

# **Um estudo comparativo entre os Frameworks Struts e JavaServer Faces**

**Anderson Silva dos Santos**

Faculdade Fortium

Curso de Pós-graduação *Lato Sensu* em Software Livre  
SRTVN, Edifício Brasília Rádio Center, 1º subsolo,  
70719-900 - Brasília/DF

[anderrsson@gmail.com](mailto:anderrsson@gmail.com)

Brasília, DF, 26 de Agosto de 2009

**Resumo.** A adoção de uma estrutura de desenvolvimento de software para internet baseada em camadas tem sido de fundamental relevância para o sucesso dos projetos web existentes e em desenvolvimento. O uso de frameworks que tenham a preocupação de se ater a um padrão de projeto já institucionalizado definido com Model View Control para simplificar a criação de aplicações web java, tem sido de grande relevância, sendo assim, fazendo uso dessa arquitetura temos os frameworks Struts e JavaServer Faces. O objetivo é comparar para esclarecer e orientar na escolha de acordo com as particularidades de cada um dos frameworks selecionados. Adaptando assim a real necessidade de negócio das empresas que se propõe a fazer uso de tais tecnologias.

**Abstract.** The adoption of a software development framework for Internet based on layers have been of fundamental relevance for the success of existing web projects and the ones that are being developed. The usage of frameworks that concern about sticking to a pattern project already institutionalized and defined as a Model View Control to simplify the creation of java web applications, has been of great importance, therefore, using this architecture we have the Strut and JavaServer Faces frameworks. The objective is compare to clarify and guide the choice according to the particularities of each of the selected frameworks. Adapting the real business needs of companies that aims to make use of such technologies.

## **1. Introdução**

A utilização de estruturas de desenvolvimento de software para web baseadas em camadas tem um índice representativo para os desenvolvedores de aplicações.

As simplificações por estruturas semelhantes representam a redução dos esforços aplicados na abstração de cada projeto envolvido. O padrão Model View Control (MVC), por se tratar de um padrão mundialmente aceito pela comunidade de desenvolvedores tem motivado a criação de diversos frameworks baseados em sua forma estrutural, sendo os mais conhecidos: Struts, JSF, Spring, Jbanana (brasileiro), Webwork, entre outros. O quantitativo de frameworks no mercado é elevado, assim faz com que a escolha do usuário dele seja cautelosa e criteriosa, para que encontrem um que se enquadre com a necessidade específica do projeto em desenvolvimento, observando os fatores positivos e negativos de cada um deles.

Dentre os frameworks citados acima, destacam-se dois: Apache Struts Web Application Framework (Struts), mantido pela Apache Software Foundation (ASF), que é considerado o mais maduro e um denominado JavaServer Faces, mantido pela Sun Microsystems; é admirado pela comunidade de desenvolvedores que consideram sua proposta um resgate ao desenvolvimento de interfaces baseadas em componentes e eventos.

## **2. Conceitos Básicos**

### **2.1. Framework**

“Um framework é considerado como uma solução para conjunto de problemas em comum, com uso de classes e interfaces, que disponibilizam objetos com capacidade de capturar funcionalidades comuns a várias aplicações. Assim sendo, um framework pode ser considerado, sob certo ponto de vista, como uma solução quase completa” [Linhadecodigo, 2009]<sup>1</sup>. “Framework é um conjunto de classes que colaboram para realizar uma responsabilidade para um domínio de um subsistema da aplicação” [FAYAD e SCHMIDT, 1997 ].

### **2.2. Model View Control**

A divisão das responsabilidades de um projeto por afinidade apresenta claramente as bases do MVC, sendo que se tem a camada de apresentação (interface da aplicação com seu mundo exterior, aquilo com que o usuário ou aplicação se depara para interagir com o software), negócio (responsável em implementar as particularidades do negócio que esta sendo executado) e controle (relacionado ao fluxo com que as ações serão executadas), estrutura que é trivial para os sistemas de informações. Quando se profere sistema de internet, há a necessidade de atenção e cuidado no que se refere o controle de fluxo, pois, quando não é gerado de forma adequada impossibilita o gerenciamento da aplicação. Portanto, o MVC, é um padrão de arquitetura que organiza a separação entre as camadas de negócio, apresentação e controle da aplicação.

---

<sup>1</sup> Disponível em: <<http://www.linhadecodigo.com.br>>. Acesso em: 01 jun. 2009.

