

JQUERY: fundamentos teóricos

ANDERSON FILGUEIRA FARIAS¹

RESUMO

As reflexões deste artigo centram-se em uma análise introdutória, de cunho teórico, de uma pequena e extensível biblioteca JavaScript denominada jQuery e sua relevância para o desenvolvimento de sistemas para internet. Inicialmente, serão abordadas a origem e a importância da linguagem JavaScript, focando sua característica de providenciar dinamismo nas páginas da internet, aspecto muito valorizado atualmente no cenário web 2.0, para em seguida serem observadas as dificuldades enfrentadas pelos programadores que utilizam apenas JavaScript “puro” e como isto é contornado usando jQuery, que possui uma baixa curva de aprendizado, e fornece um considerável ganho de produtividade, por meio de recursos, como por exemplo, a manipulação de objetos DOM, tratamento de eventos, e a característica de estender suas funcionalidades por meio de plugins para solucionar problemas mais específicos. Contudo, apesar de todas as vantagens citadas, será demonstrado que só é possível aproveitar todo o potencial da jQuery se a mesma for utilizada em conjunto com as melhores práticas de programação.

Palavras-chave: DOM. JavaScript. jQuery. Melhores práticas de programação.

ABSTRACT

This article focus on a small and extensible JavaScript library called jQuery. This library is studied from a theoretical perspective and we cover many aspects of its usage, including its relevance to the development of Internet systems. Initially, the origin and importance of the JavaScript language are discussed, including its dynamic aspects when manipulating web pages, something of extreme importance on today's Web 2.0 scenario. Then we aim on the difficulties faced by developers when using "pure" JavaScript and how jQuery handle this issues. JQuery has a low learning curve and provides us with considerable gain in productivity. Among its main resources we would highlight: DOM objects manipulation, event handling and plugins that allow us to extend its functionalities in order to solve specific problems. Besides JQuery's many advantages, we show that its full potencial can only be achieved by using it in conjunction with good programming practices.

Keywords: Best programming practices. Dom. JavaScript. jQuery..

¹ FARIAS, Anderson Filgueira. Tecnólogo em Informática e Sistemas para Internet pela Universidade Tiradentes; Pós-graduando em Sistema de Informação para Web pela Faculdade de Negócio de Sergipe(FANESE)

INTRODUÇÃO

Javascript é uma linguagem de script dinâmica que suporta a construção de objetos, baseada em protótipos, desenvolvida pela empresa Netscape, usada em milhões de páginas da internet e servidores de aplicações do mundo inteiro. Ao contrário das concepções populares, JavaScript não é "Java Interpretado", ou qualquer outro subtipo da linguagem Java; esta confusão acontece, além da semelhança entre os nomes, pelo fato da sintaxe básica do JavaScript ser intencionalmente similar tanto a Java quanto a C++ para reduzir o número de novos conceitos requeridos para aprender a linguagem.

Javascript além de ser uma linguagem de programação completa, também é uma linguagem envolvida em muitas polêmicas, que geraram (e ainda geram) muitos debates acalorados, principalmente de programadores que simplesmente a desdenham ao ponto de considerá-la uma "linguagem de programação de brinquedo". Entre as alegações está o fato da mesma não ser compilada, problemas referentes às questões de segurança e se ela seria realmente uma linguagem Orientada a Objetos. As polêmicas são tantas que Douglas Crockford, arquiteto sênior da empresa Yahoo! e também autor do livro *O Melhor do Javascript*², chegou a criar um artigo intitulado "JavaScript: A Menos Entendida Linguagem de Programação do Globo!"

Polêmicas a parte, é inegável a popularidade da linguagem JavaScript, geralmente usada para validação de formulários e para adicionar interatividade e efeitos dinâmicos nas páginas da web, ganhando um destaque ainda maior com o estabelecimento do conjunto de tecnologias conhecidas como AJAX³, que acabou sendo um dos pilares da Web 2.0⁴.

Porém, nem tudo é perfeito, desenvolvedores ainda estão envolvidos com a dificuldade de ter que lidar com navegadores diferentes, utilizando "resmas" de linha de código, sim, nestas circunstâncias, nos dias atuais, seria dispendioso trabalhar

² CROCKFORD, Douglas. **O Melhor do JavaScript**. 1. ed. Rio de Janeiro: Alta Books, 2008.

³ AJAX (acrônimo em língua inglesa de Asynchronous Javascript And XML). AJAX não é uma nova linguagem de programação, mas um novo jeito de utilizar padrões já existentes. É a arte de trocar dados com um servidor, e atualizar partes de uma página de internet sem recarregar a mesma. W3SCHOOLS. **AJAX Tutorial**. Disponível em: < <http://www.w3schools.com/ajax/default.asp> > Acesso em: 23/07/2010.

⁴ "Web 2.0 é um buzz word que define conteúdo gerado pelo usuário e com foco no compartilhamento de informações. Tudo regado a AJAX." VIEIRA, Nando. **Conceituando o que é Web 2.0**. Disponível em: < <http://web2.0br.com.br/conceito-web20/> > Acesso em: 22/07/2010.

com JavaScript puro para a programação do DOM⁵ e não usar algum tipo de biblioteca que contribua para contornar as peculiaridades específicas de cada navegador e forneça funcionalidades que tornem mais fácil trabalhar em um agnóstico ambiente de navegadores.

