

UNIVERSIDADE FEDERAL DE SANTA CATARINA

Implementação de multicamadas em Java:  
Refactoring para J2EE com Struts e Spring aplicado ao  
Sistema Informatizado para Gestão de Ouvidorias

DAVID GERSON RAMOS DE ANDRADE

ISAC RAMOS LUZ

FLORIANÓPOLIS – SC

20071/1

UNIVERSIDADE FEDERAL DE SANTA CATARINA

DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

CURSO DE SISTEMAS DE INFORMAÇÃO

Implementação de multicamadas em Java:  
Refactoring para J2EE com Struts e Spring aplicado ao  
Sistema Informatizado para Gestão de Ouvidorias

DAVID GERSON RAMOS DE ANDRADE

ISAC RAMOS LUZ

Trabalho de conclusão de curso apresentado  
como parte dos requisitos para obtenção do grau  
de Bacharel em Sistemas de Informação.

FLORIANÓPOLIS – SC

20071/1

DAVID GERSON RAMOS DE ANDRADE

ISAC RAMOS LUZ

Implementação de multicamadas em Java:  
Refactoring para J2EE com Struts e Spring aplicado ao  
Sistema Informatizado para Gestão de Ouvidorias

Trabalho de conclusão de curso apresentado como parte dos requisitos para obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Rony Reinehr Brand

Co-orientador: Prof. Vitório Bruno Mazzola, Dr.

Banca Examinadora

Prof. Vitório Bruno Mazzola, Dr.

Prof. José Eduardo De Lucca, Dr.

# SUMÁRIO

<b>LISTA DE FIGURAS .....</b>	<b>6</b>
<b>1 INTRODUÇÃO .....</b>	<b>10</b>
1.1 OUVIDORIA.....	11
1.2 TEMA .....	13
1.3 DEFINIÇÃO DO PROBLEMA .....	14
1.3.1 <i>Camada Lógica</i> .....	15
1.3.2 <i>Camada de Apresentação</i> .....	15
1.4 ESCOPO DE ESTUDO .....	15
1.5 OBJETIVO GERAL .....	16
1.6 OBJETIVOS ESPECÍFICOS .....	17
1.7 MOTIVAÇÃO .....	17
<b>2 SISTEMA INFORMATIZADO PARA GESTÃO DE OUVIDORIAS .....</b>	<b>19</b>
2.1 OUTRAS SOLUÇÕES EXISTENTES EM OUVIDORIA.....	23
<b>3 TECNOLOGIAS UTILIZADAS .....</b>	<b>26</b>
3.1 PROPOSTA J2EE SEM EJB .....	26
3.2 STRUTS.....	30
3.2.1 <i>Exemplo Prático do Struts</i> .....	32
3.3 SPRING .....	36
3.3.1 <i>Exemplo Prático do Spring</i> .....	37
3.4 JSTL .....	40
3.4.1 <i>Exemplo Prático de JSTL</i> .....	41
<b>4 DESENVOLVIMENTO .....</b>	<b>45</b>
4.1 LEVANTAMENTO DOS REQUISITOS DO SISTEMA.....	45
4.2 DIAGRAMA DE CLASSES DA CAMADA DE APRESENTAÇÃO .....	48
4.2.1 <i>Diagrama Resumido</i> .....	49
4.2.2 <i>Classes Detalhadas: Camada de Apresentação</i> .....	49
4.3 DIAGRAMA DE CLASSES DA CAMADA LÓGICA .....	54
4.3.1 <i>Diagrama de Classes Abstratas das Services</i> .....	55
4.3.2 <i>Diagrama de classes das Implementações das Services</i> .....	56
4.3.3 <i>Classes Detalhadas: Objetos Service da Camada Lógica</i> .....	56
4.3.4 <i>Diagrama de Classes de Domínio</i> .....	64
<b>5 COMPARAÇÃO DAS VERSÕES DO SOFTWARE .....</b>	<b>66</b>
5.1 PORTABILIDADE .....	66

5.2	SEGURANÇA.....	66
5.3	MANUTENÇÃO.....	67
<b>6</b>	<b>CONSIDERAÇÕES FINAIS .....</b>	<b>68</b>
6.1	CONCLUSÃO .....	68
6.2	TRABALHOS FUTUROS.....	69
	<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>72</b>
	<b>ANEXOS .....</b>	<b>74</b>
	ANEXO 1 – ARTIGO.....	74
<b>1.</b>	<b>INTRODUÇÃO .....</b>	<b>74</b>
<b>2.</b>	<b>SISTEMA INFORMATIZADO PARA GESTÃO DE OUVIDORIAS .....</b>	<b>76</b>
<b>3.</b>	<b>TECNOLOGIAS UTILIZADAS .....</b>	<b>78</b>
<b>4.</b>	<b>COMPARATIVO DE VERSÕES .....</b>	<b>82</b>
<b>5.</b>	<b>RESULTADOS .....</b>	<b>83</b>
	<b>REFERÊNCIAS .....</b>	<b>83</b>
	ANEXO 2 – TELAS DO SOFTWARE.....	85
	ANEXO 3 - CÓDIGO FONTE .....	87

## LISTA DE FIGURAS

Figura 1 – Arquitetura J2EE Tradicional	11
Figura 2 – Etapas para implantação de Ouvidorias. Fonte: [OMD]	19
Figura 3 – Arquitetura Geral do Sistema. Fonte: [OMD]	22
Figura 4 – Arquitetura de Contêiner Leve	30
Figura 5 – Arquitetura Completa do Struts. Fonte: [JSPINSIDER]	31
Figura 6 – Diagrama de Classes da Apresentação Resumido	49
Figura 7 – BaseAction	49
Figura 8 – AssuntoAction	50
Figura 9 – CadastroManifestacaoAction	50
Figura 10 – CargoAction	51
Figura 11 – CidadeAction	51
Figura 12 – ConsultaAction	51
Figura 13 – FeriadoAction	52
Figura 14 – LoginAction	52
Figura 15 – MenuAction	53
Figura 16 – NotificacaoAction	53
Figura 17 – PrazoAction	53
Figura 18 – UnidadeAction	54
Figura 19 – UsuárioAction	54
Figura 20 – Diagrama de Classes Abstratas das Services Resumido	55
Figura 21 – Diagrama de Classes da Implementação das Services Resumido	56
Figura 22 – ServiceImpl	57
Figura 23 – AssuntoServiceImpl	57
Figura 24 – CargoServiceImpl	58
Figura 25 – CaixaManifestacaoServiceImpl	58
Figura 26 – CidadeServiceImpl	59
Figura 27 – EstadoServiceImpl	59
Figura 28 – FeriadoServiceImpl	60
Figura 29 – ManifestacaoServiceImpl	60
Figura 30 – NotificacaoServiceImpl	61
Figura 31 – PerfilServiceImpl	61
Figura 32 – PrazoServiceImpl	62
Figura 33 – RecebimentoServiceImpl	62

Figura 34 – RelatorioServiceImpl	62
Figura 35 – UnidadeServiceImpl	63
Figura 36 – UsuarioServiceImpl	63
Figura 37 – Diagrama de Classes do Domínio Resumido	65
Figura 38 – Tela inicial do sistema	85
Figura 39 – Tela de consulta de manifestações	85
Figura 40 – Tela de inclusão de manifestações	86
Figura 41 – Tela de dados do manifestante	86
Figura 42 – Tela de manifestação cadastrada	87

## **RESUMO**

O processo de refactoring é uma atividade que realiza mudanças no código-fonte de uma aplicação, mas sem alterar o comportamento e propósito inicial do software.

A partir da reestruturação do Sistema Informatizado para Gerenciamento de Ouvidorias, baseada na modelagem J2EE e iniciada em 2006 por Rony Brand, a camada de persistência foi totalmente desenvolvida e testada. Porém, para que a reestruturação fosse completa, faltava o desenvolvimento das outras duas camadas.

Este trabalho tem por objetivo finalizar esta reestruturação e gerar a versão do sistema 2.0 completa, plenamente desenvolvido em multicamadas. E para que isto fosse possível foi utilizado o framework Apache STRUTS para agilizar a construção da camada de apresentação em conjunto com o framework SPRING que fará o elo entre camada lógica e as outras camadas.

Com esta adequação o software será altamente flexível, funcionando com as principais tecnologias do mercado, facilitando sua implantação nas organizações que o desejarem, aumentando a segurança e diminuindo os custos de implantação e manutenção.

Palavras-chave: ouvidoria, multicamadas, J2EE sem EJB, framework, Struts, Spring, JSTL

## **ABSTRACT**

The process of refactoring is an activity that carries through changes in the source-code of an application, but without modifying the behavior and initial intention of software.

From the reorganization of the "Sistema Informatizado para Gerenciamento de Ouvidorias", based on the modeling J2EE and initiate in 2006 by Rony Brand, the persistence layer was developed and tested tottaly. However, so that the reorganization was complete, it lacked to the development of the others two layers. This work has for objective to finish this reorganization and to generate the version of complete system 2.0, fully developed in multi-layers. E so that this was possible was used framework Apache STRUTS to speed the construction of the presentation layer in set with framework SPRING that will make the link between logical layer and the other layers.

With this adequacy, the system will be highly flexible, functioning with the main market technologies, facilitating its implantation in the organizations that desiring, increasing the security and diminishing the costs of implantation and maintenance.

Key-words: ombudsman, multi-layers, J2EE without EJB, framework, Struts, Spring, JSTL

# 1 Introdução

Através deste trabalho estaremos complementando o estudo iniciado em 2003 por Rony Brand, quando iniciou suas atividades como sócio e diretor de tecnologia da empresa OMD Soluções para Ouvidorias, onde desenvolveu a primeira versão do Sistema Informatizado para Gerenciamento de Ouvidorias.

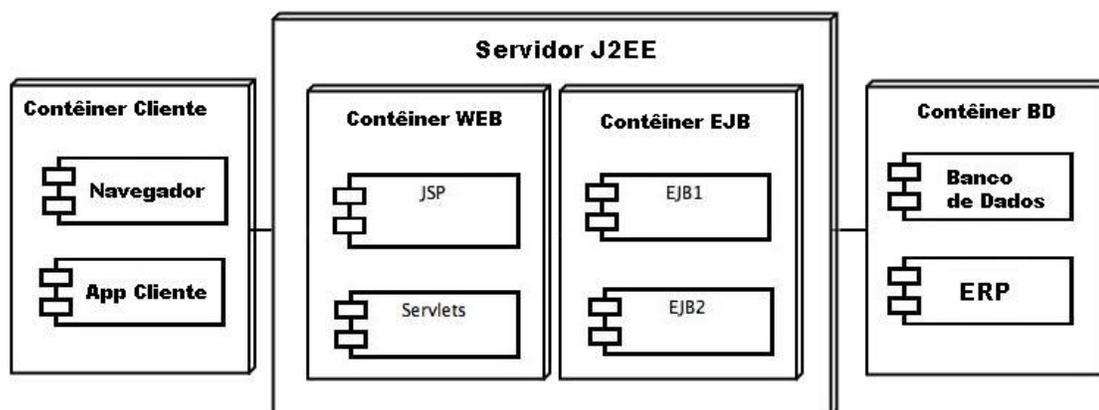
Esta primeira etapa tinha como finalidade inserir a empresa no mercado nacional e explorar um tema que começava a ganhar força na academia e que rapidamente seria disseminado na cultura das grandes organizações, o chamado CRM - *Customer Relationship Management* (Gerenciamento do Relacionamento com o Cliente).

Esta nova forma de vislumbrar os clientes, dando a eles a possibilidade de uma maior interação com as empresas, sugerindo melhorias e apontando os defeitos, fez emergir a idéia de se criar um sistema informatizado para auxiliar as organizações nesta nova cultura. Surgia assim o principal “serviço” oferecido pela OMD.

Como a grande maioria das primeiras versões de softwares, o Sistema de Ouvidorias foi concebido no ambiente acadêmico durante a graduação de seu criador utilizando as tecnologias conhecidas por ele e mais disseminadas no período (JSP/Servlets/JDBC).

Sabe-se que a evolução na área tecnológica avança numa aceleração exponencial e isto faz com que as pesquisas para melhorias nos sistemas sejam sempre constantes. Com este foco Brand resolveu inovar e em meados de 2005 iniciou um estudo, que tornaria seu trabalho de conclusão de curso, para remodelar

seu sistema de ouvidorias aplicando os conceitos de modelagem multicamadas J2EE.



**Figura 1 – Arquitetura J2EE Tradicional**

No início de 2006 ele finalizou o projeto, Persistência de Dados em Java: Um Estudo Aplicado ao Sistema Informatizado para Gerenciamento de Ouvidorias, onde conseguiu separar a persistência de dados do sistema criando uma das camadas do modelo. Neste ponto uma das premissas da nova modelagem estava resolvida, porém ainda faltavam as outras camadas para obter uma nova versão do software.

Foi neste ponto que se identificou a possibilidade de continuar o trabalho, para isso serão criadas duas novas camadas conforme modelagem sugerida, subdividindo a atual implementação para deixá-la de acordo com os padrões de desenvolvimento propostos.

## **1.1 Ouvidoria**

Conforme apresenta a OMD, as ouvidorias não devem se limitar a um instrumento de comunicação do povo com a administração pública, mas seu

conceito também deve ser aplicado em outros contextos como, entre o cliente e a empresa, o trabalhador e seu sindicato, o associado e sua entidade, o profissional e seu conselho, fornecendo um canal ágil e direto de comunicação.

Ela ainda acrescenta que:

Ter e oferecer essa importante ferramenta representa para a organização, acima de tudo, humanizar e estreitar o relacionamento com o cliente. Significa buscar soluções práticas e efetivas para os principais problemas da organização sob a ótica do cliente, identificar nas críticas as oportunidades de melhoria, inovar constantemente seus processos, produtos e serviços.

No decorrer do texto iremos aplicar o termo Cliente-Cidadão, utilizado por [BRAND 2006] em seu trabalho de conclusão para definir tanto o cliente de uma organização da iniciativa privada, bem como, o cidadão que usufrui de um serviço da administração pública.

E é neste contexto, onde o cliente-cidadão tem um papel importante para o bom funcionamento das organizações, apresentando reclamações, denúncias, elogios e/ou sugestões referentes aos serviços prestados por elas, que a empresa OMD está atuando. Provendo um serviço que vai desde o diagnóstico até a operacionalização efetiva de ouvidorias. Serviço este que vem ganhando destaque nas grandes organizações e que com este trabalho permitirá que seja aplicado às pequenas e médias empresas e também a órgãos públicos nestes moldes.

A seguir relacionamos os principais benefícios que a implantação de uma ouvidoria traz para as organizações, respeitando algumas premissas abordadas na

dissertação de [ALVES JR. 2002] e que foram consolidadas no site da empresa:

- Aproximação com os clientes-cidadãos, mediante o estabelecimento de um canal de comunicação acessível e direto;
- Identificação das áreas que estejam merecendo maior atenção dos dirigentes, definindo-se eixos prioritários de ação;
- Feedback sobre a atuação da organização, permitindo a correção de disfunções e redirecionamento das ações desenvolvidas;
- Identificação de necessidades dos usuários;
- Relacionamento democrático com a sociedade;
- Maior credibilidade e fortalecimento da imagem da organização junto à população;
- Inovação de processos produtos e serviços, a partir da participação do ouvidor no processo de discussão de políticas institucionais, propondo ações interventivas.

E partindo destas premissas é que o Sistema Informatizado para Gestão de Ouvidorias foi desenvolvido, estreitando assim o canal de comunicação entre os clientes-cidadãos e as organizações.

## **1.2 Tema**

O tema deste trabalho é o refactoring do Sistema Informatizado para Gerenciamento de Ouvidorias, que atualmente funciona com Servlets/JSP sem conceito de multicamadas, e que foi inicialmente objeto de estudo no trabalho

anterior de Rony Brand, atual orientador do projeto. A camada de persistência foi objeto de estudo do mesmo. Neste, trabalho, o resultado será o sistema plenamente desenvolvido em multicamadas, utilizando o framework Apache STRUTS para agilizar a construção da camada de apresentação em conjunto com o framework SPRING que fará o elo entre camada lógica e as outras camadas.

Após sua conclusão, teremos a segunda versão do sistema, já com as três camadas independentes e suas atribuições plenamente definidas, tornando-se um software altamente coeso, flexível, seguro e de fácil manutenção.

### **1.3 Definição do Problema**

Este Trabalho de Conclusão de Curso surgiu para dar continuidade ao trabalho feito por Rony Brand em 2006, onde foi desenvolvida a camada de persistência de dados para o Sistema Informatizado para Gestão de Ouvidorias. Na ocasião foi concluída a etapa de persistência, possibilitando ao sistema funcionar nos principais banco de dados relacionais existentes no mercado. Apesar de o sistema estar atualmente em funcionamento utilizando um servidor Servlet/JSP TomCat, e podendo ser acessado em diferentes navegadores, a idéia é adequá-lo ao padrão multicamadas e para isso dividi-lo em três camadas: uma delas contendo a lógica e as regras do negócio, a outra camada contemplando a apresentação web e interação com os usuários e a terceira camada de persistência que já encontra-se implementada e será integrada durante o desenvolvimento para a gerar a segunda versão do sistema.

Com esta adequação o software será altamente flexível funcionando com as principais tecnologias do mercado, facilitando sua implantação nas organizações

que o desejarem, aumentando a segurança e diminuindo os custos de implantação e manutenção.

### **1.3.1 Camada Lógica**

A camada lógica, também chamada de Regras de Negócio por alguns autores, cuida das necessidades da aplicação no domínio em que ela se insere. É a camada responsável por intermediar o processo de solicitação efetuado pela camada de Apresentação e buscar as informações relativas às respostas na camada de Persistência, devolvendo o resultado para a Apresentação. Qualquer tipo de cálculo ou validação será executado nesta camada.

### **1.3.2 Camada de Apresentação**

A camada de apresentação cuida da interação entre o usuário e o software. Neste trabalho, será construída uma camada de apresentação web, pois será um sistema baseado em navegadores de Internet (*browsers*) com a responsabilidade exibir a informação para o usuário e interpretar os comandos emitidos pelo usuário em ações para a camada lógica. Desta forma é possível construir uma camada bastante leve, que basicamente executa os tratamentos de telas e campos e geralmente acessa somente a segunda camada, a qual faz as requisições ao banco de dados e devolve o resultado.

## **1.4 Escopo de Estudo**

Utilizaremos para este trabalho algumas funcionalidades disponibilizada pelo

framework Spring na implementação das Regras do Negócio, o que facilitará a integração com as outras duas camadas. Já para a camada de Apresentação web utilizaremos o framework Struts, que agilizará a implementação e diminuirá o tempo de desenvolvimento da mesma, tomando devidos cuidados ao desenvolver as interfaces para possibilitar o funcionamento do sistema independente do Navegador e separando bem as duas camadas. Com estes dois frameworks em conjunto com o JDO já utilizado no trabalho anterior, esta aplicação se tornará altamente flexível, necessitando de poucas configurações para deixá-la em pleno funcionamento, utilizando apenas recursos de hardware e software mínimos exigidos na implantação.

Ao final, quando as duas camadas estiverem desenvolvidas, realizaremos uma análise comparativa entre a atual estrutura com as camadas implementadas, abordando aspectos de portabilidade, manutenção e segurança.

Os frameworks Spring e Struts foram escolhidos por possuírem diversos padrões internacionais de desenvolvimento e serem muito utilizados atualmente na implementação de aplicações comerciais. Além disso, suas interfaces facilitam a escrita e o reuso de código, diminuindo o tempo de desenvolvimento e principalmente o esforço de manutenção de código.

## **1.5 Objetivo Geral**

Estudar as tecnologias do Framework Spring para o desenvolvimento da camada Lógica e sua integração com as camadas de Apresentação e de Persistência e também do Framework Struts para nos auxiliar no desenvolvimento da camada de Apresentação WEB.

Aplicar o conhecimento adquirido no desenvolvimento da segunda versão do Sistema Informatizado para Gestão de Ouvidorias, e assim comprovar que a utilização destes frameworks auxilia nas práticas de programação, principalmente na proposta de utilização do modelo J2EE sem EJBs.

## **1.6 Objetivos Específicos**

- Descrever o Sistema Informatizado para Gestão de Ouvidorias, para esclarecer sua complexidade e justificar a necessidade da existência do desenvolvimento em multicamadas;
- Estudar os frameworks Spring e Struts para a obtenção dos melhores padrões de desenvolvimento destas ferramentas e aplicá-las neste sistema.
- Desenvolver a camada de apresentação totalmente independente.
- Desenvolver a camada lógica independente do sistema.
- Realizar uma análise comparativa entre camada atual (Servlets/JSP) versus as camadas: de Apresentação (com Struts).e Lógica (com Spring).

## **1.7 Motivação**

A partir do trabalho anterior de [BRAND 2006], que é um dos sócios da empresa OMD Soluções para Ouvidorias, foi identificada a necessidade de se alterar a modelagem de sua aplicação principal a fim de tornar todo o seu código compatível com a camada de persistência desenvolvida anteriormente.

Após a saída da incubadora GeNESS (Centro de Geração de Novos

Empreendimentos em Software e Serviços) em 2006, tendo adquirido um grau de maturidade e querendo se consolidar no mercado, a empresa observou a necessidade de remodelar o sistema para deixá-lo mais robusto e facilitando sua implantação, pois a nova versão do sistema adaptar-se-ia ao ambiente de hardware e software já existente em seus potenciais clientes.

Neste período a necessidade de um tema para a conclusão de curso fez com que os autores deste trabalho aceitassem este desafio de construir a segunda versão do software, que teria seu nome alterado de "Sistema Informatizado para Gerenciamento de Ouvidorias" para "Sistema Informatizado para Gestão de Ouvidorias", conforme as regras descritas na seção trabalhos futuros de [BRAND 2006], onde seriam construídas as camadas de Apresentação e Lógica para deixá-lo no padrão J2EE multicamadas, tornando a aplicação mais flexível, segura e de fácil manutenção.

## 2 Sistema Informatizado para Gestão de Ouvidorias

Conforme mencionado por [BRAND 2006], o Sistema Informatizado para Gestão de Ouvidorias é inovador na sua área, por se tratar de um sistema totalmente web, possibilitando ter um controle das manifestações apresentadas à organização, desde seu recebimento, passando por seu encaminhamento para as áreas responsáveis, setores ou departamentos dentro da organização, até a resposta ao cliente-cidadão, revisão desta resposta e encerramento da manifestação.

Para que o sistema funcione de acordo devem-se obedecer algumas etapas anteriores e posteriores a implantação conforme figura abaixo:



**Figura 2 – Etapas para implantação de Ouvidorias. Fonte: [OMD]**

Por este motivo, como citado por [ALVES JR 2004] no artigo Garantindo a efetividade da Ouvidoria, de nada adianta a organização possuir uma Ouvidoria para receber as manifestações de seus clientes-cidadãos se ela não estiver disposta a mudar seus processos e corrigir suas disfunções. E para que essas mudanças tenham efeito é necessário saber quais as unidades e assuntos que merecem um maior foco dos gestores, sem esquecer que o principal interessado continua sendo o cliente-cidadão. Novamente, percebe-se a necessidade de uma ferramenta que

forneça informações aos gestores para que assim possam identificar melhor os problemas e apoiá-los no processo de tomada de decisão, principalmente à medida que o número de manifestações torna-se maior. O sistema deve ainda permitir o refinamento das informações, compreendendo desde uma visão macro da organização, passando pelas suas unidades até a visão detalhada de uma manifestação específica.

A seguir é apresentado um resumo dos principais benefícios do Sistema Informatizado para Gestão de Ouvidorias:

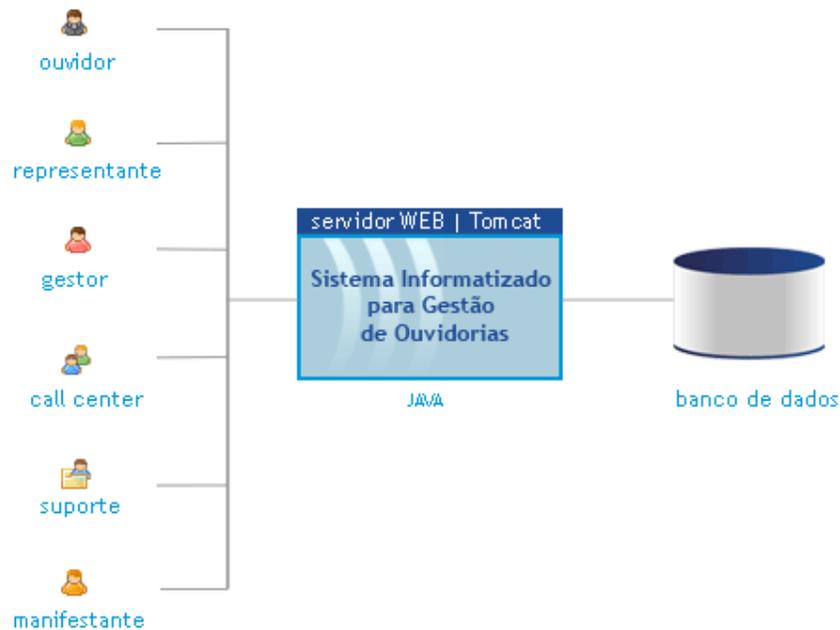
- Gerencia todo o processo e proporciona agilidade na resolução das manifestações;
- Registra e controla o recebimento de manifestações originadas por qualquer forma de atendimento: call center, atendimento pessoal, internet, carta, fax, caixa de sugestões, etc.;
- Controla prazos, identificando áreas pendentes;
- Encaminha respostas para o cidadão, disponibilizando-as para call center, Internet, carta, fax, etc.;
- Consolida informações estratégicas para a alta administração, indicando áreas críticas da organização;
- Integra ações entre as diversas áreas da organização, através de acesso descentralizado, via internet ou rede interna;
- Oferece confiabilidade e segurança no tratamento das informações;
- Possibilita navegação simples e fácil aprendizagem;

- Dispensa licença de uso por equipamento;
- Agiliza a implantação e facilita a manutenção;
- Adapta-se às necessidades e características da organização;

Além dos benefícios citados, o sistema possui outras funcionalidades que o torna diferenciado comparado com outros sistemas existentes no mercado:

- Acompanhamento da situação da manifestação pelo próprio manifestante (via Internet), com acesso por meio de código de segurança;
- Possibilidade de tratamento especial para manifestações sigilosas;
- Emissão de relatórios gerenciais dinâmicos, com a utilização do recurso “drill-down”, permitindo gerar informações estatísticas abrangendo toda a organização (visão macro) até uma manifestação específica (visão micro), facilitando a identificação das áreas que estejam merecendo maior atenção dos gestores;
- Possibilidade de efetuar diversas configurações no sistema mesmo quando ativo (configuração on-line).

