Análise Comparativa entre Bibliotecas Javascript para o Desenvolvimento de Interfaces Web Ricas

Erica Fernanda Berniche, Vânia Paula de Almeida Neris

Abstract: This paper presents a comparison between javascript libraries for the development of rich web interfaces. jQuery, Mootools, Prototype, script.aculo.us and YUI libraries were analyzed considering the following evaluation criteria: compatibility, interactivity, implementation effort, support and documentation. The results intend to support developers while taking decision about which library would better support them in the development of web applications.

Keywords: javascript libraries, rich web interfaces

Resumo: Este artigo apresenta uma análise comparativa entre bibliotecas javascript para o desenvolvimento de interfaces web ricas. As bibliotecas jQuery, Mootools, Prototype, script.aculo.us e YUI foram analisadas considerando-se os seguintes critérios de avaliação: compatibilidade entre navegadores, interatividade, esforço de implementação, suporte e documentação. Os resultados apresentados podem apoiar a tomada de decisão do desenvolvedor sobre qual biblioteca utilizar no desenvolvimento de uma aplicação web.

Palavras-Chave: bibliotecas javascript, interfaces web ricas.

I. INTRODUÇÃO

O desenvolvimento de interfaces web tornou-se rápido e prático com o surgimento das bibliotecas Javascript, que fornecem diversos recursos para otimizar a implementação de aplicações web ricas ou Rich Internet Applications (RIA), em inglês. Nessas aplicações alguns de seus recursos são similares aos de uma aplicação desktop, além de possuir suporte a animações e gráficos. Um de seus objetivos é aumentar a interatividade com seus usuários, possibilitando uma riqueza de detalhes não suportada pelas aplicações desktop.

A utilização de bibliotecas javascript pode facilitar o desenvolvimento das interfaces dessas aplicações porque disponibilizam funcionalidades complexas de forma clara e simples de serem utilizadas.

Porém, ao realizar um levantamento sobre qual biblioteca javascript utilizar no desenvolvimento das interfaces de sua aplicação, o desenvolvedor se depara com várias bibliotecas que fornecem diferentes recursos para tornar as interfaces de usuário desenvolvidas com aspecto e recursos similares a aplicações desktops.

O objetivo desse artigo é fornecer uma análise comparativa entre cinco bibliotecas Javascript – jQuery, Mootools, Prototype, script.aculo.us e YUI – disponíveis no mercado para a construção de interfaces web ricas, com o intuito de fornecer um material de apoio que auxilie o desenvolvedor na escolha de qual biblioteca utilizar em sua aplicação. As bibliotecas foram analisadas considerando-se quatro critérios: compatibilidade entre navegadores, suporte e documentação, interatividade e esforço de implementação. Considerando as comparações realizadas, uma árvore de decisão sistematizando o apoio ao desenvolvedor foi formalizada.

Visando aumentar a confiabilidade da abordagem de pesquisa adotada, alguns procedimentos foram executados como, por exemplo, a análise de compatibilidade das bibliotecas com os navegadores web para verificar se seus componentes são exibidos corretamente nesses navegadores.

Esse artigo contém seis seções. Na seção dois são apresentados outros trabalhos de comparação entre bibliotecas javascript. Na seção três é apresentada uma fundamentação teórica sobre os assuntos relacionados ao tema proposto. Na seção quatro é apresentado o método de avaliação adotado para analisar as bibliotecas selecionadas. A seção cinco apresenta os resultados obtidos na análise comparativa. Na seção seis é apresentada uma árvore de decisão formada a partir dos resultados obtidos e, por fim, na última seção, é realizada a conclusão do artigo.

II. TRABALHOS RELACIONADOS

As análises comparativas entre bibliotecas Javascript...
encontradas na literatura dão ênfase a quais componentes disponíveis nessas bibliotecas seriam mais indicados para uma funcionalidade específica. Baseados nessa abordagem, Cogo et al. (2009) analisaram várias bibliotecas com o objetivo de discutir as vantagens e desvantagens da utilização de alguns componentes presentes nessas bibliotecas e mostrar qual é a mais indicada para cada componente selecionado. Através de análises, esses autores concluíram, por exemplo, que o componente de calendário disponível na biblioteca YUI é o mais indicado entre as bibliotecas selecionadas – jQuery, YUI e QooXDoc – por sua facilidade de configuração e utilização.

