

# Economia de Tráfego e Tempo em Aplicações Web Utilizando Técnicas Ajax

**Carlos Roberto Alves Rolim<sup>1</sup>**

carlosrar@gmail.com

**Ênnyo José Barros de Araújo**

ennyo.jose@gmail.com

**Juliano César Tozzi**

jctozzi@gmail.com

União dos Institutos Brasileiros de Tecnologia – UNIBRATEC

**Resumo:** O objetivo deste artigo é explicar tecnicamente o funcionamento do AJAX, com a construção de um exemplo prático, demonstrando os resultados obtidos nas pesquisas científicas realizadas com tal implementação. Além disso, este trabalho visa fornecer embasamento teórico sobre esse novo método de desenvolvimento Web. São apresentados gráficos e dados estatísticos, os quais possibilitam a avaliação dos benefícios e limitações desse novo paradigma da programação Web, que vem apontando para o futuro do desenvolvimento das aplicações on-line. Mostra-se ainda como essa inovadora solução contribui com o estado atual da Internet, popularizada e conhecida sob o termo Web 2.0, que sumariza toda a nova estrutura da Web, bem como suas modificações e remodelagens tecnológicas e conceituais.

**Palavras-Chave:** AJAX; Web; Desenvolvimento; Sistema; Web 2.0; Internet.

**Abstract:** The main objective of this paper is to explain technically the working of AJAX, with the construction of a practical example, showing the results obtained by the scientific researches held with such implementation. Besides, this work aims to give theoretical basis about this new method of Web development. Graphics and statistics data will be presented, which make possible the evaluation from the benefits and limitations of this new paradigm of Web programming, that indicates the future of the development of on-line applications. Yet, it has been showed how this innovative solution contributes to the actual state of internet, popular and well-known under the term Web 2.0, which sums up all the new Web structure, as well as its changes and technological and concepts remodels.

**Key-Words:** AJAX; Web; Development; System; Web 2.0; Internet.

## 1. Introdução

O modelo atual de negócios e de comercialização no Brasil, como em qualquer outro país, exige cada vez mais agilidade e, conseqüentemente, resultados instantâneos. Informação consistente e atualizada é a chave para àqueles que desejam estar sempre um degrau acima dos seus concorrentes.

É notável a crescente procura das empresas por novos métodos de comunicação e de interação com os seus clientes e, com certeza, a Internet apresenta-se como a melhor opção para esse tipo de integração. No ambiente empresarial, adotar um software baseado na Internet significa otimizar o tempo e os custos com a instalação do sistema em

cada estação de trabalho, o que implica em um benefício especial: a redução de custos com manutenção. Com o advento de tecnologias capazes de tornar a utilização de sistemas on-line tão simples e dinâmicos quanto às aplicações *desktop*, observa-se, dia após dia, o aumento de novos softwares completos atuando na *Web*.

Há dois anos, as aplicações *Web* eram sinônimos de lentidão, desorganização e insegurança, além de serem soluções pobres em dinamismo e interatividade. Tal afirmação é perfeitamente compreensível, tendo em vista que a grande maioria dos sistemas on-line construídos até hoje, foram desenvolvidos de forma digamos, irregular. Inúmeras são as tecnologias disponíveis para o desenvolvimento *Web*, porém, poucos as utilizam

corretamente. É inegável que a demora na navegação de algumas aplicações Web, na grande maioria dos casos, é fruto do modo como essas páginas foram projetadas e construídas. Segundo Isabel (2006) “a lentidão de páginas em websites muitas vezes é 100% culpa de um código HTML desorganizado e a comunicação falha com os servidores”. Não se deseja que o usuário tenha um conhecimento profundo sobre acessibilidade e usabilidade; espera-se, sim, que a interface apresentada seja a mais intuitiva possível para o usuário e que, apenas, preocupe-se em atingir seu objetivo frente àquele site, com menos cliques e obtendo informações necessárias seguindo uma seqüência lógica. A Web deve fortalecer e não frustrar.

O uso de padrões no desenvolvimento e a adoção de novas técnicas de programação Web, como o AJAX, vem quebrando esses paradigmas que assolam as aplicações para Internet. Em paralelo, a forma como os sistemas são projetados, também, passa por mudanças técnicas e conceituais. A chamada Web 2.0 altera não somente a camada de apresentação ao cliente final, mas também a forma como os serviços dessas aplicações são disponibilizados, oferecendo inovadoras possibilidades ao mercado tecnológico. Segundo João Romão (2006): “Trata-se, essencialmente, de usar a Web como plataforma para aplicações e não, apenas, como meio de entrega expressa de conteúdo”. A atitude Web 2.0 não tenta resolver as coisas do modo tradicional. e sim, utilizar as possibilidades da Internet para facilitar a vida do usuário. Tal tipo de atitude cria mais oportunidades de êxito porque remete ao modelo de pequenas e eficientes equipes de desenvolvimento, acabando com os projetos que perduram por anos e consomem milhões.

As primeiras experimentações para ultrapassar a metáfora dos documentos Web estão sendo realizadas com a utilização das técnicas AJAX no desenvolvimento das aplicações. Graças ao AJAX, o foco dessas novas aplicações está saindo da simples informação e passando a total interação. Agora, além do acesso, os websites desenvolvidos sob essa perspectiva podem oferecer ao usuário final a manipulação e transformação do conteúdo. E este é um excelente caminho para remodelar todo o conteúdo disponível na Internet e transformá-lo em conhecimento.

## 2. O que é AJAX

O termo AJAX - acrônimo para Asynchronous JavaScript and XML (Javascript Assíncrono e XML) - foi criado por Jesse James Garret para definir de

forma mais concreta (ou comercial) o uso do aglomerado de tecnologias presentes em sua utilização. Segundo Garret (2005): “O AJAX não é uma tecnologia. Mas, na realidade, várias tecnologias, cada uma progredindo de modo independente e que se juntaram de maneira a poder explorar novas formas de melhorar a interação com os usuários de aplicações Web.”.

O AJAX não é uma novidade tecnológica, mas sim um novo modo de utilizar tecnologias já disponíveis. Abaixo algumas das tecnologias que compõem o AJAX:

- I. XHTML como linguagem de marcação e CSS para definição dos estilos;
- II. Exibição e interação utilizando DOM (Document Object Model);
- III. Troca de dados usando XML (Extensible Markup Language);
- IV. XMLHttpRequest funcionando de forma assíncrona na comunicação com o servidor;
- V. Javascript para fundir e manipular todas as partes do processo.

O uso das tecnologias descritas acima gera o conceito de criar um sistema interativo, em que usuário final tenha uma navegação assíncrona, ou seja, que o mesmo possa realizar diversas ações simultâneas sem ter de esperar que o servidor processe as informações para que ele possa continuar navegando. Apenas o trecho da página requisitada é modificado diminuindo, dessa forma, o tráfego de informações desnecessárias pela rede e reduzindo também o processamento por parte do servidor. A cada novo conteúdo solicitado pelo usuário, não é necessário renderizar novamente a mesma página (efeito refresh). Segundo Paulo Lomanto (2006), “Tanto para Websites quanto para intranets e sistemas corporativos, o AJAX representa uma redução de até 80% no tráfego de informações entre o servidor Web e o cliente que está utilizando o recurso”.

