

Construindo aplicações de interface rica com JavaFX

Cristiano Agostini *

Daniel Rodrigues **

Resumo

Este artigo apresenta uma introdução à Linguagem de Script JavaFX. São apresentados pontos-chave da linguagem que aproveita toda a bagagem de Java. O conhecimento de Java não é necessário, mas para quem a conhece, as principais diferenças são destacadas. JavaFX é uma linguagem que facilita significativamente a construção de interfaces, e, como toda linguagem de Script, tem foco na produtividade. É uma linguagem recém-lançada, mas com um potencial enorme. Ela roda sobre a máquina virtual Java, uma tecnologia bastante difundida, e leva ao extremo o slogan de Java, que é: "Escreva uma vez, rode em qualquer lugar", pois se pode desenvolver um código JavaFX que rode, sem alterações, em celulares, TVs (em breve) e no Desktop, seja através de um Applet, seja através do protocolo Java Web Start. A adoção da tecnologia ocorrerá a partir do momento em que as dúvidas a respeito da qualidade passarem. JavaFX é rápida, concisa e muito fácil de aprender, sua sintaxe declarativa, valoriza a expressão visual e facilita a divisão de trabalho entre desenvolvedores e designers.

Palavras-chave: JavaFX. Linguagem de Script. Aplicação de Internet Rica (RIA).

1 INTRODUÇÃO

Em um mundo cada vez mais conectado, a *Web* tornou-se o principal canal de comunicação e relacionamento. Os *softwares* começam a seguir a tendência de migrar para essa nova plataforma, que elimina vários obstáculos, como a distância, e minimiza outros, como o tempo, por exemplo. Contudo, ainda não se pode afirmar que a *Web* substitui completamente o *Desktop*,¹ pois sua usabilidade é menor. Com base nisso e na nova tecnologia JavaFX tenta-se unir a dinamicidade da *Web* e o poder do *Desktop*. Essa nova tecnologia usa a plataforma Java como base, o que faz com que sua *Runtime* já esteja disponível em mais de 4,5 bilhões de computadores no mundo (SUN, 2009), além da possibilidade de usar todo o código desenvolvido em Java e sua extensa biblioteca de códigos.

Entre as vantagens de migrar *software* para a *Web*, podem-se citar: acesso ao programa/dados em qualquer lugar conectado à internet, facilidade no compartilhamento de informações, facilidade na instalação e atualização de *software*, entre outras (MACORATTI, 2009). Porém, a sua tecnologia padrão (*XHTML*) não oferece uma experiência de navegação e utilização tão intuitiva como a que é conseguida em um sistema *Desktop* (MACORATTI, 2009).

Para suprir a lacuna, busca-se uma alternativa que una o melhor dos dois mundos (COSTA, 2008). As chamadas *Rich Internet Applications* tentam aproximar a experiência de navegação de um usuário *Web* a um usuário *Desktop*.

* Mestre em Ciência da Computação; Professor da Universidade do Oeste de Santa Catarina; analista programador da Coordenação de Tecnologia em Informação e Comunicação CTIC da Universidade do Oeste de Santa Catarina; cristiano.agosti@unoesc.edu.br

** Acadêmico do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade do Oeste de Santa Catarina; daniel.poprock@gmail.com

1.1 APLICAÇÃO DE INTERNET RICA (RICH INTERNET APPLICATION – RIA)

Primeiramente, deve-se descrever o que é o termo “Aplicação de Internet Rica”, em inglês, Rich Internet Application (RIA). O termo RIA foi cunhado pela empresa *Macromedia* (incorporada pela *Adobe*) para descrever aplicações que tenham características de *software Desktop*, mas rodem na *Web* (ADOBE, 2009).

Segundo Wikipédia (2009), algumas características que permitem classificar um *software* como RIA são:

- a) executar a aplicação em um navegador (lado cliente);
- b) manter os dados no servidor de aplicações;
- c) executar em uma *sandbox*.²

De acordo com Costa (2008), a experiência do usuário com o *software* é melhor em relação ao usuário *Web* convencional, pois várias ações são feitas localmente, o que evita a troca de informações entre cliente e servidor, tornando a aplicação mais rápida.