Outros trabalhos encontrados, como o de Lennon (2010), realizaram análises comparativas baseadas em algumas características que são comumente encontradas nessas bibliotecas como, por exemplo, suporte a Asynchronous Javascript and XML (AJAX) e componentes que melhoram a experiência do usuário com as aplicações. Em sua análise, Lennon (2010) pode verificar, por exemplo, que dentre as cinco bibliotecas analisadas – jQuery, Prototype, YUI, ExtJs e MooTools – a biblioteca ExtJs é a biblioteca com a maior quantidade de componentes e recursos que facilitam a utilização de uma aplicação web.

Em seu trabalho, Morales et al. (2011) analisaram as bibliotecas Javascript com o objetivo de avaliar seu suporte ao elemento multimídia. Uma tabela de comparação foi criada para mostrar quais bibliotecas foram avaliadas, quais arquivos eram necessários para que o componente multimídia fosse exibido e como esse componente foi exibido por cada biblioteca. Ao final da análise, Morales et al. (2011) concluíram que a biblioteca Dojo é a mais completa para a exibição desse componente, já que para as outras é necessária a inclusão de plugins para que seja exibido corretamente.

O presente trabalho difere dos trabalhos encontrados na literatura porque fornece ao desenvolvedor uma árvore de decisão baseada nos resultados obtidos através da análise comparativa realizada entre as principais bibliotecas Javascript disponíveis no mercado, considerando os critérios de compatibilidade entre navegadores, suporte e documentação, interatividade e esforço de implementação. A partir dessa árvore de decisão, o desenvolvedor pode selecionar a biblioteca adequada para construção de interfaces da sua aplicação. Além disso, analisa comparativamente critérios diferentes dos encontrados nos trabalhos relacionados como, por exemplo, a compatibilidade com os navegadores e o esforço de implementação.

III. Fundamentação Teórica

O termo Rich Internet Applications (RIA) surgiu para denominar um novo tipo de aplicações web que fornecem maior interatividade aos seus usuários. Para Morales et al. (2011), essa abordagem possui o comportamento e as características de uma aplicação desktop, além de possuir animações, conteúdo multimídia e processamento do lado cliente, permitindo que os usuários utilizem a aplicação com uma interface visual mais agradável.

De acordo com Busch e Koch (2009), algumas vantagens oferecidas por esse tipo de aplicação são a arquitetura cliente-servidor, a variedade de componentes que podem ser utilizados para a construção de interfaces e a comunicação assíncrona. Outra vantagem desse tipo de aplicação é o fato de que as páginas não precisam ser atualizadas completamente e, com isso, diminuiu-se o tempo de processamento de uma página e a sobrecarga de processamento do servidor, já que apenas uma parte da página é atualizada.

Aplicações RIA podem ser desenvolvidas utilizando-se bibliotecas Javascript para simplificar seu desenvolvimento, permitir que uma parte do processamento seja gerenciada pelo lado cliente da aplicação e fornecer suporte para diversos navegadores. Além disso, essas bibliotecas fornecem suporte a Asynchronous Javascript and XML (AJAX), permitindo que as aplicações não sejam recarregadas a cada requisição ao servidor.

Existem diversas bibliotecas Javascript disponíveis no mercado que disponibilizam esses recursos e é necessário que o desenvolvedor faça uma análise entre elas para selecionar aquela que atenda aos requisitos da sua aplicação.

Para a análise comparativa proposta para esse artigo, com o intuito de auxiliar o desenvolvedor na sua decisão, foram selecionadas as seguintes bibliotecas: jQuery Javascript Library v1.6.2, Prototype v1.7, script.aculo.us v1.9, MooTools v1.4 e YUI v3.3.0. Tais bibliotecas foram selecionadas por serem as principais utilizadas no desenvolvimento de aplicações RIA.

A) jQuery Javascript Library v1.6.2

De acordo com Silva (2008), a biblioteca jQuery fornece dinamismo e interatividade às páginas web com o intuito de tornar agradável a experiência do usuário com a aplicação, além de fornecer diversas funcionalidades ao desenvolvedor para facilitar a criação de scripts. Esses scripts visam incrementar, de forma progressiva e não obstrutiva, a usabilidade, a acessibilidade e o design, enriquecendo a experiência do usuário.

Uma das características da jQuery é o fato de permitir que os elementos de uma página sejam selecionados através de seus atributos como, por exemplo, o atributo id ou o atributo class. Com isso, esses elementos podem ser localizados sem que o desenvolvedor codifique várias linhas de código.

