4º conahpa
Congresso Nacional de Ambientes Hipermídia para Aprendizagem
Florianópolis | Brasil | 5 a 7 de Novembro de 2009 |

ANÁLISE COMPARATIVA ENTRE OS PRINCIPAIS *FRAMEWORKS* DE DESENVOLVIMENTO JAVA

COMPARATIVE ANALYSIS BETWEEN MAIN DEVELOPMENT FRAMEWORKS IN JAVA

Erik Aceiro Antonio[1]
Universidade Estadual Paulista "Júlio de Mesquita Filho" (UNESP), Rio Claro, São Paulo,
Departamento de Estatística Matemática Aplicada e Computação (DEMAC).
Universidade Hermínio Ometto (UNIARARAS), Araras, SP.
Faculdade Anhanguera, Rio Claro, SP.

Milene Ferro[2]
Universidade Estadual Paulista "Júlio de Mesquita Filho" (UNESP), Rio Claro, SP, Instituto de Biociências, Centro de Estudos de Insetos Sociais (CEIS).

Resumo
A linguagem de programação Java tem sido utilizada com frequência e se tornado cada vez mais popular no mundo todo. Esse cenário revela que existe um conjunto de novas tecnologias que vem dando suporte a linguagem de programação Java, com especial destaque para as aplicações *server-side* e os sistemas de *frameworks.* Um *framework* é um tipo especializado de *software* que permite que outras aplicações reutilizem eficientemente componentes e objetos em aplicações para a Internet e em sistemas para *desktops.* Esse artigo mostra a estrutura de *frameworks,* em especial a arquitetura MVC (*Model-View-Controller*) e os principais prós e contras de cada *framework* para o desenvolvimento em linguagem Java, são apresentados também comparativos entre os *frameworks* JSF, Struts, Stripes, Wicket, Tapestry, Velocity, Vraptor e Hibernate.

Palavras-chave: frameworks, MVC, Java

[1] eaceiro@rc.unesp.br, aceiro@gmail.com
[2] milenef@gmail.com

Abstract
The Java programming language has been used frequently and become increasingly popular worldwide. This shows that there is a set of new technologies that have supported the programming language Java with particular highlight on the server-side applications and systems frameworks. A framework is a type of specialized software that allows other applications efficiently re-use of components and objects in applications for the Internet and desktops systems. This work shows the structure of frameworks in a special form the architecture MVC (Model-View-Controller) and the main pros and cons of each framework for development in Java, are also presented comparative analysis between the frameworks JSF, Struts, Stripes, Wicket, Tapestry, Velocity, Vraptor e Hibernate.

Keywords: *frameworks, MVC, Java*

1 Introdução

A linguagem de programação Java segundo o TIOBE *Programming Community Index* (TIOBE, 2009), é considera uma linguagem de programação utilizada cada vez com mais frequência no mundo, conforme levantamento a partir de 2002. Nesse cenário há um conjunto de novas tecnologias que vem dando suporte a linguagem de programação Java, com destaque em especial para as aplicações *server-side* - J2EE (Java 2 *Enterprise Edition*) (FLANAGAN, 2005; JENDROCK, *et al*., 2008).

Atualmente as aplicações para *web* estão cada vez mais movendo o código da camada de negócios para o lado do servidor (*server-side*) e suas funções *web* estão sendo expandidas para novos *framework*, assim são desenvolvidos novos *frameworks* visando suportar esse conjunto de novas tecnologias (KOFMAN, 2009).

Esse trabalho relata a estrutura de *frameworks*. São apresentadas também as principais vantagens e desvantagens do uso de cada *framework* para o desenvolvimento em linguagem Java.

Serão mostrados resultados comparativos da aplicação e do uso dos *frameworks* no mercado de trabalho brasileiro.

Os principais *frameworks* que serão mostrados são: JSF (SUN, 2009a), Struts (APACHE, 2009a), Stripes (STRIPES, 2009), Wicket (APACHE, 2009b), Tapestry (APACHE, 2009c), Velocity (APACHE, 2009d), Vraptor (VRAPTOR, 2009) e Hibernate (HIBERNATE, 2009).

