

RichBlocks - Um *Framework* para Implantar Interfaces RIA em Sistemas Web

Jaydson Nascimento Gomes, Rodrigo Prestes Machado

Curso de Análise e Desenvolvimento de Sistemas
Faculdade de Tecnologia Senac RS (FATEC/RS)
Porto Alegre – RS – Brasil

jayalemao@gmail.com, rodrigo@prestesmachado.com.br

Abstract. *The development of Rich Internet Applications (RIA) is among the new technologies that are currently available for the development of web applications. These applications are characterized by the similarity with desktop applications, that use an order Intuitive and easier for users with a rich interface than traditional web applications. This article shows the development of a framework that seeks to remedy the current difficulties in the deployment of web interfaces and proposes to facilitate the development of these applications, deploying an RIA-based interface in Web systems. The article presents the operation of the Framework, as it was built and what benefits it brings to web systems.*

Resumo. *O desenvolvimento de Aplicações Ricas para Internet (RIA – do inglês Rich Internet Applications) esta entre as novas tecnologias que atualmente estão disponíveis para o desenvolvimento de aplicações web. Estas aplicações caracterizam-se pela semelhança com aplicações Desktop, isto visando um uso mais fácil e intuitivo aos usuários com uma interface mais rica do que aplicações web tradicionais. Este artigo mostra o desenvolvimento de um Framework que visa sanar as atuais dificuldades na implantação de interfaces web e se propõe facilitar o desenvolvimento destas aplicações, implantando uma interface baseada em RIA em sistemas web. O artigo apresenta o funcionamento do Framework, como foi construído e quais benefícios ele traz para sistemas web.*

1. Introdução

A Internet tem revolucionado o mundo dos computadores e das comunicações como nenhuma invenção foi capaz de fazer antes (LEINER, 2003). Quando em 1993 foi lançado o navegador Mosaic pela NCSA (*National Center for Supercomputing Applications*) que implementava suporte HTTP e processamento de páginas HTML, não se imaginava o quanto a Internet ainda tinha que evoluir. Naquela época o conteúdo era basicamente todo editorial, ou seja, apenas algumas pessoas e empresas publicavam conteúdo na Internet. Este conteúdo era totalmente estático e as informações eram pouco usuais e relevantes (BRIDEE, 2007). Hoje a Web já se tornou uma plataforma sólida onde aplicações podem ser feitas de forma rápida e eficiente. Com o aumento da velocidade da Internet e o acesso banda larga, as tecnologias web continuam evoluindo

para proporcionar novas experiências aos usuários e aumentar a utilidade e qualidade dessas aplicações. (LOOSLEY, 2006).

O aumento da demanda para construção de aplicações Web de maior complexidade fez com que as aplicações atuais fossem além das possibilidades de uma aplicação Web tradicional.

RIA (*Rich Internet Applications*) é mais um passo no processo evolutivo da Internet. Essas aplicações se assemelham as aplicações desktop, fazendo com que o seu uso seja mais fácil, oferecendo uma interface mais rica aos usuários. (LOOSLEY, 2006).

RIA é a revolução das aplicações Web, é mais que tecnologia é um conceito. É o uso da Internet e das tecnologias disponíveis para criar uma experiência de uso de aplicação mais intuitiva e eficiente para o usuário. É a combinação da interatividade e funcionalidade do Desktop com a abrangência e flexibilidade da Web (BRIDEE, 2007).

Em aplicações RIA, tudo que é Interface deve ser processado no cliente e o que for lógica do negócio deve ser processado no servidor (BRIDEE, 2007).

Algumas características destas aplicações podem ser citadas, como: A aplicação é iniciada a partir de uma página Web, incluindo o seu conteúdo. Os usuários que interagem com a aplicação, obtêm uma reação imediata da aplicação quando são solicitados novos dados, ou seja, não há a necessidade de renderizar a tela inteiramente.

