente. Essa mudança tem custos porque estaremos delegando ao

cliente a responsabilidade para realizar determinadas operações, para as quais não foi inicialmente desenvolvido. Com isso pode-se sobrecarregar o cliente, caso não se tomem às devidas precauções, durante a fase de desenvolvimento deste tipo de aplicações Web.

### **viii) Comportamento dos botões Back e Forward**

Como as atualizações das páginas passam a ser dinâmicas, por exemplo, os botões back e forward dos Web Browsers, podem ter comportamentos que não são os pretendidos. Para resolver este problema começam a aparecer algumas soluções, onde a maioria passa pela criação ou utilização de Frames invisíveis que guardam os diversos estados da aplicações e preenchem o histórico utilizado pelos botões back e forward.

### **ix) Bookmarking (Unique URL)**

Outro problema associado ao anterior é o fato de se tornar difícil, a tarefa do usuário guardar um link para um determinado estado da aplicação. Para este problema também existem algumas soluções, entre elas a utilização do identificador de fragmento do URL(anchor), que permite aos usuários voltarem a um determinado estado da aplicação.

### **x) Latência da rede**

Outra preocupação das aplicações AJAX é a latência de rede. Deverá existir a preocupação de quando for efetuado um pedido ao servidor, ser dado algum feedback para que o usuário não pense que a aplicação teve uma falha, ou travou, caso o pedido demore entre a solicitação para enviar, até ser mostrada a resposta. Este problema tem tendência a diminuir com o aumento da banda larga, mas nunca deixará de existir devido á natureza da Internet.

## xi) Requer conectividade permanente

Essas aplicações operam apenas baseadas na Web, como a parte da aplicação de processamento de dados de negócio fica em um servidor, isso torna necessário que a aplicação esteja sempre conectada.

### 5.4. DOM

O DOM manipula HTML, desde o conteúdo de blocos, width, height, até linhas de uma tabela. Os elementos são conhecidos como Nodes, e cada tipo tem seus próprios métodos e atributos particulares.

O conteúdo de um input do tipo text com id "cliente\_nome" pode ser acessado da seguinte maneira:

```
<script type="text/javascript">
    conteudo = document.getElementById("cliente_nome").value
</script>
```

Ou então usando a biblioteca jquery:

```
<script type="text/javascript">
    conteudo = $("#cliente_nome").val();
</script>
```

Para trocar o valor deste input:

```
<script type="text/javascript">
    conteudo = document.getElementById("cliente_nome").value = "Novo Valor";
</script>
```

Ou então usando a biblioteca jquery:

```
<script type="text/javascript">
    conteudo = $("#cliente_nome").val("Novo Valor");
</script>
```

## 6. Bibliotecas

Existem muitas bibliotecas para os desenvolvedores usarem em suas aplicações. Porém, depois de quebrar muito a cabeça, eu recomendo a jquery.

Qual é a vantagem de utilizar ela? Creio que podemos fazer uma lista:

- Compatibilidade com qualquer navegador;
- Seu peso é bem leve ficando em torno de 20kb (compressed);
- Otimização de código, você evita de programar linhas e linhas de código;
- Você consegue acessar a qualquer elemento ou combinações de elementos;
- Possui diversos efeitos visuais para deixar seu site mais atraente.

### 6.1. Iniciando com a JQuery

Faça o download da versão mais recente da JQuery [http://docs.jquery.com/Downloading\\_jQuery](http://docs.jquery.com/Downloading_jQuery). Feito o download do arquivo, basta adicionar o arquivo na sua página.

```
<script type="text/javascript" language="javascript" src="js/jquery.js"></script>
```

### 6.2. A sintaxe da JQuery

A JQuery tem uma sintaxe bem simples e intuitiva.

Suponha que você queira selecionar todas as tag's div, seu código seria o seguinte: (`"div"`). Quanto no método tradicional, você teria de usar: `document.getElementsByTagName("div")`. Esse exemplo bem básico já demonstra a simplicidade da JQuery e o tempo que você pode economizar utilizando ela.