O presente artigo tem como objetivo principal levar até o leitor os principais conceitos referente à biblioteca jQuery, e o seu potencial no desenvolvimento de sistemas para internet; como objetivo secundário tentar contribuir a este campo de estudos que sofre com a carência de material correlato.

O artigo está estruturado em outros sete tópicos. O primeiro, intitulado jQuery, descreve brevemente o objeto de estudo, em seguida são apresentadas as principais características da jQuery, entre elas o conceito de JavaScript Não Obstrutivo, e a Função Ready; IDEs Compatíveis lista algumas ferramentas muito úteis para aumentar a produtividade; em outras bibliotecas JavaScript, é feita uma breve comparação com os concorrentes da jQuery. Já na última seção, são apresentadas ao leitor, as Considerações Finais deste trabalho.

⁵ “O Document Object Model é uma API para documentos HTML e XML. Isto proporciona uma representação estrutural de um documento, habilitando você a modificar seu conteúdo e apresentação visual. Essencialmente, isto conecta páginas web à scripts ou linguagens de programação.” SVOLENSKI, Matheus Smythe. **Sobre o Document Object Model**. Disponível em: < https://developer.mozilla.org/pt/Sobre_o_Document_Object_Model> Acesso em: 22/07/2010.

JQUERY

A jQuery é uma pequena biblioteca JavaScript criada pelo americano John Resig e apresentada ao público em 2006. Ela foi desenvolvida com o intuito de permitir àqueles que usam essa linguagem, sejam programadores ou/e designers, codificar JavaScript de maneira mais fácil e eficiente, dando ênfase em um pequeno e simples código que servisse a todos as aplicações práticas que os desenvolvedores web lidam no dia a dia.

Esta biblioteca serve para adicionar interatividade e dinamismo, aumentando a usabilidade e o design, adicionando eventos visuais e animações, permitindo acessar e manipular o DOM, buscar informações no servidor sem necessidade de recarregar a página, alterar conteúdo e modificar a apresentação da página web.

A jQuery é distribuída sob uma das seguintes licenças: a MIT(Massachusetts Institute of Technology)⁶ e a GPL (GNU General Public License)⁷. O usuário é livre para escolher a que mais se adéqua ao seu projeto.

Sendo tão fácil de implementar, jQuery está popularmente crescendo, não só em pequenos projetos, mas em serviços de grande popularidade. O site msnbc.com (o site de notícias da família NBC News), as redes sociais Facebook e Twitter, por exemplo, usam efeitos jQuery. No caso doTwitter, quando você posta um novo tweet a página não recarrega, em vez disso, o código por trás da página chama uma função jQuery para fazer seu novo tweet aparecer e todos os seus outros tweets na sua pagina sejam movidos pra baixo. Outro recurso jQuery no Twitter é a notificação que você recebe quando um novo tweet chega.

Principais características da jQuery

jQuery fornece várias funcionalidades, mas aqui está uma lista das principais características:

- **Seletores de elementos DOM** - Apesar de soar redundante, seletores jQuery permitem selecionar elementos DOM para que possa aplicar-lhes

⁶ Github Social Coding. **MIT-LICENSE.txt**. Disponível em: <<http://github.com/jquery/jquery/blob/master/MIT-LICENSE.txt>> Acesso em: 15/8/ 2010.

⁷ Github Social Coding. **MIT-LICENSE.txt**. Disponível em: <<http://github.com/jquery/jquery/blob/master/GPL-LICENSE.txt>> Acesso em: 15/8/ 2010.

funcionalidade por meio de algum de seus métodos. JQuery usa uma sintaxe CSS 3.0 (e mais algumas extensões) para selecionar um ou vários elementos em um documento. Usando a sintaxe CSS você pode selecionar os elementos por id, classe CSS, filtros de atributo, por relação com outro elemento e até mesmo filtrar condições que podem estar relacionadas. Um exemplo simples: é trivial a tarefa de selecionar todas as segundas colunas dos elementos TD de uma tabela através um seletor simples como este: `$("#gdEntries td:nth-child(2)")` (trecho de código jQuery). Você pode ver uma relação básica de seletores no apêndice A

- **O objeto jQuery: O conjunto encapsulado**⁸ - O uso dos seletores retornam um objeto jQuery que é conhecido como conjunto encapsulado, que é um vetor que contém cada um dos elementos DOM selecionados. Você pode iterar o conjunto encapsulado como se fosse um vetor ou acessar os elementos individualmente, por meio do indexador (`$(sel)[0]` , por exemplo). O mais importante é que você também pode aplicar funções jQuery em todos os elementos selecionados.
- **Operações encadeadas** - No JavaScript tradicional não existe o encadeamento, o que obriga o uso de múltiplas declarações para se conseguir um efeito que em jQuery conseguimos com uma linha de código apenas. O poder real do conjunto encapsulado vem da capacidade de aplicar operações jQuery contra todos os elementos DOM selecionados simultaneamente em lote. Você pode encadear juntos vários métodos em um único comando. Efetivamente isto significa que você pode selecionar uma vez, e operar muitas vezes contra o mesmo objeto e até filtrar ou expandir o conjunto encapsulado com métodos como `.find()`, `.filter()` or `.add()`. Ex: `$("#div.notLongForThisWorld").fadeOut().addClass("removed");`
- **Manuseio simplificado de evento** - Muito do que é feito no código JavaScript, da manipulação DOM às chamadas AJAX, é tratado de forma assíncrona e, infelizmente, as implementações de DOM para manipulação de eventos variam consideravelmente entre os navegadores. JQuery fornece um mecanismo fácil para vincular e desvincular eventos e fornecer um modelo de evento normalizado para todos os navegadores suportados que torna muito

⁸ “Set wrapped”, no original em inglês. Não confundir com o conceito de encapsulamento pertencente ao paradigma da Programação Orientada a Objetos.

fácil a manipulação de eventos. Todos os eventos são chamados no contexto do elemento que gerou o evento e eles recebem uma correção e um objeto de evento normalizado do browser que é consistente.