A aplicação utiliza alguns controles de interface de usuário mais modernos como menu, menu em árvore, painel de abas, entre outros elementos gráficos de interface que não são utilizados em aplicações Web tradicionais. Permite o uso de operações comuns em aplicações Desktop, como Drag & Drop, redimensionar, uso de animações, entre outras (BRIDEE, 2007). A figura 1 ilustra as definições RIA.



Figura 1 - RIA

Atualmente existem diversas tecnologias e ferramentas disponíveis que auxiliam no desenvolvimento de RIA's, entre elas podemos citar ferramentas da Adobe como Flash, Flex e Air, Microsoft SilverLight, JavaFX, OpenLazlo, e *Frameworks* baseados em Ajax, como o Google Web ToolKit.

Um dos problemas que podem ser enfrentados ao implementar RIA utilizando alguma das ferramentas citadas anteriormente é a necessidade do aprendizado de tal ferramenta, bem como suas funcionalidades, sintaxe, padrões e dependências (ADAMS, 2007). Levando isso em conta, pode haver certa dificuldade na adoção de RIA como tecnologia no desenvolvimento de aplicações web.

Este artigo é motivado pelo constante avanço que internet vem sofrendo e pela vontade de contribuir para o crescimento e popularização de RIA como metodologia de desenvolvimento web. O fato de existirem poucas ferramentas que sejam voltadas para auxiliar no desenvolvimento focado de uma interface RIA, também faz parte da motivação.

O objetivo deste trabalho é construir um *Framework* utilizando a linguagem de programação Javascript para renderizar interfaces RIA e arquivos XML para armazenar informações de como o sistema deve funcionar. O Framework deve proporcionar de maneira simples para os desenvolvedores implantar interfaces RIA em sistemas/aplicações Web.

O artigo está dividido nas seguintes seções: A seção 2 mostra algumas das tecnologias e ferramentas existentes atualmente para desenvolver aplicações baseadas em RIA. Na seção 3 é descrito detalhadamente o funcionamento do *Framework*, como quais técnicas foram usadas, quais funções foram necessárias para o total funcionamento, como deve ser utilizado e qual é o padrão utilizado. Já na seção 4 é apresentado um estudo de caso feito para validar o *Framework*. Na seção 5 é apresentada a conclusão do artigo, cruzando os objetivos do *Framework* com os resultados obtidos no estudo de caso. A seção é destinada a base referencial do artigo.

2. Trabalhos Relacionados

As atuais ferramentas e tecnologias provêm uma série de funcionalidades que o conceito RIA propõe, como uma nova experiência aos usuários e uma interface rica.

- Adobe Flash, Flex e Air - A Adobe Systems Incorporated possui uma plataforma de tecnologias que engloba ferramentas, estruturas, servidores e serviços que funcionam juntos ajudando desenvolvedores a produzir aplicações RIA (PAREJA 2008, ZANINETTI, 2008).
- Javafx – É um produto da família Sun Microsystems, e se destina para o desenvolvimento de Aplicações RIA. Atualmente JavaFX consiste em uma linguagem de scripts (JavaFXScript) e um SO para dispositivos móveis (JavaFX Mobile) (SUN, 2008).
- MS Silverlight – Baseado em Microsoft .NET com o Microsoft Silverlight, este plug-in é compatível com vários navegadores e plataformas. Esta ferramenta permite com que aplicativos complexos sejam acessados via Web. O Silverlight é bem flexível suporta AJAX, C#, Python e Ruby, integrando-se a aplicativos da Web já existentes (MICROSOFT, 2008).
- Openlazo - É uma plataforma de código aberto para a criação de aplicações RIA. Aplicações Openlazo são escritas em XML e Javascript e de forma transparente

compiladas para o Flash. Sua API proporciona animação, layout, comunicação com servidor, dados obrigatórios entre outras funcionalidades. Openlazlo funciona em todos os browsers e em todas as plataformas (OPENLAZLO, 2008).