Agora suponhamos que você queira selecionar uma div com um id em específico: (`"id_da_div"`). Bom, mas até agora não fizemos nada de interessante. Suponhamos que você queira esconder todas as Tag's 'p' de um local em específico: (`"p", "id_da_div"`).`hide()`.

Fácil, não é? E para fazer o inverso é bem fácil, e também intuitivo. Basta usar `.show()`. A JQuery ainda conta com efeitos do tipo `fadeIn`, `fadeOut` e `fadeTo`: (`"p"`).`fadeIn("slow")`; (`"p"`).`fadeOut("slow")`; (`"p"`).`fadeTo("slow", 0.5)`;

Para pegar todos os campos de um formulário é só usar:

```
frm=$("#"+formulario);
dados=$(
    "select,input[@type=hidden],
    input[@type=text],
    input[@type=radio][@checked]", frm
).serialize();
```

## 7. Conclusão

O Ajax redefine o significado de páginas dinâmicas, pois elementos HTML são alterados conforme respostas dos aplicativos no servidor. A Web se aproxima do desktop, e os usuários passam a não distinguir mais entre aplicativos locais e remotos.

Nos próximos 3 anos, com exceção de softwares para áudio, imagem, vídeo e jogos, encontraremos diversos substitutos para softwares atualmente só encontrados em Desktops.

AJAX não é em si uma tecnologia, ou mesmo um framework de desenvolvimento de sistemas, mas sim se assemelha a um conceito de Design Pattern de utilização de Javascript, uma maneira eficiente para realizar as comunicações com o servidor de modo assíncrono, que traga benefícios significativos para as aplicações que o implementam. De certa forma, é uma nova maneira de pensar e desenvolver sistemas Web que otimizem a comunicação com o servidor, de modo que uma vez a interface da aplicação esteja no Web Browser do usuário, esta faça requisição apenas dos dados a serem exibidos e manipulados pelo usuário utilizando a interface da aplicação em seu Web Browser.

Como o AJAX em si é uma engine, códigos, em Javascript o mesmo apresenta algumas limitações quanto a suporte do Web Browser para a execução do mesmo, sendo que, caso a necessidade de implementar algo específico que não se encontre no conjunto de uma biblioteca Javascript de AJAX, o desenvolvedor deverá codificar tais códigos de maneira que os mesmos sejam compatíveis com o maior número de Web Browsers possíveis.

Atualmente temos visto uma corrida por parte dos desenvolvedores, para aprender o tal AJAX, sendo que este era algo que já existia a um bom

tempo antes de ser denominado “AJAX”; temos visto os casos de sucesso como as aplicações do Google, que implementam com sucesso o AJAX, onde vemos que realmente funciona; porem em sistemas com um volume de eventos, um tanto grande, como por exemplo, algumas animações, notamos que o mesmo apresenta algumas dificuldades de representar de maneira, que mantenha a qualidade.

O AJAX é uma boa alternativa atual, para o desenvolvimento de aplicações com interfaces mais ricas e com uma melhor metodologia de comunicação com o servidor.

## 8. Referências Bibliográficas

1 – ONTIVERO, M. S., **O que são os Web Standards?**

Acessado em: 29/04/2007.

Disponível em : <http://msontivero.wordpress.com/sobre-tabless/>

2 – PEREIRA, H. C., **Introdução a Semântica Web.**

Acessado em: 29/04/2007.

Disponível em: <http://www.revolucao.etc.br/archives/introducao-a-semantica-web/>

3 – Wikipédia.

Acessado em: 29/04/2007.

Disponível em: <http://www.wikipedia.com>

4 - REIS, Ricardo. et al. **Artigo AJAX: Introdução.**

Acessado em: 17/05/2006.

Disponível em: <http://pwp.net.ipl.pt/alunos.isel/24138/AJAX/IntroducaoAJAX.pdf>