Podemos ter uma melhor visão do sistema como um todo a partir da figura abaixo.



**Figura 3 – Arquitetura Geral do Sistema. Fonte: [OMD]**

A partir da figura podemos explicitar melhor o modelo de camadas, onde a direita vemos a camada de apresentação subdividida pelos tipos de usuários com acesso ao sistema. Ao centro a camada lógica, onde estão implementadas as regras do negócio e a esquerda a camada de persistência representada pelo banco de dados.

Após a citação de como o sistema funciona, apresenta-se à organização as características, requisitos e recursos mínimos de hardware e software para a instalação do servidor:

- Uma máquina para ser o Servidor com capacidade de processamento a partir de 1Ghz e com 512 MB ou mais de memória RAM;
- Pode ser instalado tanto em servidores Microsoft Windows ou distribuições Linux;
- Os aplicativos JVM 1.4 e TomCat 4.1.24, ou versões superiores dos

mesmos;

- Já para banco de dados temos uma diversidade maior que pode ser escolhido pelo cliente entre:
  - ✓ ORACLE Database versão 9i ou superior
  - ✓ Microsoft SQL SERVER versão 2000
  - ✓ MySQL versão 4.0.12 ou superior
  - ✓ PostgreSQL versão 7.3 ou superior
  - ✓ DB2 8.2 ou superior
  - ✓ Sybase Adaptive Server Enterprise (ASE) 15.0

Já para o acesso dos usuários cadastrados basta algumas ferramentas conforme abaixo:

- Computador com acesso à internet ou rede interna;
- Navegador Microsoft Internet Explorer, Mozilla Firefox, Netscape, Opera ou Konqueror;

## **2.1 Outras Soluções Existentes em Ouvidoria**

Algumas das empresas citadas por [BRAND 2006] continuam no mercado com a terceirização dos serviços prestados na área de ouvidorias. A empresa Ombudsman Associados, não existe mais, porém um dos sócios, Sylvio Lopes Riccó, continua na área de consultorias e atende no endereço [www.ouvidor.com.br](http://www.ouvidor.com.br). Além dele, outras empresas prestam consultoria na área da administração pública, podemos citar a Softsul – Sociedade Sul-Riograndense de Apoio ao

Desenvolvimento de Software que possui o PROJETO TIM, temos também a Cetil Sistemas de Informática, com sede em Blumenau-SC que conta com o Programa Nacional de Informatização e Modernização da Gestão Pública – PRONIM e também a Soft Micro, de Araçatuba-SP disponibiliza a solução Assessor Público - Tecnologia da Informação Aplicada a Administração Pública.

Outro concorrente que pode ser considerado indireto é a Del Grande Informática sediada em Garopaba-SC, tem seu foco voltado para a área de telefonia, possui o Sistema de Ouvidoria Intelicon, que atua junto ao PABX da empresa controlando e gravando as manifestações.

Como vimos, muitas empresas podem ser consideradas concorrentes indiretas da OMD, porém temos ainda aquelas empresas que disponibilizam um software pronto, que pode ser instalado na empresa e que são os concorrentes diretos. Uma dessas empresas é também de Santa Catarina e fica no sul do estado, a Betha Sistemas LTDA., com sede em Criciúma possui diversas soluções para Gestão Municipal, entre elas a Betha Ouvidoria, que possui algumas das características do Sistema Informatizado para Gestão de Ouvidorias como, controle de prazos, gráficos e relatórios.

Seguindo a linha de pacotes de software a empresa WebLink informática sediada em Brasília-DF, desenvolveu o Sistema para trâmites e gerenciamento de Ouvidorias utilizando o ambiente Microsoft, com tecnologia ASP e Banco de Dados SQL Server. Por possuir esta parceria, não utiliza software livre, limitando sua gama de clientes.

Já empresa SOF Informática, voltada para a área de Gerenciamento do Relacionamento com o Cliente possui o Software Exceller, que é disposto em

módulos para atender Call Center e áreas de Marketing, lançou o novo módulo Exceller Ouvidoria para abranger este mercado em crescimento.

A empresa Infoexpert, com sede em Recife-PE, citada no trabalho anterior, continua no mercado, porém alterou o nome de seu produto de PROVIDORIA para Ouvidor4i que segundo eles é um Sistema de informações Inteligentes Integradas na Internet Aplicado à Ouvidoria, porém não há no site maiores informações sobre o novo produto.

Além destes, temos os departamentos de informática de vários órgãos públicos que desenvolvem seus próprios sistemas de ouvidorias. É o caso das prefeituras de São Paulo e Rio de Janeiro, do governo da Bahia e de Brasília, de algumas universidades, e até do governo federal com o SERPRO - Serviço Federal de Processamento de Dados, que implantou seu Sistema Ouvidoria há quatro anos em fase experimental e está lançando oficialmente ao governo. Possui quatro módulos: Acionamento, Acompanhamento, Gerencial e Aprendizado e é uma solução desenvolvida sob plataforma web, banco em Software Livre e dotado de condições de acessibilidade.

Comparando a essas empresas, a OMD destaca-se pelo conjunto da obra, onde a participação durante o processo de implantação é fundamental, prestando consultoria prévia nas dependências da organização para o estabelecimento dos fluxos, responsabilidades e prazos para encaminhamento das manifestações. Juntamente a estes fatores, o sistema informatizado permite a emissão de relatórios com informações úteis para transformar a ouvidoria numa poderosa ferramenta gerencial, para identificar áreas da organização que estejam merecendo maior atenção dos gestores e assim servir de suporte ao processo de tomada de decisões.

## 3 Tecnologias Utilizadas

### 3.1 Proposta J2EE sem EJB

Com base nos estudos de [JOHNSON&HOELLER 2004] as arquiteturas J2EE tradicionais geralmente eram motivadas por stress e trabalho pesado. Isso tudo porque seu desenvolvimento e manutenção envolviam um nível de complexidade que excediam o das próprias regras de negócio e requisitos que o sistema deveria atender, o que incluía dependência da API padronizada pela Sun, conhecida como Enterprise Java Beans (EJB).

Trabalhar com EJB significa estar envolvido em uma série de conceitos que fazem parte do mesmo, como:

- Problemas na portabilidade entre servidores de aplicação J2EE, pois geralmente a migração entre servidores de aplicação significa envolver compilação ou modificação de código, além de limitar a possibilidade de escolha do mesmo. Não seria possível utilizar um contêiner web como o TomCat por exemplo.
- Perda de produtividade no desenvolvimento dos aplicativos J2EE, porque no desenvolvimento e manutenção estão envolvidos uma série de arquivos de configuração, além de envolver também a criação de outros artefatos, que não são diretamente ligados a lógica do negócio, como por exemplo: a interface home e a interface local de um bean. Multiplicando isso, pelo número total de classes, que fazem parte da lógica de negócio do sistema, a complexidade no desenvolvimento do projeto é incrementada

consideravelmente. Este fator leva o usuário a buscar alternativas de ferramentas como XDoclet para automatizar esse processo, porém é um tipo de dependência desnecessário.

Quando o principal, e muitas vezes o único, interesse em utilizar um container J2EE é somente o gerenciamento de transações, no que ele é perfeitamente usual, este projeto encaixa-se perfeitamente, não necessitando de serviços como clusterização ou endereçamento remoto (RMI) embutido no EJB.

Com o surgimento do framework Spring, que será apresentado no item 3.3, utilizar EJB apresenta grandes vantagens apenas quando é realmente necessário realizar chamadas remotas de objetos, ou seja, somente quando se utiliza SessionBeans Stateless e fazendo as chamadas com sua interface Remote (quando existe a necessidade de se ter dois servidores de aplicação), entretanto, se utilizar a interface Local, esta pode ser substituída com grandes vantagens pelo Spring pois neste caso usa-se POJOs (*Plain Old Java Object*), onde pode-se controlar transações (programáticas e declarativas), integração com as camadas de persistência e web, obtendo-se um desempenho melhor e consumindo menos recursos .

Simplicidade e foco na regra de negócio e rapidez no processo de desenvolvimento são as motivações principais na escolha deste tipo de arquitetura. Alternativas de componentes da camada de persistência, como Hibernate ou JPOX, por exemplo, já são motivos suficientes para questionar as Entities Beans como componentes de acesso aos dados.

Ainda reforçando o conceito de Orientação a Objetos, as chamadas entre classes ocorrem, endereçando as suas respectivas interfaces e não suas

implementações. Além disso, é necessária a existência de aplicações falsas, como Data Transfer Objects, que nada mais são do que classes com funções de get e set, não envolvendo lógica alguma de negócio. Neste trabalho é utilizado o padrão de projeto Domain Model, que cria uma teia de relações entre os objetos do problema, basicamente seguindo o comportamento exigido pelas regras de negócio do sistema desenvolvido.

Baseado nas informações anteriores foi adotada a metodologia de Arquitetura de Contêiner Leve (Lightweight Container Architecture). Os contêineres leves ajudam a montar componentes de projetos diferentes em uma aplicação coesa. Por trás destes contêineres há um padrão comum de como eles fazem a amarração, o conceito é conhecido como Inversão de Controle (Inversion of Control ou IoC). Seu tipo mais popular e amplamente utilizado é a Dependency Injection (Injeção de Dependências).

A idéia básica da Injeção de Dependências é ter um objeto separado, chamado montador, que popula um campo em um objeto com uma implementação apropriada para a sua interface. Existem três principais formas de Injeção de Dependências: a forma Constructor Injection (Injeção por Construtores); a Setter Injection (Injeção por Métodos Set) e a Interface Injection (Injeção por Interfaces).

Ao decidirmos pela utilização do Framework Struts e Spring (abordados mais a fundo num próximo tópico) utilizaremos a Injeção por Métodos Set, onde a Action do Struts utiliza-se de setters pra tudo que ela precisar e o Controlador do Spring chama esse setter passando o parâmetro esperado.

Neste tipo de arquitetura, a camada de negócio é composta de POJOs, que são simples objetos Java e utilizando a Injeção de Dependência, propõe-se a lidar

com as seguintes funções:

- Gerenciar o ciclo de vida de objetos de negócio;
- Controlar o gerenciamento de dependência entre objetos da camada de negócio;
- Utilizar o conceito de Orientação a Aspectos, provendo serviços como controle de transação declarativa dos objetos em execução;
- Gerenciar as transações declarativas;
- Integrar com outros frameworks e tecnologias como o Struts e o JDO.

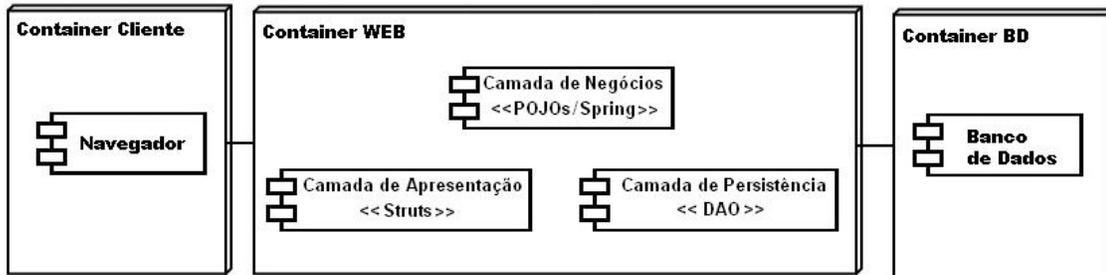
O uso desta Injeção de Dependência pode ajudar a facilitar a visualização de quais são as dependências do componente. Você pode simplesmente analisar o mecanismo de injeção, como os métodos set, para encontrá-las.

Outra razão pela preferência da Injeção de Dependência é que ela facilita a realização de testes, onde se consegue resultados rápidos e eficazes apenas substituindo implementações de serviços reais por stubs ou objetos falsos (mock).

Com os contêiners leves todas as classes JAVA são executadas pela mesma máquina virtual, ao contrário de uma arquitetura J2EE típica, onde as classes presentes na camada de apresentação e da camada de negócio, executam em máquinas virtuais diferentes, uma para cada camada.

Um lightweight container não está estritamente vinculado à arquitetura J2EE, sendo assim possível de ser executado tanto em um Contêiner Web, como Tomcat, quanto através de aplicativos stand-alone ou mesmo através de um container EJB.

A seguir uma figura que exemplifica bem o que estamos sugerindo, com as camadas bem definidas:



**Figura 4 – Arquitetura de Contêiner Leve**

Uma das razões que reforçam o conceito de Contêiner Leve é que não é necessária a implementação de padrões como Singleton ou Factory para cada objeto sugerido na camada de negócio.

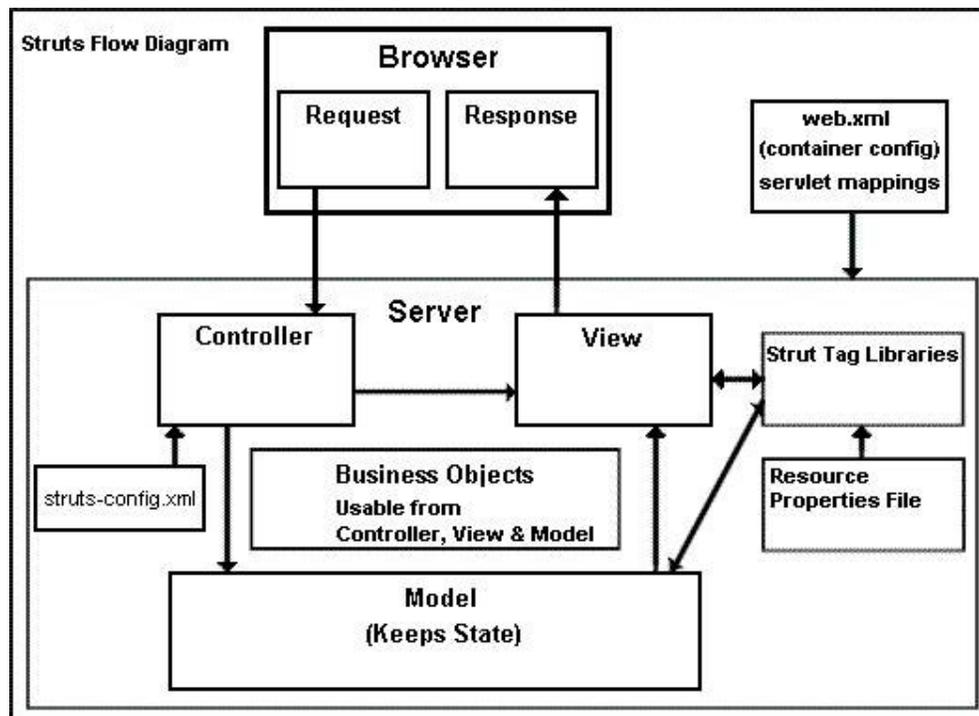
Outra característica importante é não impor requisitos especiais para a execução dos objetos da aplicação, facilitando a suas inicializações sem a necessidade de passos adicionais para fazer o deploy da aplicação.

### 3.2 Struts

O Struts é um framework que implementa o modelo MVC (Model-View-Controller), e que surgiu para melhorar o desenvolvimento de aplicações em Smalltalk e virou um padrão utilizado em diversas linguagens. Com ele podemos facilmente configurá-lo para construir novas aplicações, além de fornecer uma melhor separação entre a camada de apresentação e a camada lógica. Sua primeira versão surgiu em 2001, como parte do projeto Jakarta da Apache Software Foundation e devido ao seu sucesso tornou-se um projeto do Apache.

O nome Struts vem das estruturas que sustentam edificações, assim struts dá sustentação às aplicações. Ele possui a classe Action Servlet que representa o

Controle da aplicação, em geral não é necessário estendê-la, pois ele delega tarefa de acordo com o arquivo struts-config.xml, fazendo com que as modificações tenham impacto diminuído sobre a aplicação. Uma melhor visualização do framework Struts pode ser observada na figura abaixo:



**Figura 5 – Arquitetura Completa do Struts. Fonte: [JSPINSIDER]**

Quando as requisições chegam, elas passam pelo Controle e então ele aponta para quem deve tratá-la, se a requisição for composta por formulário(s) ela é repassada as classes que estendem ActionForm, caso contrário é encaminhado diretamente a classe que estende Action. A classe ActionForm fica responsável por guardar os dados do formulário e validá-los. No caso dos dados estarem incorretos, a requisição volta para a página que a iniciou. Se os dados estiverem corretos esta requisição será passada a uma classe de Action que é quem faz algum processamento sobre a requisição, consultando o modelo e colocando objetos para

serem exibidos na camada de apresentação.

Para este trabalho, utilizou-se de uma subclasse especializada de ActionForm chamada DynaActionForm, que permite a criação de FormBeans dinâmicos utilizando-se de configurações no arquivo de propriedades *struts-config.xml*, sem requerer do desenvolvedor a criação de uma classe Java para cada tipo de FormBean. DynaActionForm elimina a necessidade da classe de FormBean fazendo com que o tempo de desenvolvimento seja reduzido.

### 3.2.1 Exemplo Prático do Struts

#### Trecho do arquivo struts-config.xml

```
<?xml version="1.0" encoding="ISO-8859-1" ?>

<!DOCTYPE struts-config PUBLIC
"-//Apache Software Foundation//DTD Struts Configuration 1.2//EN"
"http://jakarta.apache.org/struts/dtds/struts-config_1_2.dtd">

<struts-config>
  <form-beans>
    <form-bean type="org.apache.struts.action.DynaActionForm"
name="LoginFormBean">
      <form-property type="java.lang.String" name="login"/>
      <form-property type="java.lang.String" name="senha"/>
    </form-bean>
  <action-mappings>
    <action path="/login"
type="br.com.umd.ouvidoria.web.controle.action.LoginAction"
name="LoginFormBean">
      <forward name="success" path="/WEB-INF/jsp/inicial.jsp"></forward>
      <forward name="errologin" path="/WEB-INF/jsp/index.jsp"></forward>
    </action>
  </action-mappings>
  <controller processorClass="org.apache.struts.tiles.TilesRequestProcessor"/>

```

```
</struts-config>
```

Acima, o trecho de código inicial do arquivo *struts-config.xml*, onde são inseridos os códigos referentes ao form bean LoginFormBean utilizado para efetuar a validação do usuário que irá logar-se no sistema e em seguida a configuração das actions para cada caso de retorno obtido.

A seguir, temos a implementação da classe LoginAction, onde está a implementação do código utilizado pelo sistema ao executar a Action.

### **Classe LoginAction**

```
package br.com.ond.ouvidoria.web.controle.action;

import br.com.ond.ouvidoria.logica.domain.Usuario;
import br.com.ond.ouvidoria.logica.dto.UsuarioDTO;
import br.com.ond.ouvidoria.logica.service.UsuarioService;
import br.com.ond.ouvidoria.util.Criptografia;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.DynaActionForm;

public class LoginAction extends BaseAction {

    public LoginAction() {
    }

    public ActionForward defaultMethod(ActionMapping mapping, ActionForm form,
```

```

HttpServletRequest request, HttpServletResponse response) throws Exception {
    DynaActionForm dynaform=(DynaActionForm)form;
    String login = (String)dynaform.get("login");
    String senha = Criptografia.retornaString(Criptografia.digest(
((String)dynaform.get("senha")).getBytes()));

    UsuarioDTO usuarioDTO = new UsuarioDTO();

    usuarioDTO.setLogin(login);
    usuarioDTO.setSenha(senha);

    UsuarioService usuarioService =
(UsuarioService)getWebApplicationContext().getBean("usuarioService");
    String destino;
    try {
        Usuario usuario = usuarioService.autenticar(usuarioDTO);
        HttpSession session = request.getSession(true);
        session.setAttribute("usuario", usuario);
        destino = "success";
    } catch (Exception e ){
        e.printStackTrace();
        destino = "errologin";
        request.setAttribute("erro", new Boolean(true));
    }
    return mapping.findForward(destino);
}
}
}

```

Esta classe é utilizada quando seu serviço é requisitado pelo arquivo JSP index.jsp abaixo:

#### **Trecho do index.jsp**

```
<%@ taglib prefix="c" uri="/WEB-INF/c.tld" %>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```

<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
    <title>OMD v2.0 - OMD Soluções para Ouvidorias</title>
  </head>

  <body>
    <div id="master">
      
      <div id="form">
        <form name="dados" method="post" action="login.do">
          <c:if test="{erro != null}">
            <span class="erro">Usuário e/ou senha incorretos</span><br><br>
          </c:if>
          <label>login</label><input name="login" type="text" />
          <label>senha</label><input name="senha" type="password" />
          <input type="submit" class="botao" value="entrar" style="border:none" />
        </form>
        <p><a href="#">esqueci a senha</a></p>
      </div>
      <h1>Sistema Informatizado para Gest&atilde;o de Ouvidorias </h1>
    </div>
  </body>
</html>

```

Além das actions exemplificadas anteriormente, o Struts possui um conjunto de tags bastante extenso facilitando assim a construção da camada de apresentação, suas tags são html, bean, logic, nested e tiles. Entretanto utilizando a JSP Standard Tag Library (JSTL) é possível obter grande parte das funcionalidades das tags do Struts, pois existem tags que permitem obter e modificar valores, criar condicionais,

repetição, bem como, obter suporte a internacionalização.

### **3.3 Spring**

Criado por Rod Johnson para amenizar os problemas encontrados ao utilizar a tecnologia de Enterprise Java Beans (EJB), parte principal do J2EE. A pedra fundamental do Spring Framework é o conceito de Dependency Injection (Injeção de Dependências - DI) que é uma forma de Inversion Of Control (Inversão de Controle – IoC) na qual um componente não instancia suas dependências mas o ambiente (contêiner) automaticamente as fornece.

Como o Spring é responsável pelo controle do ciclo de vida dos beans existentes na camada de negócio, e pela busca de cada recurso, minimizando a dependência do código desenvolvido do container em questão.

Ao utilizar o Spring Framework temos como grande vantagem o fato de não interferir diretamente na modelagem da aplicação. Ele afeta a arquitetura, conjunto de decisões sobre a divisão de um sistema em módulos e a forma como estes módulos interagem, do sistema, mas os objetos de negócio não sofrem muitos impactos. Além disso, o Spring induz ao uso de boas práticas de programação ao se baseia ostensivamente em interfaces ao invés de classes. Esta característica, além de permitir extensão e adaptação fácil do framework e ser um conceito da Orientação a Objetos, possibilita o uso de testes unitários com muita facilidade.

Utiliza-se o Spring para intermediar esta arquitetura de controlador web com a interface de acesso ao banco de dados principalmente quando eles são desenvolvidos por equipes diferentes, onde não é necessário o conhecimento da outra camada para continuidade do projeto.

O Spring é um framework abrangente voltado para o desenvolvimento Java corporativo. Ele inclui a possibilidade de criação de transações concentra e abstratas (reduzindo assim o tamanho de arquivos de configuração xml), bem como, se integra muito bem com frameworks de persistência, e no desenvolvimento de aplicações web e JDBC.

A principal razão de sua escolha é o controle de transações. O uso de transações declarativas possibilita que o desenvolvedor de componentes construa o componente sem precisar demarcar transações, evita a introdução de erros que surgem decorrentes do esquecimento de sessões com o banco de dados abertas e não confirmadas, já que o contêiner se encarrega disto. Isso incentiva o reuso, pois o componente é customizado com uma política de segurança mais ou menos rígida, dependendo do escopo onde é usado, sem necessidade de mudanças no código fonte.

Além disso, o código utilizado na camada de negócio, em um momento acessando a camada de persistência através de conexões JDBC, poderá ser utilizado para acessar a base utilizando outra tecnologia de dados como JPOX ou Hibernate, executando a adaptação sem alterar o código fonte, mas sim o próprio framework. Assim o Spring saberá a quantidade de recursos de conexão disponíveis para ele, de acordo com a conexão escolhida, ou seja, a escalabilidade é gerenciada pelo container.