- Ajax - (*Asynchronous Javascript And XML*) é o uso sistemático de tecnologias providas por navegadores, como Javascript e XML, para tornar páginas mais interativas com os usuários, utilizando-se de solicitações assíncronas de informações. AJAX não é somente um novo modelo, é também uma iniciativa na construção de aplicações Web mais dinâmicas e criativas, também não é uma tecnologia, são várias tecnologias conhecidas trabalhando juntas, oferecendo novas funcionalidades (MOZILLA, 2008).

3. Funcionamento do *Framework*

O *Framework* possui uma estrutura que é composta por uma série de arquivos XML que são responsáveis por armazenar todos os dados de configuração do sistema em que a interface será implantada, e bibliotecas Javascript que são o núcleo do *Framework*. São estas bibliotecas que provem à funcionalidade RIA para o sistema, lendo os arquivos XML e renderizando a interface simulando uma interface semelhante a uma aplicação Desktop tradicional. A figura 2 ilustra o funcionamento do *Framework*.

A interface gerada pelo *Framework* segue o padrão de aplicações *Desktop* onde as funcionalidades do sistema estão disponíveis em janelas e estas são acessadas através de menus.

Um menu pode possuir várias opções e cada opção de menu é responsável por abrir uma janela ou por executar alguma função Javascript já nativa no *Framework*, ou escrita pelo desenvolvedor. Isto é, quem estiver implantando a interface poderá usar funções já existentes no *Framework* e também poderá customizar a interface de acordo com a necessidade.

As janelas possuem uma série de atributos. Estes atributos definem as funcionalidades da janela, tais como título, posição, dimensões, pagina que será carregada no seu corpo e ainda algumas opções *booleanas* como: maximizar, minimizar, fechar e atualizar. Estes atributos permitem que seja possível criar uma janela definindo quais propriedades serão exibidas.

Já as funções serão definidas da maneira tradicional de definições de funções Javascript, porém estas ficam armazenadas em um arquivo XML, e só são executadas quando chamadas por alguma opção de menu.

Para definir menus, janelas e funções é necessário conhecimento em Javascript e XML, pois todas as configurações ficam armazenadas em arquivos XML e todas as funções nativas foram desenvolvidas em Javascript, qualquer função que seja criada deverá ser implementada também em Javascript.

Primeiramente, o desenvolvedor terá que mapear o sistema em que a interface será implantada. Isto é, determinar quais as opções de menu serão exibidas, quais janelas o sistema vai ter, e quais funções este sistema irá executar. Depois de mapeado, os arquivos XML devem ser editados para determinar quais menus serão necessários, quais janelas

estes menus irão abrir, ou quais funções o menu irá executar. Também é necessário definir as janelas com seus atributos e as funções Javascript que poderão ser executadas.

O *Framework* poderá ser usado em um sistema Web que não utiliza uma interface RIA, ou seja, funciona da maneira tradicional de navegação em um navegador. Após o mapeamento e configurações feitas, o *Framework* se encarregará de adequar este sistema a uma nova interface.