Ainda de acordo com Silva (2008), essa biblioteca pode ser utilizada para adicionar efeitos visuais e animações em aplicações que exigem um alto nível de interatividade com seu usuário, além de recuperar informações do servidor sem a necessidade de recarregar toda a página novamente.

Esses efeitos visuais e animações podem ser obtidos utilizando-se a biblioteca jQuery UI. Essa biblioteca é baseada na biblioteca jQuery e utiliza seus recursos com o objetivo de facilitar o desenvolvimento dessas funcionalidades e criar aplicações web interativas. Alguns componentes disponíveis na biblioteca jQuery UI são a caixa de diálogo e o calendário, demonstrados na Figura 1.
Figura 1. Componentes calendário e caixa de diálogo da biblioteca jQuery UI

B) Prototype v1.7 e script.aculo.us v1.9

Apesar de ser inspirada na linguagem de programação Ruby e sua codificação ser facilitada na plataforma Ruby on Rails, a biblioteca Prototype pode ser utilizada com qualquer plataforma de desenvolvimento web como, por exemplo, ASP.NET e J2EE.

De acordo com Dupont (2008), essa biblioteca não possui componentes para a criação de animações e efeitos visuais, mas tais componentes podem ser construídos utilizando-a. Algumas bibliotecas disponíveis no mercado a utilizam como base para a criação de componentes, como é o caso da biblioteca script.aculo.us. A Figura 2 apresenta os componentes autocomplete e drag and drop, disponíveis nessa biblioteca.

Figura 2. Componentes autocomplete e drag and drop da biblioteca script.aculo.us

C) MooTools v1.4

De acordo com Gube e Cheung (2009), MooTools é uma biblioteca javascript leve, modular e orientada a objetos. Uma das suas características é a modularidade, ou seja, é dividida em várias partes interdependentes. Umas aplicação web não precisa possuir todos os seus componentes para ser executada, reduzindo a quantidade de arquivos a serem carregados e o tempo de resposta da aplicação.

A biblioteca MooTools é dividida em duas partes principais: MooTools Core, que contém todas as funções básicas e é necessária para que seus outros arquivos sejam executados e MooTools More, que contém plugins e extensões do arquivo MooTools Core. Alguns recursos disponíveis nessa biblioteca são o accordion e o validator, demonstrados na Figura 3.

Figura 3. Componentes accordion e validator da biblioteca MooTools

D. YUI v3.3.0

Criada por desenvolvedores da empresa Yahoo!, a biblioteca YUI tem o objetivo de facilitar o desenvolvimento das aplicações web e as tornar agradáveis a seus usuários. Formada por vários utilitários e componentes, é escrita principalmente em javascript e pode ser utilizada na criação de aplicações ricas e interativas.

Essa biblioteca é formada por vários módulos, porém apenas o módulo principal, que contém as funções básicas, é obrigatório para que uma aplicação possa utilizá-la. A Figura 4 apresenta alguns recursos disponíveis nessa biblioteca, como o chart e o datatable.

Figura 4. Componentes chart e datatable da biblioteca YUI

IV. METODOLOGIA DE AVALIAÇÃO

Para que a análise comparativa pudesse ser realizada, alguns critérios de avaliação foram selecionados, a saber: compatibilidade entre navegadores, suporte e documentação,

---

4 http://jqueryui.com/demos
5 http://madrobby.github.com/scriptaculous
6 http:// mootools.net/demos/
7 http://developer.yahoo.com/yui/3/examples
interatividade e esforço de implementação. As cinco bibliotecas escolhidas foram avaliadas em todos esses critérios para que suas capacidades e limitações pudessem ser analisadas e detectadas.

A seguir, serão apresentados como cada critério de avaliação foi analisado e uma visão geral do sistema SIA COP, utilizado com estudo piloto para a realização da análise.

A) SIA COP - Sistema de Administração de Concursos para Prefeitura

O sistema SIA COP tem como objetivo auxiliar computacionalmente a Prefeitura Municipal de Cajuru no controle de concursos públicos e/ou processos seletivos na etapa subsequente a sua homologação.

A dificuldade enfrentada pela seção de Recursos Humanos da Prefeitura Municipal de Cajuru encontra-se no momento em que é necessário organizar os resultados de um concurso público. Tais informações são fornecidas por uma empresa contratada pela prefeitura para organizar e realizar esses concursos.