## 3. Por que usar AJAX?

A maneira como o conteúdo das páginas Web é apresentado vem evoluindo constantemente, de um modo próprio e inovador. No início, essa apresentação era feita apenas de forma textual, o que não despertava muito interesse no usuário. Com o passar do tempo surgiram tecnologias capazes de exibir páginas estáticas com algumas figuras, como o HTML.

Existe uma lacuna técnica entre aplicações desenvolvidas para Web e em aplicações feitas para desktops. Esta distância fica mais evidente quando

se analisa o tempo de resposta. Nos *desktops* a resposta é imediata. Numa aplicação *Web*, essa arquitetura é diferente, portanto, vejamos:

- O acesso a memória é diferente;
- A aplicação não está instalada no cliente e sim no servidor;
- Para a comunicação, existe a necessidade de se estabelecerem conexões e requisições de dados;
- Toda requisição gera um *refresh* de tela, o que impossibilita muitas aplicações de se utilizar os recursos da Internet em determinadas aplicações, para as quais o tempo de resposta é algo essencial.

O AJAX minimiza essa distância, possibilitando a construção de aplicações *Web* com grande similaridade às aplicações *desktops* e sem a predileção por uma plataforma de sistema operacional específica. Entretanto, como toda tecnologia que surge no mercado, o AJAX possui vantagens e desvantagens.

### 3.1. Vantagens

Existem inúmeras vantagens ao se usar essa técnica. Certamente, seu uso deve ser “dosado” e sempre utilizado de forma coerente para não comprometer a legibilidade e manutenção de sua aplicação. Especialistas em usabilidade e desenvolvimento elegeram alguns pontos-chave a serem considerados. Segundo Michael Mahemoff (2006), podemos considerar:

*“Tráfego mínimo – As aplicações deveriam mandar e receber do servidor o mínimo de informação possível. O AJAX pode diminuir consideravelmente o tráfego entre cliente e servidor. Usando AJAX certamente sua aplicação não estará enviando informação desnecessária”.* A largura de banda é melhor utilizada, diminuindo a latência de rede, conseqüentemente obtém-se um melhor desempenho nas aplicações;

*“Interface amigável – Aplicações AJAX introduzem um conceito diferente do modelo ‘clica-espera’ suportado atualmente na internet. Outros paradigmas do tipo ‘arrasta e solta’ e ‘duplo clique’. Não importa qual paradigma usar, é necessário que o usuário saiba o que vai fazer após”.* Podemos então, construir aplicações mais intuitivas, com um conforto visual muito melhor, além de fornecer um

ambiente mais familiar ao usuário, deixando-o mais próximo das aplicações *desktop* a que ele está acostumado;

*“Convenções - não desperdice tempo reinventando novos modelos de interação com o usuário, utilize os padrões já conhecidos do desktop e diminua a curva de aprendizado”;*

*“Acessibilidade – considere a maior faixa de clientes possíveis, considere todas as possibilidades do uso de navegadores antigos ou os do tipo texto, antevêja algumas destas variantes e programe para um público inesperado.”;*

*“Desenvolvimento” - algumas aplicações antes só disponíveis para máquinas locais podem ter o seu desenvolvimento feito para navegadores Web;*

*“Aplicações Web mais rápidas, inteligentes e seguras.”* podemos agora ter carrinhos de compras inteligentes e outras aplicações que colaborem para deixar as operações de entrada e saída de dados mais simples para o usuário final;

Com AJAX, o usuário consegue interatividade e usabilidade sem demandar um elevado consumo de recursos para que esses sejam disponibilizados. O AJAX é capaz de associar eventos a um maior número de ações do usuário. Dessa forma, os conceitos fundamentais de interatividade com o usuário, como o famoso *drag-and-drop* (arrastar e soltar), torna-se perfeitamente possível. Do ponto de vista da usabilidade, esta liberdade não é importante, apenas, porque ela permite exercer nossa imaginação, mas sim porque ela nos possibilita combinar as ações disparadas pelo usuário e as solicitações ao servidor de uma forma bem mais completa.

As vantagens da utilização do AJAX tornam-se visíveis principalmente para os usuários. É possível imaginar muitas funcionalidades em que esta interação seja conveniente: validação de dados, correção ortográfica automática, cálculos dinâmicos, emissão e gerenciamento de relatórios gerados, a partir de uma base de dados, etc.

### 3.2. Desvantagens

Como acontece em toda tecnologia, algumas vantagens podem trazer desvantagens. No

desenvolvimento de aplicações *Web* existem alguns pontos a serem considerados:

Programação mais complexa – Por ter uma estrutura mais complexa, torna-se mais difícil desenvolver utilizando-se AJAX, a curvatura de aprendizado é muito maior e os programas se tornam de difícil manutenção;

Botões avançar e voltar – fica muito difícil controlarem os botões avançar e voltar do navegador, quando se está navegando por uma página que usa AJAX, o que pode complicar a restauração de um estado anterior ao se utilizarem esses botões. Esse ainda é um desafio a ser vencido na programação, usando-se essa tecnologia;

Salvar – fica difícil e, muitas vezes, impossível salvar o conteúdo de uma site se o mesmo tiver sido desenvolvido 100% em AJAX;

Depuração dificultada - Além de ferramentas de “debugação” de JavaScript serem raras, um aumento na complexidade do programa pode trazer diminuição da produtividade de um sistema;

Portabilidade – caso o JavaScript não esteja habilitado ou não exista suporte a ele no navegador do usuário, fica impossível utilizar o sistema em questão;

Motores de busca – se o site for feito completamente em AJAX, os robôs de busca de *websites* de pesquisa não conseguem identificar e indexar o site.

Então, antes de construir páginas ou sistemas *Web*, os pontos acima devem ser fortemente levados em consideração. Segundo Nicholas C. Zakas (2006) *“A tecnologia AJAX serve para melhorar a experiência de usuário com a web, a tecnologia por detrás do desenvolvimento é simplesmente um ferramenta para atingir um determinado fim. Certifique-se que sua aplicação será útil e usável.”*.

#### 4. Quem usa?

Indiscutivelmente, a empresa que mais se destaca na utilização do AJAX, atualmente, é a Google. As maiores e mais bem sucedidas aplicações desenvolvidas pela Google (Google Maps, Gmail e orkut), possuem implementações AJAX. O Google Suggest também é um excelente

exemplo que demonstra o quanto o AJAX pode tornar uma aplicação *Web* poderosa. Ainda existem diversas outras implementações na *Web* que utilizam o AJAX no seu desenvolvimento, como a ferramenta de busca A9 da Amazon, o Flickr, gerenciador de fotos on-line, o Writely, editor de textos on-line (adquirido recentemente pela Google), o NumSum, editor de planilhas *on-line*, o netVibes, gerenciador de conteúdo personalizado, e diversos outros sistemas que de certa forma embutem essa nova técnica de desenvolver para *Web*.