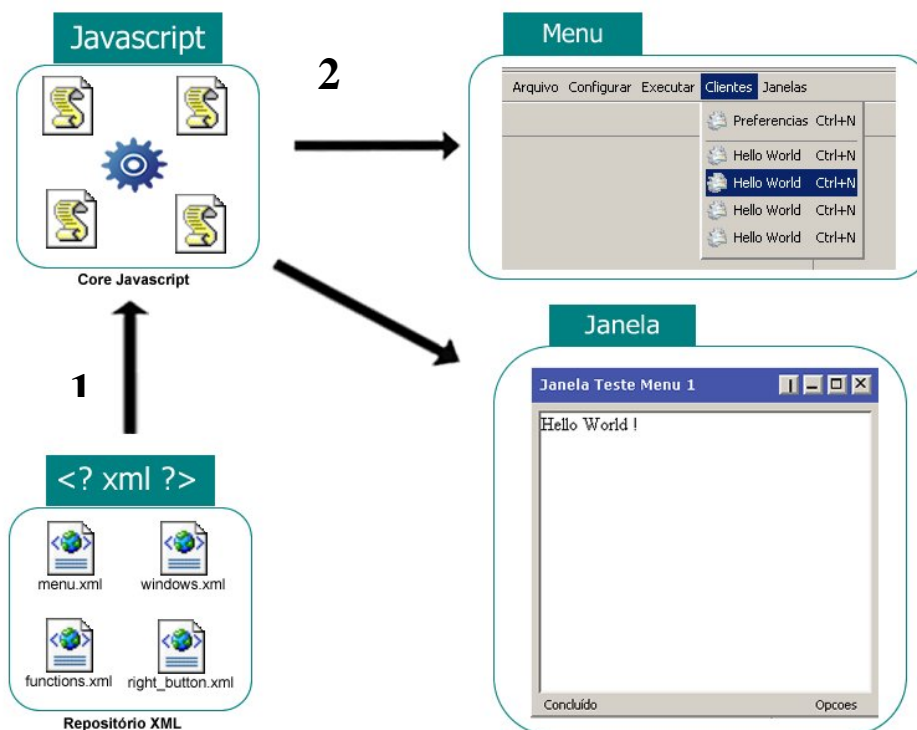


Figura 2. Funcionamento do *Framework*

Na figura 2, é representado no passo 1 o repositório de arquivos XML que as bibliotecas Javascript lêem. Já no passo 2 é ilustrada a interface renderizada pelo Javascript após feita a leitura.

3. 1. Menus

A estrutura de menus consiste em uma hierarquia simples, onde existe uma barra de menus principal que fica disposta na parte superior do sistema. Esta barra de menus possui n opções, e cada opção expande seus respectivos menus. Cada menu possui um filho submenu, e cada filho submenu pode possuir n filhos option. Como dito anteriormente, é necessário ter algum conhecimento básico em XML para poder utilizar o *Framework*, pois para definir quais menus o sistema irá ter é necessário editar o arquivo XML responsável por armazenar os dados que serão exibidos na barra de menus.

No arquivo XML a tag <menu> refere-se ao menu principal que ficará localizado no topo da tela. Este quando selecionado será expandido, mostrando o seu filho <submenu> com todos os seus respectivos filhos <option> e assim sucessivamente.

Todas as tags possuem atributos que definem a característica de cada opção, as principais se referem ao que esta opção irá executar no sistema. A tabela 1 mostra o que cada atributo da tag <option> representa. Na figura 3 é exemplificado um pedaço de código XML que define o menu, e como ele é gerado graficamente em HTML.

Tabela 1. Atributos do Menu

MENU		
Atributos	Descrição	Valores
name	Especifica o nome do menu mestre(topo)	Texto
value	Especifica o valor do menu mestre(topo), este é o valor que aparece no menu	Texto

MENU OPTION		
Atributos	Descrição	Valores
name	Especifica o nome da opção do menu, este é necessário para criar uma referência no sistema.	Texto
value	Especifica o valor que será exibido como opção no sistema	Texto
windowTarget	Especifica qual janela deve ser aberta pelo sistema quando clicado. A janela é definida por um nome único definido no arquivo windows.xml	Texto
img	Especifica a URL de uma imagem que será exibida ao lado do valor	Texto
separator	Insere uma linha para sinalizar categorias de opções	Booleano
shortcut	Especifica qual a combinação de teclas servirá como atalho para executar este Menu	Texto