Benefícios do uso de camadas (MVC):

- Controle do Fluxo e lógica do negócio – neste ponto há um arquivo que centraliza todo o controle de onde vem e para onde vão todas as requisições correspondentes ao contexto da aplicação. Aqui cada request conhece ser respectivo response.
- Reduz duplicidade de código fonte – a medida que os beans, páginas, properties, etc, vão sendo criados, o reaproveitamento do que já foi escrito é muito maior pelo fato de a aplicação ser modularizada, onde os componentes criam os objetos maiores.
- Centraliza controle – aplicações com essa estrutura detêm um controle de fluxo bastante desenvolvido, e sendo assim, sempre que ocorre um problema inesperado, não previsto e etc, a aplicação ou container, toma a atitude mais correta para o fluxo que está sendo executado.
- Aplicação mais robusta – sistemas mais confiáveis, por seguirem um padrão de projeto já institucionalizado como trivial pelo mercado de desenvolvedores. Menor número de incidentes, erros e etc.
- Portável – aqui o programa não tem uma ligação forte na plataforma em que foi desenvolvido, podendo operar com os mesmos resultados rodando sobre Linux, MacOS, Unix e etc.
- Reduz a complexidade – agrupam os componentes que tem características comuns e assim simplificam a complexidade entre eles.
- Reusabilidade – componentes, módulos, camadas, são utilizados em outros projetos, promovendo um ganho de tempo extraordinário no desenvolvimento de software.
- Padrão arquitetural – sendo uma estrutura de software conhecida e difundida entre a equipe de desenvolvedores facilita a comunicação e entendimento dos projetos.

Desvantagens do uso do MVC:

- Limitação tecnológica – algumas regras tecnológicas são quebradas por restrição tecnológica.
- Complicam – quando o tratamento tem a finalidade de resolver um problema em um sistema simples a utilização de camadas complica ou mesmo inviabiliza um estudo para o caso em questão.
- Overdose – muitos arquitetos de software com suas mentes criativas criam diversas camadas fazendo com que uma aplicação relativamente simples se torne altamente complexas pelo abuso de camadas.

A Figura 1 apresenta o modelo proposto pela estrutura MVC.

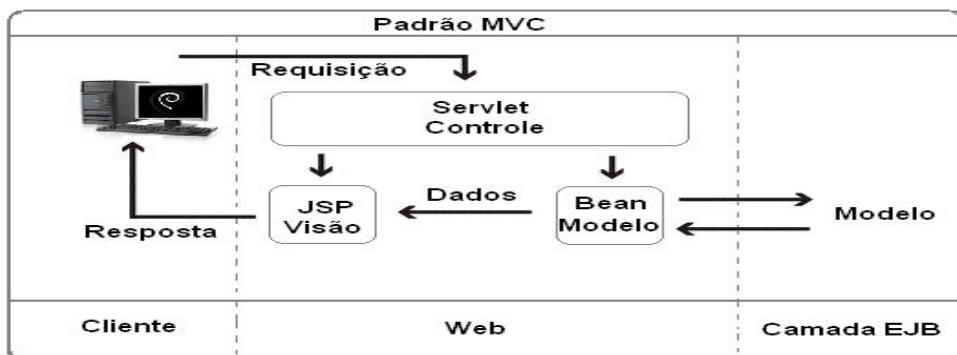


Figura 1 – Abstração utilizada na arquitetura MVC. Fonte : [Plugmasters, 2009]<sup>2</sup>

### 3. Apache Struts Web Application Framework

O projeto do framework Struts iniciou-se em maio de 2000 por Craig R. McClanahan, com o objetivo de constituir uma implementação MVC padrão para a comunidade Java. Em julho de 2001, a versão 1.0 foi lançada e em pouco tempo o objetivo foi alcançado, há hoje inúmeras aplicações baseadas neste framework. Atualmente, os engenheiros da Apache Software Foundation trabalham na versão 2 do Struts, o que mostra que, ao contrário do que muitos imaginaram, o projeto do Struts não vai estagnar.

Com base na arquitetura MVC, as aplicações que aderem ao Struts são divididas em três camadas distintas: componentes de controle, componentes de visão, incluindo um servlet e um processador de requisições; JavaServer Pages e Tag Libraries da camada de visão; e componentes de negócios e de acesso a dados da camada de modelo. A Figura 2 ilustra a arquitetura de uma aplicação MVC baseada no Struts.