Projetos como esses, anteriormente, citados demonstram que o AJAX não é apenas mais uma “mania” passageira, e sim uma poderosa nova forma de encarar a realidade *Web*. A *Adaptive Path*, empresa que formulou e originou o termo, diz que os desenvolvedores que vêm utilizando o AJAX *“apenas estão arranhando a superfície das interações ricas e a comunicabilidade que as aplicações AJAX podem prover.”*.

#### 5. Padrões

No cenário atual da tecnologia da informação marcado por uma acelerada evolução, a *Web* necessita de orientação a fim de que seja capaz de desenvolver todo o seu potencial. Esse direcionamento pode ser obtido através dos padrões *Web*.

Nos idos da programação *Web*, os códigos de formatação (HTML e CSS) eram muito complexos e sem organização, o que tornou a navegação em muitos sites da Internet um verdadeiro caos. Porém, isso vem mudando rapidamente. O código interpretado no cliente (browser) nos dias atuais é bastante simplificado. Com o lançamento do XHTML (sucessor do antigo HTML), as linhas de formatação das páginas reduziram-se drasticamente. Hoje em dia, temos códigos padronizados de fácil manutenção e melhor legibilidade.

Mas, essas mudanças não são méritos, apenas, da evolução das linguagens citadas. A maneira de pensar do desenvolvedor *Web* também mudou. É cada vez mais freqüente observar a preocupação dos programadores com relação à usabilidade e acessibilidade dos seus sites e sistemas. Antigamente, pouco se ouvia falar sobre essas características, mas hoje em dia é comum a preocupação para que os códigos sejam interpretados corretamente por diversos tipos de navegadores, que sejam acessíveis pelo teclado e que possam ser visualizados em qualquer tipo de dispositivo de acesso. E mais ainda, a questão da acessibilidade vem interferindo no desenvolvimento das aplicações *Web* a ponto dos desenvolvedores preocuparem-se até com o tipo de usuário que utiliza

sua aplicação. A analista de sistemas, Karina Guimarães (2006), define o usuário como o centro de tudo: “*As aplicações Web devem evitar que o usuário deixe de encontrar a informação desejada e abandone a visita em menos de 5 segundos.*”. Para isso, é necessário um conteúdo atrativo e disposto em uma interface agradável e bem elaborada.

Diversas são as organizações de padronização que influenciam a Internet. Inegavelmente, o mais expressivo trabalho nesta área de definição de padrões para *Web* é feito pelo W3C (World Wide Web Consortium).

O W3C é um consórcio da indústria internacional destinado a conduzir a Internet rumo ao seu completo potencial. Fundado em 1994 e liderado pelo pai da *Web*, Tim Berners-Lee, o W3C possui mais de 450 organizações membro, dentre elas destacam-se a Microsoft, a AOL, a Apple, a Sun, a Adobe e a Macromedia, além de uma grande variedade de fabricantes de hardware e software.

Os princípios fundamentais do projeto W3C são:

- **Interoperabilidade:** especificações para as linguagens e para os protocolos da *Web* devem ser compatíveis entre si, além de prover o acesso à *Web* a partir de qualquer tipo de hardware ou software;
- **Evolução:** a *Web* deve ser capaz de acomodar tecnologias futuras. Princípios de projeto, como, simplicidade, modularidade e extensibilidade aumentarão as chances da *Web* trabalhar com tecnologias emergentes, tais como dispositivos móveis e televisão digital, bem como outras que ainda estão por vir;
- **Descentralização:** este é o princípio mais novo e mais difícil de aplicar. Para fazer com que a *Web* alcance proporções mundiais, enquanto resiste a erros e quedas do sistema, sua arquitetura deve limitar ou mesmo eliminar as dependências em registros centrais com sede em grandes potências mundiais.

Estes princípios guiam as atividades do W3C, que desenvolve especificações abertas a fim de garantir a interoperabilidade, a evolução e a descentralização dos produtos relacionados com a *Web*.

## 5.1. WEB 2.0

A *Web 2.0*, termo criado por Dale Dougherty em uma conferência realizada em outubro de 2005,

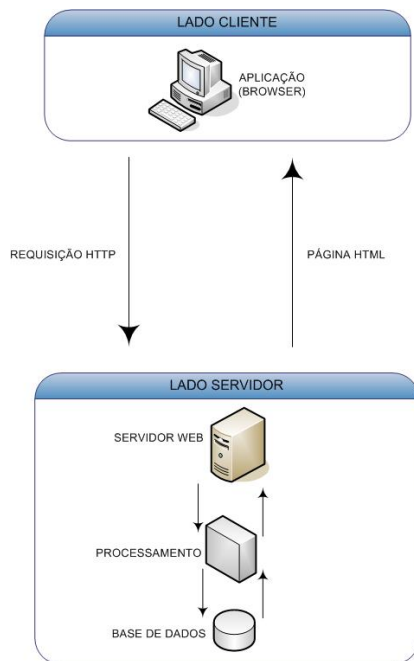
em São Francisco – EUA, retrata o estado atual da Internet. Um local acessível a todos, onde é possível desenvolver, pesquisar, informar e interagir, sem a interferência de corporações ou restrições de uso. Adotando uma ótica mais profunda, pode-se perceber a *Web 2.0* ultrapassando as barreiras físicas, a ponto de, cada vez mais, se observar o nascimento de aplicações robustas e seguras que nada deixam a desejar às tão conhecidas aplicações *desktop*. Chegará o dia em que o usuário final não necessitará de softwares, com suas licenças exorbitantes, instalados no seu PC, pois ele terá soluções alternativas disponíveis na *Web*, a qualquer hora do dia, independente de plataforma operacional ou configuração de hardware.

Keith Robinson (2006) do site Asterisk, afirma “*O conteúdo da Web 2.0 é a possibilidade democrática e sem barreiras de exercer sua possibilidade de opinar.*”. A *Web 2.0* vem ao mundo sobre a perspectiva da acessibilidade total, ou seja, as aplicações *Web* que estão surgindo dentro desse novo paradigma estarão prontas para serem executadas nos celulares, videogames e na tão esperada TV digital.

Fundamentalmente, a atitude empreendedora da *Web 2.0* tem maior possibilidade de dar certo porque foi concebida sob o prisma da colaboração e do conteúdo multiplataformas. Os usuários lêem mais, criam mais e, conseqüentemente, colaboram mais.