- **Tamanho pequeno** - JQuery é uma biblioteca bastante compacta, na sua versão 1.4.2 seu tamanho era de 71KB/160KB, mas você tem acesso a seletores, várias operações que podem ser executadas sobre o conjunto encapsulado, normalização DOM para a maioria dos browsers, funcionalidade Ajax, uma série de funções utilitárias para a manipulação do objeto array e um número de efeitos básicos funcionais. Dada a elevada utilidade da jQuery, estes 71KB/160KB de script baixado fornece um enorme custo benefício.
- **Extensibilidade fácil, através de Plugin** - JQuery é pequena e bem focada no fornecimento de funcionalidades básicas e nada mais. Para todo o resto jQuery oferece uma API de plugin que permite estender o núcleo das operações do objeto jQuery simplesmente criando uma função e passando o invólucro jQuery definido como um parâmetro. Desta forma, plugins recebem o conjunto encapsulado e podem operar sobre ele e participar no encadeamento jQuery. Se você precisar de algumas funcionalidades especiais, as chances são de que um plugin já existe com a funcionalidade que você está procurando (vide Anexo B). E se isso não acontecer, é fácil o suficiente criá-la você mesmo com a ajuda da jQuery, ou outro plugin como base.

JavaScript Não Obstrutivo

JQuery suporta o conceito de JavaScript não obstrutivo (em inglês “Unobtrusive JavaScript” ou “Unobtrusive DOM Scripting”, dependendo do autor), resumidamente é o conceito de que a página web consiga ficar o mais funcional possível, mesmo que o JavaScript falhe ou esteja desabilitado. Entre as razões do JavaScript estar desabilitado, temos:

- Há empresas que bloqueiam o JavaScript através do firewall;
- Alguns usuários podem utilizar nos seus navegadores alguma extensão para se protegerem de vulnerabilidades como XSS e CSRF. Ex: Firefox NoScript;
- Muitos dispositivos móveis ignoram completamente JavaScript.

Segundo Alex Holt(2008), os passos a serem seguidos para conseguirmos alcançar este conceito, seriam:

- **Separar a funcionalidade do JavaScript** - Separe a funcionalidade do JavaScript em uma "camada de comportamentos", de modo que esteja separada e independente do (X) HTML e CSS. (X) HTML é a marcação, CSS a apresentação e Javascript a camada de comportamento. Isto significa armazenar TODOS os códigos JavaScript em arquivos de script externos e construir páginas que não dependam de JavaScript para serem utilizáveis. Nunca incluir eventos JavaScript embutidos nos atributos. Todos os comportamentos JavaScript devem ser incluídos nos arquivos de script externos e vinculado ao documento com uma tag <script> na head da página.
- **Nunca depender de Javascript** - Para ser verdadeiramente não obstrutivo, um programador nunca deve depender de suporte Javascript para entregar conteúdo ou informação. É bom usar Javascript para melhorar a informação, torná-la mais bonita, ou mais interativa, mas nunca presumir que o navegador do usuário vai ter o JavaScript habilitado. Esta regra de ouro pode, de fato, ser aplicado a qualquer tecnologia de terceiros, como Flash ou Java. Ter todo o código JavaScript residindo em um arquivo de script externo, e a página sendo apresentada no padrão (X) HTML, significa que mesmo que o JavaScript não seja executado, a página ainda será 100% pura semântica (X) HTML .
- **Semântica e marcação acessível vem em primeiro lugar** - É muito importante que a marcação (X) HTML seja semanticamente estruturada (está fora do escopo deste artigo explicar o porquê). A regra geral, apesar de nem sempre ser verdade, é que, se a marcação da página estiver semanticamente estruturada, a mesma deverá ser acessível a uma ampla gama de dispositivos.

Função Ready

Tradicionalmente, eventos JavaScript são vinculados a um documento utilizando-se um atributo **onload** na tag **<body>** da página. Esqueça essa prática, jQuery fornece um utilitário especial sobre o objeto documento, chamado de **ready**,

que permite executar o código apenas depois que o DOM acabou de ser carregado. Essa é a chave para o JavaScript não obstrutivo, pois permite separar completamente o seu código JavaScript da sua marcação. Usando **\$(document).ready ()**, pode-se enfileirar uma série de eventos e mandá-los executar após o DOM ser inicializado. Isso significa que é possível criar efeitos para todas as suas páginas, sem alterar a marcação dos elementos em questão. Para demonstrar esta funcionalidade, na Tabela 1 recriou-se a introdução padrão para Javascript: uma mensagem de alerta "Olá Mundo". A marcação a seguir mostra como poderia ser executado um "Olá Mundo" sem e com jQuery:

Tabela 1: Exemplo da função ready()

Trecho de código sem jQuery	Trecho de código com jQuery
<pre><script language="javascript"> alert('Hello World'); </script></pre>	<pre>\$(document).ready(function () (alert ('Olá Mundo');));</pre>

Fonte: Tabela construída pelo autor(2010)

A função `$(document).ready ()` recebe outra função como argumento. (Neste caso, uma função anônima é criada inline, uma técnica que é usada em toda a documentação da jQuery.) A função passada para `$(document).ready ()` é chamado após o DOM ter terminado de carregar e executar o código dentro da função , neste caso, chamar a função `alert` do JavaScript.

