

Web Interativa com

Ajax e PHP

Juliano Niederauer

O que é Ajax?

A palavra Ajax vem da expressão *Asynchronous JavaScript and XML*. É o uso sistemático de JavaScript e XML (entre outras tecnologias) para tornar o navegador mais interativo com o usuário, utilizando-se solicitações assíncronas de informações. Isso quer dizer que podemos utilizar o Ajax para fazer uma solicitação ao servidor web sem que seja necessário recarregar a página que estamos acessando. Veremos a seguir as principais diferenças entre as páginas que utilizam esse recurso e as páginas que fazem uso do modelo tradicional de comunicação com o servidor.

1.1 Modelo “clica e espera” versus modelo interativo

Desde o surgimento da internet, o modelo de interação entre usuário e servidor via HTTP é baseado em um sistema simples de hipertexto. Ou seja, você clica em um link para requisitar um documento e então espera... o servidor responde, processando sua requisição e devolvendo-lhe o documento. Depois de usufruir desse documento, você clica em mais um link e espera... e assim por diante (veja a Figura 1.1).

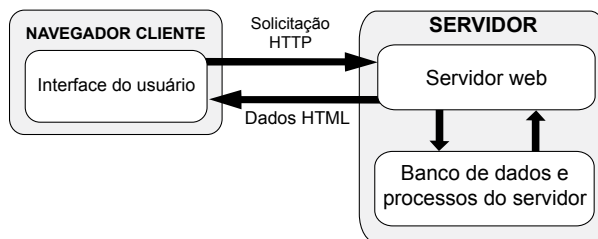


Figura 1.1 – Modelo clássico de aplicação web.

Esse sistema ainda está em uso, porém os designers e desenvolvedores sempre questionaram se esse inconveniente chama-responde teria fim. Por diversos fatores, geralmente eles recebiam um não como resposta.

Mesmo com a evolução dos navegadores (browsers), ainda havia muitas restrições, como, por exemplo, as incompatibilidades com o sistema operacional e com o navegador do usuário, falta de plug-ins instalados na máquina-cliente e baixa qualidade das conexões utilizadas pelos usuários. Diante dessas restrições, os desenvolvedores trabalharam no sentido de melhorar o modelo de interação da web, mesmo dentro do sistema chama-responde. Assim, foram criando novas técnicas para tornar as aplicações mais interessantes e fazer com que elas se tornassem tão úteis quanto suas aplicações equivalentes para desktop.

Podemos dizer que uma dessas técnicas, denominada Ajax, apesar de existir há bastante tempo, só ganhou notoriedade quando aquelas antigas restrições começaram a ser superadas. Assim como ocorreu na época em que os navegadores passaram a suportar tecnologias como CSS e XHTML, a possibilidade de utilizar o Ajax nos principais navegadores foi comemorada pelos desenvolvedores, de modo que o principal beneficiado com a adoção dessa ferramenta será o usuário final, pois a flexibilidade do Ajax irá conferir uma grande agilidade à atualização das informações na web.

O modelo de uma aplicação que utiliza Ajax está representado na Figura 1.2.

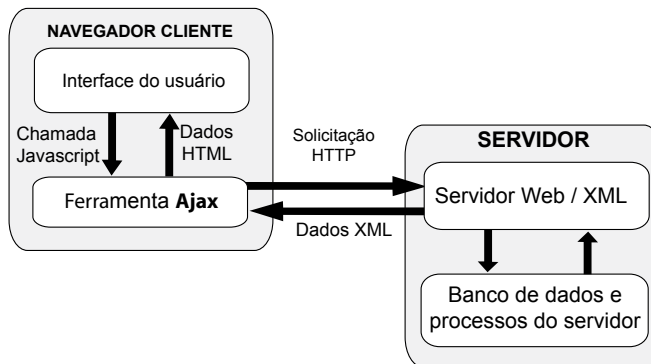


Figura 1.2 – Modelo de aplicação com Ajax.

Veja que, nesse modelo, a interação entre o navegador e o servidor web não ocorre de forma totalmente direta, mas por meio da ferramenta Ajax. E como o Ajax é ativado por uma chamada JavaScript, o usuário pode permanecer visualizando a página normalmente enquanto ocorre a comunicação com o servidor web.

O servidor processa a solicitação do Ajax (por exemplo, realizando alguma pesquisa ou simplesmente atualizando alguma informação no banco de dados) e envia uma resposta. Caso o servidor retorne dados, o Ajax poderá utilizar esses dados para fazer a atualização de apenas uma parte da página que o usuário está visualizando, sem

que seja necessário recarregá-la totalmente. Caso contrário, o usuário também poderá continuar usufruindo normalmente da página, mas ela não sofrerá qualquer alteração visual. Ou seja, é como se a página pudesse “ouvir” e “falar” simultaneamente.

Portanto, é notável que essa forma de interação entre cliente e servidor possibilitará não apenas a agilidade na atualização de informações na web, mas evitará também a retransmissão desnecessária de informações estáticas (códigos, textos, imagens etc.), melhorando o tráfego da rede e a usabilidade das páginas, que ficarão muito mais parecidas com aplicações do que com simples páginas da internet.

Conforme foi comentado no início do livro, o Ajax já existe há bastante tempo, mas sua difusão dependia do momento da web e das nossas habilidades em lidar com as tecnologias envolvidas nesse processo. Finalmente percebemos que a web está passando por esse processo de mudança e já estamos muito mais aptos a trabalhar com as tecnologias disponíveis nessa área. Veremos a seguir, como exemplo, algumas situações onde seria interessante o uso do Ajax.

1.2 Exemplos de uso prático

Existem diversos tipos de situações nas quais o uso do Ajax seria extremamente útil, dentre as quais podemos citar a validação de formulários, atualização de enquetes e de carrinhos de compras (e-commerce), conversação on-line (chats), entre outras que priorizam a atualização de informações em tempo real.

Outro exemplo bastante comum são os sites de relacionamento, os quais vêm utilizando essa técnica para facilitar a classificação de seus usuários. Um dos sites mais conhecidos, a rede social Orkut, pertencente ao Google, utiliza bastante esse recurso. Os desenvolvedores do Orkut criaram uma legenda que permite a cada usuário classificar seus amigos, conforme mostrado na Figura 1.3:



Figura 1.3 – Legenda utilizada.

Assim, cada amigo relacionado a esse usuário poderá ser classificado por ele de acordo com essa legenda, como mostra a Figura 1.4.

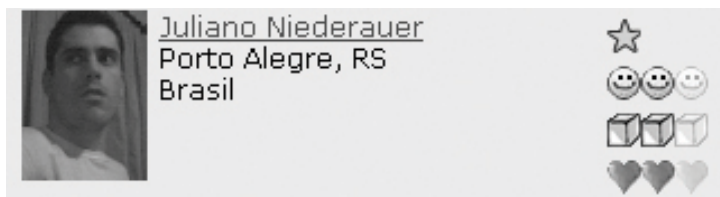


Figura 1.4 – Marcando itens sem mudar de página.