Após a realização do concurso, informações como ordem de classificação, pontuação e dados pessoais dos candidatos são entregues em planilhas eletrônicas ao setor de Recursos Humanos. Esses arquivos estão sujeitos a alterações e fraudes, já que o setor de Recursos Humanos não possui nenhum método para garantir sua integridade.

Outros pontos que devem ser observados consideram o fato de que a seção de Recursos Humanos torna-se totalmente dependente da empresa que prestou os serviços mesmo depois da realização do concurso público e a falta de integração entre os candidatos que prestaram a prova e o setor de Recursos Humanos. Esses candidatos não possuem um meio para acompanhar o andamento do concurso após a divulgação dos resultados.

Baseado nesse cenário, um sistema web foi desenvolvido para controlar e gerenciar as informações fornecidas pelas empresas responsáveis pela realização dos concursos públicos e posteriormente divulgá-las aos candidatos que obtiveram nota total igual ou superior a 50%. O sistema é capaz de extrair informações de um arquivo no formato “csv” criado a partir de um padrão pré-estabelecido entre a seção de Recursos Humanos e a empresa contratada para a realização do concurso. Também disponibiliza informações para o preenchimento de vagas de acordo com a necessidade, de forma rápida, organizada e segura, fornecendo integridade e clareza ao processo.

Os candidatos podem visualizar informações como classificação e dados estatísticos através de um perfil individual criado no sistema e atualizar seus dados pessoais, facilitando a comunicação com o setor de Recursos Humanos. Também podem encontrar informações de como tomar posse do cargo ou mesmo recusá-lo, tendo a possibilidade de imprimir um termo de desistência. O sistema também armazena imagens digitalizadas, principalmente documentos assinados, para controle e manutenção de um histórico das informações.

No contexto deste artigo, esse sistema foi utilizado para a realização da análise comparativa. Essa análise foi feita com base na interface do caso de uso “Manter Servidor Público”, que é responsável por inserir e atualizar os registros referentes aos servidores públicos da Prefeitura Municipal de Cajuru.

O protótipo da interface utilizada para a análise é exibida na Figura 5. Nesse protótipo são exibidos todos os campos que os formulários desenvolvidos devem contemplar. Para a análise comparativa entre as bibliotecas foram utilizados os componentes accordion, calendário e o plugin para a implementação de máscaras nos campos que possuem alguma formatação. Tais componentes são utilizados com o objetivo de tornar os formulários mais interativos e de fácil utilização pelo usuário do sistema.

Figura 5. Protótipo da interface de cadastro do ca-so de uso Manter Servidor Público

O componente accordion é utilizado nos casos em que se deseja organizar as informações por seções. No protótipo desenvolvido para a análise, as informações são agrupadas em duas seções: Dados Gerais, que contém informações como nome, email e telefones, e Endereço, que contém os dados sobre o endereço do servidor público que será cadastrado.

O calendário é exibido nos campos que são do tipo data como, por exemplo, data de nascimento e data de admissão do servidor público. Já o plugin para a implementação de máscaras é utilizado nos campos que possuem algum tipo de formatação como, por exemplo, CPF, telefone e CEP.

B) Critérios de avaliação

A seguir são apresentados os critérios de avaliação que foram utilizados para a análise comparativa entre as bibliotecas JQuery, Mootools, YUI, Prototype e script.aculo.us.

Compatibilidade entre navegadores

Para que a análise de compatibilidade entre navegadores pudesse ser realizada, os navegadores Internet Explorer 9.0, Firefox 7.0 e Google Chrome 14.0 foram utilizados por serem os mais populares, de acordo com o levantamento realizado pela W3Schools, exibido na Tabela 1.
Os seguintes itens foram analisados nesse critério:

I. Renderização — foi analisado como cada formulário foi renderizado pelo navegador e se os componentes presentes nos formulários foram exibidos corretamente.

II. Comportamento — foi avaliado se os componentes existentes nos formulários desenvolvidos possuem o mesmo comportamento em todos os navegadores.

Interatividade

Por interatividade entende-se a capacidade da biblioteca em fornecer recursos para tornar a aplicação mais atrativa ao usuário, além de facilitar sua utilização.

Para a avaliação desse critério, os componentes accordion, calendário e o plugin para a implementação de máscara nos campos foram avaliados.