1.1 Método de Análise

Foi utilizado o *site* TIOBE (http://www.tiobe.com) para mostrar o desempenho da linguagem Java ao longo dos anos de 2002 até 2009 (TIOBE, 2009).

Esse índice é medido como uma função dos *hits* dos principais *sites* de buscas existentes (Google, Google Blogs, MSN, Yahoo! e YouTube), sendo que a *query* utilizada nos sistemas de busca apresenta a seguinte sintaxe:

```
+"linguagem programming"                                    (1)
```

Após o envio da *query* para os sistemas de busca, no final da contagem das linguagens, o sistema TIOBE cria uma normalização para as 50 primeiras linguagens de programação encontradas, ou seja, as 50 linguagens juntas devem somar 100%. Valores que indicam falso positivo também são tratados manualmente para prevenir erros.

O resultado do TIOBE (2) permite avaliar a freqüência com que uma linguagem é utilizada a partir dos principais buscadores da *web*, também conhecidos como *web crawler* (GOOGLEGUIDE, 2009).

$$\frac{\sum_{i=1}^{n} \frac{\text{hits(PL,SE}i)}{\text{hits50(SE}i)}}{n} = \frac{\frac{\text{hits(PL,SE1)}}{\text{hits50(SE1)}} + \ldots + \frac{\text{hits(PL,SE}n)}{\text{hits50(SE}n)}}{n} \qquad (2)$$

onde,

N: Número de Search Engine
PL: Programming Language
SE: Search Engine

1.2 Linguagem de Programação Orientada a Objetos

A Linguagem Java cada vez mais vem se destacando como a principal linguagem de programação existente no mercado brasileiro e internacional. A Figura 1 mostra essa relação.

Position May 2009	Position May 2008	Delta in Position	Programming Language	Ratings May 2009	Delta May 2008	Status
1	1	=	Java	19.537%	-1.35%	A
2	2	=	C	16.128%	+0.62%	A
3	3	=	C++	11.068%	+0.26%	A
4	4	=	PHP	9.921%	-0.28%	A
5	5	=	(Visual) Basic	8.631%	-1.16%	A
6	7	↑	Python	5.548%	+0.65%	A
7	8	↑	C#	4.266%	+0.21%	A
8	9	↑	JavaScript	3.548%	+0.62%	A
9	6	↓↓↓	Perl	3.525%	-2.02%	A
10	10	=	Ruby	2.692%	+0.05%	A

Figura 1 - *Ranking* das linguagens de programação utilizadas com mais freqüência segundo o índice TIOBE (2009).

A partir da Figura 1 pode-se observar que esse cenário vem se mantendo equilibrado para a linguagem Java por mais de um ano. Os sinais em amarelo (=) indicam estabilidade no período, as setas para cima/baixo (verde/vermelho) ilustram se a

linguagem subiu ou desceu um nível no mesmo período. Todas as 10 principais linguagens mostram-se com status igual a *A* (*Active*).

Um resultado mais expressivo da popularidade da linguagem Java em todo o mundo, entre os anos de 2002 e 2009 é apresentado na Figura 2.

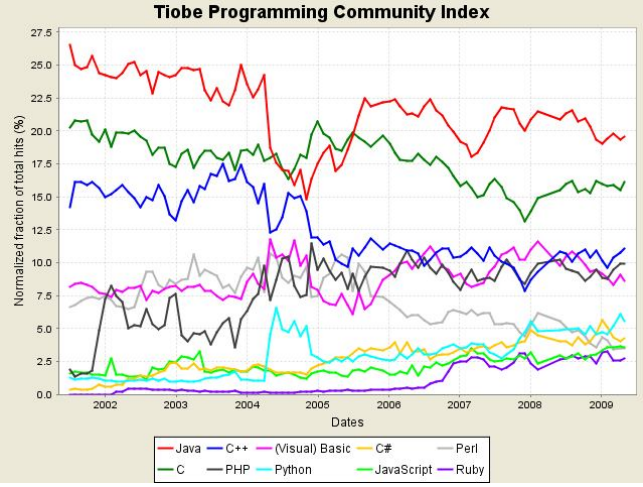

Figura 2 - Popularidade das linguagens de programação de 2002 a 2009 (TIOBE, 2009).