Por exemplo, se você considera um amigo “superconfiável”, pode preencher as três “carinhas”, clicando sobre a terceira delas. Se você é fã de um amigo, pode clicar sobre a estrela e marcá-la. A pergunta nesse caso é a seguinte: “Como fazer para atualizar o banco de dados no servidor a cada vez que o usuário clicar sobre um desses símbolos?”.

Aí é que entra o Ajax. Se estivéssemos utilizando o modelo clássico de aplicação web, ao clicar sobre um desses símbolos seria enviada uma solicitação ao servidor web e toda a página deveria ser recarregada. Com o Ajax, ao clicar sobre um dos símbolos, o navegador se comunica com o servidor por meio de uma chamada JavaScript, permitindo que o usuário continue navegando e visualizando a página como se nada tivesse acontecido. Mais adiante veremos como implementar um sistema semelhante a esse.

Outra aplicação do Ajax é no comércio eletrônico, mais especificamente na parte que envolve o carrinho de compras, ou seja, os produtos que o cliente já selecionou para comprar. Normalmente o frete a ser pago é calculado de acordo com o CEP do cliente (Figura 1.5).


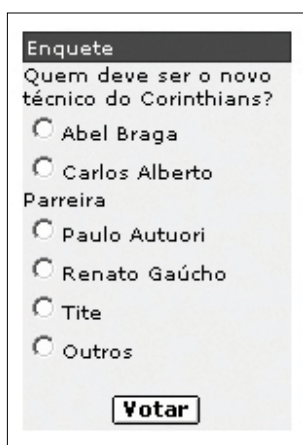
DESCRIÇÃO	QUANTIDADE	OPÇÃO PARA PRESENTE	REMOVER ITEM	COMPRAR DEPOIS	PREÇO UNITÁRIO R\$	TOTAL R\$
 Camisa Seleção Brasileira Oficial Amarela 2006 - Nike G VICTORY SPORTS	<input type="text" value="1"/>	<input type="checkbox"/>	<input type="button" value="X"/>	<input type="button" value="Comprar"/>	R\$ 169,90	R\$ 169,90
Digite o CEP do endereço de entrega para calcular o valor do serviço de entrega: <input type="text"/> - <input type="text"/> <input type="button" value="OK!"/>						
Atenção: O preenchimento deste campo não é obrigatório. Caso você tenha algum Cupom ou Vale Presente coloque seu código no espaço abaixo e em seguida clique em "Calcular" (coloque-o por completo com letras e hifens)						
Digite aqui seu Cupom/Vale-Presente: <input type="text"/>					<input type="button" value="Calcular"/>	
TOTAL						R\$ 169,90
<small>ou sem Juros em até 10X de R\$ 16,99 ou com Juros em até 12X de R\$ 15,09* *Taxa de 0,99% a.m.</small>						

Figura 1.5 – Calculando o frete do produto.

Portanto, o cliente deverá digitar seu CEP, clicar no botão **Ok** e aguardar até que a página seja recarregada com o valor do frete calculado pelo sistema. Se fosse utilizado o Ajax, no momento em que o cliente clicasse no botão **Ok**, haveria uma comunicação via JavaScript com o servidor para obter o valor do frete, que seria mostrado na tela por meio do Ajax, enquanto o resto da página permaneceria inalterado.

O mesmo procedimento poderia ser utilizado no caso de uma enquete, Figura 1.6. No momento em que o usuário respondesse a pergunta, marcando uma das opções, seria atualizada apenas a parte da página que contém essa enquete. Assim, estaríamos evitando o recarregamento desnecessário de muito conteúdo, principalmente se for um grande portal.



Enquete

Quem deve ser o novo técnico do Corinthians?

Abel Braga

Carlos Alberto Parreira

Paulo Autuori

Renato Gaúcho

Tite

Outros

Votar

Figura 1.6 – Respondendo uma enquete.

Outro exemplo seria a validação de formulários de cadastro em tempo real. É claro que se pode fazer uma validação do lado cliente por meio de uma função JavaScript, porém existem informações que precisam ser validadas no lado servidor, como, por exemplo, o nome de usuário (username).

A Figura 1.7 mostra um formulário de cadastro cuja validação é feita do modo tradicional, ou seja, o servidor web retorna uma página de resposta informando que o username escolhido já existe somente após o usuário submeter o formulário inteiro à validação. Se utilizássemos Ajax, poderíamos incluir um botão **Testar** ao lado do campo do username. Assim, ao digitar um nome de usuário, um clique no botão **Testar** ativaria a ferramenta Ajax, que verificaria em tempo real no servidor se o username digitado é válido.

Um script de bate-papo (chat) poderia ser mais facilmente criado com o uso do Ajax (Figura 1.8). Assim, o desenvolvedor evitaria o uso de tecnologias mais complexas, como, por exemplo, a comunicação por sockets.

Erro: Já existe um usuário com esse username. Escolha outro.

Para concluir seu cadastro e ter acesso a todos os nossos serviços basta você preencher os campos abaixo e escolher um nome de usuário e uma senha.

E-mail para contato

Digite seu e-mail:
(por favor digite seu e-mail correto, pois ele será muito importante para que possamos atendê-lo melhor, além de lembrar sua senha em caso de esquecimento)

Dados para acessar o site

Nome de Usuário (ex: andre43, marcosj). Esse é o nome que você vai utilizar para entrar no site. O nome de usuário (username) deve ser uma única palavra, composta somente por letras e/ou números entre 5 e 15 caracteres.

Senha (a senha deve ter entre 5 e 15 caracteres).

Confirme sua senha:

Figura 1.7 – Validação de formulário em tempo real.



Figura 1.8 – Chat.

Além das aplicações citadas, são inúmeras as situações onde poderíamos utilizar o recurso do Ajax. Por exemplo, mais adiante veremos como criar uma tabela on-line editável e um sistema de sugestões para ferramentas de busca.

Portanto, Ajax não é somente um novo modelo, mas também uma iniciativa na construção de aplicações web mais dinâmicas e criativas. O Ajax permite que várias tecnologias trabalhem juntas, cada uma fazendo sua parte, conferindo ao desenvolvedor meios poderosos para a criação de websites.

1.3 Características do Ajax

Pelos exemplos apresentados até o momento, você já pôde perceber que o principal objetivo do Ajax é melhorar a interatividade entre o usuário e o servidor. Isso significa que as páginas devem ser programadas de forma a evitar que os usuários esperem em vão. Não há porque interromper a interação com o usuário a cada vez que a aplicação necessitar de algo do servidor.

Para atingir esse objetivo, o Ajax utiliza algumas tecnologias bastante conhecidas, entre outras que são novidades para muitos desenvolvedores. A Figura 1.9 mostra as tecnologias envolvidas nesse processo, assim como a utilidade de cada uma delas:

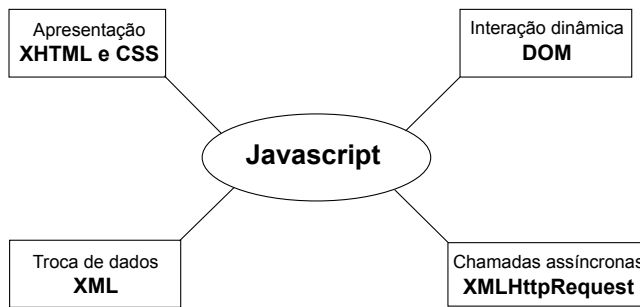


Figura 1.9 – Tecnologias utilizadas pelo Ajax.