Os itens que foram analisados são descritos abaixo:

I. Suporte ao componente — foi analisado se a biblioteca possui suporte aos componentes selecionados, ou seja, se podem ser implementados utilizando-a.

II. Suporte ao idioma — foi avaliado se o componente calendário possui suporte a tradução dos textos. Para esse componente é importante que os dias da semana, meses e demais informações estejam traduzidas para a linguagem nativa do usuário, para sua melhor compreensão e utilização.

Esforço de implementação

Nesse critério foi avaliada a complexidade de se desenvolver uma aplicação utilizando cada biblioteca. Para que essa análise pudesse ser realizada, a interface de cadastro do caso de uso “Manter Servidor Público” do sistema SIACOP foi desenvolvida utilizando-se cada uma delas.

O seguinte item foi avaliado para a análise desse critério:

I. Quantidade de linhas de código — foi analisada a quantidade de linhas de código em javascript necessários para o desenvolvimento dos componentes selecionados, com exceção dos arquivos da própria biblioteca.

Suporte e documentação

Para esse critério deve-se levar em consideração a documentação disponibilizada por cada biblioteca para que o desenvolvedor possa implementá-la sem a necessidade de conhecer profundamente seu funcionamento e o suporte da comunidade desenvolvedora aos usuários de cada biblioteca.

Os itens que foram analisados para esse critério são:

I. Documentação oficial — foi analisada a existência de documentação oficial disponibilizada no site das bibliotecas.

II. Fóruns oficiais — foi analisada a presença de fóruns oficiais presentes no site das bibliotecas.

III. Exemplos de implementação — foi avaliado a existência de exemplos de implementação no site das bibliotecas.

V. Resultados da análise

Para que a análise comparativa pudesse ser realizada, um formulário para cada biblioteca foi desenvolvido com base no protótipo da interface de cadastro do caso de uso “Manter Servidor Público”, demonstrado na Figura 5.

Os formulários e os resultados dessa análise são apresentados nas próximas seções, de acordo com cada critério de avaliação citado nas seções anteriores.

A) Formulários

Na Figura 6, o formulário à direita representa o protótipo desenvolvido com a biblioteca jQuery Javascript Library v1.6.2 e o formulário à esquerda, o protótipo implementado com as bibliotecas Prototype v1.7 e script.aculo.us v1.9.

---

8http://www.w3schools.com/browsers/browsers_stats.asp
B) Resultados

Nas próximas seções são apresentados os resultados das análises comparativas realizadas com os formulários apresentados nas Figuras 6 e 7.

Compatibilidade entre navegadores

Para o critério de compatibilidade entre navegadores foram avaliados os itens renderização e comportamento. A Tabela 2 representa os resultados obtidos com a análise do item renderização. Nessa tabela, para cada navegador, as bibliotecas que foram renderizadas corretamente são simbolizadas com um X.

<table>
<thead>
<tr>
<th>Biblioteca</th>
<th>IE 9.0</th>
<th>Firefox 7.0</th>
<th>Chrome 14.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>jQuery v1.6.2</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Prototype v1.7 e script.aculo.us v1.9</td>
<td>X</td>
<td>X</td>
<td>--</td>
</tr>
<tr>
<td>MooTools v1.4</td>
<td>X</td>
<td>X</td>
<td>--</td>
</tr>
<tr>
<td>YUI v3.3.0</td>
<td>X</td>
<td>X</td>
<td>--</td>
</tr>
</tbody>
</table>

Em todos os navegadores analisados, os formulários foram exibidos corretamente, conforme demonstrado pela Tabela 2. Um ponto que deve ser observado é que, apesar do Internet Explorer exibir os formulários corretamente, é necessário habilitar a opção que permite exibir o conteúdo bloqueado, uma vez que esse navegador restringe as páginas que possuem scripts.

Os resultados demonstrados na Tabela 3 referem-se à análise realizada para o item comportamento no critério compatibilidade entre os navegadores. Para cada componente utilizado no formulário, foi analisado se seu comportamento era o mesmo para os três navegadores.

Na Tabela 3, os componentes representados pelo símbolo X apresentaram o mesmo comportamento para os navegadores analisados.