O termo riqueza, segundo Web Design Development India (2009), significa que as mídias suportadas por uma aplicação dessa categoria são amplas e, de modo geral, para ser considerada rica, a tecnologia deve suportar, entre outras mídias: animações, múltiplas fontes, conferência *on-line*, gráficos vetoriais ou rasterizados, áudio, vídeo e outros.

Segundo Macoratti (2009), o uso de aplicações RIA justifica-se em razão de que ao contrário de uma aplicação Desktop, as RIAs não necessitam de instalação, sua atualização é feita no servidor, novas versões são usadas automaticamente, a aplicação é acessível de qualquer computador conectado à internet, ou seja, o usuário não fica preso a um computador apenas, pois os dados estão na rede, independência de plataforma (sistema operacional e *browser*), menor possibilidade de infecção de vírus nas aplicações, entre outras vantagens.

Podem-se citar vários exemplos de linguagens que permitem implementação RIA; as principais, segundo Wikipédia (2009), são:

- a) Javascript: foi a primeira linguagem que permitiu execução de código no lado do cliente e que teve adoção maciça dos navegadores;
- b) AJAX (Acrônimo para *Asynchronous Javascript and XML*) Javascript e XML assíncronos: conjunto de tecnologias e técnicas que permitem criar aplicações ricas;
- c) Adobe Flash, Adobe Flex, Adobe Air: Adobe Flash usa animações. O Adobe Flex cria interfaces em *flash* compilando em XML, e o Adobe Air une HTML, Ajax, Flash e PDF;
- d) Microsoft Silverlight: concorrente direto do *flash*, utiliza XML e é baseado na plataforma .NET 3.0;
- g) Applets Java: uma das primeiras aplicações do Java, um dos primeiros RIA, são pequenas aplicações Java embutidas na página HTML;
- h) Aplicativos Java: por meio do protocolo Java Web Start podem ser lançados via browser aplicativos Java em todo o seu poder;
- f) JavaFX: nova Tecnologia da Sun para aplicações RIA.

2 JAVA FX

De acordo com Doederlein (2009, p. 19), JavaFX Script é em parte declarativa, o que significa que ela se preocupa em dizer o que fazer e não como fazer. É orientada a objetos, roda na Máquina Virtual Java (JVM), utiliza Classes Java, tem tipagem estática e gera *bytecode*.

Sobre o JavaFX,³ a Sun (2009, tradução nossa) descreve:

JavaFX fornece um modelo unificado desenvolvimento e implantação para a construção de aplicações cliente ricas que integram mídia imersiva rica, como áudio e vídeo, gráficos, texto rico e serviços Web. JavaFX permite aos desenvolvedores de criação programarem em um contexto visual, assim, ajudando-os a trazer suas idéias para a vida mais rápido e melhor.

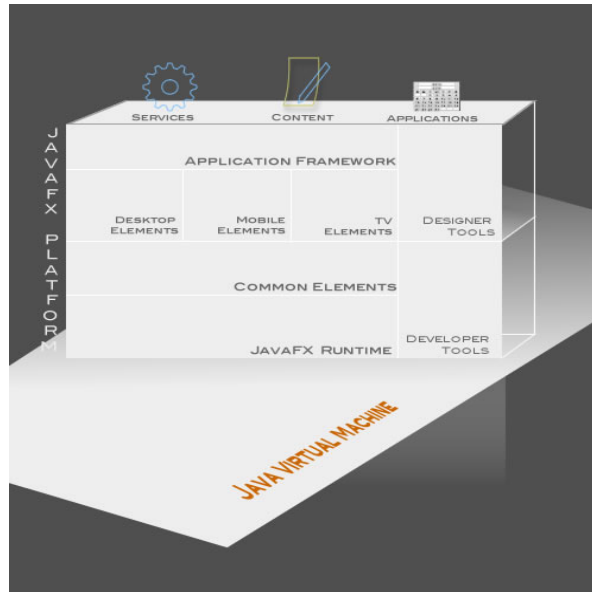
2.1 VANTAGENS

Conforme a Sun⁴ (2009) podem-se enumerar várias vantagens da linguagem JavaFX:

- a) facilita a junção de áudio, vídeo, animação, gráficos e texto. Os gráficos em 2D podem sofrer transformações em perspectiva 3D (gráficos 3D reais em breve);
- b) aspectos da animação avançada incluem *timeline*, animação por *keyframe*, e animação baseada em caminho;
- c) pode-se reutilizar bibliotecas Java, o que preserva investimentos feitos nesta tecnologia;
- d) instalação fácil;
- e) linguagem simples e declarativa.