Perceba que todo o processo gira em torno da linguagem JavaScript, pois toda a comunicação entre o usuário e o servidor ocorrerá por meio dessa linguagem, a partir da qual o Ajax será ativado. Talvez a maior novidade apresentada na figura seja o XMLHttpRequest, que é justamente a tecnologia que viabiliza o processo inteiro. Trata-se de um objeto JavaScript que torna possível a comunicação assíncrona com o servidor, sem a necessidade de recarregar a página por completo.

Algumas pessoas acham que XMLHttpRequest, objeto integrante da especificação do modelo DOM (Document Object Model), é apenas um outro nome para o Ajax, porém, como você pode ver na Figura 1.9, é apenas uma parte desse processo, que consiste em agregar diversas tecnologias para conferir interatividade à aplicação web.

O modelo DOM consiste em um conjunto de rotinas que permite o acesso e modificação de documentos XML. Portanto, entra no processo do Ajax na parte de interação dinâmica, sendo utilizado no tratamento dos dados retornados pelo servidor, ou seja, é a tecnologia que irá conferir dinamismo, aperfeiçoando a manipulação das informações em questão. Essas informações, por sua vez, poderão ser retornadas pelo servidor no conhecido formato XML (eXtensible Markup Language), um dos mais utilizados para troca de dados.

Por fim, a apresentação desse conteúdo será feita pelas populares linguagens de marcação HTML ou XHTML (eXtensible Hypertext Markup Language) e pelas folhas de estilo CSS (Cascading Style Sheets). No Capítulo 2, será feita uma revisão dos principais conceitos relativos a cada uma dessas tecnologias, para posteriormente criarmos nossa primeira aplicação com o uso do Ajax.

Agora, para complementar a explicação sobre o modelo proposto pelo Ajax, acompanhe a seguir os princípios dessa ferramenta que resumem tudo o que foi explicado até o momento.

1.3.1 O navegador hospeda uma aplicação, não conteúdo

Analisando o funcionamento do modelo “clica e espera”, comentado neste capítulo, percebemos que, em uma aplicação web clássica, o navegador executa o papel de um “terminal bobo”. Ele não sabe nada sobre as ações que o usuário executou até o momento. Todas essas informações são armazenadas em uma sessão do usuário, localizada no servidor web.

Quando o usuário entra no site ou inicia uma sessão, vários objetos são criados no lado servidor. Por exemplo, um site de comércio eletrônico, o carrinho de compras é armazenado nessa sessão. Posteriormente, o servidor web envia ao navegador do usuário a página inicial, que inclui não só códigos HTML, mas também dados do usuário, conteúdos do site e instruções de formatação.

Toda vez que o usuário interage com o site, o navegador envia uma requisição ao servidor, que retorna um outro documento, contendo a mesma mistura de cabeçalhos e dados. Assim, por ser um “terminal bobo”, o que o navegador faz é simplesmente retirar o documento anterior e exibir o novo, mesmo que os dois documentos sejam muito semelhantes. Quando o usuário efetua a saída ou fecha o navegador, a aplicação é finalizada e a sessão é destruída. Esse processo é ilustrado pela Figura 1.10.

A figura mostra o navegador fazendo quatro requisições em seqüência para o servidor web, de modo que, em cada uma delas, o servidor retorna uma página inteira para ser exibida ao usuário. Isso ocorre porque toda a lógica da aplicação está no lado do servidor.

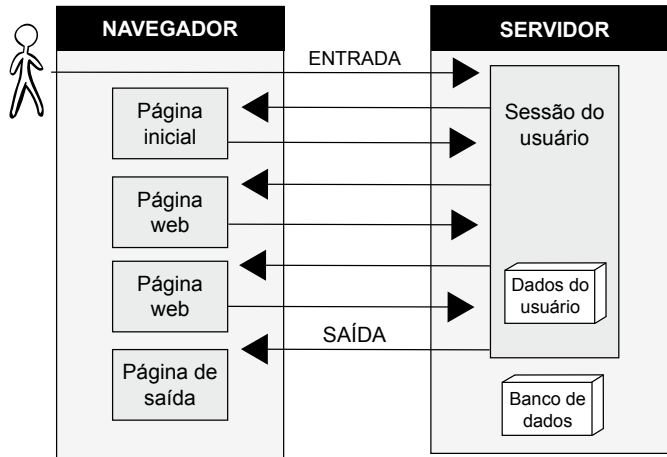


Figura 1.10 – Fluxo clássico da web.

No caso de uma aplicação Ajax, parte da lógica da aplicação é movida para o navegador (por meio da linguagem JavaScript). Nesse novo cenário, quando o usuário entra no site ou inicia uma sessão, o servidor envia ao navegador um documento mais complexo, formado em grande parte por código JavaScript. A ideia é que esse documento se torne a aplicação do cliente, permanecendo com ele por toda a sessão, mesmo que seja necessário alterar consideravelmente sua aparência em determinados momentos.

Essa aplicação terá a capacidade de tratar as informações fornecidas pelo usuário, decidindo se elas devem ser manipuladas no lado cliente ou no lado servidor, ou ainda se fará uma combinação dessas duas alternativas. Por exemplo, se houver necessidade de acessar um banco de dados, a aplicação envia uma requisição ao servidor, caso contrário ela mesma realiza o tratamento das informações.

Isso significa que nesse novo modelo o navegador pode armazenar dados sobre o estado da aplicação, visto que o mesmo documento persiste sobre toda a sessão do usuário. No caso de um web site de comércio eletrônico, por exemplo, o conteúdo de um carrinho de compras poderia ser armazenado no próprio navegador em vez de ser armazenado na sessão do servidor. A Figura 1.11 ilustra como ocorre o fluxo de informações na web por meio do uso do Ajax.

Perceba que, quando o cliente entra no site, o servidor retorna um documento mais complexo, que será a aplicação do cliente. Ao contrário do modelo clássico, a aplicação do cliente é formada por apenas uma página, que envia freqüentes requisições ao servidor. Essas requisições, que são feitas de forma assíncrona pelo objeto XMLHttpRequest, podem ser utilizadas para executar diversas operações, como, por exemplo, consultas e atualizações nos bancos de dados localizados no servidor.

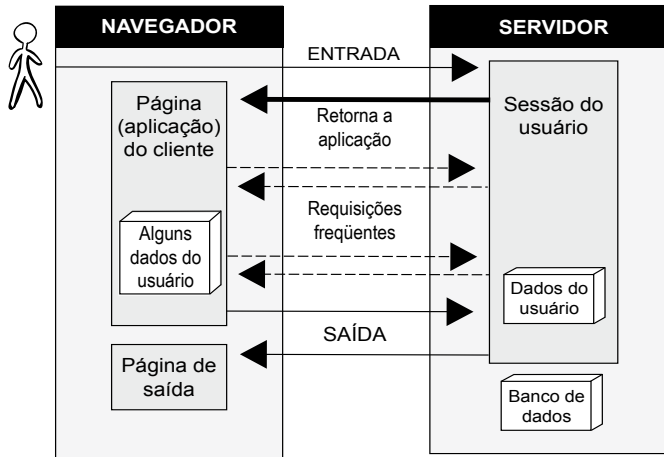


Figura 1.11 – Fluxo da aplicação utilizando Ajax.