IDEs Compatíveis

Apesar de todas as vantagens da jQuery, os projetos que a utilizam teriam uma baixa produtividade se os desenvolvedores programassem utilizando um simples editor de texto. Algumas das IDEs⁹ mais populares do momento já possuem

⁹ “Um ambiente de desenvolvimento integrado (IDE) é uma configuração de programação única na qual você tem todas as ferramentas necessárias à sua disposição. Geralmente, um editor de código-fonte que é cercado por um compilador, um depurador e outras ferramentas de desenvolvimento. Alguns ambientes de desenvolvimento integrado permitem uma certa flexibilidade, assim o desenvolvedor pode personalizar seu ambiente de trabalho de modo a obter maior produtividade.” **Ambiente de Desenvolvimento Integrados (IDE)**. Disponível em: <<http://homepages.dcc.ufmg.br/~mlbc/cursos/internet/java/>> Acesso em: 10/11/2010.

suporte à jQuery. A Tabela 2 descreve em quais IDEs o suporte é nativo e as que necessitam de instalação de plugins para funcionar, neste caso citando o nome dos mesmos. Já a Tabela 3 indica o endereço eletrônico onde cada Plugin pode ser baixado; e na Figura 1 temos um exemplo prático de edição de uma página HTML com trechos de código da jQuery, tudo isso dentro da IDE Eclipse, e graças à utilização do plugin jQueryWTP.

Tabela 2: Exemplos de IDEs que suportam a jQuery

IDE	Fabricante	Suporte Nativo?	Plugin(s)
Visual Studio 2008	Microsoft	NÃO	VS 2008 Patch KB958502 to Support "-vsdoc.js" Intellisense Files.
Visual Studio 2010	Microsoft	SIM	-----
Eclipse	Eclipse Foundation	NÃO	jQueryWTP
Dreamwaver	Abode	NÃO	jQuery API extension for Dreamweaver

Fonte: Tabela construída pelo autor(2010)

Tabela 3: Links dos plugins mencionados na Tabela 2

Plugin	Link
VS 2008 Patch KB958502	< http://code.msdn.microsoft.com/KB958502/Release/ProjectReleases.aspx?ReleaseId=1736 >
jQueryWTP	< http://sourceforge.net/project/showfiles.php?group_id=202840&package_id=241633&release_id=631407 >
jQuery API extension for Dreamweaver	< http://xtnd.us/files/jquery_API.mxp >

Fonte: Tabela construída pelo autor(2010)