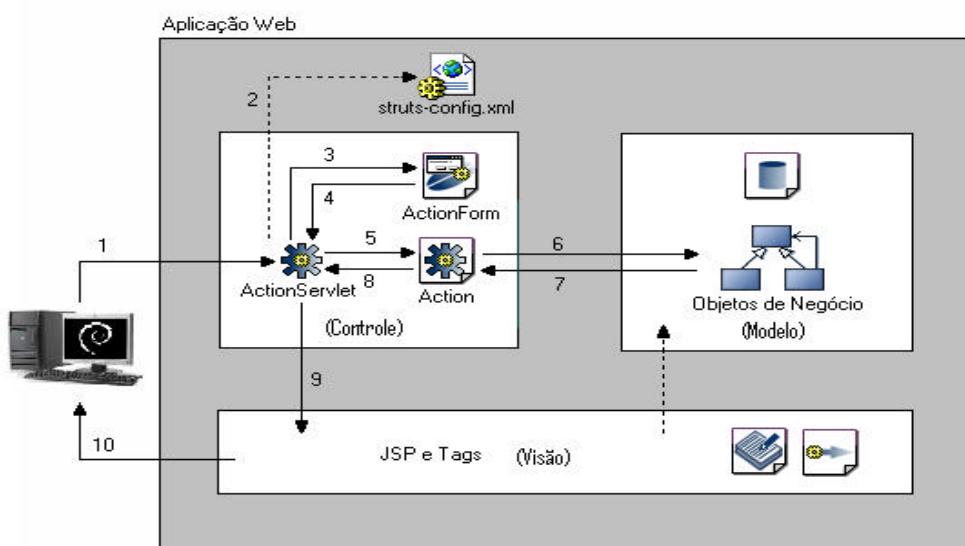


Figura 2 – Arquitetura Struts seguindo o padrão de projeto MVC. Fonte: [J2ee, 2009]<sup>3</sup>.

<sup>2</sup> Disponível em: <<http://www.plugmasters.com.br/>>. Acesso em: 04 abr. 2009.

<sup>3</sup> Disponível em: <<http://www.j2eebrasil.com.br/mostrar/32>>. Acesso em: 11 abr. 2009.

Entretanto, o Struts não implementa somente o padrão MVC, ele adere um conjunto de outros padrões também consagrados no mercado, que trazem inúmeros benefícios para a comunidade de desenvolvedores em seus projetos.

- Front Controller – padrão responsável em centralizar a lógica de processamento que ocorre na camada de apresentação.
- Command – padrão de projeto que encapsula as requisições com um objeto, desta maneira permite que sejam parametrizado clientes de diferentes requisições.
- Dispatcher – padrão de projeto que utiliza ou encapsula objetos.
- View Helper – padrão de projeto que está relacionado a formatação da apresentação dos componentes auxiliares.

### 3.1. Controlador Struts

Em conformidade a um padrão muito utilizado o Front Controller deve existir um mecanismo de centralização dos controles em uma aplicação MVC, dando assim fluxo e rumo para as requisições que são feitas pelos aplicativos web. No Struts, o ator mais importante sobre esse controle é um servlet chamado `org.apache.action.ActionServlet`. O Action Servlet quando iniciado efetua a leitura do arquivo chamado `struts-config.xml` que é onde estão todos os mapeamentos (na ocorrência de determinado evento tome a postura 1, na ocorrência de outro evento tome a postura 2) que direcionam as solicitações em ações da aplicação. Tecnicamente demonstrando, as solicitações são feitas ao ActionServlet graças ao mapeamento `servlet-mapping` definido no arquivo descriptor da aplicação, o `web.xml`. No Struts é comum o uso do conhecido \*.do lembrando que isso é uma convenção não uma regra obrigatória.