Veja também que nesse modelo existe um repositório de dados no próprio navegador, no qual serão armazenadas informações relativas às ações que o usuário executou. Assim, o desenvolvedor pode decidir se acha mais conveniente armazenar uma informação no lado cliente ou na sessão do usuário localizada no servidor.

1.3.2 O servidor fornece dados, não conteúdo

Como pudemos observar no modelo clássico, a cada requisição que o navegador faz, o servidor retorna uma página inteira, ou seja, uma mistura de dados do usuário, conteúdos do site e instruções de formatação. Entretanto, no novo modelo, precisamos de uma resposta imediata que contenha apenas as informações que são de nosso interesse.

Devemos evitar que sejam reenviados todos os conteúdos e códigos que não sofreram qualquer alteração. Por exemplo, quando um cliente digita seu CEP em um carrinho de compras, tudo que precisamos é responder com o valor do frete para o CEP digitado ou informar se ocorreu algum erro, ou seja, muitas vezes precisaremos atualizar apenas uma pequena parte de todo o documento que está sendo visualizado pelo usuário.

Por isso dizemos que, no modelo de aplicação que utiliza Ajax, o servidor deve retornar dados em vez de conteúdos. O gráfico apresentado na Figura 1.12 ilustra o tráfego de informações em uma aplicação web clássica. Cada coluna representa uma página acessada pelo usuário.

Veja que, a cada página acessada, é transferida uma determinada quantidade de informações, envolvendo dados, conteúdos e tags de formatação. No modelo proposto pelo Ajax, a idéia é priorizar a transferência dos dados e reduzir ao máximo a transferência das informações relacionadas à apresentação da página.

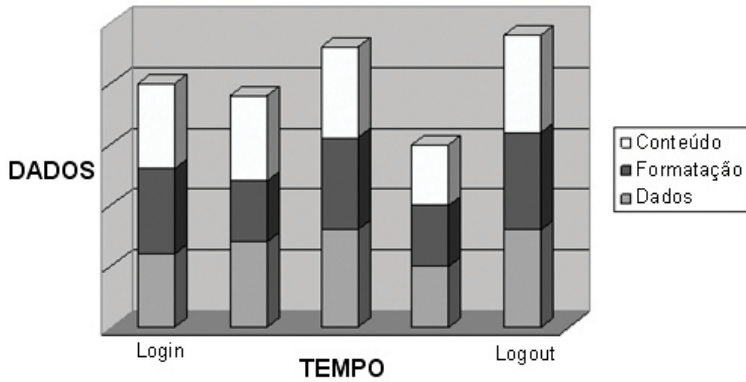


Figura 1.12 – Tráfego de informações em uma aplicação web clássica.

Para isso, quando o usuário entrar no site, seu navegador receberá uma página com uma grande camada de lógica (JavaScript), como mostra a Figura 1.13:

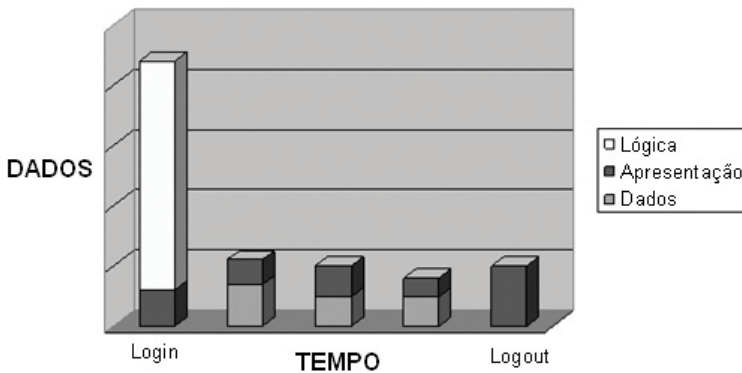


Figura 1.13 – Tráfego de informações em uma aplicação web com Ajax.

Assim, ao longo da navegação do usuário pela aplicação, o que irá trafegar entre o navegador e o servidor são basicamente dados, além de uma quantidade mínima de instruções de apresentação. Ao comparar os dois gráficos apresentados, podemos perceber que o primeiro modelo transfere muito mais informações que o segundo. Essa diferença é ilustrada pela Figura 1.14.

Perceba que em uma aplicação web clássica aumenta consideravelmente a quantidade de informações transferidas ao longo da navegação do usuário. Já a aplicação Ajax não transfere informações desnecessárias, resultando em muito menos tráfego acumulado ao longo da navegação.

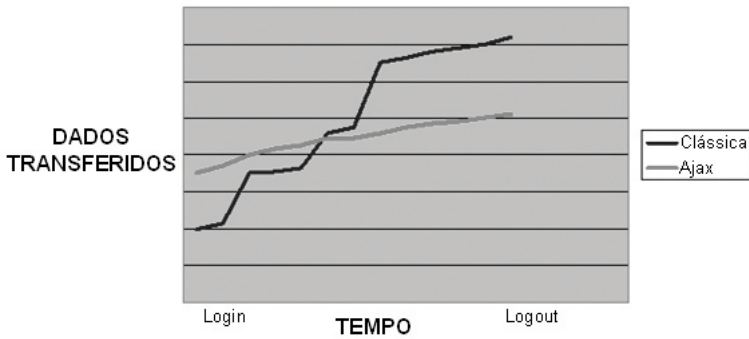


Figura 1.14 – Dados transferidos ao longo do tempo.

Isso ocorre porque, em uma aplicação Ajax, o tráfego tem sua maior intensidade no início, com um documento complexo sendo entregue quando o usuário entra no site. As comunicações subseqüentes com o servidor são muito mais eficientes. Ou seja, à medida que o tempo de interação aumentar, o custo da largura de banda será menor na aplicação Ajax do que na sua aplicação clássica equivalente.

Portanto, para melhorar a interatividade com o usuário, é fundamental que o servidor retorne apenas as informações relevantes à aplicação no momento. Esse retorno pode ser feito de diversas formas, como, por exemplo, pela utilização de um arquivo XML. Mais adiante veremos detalhes sobre as formas de retorno e em quais situações é interessante o uso de cada um delas. De qualquer forma, podemos dizer que qualquer um dos formatos escolhidos será mais eficiente do que a mistura de informações retornadas por uma aplicação web clássica.

1.3.3 Interação flexível entre usuário e aplicação

A estrutura de uma aplicação web clássica não nos permite chegar sequer próximo do nível de interação de uma aplicação para desktop, por exemplo. Ao utilizar JavaScript e folhas de estilo CSS, até conseguimos reproduzir alguns efeitos de interação em um ambiente web, mas mesmo assim a solução ainda pode ser considerada muito rudimentar.