2.2 ARQUITETURA

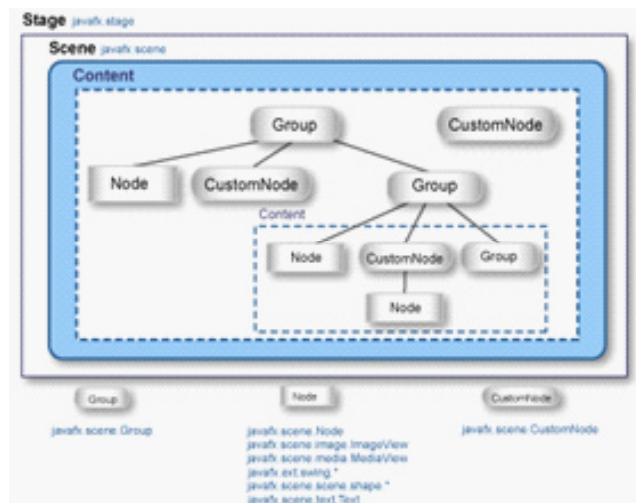
O Desenho 1 é uma representação gráfica da arquitetura de JavaFX, a qual é composta pela JavaFX Runtime (item 1), um Ambiente em Tempo de Execução específico para JavaFX que é baixado juntamente com a Java Runtime Environment (JRE). O item 2 representa a camada Common Elements (Elementos Comuns), que contém as Application Programming Interfaces (APIs) que são comuns a todas as plataformas, ou seja, se o código for desenvolvido usando apenas essas APIs, ele será executável em todos os perfis com uma JavaFX Runtime disponível. Sobre a Commons Elements, pode-se ver, destacado com os números 3, 4 e 5 as APIs específicas para Desktop, TV e dispositivos móveis. O item de número 6 representa o Framework que contém os blocos de código de aplicação; à direita, o Desenho 1 mostra, que lado a lado com essas tecnologias estão as ferramentas para desenvolvedores (item 7) e, também, as ferramentas para designers (item 8). Sobre essa plataforma é que rodam as aplicações JavaFX (item 9); todo esse conjunto roda sobre a Máquina Virtual Java (item 10).



Desenho 1: Arquitetura de JavaFX
Fonte: JavaFX Overview (2009b).

2.3 INTRODUÇÃO À LINGUAGEM JAVAFX SCRIPT

Segundo Doederlein (2009, p. 19-20), JavaFX lembra bastante Java; pode-se ver em um código-fonte com sintaxe e declarações conhecidas a um programador Java, como *class*, *import*, *package*, delimitação de bloco por chaves, etc., porém sua API acrescenta novos termos e conceitos que serão, posteriormente expostos, de forma resumida. Segundo Weaver et al. (2009, p. 99), a metáfora do JavaFX é um teatro; há um palco (Stage) e, dentro desse palco, uma ou mais cenas (Scene).



Desenho 2: Composição de um Stage
Fonte: Castilho et al. (2009).

- a) Stage – é o nível mais alto em um programa JavaFX como se vê no Desenho 2. Ele contém todos os Scenes (SUN, 2009). A Stage é que contém a interface com o usuário, ela pode ser implantada no Desktop (tendo uma barra de títulos e bordas ou não), em um navegador (como um applet dentro de uma área retangular) ou em dispositivos móveis (diretamente na tela do dispositivo) (WEAVER et al., 2009);
- b) Scene – uma superfície onde se pode “desenhar” conteúdo gráfico. JavaFX desenha tudo em um Scene e todos os objetos dentro de um Scene são objetos gráficos (descendem de Node); contém os Nodes (SUN, 2009);
- c) Custom Node – subclasse de Node, pode ter um Node filho (SUN, 2009);
- d) Group – sequência de Nodes agrupados (SUN, 2009);
- e) Node – é um elemento numa Scene (SUN, 2009).