<table>
<thead>
<tr>
<th>Biblioteca</th>
<th>Accordion</th>
<th>Calendário</th>
<th>Máscara</th>
</tr>
</thead>
<tbody>
<tr>
<td>jQuery v1.6.2</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Prototype v1.7 e script.aculo.us v1.9</td>
<td>X</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>MooTools v1.4</td>
<td>X</td>
<td>X</td>
<td>--</td>
</tr>
<tr>
<td>YUI v3.3.0</td>
<td>X</td>
<td>X</td>
<td>--</td>
</tr>
</tbody>
</table>

Um ponto que pode ser observado diz respeito ao componente calendário. Para as bibliotecas jQuery v1.6.2 e MooTools v1.4 esse componente é exibido abaixo do campo selecionado, conforme pode ser observado nas Figuras 6 e 7. Já para a biblioteca YUI v3.3.0, encontra-se no próprio formulário, conforme demonstrado na Figura 7.

Os componentes que não foram selecionados na Tabela 3 não são suportados pelas bibliotecas correspondentes. Uma análise detalhada é realizada na seção 5.2.2.

Interatividade

No critério de avaliação denominado interatividade, os itens suporte ao componente e suporte ao idioma foram analisados. A Tabela 4 exibe os resultados para o item suporte ao componente.

<table>
<thead>
<tr>
<th>Biblioteca</th>
<th>Accordion</th>
<th>Calendário</th>
<th>Máscara</th>
</tr>
</thead>
<tbody>
<tr>
<td>jQuery v1.6.2</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Prototype v1.7 e script.aculo.us v1.9</td>
<td>X</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>MooTools v1.4</td>
<td>X</td>
<td>X</td>
<td>--</td>
</tr>
<tr>
<td>YUI v3.3.0</td>
<td>X</td>
<td>X</td>
<td>--</td>
</tr>
</tbody>
</table>

Pode ser observado, analisando as Tabelas 3 e 4, que os componentes suportados pelas bibliotecas apresentam o mesmo comportamento em todos os navegadores avaliados.

Durante a realização dessa pesquisa não foi possível encontrar um plugin para a implementação de máscaras nos campos que possuem alguma formatação para as bibliotecas Prototype v1.7, script.aculo.us v1.9 e YUI v3.3.0. Para a biblioteca MooTools, existe um plugin chamado Meio.Mask, disponível em seu site, porém esse plugin só é executado com a versão 1.3 dessa biblioteca. Com a versão 1.4 utilizada por esse artigo, não foi possível implementá-la. Já as bibliotecas Prototype v1.7 e script.aculo.us v1.9 possuem apenas o componente accordion disponível para utilização.

Com relação ao item suporte ao idioma, observou-se que as bibliotecas que possuem esse recurso são jQuery v1.6.2 e MooTools v1.4. Para a biblioteca jQuery v1.6.2, um arquivo javascript com a tradução do calendário deve ser adicionado a página para que o componente possa ser traduzido. No site
dessa biblioteca existem 54 arquivos com suporte para diversos idiomas como, por exemplo, russo e chinês. Já para a biblioteca MooTools v1.4, a tradução para o idioma desejado deve ser feita pelo desenvolvedor da aplicação, na própria configuração do componente, conforme exibido na Figura 8.

Figura 8. Exemplo de tradução para a biblioteca MooTools v1.4

Esforço de implementação

Nesse critério de avaliação foram analisados os itens quantidade de linhas de código e a existência de plugins para o desenvolvimento dos componentes selecionados.

Para o primeiro item foi considerado a quantidade de linhas de código desenvolvidas em javascript necessárias para implementar os componentes accordion, calendário e máscara, com exceção dos arquivos da própria biblioteca.

A Tabela 5 representa os resultados obtidos para o desenvolvimento de um componente de cada tipo.

Tabela 5. Quantidade de linhas de código utilizada para cada componente

<table>
<thead>
<tr>
<th>Biblioteca</th>
<th>Accordion</th>
<th>Calendário</th>
<th>Máscara</th>
</tr>
</thead>
<tbody>
<tr>
<td>jQuery v1.6.2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Prototype v1.7 e</td>
<td>5</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>script.aculo.us v1.9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MooTools v1.4</td>
<td>1</td>
<td>7</td>
<td>--</td>
</tr>
<tr>
<td>YUI v3.3.0</td>
<td>8</td>
<td>11</td>
<td>--</td>
</tr>
</tbody>
</table>

Com relação ao componente calendário, não foram consideradas na biblioteca jQuery v1.6.2 as linhas de código do arquivo de suporte ao idioma por se tratar de um arquivo disponibilizado pela própria biblioteca. Entretanto, para a biblioteca MooTools v1.4, essas linhas de código foram contabilizadas, já que existe a necessidade de adaptação do componente para o idioma escolhido.