O efeito da *Web 2.0* para os desenvolvedores também é notório. Todo programador sabe o quão tortuoso é ter de modificar o escopo de um sistema que não segue nenhum padrão de desenvolvimento. É justamente aí que se percebem as metodologias da *Web 2.0*, que propiciam ao programador maneiras práticas de alterar módulos do sistema sem alterar o projeto como um todo.

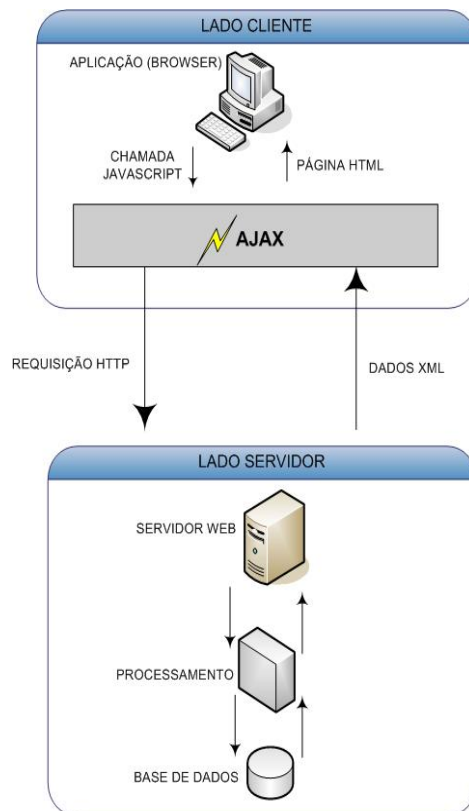
## 6. Funcionamento das páginas WEB



**Figura 1: Formato tradicional de comunicação entre cliente/servidor**

Em uma aplicação *Web* tradicional, a maior parte das ações do usuário na interface ativa uma requisição HTTP ao servidor. Este, por sua vez, processa a solicitação (retornando dados, interpretando números, etc.) e, então, retorna uma página no formato (x)HTML para o *browser* do usuário.

Toda vez que o usuário interage com o *site*, um outro documento é enviado para o navegador, contendo o mesmo conteúdo de cabeçalhos e dados. 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 passar pelo mesmo mecanismo.



**Figura 2: Formato de comunicação cliente/servidor com AJAX.**

Em uma aplicação que utiliza as técnicas AJAX, parte da lógica é movida para o navegador. Dessa forma, quando o usuário acessa a aplicação, um documento de conteúdo mais complexo é entregue ao *browser*. Este documento permanecerá com o usuário durante toda a sessão. Mesmo que o usuário altere sua aparência original, o script que compõe essa página ficará encarregado de responder essas informações e decidirá o que deverá ser feito sem que haja uma única atualização de tela. Ao invés de carregar uma página inteira no início da sessão, o navegador, na verdade, carrega uma ferramenta AJAX, que fica responsável por renderizar toda a interface visualizada pelo usuário e ainda realizar a comunicação com o usuário. Tudo isso deve ocorrer assincronamente, ou seja, um processo deve estar acontecendo independente do outro, tornando assim a navegação mais eficaz. Isso garante com que o usuário nunca se depare com uma página em branco no seu navegador, esperando por uma resposta do servidor.

As ações de um usuário durante sua navegação dentro da aplicação gerariam, todas elas (por mais simples que fossem), uma solicitação HTTP. Com AJAX essa requisição é feita pelo javascript associado a objetos XML. Assim sendo, qualquer resposta para essas ações do usuário são tratadas pelo próprio AJAX. Se essa ferramenta necessitar de

algo que está no servidor, para responder a essas ações, requisições assíncronas serão utilizadas, sem prejudicar a navegação do usuário.

## 7. O AJAX e as linguagens de servidor

Para que o AJAX torne-se funcional, faz-se necessário o uso de uma linguagem de servidor que permita a integração da aplicação que está atuando no lado cliente (*browser*), com os dados que se encontram armazenados no servidor. Existem diversas linguagens servidoras disponíveis, dentre elas o PHP, o ASP.net, a Perl, o Cold Fusion, JSP e Servlets, dentre outras. Porém, o que é mais plausível quando se fala em interação de uma aplicação que utiliza AJAX em uma linguagem de servidor, seja ela qual for, é a capacidade que essa linguagem possui, para que o desenvolvedor possa implementar scripts organizados, preferencialmente, orientados a objetos, capazes de fornecer uma base sustentável para atuar em perfeita sincronia com as requisições solicitadas pelo usuário durante sua navegação.

Soluções bem desenvolvidas no lado servidor, estruturadas em classes e providas de métodos que possibilitem a reusabilidade, bem como a prática manutenção dos mesmos, agilizam a movimentação e distribuição dos dados. E se tratando de AJAX, esse é um fator indispensável para o bom funcionamento das aplicações desenvolvidas sob essa nova tendência.

## 8. Soap e webService + AJAX

A Internet vem se popularizando num ritmo quase que frenético e junto com essa popularização surge a necessidade da integração dos sistemas que a utilizam. A definição de um protocolo simples e baseado em XML era fundamental para realizar a integração entre sistemas distintos. Apesar do SOAP (Simple Object Access Protocol – Objeto Simples de Protocolo de Acesso) não ter sido desenvolvido com esse objetivo, adequou-se perfeitamente ao mundo da integração on-line.

O SOAP é um protocolo elaborado para facilitar a chamada remota de funções via Internet, permitindo assim que dois programas diferentes comuniquem-se de uma maneira tecnicamente muito semelhante à invocação de documentos Web.

O SOAP oferece diversas vantagens não encontradas em outros protocolos como o DCOM, CORBA ou mesmo o TCP/IP:

- É simples e fácil de implementar;

- É um padrão adotado pela W3C, além de diversas outras empresas;
- Utiliza os mesmos padrões da Web para quase tudo, a comunicação é feita via HTTP com pacotes virtualmente idênticos; os protocolos de autenticação e encriptação são os mesmos; a manutenção de estado é feita da mesma forma; é normalmente implementado pelo próprio servidor *Web*;
- Passa por Firewalls e roteadores que o vêem como uma comunicação HTTP;
- Os dados e as funções são descritos em XML, tornando-o fácil de usar e bastante robusto;
- Independe de sistema operacional e CPU;
- Pode ser usado tanto de forma anônima como com autenticação (login/senha);

As solicitações SOAP podem ser realizadas em três padrões: GET, POST e SOAP. Os padrões GET e POST são idênticos aos pedidos feitos pelos *browsers*. O SOAP é um padrão que se assemelha ao POST, porém seus pedidos são feitos em XML e possibilitam a utilização de recursos mais aprimorados como passar estruturas e *arrays*. O XML descreve os dados em tempo de execução e evita problemas causados por inadvertidas mudanças nas funções, já que os objetos chamados têm a possibilidade de sempre validar os argumentos passados por essas funções, tornando o protocolo muito mais confiável.