Segundo Doederlein (2009, p. 40), em um primeiro contato com o código JavaFX Script, um programador Java5 logo percebe algumas diferenças, como a falta de uma declaração de classe, e a definição de funções, variáveis e sentenças espalhadas pelo escopo global do Script; o return é implícito na função; JavaFX utiliza sua própria sintaxe para instanciar novos objetos, chamada de Objeto Literal, como se pode observar no Quadro 1.

```
nomeDaClasse{
    propriedade1: valor
    propriedade2: valor
}
```

Quadro 1: Instanciação de Objeto

Os iniciadores são separados por vírgula, ponto e vírgula ou espaços. As variáveis e constantes são declaradas da maneira retratada no Quadro 2.

```
var valorInicial : Integer ;
def constante : String = "valor";
```

Quadro 2: Declaração de Variáveis

JavaFX conta com uma *Inference Engine*,⁶ que infere o tipo da variável ou constante em tempo de compilação, baseando-se na primeira atribuição de valor. A variável é estaticamente tipada (ANDERSON; ANDERSON, 2009, p. 44). Contudo, é perfeitamente possível determinar explicitamente o tipo, como no segundo ou terceiro exemplo. Nos casos em que o mecanismo de inferência não conseguir determinar qual o tipo da variável, o compilador avisará (ANDERSON; ANDERSON, 2009, p. 19).

2.3.1 Tipos de variáveis

Tipos primitivos de JavaFX têm sempre uma classe de Base e valores-padrão para quando uma variável é declarada sem um inicializador, não admitem null (exceto para tipos de Classe) (WEAVER et al., 2009, p. 46), e são tipos de valor, ou seja, para comparar duas Strings, pode-se usar o operador `==`, diferentemente de Java, onde a comparação usando esse operador compararia as referências ao objeto, em JavaFX é feita uma comparação de valores (DOEDERLEIN, 2009c).

- a) *Boolean* – armazena os valores *true* ou *false* (ANDERSON; ANDERSON, 2009, p. 61), é importante destacar que, diferentemente de Java, os operadores lógicos são escritos literalmente como *and*, *or* e *not* (WEAVER et al., 2009, p. 47);
- b) *Integer* – tem como Classe de base a classe *java.lang.Integer*, o valor padrão é 0;
- c) *Number* – tem como classe de base a classe *java.lang.Float*, o valor padrão é 0.0;
- d) *String* – tem como valor padrão uma *String* vazia, ou seja, `""`. Como os outros tipos de valor, *String* não aceita *null*. Pode ser usada tanto aspas duplas quanto aspas simples para delimitá-lo. A concatenação de duas *Strings* em sequência é automática, sem o uso do caractere de soma (+), ou de vírgula (.). Todos os métodos de *java.lang.String*, que é a classe de base de *String* podem ser usados (WEAVER et al., 2009, p. 55).
- e) *Duration* – é usado para representação de tempo, em animações; ele é criado com um *Integer* ou *number* sucedido por uma unidade de tempo sem espaço entre eles (WEAVER et al., 2009, p. 57). Ao contrário dos outros tipos primitivos de JavaFX que têm classes base em *java.lang.**, o tipo *Duration* tem como classe base uma classe *JavaFX*, *javafx.lang.Duration* (WEAVER et al., 2009, p. 57).
- f) *Sequences* – segundo Anderson e Anderson (2009, p. 21), são coleções de objetos que você pode acessar sequencialmente. O valor padrão de uma *sequence* é uma *sequence* vazia, e o tipo de uma *sequence* é o tipo dos seus elementos seguido por colchetes, um exemplo é *Boolean[]*. Não é possível ter elementos *null* no interior de uma *sequence* (WEAVER et al., 2009, p. 59).

2.3.2 Características especiais

- a) *Binding* – uma das técnicas mais poderosas de JavaFX (ANDERSON; ANDERSON, 2009, p. 31) é o recurso de *Binding*, em uma tradução livre, vinculação. Por meio da palavra-chave *bind* pode-se vincular uma variável a outra variável, a uma expressão ou função. Quando você atualiza algum valor da expressão a qual a variável foi vinculada, o valor do objeto dependente é atualizado, como é demonstrado no Quadro 3;

```
var altura = 4;
var base = 3;
function calculaHipotenusa(base, altura){
    return Math.sqrt(base*base + altura*altura);
};
var hipotenusa = bind calculaHipotenusa(base, altura);
println(hipotenusa);
base = 30;
println(hipotenusa);
altura = 40;
println(hipotenusa);
```