### 3.3.1 Exemplo Prático do Spring

<b>Trecho de applicationContext-service.xml</b>
---

<?xml version="1.0" encoding="UTF-8"?>
--

```

<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN"
'http://www.springframework.org/dtd/spring-beans.dtd'>
<beans>
  <bean id="abstractTxDefinition" abstract="true"
class="org.springframework.transaction.interceptor.TransactionProxyFactoryBean">
    <property name="transactionManager" ref="transactionManager"/>
    <property name="transactionAttributes">
      <props>
        <prop key="get*">PROPAGATION_REQUIRED,readOnly</prop>
        <prop key="pesquisar*">PROPAGATION_REQUIRED,readOnly</prop>
        <prop key="save*">PROPAGATION_REQUIRED</prop>
        <prop key="delete*">PROPAGATION_REQUIRED</prop>
        <prop key="*">PROPAGATION_REQUIRED,readOnly</prop>
      </props>
    </property>
  </bean>
  <!-- Manifestacao -->
  <bean id="manifestacaoService" parent="abstractTxDefinition">
    <property name="target">
      <bean
class="br.com.ond.ouvidoria.logica.service.impl.ManifestacaoServiceImpl">
        <property name="dao" ref="manifestacaoDAO"/>
      </bean>
    </property>
  </bean>
  <!-- Manifestacao -->
</beans>

```

### **Classe ManifestacaoServiceImpl**

```

package br.com.ond.ouvidoria.logica.service.impl;

import br.com.ond.ouvidoria.logica.domain.Estado;
import br.com.ond.ouvidoria.logica.domain.Manifestacao;

```

```

import br.com.omd.ouvidoria.logica.dto.ConsultaManifestacaoDTO;
import br.com.omd.ouvidoria.logica.service.ManifestacaoService;
import br.com.omd.ouvidoria.persistencia.geral.dao.ManifestacaoDAO;
import java.util.Date;
import java.util.HashMap;
import java.util.List;

public class ManifestacaoServiceImpl extends ServiceImpl implements
ManifestacaoService{

    public ManifestacaoServiceImpl() {
    }

    public List getDadosOcorrencia() throws Exception {
        return ((ManifestacaoDAO)dao).getDadosOcorrencia();
    }

    public HashMap getDadosCadastro() throws Exception {
        return ((ManifestacaoDAO)dao).getDadosCadastro();
    }

    public HashMap getDadosCadastroManifestacao() throws Exception {
        return ((ManifestacaoDAO)dao).getDadosCadastroManifestacao();
    }

    public Manifestacao salvarExternaAnonima(Manifestacao manifestacao) throws
Exception {
        return
((ManifestacaoDAO)dao).salvarExternaAnonima(this.incluiDatas(manifestacao));
    }

    public Manifestacao salvarExternaManifestante(Manifestacao manifestacao)
throws Exception {
        return
((ManifestacaoDAO)dao).salvarExternaManifestante(this.incluiDatas(manifestacao));
    }

    public List getEstados() throws Exception {
        return ((ManifestacaoDAO)dao).getEstados();
    }

    public List getCidadesPorEstado(Estado estado) throws Exception{
        return ((ManifestacaoDAO)dao).getCidadesPorEstado(estado);
    }
}

```

```

    }
    public Manifestacao salvarInternaAnonima(Manifestacao manifestacao) throws
Exception {
        return
((ManifestacaoDAO)dao).salvarInternaAnonima(this.incluiDatas(manifestacao));
    }
    public Manifestacao salvarInternaManifestante(Manifestacao manifestacao)
throws Exception {
        return
((ManifestacaoDAO)dao).salvarInternaManifestante(this.incluiDatas(manifestacao));
    }
    private Manifestacao incluiDatas(Manifestacao manifestacao){
        Date dataAtual = new Date();
        manifestacao.setDataHoraRecebimento(dataAtual);
        manifestacao.setDataUltimaModificacao(dataAtual);
        return manifestacao;
    }
    public HashMap getDadosConsulta() throws Exception {
        return ((ManifestacaoDAO)dao).getDadosConsulta();
    }
    public List consultar(ConsultaManifestacaoDTO dto) throws Exception {
        return ((ManifestacaoDAO)dao).consultar(dto);
    }
    public Date getMenorDataRecebimento() throws Exception {
        return ((ManifestacaoDAO)dao).getMenorDataRecebimento();
    }
}
}

```

### 3.4 JSTL

JSP Standard Tag Library (Biblioteca Padrão de Tags JSP) consiste em uma coleção de bibliotecas, tendo cada uma um propósito bem definido, que permitem escrever páginas JSPs sem código Java, aumentando assim a legibilidade do código e a interação entre desenvolvedores e web designers.

Uma página JSTL é uma página JSP contendo um conjunto de tags JSTLs onde cada tag realiza um determinado tipo de processamento e faz parte de uma biblioteca JSTL, sendo algo muito melhor do que a utilização de código Java dentro de JSP como ocorria no passado, na arquitetura de uma camada somente (Modelo 1). Pode-se utilizar várias bibliotecas JSTLs para apenas uma página JSTL.

Surgiu para criar páginas dinâmicas e complexas sem a necessidade de escrever código Java dentro delas, Fornecendo tags que tornam fáceis as tarefas que exigiriam várias linhas de código Java, como formatação de números e datas seguindo configurações regionais de cada usuário, iteração de listas, operações condicionais, tags para manipular documentos XML, internacionalização e tags SQL (O uso desta tag não é aconselhável pois utiliza diretamente o bancos tornando o sistema com apenas uma camada). Também introduz uma nova linguagem de expressões que simplifica o desenvolvimento de páginas JSP.

### 3.4.1 Exemplo Prático de JSTL

#### Trecho de manifestacao.jsp

```
<%@ taglib prefix="c" uri="/WEB-INF/c.tld" %>
<%@ taglib prefix="fmt" uri="/WEB-INF/fmt.tld" %>
<c:if test="{usuario == null || usuario.unidade == null}">
  <c:redirect url="../sair.do"/>
</c:if>
<html>
<head>
  <title> OMD v2.0 - OMD Soluções para Ouvidorias</title>
</head>
<body>
  <form name="dados" action="manifestacao.do" method="post">
```

```

<input type="hidden" name="method" value="criaManifestacao">
<tr>
  <td colspan="4" class="tit1"> Dados da Ocorrência do Fato</td>
</tr>
<jsp:useBean id="now" class="java.util.Date" />
<input type="hidden" name="dataAtual" value="<fmt:formatDate value='${now}'
pattern='dd/MM/yyyy'/>">
<input type="hidden" name="horaAtual" value="<fmt:formatDate value='${now}'
pattern='HH:mm'/>">
<tr>
  <td class="label">Unidade*:</td>
  <td colspan="3"><select name="unidade" onchange='mostrar(dados.unidade,
dados.rotuloOutraUnidade, dados.outraUnidade)'>
    <c:forEach var="unidade" items="${unidades}">
      <option value="<c:out value='${unidade.nome}'/>"> <c:out
value="${unidade.nome}'/></option>
    </c:forEach> </select></td>
</tr>
</form>
</body>
</html>

```

O exemplo apresenta um teste de validação de senha, sem código java, porém utilizando estrutura condicional.

Na primeira linha, temos o seguinte:

```
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core" %> e
```

Essa declaração informa ao compilador de JSPs que ele deve utilizar uma biblioteca de tags, cuja tags é reconhecida pelo prefixo "c", de modo a evitar possíveis conflitos com tags de mesmo nome de outras bibliotecas.

A biblioteca de tags é identificada pelo atributo uri.

As bibliotecas padrão disponibilizadas pela Sun para a utilização do JSTL são:

Biblioteca JSTL	Prefixo	URI	Tipos de Uso	exemplo
Core	c	http://java.sun.com/jstl/core	Acessar e modificar dados em memória Comandos condicionais Loop	<c:if> <c:forEach>
Processamento de XML	x	http://java.sun.com/jstl/xml	Leitura de documentos Impressão de partes de documentos XML Tomada de decisão baseado no conteúdo de um documento XML	<x:read>
Internacionalização e formatação	fmt	http://java.sun.com/jstl/fmt	Leitura e impressão de números Leitura e impressão de datas Ajuda a sua aplicação funcionar em mais de uma língua	<fmt:formatDate>

As bibliotecas do JSTL fornecem tags para executar trechos repetidamente, de maneira a simular os comandos da linguagem Java.

Na maioria das vezes, os scriptlets de JSP são usados para executar tarefas condicionais (if..else) ou interativas (for, while). Os tags JSTL da biblioteca core fornecem estas funcionalidades sem ter de recorrer aos scriptlets. O <c: if > e <c: forEach > Tag fornecem as operações condicionais e interativas básicas. Por exemplo, um simples if seria lido:

```
<c:if test="{usuario == null || usuario.unidade == null}">
  <c:redirect url="../sair.do"/>
</c:if>
```

Estruturas de loop podem ser executadas conforme:

```
<c:forEach var="unidade" items="{unidades}">
```

```
<option value="<c:out value='${unidade.nome}'/>"> <c:out  
value='${unidade.nome}'/></option>
```

```
</c:forEach>
```

Finalmente, uma rápida menção da tag format. Estas são separadas das Tags core e têm que ser importados conforme:

```
<%@ taglib prefix="fmt" uri="http://java.sun.com/jstl/fmt" %>
```

## 4 Desenvolvimento

Neste item é contemplada agora uma visão mais detalhada das camadas desenvolvidas. Primeiro, explicando as classes envolvidas na camada de apresentação, gerenciadas pelo Struts. Em seguida, a parte mais extensa deste trabalho, que são as classes envolvidas na camada de negócio, gerenciadas pelo Spring.

### 4.1 Levantamento dos Requisitos do Sistema

Como este trabalho dá continuidade ao trabalho de Brand, os requisitos do projeto permanecem os mesmos. Sendo assim, citando [BRAND 2006], temos os seguintes requisitos:

- Inserir, atualizar, excluir e consultar **usuários** do sistema: nome, apelido, *login* (único), senha, unidade, cargo, perfil de acesso, telefone e email;
- Inserir, atualizar, excluir e consultar a entidade de **assunto** de manifestação que contém um nome que não pode ser repetido;
- Inserir, atualizar, excluir e consultar a entidade **cargo** de um funcionário que contém um nome que não pode ser repetido;
- Inserir, atualizar, excluir e consultar a entidade **cidade**, que contém um nome e um estado. Não podendo existir duas ou mais cidades com mesmo nome no mesmo estado;

- Inserir, atualizar, excluir e consultar os **prazos** do sistema contendo: o nível, situação e a quantidade de dias úteis do prazo. Não podendo existir dois ou mais prazos com o mesmo nível e situação;

- Inserir, atualizar, excluir e consultar datas de **feriados** e dias não úteis, estas datas devem ser únicas. Os feriados serão utilizados para calcular o prazo da manifestação, dessa forma deve haver a possibilidade de consultar entre um período tanto retornando todas as datas de feriado, bem como contando a quantidade de feriados daquele período, para a consulta pode ou não ser informada uma data inicial e/ou uma data final;

- Inserir, atualizar, excluir e consultar a entidade **modo**. O modo terá um nome, que não se repetir e identifica como o manifestante chegou a Ouvidoria para se manifestar, bem como a forma desejada para envio da resposta. Há modos pode ser retornados e outros que não podem, deve haver uma consulta também por este campo;

- Consultar classificações das **manifestações**, possíveis forma de identificação do **Manifestante** e também perfis de acesso ao sistema;

- Inserir e consultar **manifestações** que contém data, hora, unidade, envolvidos e descrição da ocorrência do fato, caso o usuário não encontre a unidade é registrada outra unidade, neste cadastro também deve ser armazenado a forma de identificação do Manifestante, o modo de resposta desejado por ele, a classificação da manifestação e um assunto, da mesma forma caso o usuário não encontre o assunto desejado é registrado outro assunto. Ao finalizar o cadastro é gravado um código de segurança para a manifestação e data prevista de conclusão da manifestação;

- O cadastro do **Manifestante** contém o nome, endereço, cidade, estado, telefones e email, caso não encontre a cidade é registrada outra cidade;

- Registrar o recebimento da manifestação: data e hora do recebimento, modo de entrada, e ainda o usuário e seu cargo;

- Registrar o **encaminhamento** das **manifestações** registrando a data, data prevista para resposta, unidade de origem e destino e ainda o usuário e seu cargo;

- Registrar **respostas** às **manifestações**, contendo: data, origem, reposta, quantidade de dias úteis utilizados na resposta, unidade de origem e destino e ainda o usuário e seu cargo;

- Registrar **comentários** realizados pela Ouvidoria: data, comentário, data prevista para resposta, unidade de origem e destino e ainda o usuário e seu cargo;

- Registrar a conclusão de uma **manifestação**: data e hora da conclusão, unidade de origem e destino usuário, cargo e unidade responsável pela resposta, bem como a quantidade de dias úteis utilizados na manifestação;

- Fornecer uma consulta para informar a quantidade de cada uma das **outras unidades**, bem como fornecer um mecanismo para substituir estas outras unidades por unidades cadastradas, ao final do processo são excluídas as outras unidades;

- Fornecer uma consulta para informar a quantidade de cada um dos

**outros assuntos**, bem como fornecer um mecanismo para substituir estes outros assuntos por assuntos cadastrados, ao final do processo são excluídos os outros assuntos;

- Fornecer uma consulta para informar a quantidade de cada uma das **outras cidades**, bem como fornecer um mecanismo para substituir estas outras cidades por cidades cadastradas, ao final do processo são excluídas as outras cidades. [BRAND,34-35]

Como este trabalho é uma continuidade de um processo de refactoring iniciado com o trabalho anterior, alterar as regras de negócio e funcionalidades características do Sistema Automatizado para Gestão de Ouvidorias não foi cogitado. Entretanto, o tópico seguinte tem por objetivo aprofundar mais as outras duas camadas escolhidas e suas atribuições e características de um projeto J2EE sem EJB.

## **4.2 Diagrama de Classes da Camada de Apresentação**



Classe base que possui os principais métodos utilizados por todas as classes da camada de apresentação. Basicamente, são os métodos responsáveis pela carga dos dados e exibi-los na página que foi requisitada pelo usuário.

AssuntoAction
+ addAssunto(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward
+ AssuntoAction()
+ changeAssuntos(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward
+ defaultMethod(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward
+ deleteAssunto(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward
+ editAssunto(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward
+ getAssunto(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward
+ getAssuntos(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward

**Figura 8 – AssuntoAction**

Classe responsável por todos os eventos envolvidos na exibição das páginas de consulta, exclusão, edição, inserção de assuntos das manifestações e pela troca de um assunto que tenha sido inserido de forma especifica por outro assunto padrão do sistema.

CadastroManifestacaoAction
+ CadastroManifestacaoAction()
+ criaManifestacao(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward
+ criaManifestante(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward
+ defaultMethod(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward
+ getCidades(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward
+ getDadosCadastroManifestacao(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward

**Figura 9 – CadastroManifestacaoAction**

Classe responsável pela página de cadastramento da manifestação.

CargoAction
<pre> + addCargo(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward + CargoAction() + defaultMethod(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward + deleteCargo(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward + editCargo(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward + getCargo(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward + getCargos(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward </pre>

**Figura 10 – CargoAction**

Classe responsável por todos os eventos envolvidos na exibição das páginas de consulta, exclusão, edição e inserção de cargos dos usuários do sistema.

CidadeAction
<pre> + CidadeAction() + consultaCidade(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward + criaCidade(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward + defaultMethod(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward + editaCidade(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward + excluiCidade(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward + getCidades(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward + montaDadosTela(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward + outraCidade(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward </pre>

**Figura 11 – CidadeAction**

Classe responsável por todos os eventos envolvidos na exibição das páginas de consulta, exclusão, edição, inserção de cidades em que se localizam as unidades da organização, e pela troca de uma cidade que tenha sido inserido de forma específica por outra cidade padrão do sistema.

ConsultaAction
<pre> + ConsultaAction() + defaultMethod(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward + getCidades(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward + montaDadosTela(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward + resultadoConsulta(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward </pre>

**Figura 12 – ConsultaAction**

Classe responsável pela exibição da página de consulta de manifestações, exibição da página com o resultado da consulta e pela submissão dos dados inseridos pelo usuário no formulário de consultas.

FeriadoAction
+ addFeriado(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward + defaultMethod(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward + deleteFeriado(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward + editFeriado(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward + FeriadoAction() + getFeriadoPorPeriodo(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward + getFeriados(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward

**Figura 13 – FeriadoAction**

Classe responsável pela inserção, edição, consulta e exclusão dos feriados e dias não úteis que a organização considera, influenciando no calculo dos prazos de resposta das manifestações.

LoginAction
+ defaultMethod(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward + LoginAction()

**Figura 14 – LoginAction**

Classe simples responsável pela exibição e submissão da página inicial do sistema, que exige do o usuário membro da organização ou da ouvidoria os dados de acesso.

MenuAction
+ defaultMethod(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward + MenuAction()

### Figura 15 – MenuAction

Classe responsável por montar e exibir o menu com estrutura em árvore. Sendo que os dados e funções presentes na referida árvore dependem do nível do usuário que acessou o sistema naquele momento. Esta página não é exibida para o manifestante.

NotificacaoAction
+ atualizaHabilitado(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward
+ defaultMethod(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward
+ NotificacaoAction()

### Figura 16 – NotificacaoAction

Classe responsável pela exibição e captura dos eventos da página de notificação, que consiste num simples jsp com um checkbox, onde a alteração do status do mesmo(marcado ou não) é repassado para esta que repassa para o sistema que o responsável pela Ouvidoria na unidade, deve ser notificado por email,sobre o encaminhamento da manifestação.

PrazoAction
+ defaultMethod(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward
+ editPrazo(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward
+ getPrazos(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward
+ PrazoAction()

### Figura 17 – PrazoAction

Classe responsável pelos eventos da página de inserção, edição, consulta e exclusão dos prazos para resposta da manifestação de acordo com a hierarquia dos níveis, em que são classificadas as unidades da organização.

UnidadeAction
+ addUnidade(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward
+ changeUnidades(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward
+ defaultMethod(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward
+ deleteUnidade(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward
+ editUnidade(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward
+ exibeUnidade(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward
+ exibeUnidadeEdicao(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward
+ getCidadesOnAddUsuario(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward
+ getCidadesOnEditUsuario(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward
+ getUnidade(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward
+ getUnidades(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward
+ UnidadeAction()

**Figura 18 – UnidadeAction**

Classe responsável por todos os eventos envolvidos na exibição das páginas de consulta, exclusão, edição, inserção de unidades da organização e pela troca de um assunto que tenha sido inserido de forma específica por outra unidade padrão do sistema.

UsuarioAction
+ addUsuario(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward
+ defaultMethod(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward
+ deleteUsuario(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward
+ editUsuario(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward
+ exibeUsuario(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward
+ exibeUsuarioEdicao(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward
+ getUsuario(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward
+ getUsuarios(mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :HttpServletResponse) : ActionForward
+ UsuarioAction()

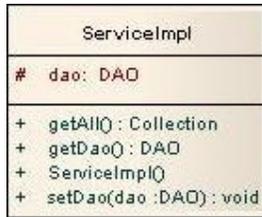
**Figura 19 – UsuárioAction**

Classe responsável por todos os eventos envolvidos na exibição das páginas de consulta, exclusão, edição, inserção de usuários do sistema.

### 4.3 Diagrama de Classes da Camada Lógica

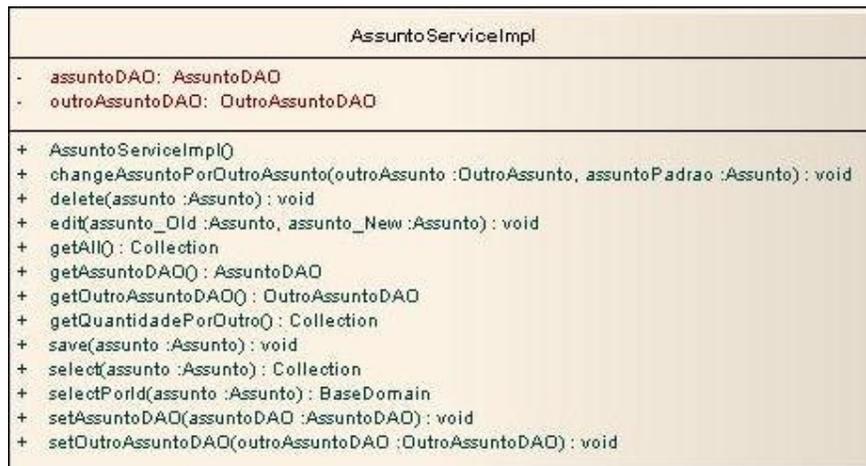






**Figura 22 – ServiceImpl**

Classe base, de com generalização do método de pesquisa geral, bem como a injeção do DAO genérico responsável por esta pesquisa. São subclasses do mesmo: NotificacaoServiceImpl, CidadeServiceImpl e ManifestacaoServiceImpl.



**Figura 23 – AssuntoServiceImpl**

Implementação da Interface AssuntoService, que fornece os métodos necessários para pesquisar todos os Assuntos, pesquisar um determinado Assunto por Id, excluir um determinado assunto, efetuar a troca de um assunto por um assunto padrão do sistema, alterar um determinado assunto, inserir um novo assunto e é injetado no mesmo( seguindo a injeção de dependência) as classes AssuntoDAO e OutroAssuntoDAO. Este último, responsável pela troca de um assunto inserido por um assunto padrão do sistema.

CargoServiceImpl	
-	cargoDAO: CargoDAO
+	CargoServiceImpl()
+	delete(cargoDTO :CargoDTO) : void
+	edit(cargoDTO_Old :CargoDTO, cargoDTO_New :CargoDTO) : void
+	getAll() : Collection
+	getCargoDAO() : CargoDAO
+	save(cargoDTO :CargoDTO) : void
+	select(cargoDTO :CargoDTO) : Collection
+	selectPorId(cargoDTO :CargoDTO) : BaseDomain
+	setCargoDAO(cargoDAO :CargoDAO) : void

**Figura 24 – CargoServiceImpl**

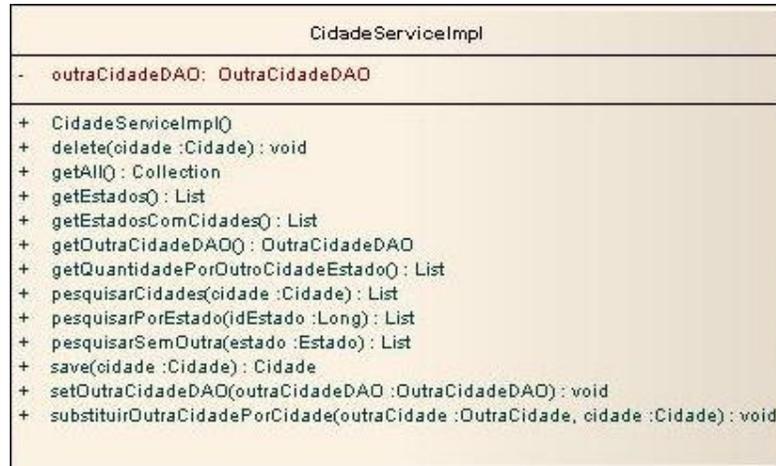
Implementação da Interface CargoService, que fornece os métodos necessários para pesquisar todos os Cargos, pesquisar um determinado Cargo por Id, excluir um determinado Cargo, inserir um novo Cargo e é injetado no mesmo (segundo a injeção de dependência) a classe CargoDAO.

CaixaManifestacaoServiceImpl	
-	dao: DAO
+	atualizarNova(manifestacao :Manifestacao) : void
+	cadastrarReposta(manifestacao :Manifestacao, resposta :Resposta) : void
+	CaixaManifestacaoServiceImpl()
+	contaManifestacoesPorSituacao(usuario :Usuario) : List
+	encaminhar(manifestacao :Manifestacao, arquivoNotificacaoEncaminhamento :File, email :Email) : void
+	enviarResposta(usuario :Usuario, codigoManifestacao :Long, numeroRespostaSelecionada :Integer) : Manifestacao
+	getAguardandoResposta(unidadeParametro :Unidade, unidadeUsuario :Unidade) : List
+	getARevisar(unidadeParametro :Unidade, unidadeUsuario :Unidade) : List
+	getDadosNova(idUnidade :Long) : HashMap
+	getDadosResponder(idUnidade :Long) : HashMap
+	getDao() : DAO
+	getDetalhes(codigoManifestacao :Long) : Manifestacao
+	getEncerradas(unidadeParametro :Unidade, usuario :Usuario) : List
+	getNova(codigoManifestacao :Long) : Manifestacao
+	getNovas(unidade :Unidade) : List
+	getRespondidas(usuario :Usuario) : List
+	getUnidade(idUnidade :Long) : Unidade
+	getUnidadesNivelAbaixoSemOutraSemNenhuma(idUnidade :Long) : List
+	setDao(dao :DAO) : void
+	setNova(idManifestacao :Long, nova :boolean) : void

**Figura 25 – CaixaManifestacaoServiceImpl**

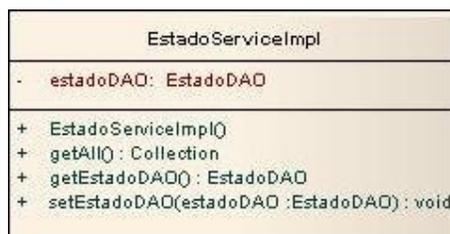
Classe responsável pelo processamento dos dados necessários para montar a

## Caixa de Manifestações do Menu.



**Figura 26 – CidadeServiceImpl**

Implementação da Interface `CidadeService`, que fornece os métodos necessários para pesquisar todos os Cidades, pesquisar um determinado Cidade por Id, excluir um determinado assunto, pesquisar Cidades por Estado, efetuar a troca de uma Cidade por um Cidade padrão do sistema, alterar um determinado assunto, inserir um novo assunto e é injetado no mesmo( seguindo a injeção de dependência) as classes `CidadeDAO` e `OutraCidadeDAO`. Este último, responsável pela troca de uma cidade inserida por uma Cidade padrão do sistema.



**Figura 27 – EstadoServiceImpl**

Implementação da Interface `EstadoService`, que fornece os métodos

necessários para pesquisar todos os Estados e é injetado no mesmo (segundo a injeção de dependência) a classe EstadoDAO.



**Figura 28 – FeriadoServiceImpl**

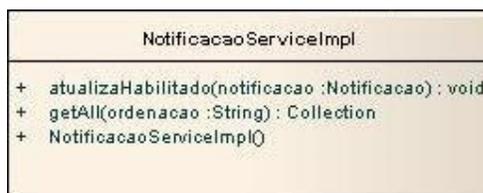
Classe responsável pela implementação da lógica envolvida na inserção, alteração, edição, exclusão e consulta de Feriados (e dias não úteis em geral) da Organização, no Sistema.