Observe no gráfico que temos uma prevalência do uso da linguagem Java entre 2002 e 2009, conforme apresentado em vermelho. Pensando nisso, torna-se muito relevante a identificação e reconhecimento das principais tecnologias existentes atualmente que norteiam a Linguagem Java.

2. Frameworks

Segundo Larman (2007) um *framework* é a representação de um conjunto de objetos extensíveis que possuem funções relacionadas.

A assinatura de um *framework* fornece um conjunto de pontos extensíveis (*core*) que podem ser plugados a um novo sistema pelo programador, esses pontos são estratégias diferentes para a reutilização de componentes, objetos e *portlets*. Temos como exemplos os pacotes AWT e Swing da linguagem Java. Esses *frameworks* fornecem muitas classes e interfaces, onde desenvolvedores podem estender essas classes e sobrepor métodos. Desenvolvedores também podem plugar eventos diferentes para os ouvintes (*listeners*) tais como os objetos do tipo `JButton`. Esses *listeners* são tratados segundo o padrão Observer-GoF (Gamma *et. al*, 1994).

JSF (SUN, 2009a)

Desenvolvido pelo comitê JCP (*Java Community Process*) com o *Java Specification Request* (JSR – 314), o padrão *JavaServer Faces* (JSF), disponibilizado pela Sun, fornece um conjunto de tecnologias para a construção de interface do usuário *Server side*, proporcionando um conjunto de ferramentas para a criação de aplicações mais fáceis. A versão atual do *JavaServer Face* é 2.1 de 3 de maio de 2009.

Os principais componentes do *framework* JSF são divididos em duas categorias:

- API
  - representa componentes da Interface do Usuário (UI)
  - ciclo de vida (estados)
  - *handle-event*
  - validação *server-side*
  - conversão de dados
  - definição da navegação de páginas
  - suporte a internacionalização
- *tags* customizadas
  - representa objetos em páginas JSP

Assim, o comitê JCP tenta proporcionar um conjunto de tecnologias baseado no padrão MVC para criar respostas da Interface Gráfica do Usuário (GUI) mais fácil via chamadas Ajax, minimizando a quantidade de código, gerando uma maior quantidade de processamento *server side*, diminuindo a quantidade de código JavaScript do lado do cliente e proporcionando modularidade para outras aplicações que se interpõem ao uso do JSF. A Figura 3 mostra a estrutura típica de uma arquitetura MVC Nível 2.

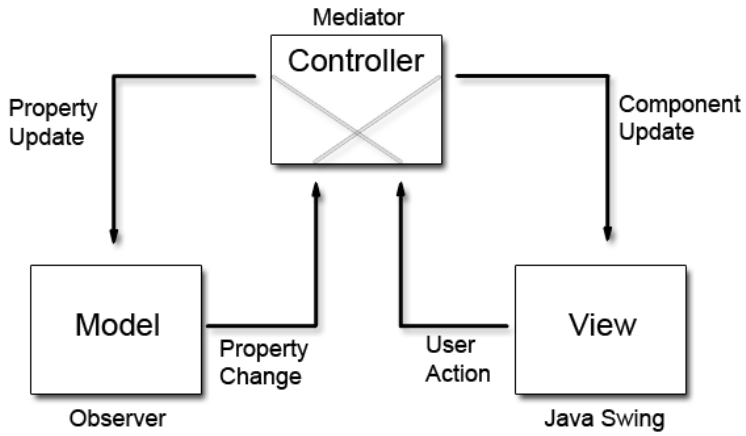

Figura 3 - Padrão MVC com controlador entre o Model e a View (SUN, 2009b).