Quadro 3: Bind em funções

- b) Expressões de Bloco – conforme Anderson e Anderson (2009, p. 50), JavaFX é uma linguagem de expressão, o que significa que vários tipos de expressão em JavaFX retornam valores, que podem ser usados, por exemplo, em uma atribuição. Uma Expressão de Bloco nada mais é do que um conjunto de expressões delimitadas por colchetes ({}). O tipo e o valor do retorno são o tipo e o valor da última expressão do bloco.
- c) Expressão IF – também retorna valor, então pode-se associar o resultado de uma expressão *if* a uma variável, o retorno da expressão *if* é do mesmo tipo e valor do retorno da expressão *then* ou *else*, por exemplo;
- d) Expressão for – é usada para fazer alguma ação em um determinado número de vezes. Ele retorna uma *sequence* formada pelos retornos de cada iteração, essa *sequence* pode ser armazenada em uma variável, como se nota no Quadro 4;

```
var nomesDasPessoas = for (pessoa in pessoas) {
    "{pessoa.nome} {pessoa.sobrenome}";
}
```

Quadro 4: Abordagem JavaFX para for

- e) Expressões *while* – não retornam valor, elas atuam e têm sintaxe exatamente como no Java; palavra-chave *while*, seguida de uma expressão booleana entre parênteses e um bloco para ser executado enquanto a condição retornar *true*;
- f) Funções – funções em JavaFX são criadas pela palavra-chave *function*, seguida do nome da função e de parênteses que delimitam os parâmetros que porventura existam, e, por último, um bloco delimitado por chaves, como no Quadro 3;
- g) *Triggers* – segundo Weaver et al. (2009, p. 215) as *triggers* são uma parte opcional de uma declaração de variável. Ela é caracterizada pelo uso das palavras-chave *on* *replace* seguidas de um bloco de código a ser executado sempre que a variável mudar;
- h) *Event Handlers* – os *Nodes* de JavaFX contêm variáveis que armazenam funções que são chamadas quando determinado evento acontece, exemplos de *event handlers* são: *onMousePressed*, *onKeyTyped*, *onMouseEntered*.

4 CONCLUSÃO

Quando se falam em novas tecnologias, logo vem a mente conceitos, como modernidade, evolução, agilidade. Ao surgir nova tecnologia criada pela *Sun Microsystems*, a expectativa é ainda maior. JavaFX desapontou quem esperava uma ferramenta como o Adobe Flash, já que não conta com editor visual (até a data de elaboração deste artigo). JavaFX, nesses *releases* iniciais, deve um pouco, em quesitos, como controles de interface com o usuário, ferramenta de desenvolvimento visual e algumas incompatibilidades no Linux. Todavia, de maneira geral, percebe-se que o potencial da linguagem é grande, a sua sintaxe declarativa é interessante, pois traz uma visão hierárquica da construção da GUI, além do mais, apresenta recursos extremamente produtivos, como o *bind* e as *closures* (variáveis com tipo de função). JavaFX tem uma grande vantagem sobre os concorrentes, há milhares de máquinas virtuais Java instaladas.

As novas *JREs* a partir da versão 6u10 permitem arrastar um Applet para fora do browser, isso não é um recurso de JavaFX e sim do Java, mas quem mais se beneficiará é JavaFX, aliás, essa é a maneira mais fácil

de instalar um aplicativo, pois ela cria um atalho para o programa de forma transparente para o usuário. O desenvolvimento da aplicação é rápido, depois que se conhece a linguagem; as opções que se tem para a publicação são bem atraentes. Não há previsão para o próximo *release* de JavaFX, mas com certeza, ele é muito esperado por aqueles que já conhecem JavaFX.

Building rich interface applications with JavaFX

Abstract

This article presents an introduction to JavaFX Script language. Are presented main points of the language that takes advantage of all the baggage of Java. Knowledge of Java, is not required, but to who knows it, the main differences are highlighted. JavaFX is a language that and very easy to build interfaces, and like any Script language focuses on productivity. It's a recently launched language, but with huge potential. It runs on the Java virtual machine, a technology widespread, and leads to end the slogan of Java, which is: Write once, run anywhere, because you can develop a JavaFX code that will run unchanged in Cell Phones, TV's (Soon) and Desktop, whether through an applet, or through the Java protocol Web Start. The adoption of the technology will occur from the time when doubts about the quality pass. JavaFX is quick, concise, and very easy to learn, its declarative syntax, values the expression visual and facilitates the division of labor between developers and designers

Keywords: JavaFX. Scripting Language. RIA – Rich Internet Application.