**Figura 29 – ManifestacaoServiceImpl**

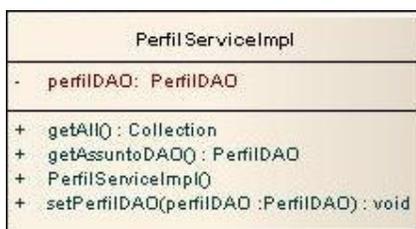
Classe responsável pela implementação da lógica envolvida na inserção e consulta de Manifestações no Sistema. Além de outras funções como verificação da

menor data de recebimento da Manifestação, dados de cadastro da Manifestação, dados da ocorrência da Manifestação em si e outras funções em geral.



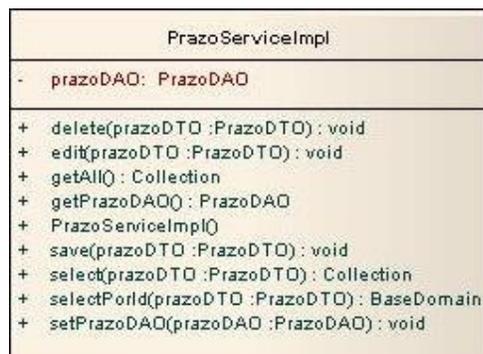
**Figura 30 – NotificacaoServiceImpl**

Classe que atualiza a instância do tipo Notificação, presente no módulo de Configurações, que informa se o responsável da Unidade deve ou não receber a notificação do Sistema de novas manifestações inseridas.



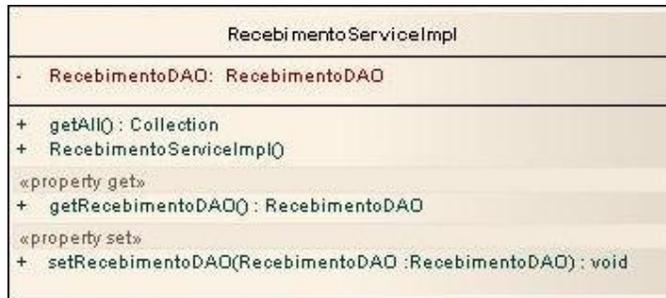
**Figura 31 – PerfilServiceImpl**

Classe responsável pela definição do Perfil de acesso do Usuário no Sistema.



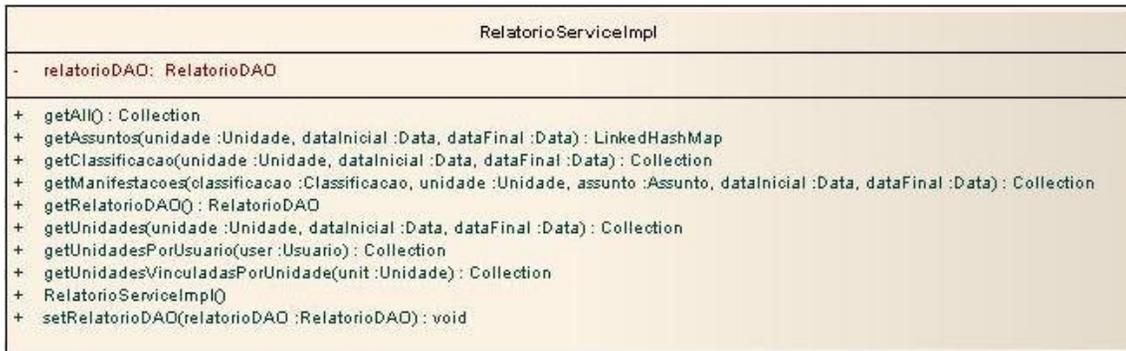
**Figura 32 – PrazoServiceImpl**

Classe responsável pela lógica referente às funções de inserção, exclusão, edição e pesquisa dos Prazos de resposta de Manifestações existentes em uma Unidade.



**Figura 33 – RecebimentoServiceImpl**

Classe responsável pela pesquisa de manifestações que foram recebidas no sistema.



**Figura 34 – RelatorioServiceImpl**

Classe responsável por todos os eventos referentes à montagem e exibição de Relatórios (Por Unidade, Por Classificação, Por Assunto).

UnidadeServiceImpl	
-	outraUnidadeDAO: OutraUnidadeDAO
-	unidadeDAO: UnidadeDAO
+	changeUnidadePorOutraUnidade(outraUnidade :OutraUnidade, unidadePadrao :Unidade) : void
+	delete(unidade :Unidade) : void
+	edit(unidade_Old :Unidade, unidade_New :Unidade) : void
+	getAll() : Collection
+	getAllSemOutraSemNenhuma() : Collection
+	getOutraUnidadeDAO() : OutraUnidadeDAO
+	getQuantidadePorOutro() : Collection
+	getUnidadeDAO() : UnidadeDAO
+	save(unidade :Unidade) : void
+	select(unidade :Unidade) : Collection
+	selectOutraUnidadePorId(outra_unidade :OutraUnidade) : BaseDomain
+	selectPorId(unidade :Unidade) : BaseDomain
+	setOutraUnidadeDAO(outraUnidadeDAO :OutraUnidadeDAO) : void
+	setUnidadeDAO(unidadeDAO :UnidadeDAO) : void
+	UnidadeServiceImpl()

**Figura 35 – UnidadeServiceImpl**

Implementação da Interface UnidadeService, que fornece os métodos necessários para pesquisar todas as Unidades, pesquisar uma determinada Unidade por Id, excluir um determinado assunto, efetuar a troca de uma unidade por uma unidade padrão do sistema, alterar uma determinada unidade, inserir uma nova unidade e é injetada no mesmo( seguindo a injeção de dependência) as classes UnidadeDAO e OutraUnidadeDAO. Este último, responsável pela troca de uma unidade inserida por uma unidade padrão do sistema.

UsuarioServiceImpl	
-	usuarioDAO: UsuarioDAO
+	autenticar(usuarioDTO :UsuarioDTO) : Usuario
+	delete(usuario :Usuario) : void
+	edit(usuario_Old :Usuario, usuario_New :Usuario) : void
+	getAll() : Collection
+	getUsuarioDAO() : UsuarioDAO
+	save(usuario :Usuario) : void
+	select(usuario :Usuario) : Collection
+	selectPorId(usuario :Usuario) : BaseDomain
+	setUsuarioDAO(usuarioDAO :UsuarioDAO) : void
+	UsuarioServiceImpl()

**Figura 36 – UsuarioServiceImpl**

Classe responsável pela lógica das funções de inserção, exclusão, edição,



### **Figura 37 – Diagrama de Classes do Domínio Resumido**

O diagrama de classes completo exibindo todas as relações entre as classes domain do sistema é bastante complexo, porém não foi alterado. Todas as classes exibidas acima estão descritas no trabalho anterior referenciado como [BRAND 2006].

## **5 Comparação das Versões do Software**

Ao final do Desenvolvimento das Camadas de Apresentação e Lógica, pode-se realizar uma análise das versões 1.0 e 2.0 do software. Para isto deve-se levar em conta os seguintes critérios: portabilidade, segurança e manutenção.

### **5.1 Portabilidade**

A maior portabilidade do software ficou visível, uma vez que a versão anterior exigia determinadas características para ser implantada, com a implementação atual, com pequenas modificações no código fonte, pode-se instalá-lo em diferentes plataformas.

### **5.2 Segurança**

Com o uso de transações declarativas possibilitou-se ao desenvolvedor que se construa os componentes sem precisar demarcar transações, incentivando o reuso já que o escopo da transação é determinado sem que seja necessário recompilar o código fonte do componente. Isto também evita a introdução de erros que podem surgir decorrentes do esquecimento de sessões com o banco de dados abertas e não confirmadas, já que o contêiner se encarrega disto.

Na versão 1.0 tínhamos os dados acessados diretamente da apresentação para o banco de dados, o que era um risco que o sistema corria.

### **5.3 Manutenção**

Utilizando camadas bem definidas e transparentes a versão 2.0 facilita as futuras manutenções do software, pois sabe-se exatamente onde devem ser feitas as correções ou melhorias após a reestruturação das classes do domínio do problema e suas respectivas funções no sistema.

## **6 Considerações Finais**

### **6.1 Conclusão**

No trabalho anterior, [BRAND 2006] adéqua a primeira versão do sistema, que inicialmente utilizava o contêiner Servlet TomCat que acessava diretamente o banco de dados MySQL com JDBC, para deixá-lo nos moldes dos softwares atuais utilizando o paradigma de orientação à objetos e o desenvolvimento em multicamadas. Para isto foi feito um estudo de caso e remodelado o sistema de modo a integrar o aspecto de multicamadas. Na ocasião, o foco foi criar a camada de persistência para tornar o sistema independente do banco de dados.

A partir deste trabalho, foi realizado um novo estudo para tornar a nova versão do software totalmente multicamada, o que garantiria maior segurança das informações, maior flexibilidade na implantação e menor esforço na manutenção.

A estruturação e implementação das camadas Web e Lógica para o sistema juntamente com a camada de Persistência desenvolvida anteriormente fará com que os custos com a manutenção do software sejam reduzidos e proporcionam à organização oferecer seus serviços a variadas infra-estruturas disponíveis em seus clientes alterando o mínimo de código possível de acordo com a infra-estrutura disponibilizada pelo cliente.

Após a comparação efetuada no capítulo anterior entre a atual camada que utiliza Servlet/JSP TomCat, com as novas camadas de Apresentação e Lógica construídas com o auxílio dos frameworks Strus e Spring pode-se afirmar que o desenvolvimento em multicamadas é bastante benéfico, principalmente com a

modelagem de Contêineres Leves, pois utiliza-se da estrutura J2EE que é bem difundida, porém sem a necessidade de esforços dispendiosos com o EJB para aplicações que não utilizarão as chamadas remotas, que foi o caso proposto.

O sucesso desta modelagem juntamente com as ferramentas propostas, lembrando que a escolha dos frameworks em questão foi puramente por se adequarem ao design da arquitetura com lightweight container, que por sua vez oferece a simplicidade e ainda assim os recursos suficientes pra implementar um sistema do tipo que é a Gestão de Ouvidorias, foi posto em prova na primeira implantação deste sistema em uma empresa real, onde necessitou-se que o código fonte fosse adaptado para que o acesso ao banco de dados utilizado na organização, que possui licença para uso, diferentemente do banco de dados com código livre utilizado nos testes de desenvolvimento.

Levando-se em consideração o tempo de programação utilizado para tal adaptação, pode-se concluir que a remodelagem do sistema proporcionará a empresa uma diminuição nos custos de manutenção e implantação do software e que os objetivos deste projeto foram alcançados com sucesso.

## **6.2 Trabalhos Futuros**

Após a transformação do sistema para o modelo multicamadas, podemos sugerir para trabalhos futuros um estudo para melhorar a camada de Apresentação, apesar de ela estar adequada ao sistema, como utilizamos o Struts, uma nova alternativa seria utilizar o SpringMVC, pois como as outras duas camadas estão utilizando o framework Spring, a integração final seria facilitada.

Como mencionado por [BRAND 2006], a camada de Apresentação também

poderia ser implementada com JSF (Java Server Faces) e AJAX, tornando a aplicação mais iterativa, já que possui grande suporte a eventos, assemelhando-se a iteratividade de aplicações desktop.

Outro ponto a ser melhorado seria a interface web, que não obteve muitas modificações da primeira para a segunda versão. O desafio seria tornar seu layout mais atrativo para os usuários que irão utilizá-lo no seu dia-a-dia.

Em função do tempo não foi possível utilizar-se de todas as ferramentas disponibilizadas pelo framework Struts como a Tag Tiles, o Validator e a Internacionalização, que podem ser agregados em outros trabalhos.

A Tag Tiles merece maior destaque, pois facilita a construção de Layouts, ou seja, com ele novas páginas se encaixarão no molde especificado, retirando a necessidade de copiar aspectos comuns em diversas páginas. Assim, aumentamos o reuso e evitamos redundâncias de código HTML, já que todas as JSPs fazem parte de um layout e em tempo de compilação as páginas são montadas preenchendo os encaixes.

O Validator facilita a validação de dados tanto no cliente como no servidor, já que para uma boa validação é necessário utilizá-la tanto do lado do cliente, pois evita o tráfego de informações incorretas pela rede, economizando tempo e largura de banda, bem como, utilizar validação do lado do servidor, evita que páginas modificadas enviem dados incorretos e acabem gerando resultados indesejados.

O suporte a internacionalização da aplicação é outro diferencial e facilitador do framework. Com ele é possível agrupar os rótulos e mensagens de erros num arquivo *properties* que pode ser facilmente substituído ao modificar o local do cliente, assim para mudar a localização basta trocar este arquivo de propriedades.



## REFERÊNCIAS BIBLIOGRÁFICAS

- [BRAND 2006] BRAND, Rony Reinehr. Persistência de Dados em Java: Um Estudo Aplicado ao Sistema Informatizado para Gerenciamento de Ouvidorias. Dissertação de Graduação. UFSC, Florianópolis, 2006. Disponível em <[http://www.omd.com.br/download/tcc\\_rony.pdf](http://www.omd.com.br/download/tcc_rony.pdf)>. Acesso em 05 de Setembro de 2006.
- [ALVES JR 2002] ALVES JR., Mário Nelson. A contribuição estratégica das ouvidorias para a melhoria dos serviços prestados pelas organizações: Um estudo de caso na Secretaria de Estado da Saúde de santa catarina. Dissertação de Mestrado. UDESC, Florianópolis, 2002. Disponível em <<http://www.omd.com.br/download/disserta.pdf>>. Acesso em 13 de Fevereiro de 2007.
- [OMD] OMD Soluções para Ouvidorias. Disponível em: <<http://www.omd.com.br>>. Acesso em 27 de Maio de 2007.
- [JOHNSON&HOELLER 2004] JOHNSON, Rod; HOELLER, Juergen. Expert one-on-one J2EE development without EJB. EUA, Wrox, 2004.
- [TODD& SZOLKOWSKI 2003] TODD, Nick ; SZOLKOWSKI, Mark: Java Server Pages: O guia do desenvolvedor. Campus, 2003
- [HUSTED 2004] HUSTED, Ted; DUMOULIN, Cedric; FRANCISCUS, George; WINTERFELDT, David : Struts em ação. Moderna, 2004.
- [STRUTS] Apache Struts Framework. Disponível em: <<http://struts.apache.org/1.2.9/index.html>>. Acesso em 13 de Setembro de 2006.
- [CORREA 2004] CORRÊA, Alice Alves. Avaliação do Framework Struts para Implementação de Aplicações Web usando Padrão Modelo-Visão-Controlador.

Dissertação de Graduação. UFSC, Florianópolis, 2004. Disponível em <[http://projetos.inf.ufsc.br/arquivos\\_projetos/projeto\\_6/tcc\\_struts\\_final\\_completo.pdf](http://projetos.inf.ufsc.br/arquivos_projetos/projeto_6/tcc_struts_final_completo.pdf)>. Acesso em 15 de Setembro de 2006.

[JSPINSIDER] JSP Insider. Disponível em: <<http://www.jspinsider.com/content/jsp/struts/strutsintro.jsp>>. Acesso em 03 de Março de 2007.

[SPRING] Spring Framework. Disponível em: <<http://www.springframework.org>>. Acesso em 13 de Setembro de 2006.

[WALLS&BREIDENBACH 2005] WALLS, Craig; BREIDENBACH, Ryan. Spring in Action. EUA, Manning, 2005.

[RAIBLE 2004] RAIBLE, Matt. Spring Live. EUA, SourceBeat, 2004

[CALÇADO 2006] CALÇADO, Phillip. Curso Spring Framework. Disponível em: <<http://fragmental.com.br/files/cursos/spring-112006/apostila-spring-112006.pdf>>. Acesso em 01 de Dezembro de 2006.

[BAYERN 2002] BAYERN, Shawn. JSTL in Action. EUA, Manning Publications Co., 2002.

[BERGSTEN 2003] BERGSTEN, Hans. JavaServer Pages 3rd Edition. EUA, O'Reilly, 2003.

## ANEXOS

### Anexo 1 – Artigo

# Implementação de multicamadas em Java: Refactoring para J2EE com Struts e Spring aplicado ao Sistema Informatizado para Gestão de Ouvidorias

David Gerson Ramos de Andrade, Isac Ramos Luz

Departamento de Informática e Estatísticas (INE) – Centro Tecnológico (CTC) Universidade Federal de Santa Catarina (UFSC) – Caixa Postal 476 – CEP 88040-900 Campus Universitário – Florianópolis, SC – Brasil

{dgerson, sufista}@inf.ufsc.br

**Resumo.** Este artigo refere-se ao trabalho de refactoring para J2EE do Sistema Informatizado para Gerenciamento de Ouvidorias, que atualmente funciona com o contêiner Servlet TomCat e JSP que acessa diretamente o banco de dados MySQL através JDBC sem o conceito de multicamadas. A camada de persistência foi objeto de estudo de Rony Brand anteriormente e já está implementada. Neste trabalho, o resultado será o sistema plenamente desenvolvido em multicamadas, utilizando o framework Apache STRUTS para agilizar a construção da camada de apresentação em conjunto com o framework SPRING que fará o elo entre camada lógica e as outras camadas. Após sua conclusão, teremos a segunda versão do sistema, já com as três camadas independentes e suas atribuições plenamente definidas, tornando-se um software altamente coeso, flexível, seguro e de fácil manutenção.

**Abstract.** This paper mentions the work to it of refactoriong for J2EE of “Sistema Informatizado para Gestão de Ouvidorias”, that currently it functions with container Servlet TomCat and JSP that it has access the data base MySQL through JDBC directly without the concept of multi-layers. The persistence layer was object of study of Rony Brand previously and already it is implemented. In this work, the result will be the system fully developed in multi-layers, using framework Apache STRUTS to speed the construction of the presentation layer in set with framework SPRING that will make the link between service layer and the other layers. After its conclusion, we will have the second version of the system, already with the three independent layers and its attributions fully defined, becoming a highly concise software, flexible, safe and of easy maintenance.

## 1. Introdução

Através deste trabalho estaremos complementando o estudo iniciado em 2003 por Rony Brand, quando iniciou suas atividades como sócio e diretor de tecnologia da empresa OMD Soluções para Ouvidorias, onde desenvolveu a primeira versão do Sistema Informatizado para Gerenciamento de Ouvidorias.

A primeira etapa tinha como finalidade inserir a empresa no mercado nacional e explorar um tema que começava a ganhar força na academia e que rapidamente seria

disseminado na cultura das grandes organizações, o chamado CRM - *Customer Relationship Management* (Gerenciamento do Relacionamento com o Cliente).

Esta nova forma de vislumbrar os clientes, dando a eles a possibilidade de uma maior interação com as empresas, sugerindo melhorias e apontando os defeitos, fez emergir a idéia de se criar um sistema informatizado para auxiliar as organizações nesta nova cultura. Surgia assim o principal “serviço” oferecido pela OMD.

Como a grande maioria das primeiras versões de softwares, o Sistema de Ouvidorias foi concebido no ambiente acadêmico durante a graduação de seu criador utilizando as tecnologias conhecidas por ele e mais disseminadas no período (JSP/Servlets/JDBC).

Sabe-se que a evolução na área tecnológica avança numa aceleração exponencial e isto faz com que as pesquisas para melhorias nos sistemas sejam sempre constantes. Com este foco Brand resolveu inovar e em meados de 2005 iniciou um estudo, que tornaria seu trabalho de conclusão de curso, para remodelar seu sistema de ouvidorias aplicando os conceitos de modelagem multicamadas J2EE.

No início de 2006 ele finalizou o projeto, Persistência de Dados em Java: Um Estudo Aplicado ao Sistema Informatizado para Gerenciamento de Ouvidorias, onde conseguiu separar a persistência de dados do sistema criando uma das camadas do modelo. Neste ponto uma das premissas da nova modelagem estava resolvida, porém ainda faltavam as outras camadas para obter uma nova versão do software.

Foi a partir deste momento que se identificou a possibilidade de dar continuidade ao trabalho, para isso serão criadas duas novas camadas conforme modelagem sugerida, subdividindo a atual implementação para deixá-la de acordo com os padrões de desenvolvimento propostos.

## **1.1. Ouvidoria**

Conforme apresenta a OMD, as ouvidorias não devem se limitar a um instrumento de comunicação do povo com a administração pública, mas seu conceito também deve ser aplicado em outros contextos como, entre o cliente e a empresa, o trabalhador e seu sindicato, o associado e sua entidade, o profissional e seu conselho, fornecendo um canal ágil e direto de comunicação.

No decorrer do texto iremos aplicar o termo Cliente-Cidadão, utilizado por [BRAND 2006] em seu trabalho de conclusão para definir tanto o cliente de uma organização da iniciativa privada, bem como, o cidadão que usufrui de um serviço da administração pública.

E é neste contexto, onde o cliente-cidadão tem um papel importante para o bom funcionamento das organizações, apresentando reclamações, denúncias, elogios e/ou sugestões referentes aos serviços prestados por elas, que a empresa OMD está atuando. Provendo um serviço que vai desde o diagnóstico até a operacionalização efetiva de ouvidorias. Serviço este que vem ganhando destaque nas grandes organizações e que com este trabalho permitirá que seja aplicado às pequenas e médias empresas e também a órgãos públicos nestes moldes.

A seguir relacionamos os principais benefícios que a implantação de uma ouvidoria traz para as organizações, respeitando algumas premissas abordadas na dissertação de [ALVES JR. 2002] e que foram consolidadas no site da empresa:

- Aproximação com os clientes-cidadãos, mediante o estabelecimento de um canal de comunicação acessível e direto;

- Identificação das áreas que estejam merecendo maior atenção dos dirigentes, definindo-se eixos prioritários de ação;
- Feedback sobre a atuação da organização, permitindo a correção de disfunções e redirecionamento das ações desenvolvidas;
- Identificação de necessidades dos usuários;
- Relacionamento democrático com a sociedade;
- Maior credibilidade e fortalecimento da imagem da organização junto à população;
- Inovação de processos produtos e serviços, a partir da participação do ouvidor no processo de discussão de políticas institucionais, propondo ações interventivas.

E partindo destas premissas é que o Sistema Informatizado para Gestão de Ouvidorias foi desenvolvido, estreitando assim o canal de comunicação entre os clientes-cidadãos e as organizações.

## **1.2. Motivação**

A partir do trabalho anterior de [BRAND 2006], que é um dos sócios da empresa OMD Soluções para Ouvidorias, foi identificada a necessidade de se alterar a modelagem de sua aplicação principal a fim de tornar todo o seu código compatível com a camada de persistência desenvolvida anteriormente.

Após a saída da incubadora GeNESS (Centro de Geração de Novos Empreendimentos em Software e Serviços) em 2006, tendo adquirido um grau de maturidade e querendo se consolidar no mercado, a empresa observou a necessidade de remodelar o sistema para deixá-lo mais robusto e facilitando sua implantação, pois a nova versão do sistema adaptar-se-ia ao ambiente de hardware e software já existente em seus potenciais clientes.

Neste período a necessidade de um tema para a conclusão de curso fez com que os autores deste trabalho aceitassem este desafio de construir a segunda versão do software, que teria seu nome alterado de "Sistema Informatizado para Gerenciamento de Ouvidorias" para "Sistema Informatizado para Gestão de Ouvidorias", conforme as regras descritas na seção trabalhos futuros de [BRAND 2006], onde seriam construídas as camadas de Apresentação e Lógica para deixá-lo no padrão J2EE multicamadas, tornando a aplicação mais flexível, segura e de fácil manutenção.

## **2. Sistema Informatizado para Gestão de Ouvidorias**

Conforme mencionado por [BRAND 2006], o Sistema Informatizado para Gestão de Ouvidorias é inovador na sua área, por se tratar de um sistema totalmente web, possibilitando ter um controle das manifestações apresentadas à organização, desde seu recebimento, passando por seu encaminhamento para as áreas responsáveis, setores ou departamentos dentro da organização, até a resposta ao cliente-cidadão, revisão desta resposta e encerramento da manifestação.

Para que o sistema funcione de acordo devem-se obedecer algumas etapas anteriores e posteriores a implantação conforme figura abaixo:

## ))) Etapas de organização



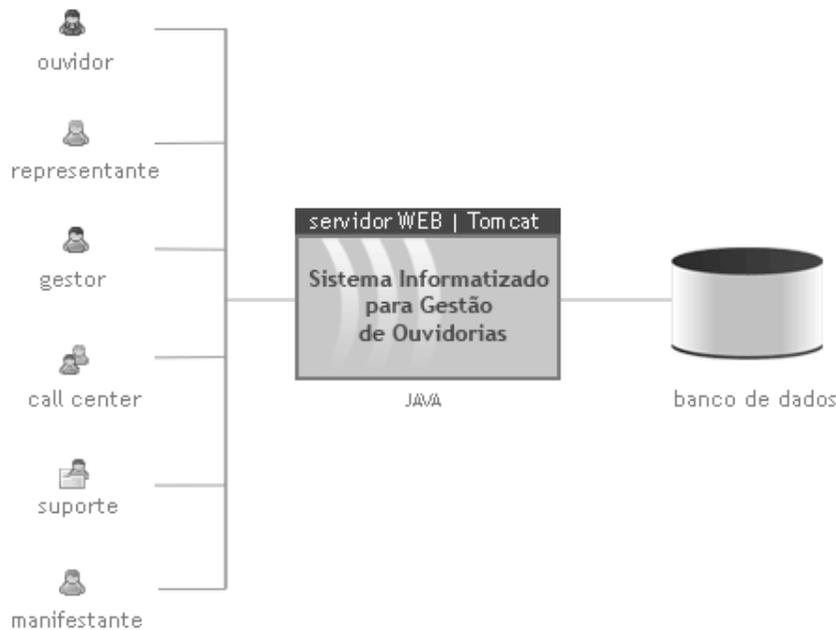
**Figura 1 – Etapas para implantação de Ouvidorias. Fonte: [OMD]**

Por este motivo, como citado por [ALVES JR 2004] no artigo Garantindo a efetividade da Ouvidoria, de nada adianta a organização possuir uma Ouvidoria para receber as manifestações de seus clientes-cidadãos se ela não estiver disposta a mudar seus processos e corrigir suas disfunções. E para que essas mudanças tenham efeito é necessário saber quais as unidades e assuntos que merecem um maior foco dos gestores, sem esquecer que o principal interessado continua sendo o cliente-cidadão. Novamente, percebe-se a necessidade de uma ferramenta que forneça informações aos gestores para que assim possam identificar melhor os problemas e apoiá-los no processo de tomada de decisão, principalmente à medida que o número de manifestações torna-se maior. O sistema deve ainda permitir o refinamento das informações, compreendendo desde uma visão macro da organização, passando pelas suas unidades até a visão detalhada de uma manifestação específica.