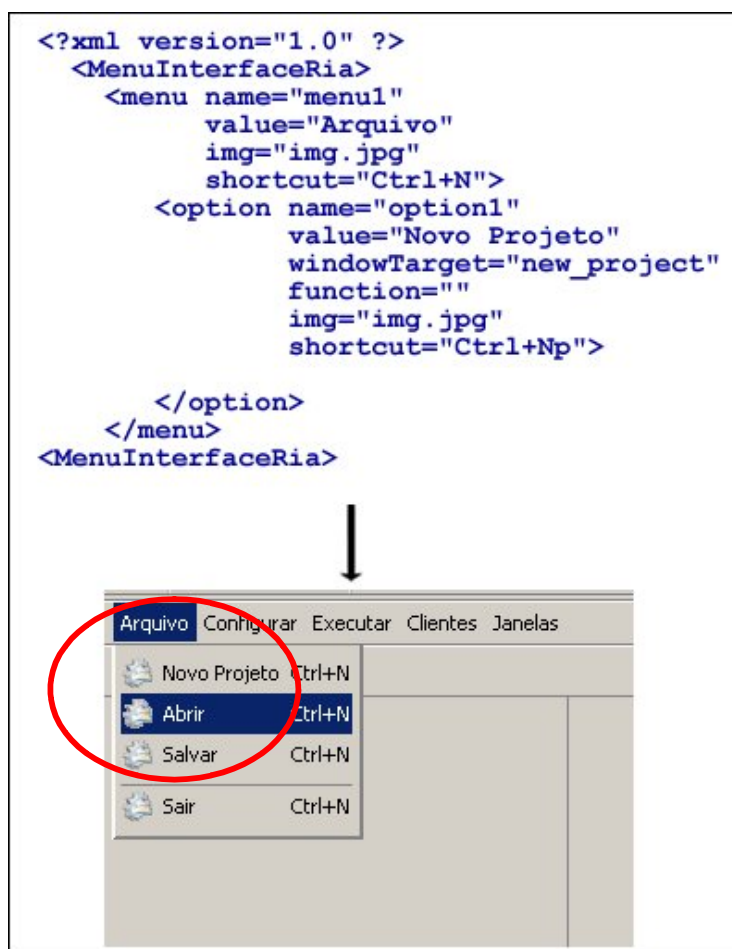


Figura 3 – Menu renderizado baseado no arquivo xml

A figura 3 acima apresenta um trecho de código XML e o resultado gerado pelo Javascript em HTML.

3. 2. Janelas

Sistemas web tradicionalmente funcionam rodando em uma página no browser, e a cada requisição feita para o servidor, esta página é novamente carregada retornando outra página. A interface do *Framework* funciona semelhante a uma aplicação Desktop, não havendo re-carregamento da página. O sistema roda apenas em uma página, que exibe outras páginas carregadas dinamicamente dentro de elementos HTML, assim sendo possível abrir inúmeras opções do sistema de uma só vez.

Estas janelas também são definidas em um arquivo XML que armazena todas as janelas do sistema. Cada janela possui atributos que definem suas características. A tabela 2 exibe estes atributos e a descrição de cada um. Na figura 4 é exemplificado um pedaço de código XML que define a janela, e como ela é gerada graficamente em HTML.

Tabela 2 - Atributos da Janela

WINDOW		
Atibutos	Descrição	Valores
name	Especifica o nome da janela, este é necessário para criar uma referência no sistema para poder ser chamado por algum menu ou função.	Texto
title	Especifica o título da janela, será mostrado na parte superior da janela.	Texto
description	Especifica uma descrição para a janela, necessário apenas para fins de documentação	Texto
defaultTop	Especifica a posição top da janela na tela.	Pixels
defaultLeft	Especifica a posição left da janela na tela.	Pixels
defaultWidth	Especifica a largura da janela.	Pixels
defaultHeight	Especifica a altura da janela.	Pixels
minWidth	Especifica a largura mínima da janela.	Pixels
minHeight	Especifica a altura mínima da janela.	Pixels
pageSrc	Especifica qual URL deve ser carregada no corpo da janela.	Texto
ajax	Especifica o método de carregamento do conteúdo da janela, se true indica que será carregada via Ajax, se false indica que o conteúdo será carregado em um Iframe.	Booleano
newInstance	Atributo que define se pode haver mais de uma instancia da mesma janela no sistema.	Booleano
minimize	Atributo que define se a janela pode ou não ser minimizada	Booleano
maximize	Atributo que define se a janela pode ou não ser maximizada	Booleano
close	Atributo que define se a janela pode ou não ser fechada	Booleano
drag	Atributo que define se a janela pode ou não ser arrastada	Booleano
resize	Atributo que define se a janela pode ou não ser redimensionada	Booleano
defaultFooter	Texto que aparece no rodapé da janela	Texto