Notas explicativas

¹Qualquer *software* instalado e executado em um computador.

²Ambiente separado em que uma aplicação roda com restrições (SINGH, 2004).

³ "JavaFX provides a unified development and deployment model for building rich client applications that integrate rich immersive media such as audio and video, graphics, rich text and web services. JavaFX allows creative developers to program in a visual context thus helping them to bring their ideas to life quicker and better." (SUN, 2009).

⁴ "JavaFX makes it easy to integrate video, audio, graphics, animation and rich text. Applications can combine audio/video and 2D graphics with 3D perspective transform (true 3D graphics coming soon) [...] Advanced animation features include timelines, keyframe animation, tweening, and path based animation [...]Java developers can preserve existing investment by reusing Java libraries in JavaFX applications [...]With Java SE

⁵ update 10 (and above) you can Drag-to-Install JavaFX applications simply by dragging them out of your browser - you can then relaunch the applications directly from your desktop [...]It is a simple, easy-to-learn, declarative scripting language."

⁶ Linguagem de Programação criada pela Sun Microsystems, roda em uma máquina virtual. Mecanismo que deduz o tipo da variável em tempo de compilação.

REFERÊNCIAS

ADOBE. **Rich Internet Applications**. 2009. Disponível em: <<http://www.adobe.com/devnet/projects/ria/>>. Acesso em: 6 nov. 2009

ANDERSON, Gail; ANDERSON, Paul. **Essential JavaFX**. Crawfordsville: Prentice Hall, 2009. 261 p.

CASTILLO, Cindy et al. JavaFX Media Browser Tutorial – Loading and Displaying Images, **Sun Microsystems**. 2009. Disponível em: <http://www.javafx.com/docs/tutorials/mediabrowser/module1_task1.jsp>. Acesso em: 19 out. 2009.

COSTA, Henry Franklin Dualibe da. **Internet Rica (RIA)**. 2008. Disponível em: <<http://www.henry.eti.br/pagina.php?IdPagina=258>>. Acesso em: 21 jun. 2009.

DOEDERLEIN, Osvaldo Pinali. Apresentando a JavaFX 1.0 e JavaFX Mobile, **Java Magazine**, Rio de Janeiro, ano 7, n. 67, maio 2009a.

_____. Começando a Programar em JavaFX, **Java Magazine**, Rio de Janeiro, ano 7, n. 68, jun. 2009b.

_____. JavaFX. Uma nova definição para linguagem dinâmica. **Java Magazine**, Rio de Janeiro, ano 7, n. 69, jul. 2009c.

_____. Uma “Aplicação Comum” em JavaFX. **Java Magazine**, Rio de Janeiro, ano 7, n. 72, out. 2009d.

JAVAFX FOR DESIGNERS E DEVELOPERS. **Sun Microsystems**. 2009a. Disponível em: <<http://www.javafx.com/about/at-a-glance.jsp>>. Acesso em: 17 fev. 2009.

JAVAFX OVERVIEW. **Sun Microsystems**. 2009b. Disponível em: <<http://www.javafx.com/about/overview/>>. Acesso em: 16 fev. 2009.

MACORATTI, José Carlos. **Desenvolvendo para Desktop ou para Web?** Disponível em: <http://www.macoratti.net/vbn_dkwb.htm>. Acesso em: 16 jun. 2009.

SINGH, Amit Sandboxing. **Kernelthread.com**. 2004. Disponível em: <<http://www.kernelthread.com/publications/security/sandboxing.html>>. Acesso em: 19 out. 2009.

SUN. **Learn About Java Technology**. 2009. Disponível em: <<http://www.java.com/en/about/>>. Acesso em: 23 jun. 2009.

WEAVER, James L. et al. **Pro JavaFX Platform: Script, Desktop and Mobile RIA with JavaTM Technology**. Berkeley: Apress Books, 2009. 543 p.

WEB DESIGN DEVELOPMENT INDIA. **Rich Internet Applications**. 2009. Disponível em: <<http://www.web-design-development-india.com/Rich-Internet-Applications.htm>>. Acesso em: 10 out. 2009.

WIKIPÉDIA. **Internet Rica**. 2009. Disponível em: <http://pt.wikipedia.org/wiki/Internet_rica>. Acesso em: 23 jun. 2009.