Abaixo explicitamos as funcionalidades que torna este sistema diferenciado comparado com outros sistemas existentes no mercado:

- Acompanhamento da situação da manifestação pelo próprio manifestante (via Internet), com acesso por meio de código de segurança;
- Possibilidade de tratamento especial para manifestações sigilosas;
- Emissão de relatórios gerenciais dinâmicos, com a utilização do recurso "drill-down", permitindo gerar informações estatísticas abrangendo toda a organização (visão macro) até uma manifestação específica (visão micro), facilitando a identificação das áreas que estejam merecendo maior atenção dos gestores;
- Possibilidade de efetuar diversas configurações no sistema mesmo quando ativo (configuração on-line).

Podemos ter uma melhor visão do sistema como um todo a partir da figura abaixo.



**Figura 2 – Arquitetura Geral do Sistema. Fonte: [OMD]**

A partir da figura podemos explicitar melhor o modelo de camadas, onde a direita vemos a camada de apresentação subdividida pelos tipos de usuários com acesso ao sistema. Ao centro a camada lógica, onde estão implementadas as regras do negócio e a esquerda a camada de persistência representada pelo banco de dados.

### 3. Tecnologias Utilizadas

#### 3.1. Proposta J2EE sem EJB

Com base nos estudos de [JOHNSON&HOELLER 2004] as arquiteturas J2EE tradicionais geralmente eram motivos de stress e trabalho pesado. Isso tudo porque seu desenvolvimento e manutenção envolviam um nível de complexidade que excediam o das próprias regras de negócio e requisitos que o sistema deveria atender, o que incluía dependência da API padronizada pela Sun, conhecida como Enterprise Java Beans (EJB).

Trabalhar com EJB significa estar envolvido em uma série de conceitos que fazem parte do mesmo, como:

- Problemas na portabilidade entre servidores de aplicação J2EE, pois geralmente a migração entre servidores de aplicação significa envolver compilação ou modificação de código, além de limitar a possibilidade de escolha do mesmo. Não seria possível utilizar um contêiner web como o TomCat por exemplo.

- Perda de produtividade no desenvolvimento dos aplicativos J2EE, porque no desenvolvimento e manutenção estão envolvidos uma série de arquivos de configuração, além de envolver também a criação de outros artefatos, que não são diretamente ligados a lógica do negócio, como por exemplo: a interface home e a interface local de um bean. Multiplicando isso, pelo número total de classes, que fazem parte da lógica de negócio do sistema, a complexidade no desenvolvimento do projeto é incrementada consideravelmente. Este fator leva o usuário a buscar alternativas de ferramentas como XDoclet para automatizar esse processo, porém é um tipo de dependência desnecessário.

Quando o principal, e muitas vezes o único, interesse em utilizar um container J2EE é somente o gerenciamento de transações, no que ele é perfeitamente usual, este projeto encaixa-se perfeitamente, não necessitando de serviços como clusterização ou endereçamento remoto (RMI) embutido no EJB.

Com o surgimento do framework Spring, que será apresentado no item 3.3, utilizar EJB apresenta grandes vantagens apenas quando é realmente necessário realizar chamadas remotas de objetos, ou seja, somente quando se utiliza SessionBeans Stateless e fazendo as chamadas com sua interface Remote (quando existe a necessidade de se ter dois servidores de aplicação), entretanto, se utilizar a interface Local, esta pode ser substituída com grandes vantagens pelo Spring pois neste caso usa-se POJOs (Plain Old Java Object), onde pode-se controlar transações (programáticas e declarativas), integração com as camadas de persistência e web, obtendo-se um desempenho melhor e consumindo menos recursos .

Simplicidade e foco na regra de negócio e rapidez no processo de desenvolvimento são as motivações principais na escolha deste tipo de arquitetura. Alternativas de componentes da camada de persistência, como Hibernate ou JPOX, por exemplo, já são motivos suficientes para questionar as Entities Beans como componentes de acesso aos dados.

Ainda reforçando o conceito de Orientação a Objetos, as chamadas entre classes ocorrem, endereçando as suas respectivas interfaces e não suas implementações. Além disso, é necessária a existência de aplicações falsas, como Data Transfer Objects, que nada mais são do que classes com funções de get e set, não envolvendo lógica alguma de negócio. Neste trabalho é utilizado o padrão de projeto Domain Model, que cria uma teia de relações entre os objetos do problema, basicamente seguindo o comportamento exigido pelas regras de negócio do sistema desenvolvido.

Baseado nas informações anteriores foi adotada a metodologia de Arquitetura de Contêiner Leve (Lightweight Container Architecture). Os contêineres leves ajudam a montar componentes de projetos diferentes em uma aplicação coesa. Por trás destes contêineres há um padrão comum de como eles fazem a amarração, o conceito é conhecido como Inversão de Controle (Inversion of Control ou IoC). Seu tipo mais popular e amplamente utilizado é a Dependency Injection (Injeção de Dependências).

A idéia básica da Injeção de Dependências é ter um objeto separado, chamado montador, que popula um campo em um objeto com uma implementação apropriada para a sua interface. Existem três principais formas de Injeção de Dependências: a forma Constructor Injection (Injeção por Construtores); a Setter Injection (Injeção por Métodos Set) e a Interface Injection (Injeção por Interfaces).

Ao decidirmos pela utilização do Framework Struts e Spring (abordados mais a fundo num próximo tópico) utilizaremos a Injeção por Métodos Set, onde a Action do Struts utiliza-se de setters pra tudo que ela precisar e o Controlador do Spring chama esse setter passando o parâmetro esperado.

Neste tipo de arquitetura, a camada de negócio é composta de POJOs, que são simples objetos Java e utilizando a Injeção de Dependência, propõe-se a lidar com as seguintes funções:

- Gerenciar o ciclo de vida de objetos de negócio;
- Controlar o gerenciamento de dependência entre objetos da camada de negócio;
- Utilizar o conceito de Orientação a Aspectos, provendo serviços como controle de transação declarativa dos objetos em execução;

- Gerenciar as transações declarativas;
- Integrar com outros frameworks e tecnologias como o Struts e o JDO.

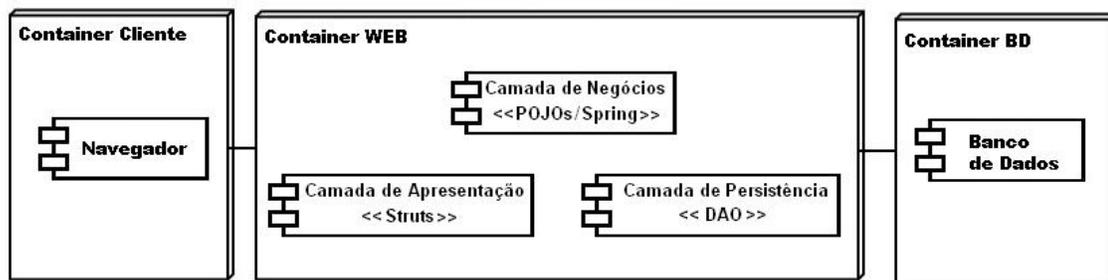
O uso desta Injeção de Dependência pode ajudar a facilitar a visualização de quais são as dependências do componente. Você pode simplesmente analisar o mecanismo de injeção, como os métodos set, para encontrá-las.

Outra razão pela preferência da Injeção de Dependência é que ela facilita a realização de testes, onde se consegue resultados rápidos e eficazes apenas substituindo implementações de serviços reais por stubs ou objetos falsos (mock).

Com os contêineres leves todas as classes JAVA são executadas pela mesma máquina virtual, ao contrário de uma arquitetura J2EE típica, onde as classes presentes na camada de apresentação e da camada de negócio, executam em máquinas virtuais diferentes, uma para cada camada.

Um lightweight container não está estritamente vinculado à arquitetura J2EE, sendo assim possível de ser executado tanto em um Contêiner Web, como Tomcat, quanto através de aplicativos stand-alone ou mesmo através de um container EJB.

A seguir uma figura que exemplifica bem o que estamos sugerindo, com as camadas bem definidas:



**Figura 3 – Arquitetura de Contêiner Leve**

Uma das razões que reforçam o conceito de Contêiner Leve é que não é necessária a implementação de padrões como Singleton ou Factory para cada objeto sugerido na camada de negócio.

Outra característica importante é não impor requisitos especiais para a execução dos objetos da aplicação, facilitando a suas inicializações sem a necessidade de passos adicionais para fazer o deploy da aplicação.

### 3.2. Struts

O Struts é um framework que implementa o modelo MVC (Model-View-Controller), e que surgiu para melhorar o desenvolvimento de aplicações em Smalltalk e virou um padrão utilizado em diversas linguagens. Com ele podemos facilmente configurá-lo para construir novas aplicações, além de fornecer uma melhor separação entre a camada de apresentação e a camada lógica. Sua primeira versão surgiu em 2001, como parte do projeto Jakarta da Apache Software Foundation e devido ao seu sucesso tornou-se um projeto do Apache.

O nome Struts vem das estruturas que sustentam edificações, assim struts dá sustentação às aplicações. Ele possui a classe Action Servlet que representa o Controle da aplicação, em geral não é necessário estendê-la, pois ele delega tarefa de acordo com o

arquivo `struts-config.xml`, fazendo com que as modificações tenham impacto diminuído sobre a aplicação.

Quando as requisições chegam, elas passam pelo Controle e então ele aponta para quem deve tratá-la, se a requisição for composta por formulário(s) ela é repassada as classes que estendem `ActionForm`, caso contrário é encaminhado diretamente a classe que estende `Action`. A classe `ActionForm` fica responsável por guardar os dados do formulário e validá-los. No caso dos dados estarem incorretos, a requisição volta para a página que a iniciou. Se os dados estiverem corretos esta requisição será passada a uma classe de `Action` que é quem faz algum processamento sobre a requisição, consultando o modelo e colocando objetos para serem exibidos na camada de apresentação.

Para este trabalho, utilizou-se de uma subclasse especializada de `ActionForm` chamada `DynaActionForm`, que permite a criação de `FormBeans` dinâmicos utilizando-se de configurações no arquivo de propriedades `struts-config.xml`, sem requerer do desenvolvedor a criação de uma classe Java para cada tipo de `FormBean`. `DynaActionForm` elimina a necessidade da classe de `FormBean` fazendo com que o tempo de desenvolvimento seja reduzido.

### 3.3. Spring

Criado por Rod Johnson para amenizar os problemas encontrados ao utilizar a tecnologia de Enterprise Java Beans (EJB), parte principal do J2EE. A pedra fundamental do Spring Framework é o conceito de Dependency Injection (Injeção de Dependências - DI) que é uma forma de Inversion Of Control (Inversão de Controle – IoC) na qual um componente não instancia suas dependências mas o ambiente (contêiner) automaticamente as fornece.

Como o Spring é responsável pelo controle do ciclo de vida dos beans existentes na camada de negócio, e pela busca de cada recurso, minimizando a dependência do código desenvolvido do container em questão.

Ao utilizar o Spring Framework temos como grande vantagem o fato de não interferir diretamente na modelagem da aplicação. Ele afeta a arquitetura, conjunto de decisões sobre a divisão de um sistema em módulos e a forma como estes módulos interagem, do sistema, mas os objetos de negócio não sofrem muitos impactos. Além disso, o Spring induz ao uso de boas práticas de programação ao se baseia ostensivamente em interfaces ao invés de classes. Esta característica, além de permitir extensão e adaptação fácil do framework e ser um conceito da Orientação a Objetos, possibilita o uso de testes unitários com muita facilidade.

Utiliza-se o Spring para intermediar esta arquitetura de controlador web com a interface de acesso ao banco de dados principalmente quando eles são desenvolvidos por equipes diferentes, onde não é necessário o conhecimento da outra camada para continuidade do projeto.

O Spring é um framework abrangente voltado para o desenvolvimento Java corporativo. Ele inclui a possibilidade de criação de transações concentra e abstratas (reduzindo assim o tamanho de arquivos de configuração xml), bem como, fazer a integração com frameworks de persistência, e no desenvolvimento de aplicações web e JDBC.

A principal razão de sua escolha é o controle de transações. O uso de transações declarativas possibilita que o desenvolvedor de componentes construa o componente sem precisar demarcar transações, evita a introdução de erros que surgem decorrentes do esquecimento de sessões com o banco de dados abertas e não confirmadas, já que o contêiner se encarrega disto. Isso incentiva o reuso, pois o componente é customizado com uma política

de segurança mais ou menos rígida, dependendo do escopo onde é usado, sem necessidade de mudanças no código fonte.

Além disso, o código utilizado na camada de negócio, em um momento acessando a camada de persistência através de conexões JDBC, poderá ser utilizado para acessar a base utilizando outra tecnologia de dados como JPOX ou Hibernate, executando a adaptação sem alterar o código fonte, mas sim o próprio framework. Assim o Spring saberá a quantidade de recursos de conexão disponíveis para ele, de acordo com a conexão escolhida, ou seja, a escalabilidade é gerenciada pelo container.

### **3.4. JSTL**

JSP Standard Tag Library (Biblioteca Padrão de Tags JSP) consiste em uma coleção de bibliotecas, tendo cada uma um propósito bem definido, que permitem escrever páginas JSPs sem código Java, aumentando assim a legibilidade do código e a interação entre desenvolvedores e web designers.

Uma página JSTL é uma página JSP contendo um conjunto de tags JSTLs onde cada tag realiza um determinado tipo de processamento e faz parte de uma biblioteca JSTL, sendo algo muito melhor do que a utilização de código Java dentro de JSP como ocorria no passado, na arquitetura de uma camada somente (Modelo 1). Pode-se utilizar várias bibliotecas JSTLs para apenas uma página JSTL.

Surgiu para criar páginas dinâmicas e complexas sem a necessidade de escrever código Java dentro delas, Fornecendo tags que tornam fáceis as tarefas que exigiriam várias linhas de código Java, como formatação de números e datas seguindo configurações regionais de cada usuário, iteração de listas, operações condicionais, tags para manipular documentos XML, internacionalização e tags SQL (O uso desta tag não é aconselhável pois utiliza diretamente o bancos tornando o sistema com apenas uma camada). Também introduz uma nova linguagem de expressões que simplifica o desenvolvimento de páginas JSP.

## **4. Comparativo de versões**

Ao final do Desenvolvimento das Camadas de Apresentação e Lógica, pode-se realizar uma análise das versões 1.0 e 2.0 do software. Para isto deve-se levar em conta os seguintes critérios: portabilidade, segurança e manutenção.

### **4.1. Portabilidade**

A maior portabilidade do software ficou visível, uma vez que a versão anterior exigia determinadas características para ser implantada, com a implementação atual, com pequenas modificações no código fonte, pode-se instalá-lo em diferentes plataformas.

### **4.2. Segurança**

Com o uso de transações declarativas possibilitou-se ao desenvolvedor que se construa os componentes sem precisar demarcar transações, incentivando o reuso já que o escopo da transação é determinado sem que seja necessário recompilar o código fonte do componente. Isto também evita a introdução de erros que podem surgir decorrentes do esquecimento de sessões com o banco de dados abertas e não confirmadas, já que o contêiner se encarrega disto.

Na versão 1.0 tínhamos os dados acessados diretamente da apresentação para o banco de dados, o que era um risco que o sistema corria.

### **4.3. Manutenção**

Utilizando camadas bem definidas e transparentes a versão 2.0 facilita as futuras manutenções do software, pois sabe-se exatamente onde devem ser feitas as correções ou melhorias após a reestruturação das classes do domínio do problema e suas respectivas funções no sistema.

## **5. Resultados**

No trabalho anterior, [BRAND 2006] adéqua a primeira versão do sistema, que inicialmente utilizava o contêiner Servlet TomCat que acessava diretamente o banco de dados MySQL com JDBC, para deixá-lo nos moldes dos softwares atuais utilizando o paradigma de orientação à objetos e o desenvolvimento em multicamadas. Para isto foi feito um estudo de caso e remodelado o sistema de modo a integrar o aspecto de multicamadas. Na ocasião, o foco foi criar a camada de persistência para tornar o sistema independente do banco de dados.

A partir deste trabalho, foi realizado um novo estudo para tornar a nova versão do software totalmente multicamada, o que garantiria maior segurança das informações, maior flexibilidade na implantação e menor esforço na manutenção.

A estruturação e implementação das camadas Web e Lógica para o sistema juntamente com a camada de Persistência desenvolvida anteriormente fará com que os custos com a manutenção do software sejam reduzidos e proporcionam à organização oferecer seus serviços a variadas infra-estruturas disponíveis em seus clientes alterando o mínimo de código possível de acordo com a infra-estrutura disponibilizada pelo cliente.

Após a comparação efetuada no capítulo anterior entre a atual camada que utiliza Servlet/JSP TomCat, com as novas camadas de Apresentação e Lógica construídas com o auxílio dos frameworks Strus e Spring pode-se afirmar que o desenvolvimento em multicamadas é bastante benéfico, principalmente com a modelagem de Contêineres Leves, pois utiliza-se da estrutura J2EE que é bem difundida, porém sem a necessidade de esforços dispendiosos com o EJB para aplicações que não utilizarão as chamadas remotas, que foi o caso proposto.

O sucesso desta modelagem juntamente com as ferramentas propostas, lembrando que a escolha dos frameworks em questão foi puramente por se adequarem ao design da arquitetura com lightweight container, que por sua vez oferece a simplicidade e ainda assim os recursos suficientes pra implementar um sistema do tipo que é a Gestão de Ouvidorias, foi posto em prova na primeira implantação deste sistema em uma empresa real, onde necessitou-se que o código fonte fosse adaptado para que o acesso ao banco de dados utilizado na organização, que possui licença para uso, diferentemente do banco de dados com código livre utilizado nos testes de desenvolvimento.

Levando-se em consideração o tempo de programação utilizado para tal adaptação, pode-se concluir que a remodelagem do sistema proporcionará a empresa uma diminuição nos custos de manutenção e implantação do software e que os objetivos deste projeto foram alcançados com sucesso.

## **Referências**

[BRAND 2006] BRAND, Rony Reinehr. Persistência de Dados em Java: Um Estudo Aplicado ao Sistema Informatizado para Gerenciamento de Ouvidorias. Dissertação de Graduação. UFSC, Florianópolis, 2006. Disponível em <[http://www.ond.com.br/download/tcc\\_rony.pdf](http://www.ond.com.br/download/tcc_rony.pdf)>. Acesso em 05 de Setembro de 2006.

- [ALVES JR 2002] ALVES JR., Mário Nelson. A contribuição estratégica das ouvidorias para a melhoria dos serviços prestados pelas organizações: Um estudo de caso na Secretaria de Estado da Saúde de Santa Catarina. Dissertação de Mestrado. UDESC, Florianópolis, 2002. Disponível em <<http://www.ond.com.br/download/disserta.pdf>>. Acesso em 13 de Fevereiro de 2007.
- [OND] OND Soluções para Ouvidorias. Disponível em: <<http://www.ond.com.br>>. Acesso em 27 de Maio de 2007.
- [JOHNSON&HOELLER 2004] JOHNSON, Rod; HOELLER, Juergen. Expert one-on-one J2EE development without EJB. EUA, Wrox, 2004.
- [TODD& SZOLKOWSKI 2003] TODD, Nick ; SZOLKOWSKI, Mark: Java Server Pages: O guia do desenvolvedor. Campus, 2003
- [HUSTED 2004] HUSTED, Ted; DUMOULIN, Cedric; FRANCISCUS, George; WINTERFELDT, David : Struts em ação. Moderna, 2004.
- [STRUTS] Apache Struts Framework. Disponível em: <<http://struts.apache.org/1.2.9/index.html>>. Acesso em 13 de Setembro de 2006.
- [CORREA 2004] CORRÊA, Alice Alves. Avaliação do Framework Struts para Implementação de Aplicações Web usando Padrão Modelo-Visão-Controlador. Dissertação de Graduação. UFSC, Florianópolis, 2004. Disponível em <[http://projetos.inf.ufsc.br/arquivos\\_projetos/projeto\\_6/tcc\\_struts\\_final\\_completo.pdf](http://projetos.inf.ufsc.br/arquivos_projetos/projeto_6/tcc_struts_final_completo.pdf)>. Acesso em 15 de Setembro de 2006.
- [JSPINSIDER] JSP Insider. Disponível em: <<http://www.jspinsider.com/content/jsp/struts/strutsintro.jsp>>. Acesso em 03 de Março de 2007.
- [SPRING] Spring Framework. Disponível em: <<http://www.springframework.org>>. Acesso em 13 de Setembro de 2006.
- [WALLS&BREIDENBACH 2005] WALLS, Craig; BREIDENBACH, Ryan. Spring in Action. EUA, Manning, 2005.
- [RAIBLE 2004] RAIBLE, Matt. Spring Live. EUA, SourceBeat, 2004
- [CALÇADO 2006] CALÇADO, Phillip. Curso Spring Framework. Disponível em: <<http://fragmental.com.br/files/cursos/spring-112006/apostila-spring-112006.pdf>>. Acesso em 01 de Dezembro de 2006.
- [BAYERN 2002] BAYERN, Shawn. JSTL in Action. EUA, Manning Publications Co., 2002.
- [BERGSTEN 2003] BERGSTEN, Hans. JavaServer Pages 3rd Edition. EUA, O'Reilly, 2003.

## Anexo 2 – Telas do Software

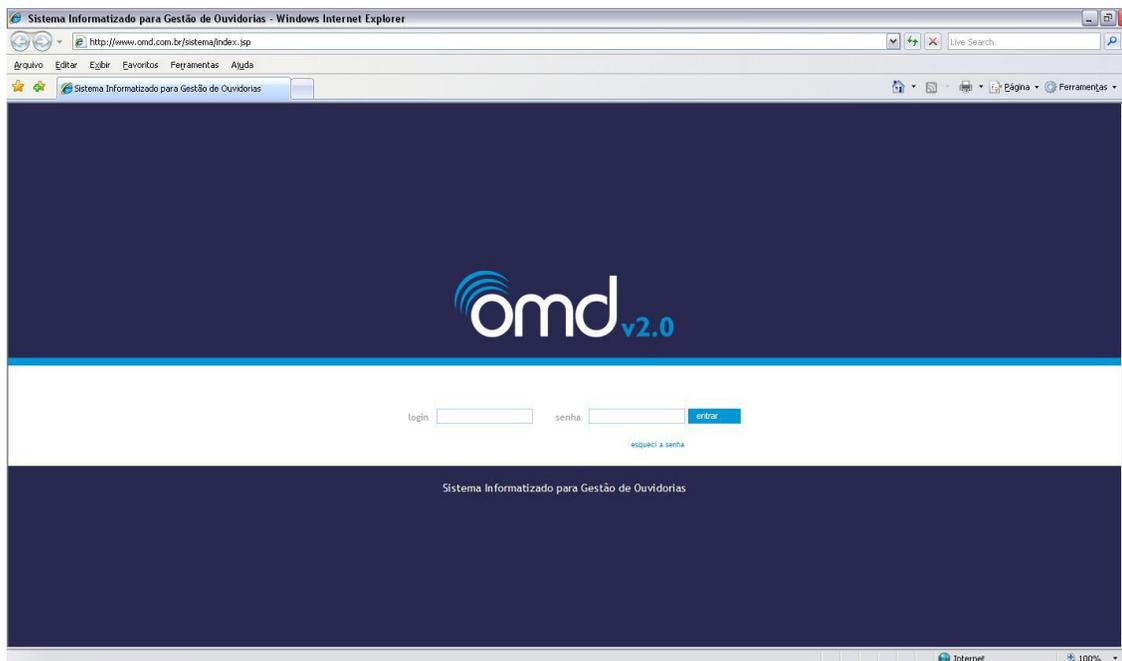


Figura 38 – Tela inicial do sistema

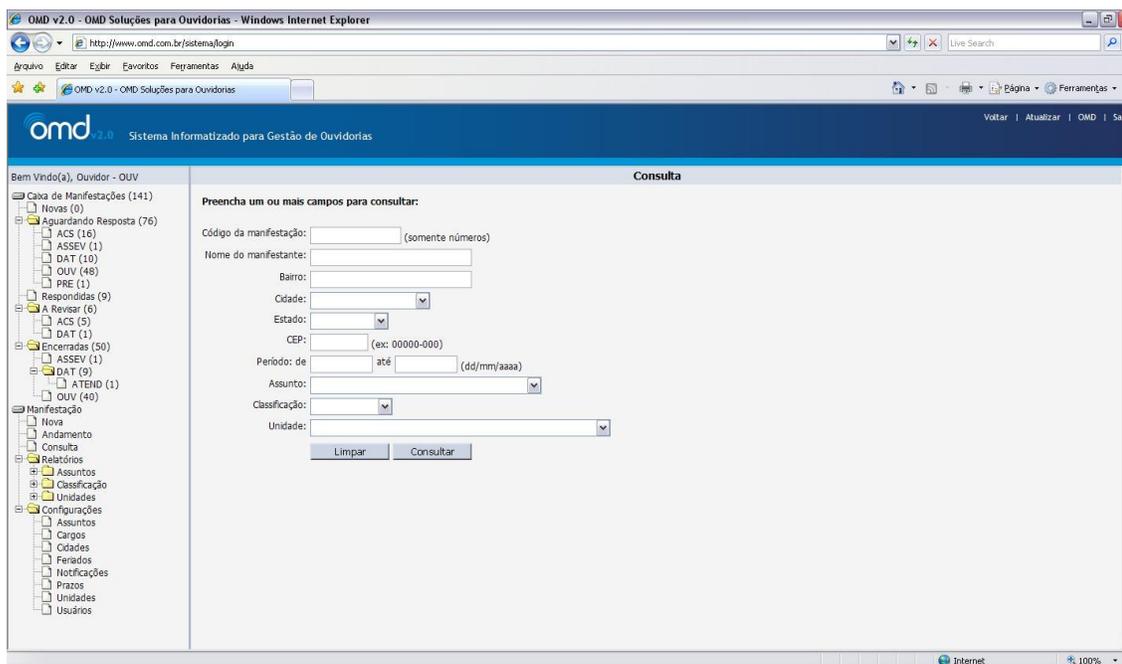


Figura 39 – Tela de consulta de manifestações

OMD v2.0 - OMD Soluções para Ouvidorias - Microsoft Internet Explorer

Arquivo Editar Exibir Favoritos Ferramentas Ajuda

Endereço http://www.umd.com.br/sistema/login Ir Links

Volta | Atualizar | OMD | Sair

Bem Vindo(a), Ouvidor - OUV

**Nova Manifestação**

**Dados da Ocorrência do Fato**

Data\*: 16/06/2006 (dd/mm/aaaa) Hora\*: 14:00 (hh:mm)

Unidade\*: Posto de Atendimento de Imitubua

Envolvidos: Atendente Fulano

Descrição\*: Houve discussão com o atendente em questão quanto aos procedimentos de atendimento.