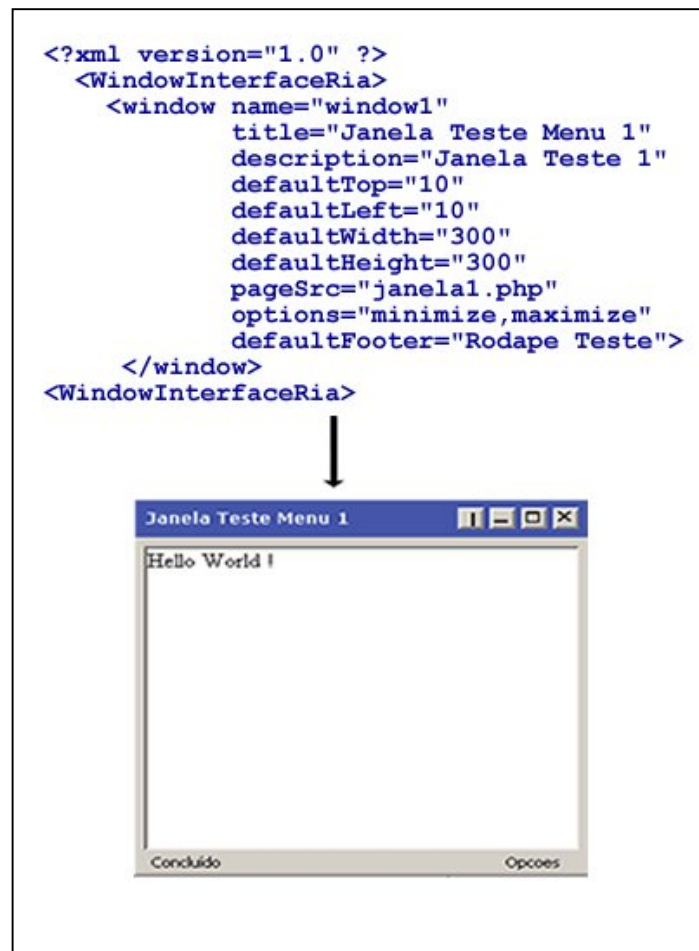


Figura 4. Janela renderizada baseada no arquivo xml

A figura 4 acima apresenta um trecho de código XML e o resultado gerado pelo Javascript em HTML.

3.3. Funções Javascript

Além de funções nativas do *Framework*, ainda existe a possibilidade de criar novas funções Javascript, para fazer a interface se adequar ao sistema em que esta sendo implantada a interface.

Para definir novas funções é preciso conhecer a sintaxe Javascript, e também conhecer a estrutura do *Framework*. As funções também ficam armazenadas em um arquivo XML. A figura 5 ilustra o arquivo XML de criação de funções.

```

<?xml version="1.0" ?>
<FucntionInterfaceRia>
  <function name="function1">
    document.write('hello world !');
    alert('Function 1');
  </function>
</FucntionInterfaceRia>

```

Figura 5. Função Javascript no arquivo xml

4. Estudo de Caso

Para a validação do Framework foi feito um estudo de caso. Dez desenvolvedores atuantes e especializados em desenvolvimento web foram selecionados para testar o funcionamento do Framework em situações reais de desenvolvimento de sistemas/aplicações web durante 3 semanas.

O objetivo deste estudo foi obter sugestões de melhorias, possíveis bugs, e também analisar a facilidade de desenvolvimento e a utilidade do Framework.