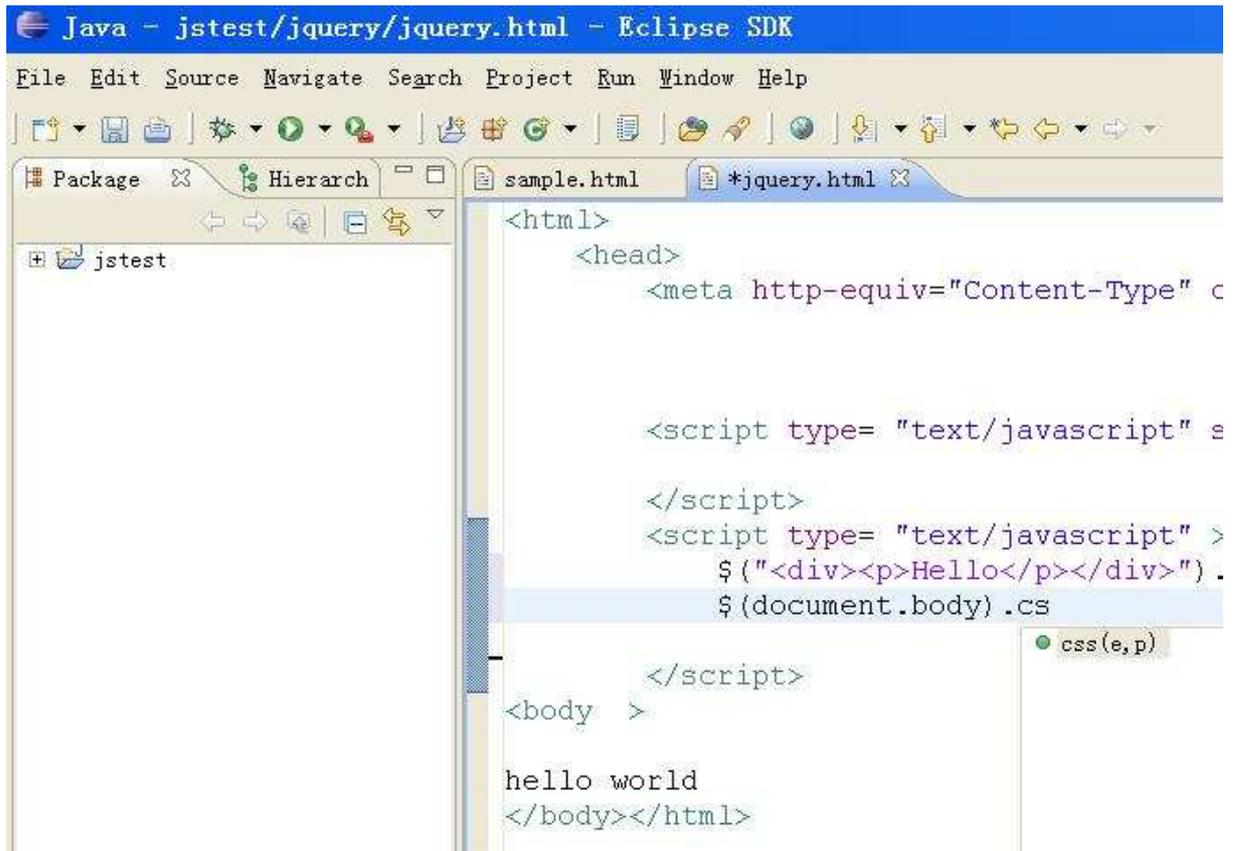


Figura 1: Eclipse utilizando o Plugin jQueryWTP

Fonte: jQueryWTP. Disponível em: <<http://www.langtags.com/jquerywtp/index.html>> Acesso em: 11/08/2010.

Outras bibliotecas JavaScript

jQuery não é a única biblioteca JavaScript no mercado, havendo outras opções bastante divulgadas, como Mootools, Prototype, DOJO, e etc.. Não é o objetivo deste trabalho, detalhar as peculiaridades de cada biblioteca e compará-las com as da jQuery, sendo que tal tarefa demandaria um artigo a parte, mas tentou-se fazer um apanhado geral.

Entre as comparações, temos a Tabela 4 que confronta várias características das bibliotecas Prototype, jQuery, Yui, ExtJS e MooTolls, como licença de distribuição, compatibilidade de navegadores, suporte à AJAX e etc.. Já ATES(2006) analisou o Prototype, Dojo e jQuery, concluindo que esta é uma mistura dos outros dois, posicionando-se precisamente entre os dois. Segundo ele, a jQuery usa uma sintaxe abreviada muito semelhante ao Prototype; apesar disso, há duas diferenças muito importantes: primeiramente, jQuery permite o "encadeamento" de funções e métodos para elas, e segundo, ao invés de pegar qualquer valor ID ou um

objeto DOM como argumento, você pode também fornecer à função dólar(\$) da jQuery uma expressão no formato XPath ou CSS. Por outro lado, jQuery vem com um monte de funcionalidades embutidas que vão além dos recursos do Prototype e que tem uma ligeira sobreposição com o que Dojo oferece, por exemplo, algo semelhante aos widgets do Dojo é o sistema de plugins embutidos da jQuery, que permite estender a jQuery com os métodos criados pelo usuário, que ele pode encadear da mesma forma que acontece nos métodos padrões jQuery. Curiosamente, jQuery excede o aspecto em que Prototype e Dojo são fracos: documentação. Quanto as suas deficiências em relação às outras duas bibliotecas, enquanto jQuery oferece uma série de recursos úteis e algumas animações básicas, ela não pode competir com Dojo, que oferece muito mais, e nem pode competir também com o Prototype, no que se refere à quantidade de funções utilitárias e manipuladores de eventos. Em suma, ATES acredita que se você estiver procurando por um conjunto maciço de recursos, use Dojo. Se você está procurando otimizar as suas técnicas de codificação de JavaScript e reduzir o tempo de desenvolvimento, experimente Prototype. E, finalmente, se você quiser ter o melhor dos dois mundos e não têm medo de escrever uma quantidade razoável de funcionalidades, jQuery é uma opção que vale ser considerada.

Em relação à Mootools, NEWTON(2009) esclarece que tanto ela quanto a jQuery são excelentes opções, contudo, elas possuem focos diferentes, segundo ele, enquanto o escopo da jQuery é o DOM (mudar propriedades CSS, animar coisas, procurar conteúdo através de AJAX, etc), Mootools é focado no JavaScript como um todo, envolvendo além do DOM, sistema de herança, melhoria para os objetos nativos (ex: Function, String) e etc.. Ou seja, tudo que a jQuery pode fazer, o Mootools também consegue, mas o inverso não é verdade.

Pelo que foi visto surge a dúvida: “Mas, se existem bibliotecas mais completas porque a jQuery é tão popular?”. Entre os possíveis motivos pode-se considerar que: 99% de todo código JavaScript pelo mundo afora roda em navegadores, e que a necessidade mais comum neste ambiente seja justamente a manipulação do DOM; a curva de aprendizado da jQuery é menor que as demais bibliotecas, em parte por não ser preciso ter um bom conhecimento prévio de JavaScript, e pelas codificação nas demais bibliotecas ser de forma mais explícita; e finalmente, o fato da comunidade de usuários jQuery ser de longe a maior, e a mesma promover a

biblioteca de maneira mais ativa. Em suma, a escolha da biblioteca dependerá do que você realmente precisa realizar e seu estilo de programar.