**Dados do Cadastro**

Manifestação\*: Reclamação Forma de Entrada\*: Internet

Identificação\*: Aberta Forma de Resposta\*: E-mail

Assunto\*: Atendimento - Meu

\* - Campos Obrigatórios

Voltar Limpar Continuar

Figura 40 – Tela de inclusão de manifestações

OMD v2.0 - OMD Soluções para Ouvidorias - Microsoft Internet Explorer

Arquivo Editar Exibir Favoritos Ferramentas Ajuda

Endereço http://www.umd.com.br/sistema/login Ir Links

Volta | Atualizar | OMD | Sair

Bem Vindo(a), Ouvidor - OUV

**Dados do Manifestante**

Nome\*: Sicreno de Tal

E-mail\*: email@com.br

Telefone 1: (ddd telefone)

Telefone 2: (ddd telefone)

\* - Campos Obrigatórios

Voltar Limpar Confirmar

Figura 41 – Tela de dados do manifestante

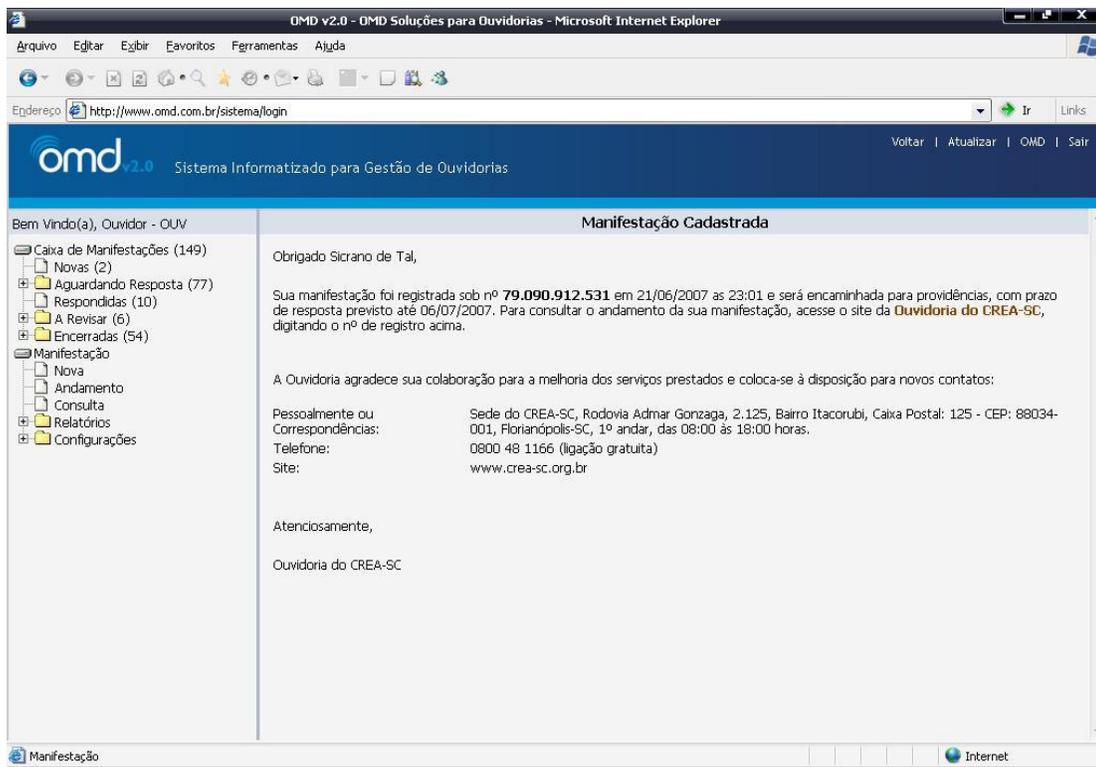


Figura 42 – Tela de manifestação cadastrada

### Anexo 3 - Código Fonte

**context.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<Context debug="0" docBase="tccOuvidoria" path="/ouvidoria" reloadable="true">
  <Logger className="org.apache.catalina.logger.FileLogger" prefix="tccOuvidoria."
suffix=".log" timestamp="true"/>
  <Resource auth="Container" name="jdbc/ouvidoria" type="javax.sql.DataSource"/>
  <ResourceParams name="jdbc/ouvidoria">
    <parameter>
      <name>factory</name>
      <value>org.apache.commons.dbcp.BasicDataSourceFactory</value>
    </parameter>
    <parameter>
      <name>maxActive</name>
```

```
<value>100</value>
</parameter>
<parameter>
  <name>maxIdle</name>
  <value>10</value>
</parameter>
<parameter>
  <name>maxWait</name>
  <value>10000</value>
</parameter>
<parameter>
  <name>username</name>
  <value>ouvidoria</value>
</parameter>
<parameter>
  <name>password</name>
  <value>Consulta</value>
</parameter>
<parameter>
  <name>driverClassName</name>
  <value>com.mysql.jdbc.Driver</value>
</parameter>
<parameter>
  <name>url</name>
  <value>jdbc:mysql://localhost:3306/ouvidoria_crea-
sc?autoReconnect=true</value>
</parameter>
</ResourceParams>
</Context>
```

### ***struts-config.xml***

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
```

```
<!DOCTYPE struts-config PUBLIC
```

```
"-//Apache Software Foundation//DTD Struts Configuration 1.2//EN"
```

```
"http://jakarta.apache.org/struts/dtds/struts-config_1_2.dtd">
```

```
<struts-config>
```

```
  <form-beans>
```

```
    <form-bean type="org.apache.struts.action.DynaActionForm"  
name="LoginFormBean">
```

```
      <form-property type="java.lang.String" name="login"/>
```

```
      <form-property type="java.lang.String" name="senha"/>
```

```
    </form-bean>
```

```
    <form-bean type="org.apache.struts.action.DynaActionForm"  
name="CargoFormBean">
```

```
      <form-property type="java.lang.String" name="nome"/>
```

```
      <form-property type="java.lang.String" name="cargoconsulta"/>
```

```
      <form-property type="java.lang.String" name="cargoesedit"/>
```

```
      <form-property type="java.util.Collection" name="listaCargos"/>
```

```
      <form-property type="java.lang.String" name="idCargosAddCombo"/>
```

```
      <form-property type="java.lang.String" name="idCargosEditarCombo"/>
```

```
      <form-property type="java.lang.String" name="idCargosRemoverCombo"/>
```

```
      <form-property type="java.lang.String" name="senha"/>
```

```
    </form-bean>
```

```
    <form-bean type="org.apache.struts.action.DynaActionForm"  
name="AssuntoFormBean">
```

```
      <form-property type="java.lang.String" name="assuntounovo"/>
```

```
      <form-property type="java.lang.String" name="assuntoconsulta"/>
```

```
      <form-property type="java.lang.String" name="assuntoalterado"/>
```

```
      <form-property type="java.util.Collection" name="listaAssuntos"/>
```

```
      <form-property type="java.util.Collection" name="listaOutrosAssuntos"/>
```

```
      <form-property type="java.lang.String" name="idAssuntosAddCombo"/>
```

```
      <form-property type="java.lang.String" name="idAssuntosEditarCombo"/>
```

```
      <form-property type="java.lang.String" name="idAssuntosRemoverCombo"/>
```

```
      <form-property type="java.lang.String" name="idOutrosAssuntosCombo"/>
```

```
      <form-property type="java.lang.String" name="idAssuntosChangeCombo"/>
```

```
      <form-property type="java.lang.String" name="senha"/>
```

```
</form-bean>
```

```
<form-bean type="org.apache.struts.action.DynaActionForm"  
name="FeriadoFormBean">  
  <form-property type="java.lang.String" name="feriadonovo"/>  
  <form-property type="java.lang.String" name="feriadoalterado"/>  
  <form-property type="java.lang.String" name="datainicio"/>  
  <form-property type="java.lang.String" name="datafim"/>  
  <form-property type="java.util.Collection" name="feriados"/>  
  <form-property type="java.lang.String" name="idFeriadosConsultarCombo"/>  
  <form-property type="java.lang.String" name="idFeriadosEditarCombo"/>  
  <form-property type="java.lang.String" name="idFeriadosRemoverCombo"/>  
  <form-property type="java.lang.String" name="senha"/>  
</form-bean>
```

```
<form-bean type="org.apache.struts.action.DynaActionForm"  
name="UnidadeFormBean">  
  <form-property type="java.lang.String" name="consultaUnidade"/>  
  <form-property type="java.util.Collection" name="listaRepresentantes"/>  
  <form-property type="java.lang.String" name="idComboRepresentantes"/>  
  <form-property type="java.util.Collection" name="listaVinculos"/>  
  <form-property type="java.lang.String" name="idComboVinculos"/>  
  <form-property type="java.util.Collection" name="listaCidades"/>  
  <form-property type="java.lang.String" name="idComboCidades"/>  
  <form-property type="java.util.Collection" name="listaEstados"/>  
  <form-property type="java.lang.String" name="idComboEstados"/>  
  <form-property type="java.lang.String" name="nomeUnidade"/>  
  <form-property type="java.lang.String" name="siglaUnidade"/>  
  <form-property type="java.lang.String" name="enderecoUnidade"/>  
  <form-property type="java.lang.String" name="bairroUnidade"/>  
  <form-property type="java.lang.String" name="cepUnidade"/>  
  <form-property type="java.lang.String" name="dddUmUnidade"/>  
  <form-property type="java.lang.String" name="telUmUnidade"/>  
  <form-property type="java.lang.String" name="dddDoisUnidade"/>  
  <form-property type="java.lang.String" name="telDoisUnidade"/>
```

```
<form-property type="java.lang.String" name="dddFaxUnidade"/>
<form-property type="java.lang.String" name="faxUnidade"/>
<form-property type="java.lang.String" name="emailUnidade"/>
<form-property type="java.lang.String" name="senhaUnidade"/>
<form-property type="java.util.Collection" name="listaOutrasUnidades"/>
<form-property type="java.lang.String" name="idComboOutrasUnidades"/>
<form-property type="java.util.Collection" name="listaUnidades"/>
<form-property type="java.lang.String" name="idComboUnidadesTroca"/>
<form-property type="java.lang.String" name="idComboUnidadeExcluir"/>
<form-property type="java.lang.String" name="senha"/>
<form-property type="java.util.Collection"
name="listaUnidadesEncontradas"/>
<form-property type="java.util.Collection" name="unidadeExibirEdicao"/>
<form-property type="java.lang.String" name="unidadeIdEditar"/>
<form-property type="java.lang.String" name="unidadeNome"/>
<form-property type="java.lang.String" name="unidadeSigla"/>
<form-property type="java.lang.String" name="unidadeEndereço"/>
<form-property type="java.lang.String" name="unidadeBairro"/>
<form-property type="java.lang.String" name="unidadeCEP"/>
<form-property type="java.lang.String" name="unidadeRepresentante"/>
<form-property type="java.lang.String" name="unidadeVínculo"/>
<form-property type="java.lang.String" name="unidadeEmail"/>
<form-property type="java.lang.String" name="unidadeTelefone1"/>
<form-property type="java.lang.String" name="unidadeTelefone2"/>
<form-property type="java.lang.String" name="unidadeFax"/>
<form-property type="java.lang.String" name="unidadeCidade"/>
<form-property type="java.lang.String" name="unidadeEstado"/>
<form-property type="br.com.umd.ouvidoria.logica.domain.Unidade"
name="unidadeedit"/>
</form-bean>
```

```
<form-bean type="org.apache.struts.action.DynaActionForm"
name="UsuarioFormBean">
<form-property type="java.lang.String" name="consultaUsuario"/>
<form-property type="java.util.Collection" name="listaPerfis"/>
```

```
<form-property type="java.lang.String" name="idComboUsuarioPerfil"/>
<form-property type="java.util.Collection" name="listaCargos"/>
<form-property type="java.lang.String" name="idComboUsuarioCargo"/>
<form-property type="java.util.Collection" name="listaUnidades"/>
<form-property type="java.lang.String" name="idComboUsuarioUnidade"/>
<form-property type="java.lang.String" name="nomeCadastroUsuario"/>
<form-property type="java.lang.String" name="loginCadastroUsuario"/>
<form-property type="java.lang.String" name="apelidoCadastroUsuario"/>
<form-property type="java.lang.String" name="senhaCadastroUsuario"/>
<form-property type="java.lang.String" name="dddCadastroUsuario"/>
<form-property type="java.lang.String" name="telefoneCadastroUsuario"/>
<form-property type="java.lang.String" name="emailCadastroUsuario"/>
<form-property type="java.lang.String" name="nomeEdicaoUsuario"/>
<form-property type="java.lang.String" name="loginEdicaoUsuario"/>
<form-property type="java.lang.String" name="apelidoEdicaoUsuario"/>
<form-property type="java.lang.String" name="senhaEdicaoUsuario"/>
<form-property type="java.lang.String" name="telefoneEdicaoUsuario"/>
<form-property type="java.lang.String" name="dddEdicaoUsuario"/>
<form-property type="java.lang.String" name="emailEdicaoUsuario"/>
<form-property type="java.util.Collection" name="listaUsuarios"/>
<form-property type="java.lang.String" name="idComboUsuarioExcluir"/>
<form-property type="java.lang.String" name="senhaConferencia"/>
<form-property type="java.lang.String" name="idUsuario"/>
<form-property type="java.lang.String" name="nomeUsuario"/>
<form-property type="java.lang.String" name="apelidoUsuario"/>
<form-property type="java.lang.String" name="cargoUsuario"/>
<form-property type="java.lang.String" name="telefoneUsuario"/>
<form-property type="java.lang.String" name="emailUsuario"/>
<form-property type="java.lang.String" name="perfilUsuario"/>
<form-property type="java.lang.String" name="unidadeUsuario"/>
<form-property type="java.util.Collection" name="listaUsuariosEncontrados"/>
<form-property type="java.lang.String" name="usuarioldEditar"/>
<form-property type="java.util.Collection" name="unidadeExibirEdicao"/>
</form-bean>
```

```
<form-bean type="org.apache.struts.action.DynaActionForm"
name="PrazoFormBean">
    <form-property type="java.lang.String" name="prazoedit"/>
    <form-property type="java.lang.String" name="idPrazo"/>
    <form-property type="java.util.Collection" name="prazos"/>
</form-bean>

<form-bean type="org.apache.struts.action.DynaActionForm"
name="ManifestacaoFormBean">
    <form-property type="java.lang.String" name="data"/>
    <form-property type="java.lang.String" name="hora"/>
    <form-property type="java.lang.String" name="unidade"/>
    <form-property type="java.lang.String" name="outraUnidade"/>
    <form-property type="java.lang.String" name="envolvidos"/>
    <form-property type="java.lang.String" name="descricao"/>
    <form-property type="java.lang.String" name="identificacao"/>
    <form-property type="java.lang.String" name="modoRetorno"/>
    <form-property type="java.lang.Long" name="modoEntrada"/>
    <form-property type="java.lang.Long" name="classificacao"/>
    <form-property type="java.lang.String" name="assunto"/>
    <form-property type="java.lang.String" name="outroAssunto"/>
    <form-property type="java.lang.String" name="nome"/>
    <form-property type="java.lang.String" name="endereco"/>
    <form-property type="java.lang.String" name="bairro"/>
    <form-property type="java.lang.Long" name="estado"/>
    <form-property type="java.lang.String" name="cidade"/>
    <form-property type="java.lang.String" name="cep"/>
    <form-property type="java.lang.String" name="outraCidade"/>
    <form-property type="java.lang.String" name="dddTelefone1"/>
    <form-property type="java.lang.String" name="telefone1"/>
    <form-property type="java.lang.String" name="dddTelefone2"/>
    <form-property type="java.lang.String" name="telefone2"/>
    <form-property type="java.lang.String" name="fax"/>
    <form-property type="java.lang.String" name="dddFax"/>
    <form-property type="java.lang.String" name="email"/>
    <form-property type="br.com.umd.ouvidoria.logica.domain.Manifestacao"
```

```
name="manifestacao"/>
  </form-bean>
  <form-bean type="org.apache.struts.action.DynaActionForm"
name="NovasFormBean">
    <form-property type="java.lang.Long" name="codigo"/>
  </form-bean>
  <form-bean type="org.apache.struts.action.DynaActionForm"
name="EncaminharFormBean">
    <form-property type="java.lang.Long" name="codigo"/>
    <form-property type="java.lang.String" name="unidade"/>
    <form-property type="java.lang.String" name="outraUnidade"/>
    <form-property type="java.lang.String" name="envolvidos"/>
    <form-property type="java.lang.String" name="descricao"/>
    <form-property type="java.lang.Long" name="classificacao"/>
    <form-property type="java.lang.String" name="assunto"/>
    <form-property type="java.lang.String" name="outroAssunto"/>
    <form-property type="java.lang.Long" name="unidadeEncaminhar"/>
    <form-property type="java.lang.String" name="comentario"/>
  </form-bean>
  <form-bean type="org.apache.struts.action.DynaActionForm"
name="AguardandoRespostaFormBean">
    <form-property type="java.lang.Long" name="idUnidade"/>
  </form-bean>
  <form-bean type="org.apache.struts.action.DynaActionForm"
name="DetalhesFormBean">
    <form-property type="java.lang.Long" name="codigo"/>
  </form-bean>
  <form-bean type="org.apache.struts.action.DynaActionForm"
name="CidadeFormBean">
    <form-property type="java.lang.Long" name="id"/>
    <form-property type="java.lang.String" name="cidade"/>
    <form-property type="java.lang.Long" name="estado"/>
    <form-property type="java.lang.String" name="senhaConferencia"/>
    <form-property type="java.lang.String" name="outraCidade"/>
  </form-bean>
  <form-bean type="org.apache.struts.action.DynaActionForm"
```

```
name="ResponderFormBean">
    <form-property type="java.lang.Long" name="codigo"/>
    <form-property type="java.lang.String" name="data"/>
    <form-property type="java.lang.Long" name="unidade"/>
    <form-property type="java.lang.String" name="responsavel"/>
    <form-property type="java.lang.Long" name="cargo"/>
    <form-property type="java.lang.String" name="resposta"/>
</form-bean>

<form-bean type="org.apache.struts.action.DynaActionForm"
name="RespondidasFormBean">
    <form-property type="java.lang.Long" name="codigo"/>
</form-bean>

<form-bean type="org.apache.struts.action.DynaActionForm"
name="ConsultaFormBean">
    <form-property type="java.lang.String" name="codigo"/>
    <form-property type="java.lang.String" name="nome"/>
    <form-property type="java.lang.String" name="bairro"/>
    <form-property type="java.lang.String" name="idEstado"/>
    <form-property type="java.lang.String" name="idCidade"/>
    <form-property type="java.lang.String" name="cep"/>
    <form-property type="java.lang.String" name="datainicio"/>
    <form-property type="java.lang.String" name="datafim"/>
    <form-property type="java.lang.String" name="idAssunto"/>
    <form-property type="java.lang.String" name="idClassificacao"/>
    <form-property type="java.lang.String" name="idUnidade"/>
</form-bean>

<form-bean type="org.apache.struts.action.DynaActionForm"
name="EnviarRespostaFormBean">
    <form-property type="java.lang.Long" name="codigo"/>
    <form-property type="java.lang.Integer"
name="numeroRespostaSelecionada"/>
</form-bean>
```

```

    <form-bean type="org.apache.struts.action.DynaActionForm"
name="RelatorioFormBean">
        <form-property type="java.lang.String" name="dataInicial"/>
        <form-property type="java.lang.String" name="dataFinal"/>
        <form-property type="java.lang.String" name="usuario_unidade_id"/>
    </form-bean>

    <form-bean type="org.apache.struts.action.DynaActionForm"
name="MenuFormBean">
        <form-property type="java.lang.String" name="usuario_unidade_id"/>
        <form-property type="java.lang.String" name="unidade_id"/>
        <form-property type="java.lang.String" name="unidade"/>
    </form-bean>

    <form-bean type="org.apache.struts.action.DynaActionForm"
name="EncerradasFormBean">
        <form-property type="java.lang.Long" name="idUnidade"/>
    </form-bean>

    <form-bean type="org.apache.struts.action.DynaActionForm"
name="ARevisarFormBean">
        <form-property type="java.lang.Long" name="idUnidade"/>
    </form-bean>
</form-beans>
<global-exceptions>

</global-exceptions>

<global-forwards>
    <forward name="index" path="/index.do"/>
</global-forwards>

<action-mappings>
    <action path="/login"
type="br.com.umd.ouvidoria.web.controle.action.LoginAction"
name="LoginFormBean">
        <forward name="success" path="/WEB-INF/jsp/inicial.jsp"></forward>
        <forward name="errologin" path="/WEB-INF/jsp/index.jsp"></forward>
    </action>
</action-mappings>