O SOAP define, ainda, um padrão chamado WSDL, que descreve objetos e métodos disponíveis através de páginas XML acessíveis em toda a Web. O WSDL funciona da seguinte maneira: quem publicar um serviço também cria as páginas. Quem desejar invocar o serviço, pode utilizar essas páginas como uma espécie de documentação de chamada e, ainda, usá-las antes de chamar as funções que verificam se alguma coisa foi alterada.

O SOAP e o AJAX são integrados de forma bastante simples, partindo do princípio de que os dois utilizam documentos XML como ferramenta básica de execução. Utilizando AJAX com SOAP, é possível chamar métodos implementados em aplicações desenvolvidas pelo próprio programador e ainda acessar aplicações de outros desenvolvedores, fazendo com que as aplicações se tornem totalmente interligadas a qualquer servidor que ofereça serviços Web pelo protocolo SOAP.

## 8.1 WebServices

Um Webservice é um componente acessível, através de protocolos padrões da Internet (como o SOAP). Sendo componentes, esses serviços possuem uma funcionalidade que pode ser reusada sem a preocupação de como a mesma é implementada. Os Webservices combinam os melhores aspectos do desenvolvimento baseado em componentes e a Internet. A enciclopédia virtual Wikipédia define-os da seguinte maneira: “*Webservices é um conjunto de operações/métodos que podem ser acessados remotamente*”.

Os Webservices podem ser utilizados em quase tudo que envolva troca de dados e informações. Qualquer plataforma que interprete rotinas HTTP e trabalhe com XML pode usar os dados/informações dos webservices, sem restrição alguma.

Associar os Webservices ao uso do AJAX é algo imprescindível para torná-los ainda mais robustos em uma aplicação. Com eles podem-se acessar rotinas de validação de cartão de crédito, endereçamento postal (CEP), calcular valores de fretes dos sites de comércio eletrônico, aperfeiçoar um sistema de notícias, enfim, um leque gigantesco de possibilidades se abre diante do desenvolvedor que consome os recursos dos Webservices.

## 9. Aplicação

Com o intuito de obter subsídios que pudessem ratificar tudo o que foi exposto sobre o AJAX, foi necessário o desenvolvimento de um sistema piloto, com duas versões diferentes. Um que utiliza técnicas AJAX em seu conteúdo, processamento e fluxo de dados. E um outro com implementações de uma aplicação Web normal, ou seja, um sistema construído, apenas, com o uso dos recursos oferecidos pela linguagem de servidor PHP, sem atualizações dinâmicas ou links rápidos na estrutura da aplicação.

A escolha da aplicação foi um ponto importante. As pesquisas visam atingir soluções práticas que levem algum benefício a sociedade, seja com o melhoramento de um processo ou com a invenção de um produto. Neste contexto, a escolha de uma loja virtual é bastante pertinente.

Dessa forma, iremos expor, nos próximos parágrafos, gráficos gerados a partir de testes comparativos realizados com esses dois modelos de sistemas, atestando então todos os benefícios que o AJAX pode proporcionar a uma aplicação Web.

### 9.1 Especificação do exemplo

Os modelos construídos são todos multiplataforma. Pode-se facilmente construir os modelos em uma plataforma e testá-los em uma outra. Este modelo de análise foi construído completamente em plataforma WINDOWS®, não implicando que o mesmo não possa ser desenvolvido em outra plataforma, como o LINUX, por exemplo. Todas as especificações estão descritas abaixo.

#### 9.1.1 Ferramentas utilizadas

Os dois modelos de aplicação foram construídos utilizando as seguintes ferramentas:

- Para a Base de dados, MySQL 5.0;
- Como linguagem servidora foi utilizado o PHP 5.1.1;
- Na camada do cliente utilizamos: XHTML, XML, CSS e Javascript.

#### 9.1.2 Áreas internas da aplicação

As aplicações são baseadas na estrutura de uma loja virtual, pois neste tipo de sistema podemos encontrar ferramentas e funcionalidades em que o uso de técnicas AJAX é facilmente visualizado. A aplicação, esta dividida em duas áreas.

##### 9.1.2.1 Área pública

Na área pública teremos a vitrine on-line, que conta com a exibição dos produtos à venda dispostos por categorias. Nesta mesma área está o “carrinho de compras”, no qual os produtos adicionados ao carrinho aparecem listados. Ao encerrar sua navegação pelo site, o usuário pode finalizar então seu pedido confirmando assim a sua compra.

##### 9.1.2.2 Área administrativa

Na área administrativa, construímos dois formulários de cadastrado básicos a um comércio eletrônico: o cadastro de produtos e o de fornecedores.

## 9.2 Loja virtual

Ao acessar a loja virtual, o usuário poderá visualizar um menu horizontal no topo do site, no qual são apresentadas as categorias dos produtos a venda na loja virtual. Clicando em cada categoria serão expostos os seus respectivos produtos em uma área específica do site. Esses produtos estão representados por fotos, contendo, ainda, uma breve descrição e o seu preço atual.



### 9.2.1 Área pública (AJAX)

Utilizamos um sistema de paginação bastante eficiente que proporciona ao usuário uma navegação rápida e agradável pelos produtos ofertados no site. Ao ir-se de uma página para outra, apenas o

conteúdo central, ou seja, os produtos e suas descrições, são alternados, conservando, assim, toda a estrutura em torno desta área, evitando um desperdício de tráfego de rede e economizando tempo de navegação

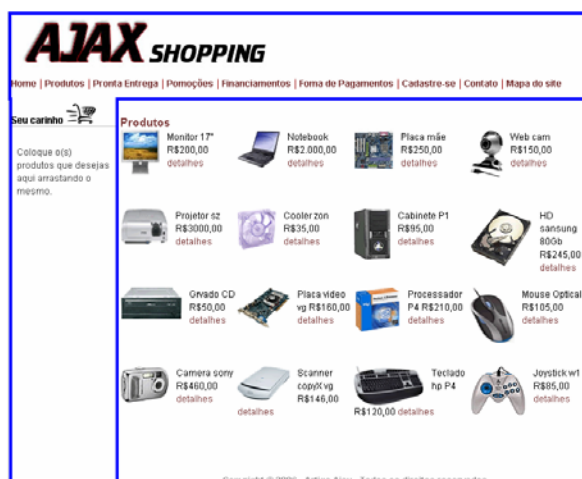
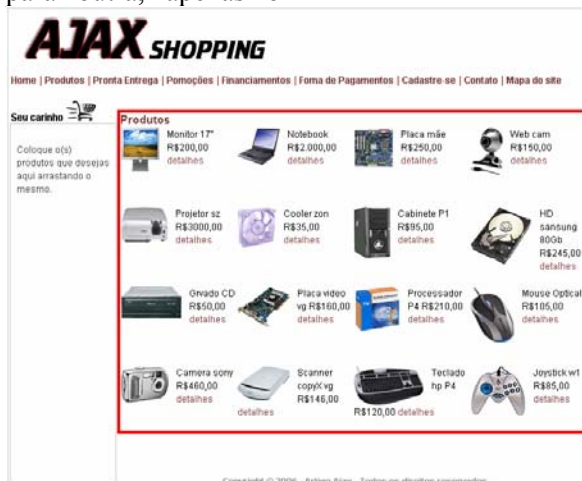


Imagem 1. Em azul, a área padrão do sistema. Em vermelho, a área que sofre modificações.