## 4. Framework JavaServer Face

Em maio de 2001 um grupo de empresas reuniu-se no Java Community Process (O processo JCP utiliza os *Java Specification Requests* (JSRs), que são documentos formais que descrevem as especificações propostas e tecnologias que pretendem adicionar na plataforma Java. São conduzidas revisões públicas e formais antes que um JSR se torne *final* e seja votado pelo comitê executivo do JCP. Um JSR que chega ao status final torna-se uma implementação de referência que provê uma versão livre da tecnologia na forma de código fonte e um Technology Compatibility Kit que permite a verificação da API especificada) com a finalidade de definir uma arquitetura de desenvolvimento de interfaces de usuário processadas no servidor, e assim foi dado o nome de JavaServer Faces. No primeiro dia do mês de março de 2004, a JSR 127 foi aprovada com 100% de aceitação entre os votantes, conforme Figura 3.

Apache Software Foundation	<input checked="" type="checkbox"/>	Apple Computer, Inc.	<input type="checkbox"/>	BEA Systems	<input checked="" type="checkbox"/>	Borland Software Corporation	<input checked="" type="checkbox"/>
Caldera Systems	<input checked="" type="checkbox"/>	Fujitsu Limited	<input checked="" type="checkbox"/>	Hewlett-Packard	<input checked="" type="checkbox"/>	IBM	<input checked="" type="checkbox"/>
IONA Technologies PLC	<input checked="" type="checkbox"/>	Lea, Doug	<input checked="" type="checkbox"/>	Macromedia, Inc.	<input checked="" type="checkbox"/>	Monson-Haefel, Richard	<input checked="" type="checkbox"/>
Nokia Networks	<input checked="" type="checkbox"/>	Oracle	<input checked="" type="checkbox"/>	SAP AG	<input checked="" type="checkbox"/>	Sun Microsystems, Inc.	<input checked="" type="checkbox"/>
Legenda: Sim: <input checked="" type="checkbox"/> Não <input checked="" type="checkbox"/> Abstenção: <input type="checkbox"/> Não votou: <input type="checkbox"/>							

Figura 3 – Resultado da votação da JSR-127. Fonte: [JCP, 2009]<sup>4</sup>.

Dentro das empresas que foram supracitados, pode-se observar claramente que o pensamento sobre os problemas e soluções apresentados são sanados pela proposta oferecida pelo framework, assim tornando-se reconhecido pelos magnatas da área.

A intenção do framework JSF é adicionar facilidades no desenvolvimento de interfaces a grande plataforma Java 2 Platform, Enterprise Edition (*J2EE*). O framework JSF simplifica o desenvolvimento de grandes fluxos de interfaces com o usuário, com seu modelo de componentes e eventos ligado a um eficiente ciclo de processamento de requisições.

Principais elementos de JSF:

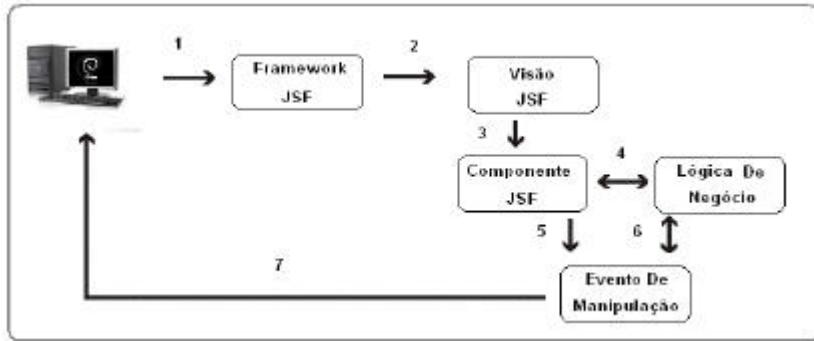
- Um conjunto de componentes de interface e uma API para extensão e criação de novos componentes, gerenciamento de seu estado e de seus eventos.
- Duas tag libraries com componentes de interface Web e de manipulação de objetos no lado servidor.

Para que esse framework seja realmente útil, deve ser utilizado com algum ambiente integrado de desenvolvimento (IDE) que promova a interação com seus componentes e arquivos de configuração. O seu uso em uma boa IDE faz com que a produtividade tenha um ganho muito significativo, um aumento na manipulação dos componentes que o framework dispõe para a criação das interfaces web.

JSF tem também sua arquitetura baseada no padrão MVC de forma similar ao Swing e de outros frameworks de interface gráfica para desktop. A Figura 4 demonstra os elementos JSF na arquitetura MVC.

---

<sup>4</sup> Disponível em: <<http://jcp.org/>>. Acesso em: 16 mai. 2009.