```

```
</action>

<action path="/index" forward="/WEB-INF/jsp/index.jsp"></action>
<action path="/inicial" forward="/WEB-INF/jsp/inicial.jsp"></action>
<action path="/bemvindo" forward="/WEB-INF/jsp/bemVindo.jsp"></action>
<action path="/topo" forward="/WEB-INF/jsp/topo.jsp"></action>
<action path="/separador" forward="/WEB-INF/jsp/separador.htm"></action>
<action path="/sair" forward="/WEB-INF/jsp/sair.jsp"></action>
<action path="/externo/consulta" forward="/WEB-
INF/jsp/externo/consultaAndamento.jsp"></action>

<action path="/menu"
type="br.com.omb.ouvidoria.web.controle.action.MenuAction"
name="MenuFormBean">
    <forward name="success" path="/WEB-INF/jsp/menu.jsp"></forward>
</action>

<action path="/configuracoes/cargos"
type="br.com.omb.ouvidoria.web.controle.action.CargoAction"
name="CargoFormBean">
    <forward name="success" path="/WEB-
INF/jsp/configuracoes/cargos.jsp"></forward>
    <forward name="resultadoConsulta" path="/WEB-
INF/jsp/configuracoes/resultadoConsulta.jsp"></forward>
    <forward name="errologin" path="/WEB-INF/jsp/index.jsp"></forward>
    <forward name="resultadoSucesso" path="/WEB-
INF/jsp/configuracoes/resultado.jsp"></forward>
    <forward name="erro" path="/WEB-
INF/jsp/configuracoes/erro.jsp"></forward>
</action>

<action path="/configuracoes/feriados"
type="br.com.omb.ouvidoria.web.controle.action.FeriadoAction"
name="FeriadoFormBean">
    <forward name="success" path="/WEB-
INF/jsp/configuracoes/feriados.jsp"></forward>
    <forward name="resultadoConsulta" path="/WEB-
INF/jsp/configuracoes/resultadoConsultaFeriado.jsp"></forward>
```

```
        <forward name="errologin" path="/WEB-INF/jsp/index.jsp"></forward>
        <forward name="resultadoSucesso" path="/WEB-
INF/jsp/configuracoes/resultado.jsp"></forward>
        <forward name="erro" path="/WEB-
INF/jsp/configuracoes/erro.jsp"></forward>
    </action>
    <action path="/configuracoes/prazos"
type="br.com.umd.ouvidoria.web.controle.action.PrazoAction"
name="PrazoFormBean">
        <forward name="success" path="/WEB-
INF/jsp/configuracoes/prazos.jsp"></forward>
        <forward name="resultadoConsulta" path="/WEB-
INF/jsp/configuracoes/resultado.jsp"></forward>
        <forward name="errologin" path="/WEB-INF/jsp/index.jsp"></forward>
        <forward name="resultadoSucesso" path="/WEB-
INF/jsp/configuracoes/resultado.jsp"></forward>
        <forward name="erro" path="/WEB-
INF/jsp/configuracoes/erro.jsp"></forward>
    </action>

    <action path="/configuracoes/unidades"
type="br.com.umd.ouvidoria.web.controle.action.UnidadeAction"
name="UnidadeFormBean">
        <forward name="success" path="/WEB-
INF/jsp/configuracoes/unidades.jsp"></forward>
        <forward name="exibicao" path="/WEB-
INF/jsp/configuracoes/unidade.jsp"></forward>
        <forward name="edicao" path="/WEB-
INF/jsp/configuracoes/editarUnidade.jsp"></forward>
        <forward name="resultadoConsulta" path="/WEB-
INF/jsp/configuracoes/resultadoConsultaUnidade.jsp"></forward>
        <forward name="errologin" path="/WEB-INF/jsp/index.jsp"></forward>
        <forward name="resultadoSucesso" path="/WEB-
INF/jsp/configuracoes/resultado.jsp"></forward>
        <forward name="erro" path="/WEB-
INF/jsp/configuracoes/erro.jsp"></forward>
    </action>

    <action path="/configuracoes/usuarios"
type="br.com.umd.ouvidoria.web.controle.action.UsuarioAction"
```

```
name="UsuarioFormBean">
    <forward name="success" path="/WEB-
INF/jsp/configuracoes/usuarios.jsp"></forward>
    <forward name="exibicao" path="/WEB-
INF/jsp/configuracoes/usuario.jsp"></forward>
    <forward name="edicao" path="/WEB-
INF/jsp/configuracoes/editarUsuario.jsp"></forward>
    <forward name="resultadoConsulta" path="/WEB-
INF/jsp/configuracoes/resultadoConsultaUsuario.jsp"></forward>
    <forward name="errologin" path="/WEB-INF/jsp/index.jsp"></forward>
    <forward name="resultadoSucesso" path="/WEB-
INF/jsp/configuracoes/resultado.jsp"></forward>
    <forward name="erro" path="/WEB-
INF/jsp/configuracoes/erro.jsp"></forward>
    </action>

    <action path="/configuracoes/assuntos"
type="br.com.umd.ouvidoria.web.controle.action.AssuntoAction"
name="AssuntoFormBean">
    <forward name="success" path="/WEB-
INF/jsp/configuracoes/assuntos.jsp"></forward>
    <forward name="resultadoConsulta" path="/WEB-
INF/jsp/configuracoes/resultadoConsulta.jsp"></forward>
    <forward name="errologin" path="/WEB-INF/jsp/index.jsp"></forward>
    <forward name="resultadoSucesso" path="/WEB-
INF/jsp/configuracoes/resultado.jsp"></forward>
    <forward name="erro" path="/WEB-
INF/jsp/configuracoes/erro.jsp"></forward>
    </action>

    <action path="/externo/cadastro"
type="br.com.umd.ouvidoria.web.controle.action.CadastroExternoManifestacaoActio
n" name="ManifestacaoFormBean">
    <forward name="dadosOcorrencia" path="/WEB-
INF/jsp/externo/cadastroDadosOcorrencia.jsp"/>
    <forward name="dadosCadastro" path="/WEB-
INF/jsp/externo/cadastroDadosCadastro.jsp"/>
    <forward name="cadastroManifestante" path="/WEB-
INF/jsp/externo/cadastroManifestante.jsp"/>
    <forward name="manifestacaoCadastrada" path="/WEB-
INF/jsp/externo/resultadoCadastro.jsp"/>
```

```
</action>

    <action path="/cadastro/manifestacao"
type="br.com.omb.ouvidoria.web.controle.action.CadastroManifestacaoAction"
name="ManifestacaoFormBean">
        <forward name="cadastroManifestacao" path="/WEB-
INF/jsp/cadastro/manifestacao.jsp"/>
        <forward name="cadastroManifestante" path="/WEB-
INF/jsp/cadastro/manifestante.jsp"/>
        <forward name="manifestacaoCadastrada" path="/WEB-
INF/jsp/externo/resultadoCadastro.jsp"/>
    </action>

    <action path="/caixa/novas"
type="br.com.omb.ouvidoria.web.controle.action.caixa.NovasAction"
name="NovasFormBean">
        <forward name="novas" path="/WEB-INF/jsp/caixa/novas.jsp"/>
        <forward name="nova" path="/WEB-INF/jsp/caixa/nova.jsp"/>
        <forward name="erro" path="/WEB-INF/jsp/caixa/erro.jsp"/>
    </action>

    <action path="/caixa/encaminhar"
type="br.com.omb.ouvidoria.web.controle.action.caixa.EncaminharAction"
name="EncaminharFormBean">
        <forward name="encaminhar" path="/WEB-INF/jsp/caixa/encaminhar.jsp"/>
        <forward name="comentar" path="/WEB-INF/jsp/caixa/comentar.jsp"/>
        <forward name="resultado" path="/WEB-INF/jsp/caixa/resultado.jsp"/>
    </action>

    <action path="/caixa/aguardandoResposta"
type="br.com.omb.ouvidoria.web.controle.action.caixa.AguardandoRespostaAction"
name="AguardandoRespostaFormBean">
        <forward name="aguardandoResposta" path="/WEB-
INF/jsp/caixa/aguardandoResposta.jsp"/>
    </action>

    <action path="/caixa/detalhes"
type="br.com.omb.ouvidoria.web.controle.action.caixa.DetalhesAction"
name="DetalhesFormBean">
        <forward name="detalhes" path="/WEB-INF/jsp/caixa/detalhes.jsp"/>
        <forward name="erro" path="/WEB-INF/jsp/caixa/erro.jsp"/>
    </action>

    <action path="/configuracoes/cidades"
```

```

type="br.com.omb.ouvidoria.web.controle.action.CidadeAction"
name="CidadeFormBean">
    <forward name="cidades" path="/WEB-INF/jsp/configuracoes/cidades.jsp"></forward>
    <forward name="resultadoConsulta" path="/WEB-INF/jsp/configuracoes/resultadoConsultaCidades.jsp"></forward>
    <forward name="success" path="/WEB-INF/jsp/configuracoes/resultado.jsp"></forward>
    <forward name="erro" path="/WEB-INF/jsp/configuracoes/erro.jsp"></forward>
</action>
<action path="/caixa/responder"
type="br.com.omb.ouvidoria.web.controle.action.caixa.ResponderAction"
name="ResponderFormBean">
    <forward name="responder" path="/WEB-INF/jsp/caixa/responder.jsp"/>
    <forward name="resultado" path="/WEB-INF/jsp/caixa/resultado.jsp"/>
</action>
<action path="/caixa/respondidas"
type="br.com.omb.ouvidoria.web.controle.action.caixa.RespondidasAction"
name="RespondidasFormBean">
    <forward name="respondidas" path="/WEB-INF/jsp/caixa/respondidas.jsp"/>
</action>
<action path="/consulta/index"
type="br.com.omb.ouvidoria.web.controle.action.ConsultaAction"
name="ConsultaFormBean">
    <forward name="consulta" path="/WEB-INF/jsp/consulta/index.jsp"></forward>
    <forward name="resultadoConsulta" path="/WEB-INF/jsp/consulta/resultadoConsulta.jsp"></forward>
    <forward name="erro" path="/WEB-INF/jsp/configuracoes/erro.jsp"></forward>
</action>
<action path="/caixa/enviar"
type="br.com.omb.ouvidoria.web.controle.action.caixa.EnviaRespostaAction"
name="EnviarRespostaFormBean">
    <forward name="enviar" path="/WEB-INF/jsp/caixa/enviar.jsp"/>
    <forward name="resultado" path="/WEB-INF/jsp/caixa/resultado.jsp"/>
</action>
<action path="/caixa/encerradas"
type="br.com.omb.ouvidoria.web.controle.action.caixa.EncerradasAction"

```

```
name="EncerradasFormBean">
    <forward name="encerradas" path="/WEB-INF/jsp/caixa/encerradas.jsp"/>
</action>
    <action path="/relatorios/index"
type="br.com.ond.ouvidoria.web.controle.action.relatorio.RelatorioAction"
name="RelatorioFormBean">
    <forward name="success" path="/WEB-
INF/jsp/relatorios/index.jsp"></forward>
    <forward name="resultado" path="/WEB-
INF/jsp/relatorios/resultado.jsp"></forward>
</action>
    <action path="/relatorios/assuntos"
type="br.com.ond.ouvidoria.web.controle.action.relatorio.RelatorioAssuntoAction"
name="RelatorioFormBean">
    <forward name="success" path="/WEB-
INF/jsp/relatorios/assuntos.jsp"></forward>
</action>
    <action path="/relatorios/classificacao"
type="br.com.ond.ouvidoria.web.controle.action.relatorio.RelatorioClassificacaoActio
n" name="RelatorioFormBean">
    <forward name="success" path="/WEB-
INF/jsp/relatorios/classificacao.jsp"></forward>
</action>
    <action path="/relatorios/manifestacoes"
type="br.com.ond.ouvidoria.web.controle.action.relatorio.RelatorioManifestacaoActio
n" name="RelatorioFormBean">
    <forward name="success" path="/WEB-
INF/jsp/relatorios/manifestacoes.jsp"></forward>
</action>
    <action path="/relatorios/unidades"
type="br.com.ond.ouvidoria.web.controle.action.relatorio.RelatorioUnidadeAction"
name="RelatorioFormBean">
    <forward name="success" path="/WEB-
INF/jsp/relatorios/unidades.jsp"></forward>
</action>

    <action path="/configuracoes/notificacoes"
type="br.com.ond.ouvidoria.web.controle.action.NotificacaoAction" >
    <forward name="success" path="/WEB-
INF/jsp/configuracoes/resultado.jsp"></forward>
```

```

        <forward name="inicial" path="/WEB-
INF/jsp/configuracoes/notificacoes.jsp"></forward>
        <forward name="erro" path="/WEB-
INF/jsp/configuracoes/erro.jsp"></forward>
    </action>
    <action path="/caixa/arevisar"
type="br.com.ond.ouvidoria.web.controle.action.caixa.ARevisarAction"
name="ARevisarFormBean">
        <forward name="arevisar" path="/WEB-INF/jsp/caixa/arevisar.jsp"/>
    </action>

    <action path="/caixa/index"
type="br.com.ond.ouvidoria.web.controle.action.caixa.CaixaManifestacoesAction">
        <forward name="resultado" path="/WEB-INF/jsp/caixa/index.jsp"></forward>
        <forward name="erro" path="/WEB-INF/jsp/caixa/erro.jsp"></forward>
    </action>
</action-mappings>

<controller processorClass="org.apache.struts.tiles.TilesRequestProcessor"/>

<message-resources parameter="Mensagens"/>

<!-- ===== Tiles plugin
===== -->
<!--
Paths found in Tiles definitions are relative to the main context.
-->
<plug-in className="org.apache.struts.tiles.TilesPlugin" >
    <set-property property="definitions-config" value="/WEB-INF/tiles-defs.xml" />
    <set-property property="moduleAware" value="true" />
</plug-in>

<!-- ===== Validator plugin
===== -->
<plug-in className="org.apache.struts.validator.ValidatorPlugIn">
    <set-property
        property="pathnames"

```

```

        value="/WEB-INF/validator-rules.xml,/WEB-INF/validation.xml"/>
    </plug-in>

</struts-config>

validation.xml
<?xml version="1.0" encoding="ISO-8859-1" ?>

<!DOCTYPE form-validation PUBLIC
    "-//Apache Software Foundation//DTD Commons Validator Rules Configuration
    1.1.3//EN"
    "http://jakarta.apache.org/commons/dtds/validator_1_1_3.dtd">

<form-validation>

<!--
    This is a minimal Validator form file with a couple of examples.
-->

<global>

    <!-- An example global constant
    <constant>
        <constant-name>postalCode</constant-name>
        <constant-value>^\d{5}\d*$</constant-value>
    </constant>
    end example-->

</global>

<formset>

    <!-- An example form -->
    <form name="logonForm">
        <field

```

```

        property="username"
        depends="required">
            <arg key="logonForm.username"/>
    </field>
    <field
        property="password"
        depends="required,mask">
            <arg key="logonForm.password"/>
            <var>
                <var-name>mask</var-name>
                <var-value>^[0-9a-zA-Z]*$</var-value>
            </var>
    </field>
</form>

</formset>

<!-- An example formset for another locale -->
<formset language="fr">

    <constant>
        <constant-name>postalCode</constant-name>
        <constant-value>^[0-9a-zA-Z]*$</constant-value>
    </constant>

    <!-- An example form -->
    <form name="logonForm">
        <field
            property="username"
            depends="required">
                <arg key="logonForm.username"/>
        </field>
        <field
            property="password"

```

```

        depends="required,mask">
            <arg key="logonForm.password"/>
            <var>
                <var-name>mask</var-name>
                <var-value>^[0-9a-zA-Z]*$</var-value>
            </var>
        </field>
    </form>

</formset>

</form-validation>

```

### ***validator-rules.xml***

```

<!DOCTYPE form-validation PUBLIC
    "-//Apache Software Foundation//DTD Commons Validator Rules Configuration
    1.1.3//EN"
    "http://jakarta.apache.org/commons/dtds/validator_1_1_3.dtd">
<!--
    $Id: validator-rules.xml,v 1.1 2006/12/02 14:48:47 rony Exp $

```

This file contains the default Struts Validator pluggable validator definitions. It should be placed somewhere under /WEB-INF and referenced in the struts-config.xml under the plug-in element for the ValidatorPlugIn.

```

    <plug-in className="org.apache.struts.validator.ValidatorPlugIn">
        <set-property property="pathnames" value="/WEB-INF/validator-rules.xml,
            /WEB-INF/validation.xml"/>
    </plug-in>

```

These are the default error messages associated with each validator defined in this file. They should be added to your projects ApplicationResources.properties file or you can associate new ones by modifying the

pluggable validators msg attributes in this file.

# Struts Validator Error Messages

errors.required={0} is required.

errors.minlength={0} can not be less than {1} characters.

errors.maxlength={0} can not be greater than {1} characters.

errors.invalid={0} is invalid.

errors.byte={0} must be a byte.

errors.short={0} must be a short.

errors.integer={0} must be an integer.

errors.long={0} must be a long.

errors.float={0} must be a float.

errors.double={0} must be a double.

errors.date={0} is not a date.

errors.range={0} is not in the range {1} through {2}.

errors.creditcard={0} is an invalid credit card number.

errors.email={0} is an invalid e-mail address.

Note: Starting in Struts 1.2.0 the default javascript definitions have been consolidated to commons-validator. The default can be overridden by supplying a <javascript> element with a CDATA section, just as in struts 1.1.

-->

<form-validation>

<global>

<validator name="required"

    classname="org.apache.struts.validator.FieldChecks"

    method="validateRequired"

```
methodParams="java.lang.Object,  
    org.apache.commons.validator.ValidatorAction,  
    org.apache.commons.validator.Field,  
    org.apache.struts.action.ActionMessages,  
    org.apache.commons.validator.Validator,  
    javax.servlet.http.HttpServletRequest"  
msg="errors.required"/>
```

```
<validator name="requiredif"  
    classname="org.apache.struts.validator.FieldChecks"  
    method="validateRequiredIf"  
    methodParams="java.lang.Object,  
        org.apache.commons.validator.ValidatorAction,  
        org.apache.commons.validator.Field,  
        org.apache.struts.action.ActionMessages,  
        org.apache.commons.validator.Validator,  
        javax.servlet.http.HttpServletRequest"  
    msg="errors.required"/>
```

```
<validator name="validwhen"  
    msg="errors.required"  
    classname="org.apache.struts.validator.validwhen.ValidWhen"  
    method="validateValidWhen"  
    methodParams="java.lang.Object,  
        org.apache.commons.validator.ValidatorAction,  
        org.apache.commons.validator.Field,  
        org.apache.struts.action.ActionMessages,  
        org.apache.commons.validator.Validator,  
        javax.servlet.http.HttpServletRequest"/>
```

```
<validator name="minlength"  
    classname="org.apache.struts.validator.FieldChecks"  
    method="validateMinLength"
```

```
methodParams="java.lang.Object,  
    org.apache.commons.validator.ValidatorAction,  
    org.apache.commons.validator.Field,  
    org.apache.struts.action.ActionMessages,  
    org.apache.commons.validator.Validator,  
    javax.servlet.http.HttpServletRequest"  
depends=""  
msg="errors.minlength"  
jsFunction="org.apache.commons.validator.javascript.validateMinLength"/>
```

```
<validator name="maxlength"  
    classname="org.apache.struts.validator.FieldChecks"  
    method="validateMaxLength"  
methodParams="java.lang.Object,  
    org.apache.commons.validator.ValidatorAction,  
    org.apache.commons.validator.Field,  
    org.apache.struts.action.ActionMessages,  
    org.apache.commons.validator.Validator,  
    javax.servlet.http.HttpServletRequest"  
depends=""  
msg="errors.maxlength"  
jsFunction="org.apache.commons.validator.javascript.validateMaxLength"/>
```

```
<validator name="mask"  
    classname="org.apache.struts.validator.FieldChecks"  
    method="validateMask"  
methodParams="java.lang.Object,  
    org.apache.commons.validator.ValidatorAction,  
    org.apache.commons.validator.Field,  
    org.apache.struts.action.ActionMessages,  
    org.apache.commons.validator.Validator,
```

```
        javax.servlet.http.HttpServletRequest"
depends=""
msg="errors.invalid"/>
```

```
<validator name="byte"
  classname="org.apache.struts.validator.FieldChecks"
  method="validateByte"
  methodParams="java.lang.Object,
    org.apache.commons.validator.ValidatorAction,
    org.apache.commons.validator.Field,
    org.apache.struts.action.ActionMessages,
    org.apache.commons.validator.Validator,
    javax.servlet.http.HttpServletRequest"
  depends=""
  msg="errors.byte"
jsFunctionName="ByteValidations"/>
```

```
<validator name="short"
  classname="org.apache.struts.validator.FieldChecks"
  method="validateShort"
  methodParams="java.lang.Object,
    org.apache.commons.validator.ValidatorAction,
    org.apache.commons.validator.Field,
    org.apache.struts.action.ActionMessages,
    org.apache.commons.validator.Validator,
    javax.servlet.http.HttpServletRequest"
  depends=""
  msg="errors.short"
jsFunctionName="ShortValidations"/>
```

```
<validator name="integer"
```

```
        classname="org.apache.struts.validator.FieldChecks"
        method="validateInteger"
methodParams="java.lang.Object,
        org.apache.commons.validator.ValidatorAction,
        org.apache.commons.validator.Field,
        org.apache.struts.action.ActionMessages,
        org.apache.commons.validator.Validator,
        javax.servlet.http.HttpServletRequest"
        depends=""
        msg="errors.integer"
jsFunctionName="IntegerValidations"/>
```

```
<validator name="long"
        classname="org.apache.struts.validator.FieldChecks"
        method="validateLong"
methodParams="java.lang.Object,
        org.apache.commons.validator.ValidatorAction,
        org.apache.commons.validator.Field,
        org.apache.struts.action.ActionMessages,
        org.apache.commons.validator.Validator,
        javax.servlet.http.HttpServletRequest"
        depends=""
        msg="errors.long"/>
```

```
<validator name="float"
        classname="org.apache.struts.validator.FieldChecks"
        method="validateFloat"
methodParams="java.lang.Object,
        org.apache.commons.validator.ValidatorAction,
        org.apache.commons.validator.Field,
        org.apache.struts.action.ActionMessages,
```

```
        org.apache.commons.validator.Validator,  
        javax.servlet.http.HttpServletRequest"  
    depends=""  
    msg="errors.float"  
jsFunctionName="FloatValidations"/>  
  
<validator name="double"  
    classname="org.apache.struts.validator.FieldChecks"  
    method="validateDouble"  
    methodParams="java.lang.Object,  
        org.apache.commons.validator.ValidatorAction,  
        org.apache.commons.validator.Field,  
        org.apache.struts.action.ActionMessages,  
        org.apache.commons.validator.Validator,  
        javax.servlet.http.HttpServletRequest"  
    depends=""  
    msg="errors.double"/>  
  
<validator name="date"  
    classname="org.apache.struts.validator.FieldChecks"  
    method="validateDate"  
    methodParams="java.lang.Object,  
        org.apache.commons.validator.ValidatorAction,  
        org.apache.commons.validator.Field,  
        org.apache.struts.action.ActionMessages,  
        org.apache.commons.validator.Validator,  
        javax.servlet.http.HttpServletRequest"  
    depends=""  
    msg="errors.date"  
jsFunctionName="DateValidations"/>  
  
<validator name="intRange"
```

```
        classname="org.apache.struts.validator.FieldChecks"
        method="validateIntRange"
methodParams="java.lang.Object,
        org.apache.commons.validator.ValidatorAction,
        org.apache.commons.validator.Field,
        org.apache.struts.action.ActionMessages,
        org.apache.commons.validator.Validator,
        javax.servlet.http.HttpServletRequest"
depends="integer"
        msg="errors.range"/>
```

```
<validator name="floatRange"
        classname="org.apache.struts.validator.FieldChecks"
        method="validateFloatRange"
methodParams="java.lang.Object,
        org.apache.commons.validator.ValidatorAction,
        org.apache.commons.validator.Field,
        org.apache.struts.action.ActionMessages,
        org.apache.commons.validator.Validator,
        javax.servlet.http.HttpServletRequest"
depends="float"
        msg="errors.range"/>
```

```
<validator name="doubleRange"
        classname="org.apache.struts.validator.FieldChecks"
        method="validateDoubleRange"
methodParams="java.lang.Object,
        org.apache.commons.validator.ValidatorAction,
        org.apache.commons.validator.Field,
        org.apache.struts.action.ActionMessages,
        org.apache.commons.validator.Validator,
        javax.servlet.http.HttpServletRequest"
depends="double"
```

```
msg="errors.range"/>
```

```
<validator name="creditCard"  
  classname="org.apache.struts.validator.FieldChecks"  
  method="validateCreditCard"  
  methodParams="java.lang.Object,  
    org.apache.commons.validator.ValidatorAction,  
    org.apache.commons.validator.Field,  
    org.apache.struts.action.ActionMessages,  
    org.apache.commons.validator.Validator,  
    javax.servlet.http.HttpServletRequest"  
  depends=""  
  msg="errors.creditcard"/>
```

```
<validator name="email"  
  classname="org.apache.struts.validator.FieldChecks"  
  method="validateEmail"  
  methodParams="java.lang.Object,  
    org.apache.commons.validator.ValidatorAction,  
    org.apache.commons.validator.Field,  
    org.apache.struts.action.ActionMessages,  
    org.apache.commons.validator.Validator,  
    javax.servlet.http.HttpServletRequest"  
  depends=""  
  msg="errors.email"/>
```

```
<validator name="url"  
  classname="org.apache.struts.validator.FieldChecks"  
  method="validateUrl"  
  methodParams="java.lang.Object,  
    org.apache.commons.validator.ValidatorAction,  
    org.apache.commons.validator.Field,
```

```
        org.apache.struts.action.ActionMessages,  
        org.apache.commons.validator.Validator,  
        javax.servlet.http.HttpServletRequest"  
    depends=""  
    msg="errors.url"/>
```

```
<!--
```

This simply allows struts to include the validateUtilities into a page, it should not be used as a validation rule.

```
-->
```

```
<validator name="includeJavaScriptUtilities"  
    classname=""  
    method=""  
    methodParams=""  
    depends=""  
    msg=""  
    jsFunction="org.apache.commons.validator.javascript.validateUtilities"/>
```

```
</global>
```

```
</form-validation>
```

### **web.xml**

```
<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE web-app PUBLIC  
"-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"  
"http://java.sun.com/dtd/web-app_2_3.dtd">
```

```
<web-app id="WebApp_ID">
```

```
    <!-- Configuração para envio de e-mails através do sistema -->
```

```
    <!-- Servidor de E-mail TCE/AM
```

```
    <context-param>
```

```
        <param-name>servidorEmail</param-name>
```

```
        <param-value>10.0.0.4</param-value>
```

```
</context-param>
<context-param>
  <param-name>apelidoEmail</param-name>
  <param-value>Ouvidoria TCE-AM</param-value>
</context-param>
<context-param>
  <param-name>email</param-name>
  <param-value>ouvidoria@tce.am.gov.br</param-value>
</context-param>
<context-param>
  <param-name>usuarioEmail</param-name>
  <param-value>ouvidoria@tce.am.gov.br</param-value>
</context-param>
<context-param>
  <param-name>senhaEmail</param-name>
  <param-value>ouvidoriaam </param-value>
</context-param>
-->
<!--Servidor Email OMD-->
<context-param>
  <param-name>servidorEmail</param-name>
  <param-value>smtp.omb.com.br</param-value>
</context-param>
<context-param>
  <param-name>apelidoEmail</param-name>
  <param-value>OMB Soluções para Ouvidorias</param-value>
</context-param>
<context-param>
  <param-name>senhaEmail</param-name>
  <param-value>sala602</param-value>
</context-param>
<context-param>
  <param-name>email</param-name>
  <param-value>ouvidoria@omb.com.br</param-value>
```

```

</context-param>
<context-param>
  <param-name>usuarioEmail</param-name>
  <param-value>ouvidoria@omd.com.br</param-value>
</context-param>

<!-- Arquivos de configuração do Spring -->
<context-param>
  <param-name>contextConfigLocation</param-name>
  <param-value>/WEB-INF/classes/applicationContext-service.xml, /WEB-
INF/classes/applicationContext-dao.xml</param-value>
</context-param>

<!--
Criado por causa do Spring.
Este listner se encarrega de criar o objeto da classe WebApplicationContext
usando
os dados do arquivo de configuração applicationContext.xml e colocar o objeto
criado
dentro do servlet context para ser usado pelos actions.
-->
<listener>
  <listener-
class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>

<!-- Configuração do Struts -->
<servlet>
  <servlet-name>action</servlet-name>
  <servlet-class>org.apache.struts.action.ActionServlet</servlet-class>
  <init-param>
    <param-name>config</param-name>
    <param-value>/WEB-INF/struts-config.xml</param-value>
  </init-param>
  <init-param>
    <param-name>mensagens_omd</param-name>