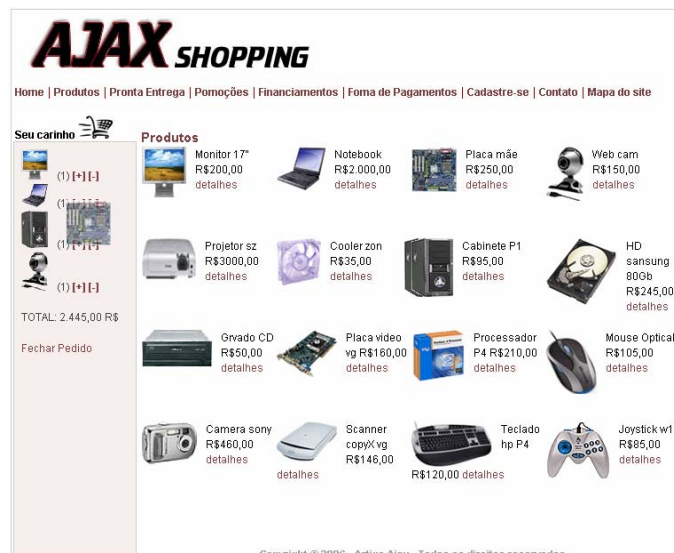
### 9.2.2 Área pública (s/ AJAX)

Na área pública temos a paginação de produtos sem a utilização de AJAX, então, a cada mudança na paginação, temos algumas requisições ao servidor e uma nova página é montada com todos os próximos 16 produtos e seus respectivos menus e o “carrinho de compras”.

### 9.2.3 Realizar compra (AJAX)

Para realizar um compra nessa loja virtual, basta que o usuário clique sobre o ícone (foto) do produto desejado e o arraste para o “carrinho de compras”, disposto lateralmente à área de exibição de produtos, e que estará disponível e acessível durante toda a

sessão de navegação do usuário. Cada produto adicionado ao “carrinho de compras” é simbolizado, dentro do mesmo, por uma versão miniatura da sua foto. Existe ainda um link dedicado somente ao aumento e diminuição da quantidade de cada item inserido no “carrinho de compras”. Tudo isso é realizado sem uma única atualização de página, de forma que o usuário apenas terá de esperar por uma resposta do servidor no momento em que finalizar o seu pedido (clcando sobre o link “fechar pedido”), em que todos os dados acumulados, durante a navegação, são processados e armazenados definitivamente. O *screenshot* abaixo, mostra o exato momento que o usuário arrasta um produto para o seu “carrinho de compras”.



**Imagem 2. Tela principal da loja virtual**

#### 9.2.4 Realizar compra (s/ AJAX)

Nesta versão da aplicação, a área pública do sistema também apresenta um local destinado à exibição dos produtos anunciados na loja virtual, oferecendo, ainda, a mesma opção de montar um pedido, inserindo itens no “carrinho de compras”. Entretanto, diferente da aplicação desenvolvida com técnicas AJAX, nesta o cliente deve clicar no link “comprar” existente ao lado de cada produto, esperar pelo envio da requisição ao servidor, visualizar seu pedido na página que possui o “carrinho de compras” e confirmar o pedido ou, se desejar, ir para o link “continuar comprando”, o qual o leva de volta para a relação de produtos, recarregando toda a página novamente. O sistema de paginação, que organiza os produtos listados na página principal, não possui nenhum espécie de mecanismo de montagem otimizada de templates ou atualização de conteúdo através de iframes. Os produtos são mostrados, de 16 em 16, com um link de navegação entre as páginas, que a cada clique, toda a estrutura do documento é remontada e os próximos 16 produtos são exibidos ao usuário.

#### 9.2.5 Área de cadastros (AJAX)

Na parte dedicada à administração da loja virtual, existem dois tipos de cadastros disponíveis

ao administrador do sistema: o cadastro de fornecedores e o de produtos. O primeiro, contém campos destinados à entrada de dados referentes aos fornecedores de produtos à loja virtual. Todos os campos contam com validação de dados (data, valor monetário, quantidade, e-mail, CNPJ, etc.). Desta maneira, todos os possíveis erros de digitação ou omissão de caracteres são apresentados durante o processo de cadastramento, possibilitando a correção imediata. Nesse formulário, embutimos o consumo de um WebService bastante útil para preenchimento de endereços. Dado um determinado CEP, o serviço retorna o nome da rua, bairro, cidade e estado pertencentes ao CEP informado, sem haver a necessidade de remontar toda a página só para exibir esses novos dados recolhidos do servidor. Neste ponto podemos observar perfeitamente o quanto o AJAX é poderoso, pois seria no mínimo tedioso pensar em consumir um WebService deste tipo, se tivéssemos de aguardar por uma resposta do servidor, guardar as informações já digitadas nos campos anteriores, atualizar o documento, remontar o formulário com os dados armazenados em seus respectivos campos e exibir os novos dados retornados. Com o AJAX, a atualização é imediata e o resultado torna-se instantâneo.

### CADASTRO DE FORNECEDOR

\* Razão Social:  ?

Nome Fantasia:

\* CNPJ:  ?

\* Insc. Estadual:  ?

Insc. Municipal:  ?

CEP:  ?

\* Endereço:  ?

\* Número:  \* Bairro:

\* Cidade:  Estado:

Complemento:

Referência:  ?

Telefone:

Telefone Alt.:

Fax:

Celular:

E-mail:  ?

Site:

Observações:

500

CADASTRAR PRODUTOS PARA ESTE FORNECEDOR?  Sim  Não

*Imagem 3: Formulário para cadastro de fornecedores.*

Do formulário de cadastro de fornecedores, pode-se acessar diretamente o formulário para cadastro de produtos, no caso do administrador do sistema, ao mesmo tempo em que cadastra um novo fornecedor, cadastrar também os produtos fornecidos por este fornecedor. O formulário para cadastro de produtos é exibido na mesma tela como uma janela sobrepondo o cadastro de fornecedores. Assim que for efetuado o cadastramento do novo produto, o mesmo já será mostrado em uma listagem contendo todos os produtos cadastrados no sistema, mais uma vez,

sem nenhum refresh de página. Outro recurso muito interessante utilizado nos formulários de cadastro, é a atualização de um select a partir de outro select, select's aninhados. Por exemplo, no cadastro de produtos, o administrador deve selecionar a sub-categoria à qual pertence o produto que está sendo cadastrado, porém, esta sub-categoria, por sua vez, pertence a uma categoria. Caso o administrador selecione uma categoria X e suas sub-categorias serão imediatamente exibidas no campo seguinte.