Tabela 4: Comparativo das bibliotecas Javascript

	Prototype	jQuery	Yui	ExtJS	MooTolls
Latest Version	1.6.1	1.4.1	3.00	3.1	1.2.4
License	MIT	MIT & GPL	BSD	Commercial & GPL v3	MIT
Browser Compatibility					
IE Support	6.0+	6.0+	6.0+	6.0+	6.0+
Firefox Support	1.5+	2.0+	3.0+	1.5+	2.0+
Safari Support	2.0.4+	3.0+	4.0+	3.0+	2.0+
Opera Support	9.25+	9.0+	10.0+	9.0+	9.0+
Chrome Support	1.0+	1.0+	Not verified	Not verified	Not verified
Core Features					
Ajax Support	Yes	Yes	Yes	Yes	Yes
DOM Manipulation	Yes	Yes	Yes	Yes	Yes
DOM Traversal	Yes	Yes	Yes	Yes	Yes
Event Handling	Yes	Yes	Yes	Yes	Yes
JSON	Yes	Yes	Yes	Yes	Yes
Selectors	Yes	Yes	Yes	Yes	Yes
UX/UI Enhancements					
Accordion	No	jQuery UI	No	Yes	MooTools More
Animation	scriptaculous	Yes	Yes	Yes	Yes
Auto Completion	scriptaculous	No	Yes	Yes	No
Browser History	scriptaculous	No	Yes	Yes	No
Calendar	No	jQuery UI	Yes	Yes	No
Charting	No	No	Yes	Yes	No
Drag and Drop	scriptaculous	jQuery UI	Yes	Yes	MooTools More
Grids	No	No	Yes	Yes	MooTools More
Progress Bar	No	jQuery UI	Yes	Yes	No
Resizing	No	jQuery UI	Yes	Yes	No
Rich Text Editor	No	No	Yes	Yes	No
Slider	scriptaculous	jQuery UI	Yes	Yes	MooTools More
Tabs	No	jQuery UI	Yes	Yes	No
Themes	No	jQuery UI	Yes	Yes	MooTools More
Tree Views	No	No	Yes	Yes	No

Fonte: LENNON, Joe. Compare JavaScript frameworks. Disponível em: <
<http://www.ibm.com/developerworks/web/library/wa-jsframeworks/index.html>> Acesso em: 13/11/2010.
 (fonte reeditada pelo autor)

Outro questionamento comum é de como proceder na situação de implantar a jQuery em um ambiente que já utiliza um outra biblioteca JavaScript (ex: sistema legado). O problema de conflito surge se alguma das outras bibliotecas também utilizar algum identificador chamado **jQuery** ou **\$**. O primeiro caso é bem improvável de acontecer, já o segundo é bem comum, pois, por exemplo, a biblioteca Prototype também usa como identificador o símbolo do cifrão (\$). Para eliminar este conflito, na jQuery há a função `noConflict()`, que deve ser chamada após a bibliotecas conflitantes terem sido carregadas, isto irá alterar o significado do \$ definido pela biblioteca não jQuery

CONSIDERAÇÕES FINAIS

Procurou-se demonstrar neste trabalho que atualmente a internet é um ambiente dinâmico, e os seus usuários valorizam cada vez mais o estilo e funcionalidades dos sites, estas características podem ser obtidas com o auxílio da linguagem JavaScript para a criação de sites interessantes e interativos, porém, apesar de ser uma linguagem com bastante recursos, ainda faz com que os desenvolvedores gastem boa parte do seu tempo digitando inúmeras linhas de código, para, por exemplo, selecionar elementos da página e em seguida manipular suas propriedades, seja para validação de dados ou criar algum belo efeito gráfico.

Diante dessas dificuldades, os desenvolvedores começaram a utilizar bibliotecas JavaScript como jQuery para automatizar tarefas comuns e simplificar as mais complicadas. Uma das razões da jQuery ser uma escolha popular é a sua capacidade para ajudar em uma ampla gama de tarefas; pode-se afirmar que John Resig ao criar a jQuery percebeu que jamais conseguiria desenvolver uma biblioteca que atendesse a necessidade de todos os usuários, por isso focou sua criação em livrar os programadores das tarefa repetitiva de manipulação de elementos, em contrapartida, possibilitou qualquer um estender facilmente seus recursos através de plugins, adicionando as funcionalidades que achasse necessárias.

Além disso, a maioria dos conceitos do projeto da biblioteca é emprestada da estrutura HTML e Cascading Style Sheets (CSS), ocasionando uma redução na curva de aprendizado, tanto para designers com pouca experiência em programação como para desenvolvedores web com mais experiência com essas tecnologias. Vale ainda ressaltar que ao trabalharmos com a biblioteca conforme o conceito de Javascript não obstrutivo, que, aliás, muito se assemelha ao padrão MVC¹⁰, nossas páginas ganham a capacidade de tolerâncias a falhas.

Diante de todo exposto, nota-se que jQuery está se tornando uma das bibliotecas JavaScript mais adotadas pela comunidade de desenvolvimento de aplicações Web, e provavelmente continuará a crescer em importância a medida que Aplicações de Internet Rica tornam-se cada vez mais predominantes.