**Figura 4 – JavaServer Faces na Arquitetura MVC** Fonte: [SUNNYWEAR, 2009]<sup>5</sup>.

Na camada de visão como JSF, os componentes são produzidos por classes chamadas Renderers, para a apresentação em um mecanismo de saída. Um grupo de Renderers juntos formam um RenderKit. A implementação JSF de referência inclui somente um RenderKit voltados para clientes, o qual fabrica tags HTML para cada um dos componentes JSF. No decorrer do tempo, novos RenderKits serão desenvolvidos, permitindo que os componentes JSF possam ser convertidos para outras linguagens de marcação como WML e XML.

## 5. Realização da escolha

Observando os dois frameworks, pode-se assegurar que ambos contemplam as operações básicas de navegação, validação e internacionalização. O Struts assume um número bem maior de aplicativos na web por ser mais maduro. O JSF destaca-se pelo seu modelo de componentes de interface e eventos processados no lado servidor.

Para que interfaces de usuário complexas sejam desenvolvidas nas tecnologias tradicionais, como Servlets, JSPs e até mesmo Struts, faz-se necessária a implementação de bastantes códigos nas páginas, a fim de apresentar os dados e tratar os eventos do usuário. Em uma implementação JSF, os elementos de interface são representados por componentes padronizados e as ações do usuário são representadas por eventos manipulados pela aplicação no lado servidor. Este tipo de modelo de componentes baseado em eventos facilitou o desenvolvimento de complexas interfaces de aplicações desktop durante os últimos anos e pode também simplificar o desenvolvimento Web [Plugmasters, 2009]<sup>6</sup>.

Alguns autores defendem uma utilização híbrida do Apache Struts e do JSF, como se eles fossem complementares. Segundo eles, um cenário ideal seria utilizar componentes de interface JSF em aplicações Struts. Entretanto, o JSF possui suporte a todas as etapas existentes desde a origem da requisição até a geração do conteúdo de resposta. Uma abordagem híbrida, na qual JSF e Struts trabalham de forma complementar, pode ser muito interessante para a migração gradual de aplicações Struts para a tecnologia JSF, mas para o desenvolvimento de novas aplicações, em geral, o JSF é mais recomendado. A migração de Struts para JSF pode começar pela substituição das tags Struts HTML por componentes JSF, com o auxílio do projeto struts-

<sup>5</sup> Disponível em: <<http://www.sunnywear.org/>>. Acesso em: 04 jun. 2009.

<sup>6</sup> Disponível em: <<http://www.plugmasters.com.br/>>. Acesso em: 13 jun. 2009.

faces[Plugmasters, 2009]<sup>7</sup>.

### 5.1. Maturidade

O Struts é um framework J2EE desde 2000 e alcançou um elevado grau de maturidade. Existem milhares de desenvolvedores espalhados pelo mundo, sem falar na quantidade de livros, apostilas, listas e dezenas de artigos disponíveis. As IDEs mais consagradas e importantes do mercado suportam esse framework. O JSF surgiu em 2004, mais novo, mas tem a vantagem de ser o primeiro framework J2EE que contempla as especificações incluídas na última versão J2EE, que exige que funcione em todos os servidores. No entanto, os resultados ainda não são conhecidos de forma maciça, a tendência é a superação. Sendo assim, o Struts é o mais maduro dos dois frameworks, teve grande aceitação e continua a evoluir em Struts 1 e Struts 2.

### 5.2. Flexibilidade e Controle com Fluxo

Tanto o Struts quanto o JSF implantam um controlador padrão centralizado no tratamento de suas requisições realizadas pelos clientes. O JSF adiciona muitos benefícios para o controlador e podendo assim oferecer a capacidade de lidar com múltiplos eventos em uma página, enquanto, o struts pode apenas tratar um por página.

### 5.3. Validações

No Struts a validação é feita pelos ActionForm e de forma completa do objeto, ou seja, de todos os atributos que o formulário conter. No que tangem o conversão de dados é utilizado o padrão JavaBeans. Por outro lado o JSF irá validar cada componente de forma individual do formulário, mas pode também validar usando validadores de padrões definidos (casos mais genéricos).