**Imagem 4. Formulário para cadastro de produtos, acessado através do cadastro de fornecedores.**

### 9.2.6 Área de cadastros (s/ AJAX)

No sistema sem AJAX o preenchimento dos cadastros é realizado de forma individual. Existem telas separadas para fornecedores e produtos. Toda a digitação é realizada e submetida ao servidor, que realiza as inclusões e remonta uma nova página no cliente para a digitação de um novo fornecedor ou produto.

## 10. Resultados

### 10.1 Especificações dos equipamentos utilizados

- Computador cliente: AMD Athlon 64 2800+ com 1Gbyte de RAM.  
Interfaces de rede: Para acessar internet: Marvell Yukon 88E8001 PCI GIGABIT ETHERNET;  
Para acesso a rede local: SIS 900 PCI ETHERNET;
- Computador servidor: AMD Sempron 2300+ com 512 de RAM  
Interface de rede: Para acesso a rede local: SIS 900 PCI ETHERNET;
- Equipamentos acessórios: HUB LG 10/100

- Modem Externo Motorola: Cabo de par trançado para interligação dos pontos da rede.

### 10.2 Resultado dos testes.

Os dois sistemas implementados passaram por diversos tipos de testes e análises, gerando resultados que tanto ajudam a entender o funcionamento do AJAX, como também comprovam toda a teoria que o envolve.

A pesquisa foi feita com os sistemas hospedados localmente (com exceção para as análises de tempo de resposta, onde os sistemas foram hospedados na Web) em uma máquina servidora, tendo o servidor *Web Apache 2.0*, o *PHP versão 5.1.1* e o *MySQL 5.0* devidamente instalados. Esse servidor foi acessado pela máquina cliente por intermédio de uma rede local de 10/100 MBytes. Os testes foram executados no dia 22/07/06 (sábado), das 16:08 às 17:45. O Mozilla Firefox versão 1.0.6 foi o *browser* escolhido para acessar as aplicações, e o software responsável pela a análise de tráfego de rede foi o *Etheral – Network Protocol Analyzer versão 0.99*, disponível em <http://www.ethereal.com>;

- **Tempo de execução:** aqui, podemos observar a quantidade de tempo despendido para a execução de uma determinada ação ou processamento no sistema desenvolvido com

AJAX, e o tempo necessário para efetuar a mesma ação no sistema sem AJAX. Cinco pessoas diferentes participaram deste teste, obedecendo aos critérios: todos preencheram o formulário de fornecedores e de produtos com os mesmos dados; realizaram um pedido com os

mesmos produtos e os mesmos valores. O tempo de cada pessoa foi cronometrado e, ao final todas as cinco marcas obtidas foram divididas pelo número de participantes do teste (cinco), obtendo-se assim as médias para cada ação.

	SISTEMA COM AJAX	SISTEMA SEM AJAX
Cadastro de fornecedor e produtos:	1,01 minuto (média)	2,06 minutos (média)
Composição de pedido:	13,14 segundos (média)	22,05 segundos (média)

- **Tempo de resposta:** essa análise apresenta o tempo de resposta do servidor para cada uma das ações apresentadas abaixo. Para esse teste, hospedamos os sistemas na Web, em um

servidor de plataforma Windows 2003 Server com suporte a PHP 5.1.1 e MySQL 5.0. A aplicação foi acessada em uma conexão banda larga de 100 Kbps.

	SISTEMA COM AJAX	SISTEMA SEM AJAX
Login:	2,9 segundos (média)	4,9 segundos (média)
Cadastro de fornecedor:	1,8 segundos (média)	3,9 segundos (média)
Cadastro de produto:	1,3 segundos (média)	3,0 segundos (média)
Fechamento de pedido:	0,8 segundos (média)	5,2 segundos (média)

- **Tráfego de Rede:** foi último resultado a ser apresentado e, com certeza o mais interessante de todos eles. Nesse teste, analisamos a quantidade de bytes que trafegaram na rede durante a realização de um mesmo processo. Cada ação foi executada por três vezes, e assim pudemos observar, claramente, a economia de bytes enviados e recebidos na aplicação com implementações AJAX, em relação ao sistema normal.

- **Processo de login**

	SISTEMA COM AJAX	SISTEMA SEM AJAX
Teste 1	Tráfego Total: 1.218 bytes Bytes enviados (cliente-servidor): 386 bytes Bytes recebidos (servidor-cliente): 832 bytes	Tráfego Total: 9.284 bytes Bytes enviados (cliente-servidor): 906 bytes Bytes recebidos (servidor-cliente): 8.378 bytes
Teste 2	Tráfego Total: 1.183 bytes Bytes enviados (cliente-servidor): 345 bytes Bytes recebidos (servidor-cliente): 828 bytes	Tráfego Total: 9.284 bytes Bytes enviados (cliente-servidor): 906 bytes Bytes recebidos (servidor-cliente): 8.378 bytes
Teste 3	Tráfego Total: 1.218 bytes Bytes enviados (cliente-servidor): 386 bytes Bytes recebidos (servidor-cliente): 832 bytes	Tráfego Total: 9.561 bytes Bytes enviados (cliente-servidor): 906 bytes Bytes recebidos (servidor-cliente): 8.657 bytes

- **Cadastro de fornecedores**

	SISTEMA COM AJAX	SISTEMA SEM AJAX
Teste 1	Tráfego Total: 1.483 bytes Bytes enviados (cliente-servidor): 359 bytes Bytes recebidos (servidor-cliente): 1.124 bytes	Tráfego Total: 10.733 bytes Bytes enviados (cliente-servidor): 8.852 bytes Bytes recebidos (servidor-cliente): 8.186 bytes
Teste 2	Tráfego Total: 1.483 bytes Bytes enviados (cliente-servidor): 359 bytes Bytes recebidos (servidor-cliente): 1.124 bytes	Tráfego Total: 10.733 bytes Bytes enviados (cliente-servidor): 8.852 bytes Bytes recebidos (servidor-cliente): 8.186 bytes
Teste 3	Tráfego Total: 1.483 bytes Bytes enviados (cliente-servidor): 359 bytes Bytes recebidos (servidor-cliente): 1.124 bytes	Tráfego Total: 10.721 bytes Bytes enviados (cliente-servidor): 8.852 bytes Bytes recebidos (servidor-cliente): 1.869 bytes

- **Cadastro de produtos**

	SISTEMA COM AJAX	SISTEMA SEM AJAX
Teste1	Tráfego Total: 1.322 bytes Bytes enviados (cliente-servidor): 358 bytes Bytes recebidos (servidor-cliente): 964 bytes	Tráfego Total: 15.973 bytes Bytes enviados (cliente-servidor): 12.382 bytes Bytes recebidos (servidor-cliente): 3.591 bytes
Teste2	Tráfego Total: 1.322 bytes Bytes enviados (cliente-servidor): 358 bytes Bytes recebidos (servidor-cliente): 964 bytes	Tráfego Total: 16.226 bytes Bytes enviados (cliente-servidor): 12.449 bytes Bytes recebidos (servidor-cliente): 3.591 bytes
Teste3	Tráfego Total: 1.610 bytes Bytes enviados (cliente-servidor): 466 bytes Bytes recebidos (servidor-cliente): 1.144 bytes	Tráfego Total: 16.226 bytes Bytes enviados (cliente-servidor): 12.449 bytes Bytes recebidos (servidor-cliente): 3.591 bytes