Esse padrão tem como principal objetivo separar os interesses relativos da camada de apresentação da lógica e regras de negócios da aplicação. Nesse padrão o controlador recebe um evento disparado por uma GUI (*Graphic User Interface*) ou mensagens via métodos (GET/POST) do protocolo HTTP. A partir desse *post* o controlador processa a informação e armazena ou reencaminha para outro controlador. A atualização do modelo e da apresentação depende do controlador por meio de *forward*.

Struts (APACHE, 2009a)

Com a versão 2.1.6 de 13 de janeiro de 2009 a *Apache Software Foundation* (http://www.apache.org), vem mantendo um *framework open source* para desenvolvimento de aplicações *web*.

Aplicações *web* geralmente misturam páginas *web*, código de banco de dados, *design* de página e controle de fluxo, sendo que gerenciar a complexidade dessas aplicações tem se tornado cada vez mais difícil.

Uma maneira de se separar as partes do *software,* permitindo um controle maior é utilizar a arquitetura MVC (*Model-View-Controll*). O *Model* representa o negócio ou código do banco de dados, o *View* representa a página desenvolvida e o *Controller* representa o fluxo ou navegação. O *framework Struts* é desenvolvido para ajudar desenvolvedores a criarem suas páginas *web* utilizando o modelo MVC.

O *framework* fornece três pontos chaves:

- *Request* – manipulador (*handler*) fornecido pelo desenvolvedor da aplicação que é mapeado para a URI correspondente;
- *Response* – manipulador que transfere o controle para outro recurso;
- *Biblioteca de tags* – interativa troca de mensagens com o *form* e Server *pages*.

O *framework* trabalha bem com outras tecnologias como Ajax e SOAP.

Atualmente existem dois projetos: Struts 1 que é a melhor escolha para grupos de usuários que querem um *framework* para resolver problemas comuns e o projeto Struts 2 que era conhecido como WebWork 2 e atualmente fornecem o Struts 2.x, para equipes que necessitam de um *framework* para resolver problemas complexos com soluções sofisticadas.

Stripes (STRIPES, 2009)

Lançado em março de 2009 com a versão 1.5.1, o *framework* Stripes possui a licença *open source*. O projeto Stripes tem o objetivo de construir aplicações *web* para a plataforma Java eliminando a quantidade de configurações que outros *frameworks* como Struts, WebWork 2 e Spring-MVC requerem. Além de não ser necessário aprender toda uma nova linguagem para começar a desenvolver aplicações em Java.

Os principais objetivos do *framework* Stripes são: (1) Tornar o desenvolvimento Java para *web* mais fácil; (2) Fornecer ferramentas simples, ainda poderosas para a solução de problemas; (3) Fazer com o *framework* Stripes um desenvolvimento rápido em menos de 30 minutos; (4) Fazer isso apenas com a extensão do Stripes, sem configurações adicionais.

Características principais:

- Zero de configurações extras por pagina/ação (anotações usadas para descobrir os *ActionBeans*);
- Poderosa ferramenta de ligação;
- Fácil de usar os esquemas de conversão;
- Capacidade de reutilizar os *ActionBeans*;
- Construído para suportar múltiplos eventos.

Wicket (APACHE, 2009b)

Com a atual separação entre marcação/lógica, o *framework* Wicket tem como objetivo proporcionar o uso de POJO (*Plain Old Java Object*) para o desenvolvimento de aplicações *web*, tornando assim o desenvolvimento mais simples e agradável.

Páginas em Wicket são objetos Java puros fáceis de desenvolver, pois exigem o mínimo de conhecimento do *framework* para o desenvolvimento. O usuário pode trabalhar com o Wicket sem precisar de uma ferramenta para edição ou configuração de *tags* específicas.

Tapestry (APACHE, 2009c)

O Tapestry é um *framework* open source que tem como objetivo principal fornecer a desenvolvedores Java a capacidade de criação de páginas dinâmicas, sendo robusto e escalável. O Tapestry complementa a API *Java Servlet Container*.