¹⁰ O padrão MVC(Model-View-Controller) separa a modelgem de domínio, a apresentação, e as ações baseadas na entrada de dados do usuário em três classes distintas. BURBECK, Steve. **How to use Model-View-Controller (MVC)**. Disponível em: <<http://st-www.cs.illinois.edu/users/smarch/st-docs/mvc.html>> Acesso em: 08/11/2010

REFERÊNCIAS

BEIGHLEY, Lynn. **jQuery For DUMMIES** Hoboken: Wiley, 2010

SILVA, Murício Samy. **jQuery: A Biblioteca do Programador JavaScript**. 2. ed. São Paulo: Novatec, 2010.

jQuery JavaScript Library. **DOCUMENTATION** Disponível em: <http://docs.jquery.com/Main_Page> Acesso em: 30/07/2010.

Por onde começar com a jQuery. Disponível em: <http://i18n.2kminterativa.com.br/jquery/jquery-getting-started-pt_br.html> Acesso em: 03/08/2010.

STRAHL, Rick. **An introduction to jQuery: Part 1- The Client Side.** Disponível em: <<http://www.west-wind.com/presentations/jquery/default.aspx>> Acesso em: 01/08/2010.

GRABANSKI, Marc. **Great Ways to Learn jQuery.** Disponível em: <<http://www.learningjquery.com/2010/07/great-ways-to-learn-jquery>> Acesso em: 01/08/2010.

HOLT, Alex. **jQuery and JavaScript Coding: Examples and Best Practices.** Disponível em: <<http://www.smashingmagazine.com/2008/09/16/jquery-examples-and-best-practices/>> Acesso em: 02/08/2010.

LA, Nick. **jQuery Tutorials for Designers.** Disponível em: <<http://www.webdesignerwall.com/tutorials/jquery-tutorials-for-designers/>> Acesso em: 02/08/2010.

WAY, Jeffrey. **15 Resources To Get You Started With jQuery From Scratch.** Disponível em: <<http://net.tutsplus.com/javascript-ajax/15-resources-to-get-you-started-with-jquery-from-scratch/>> Acesso em: 26/07/2010.

WILSON, Simon. **Unobtrusive JavaScript with jQuery.** Disponível em: <<http://simonwillison.net/static/2008/xtech/>> Acesso em: 29/07/2010.

CROCKFORD, Douglas. **JavaScript: A Menos Entendida Linguagem de Programação do Globo!** Disponível em: <<http://javascript.crockford.com/pt/javascript.html>> Acesso em: 08/08/2010.

DOYLE, Matt. **Using jQuery Selectors to Locate Page Elements**. Disponível em: <<http://www.elated.com/articles/jquery-selectors/#basic-selectors>> Acesso em: 22/07/2010.

Mozilla Developer Center. **About JavaScript**. Disponível em: <https://developer.mozilla.org/en/JavaScript/About_JavaScript> Acesso em: 07/08/2010.

ARNO, David. **Why JavaScript is a toy language**. Disponível em: <<http://www.davidarno.org/2010/05/18/why-javascript-is-a-toy-language/>> Acesso em: 30/07/2010.

KOCH, Peter-Paul. **JavaScript and "serious" programmers**. Disponível em: <http://www.quirksmode.org/blog/archives/2005/11/javascript_and_1.html> Acesso em: 12/08/2010.

jQueryWTP. Disponível em: <<http://www.langtags.com/jquerywtp/index.html>> Acesso em: 11/08/2010.

GUTHRIE, Scott. **jQuery Intellisense in VS 2008**. Disponível em: <<http://weblogs.asp.net/scottgu/archive/2008/11/21/jquery-intellisense-in-vs-2008.aspx>> Acesso em: 03/08/2010.

ARORA, Pushkar. **Visual Studio News – jQuery Integration, VS 2010 And .NET Framework 4.0 Unveiled**. Disponível em: <<http://pushkararora.com/net/visual-studio-news-jquery-integration-vs-2010-and-net-framework-40-unveiled/>> Acesso em: 02/08/2010.

DESMOND, Michael. **MIX10: Microsoft Doubles Down on jQuery**. Disponível em: <<http://visualstudiomagazine.com/articles/2010/03/16/mix10-jquery-announcement.aspx>> Acesso em: 02/08/2010.

ASP.NET 4 and Visual Studio 2010 Web Development Overview. Disponível em: <<http://www.asp.net/learn/whitepapers/aspnet4>> Acesso em: 03/08/2010.

Twitter. **jQuery(jquery) on Twitter**. Disponível em: <<http://twitter.com/jquery>> Acesso em: 19/8/2010.

CHAFFER, Jonathan; SWEDBERG, Karl. **Learning J Query 1.3**. Disponível em: <<http://www.slideshare.net/VadivelKV/learning-j-query-13>> Acesso em: 02/08/2010.

jQuery JavaScript Library. **Downloading jQuery**. Disponível em: <http://docs.jquery.com/Downloading_jQuery> Acesso em: 16/08/2010.

jquery-how-it-works.gif - Altura: 387 pixels. Largura 806 pixels. True Color 8 bits. 18,5KB. Formato GIF. Disponível em: <<http://www.webdesignerwall.com/wp-content/uploads/2008/02/jquery-how-it-works.gif>> Acesso em: 30/07/2010.