Devemos ter em mente que um navegador web só conhece duas maneiras de enviar entradas de dados para outro computador: hiperlinks e formulários HTML. Quem conhece o ambiente de programação web sabe que esses dois métodos são chamados de GET e POST, respectivamente.

No caso dos hiperlinks, para oferecer uma melhoria na interface, eles podem estar vinculados a imagens e folhas de estilo (CSS), como, por exemplo, para definir efeitos a serem aplicados quando o mouse estiver sobre eles. Em relação aos formulários HTML, eles nos oferecem um subconjunto básico de componentes de interface com o

usuário, como caixas de texto, listas de seleção, botões de rádio e caixas de checagem (checkboxes). Entretanto, esses componentes não são suficientes para garantir uma boa interatividade com o usuário. Não existem, por exemplo, tabelas para edição (grids), caixas de combinação ou controles de seleção em árvores como existem nas aplicações para desktop.

Outra possibilidade que os hiperlinks e formulários nos oferecem é de apontá-los para uma função JavaScript. Essa é uma técnica bastante comum em páginas web para realizar validações de formulários no próprio navegador do usuário, as quais, apesar de incluírem a verificação de campos vazios, valores fora de intervalo e assim por diante, para posteriormente os dados serem submetidos ao servidor, são insuficientes, pois o código JavaScript só roda no navegador, ou seja, antes da página ser enviada. Portanto, depois que a página for enviada, o usuário terá de aguardar uma validação extra no lado servidor, a fim de verificar se não houve uma tentativa de burlar a segurança da aplicação.

No caso do Ajax, a interação com o usuário tende a ser flexível, contínua e a fluir de forma mais amigável. Não será mais necessário aguardar ao clicar em um hiperlink ou submeter um formulário. Conceitos mais sofisticados, como, por exemplo, o “arrastar e soltar”, tornam-se praticáveis, fazendo com que a interface se assemelhe à interface de uma aplicação para desktop. Dessa forma, torna-se possível combinar a interação do usuário e as solicitações ao servidor de maneira mais completa. Portanto, o modelo proposto pelo Ajax habilita o servidor a trabalhar de forma conjunta com o usuário, proporcionando a este uma experiência muito mais agradável.

1.3.4 Disciplina na codificação

Já vimos que uma aplicação Ajax é um código JavaScript complexo que se comunica com o servidor enquanto o usuário continua trabalhando. Embora seja um modelo que descende das aplicações web clássicas, é pequena a similaridade entre essas duas formas de programar. Devemos ter sempre em mente essas diferenças para criar aplicações eficientes.

Codificar utilizando Ajax é bastante diferente de codificar uma aplicação clássica para a web. Com Ajax, o código fornecido no início da aplicação deve ser executado até que ela seja encerrada, sem interrupções. Para atingirmos esse objetivo, devemos escrever códigos de alto desempenho e de fácil manutenção. Normalmente esses códigos serão muito maiores do que outros escritos em aplicações web clássicas. Portanto, é necessária muita disciplina para desenvolver uma aplicação Ajax.

A maioria dos desenvolvedores web costuma utilizar JavaScript apenas em certas ocasiões, devido à limitação que o modelo clássico apresenta. Esse modelo é baseado

em páginas e não possibilita que os scripts permaneçam ao longo do tempo. Assim, a linguagem JavaScript perdeu importância, e muitos desenvolvedores a menosprezam.

Nas aplicações Ajax, o JavaScript ganha muita importância, pois é o centro do processo. E como a base de código dessa linguagem passa a ser maior, tornam-se necessárias boas práticas na construção desse código. É recomendável que o desenvolvedor crie um código organizado e que possa ser entendido facilmente por outros profissionais, principalmente quando houver uma equipe envolvida no projeto. Nesse caso, é importante que a equipe siga padrões de codificação, a fim de tornar a aplicação bastante manutenível.

1.4 O contexto do Ajax na web

Muito se fala sobre o Ajax, mas grande parte dos usuários e desenvolvedores web ainda tem diversas dúvidas sobre o real benefício que esse novo modelo de programação lhes trará.

Os questionamentos envolvem os mais variados aspectos, como, por exemplo, as mudanças na forma de programar, as limitações de acessibilidade por parte do usuário, a comparação com outras tecnologias, a necessidade ou não de utilizar determinados mecanismos, a integração com algumas linguagens de programação e as situações nas quais o uso do Ajax não é recomendado. Começaremos a esclarecer essas questões.

1.4.1 O Ajax não tem dono

Primeiramente, é importante destacar que o Ajax não é propriedade de empresa alguma. Não é algo que você possa baixar da internet. Trata-se apenas de uma abordagem, isto é, uma nova maneira de se pensar a arquitetura de aplicações web que utilizam certas tecnologias.

Algumas pessoas acreditavam que o Ajax foi inventado pela Google, proprietária do site de buscas mais utilizado no mundo. Na verdade, esse comentário surgiu porque a Google foi uma das primeiras empresas a utilizar essa técnica em suas aplicações, como, por exemplo, no Google Maps (Figura 1.15), Gmail, Google News, Orkut, entre outras.

O surgimento do Ajax não está relacionado ao site Google. Várias outras empresas também já utilizam com sucesso esse novo modelo, como, por exemplo, a Amazon.com, a maior livreria virtual do mundo. Mais adiante, veremos o que é necessário tecnicamente para que os usuários e os desenvolvedores possam usufruir do Ajax.

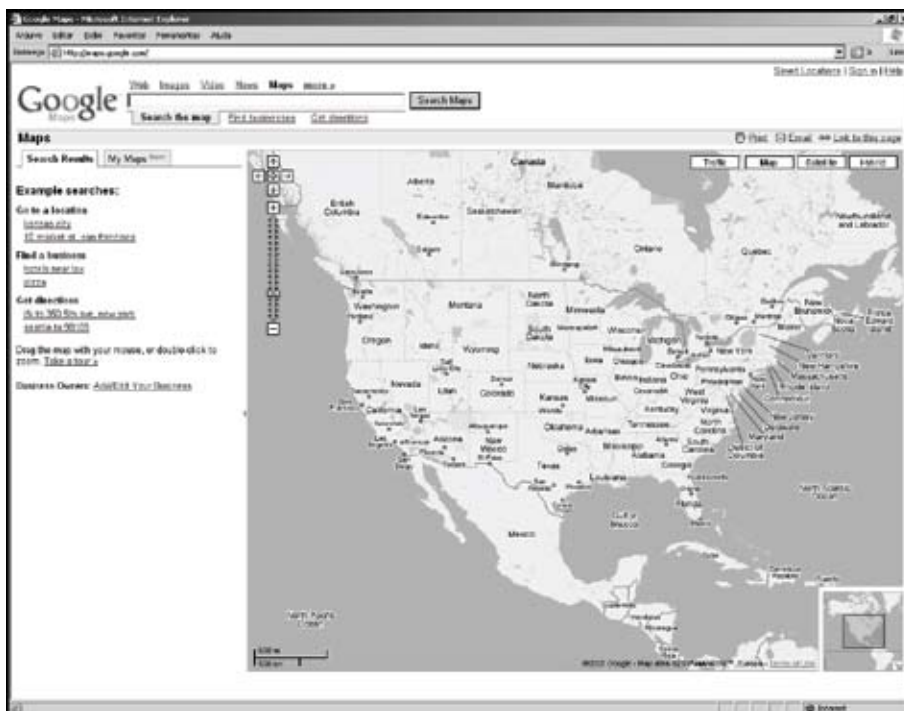


Figura 1.15 – Google Maps utiliza Ajax: alto nível de interação.