A estrutura do Tapestry consiste em um conjunto de páginas que representam um componente. Possui a responsabilidade de acionar e disparar estados para clientes e servidores, validação de entrada de dados, localização, internacionalização e geração de relatório de exceções. A versão atual do *framework* é 5.1.0.5.

Velocity (APACHE, 2009d)

O Velocity é um *framework* de motor de *template* (modelo) para a linguagem Java. Permite que qualquer desenvolvedor possa usar uma linguagem simples, embora poderosa, linguagem de *template* para referenciar objetos definidos em código Java.

Usa o padrão MVC criando uma separação entre as camadas de apresentação desenvolvidas por *web designers* e a lógica de controle desenvolvida em Java. O Velocity separa o código das páginas *webs*, criando páginas *web* de mais fácil manutenção, viabilizando assim todo o ciclo de vida do *site*. Fornece também uma alternativa viável entre JSP (*Java Server Pages*) ou PHP.

Pode ser gerado a partir do framework *templates* para SQL, PostScript e XML. Pode ser utilizado como ferramenta *standalone* ou como um componente integrador de outros sistemas.

Hibernate (HIBERNATE, 2009)

O Hibernate é um *framework* que tem alto desempenho para persistência de objetos usando o modelo relacional e também serviços de consulta (*query*). O modelo de desenvolvimento do Hibernate deixa que o programador desenvolva a camada de persistência utilizando o paradigma orientado a objetos, incluindo associação, herança, polimorfismo, composição e *collections.* Hibernate proporciona ao desenvolvedor a capacidade de expressar *queries* em sua própria extensão de SQL (HQL), da mesma maneira que comandos em SQL, ou com um critério de orientação a objetos.

Diferente de outros *frameworks*, a proposta do Hibernate não é esconder o poder do SQL e sim fornecer uma maneira fácil de integrar suas aplicações orientadas a objetos em Java e .Net.

Mantido sobre projeto *open source* e licença LGPL está disponível atualmente com a versão 3.2.4 de 29 de janeiro de 2009.

VRaptor (VRAPTOR, 2009)

O *framework* VRaptor é um controlador MVC desenvolvido por alunos da Universidade de São Paulo (USP) para *web* focado no desenvolvimento ágil. Através da inversão de controle e injeção de dependências, ele elimina o tempo de trabalho que seria perdido com o código repetitivo: validações, conversões, direcionamentos e *lookups*.

Ele se adapta perfeitamente para trabalhar com o JSP na camada de apresentação e com Hibernate na persistência. Totalmente baseado em anotações do Java 5.0. As anotações garantem uma maneira simples de trabalhar com programação para a *web*.

3 Análise Comparativa entre os Frameworks

Foram feitas diversas amostras dos principais sistemas de *framework a* partir dos motores de *web crawlers* existentes. Essas análises comparativas foram feitas entre: Forum GUJ, Livros do Google Books, Catcho.com.

Forum GUJ

Para permitir identificar o nível de uso dos principais *frameworks* foram realizadas amostras do Forum Grupo de Usuários Java - GUJ (Figura 4).

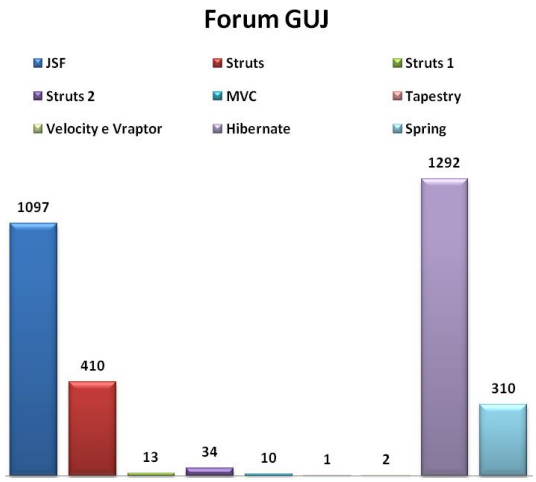

Figura 4 - Taxa de uso do GUJ para os *frameworks*.