- **Processo de compra (inclusão de 4 itens no carrinho de compras e finalização do pedido, com exibição dos dados da compra)**

	SISTEMA COM AJAX	SISTEMA SEM AJAX
Teste1	Tráfego Total: 12.556 bytes Bytes enviados (cliente-servidor): 7.845 bytes Bytes recebidos (servidor-cliente): 4.711 bytes	Tráfego Total: 91.947 bytes Bytes enviados (cliente-servidor): 69.450 bytes Bytes recebidos (servidor-cliente): 3.591 bytes
Teste2	Tráfego Total: 12.441 bytes Bytes enviados (cliente-servidor): 7.190 bytes Bytes recebidos (servidor-cliente): 4.681 bytes	Tráfego Total: 91.949 bytes Bytes enviados (cliente-servidor): 69.450 bytes Bytes recebidos (servidor-cliente): 22.499 bytes
Teste3	Tráfego Total: 12.441 bytes Bytes enviados (cliente-servidor): 7.190 bytes Bytes recebidos (servidor-cliente): 4.681 bytes	Tráfego Total: 97.204 bytes Bytes enviados (cliente-servidor): 72.473 bytes Bytes recebidos (servidor-cliente): 24.725 bytes

## 11. Conclusão

A aplicação do AJAX na *Web 2.0* ainda não está sendo amplamente explorada. A própria *Web 2.0* encontra-se em fase embrionária, em que métodos de utilização e, até mesmo, conceitos e fundamentos ainda estão sendo calcados. Trata-se não apenas de remodelar a *Web* atual, mas sim de encorajar, desenvolvedores e utilizadores, a entenderem que a Internet deixou de ser, apenas, um veículo dinâmico de intercomunicação e de informação e passou à categoria de plataforma, na qual tudo, absolutamente tudo, acontece. A idéia de plataforma é ainda mais vasta. Emprega o uso da rede como plataforma de serviços, e tais aplicações podem localizar uma determinada funcionalidade e esta ser disponibilizada, sem a necessidade de uma nova requisição.

A técnica AJAX trouxe uma visão inovadora para o desenvolvimento *Web* e vem ajudando a embasar tudo aquilo a que se propõe a *Web 2.0*. Os paradigmas de interação entre aplicação/usuário estão sendo profundamente modificados por esse novo prisma das aplicações on-line. Nota-se intensamente a migração de softwares *desktop* para a “plataforma” *Web* e mesmo os sistemas que já atuam sob essa modalidade se encontram em processo de reformulação para atender às novas características da Internet, ditadas pela *Web 2.0*. Aliar o dinamismo e a agilidade dos resultados encontrados nas aplicações *desktop* à independência e portabilidade

oferecidos pelos sistemas *Web* tornou-se a meta de todo desenvolvedor de softwares.

Experiências realizadas com AJAX têm mostrado uma melhoria indiscutível na acessibilidade e na economia de tempo de navegação, que antes eram tidas como as grandes vilãs dos sistemas *Web*. Sites e portais que fazem uso do AJAX obtêm reduções drásticas na quantidade de bytes despendidas nas suas larguras de banda. Conseguiram também aumentar a quantidade de acessos a suas páginas depois que começaram a criar interfaces mais agradáveis ao usuário. Diego Eis, em um artigo para o site Tableless, citou o exemplo do portal da ESPN que consumia cerca de 2 Terabytes por dia e conseguiu a façanha de reduzir esse número quase que 90% com a simples adoção dos padrões *Web* de desenvolvimento.

A principal intenção com a publicação deste trabalho foi a de difundir essa técnica tão interessante, demonstrando suas principais aplicações, seus benefícios e suas inovações, além de expor a comunidade de desenvolvedores e usuários da Internet que estamos embarcando em uma nova era, na qual os paradigmas vêm sendo derrubados e novas óticas sobre programação e engenharia de softwares estão eclodindo.

Finalizamos com uma afirmação daquele que, digamos, “inventou” o AJAX, Jesse James Garret (2005): “*Os enormes desafios em criar aplicações AJAX não são técnicas. O coração das tecnologias AJAX é maduro, estável, e bem conhecido. Ao contrário disso, os desafios são para os*

*desenvolvedores dessas aplicações esquecerem o que nós achamos saber sobre as limitações Web, e começarem a imaginar uma longa e rica gama de possibilidades.”.*

## 12. Referências

MARQUEZI, Dagomir. A revolução do AJAX. Info Exame – Ano 21 – nº. 240 – página 42. Março, 2006.

MOREIRA, Daniela. AJAX leva aplicações para muito além do desktop. IDG Now! Disponível em: <<http://old.idgnow.com>> - Acesso em: 22/05/06.

EIS, Diego. Uma das vantagens dos padrões. Tableless. Disponível em: <<http://www.tableless.com.br>> - Acesso em: 15/05/06.

FERREIRA, Elcio. AJAX para quem só ouviu falar. Tableless. Disponível em: <<http://www.tableless.com.br>> - Acesso em: 15/05/06;

FERREIRA, Elcio. AJAX: encarando o mundo real. Tableless. Disponível em: <<http://www.tableless.com.br>> - Acesso em: 15/05/06;

ASLENSON, Ryan and SCHUTTA, Nathaniel T. Foundations of AJAX – [forums.apress.com](http://forums.apress.com)

GARRET, Jesse James. AJAX: A New Approach to Web Applications. Adaptive Path. Disponível em: <<http://www.adaptivepath.com>> - Acesso em: 09/05/06.

LOMANTO, Paulo. AJAX – Olhar crítico sob sua utilização. iMasters. Disponível em: <<http://www.imasters.com.br>> - Acesso em: 22/05/09.

SOARES, Wallace. AJAX – Guia Prático. Editora Érica – 1ª Edição, 2006.

WIKIPÉDIA. AJAX. Disponível em: <<http://www.wikipedia.org>> - Acesso em: 22/05/06.

HOLZNER, Steve. AJAX for dummies – Editora Wiley Publishing Inc. – Copyright 2006.

ULLMAN, Larry. Visual QuickPro Guide PHP and MySQL for dynamic Web sites – Editora Peachpit Press – 2ª Edição, 2005.

## Responsabilidade de autoria

As informações contidas neste artigo são de inteira responsabilidade de seus autores. As opiniões nele emitidas não representam, necessariamente, pontos de vista da Instituição e/ou do Conselho Editorial.