1.4.2 Programação, acessibilidade e compatibilidade

Por ser uma nova abordagem de programação, alguns desenvolvedores podem pensar que as aplicações Ajax são mais fáceis de desenvolver do que as aplicações web tradicionais. Na realidade, isso nem sempre ocorre. As aplicações Ajax podem conter códigos JavaScript complexos no lado cliente, o que aumenta a possibilidade de erros (bugs).

Devemos ainda levar em consideração as peculiaridades dos diferentes tipos de navegadores existentes no mercado, como Internet Explorer, Mozilla Firefox, Opera, Safari, entre outros. Nossa aplicação precisa estar preparada para lidar com todos eles. Por exemplo, para começar a usar o objeto XMLHttpRequest, devemos verificar o tipo de navegador que o cliente está utilizando:

```
function IniciaAjax() {
    var req;
    try {
        req = new XMLHttpRequest("Microsoft.XMLHTTP"); // Internet Explorer
    } catch(e) {
        try {
            req = new XMLHttpRequest("Msxml2.XMLHTTP");
        } catch(ex) {
```

```
    try {
        req = new XMLHttpRequest();    // Mozilla, Safari,...
    } catch(exc) {
        alert("Este browser não tem recursos para uso do Ajax!");
        req = null;
    }
}
}
return req;
}
```

Não se preocupe em entender esse programa agora. Esse é um pequeno script utilizado para efetuar uma requisição com o Ajax. Apenas note que são feitas tentativas de criação do objeto para diferentes navegadores (uma para o Internet Explorer, outra para o Mozilla e assim por diante).

Detalhes como esse aumentam a complexidade de programação, e por isso torna-se necessário o uso de plataformas mais adequadas e de boas ferramentas de desenvolvimento, visando a diminuição no índice de erros.

Outro aspecto a ser analisado é a questão da acessibilidade de uma aplicação que utiliza Ajax. Os desenvolvedores se mostram preocupados ao lidar com questões que possam limitar o funcionamento da aplicação. Por exemplo, “será que o botão **Voltar** do navegador deixará de funcionar? E será que a aplicação irá funcionar para usuários que desabilitaram o JavaScript de seus navegadores? Além disso, que tipo de cuidados devemos tomar para garantir a segurança de uma aplicação Ajax?”

É claro que ainda há muito trabalho a ser feito para determinar as limitações do Ajax, porém, à medida que esse novo modelo for ganhando popularidade, os desenvolvedores irão cada vez mais trabalhar em meios de lidar com eventuais dificuldades de uso dessa ferramenta, ou seja, assim como ocorre com qualquer novidade no mundo da programação, o crescimento da comunidade de desenvolvedores desvendará muitas dessas questões, tornando a ferramenta mais eficiente ao longo do tempo.

1.4.3 Comparando Ajax com o Flash

Quando o Ajax começou a ganhar espaço na mídia, muitas pessoas questionaram: “Será que o Ajax irá acabar com o Flash?”. Logicamente a resposta é não. Inclusive essas duas ferramentas podem ser mescladas, como ocorre no site de compartilhamento de imagens flickr (Figura 1.16).

Nesse site, os desenvolvedores utilizaram em conjunto essas duas tecnologias para fazer um *photostream*, isto é, uma interface que possibilita ao usuário navegar em um determinado conjunto de fotos sem que ele precise trocar de página.

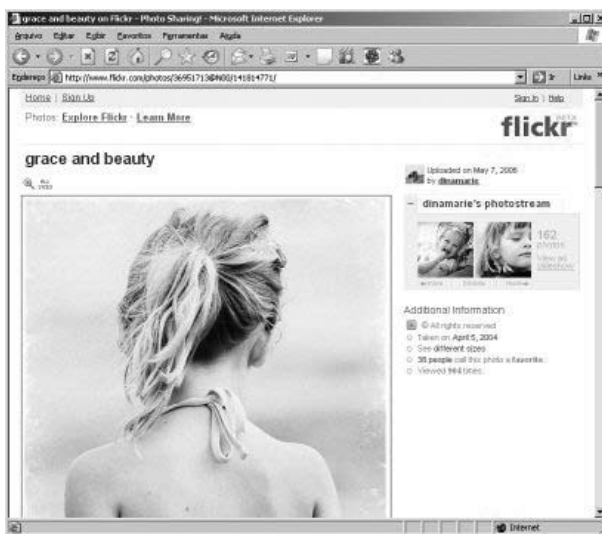


Figura 1.16 – Site flickr mistura as tecnologias Ajax e Flash.

Na verdade, os propósitos do Ajax e do Flash são distintos, embora algumas vezes eles possam ser utilizados para executar tarefas semelhantes. Nesses casos, podemos decidir se a melhor solução para um determinado problema é o Ajax ou se o mais indicado é o Flash, o qual, costuma ser utilizado para criação de animações e exibição de informações ao usuário de uma forma mais atraente em termos visuais. O Ajax, por sua vez, tem como principal objetivo a interação assíncrona com o servidor, possibilitando a alteração de apenas um trecho da página web, sem recarregá-la.

1.4.4 Onde entra o PHP nessa história?

Por ter adquirido este livro, você já deve ter ao menos um conhecimento básico da linguagem PHP, cuja primeira versão surgiu em 1995, quando Rasmus Lerdorf criou para uso pessoal uma ferramenta chamada PHP/FI (Personal Home Page/Forms Interpreter). Ele não imaginava, mas estava criando uma das mais poderosas linguagens para o desenvolvimento de aplicações na web. O PHP (sigla que é um acrônimo recursivo para PHP: Hypertext Preprocessor) conquistou muito espaço nos últimos anos.

Você pode estar se perguntando: “Se o Ajax é baseado em tecnologias como JavaScript e XML, então qual é a importância do PHP nesse processo?” Com certeza a importância é grande. Na verdade faz muito mais sentido utilizar o Ajax em conjunto com uma linguagem de programação do lado servidor do que utilizá-lo isoladamente. Essa linguagem pode ser PHP, ASP ou qualquer outra utilizada para criação de páginas dinâmicas na web. Neste livro, utilizaremos a linguagem PHP, uma das mais populares entre os desenvolvedores.

Como o objetivo do Ajax é poder realizar chamadas assíncronas ao servidor web enquanto o usuário usufrui da aplicação, utilizaremos programas PHP para receber essas chamadas e realizar o processamento das mesmas, retornando as informações necessárias à aplicação do cliente. Por exemplo, se uma aplicação de comércio eletrônico necessitar do preço de algum produto, ela fará uma chamada (por meio do Ajax) para um programa PHP, que será responsável por executar uma consulta no banco de dados, obter esse preço e retorná-lo ao navegador do usuário.

Portanto, em praticamente todos nossos exemplos teremos uma linguagem de programação no lado cliente (JavaScript) e outra no lado servidor (PHP).