Esse gráfico apresenta a taxa com a qual os usuários (Desenvolvedores Java) estão utilizando os principais *frameworks*. Pode-se observar como resultado mais significativo para um grupo total de 3.169 mensagens postadas, aproximadamente 35% de todo o volume de mensagens enviadas corresponde ao *framework* JSF. O

framework Hibernate aparece na lista com uma taxa de aproximadamente 45% das mensagens sendo enviadas. Isso mostra a grande preocupação dos usuários em trabalhar com tipos diferentes de banco de dados, além da acentuada busca por um *framework* mais flexível como o JSF.

É importante ressaltar que os *frameworks* Stripes e Wicket não chegaram a receber pontuação. Esses resultados mostram que os usuários ainda desconhecem esses tipos de *frameworks*. Outro ponto importante é a pontuação do *framework* Tapestry, que possuía apenas uma única mensagem. Essa mensagem mostrava a preocupação do desenvolvedor em mostrar para a comunidade a existência do *framework* Tapestry.

Livros do Google Books

Foram realizadas pesquisas a partir do *site* http://books.google.com, buscando por literaturas em inglês, a partir da ferramenta de busca e compartilhamento do Google. A Figura 5 mostra os resultados dessa avaliação.

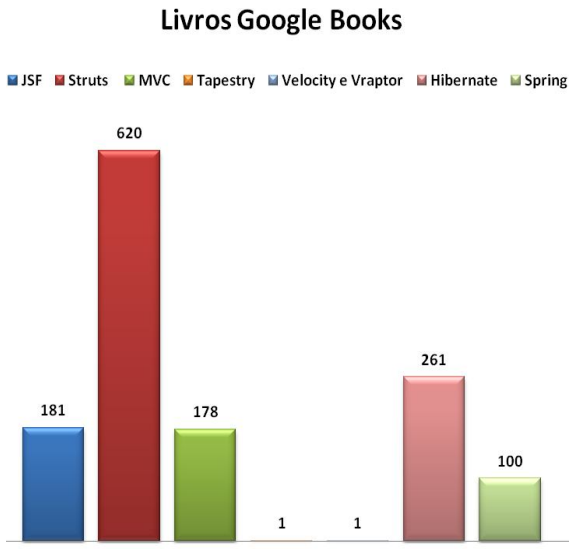

Figura 5 - Taxa de Livros produzidos para os *frameworks*.

Com a Figura 5 observa-se a grande quantidade de literatura já produzida para o *framework* Struts, assim do total (1.342), 46% da literatura corresponde ao Struts. Essa amostra indica livros completos ou apenas parte de livros como complemento de um guia de referência. A quantidade de livros produzidos para o framework Tapestry, Wicket e Stripes não corresponde a 1% de todo o volume pesquisado. Essa pesquisa foi realizada utilizando-se a *query* de busca do Google configurada para:

```
allintitle: “NomeDoFramework”
```

Empregos na catho.com.br

Um outro tipo de pesquisa realizada, foi uma busca no *site* da Catho, que mapeia a demanda por empregos em cada área do desenvolvimento Java, para os principais *frameworks* existentes no mercado (Figura 6).

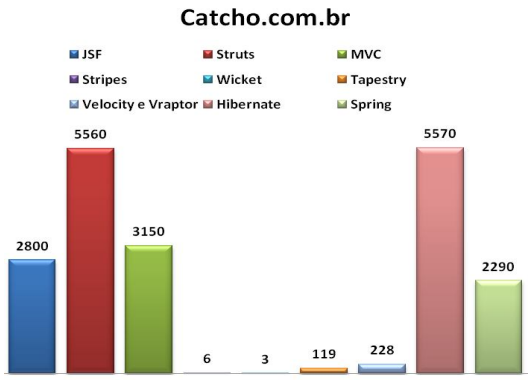

Figura 6 - Taxa de empregos para desenvolvedores Java nos principais *frameworks* (Fonte: catho.com.br).