O Framework é um software livre de código aberto que está sob a licença GNU *General Public License* v3. Todo seu código está disponível no Google Code (repositório de softwares livres), isto facilitou o estudo de caso, pois os desenvolvedores puderam ter acesso a todo o código fonte e a documentação sobre o *Framework*.

A proposta consistiu em passar um exercício simples para cada desenvolvedor executar uma série de tarefas utilizando o Framework simulando um desenvolvimento real de alguma aplicação. Abaixo segue o exercício passado para os desenvolvedores:

O exercício é simples, a idéia é mostrar como o Framework funciona e o quanto pode ser feito com ele.

Requisitos

A Aplicação deve exibir uma tela com o seu nome.

A Aplicação deve exibir uma tela com um vídeo do YouTube.

A Aplicação deve exibir uma tela com um formulário HTML (Não é preciso submeter o formulário, a não ser que quem estiver desenvolvendo tenha tempo para fazer isso).

Os desenvolvedores receberam um link para fazer *download* da versão mais atualizada do *Framework*. Dentre os arquivos baixados, um arquivo *help* foi disponibilizado visando auxiliar na hora de executar o exercício.

Este arquivo *help* possui exemplos de código e especificações de como o *Framework* funciona, mostrando como configurar os arquivos XML para o sistema funcionar integrado com a interface.

À medida que os desenvolvedores foram executando os testes e concluindo o exercício, foi solicitado o preenchimento de uma pesquisa on-line para colher dados sobre o teste.

A pesquisa conteve as seguintes questões e poderiam ser avaliadas com os conceitos Baixa, Media e Alta:

- Facilidade no desenvolvimento
- Cumprimento efetivo das funcionalidades que se propõe a fornecer.
- Utilidade do Framework.
- Satisfação com o resultado.
- Sugestões de melhorias (texto livre)
- Principais Dúvidas (texto livre)
- Considerações finais (texto livre)
- De uma forma geral conceitue o Framework

As questões de texto livre tiveram o objetivo de colher dados para trabalhos futuros.

4.2 Resultados

A tabela 3 exibe os resultados obtidos com o a pesquisa feita no estudo de caso e a tabela 4 exibe algumas considerações finais dos desenvolvedores.

Tabela 3 - Resultado Estudo de Caso

1.	Facilidade no desenvolvimento	
	Baixa	29%
	Média	43%
	Alta	29%
2.	Cumprimento efetivo das funcionalidades que se propõe a fornecer	
	Baixa	0%
	Média	0%
	Alta	100%
3.	Utilidade do Framework	
	Baixa	0%
	Média	14%
	Alta	86%
4.	Satisfação com o resultado.	
	Baixa	0%
	Média	0%
	Alta	100%
5.	De uma forma geral conceitue o Framework	
	Baixa	0%
	Média	0%
	Alta	100%

Tabela 4 – Principais sugestões dos desenvolvedores

Sugestões de melhorias
Quando tem erro no XML, o menu não renderiza, e também não acusa erro... talvez fosse legal abrir um alert na tela avisando que o XML não tá legal
Cuidar a performance ao carregar a página. Talvez carregue o esqueleto da página e depois carregue alguns menus.
Verificar detalhes no funcionamento em linux.
A idéia do RichBlocks é excelente agora só resta usar e contribuir com novas melhorias que só com o uso será possível.
Centralizar um ponto de desenvolvimento dos XMLs.
Acho que foi feito um bom trabalho na documentação. Entretanto, quando falar para editar o arquivo menu.xml é legal dizer que ele se encontra no diretório conf.
Existe alguma maneira de se eliminar o atributo name das tags? Aquele atributo serve apenas para o sistema certo?
Funcionar nas plataformas Safari e Chrome.
Implantar algum método de integração entre janelas.
Fornecer uma interface gráfica para editar os arquivos XML.

Após análise das sugestões alguns “Bugs” foram consertados, como por exemplo, uma falha na renderização do Javascript para o navegador Internet Explorer 7, onde os botões das janelas não apareciam. Este problema acontecia apenas quando usado o protocolo //file no navegador. Para a plataforma Chrome e Safári, o método de carregamento dos arquivos XML deverá se revisto, pois as 2 plataformas não aceitam a manipulação de arquivos XML.