1.4.5 Usar XML é interessante, mas não obrigatório

O termo “Ajax”, introduzido por Jesse James Garrett, da Adaptive Path, vem da expressão Asynchronous JavaScript and XML. Apesar do nome, ao tomar conhecimento dessa nova abordagem de programação, você poderá se questionar se realmente é necessário utilizar a tecnologia XML em uma aplicação Ajax.

A verdade é que não somos obrigados a utilizar XML. Observando o diagrama das tecnologias envolvidas no Ajax, apresentado neste capítulo (Figura 1.9), percebemos que o XML entra na parte de “Troca de dados”, ou seja, é a linguagem que o servidor web utiliza para retornar os dados solicitados pela aplicação do usuário. No entanto, esse retorno não precisa ser necessariamente em formato XML; pode inclusive ser um fluxo de texto simples, como usaremos várias vezes no decorrer deste livro.

Então por que o XML é utilizado? Isso ocorre porque a tecnologia XML evolui a passos largos e vem se tornando um padrão para troca de dados, não apenas na web mas também entre aplicações desktop que rodam em diferentes plataformas.

Essa preferência se deve principalmente à facilidade na manipulação dos dados de um arquivo XML, pois é um formato que nos disponibiliza dados estruturados. Dessa forma, podemos utilizar as funções DOM para acessar e manipular esses dados da forma que acharmos conveniente. O mesmo não ocorreria se o retorno fosse feito como um fluxo de texto simples, no qual não há qualquer estruturação dos dados.

Portanto, para troca de informações entre a aplicação e o servidor, sempre que possível, procure fazer uso de marcações XML, pois elas ajudarão na manipulação dos dados e, conseqüentemente, sua aplicação ficará mais organizada, com um código-fonte mais fácil de ser mantido.

1.4.6 Quando não vale a pena usar Ajax

É importante destacar que as aplicações Ajax nem sempre proporcionam uma experiência melhor ao usuário do que as aplicações tradicionais. Na realidade, o Ajax nos fornece

um determinado poder, que consiste na flexibilidade para a criação de interação na web. Contudo, assim como ocorre em nossa vida, quanto mais poder tivermos, maior deve ser nossa precaução em exercê-lo. Por isso, torna-se necessária uma dose de cautela no sentido de não desagradar a experiência do usuário, mas aprimorá-la.

Sendo assim, em que situações o uso do Ajax não é recomendado? Na realidade, devemos utilizar o Ajax sempre que possível. Como já foi dito, não devemos deixar o usuário esperando em vão devido ao recarregamento desnecessário de informações. No entanto, se você for desenvolver uma aplicação onde para cada passo do usuário seja necessário mostrar uma página totalmente diferente da anterior, talvez seja melhor criar uma aplicação no modelo clássico da web.

É importante lembrar que a utilização de Ajax aumenta bastante o tempo de planejamento de um site, já que todo o processo de interação com o usuário deverá ser exaustivamente discutido antes de o desenvolvedor começar a implementação. Isso significa que o uso dessa nova abordagem pode aumentar bastante os custos do projeto.

1.5 Ajax e a Web 2.0

Se você costuma ler com certa frequência publicações da área de tecnologia, provavelmente já deve ter lido a respeito da Web 2.0. Conforme foi comentado no início deste livro, a idéia da Web 2.0 é fazer com que o usuário utilize a web para acessar aplicações, e não simples páginas estáticas com pouca interatividade. Surge então a pergunta: “Qual é o papel do Ajax nesse processo?”

O Ajax surgiu como um protagonista da Web 2.0, pois ele modifica o modo como os navegadores interagem com as informações disponíveis na internet. Portanto, podemos dizer que o Ajax é um dos primeiros passos dessa nova geração da internet.

Vamos imaginar a seguinte situação: você baixa um arquivo CDR (CorelDraw) da internet e tenta abri-lo em seu computador. Se você não tiver o programa necessário para abri-lo, no caso o CorelDraw, o Windows exibirá uma janela para escolha do programa a ser usado, semelhante à janela mostrada na Figura 1.17.

Nessa situação, o usuário estará impossibilitado de abrir o arquivo, pois ele não possui o aplicativo instalado em sua máquina. Então, ele deve obter o CD de instalação do programa e instalá-lo em seu computador, para então finalmente poder usufruir do arquivo.

Na nova geração da internet a situação muda. Não será necessário passar por todo esse processo, pois a Web 2.0 tem como característica executar os softwares diretamente pela internet. Ninguém precisará ter o programa instalado em seu computador. Basta estar on-line. Veja na Tabela 1.1 os dois lados da Web 2.0.



Figura 1.17 – Lista exibida pelo sistema ao não encontrar o aplicativo.

Tabela 1.1 – Benefícios e desvantagens da Web 2.0

Benefícios	Desvantagens
<ul style="list-style-type: none"> • O uso de softwares on-line elimina não só os custos de licença para instalação nas máquinas, mas também o tempo perdido nesse processo de instalação. O usuário poderá pagar apenas de acordo com seu tempo de utilização. • As atualizações dos softwares são facilitadas, pois ocorrem on-line. • Os dados podem ser salvos no servidor e acessados de qualquer computador com conexão à internet. Além disso, eles ficam protegidos em relação a falhas no HD do usuário. 	<ul style="list-style-type: none"> • Utilizar softwares on-line requer uma conexão de alta velocidade. As conexões de banda larga vêm crescendo rapidamente no mundo todo, mas grande parte dos usuários de internet ainda utiliza conexões discadas. • Se a conexão cair, o usuário terá seu trabalho interrompido e possivelmente perderá dados não salvos. • Como os dados são armazenados no servidor, surge a questão da privacidade. Por exemplo, a empresa responsável pelo servidor terá acesso a informações que podem ser confidenciais para o usuário. Além disso, existe o risco de invasão do servidor e roubo de dados.

É notório que o modelo clássico (baseado em páginas) não nos permite reproduzir com eficiência os aplicativos do desktop na web. Por exemplo, é difícil imaginar um software com todos os recursos do CorelDraw rodando no próprio navegador do usuário.

Surgem então os questionamentos: “Será que demorará para a web atingir esse nível e reproduzir fielmente as aplicações desktop? E será que não ficará pesado para o navegador carregar essas aplicações?”. São perguntas difíceis de serem respondidas, mas sabemos que o Ajax já está acelerando esse processo. Quanto ao carregamento da aplicação, a idéia é carregar no navegador apenas os recursos necessários para as ações

que o usuário está executando. Os demais recursos seriam carregados posteriormente por meio das chamadas assíncronas do Ajax ao servidor.

Conhecendo o conceito de Web 2.0, fica clara a importância do Ajax nesse processo, pois ele diminui muito a distância entre as aplicações para desktop e as aplicações para a web, para a qual muitos aplicativos que foram consagrados no desktop já estão migrando com muito sucesso. Já surgiram inclusive boatos de que algumas empresas estariam desenvolvendo um conjunto de aplicativos on-line (editores de texto, planilhas eletrônicas etc.) para concorrer com o Office da Microsoft.