### 5.4. Navegabilidade

A naveabilidade possui características fundamentais em ambos os frameworks. Em ambos os casos tem-se um modelo de navegação declaratória que ficam contidas dentro de uma arquivo de configuração XML. Existem dois tipos de navegação, a estática e a dinâmica na primeira quando redirecionar diretamente para a próxima pagina e na segunda quando houver necessidade da realização de alguma ação ou lógica de negocio para a partir daí ir para a página. O Struts e o JSF suportam os dois tipos de navegação. No Struts há os ActionForward que são objetos que definem os locais por onde vão passar as requisições e mapeamentos elaborados. No JSF é feito através de objetos listeners que ficam ouvindo os eventos. Estes listeners são responsáveis pelos tratamentos e retorno lógico, utilizado pela aplicação para apresentar suas paginas.

---

<sup>7</sup> Disponível em: <<http://www.plugmasters.com.br/>>. Acesso em: 13 jun. 2009.

## **6. Conclusão**

Destarte, não existe um molde para que um framework MVC seja selecionando, entretanto, podem-se observar algumas das particularidades que podem influenciar na escolha. Em termos de aplicações é possível construir aplicações que utilizem tanto o framework Struts quanto o JSF, em ambos os casos há uma estrutura bem definida que satisfaz o negócio das organizações. Para aplicações que foram feitas utilizando o Struts ou que estejam em processo de conclusão do projeto ou ainda que dependam de manutenção recomenda-se que seja mantida (mudanças de framework no meio do projeto acarretaria diminuição na produtividade da equipe de desenvolvimento e assim um possível atraso no cronograma) e acompanhada as evoluções do framework pelo mantenedor, no caso do Struts, Apache Software Foundation. O JSF tem inúmeros motivos para ser escolhido, possui estrutura extensível e flexível, a quantidade de componentes na camada de visualização é superior e mais fáceis de trabalhar, atualizada. Existem linhas de estudo que acreditam que a melhor opção seja o uso simultâneo, onde o Struts faria uso dos recursos da interface do JSF (aumento considerável da complexidade). A decisão deve considerar um conjunto de fatores na escolha de um framework, sendo eles: equipe (pessoas capacitadas com a tecnologia utilizada), cronograma (prazos que possam ser seguidos), recurso (capital para execução de todo o projeto) para que não seja um projeto mal sucedido.

## **7. Referências Bibliográficas**

- Apache Struts. Struts. ?. Disponível em: <<http://struts.apache.org/>>. Acesso em: 04 abr. 2009.
- DFJUG. **Struts e JSF**. ?. Disponível em: <<http://www.djug.org/>>. Acesso em: 13 jun. 2009.
- FAYAD, M. E. & SCHIMIDT, D. C. (1997): **Object-oriented application frameworks**, Communications of the ACM, NY 1997.
- GEARY, David, HORSTAMNN, Cay. **Core JavaServer Faces**. Tradução: Marcelo Soares. Rio de Janeiro 2005.
- HUSTED, Ted. **Struts em Ação**. Tradução: Eveline Vieira Machado. 1.ed. Rio de Janeiro : Ciência Moderna, 2004.
- J2EE Brasil. JSF. ?. Disponível em: <<http://www.j2eebrasil.com.br/>>. Acesso em: 06 jun. 2009.
- Java Community Process. JCP. ?. Disponível em: <<http://jcp.org/>>. Acesso em: 09 mai. 2009.
- Java Platform, Enterprise Edition (Java EE). ?. Disponível em: <<http://java.sun.com/javaee/index.jsp>>. Acesso em: 13 mai. 2009.
- NASCIMENTO, Givanildo Santana. **Apache Struts ou JavaServer Faces**. ?. Disponível em: <<http://www.plugmasters.com.br/>>. Acesso: 21 mai. 2009.

ORACLE Technology Network. **Tutorial: Building a Web Application with Struts.** ?. Disponível em <[http://otn.oracle.com/sample\\_code/tutorials/bc4jvsrm/struts/toc.htm](http://otn.oracle.com/sample_code/tutorials/bc4jvsrm/struts/toc.htm)>. Acesso em: 03 jun. 2009.

KURNIAWAN, Budi. **Struts 2 Projeto e Programação.** 1.ed. Rio de Janeiro : Ciência Moderna, 2008.

PRESSMAN, Roger S. Engenharia de software. Tradução: Rosângela Delloso Penteado. 6<sup>a</sup> edição, São Paulo, 2006.

Sun Microsystems. JSF and Struts. ?. Disponível em: <<http://www.sun.com/>>. Acesso em: 06 jun. 2009.