5. Conclusão

Com base nos resultados obtidos na pesquisa, foi possível verificar que os objetivos do *Framework* foram satisfatórios. Um dos principais problemas levantados no artigo foi à dificuldade de aprendizado e utilização de algumas ferramentas existentes no mercado, voltadas para o desenvolvimento de aplicações baseadas em RIA. O estudo de caso mostra que o *Framework* facilita a implantação de interfaces RIA em sistemas/aplicações Web de uma maneira satisfatória para os desenvolvedores. Utilizando apenas arquivos XML o desenvolvedor consegue criar uma aplicação Web dinâmica e rica, utilizando o conceito RIA como base para a interface.

Este artigo mostrou como é possível criar aplicações ricas para internet utilizando um *Framework*, porém ainda tem que evoluir. Mais funcionalidades podem ser acrescentadas, e soluções para problemas específicos também podem ser resolvidas. Para isso alguns trabalhos futuros podem ser citados:

- Interface para manipular os arquivos XML.

Isso seria uma aplicação, que pode ser desenvolvida em qualquer linguagem de programação que manipule arquivos XML. Esta aplicação proveria uma interface onde os desenvolvedores poderiam manipular o XML graficamente.

- Criar métodos de Integração entre janelas.

Isso seria uma atualização no núcleo do *Framework*, desenvolvendo métodos para para integrar janelas de forma mais simples para o desenvolvedor.

6. Referências

LOOSLEY, Chris, Senior Director SLM Technologies Keynote Systems. Rich Internet Applications: Design, Measurement and Management Challenges 2006. http://www.keynote.com/docs/whitepapers/RichInternet_5.pdf. Acessado em Abril de 2008.

LEINER, Barry, Director of the Research Institute for Advanced Computer Science Internet Society, A Brief History of the Internet. <http://www.isoc.org/internet/history/brief.shtml>. Acessado em Dezembro de 2008.

BRIDEE, Erko, Analista & Desenvolvedor Java - Wise Systems - Curitiba - PR, Web 2.0 e RIA. <http://www.slideshare.net/erko/palestra-web-20-ria>. Acessado em Novembro de 2008.

PAREJA, Silvana, Adobe Systems Incorporated, Plataforma de Tecnologia Adobe para RIAs Impulsiona Inovação na Web, São Paulo, 26 de fevereiro de 2008. http://www.adobe.com/br/aboutadobe/pressroom/pr/feb2008/PR_regional_AIR.pdf. Acessado em Abril de 2008.

ZANINETTI, Gabriela, Adobe Systems Incorporated, Plataforma de Tecnologia Adobe para RIAs Impulsiona Inovação na Web, São Paulo, 26 de fevereiro de 2008. http://www.adobe.com/br/aboutadobe/pressroom/pr/feb2008/PR_regional_AIR.pdf. Acessado em Abril de 2008.

ADAMS, Rob, Adobe Systems Incorporated, Designing for Flex : Overview and introduction to Flex, San Francisco, California, 2007. http://www.adobe.com/devnet/flex/articles/fig_pt1.html. Acessado em Junho de 2008.

MOZILLA, AJAX: Como começar, disponível em:

http://developer.mozilla.org/pt/docs/AJAX:Como_come%C3%A7ar . Acessado em Jun. de 2008.

SUN, JavaFX, disponível em: <http://www.sun.com/software/javafx/index.jsp>. Acessado em Jun. de 2008.

OPENLAZLO, disponível em: <http://www.openlaszlo.org/>. Acessado em Jun. de 2008.

MICROSOT, SilverLight, disponível em: <http://www.microsoft.com/silverlight/>. Acessado em Jun. de 2008.

GOOGLE, Web ToolKit, disponível em: <http://code.google.com/webtoolkit/>. Acessado em Jun. de 2008