Na realidade, desde que o Google lançou uma aplicação chamada Gmail, a comunidade web percebeu que era possível fazer o navegador se comportar como uma aplicação “de verdade”. O Gmail chegou e conquistou seu espaço, mesmo em um mercado saturado como o de e-mails na web (webmail). A partir daí, o usuário começou a se ver livre não só do sistema operacional, mas também do conceito de “meu computador”.

Previsões à parte, é muito difícil dizer se realmente o Ajax dominará a Internet do futuro ou se surgirá um modelo inovador para substituí-lo, porém a idéia da Web 2.0 começa a tomar forma a partir do Ajax, e os desenvolvedores devem estar preparados para criar aplicações condizentes com a nova geração da web.

1.6 Requisitos e instalações de softwares

A grande vantagem das aplicações Ajax é que elas rodam no próprio navegador web. Isso significa que, para usufruir dessas aplicações, basta que o usuário possua uma versão mais recente (após 2001) de um dos principais navegadores utilizados no mercado, como por exemplo:

- **Mozilla Firefox:** <http://www.mozilla.com/firefox/>
- **Internet Explorer 5+:** <http://www.microsoft.com/ie/>
- **Opera:** <http://www.opera.com/>
- **Konqueror:** <http://www.konqueror.org/>
- **Safari:** <http://www.apple.com/safari/>

Para testarmos os exemplos apresentados neste livro, serão utilizados os seguintes softwares:

- **No lado cliente:** Internet Explorer (navegador).
- **No lado servidor:** PHP (linguagem de programação), Apache (servidor web), MySQL (banco de dados).

Se você não tiver à disposição um servidor contendo os softwares necessários, pode instalar o pacote PHP + Apache + MySQL em sua máquina, fazendo-a executar o papel de servidor e possibilitando a você testar os exemplos do livro.

No Windows, uma boa opção para instalar esses três softwares é obter o pacote EasyPHP, disponível para download em <http://www.easyphp.org/>. Com esse pacote, podem ser instalados o PHP, o Apache e o MySQL, a partir de um único arquivo de instalação.

No entanto, se você for um usuário um pouco mais experiente, é recomendável instalar individualmente cada um desses softwares, visto que nem sempre o EasyPHP contém as versões mais atuais de cada um deles. Em nosso caso, faremos manualmente a instalação das seguintes versões:

- PHP 5, disponível em <http://www.php.net>.
- Apache 2.0, disponível em <http://httpd.apache.org>.
- MySQL 5, disponível em <http://www.mysql.com>.

Se você resolver também fazer a instalação e a configuração manual dos softwares, o procedimento para o Windows é o seguinte:

- 1. Instalação do PHP** – na seção de downloads do site do PHP, faça o download da versão para Windows do PHP 5. Você pode baixar o pacote completo, disponível em formato ZIP. Em seguida, descompacte o arquivo no seu computador (normalmente cria-se a pasta `c:\PHP` para isso). Renomeie o arquivo `php.ini-dist` para `php.ini`, a fim de que o PHP possa localizar seu arquivo de configuração. Por fim, copie o arquivo `php5ts.dll` para a pasta de sistema do seu Windows (no Windows XP, a pasta é `C:\WINDOWS\system32`), pois o PHP precisará encontrar esse arquivo. Faça o mesmo para o arquivo `libmysql.dll`, que será usado posteriormente no acesso ao MySQL.
- 2. Instalação do Apache** – no site do Apache, na seção de downloads, há pastas com várias versões do software. No caso do Windows, uma boa alternativa é acessar uma pasta chamada `win32`, localizada dentro da pasta `binaries`. Nela, você encontrará algumas distribuições binárias do produto, ou seja, você poderá baixar um arquivo (normalmente com extensão `.msi`) e executá-lo para que o Apache seja automaticamente instalado em sua máquina, como mostrado na Figura 1.18.

Se durante o processo de instalação for solicitado o nome do servidor (server name), você pode digitar `localhost`. Se ocorrer algum erro ao iniciar o Apache, será exibida uma mensagem indicando onde está o problema no arquivo de configuração `httpd.conf`, localizado no diretório `conf` do Apache. Algumas vezes, esse erro pode ocorrer em virtude da falta de alguma informação, como, por exemplo, o e-mail do administrador. Se ocorrer, basta fornecer as informações solicitadas

para que o Apache seja iniciado corretamente. Para verificar se o Apache está rodando, abra seu navegador e digite na barra de endereços:

http://localhost/

Deve aparecer uma página indicando que o Apache está em execução (Figura 1.19).



Figura 1.18 – Download do arquivo de instalação do Apache.



Figura 1.19 – Testando o funcionamento do Apache.

Para que suas aplicações possam ser acessadas pelo navegador, elas devem ser colocadas na pasta raiz do servidor (`DocumentRoot`), por padrão chamada de `htdocs`. Por exemplo:

```
C:\Arquivos de programas\Apache Group\Apache2\htdocs
```

A pasta raiz pode ser modificada por meio da alteração do valor da diretiva `DocumentRoot` do arquivo de configuração `httpd.conf`.

- 3. Instalação do MySQL** – faça o download do MySQL 5 no site oficial do produto. Assim como no Apache, você também poderá obter um arquivo binário para Windows, com extensão `msi`. Basta fazer o download desse arquivo e executá-lo para que o MySQL seja instalado.
- 4. Configuração do PHP no Apache** – agora que você já tem os três softwares necessários para montar seu servidor, falta apenas configurá-los a fim de que funcionem em conjunto. Para habilitar o PHP como um módulo do Apache, você precisará acrescentar algumas linhas no arquivo `httpd.conf`. Essas linhas podem ser encontradas no arquivo `install.txt` que acompanha a distribuição do PHP. Por exemplo, para instalar o PHP como módulo do Apache 2.0, devem ser acrescentadas as seguintes linhas:

```
LoadModule php5_module "C:/PHP/php5apache2.d11"  
AddType application/x-httpd-php .php  
PHPIIniDir "C:/PHP"
```

A primeira linha carrega o módulo do PHP, a segunda faz com que o Apache reconheça os scripts com extensão `.php`, e a terceira indica a localização do arquivo de configuração `php.ini`. Consulte o arquivo `install.txt` para orientações quanto a outras versões. Após incluir as linhas, reinicie o Apache para que elas tenham efeito.

- 5. Configuração do MySQL no PHP** – para finalizar, precisamos habilitar a extensão do MySQL no arquivo de configuração do PHP, o `php.ini`. Neste livro, utilizaremos a extensão `mysql_i`, visto que nossa versão do MySQL é superior a 4.1, e, portanto, a antiga extensão do MySQL não funcionará. A linha que habilita essa extensão é:

```
extension=mysql_i.d11
```

Se essa linha estiver comentada, retire o comentário. Caso ela não exista, deve ser inserida. Se sua versão do MySQL for anterior a 4.1, habilite o arquivo `php_mysql.d11` em vez do `php_mysql_i.d11`. Verifique ainda se a diretiva `extension_dir` está apontando para o diretório correto das extensões do PHP (normalmente chamado de “`ext`”). Por exemplo:

```
extension_dir = "C:/PHP/ext/"
```