Jquerywtp.jpg - Altura: 486 pixels. Largura 689 pixels. True Color 24 bits. 45KB. Formato JPEG. Disponível em: <<http://www.langtags.com/jquerywtp/jquerywtp.jpg>> Acesso em: 30/07/2010.

XTND.US. **jQuery API extension for Dreamweaver**. Disponível em: <<http://xtnd.us/dreamweaver/jquery>> Acesso em: 27/07/2010.

MENEZES, Pedro. **Os 40 Melhores Plugins da jQuery para facilitar a vida do Desenvolvedor Web**. Disponível em: <<http://pedromenezes.com/os-40-melhores-plugins-da-jquery-para-facilitar-a-vida-do-desenvolvedor-web>> Acesso em: 28/07/2010.

ATES, Faruk. **Prototype, Dojo, and jQuery**. Disponível em: <<http://www.peachpit.com/guides/content.aspx?g=webdesign&seqNum=281>> Acesso em: 10/11/2010.

NEWTON, Aaron. **jQuery vs MooTools**. Disponível em: <http://jqueryvsmootools.com/index_pt-br.html> Acesso em: 10/11/2010.

LENNON, Joe. **Compare JavaScript frameworks**. Disponível em: <<http://www.ibm.com/developerworks/web/library/wa-jsframeworks/index.html>> Acesso em: 13/11/2010. (fonte reeditada pelo autor)

W3SCHOOLS. **AJAX Tutorial**. Disponível em: <<http://www.w3schools.com/ajax/default.asp>> Acesso em: 23/07/2010.

VIEIRA, Nando. **Conceituando o que é Web 2.0**. Disponível em: <<http://web2.0br.com.br/conceito-web20/>> Acesso em: 22/07/2010.

SVOLENSKI, Matheus Smythe. **Sobre o Document Object Model**. Disponível em: <https://developer.mozilla.org/pt/Sobre_o_Document_Object_Model> Acesso em: 22/07/2010.

Ambiente de Desenvolvimento Integrados (IDE). Disponível em: <<http://homepages.dcc.ufmg.br/~mlbc/cursos/internet/java/>> Acesso em: 10/11/2010.

ANEXO A

SELETORES BÁSICOS

Seletor	Descrição	Exemplo
All	Selecionando todos os elementos da página, incluindo head, body e assim por diante.	<code>\$("*")</code>
Element	Seleciona todos os elementos com o nome de uma tag passada.	<code>\$("p")</code> <code>\$("div")</code>
Class	Seleciona todos os elementos com uma classe CSS passada.	<code>\$(".myClass")</code> <code>\$("p.myClass")</code>
ID	Seleciona um simples elemento com um atributo id passado.	<code>\$("#myID")</code> <code>\$("p.#myID")</code>
Fonte: DOYLE, Matt. Using jQuery Selectors to Locate Page Elements . Disponível em: http://www.elated.com/articles/jquery-selectors/#basic-selectors Acesso em: 22/07/2010. (fonte reeditada pelo autor)		

ANEXO B

20 EXEMPLOS DE PLUGINS PARA JQUERY

1. Interface - opções para movimentar divs, mudar tamanhos, criar slideshows, auto-completar, criar carrinho de compras e outras coisas.	2. jQuery Flash Plugin - “simples, mas poderoso” ~ detecta se o Flash está instalado, pede para instalar, é acessível e não obstrutivo! Possui códigos legais feitos com ele como inserir um mp3 player em Flash e vídeos do YouTube.
3. TableSorter - permite que o usuário edite a ordem dos dados de tabelas sem precisar recarregar a página.	4. jQMinMax - adiciona suporte a min-width, max-width, min-height e max-height onde eles não são suportados.
5. QuickSearch - plugin não-obstrutivo que busca textos e possui várias opções.	6. Tabs version 2 - sistema de abas com diversos efeitos.
7. autocomplete - faça sistemas de autocompletar facilmente com esse plugin.	8. autohelp - ajuda o usuário a preencher formulários dando dicas do que escrever.
9. editable - edite elementos com um click.	10. AutoScroll - segure Control e mova o mouse para rolar a página.
11. jQuery History - grave o histórico e ative a opção de voltar no navegador.	12. CheckBox - mude as checkboxes de seus formulários com esse plugin e CSS.
13. Chili - colore a sintaxe de códigos de diversas linguagens com CSS.	14. ColorGradient - cria gradient em elementos com as cores selecionadas.
15. Gradient - cria efeito gradient no fundo de elementos.	16. Curvy Corners - adicione bordas arredondadas em elementos facilmente. Sem Photoshop!
17. GreyBox Redux - mostre fotos, sites e outras coisas em uma janela no estilo LightBox.	18. jCarousel - controla listas verticais e horizontais com conteúdo dinâmico ou estático e com animação.
19. jScrollPane - mude a aparência das scrollbars.	20. MouseGestures - identifique os movimentos do mouse.
<p>Fonte: MENEZES, Pedro. Os 40 Melhores Plugins da jQuery para facilitar a vida do Desenvolvedor Web. Disponível em: < http://pedromenezes.com/os-40-melhores-plugins-da-jquery-para-facilitar-a-vida-do-desenvolvedor-web> Acesso em: 28/07/ 2010. (fonte reeditada pelo autor)</p>	