Essa análise ilustra que para um total de 19.726 empregos para programadores em todo o Brasil, 14% são para desenvolvedores JSF e 28% são para desenvolvedores Struts e Hibernate. Embora MVC não seja relativamente um *framework*, a pesquisa mostra a preocupação das empresas em possuírem ambientes de desenvolvimento que utilizem *frameworks* que tenham o padrão MVC, permitindo assim a flexibilidade entre diferentes tipos de camadas de apresentação, modelos e banco de dados, seja um SGBD (Sistema de Gerenciamento de Banco de Dados) ou um sistema embarcado para persistência via XML.
Essa pesquisa foi realizada com a seguinte query de consulta no Google:

```
vaga NomeDoFramework +programador inurl:catho.com
```

4. Conclusão

O presente trabalho permitiu constatar a importância dos sistemas de *web crawler* para a busca e análise do uso de novas tecnologias que oferecem suporte a linguagem de programação Java. Além disso, proporcionou resultados qualitativos e quantitativos a respeito dos principais *frameworks* disponíveis.

Pode-se notar dentre os *frameworks* apresentados, que a tecnologia JSF embora recente, vem recebendo atenção tanto da comunidade Java como do mercado de trabalho. Essa tecnologia assim como Wicket e Striper, vem mostrando acentuado interesse em encontrar o melhor parâmetro que caracterize a razão configuração por página, que ainda se mostra elevado nos *frameworks* Struts e Spring-MVC. Os dados apresentados revelam também que a grande parte de usuários e empresas brasileiras estão dando atenção ao uso de padrões MVC (*Model-View-Controller*) e também ao uso de padrões flexíveis para bancos de dados.

5 Referências Bibliográficas

APACHE, Struts. Disponível em: <http://struts.apache.org/>. Acesso em 11 de maio de 2009.

_____, Wicket. Disponível em: <http://wicket.apache.org/>. Acesso em 11 de maio de 2009.

_____, Tapestry. Disponível em: <http://tapestry.apache.org/>. Acesso em 11 de maio de 2009.

_____, Velocity. Disponível em: <http://velocity.apache.org/moving.html>. Acesso em 11 de maio de 2009.

Flanagan, D. Java in a nutshell. 5 ed. Editora O'Reilly, 2005, 1224 páginas.

GOOGLEGUIDE. Disponível em: <http://www.googleguide.com/google_works.html>. Acesso em 29 de jun. de 2009.

Gamma, E., Helm, R., Johnson, R., Vlissides, J. M. Design Patterns: Elements of Reusable Object-Oriented Software (Hardcover). Addison-Wesley, 1994.

HIBERNATE,Hibernate. Disponível em: <https://www.hibernate.org/>. Acesso em 11 de maio de 2009.

Jendrock, E., Ball, J., Carson D., Evans, I., Fordin, S., Haase, S. The Java EE 5 Tutorial For Sun Java System Application Server 9.1. out. 2008. Disponível em: <http://java.sun.com/javaee/5/docs/tutorial/doc/index.html>. Acesso em 11 de maio. 2009.

Kofman V. Modern Java Frameworks for Web Development. Disponível em: <http://www.developer.com/design/article.php/10925_3523506_1>. Acesso em 11 de maio de 2009.

SUN. Java ServerFaces (JSF). Disponível em: <http://java.sun.com/javaee/javaserverfaces/>. Acesso em 11 de maio de 2009.

_____, Java SE Application Design With MVC Robert Eckstein, Março de 2007. Disponívem em: <http://java.sun.com/developer/technicalArticles/javase/mvc/>. Acesso em 29 de junto de 2009.

TIOBE. TIOBE Programming Community Index. Disponível em: <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>. Acesso em 11 de maio de 2009.

VRAPTOR, Vraptor. Disponível em: <http://www.vraptor.org/>. Acesso em 11 de maio de 2009.