O componente accordion pode ser desenvolvido utilizando-se todas as bibliotecas selecionadas, porém sua implementação é facilitado nas bibliotecas jQuery v1.6.2 e MooTools v1.4. Para essas bibliotecas, o desenvolvedor da aplicação deve somente informar qual elemento da página será o responsável por gerar esse componente. Já par as bibliotecas YUI v3.3.0 e Prototype v1.7 e script.aculo.us v1.9, além dessas informações, é necessário acrescentar o comando para gerar o componente, pois sem esse comando o accordion não é exibido.

Suporte e documentação

A existência de documentação, fóruns oficiais e exemplos de implementação foram analisados para esse critério de avaliação. A Tabela 6 apresenta os resultados obtidos após a análise realizada nos sites das bibliotecas selecionadas.

Como nos critérios anteriores, os resultados que foram alcançados para o item avaliado são representados simbolicamente por um X.

Tabela 6. Resultados para o critério de suporte e documentação

<table>
<thead>
<tr>
<th>Biblioteca</th>
<th>Documentação</th>
<th>Fóruns</th>
<th>Exemplos de implementação</th>
</tr>
</thead>
<tbody>
<tr>
<td>jQuery v1.6.2</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Prototype v1.7 e</td>
<td>--</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>script.aculo.us v1.9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MooTools v1.4</td>
<td>X</td>
<td>--</td>
<td>X</td>
</tr>
<tr>
<td>YUI v3.3.0</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

Conforme visualizado na Tabela 6, apenas a biblioteca MooTools v1.4 não possui fórum oficial, porém seus desenvolvedores criaram um fórum para discutir problemas e outras questões relacionadas à essa biblioteca. Com base na documentação e nos exemplos de implementação é possível desenvolver aplicações utilizando-se as cinco bibliotecas sem a necessidade de um conhecimento aprofundado sobre seu funcionamento.

V. APOIO À ESCOLHA DE FRAMEWORKS RIA

Com base nos resultados obtidos após a análise comparativa entre as bibliotecas javascript, uma árvore de decisão foi criada, conforme demonstrado na Figura 9, com o intuito de auxiliar o desenvolvedor na escolha de qual biblioteca utilizar no desenvolvimento de sua aplicação.

Figura 9. Árvore de decisão baseada nos resultados obtidos com a análise comparativa

Nessa árvore os critérios de avaliação são vinculados aos seus itens e relacionados com as bibliotecas selecionadas. Para que o relacionamento dos critérios de avaliação com as bibliotecas fosse realizado, os melhores resultados em cada critério foram levados em consideração.
O desenvolvedor que utilizar a árvore de decisão para auxiliar na sua tomada de decisão deve selecionar um critério de avaliação e/ou item e verificar qual biblioteca possui apoio ao que foi selecionado. Por exemplo, se sua aplicação deve possuir suporte a idiomas e baixo esforço de implementação, as bibliotecas que podem ser utilizadas são MooTools v1.4 e jQuery v1.6.

VI. CONCLUSÃO

A existência de diversas bibliotecas para o desenvolvimento de aplicações web ricas torna difícil a escolha do desenvolvedor sobre qual biblioteca utilizar em sua aplicação. Uma análise comparativa entre elas é fundamental para esclarecer qual apresenta o maior número de recursos e facilidades de utilização, ocasionando a diminuição do tempo de desenvolvimento e uma maior interatividade entre a aplicação e seu usuário.

Com a análise comparativa realizada, foi possível gerar uma árvore de decisão que representa os resultados obtidos com a análise dos critérios de avaliação propostos. Baseada nessa árvore e analisando suas combinações o desenvolvedor é capaz de tomar a decisão sobre qual biblioteca entre as analisadas atende os requisitos de desenvolvimento da sua aplicação.

Apesar de a análise comparativa utilizar diversos critérios de avaliação, durante o seu desenvolvimento sentiu-se a necessidade de verificar o comportamento das bibliotecas analisadas com relação à integração com os frameworks para desenvolvimento de aplicações web como, por exemplo, JSF (JavaServer Faces) e JSP (JavaServer Pages), caracterizando um trabalho futuro.

REFERÊNCIAS