```

```
        <param-value>Mensagens</param-value>
    </init-param>
    <init-param>
        <param-name>debug</param-name>
        <param-value>2</param-value>
    </init-param>
    <init-param>
        <param-name>detail</param-name>
        <param-value>2</param-value>
    </init-param>
    <load-on-startup>2</load-on-startup>
</servlet>
<servlet-mapping>
    <servlet-name>action</servlet-name>
    <url-pattern>*.do</url-pattern>
</servlet-mapping>
<welcome-file-list>
    <welcome-file>/WEB-INF/jsp/index.jsp</welcome-file>
</welcome-file-list>
<taglib>
    <taglib-uri>/tags/struts-bean</taglib-uri>
    <taglib-location>/WEB-INF/struts-bean.tld</taglib-location>
</taglib>

<taglib>
    <taglib-uri>/tags/struts-html</taglib-uri>
    <taglib-location>/WEB-INF/struts-html.tld</taglib-location>
</taglib>

<taglib>
    <taglib-uri>/tags/struts-logic</taglib-uri>
    <taglib-location>/WEB-INF/struts-logic.tld</taglib-location>
</taglib>
```

```
<taglib>
  <taglib-uri>/tags/struts-nested</taglib-uri>
  <taglib-location>/WEB-INF/struts-nested.tld</taglib-location>
</taglib>

<taglib>
  <taglib-uri>/tags/struts-tiles</taglib-uri>
  <taglib-location>/WEB-INF/struts-tiles.tld</taglib-location>
</taglib>
</web-app>
```

```
..\src\java\Mensagens.properties
```

```
login.usuario=Usuário
```

```
login.senha=Senha
```

```
login.usuario.null=Campo Usuário deve ser preenchido.
```

```
login.senha.null=Campo Usuário deve ser preenchido.
```

```
login.erro=Usuário e/ou Senha incorretos.
```

```
cargos.pagina.titulo=OMD v2.0 - OMD Soluções para Ouvidorias
```

```
cargos.label.titulo=Configurações de Cargos
```

```
cargos.label.titulo.cadastrados=Cargos Cadastrados:
```

```
cargos.label.novo=Novo Cargo
```

```
cargos.label.editar=Editar
```

```
cargos.label.consultar=Consulta
```

```
cargos.label.excluir=Excluir
```

```
cargos.label.alerta=Atenção: A alteração do cargo acarretará na mudança de todos os usuários que possuem este cargo.
```

```
cargos.label.substituir=Substituir por:
```

```
cargos.label.cargo=Cargo:
```

```
cargos.label.senha=Senha:
```

```
cargos.label.digite=Digite o cargo que você deseja consultar
```

```
prazos.pagina.titulo=OMD v2.0 - OMD Soluções para Ouvidorias
```

```
prazos.label.configuracoes=Configurações de Prazos
```

```
prazos.label.titulo.nivel=Nível
```

prazos.label.titulo.situacao=Situação

prazos.label.prazo=Prazo

prazos.label.editar=Editar

prazos.label.atencao=Atenção: A alteração do prazo acarretará na mudança de manifestações em andamento e de novas manifestações.

prazos.label.diasuteis=Dias Úteis:

prazos.label.sequencia=(Nível, Situação, Dias Úteis)

prazos.label.zero=Deve ser maior que Zero (0)

..\src\java\applicationContext-service.xml

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN"
```

```
'http://www.springframework.org/dtd/spring-beans.dtd'>
```

```
<beans>
```

```
  <bean id="abstractTxDefinition" abstract="true"
```

```
class="org.springframework.transaction.interceptor.TransactionProxyFactoryBean">
```

```
  <property name="transactionManager" ref="transactionManager"/>
```

```
  <property name="transactionAttributes">
```

```
    <props>
```

```
      <prop key="get*">PROPAGATION_REQUIRED,readOnly</prop>
```

```
      <prop key="pesquisar*">PROPAGATION_REQUIRED,readOnly</prop>
```

```
      <prop key="save*">PROPAGATION_REQUIRED</prop>
```

```
      <prop key="delete*">PROPAGATION_REQUIRED</prop>
```

```
      <prop key="*">PROPAGATION_REQUIRED,readOnly</prop>
```

```
    </props>
```

```
  </property>
```

```
</bean>
```

```
<!-- Cargo -->
```

```
<bean id="cargoService" parent="abstractTxDefinition">
```

```
  <property name="target">
```

```
    <bean class="br.com.ond.ouvidoria.logica.service.impl.CargoServiceImpl">
```

```
      <property name="cargoDAO" ref="cargoDAO"/>
```

```
    </bean>
```

```

    </property>
</bean>
<!-- Cargo -->

<!-- CaixaManifestacao -->
<bean id="caixaManifestacaoService" parent="abstractTxDefinition">
    <property name="target">
        <bean
class="br.com.ond.ouvidoria.logica.service.impl.CaixaManifestacaoServiceImpl">
            <property name="dao" ref="caixaManifestacaoDAO"/>
        </bean>
    </property>
</bean>
<!-- CaixaManifestacao -->

<!-- Assunto -->
<bean id="assuntoService" parent="abstractTxDefinition">
    <property name="target">
        <bean class="br.com.ond.ouvidoria.logica.service.impl.AssuntoServiceImpl">
            <property name="assuntoDAO" ref="assuntoDAO"/>
            <property name="outroAssuntoDAO" ref="outroAssuntoDAO"/>
        </bean>
    </property>
</bean>
<!-- Assunto -->

<!-- Unidade -->
<bean id="unidadeService" parent="abstractTxDefinition">
    <property name="target">
        <bean
class="br.com.ond.ouvidoria.logica.service.impl.UnidadeServiceImpl">
            <property name="unidadeDAO" ref="unidadeDAO"/>
            <property name="outraUnidadeDAO" ref="outraUnidadeDAO"/>
        </bean>
    </property>

```

```

</bean>
<!-- Unidade -->

<!-- Prazos -->
<bean id="prazoService" parent="abstractTxDefinition">
  <property name="target">
    <bean class="br.com.ond.ouvidoria.logica.service.impl.PrazoServiceImpl">
      <property name="prazoDAO" ref="prazoDAO"/>
    </bean>
  </property>
</bean>
<!-- Prazos -->

<!-- Usuario -->
<bean id="usuarioService" parent="abstractTxDefinition">
  <property name="target">
    <bean class="br.com.ond.ouvidoria.logica.service.impl.UsuarioServiceImpl">
      <property name="usuarioDAO" ref="usuarioDAO"/>
    </bean>
  </property>
</bean>
<!-- Usuario -->

<!-- Relatorio -->
<bean id="relatorioService" parent="abstractTxDefinition">
  <property name="target">
    <bean
class="br.com.ond.ouvidoria.logica.service.impl.RelatorioServiceImpl">
      <property name="relatorioDAO" ref="relatorioDAO"/>
    </bean>
  </property>
</bean>
<!-- Relatorio -->
<!-- Manifestacao -->
<bean id="manifestacaoService" parent="abstractTxDefinition">

```

```
<property name="target">
  <bean
class="br.com.ond.ouvidoria.logica.service.impl.ManifestacaoServiceImpl">
    <property name="dao" ref="manifestacaoDAO"/>
  </bean>
</property>
</bean>
<!-- Manifestacao -->

<!-- Feriado -->
<bean id="feriadoService" parent="abstractTxDefinition">
  <property name="target">
    <bean class="br.com.ond.ouvidoria.logica.service.impl.FeriadoServiceImpl">
      <property name="feriadoDAO" ref="feriadoDAO"/>
    </bean>
  </property>
</bean>
<!-- Feriado -->

<!-- Perfil -->
<bean id="perfilService" parent="abstractTxDefinition">
  <property name="target">
    <bean class="br.com.ond.ouvidoria.logica.service.impl.PerfilServiceImpl">
      <property name="perfilDAO" ref="perfilDAO"/>
    </bean>
  </property>
</bean>
<!-- Perfil -->

<!-- Cidade -->
<bean id="cidadeService" parent="abstractTxDefinition">
  <property name="target">
    <bean class="br.com.ond.ouvidoria.logica.service.impl.CidadeServiceImpl">
      <property name="dao" ref="cidadeDAO"/>
      <property name="outraCidadeDAO" ref="outraCidadeDAO"/>
    </bean>
  </property>
</bean>
```

```

        </bean>
    </property>
</bean>
<!-- Cidade -->

<!-- Estado -->
<bean id="estadoService" parent="abstractTxDefinition">
    <property name="target">
        <bean class="br.com.omd.ouvidoria.logica.service.impl.EstadoServiceImpl">
            <property name="estadoDAO" ref="estadoDAO"/>
        </bean>
    </property>
</bean>
<!-- Estado -->

<!-- Notificacao -->
<bean id="notificacaoService" parent="abstractTxDefinition">
    <property name="target">
        <bean
class="br.com.omd.ouvidoria.logica.service.impl.NotificacaoServiceImpl">
            <property name="dao" ref="notificacaoDAO"/>
        </bean>
    </property>
</bean>
<!-- Notificacao -->
</beans>

..\src\java\br\com\omd\ouvidoria\logica\domain\BaseDomain.java
/*
 * BaseDomain.java
 *
 */
package br.com.omd.ouvidoria.logica.domain;

public class BaseDomain {

```

```

protected Long id;

/** Creates a new instance of Identitidade */
public BaseDomain() {
}

public Long getId() {
    return id;
}

public void setId(Long id) {
    this.id = id;
}
}

..\src\java\br\com\omd\ouvidoria\logica\service\AssuntoService.java
/*
 * AssuntoService.java
 *
 */
package br.com.omd.ouvidoria.logica.service;

import br.com.omd.ouvidoria.logica.domain.Assunto;
import br.com.omd.ouvidoria.logica.domain.outros.OutroAssunto;
import br.com.omd.ouvidoria.logica.dto.AssuntoDTO;
import java.util.Collection;

public interface AssuntoService extends Service {

    public void save(Assunto assunto ) throws Exception;

    public void delete(Assunto assunto ) throws Exception;

```

```
public Collection select(Assunto assunto ) throws Exception;

public Collection getAll() throws Exception;

public void edit(Assunto assunto_Old,Assunto assunto_New) throws Exception;

public Collection getQuantidadePorOutro() throws Exception;

    public void changeAssuntoPorOutroAssunto(OutroAssunto outroAssunto,Assunto
assuntoPadrao) throws Exception;
}
```

```
..\src\java\br\com\omd\ouvidoria\logica\service\CaixaManifestacaoService.java
```

```
/*
```

```
* CaixaManifestacaoService.java
```

```
*
```

```
*/
```

```
package br.com.omd.ouvidoria.logica.service;
```

```
import br.com.omd.ouvidoria.logica.domain.Manifestacao;
```

```
import br.com.omd.ouvidoria.logica.domain.Resposta;
```

```
import br.com.omd.ouvidoria.logica.domain.Unidade;
```

```
import br.com.omd.ouvidoria.logica.domain.Usuario;
```

```
import br.com.omd.ouvidoria.util.Email;
```

```
import java.io.File;
```

```
import java.util.HashMap;
```

```
import java.util.LinkedHashMap;
```

```
import java.util.List;
```

```
public interface CaixaManifestacaoService extends Service {
```

```
    public List getNovas(Unidade unidade) throws Exception;
```

```
    public Manifestacao getNova(Long codigoManifestacao) throws Exception;
```

```
    public HashMap getDadosNova(Long idUnidade) throws Exception;
```

```
    public void atualizarNova(Manifestacao manifestacao) throws Exception;
```

```

    public void encaminhar(Manifestacao manifestacao, File
arquivoNotificacaoEncaminhamento, Email email) throws Exception;
    public List getAguardandoResposta(Unidade unidadeParametro, Unidade
unidadeUsuario) throws Exception;
    public Manifestacao getDetalhes(Long codigoManifestacao) throws Exception;
    public List getUnidadesNivelAbaixoSemOutraSemNenhuma(Long idUnidade)
throws Exception;
    public HashMap getDadosResponder(Long idUnidade) throws Exception;
    public void cadastrarReposta(Manifestacao manifestacao, Resposta resposta)
throws Exception;
    public Unidade getUnidade(Long idUnidade) throws Exception;
    public List getRespondidas(Usuario usuario) throws Exception;
    public List getARevisar(Unidade unidadeParametro, Unidade unidadeUsuario)
throws Exception;
    public Manifestacao enviarResposta(Usuario usuario, Long codigoManifestacao,
Integer numeroRespostaSelecionada) throws Exception;
    public List getEncerradas(Unidade unidadeParametro, Usuario usuario) throws
Exception;
    public List contaManifestacoesPorSituacao(Usuario usuario) throws Exception;
    public void setNova(Long idManifestacao, boolean nova) throws Exception;
}

```

```

..\src\java\br\com\omd\ouvidoria\logica\service\CargoService.java

```

```

/*

```

```

 * CargoService.java

```

```

 *

```

```

 */

```

```

package br.com.omd.ouvidoria.logica.service;

```

```

import br.com.omd.ouvidoria.logica.dto.CargoDTO;

```

```

import java.util.Collection;

```

```

public interface CargoService extends Service {

```

```

    public void save(CargoDTO cargoDTO) throws Exception;

```

```

    public void delete(CargoDTO cargoDTO) throws Exception;

```

```
public Collection select(CargoDTO cargoDTO) throws Exception;

public Collection getAll() throws Exception;

public void edit(CargoDTO cargoDTO_Old,CargoDTO cargoDTO_New) throws
Exception;
}
```

```
..\src\java\br\com\omd\ouvidoria\logica\service\CidadeService.java
```

```
/*
```

```
* CidadeService.java
```

```
*
```

```
*/
```

```
package br.com.omd.ouvidoria.logica.service;
```

```
import br.com.omd.ouvidoria.logica.domain.Cidade;
```

```
import br.com.omd.ouvidoria.logica.domain.Estado;
```

```
import br.com.omd.ouvidoria.logica.domain.outros.OutraCidade;
```

```
import java.util.List;
```

```
public interface CidadeService extends Service {
```

```
    public List pesquisarSemOutra(Estado estado) throws Exception;
```

```
    public List getQuantidadePorOutroCidadeEstado() throws Exception;
```

```
    public List getEstados() throws Exception;
```

```
    public List getEstadosComCidades() throws Exception;
```

```
    public List pesquisarPorEstado(Long idEstado) throws Exception;
```

```
    public Cidade save(Cidade cidade) throws Exception;
```

```
public List pesquisarCidades(Cidade cidade) throws Exception;

public void delete(Cidade cidade) throws Exception;

public void substituirOutraCidadePorCidade(OutraCidade outraCidade, Cidade
cidade) throws Exception;
}
```

```
..\src\java\br\com\omd\ouvidoria\logica\service\EstadoService.java
```

```
/*
```

```
* EstadoService.java
```

```
*
```

```
*/
```

```
package br.com.omd.ouvidoria.logica.service;
```

```
import java.util.Collection;
```

```
public interface EstadoService extends Service {
```

```
    public Collection getAll() throws Exception;
```

```
}
```

```
..\src\java\br\com\omd\ouvidoria\logica\service\FeriadoService.java
```

```
/*
```

```
* FeriadoService.java
```

```
*
```

```
*/
```

```
package br.com.omd.ouvidoria.logica.service;
```

```
import br.com.omd.ouvidoria.logica.domain.Feriado;
```

```
import br.com.omd.ouvidoria.logica.dto.FeriadoDTO;
```

```
import java.util.Collection;
```

```
public interface FeriadoService extends Service {

    public void save(Feriado feriado ) throws Exception;

    public void delete(Feriado feriado ) throws Exception;

    public Collection select(Feriado feriadoInicio, Feriado feriadoFim) throws
Exception;

    public Collection getAll() throws Exception;

    public void edit(Feriado feriado_Old,Feriado feriado_New) throws Exception;
}
```

```
..\src\java\br\com\omd\ouvidoria\logica\service\ManifestacaoService.java
```

```
/*
```

```
* ManifestacaoService.java
```

```
*
```

```
*/
```

```
package br.com.omd.ouvidoria.logica.service;
```

```
import br.com.omd.ouvidoria.logica.domain.Estado;
```

```
import br.com.omd.ouvidoria.logica.domain.Manifestacao;
```

```
import br.com.omd.ouvidoria.logica.dto.ConsultaManifestacaoDTO;
```

```
import java.util.Date;
```

```
import java.util.HashMap;
```

```
import java.util.List;
```

```
public interface ManifestacaoService extends Service {
```

```
    public List getDadosOcorrencia() throws Exception;
```

```
    public HashMap getDadosCadastro() throws Exception;
```

```
    public HashMap getDadosCadastroManifestacao() throws Exception;
```

```
    public List getEstados() throws Exception;
```

```
    public List getCidadesPorEstado(Estado estado) throws Exception;
```

```
    public Manifestacao salvarInternaAnonima(Manifestacao manifestacao) throws
Exception;
    public Manifestacao salvarInternaManifestante(Manifestacao manifestacao)
throws Exception;
    public Manifestacao salvarExternaAnonima(Manifestacao manifestacao) throws
Exception;
    public Manifestacao salvarExternaManifestante(Manifestacao manifestacao)
throws Exception;
    public HashMap getDadosConsulta() throws Exception;
    public List consultar(ConsultaManifestacaoDTO dto) throws Exception;
    public Date getMenorDataRecebimento() throws Exception;
}
```

```
..\src\java\br\com\omd\ouvidoria\logica\service\NotificacaoService.java
```

```
/*
```

```
* NotificacaoService.java
```

```
*
```

```
*/
```

```
package br.com.omd.ouvidoria.logica.service;
```

```
import br.com.omd.ouvidoria.logica.domain.Notificacao;
```

```
import java.util.Collection;
```

```
import java.util.List;
```

```
public interface NotificacaoService extends Service {
```

```
    public Collection getAll(String ordenacao) throws Exception;
```

```
    public void atualizaHabilitado(Notificacao notificacao) throws Exception;
```

```
}
```

```
..\src\java\br\com\omd\ouvidoria\logica\service\OutraUnidadeService.java
```

```
/*
```

```
* OutraUnidadeService.java
```

```
*
```

```
*/  
package br.com.omd.ouvidoria.logica.service;  
  
import br.com.omd.ouvidoria.logica.domain.Unidade;  
import br.com.omd.ouvidoria.logica.domain.outros.OutraUnidade;  
import java.util.Collection;  
  
public interface OutraUnidadeService extends Service {  
  
    public Collection getAll() throws Exception;  
  
    public void edit(OutraUnidade outraUnidade, Unidade unidadePadrao) throws  
Exception;  
}
```

```
..\src\java\br\com\omd\ouvidoria\logica\service\OutroAssuntoService.java
```

```
/*
```

```
* OutroAssuntoService.java
```

```
*
```

```
*/
```

```
package br.com.omd.ouvidoria.logica.service;
```

```
import br.com.omd.ouvidoria.logica.domain.Assunto;  
import br.com.omd.ouvidoria.logica.domain.outros.OutroAssunto;  
import java.util.Collection;
```

```
public interface OutroAssuntoService extends Service {
```

```
    public Collection getAll() throws Exception;
```

```
    public void edit(OutroAssunto outroAssunto, Assunto assuntoPadrao) throws  
Exception;
```

```
}
```

```
..\src\java\br\com\omd\ouvidoria\logica\service\PerfilService.java
```

```
/*
 * PerfilService.java
 *
 */
package br.com.ond.ouvidoria.logica.service;

import br.com.ond.ouvidoria.logica.domain.Assunto;
import br.com.ond.ouvidoria.logica.domain.outros.OutroAssunto;
import java.util.Collection;

public interface PerfilService extends Service {

    public Collection getAll() throws Exception;
}

..\src\java\br\com\ond\ouvidoria\logica\service\PrazoService.java
/*
 * PrazoService.java
 *
 */
package br.com.ond.ouvidoria.logica.service;

import br.com.ond.ouvidoria.logica.dto.PrazoDTO;
import java.util.Collection;

public interface PrazoService extends Service {

    public void save(PrazoDTO prazoDTO) throws Exception;

    public void delete(PrazoDTO prazoDTO) throws Exception;

    public Collection select(PrazoDTO prazoDTO) throws Exception;

    public Collection getAll() throws Exception;
```

```
    public void edit(PrazoDTO prazoDTO) throws Exception;
}
```

```
..\src\java\br\com\omd\ouvidoria\logica\service\RecebimentoService.java
```

```
/*
```

```
 * RecebimentoService.java
```

```
 *
```

```
*/
```

```
package br.com.omd.ouvidoria.logica.service;
```

```
import java.util.Collection;
```

```
public interface RecebimentoService extends Service{
```

```
    public Collection getAll() throws Exception;
```

```
}
```

```
..\src\java\br\com\omd\ouvidoria\logica\service\RelatorioService.java
```

```
/*
```

```
 * RelatorioService.java
```

```
 *
```

```
*/
```

```
package br.com.omd.ouvidoria.logica.service;
```

```
import br.com.omd.ouvidoria.logica.domain.Assunto;
```

```
import br.com.omd.ouvidoria.logica.domain.Classificacao;
```

```
import br.com.omd.ouvidoria.logica.domain.Unidade;
```

```
import br.com.omd.ouvidoria.logica.domain.Usuario;
```

```
import br.com.omd.ouvidoria.util.Data;
```

```
import java.util.Collection;
```

```
import java.util.LinkedHashMap;
```

```
public interface RelatorioService extends Service{
```

```
    public Collection getAll() throws Exception;
```

```
    public LinkedHashMap getAssuntos(Unidade unidade,Data dataInicial, Data
dataFinal) throws Exception;

    public Collection getManifestacoes(Classificacao classificacao,Unidade
unidade,Assunto assunto,Data dataInicial, Data dataFinal) throws Exception;

    public Collection getClassificacao(Unidade unidade,Data dataInicial, Data
dataFinal) throws Exception ;

    public Collection getUnidades(Unidade unidade,Data dataInicial, Data dataFinal)
throws Exception ;

    public Collection getUnidadesVinculadasPorUnidade(Unidade unit) throws
Exception;

    public Collection getUnidadesPorUsuario(Usuario user) throws Exception;
}
```

```
..\src\java\br\com\omd\ouvidoria\logica\service\Service.java
```

```
/*
```

```
* Service.java
```

```
*
```

```
*/
```

```
package br.com.omd.ouvidoria.logica.service;
```

```
import br.com.omd.ouvidoria.logica.domain.BaseDomain;
```

```
import java.util.Collection;
```

```
public interface Service {
```

```
    static String ORDENAR_POR_NOME = "nome ASC";
```

```
    static String ASCENDENTE="ascending";
```

```
    static String ORDENAR_POR_DESCRICA0 = "descricao ASC";
```

```
    public Collection getAll() throws Exception;
```

```
}
```

```
..\src\java\br\com\omd\ouvidoria\logica\service\UnidadeService.java
```

```
/*
```

```
* UnidadeService.java
```

```

*
*/
package br.com.omd.ouvidoria.logica.service;

import br.com.omd.ouvidoria.logica.domain.BaseDomain;
import br.com.omd.ouvidoria.logica.domain.Unidade;
import br.com.omd.ouvidoria.logica.domain.outros.OutraUnidade;
import br.com.omd.ouvidoria.logica.dto.UnidadeDTO;
import java.util.Collection;

public interface UnidadeService extends Service {

    public void save(Unidade Unidade ) throws Exception;

    public void delete(Unidade Unidade ) throws Exception;

    public Collection select(Unidade Unidade ) throws Exception;

    public BaseDomain selectPorId(Unidade unidade) throws Exception;

    public Collection getAll() throws Exception;

    public void edit(Unidade unidade_Old,Unidade unidade_New) throws Exception;

    public Collection getQuantidadePorOutro() throws Exception;

    void changeUnidadePorOutraUnidade(OutraUnidade outraUnidade, Unidade
unidadePadrao)throws Exception;

    public Collection getAllSemOutraSemNenhuma()throws Exception;
}

..\src\java\br\com\omd\ouvidoria\logica\service\UsuarioService.java
/*
* UsuarioService.java

```

```

*
*/
package br.com.omd.ouvidoria.logica.service;

import br.com.omd.ouvidoria.logica.domain.BaseDomain;
import br.com.omd.ouvidoria.logica.domain.Usuario;
import br.com.omd.ouvidoria.logica.dto.UsuarioDTO;
import java.util.Collection;

public interface UsuarioService extends Service {

    public Usuario autenticar(UsuarioDTO usuario) throws Exception;

    public void save(Usuario Usuario ) throws Exception;

    public void delete(Usuario Usuario ) throws Exception;

    public Collection select(Usuario Usuario ) throws Exception;

    public BaseDomain selectPorId(Usuario usuario ) throws Exception;

    public Collection getAll() throws Exception;

    public void edit(Usuario usuario_Old,Usuario usuario_New) throws Exception;
}

..\src\java\br\com\omd\ouvidoria\logica\service\impl\AssuntoServiceImpl.java
/*
 * AssuntoServiceImpl.java
 *
 */
package br.com.omd.ouvidoria.logica.service.impl;

import br.com.omd.ouvidoria.logica.domain.BaseDomain;

```

```

import br.com.omd.ouvidoria.logica.domain.Assunto;
import br.com.omd.ouvidoria.logica.domain.outros.OutroAssunto;
import br.com.omd.ouvidoria.logica.service.AssuntoService;
import br.com.omd.ouvidoria.persistencia.geral.dao.AssuntoDAO;
import br.com.omd.ouvidoria.persistencia.geral.outros.dao.OutroAssuntoDAO;
import java.util.Collection;

public class AssuntoServiceImpl implements AssuntoService{

    private AssuntoDAO assuntoDAO;

    private OutroAssuntoDAO outroAssuntoDAO;

    /** Creates a new instance of AssuntoServiceImpl */
    public AssuntoServiceImpl() {

    }

    public void save(Assunto assunto ) throws Exception {
        assuntoDAO.save(assunto);
    }

    public void edit(Assunto assunto_Old,Assunto assunto_New) throws Exception {

        Assunto assuntoPronto=new Assunto();
        Assunto assuntoParaAlterar=(Assunto)selectPorId(assunto_Old);

        if (assuntoParaAlterar!=null){
            assuntoParaAlterar.setNome(assunto_New.getNome());

            if (assuntoParaAlterar.getId().equals(assunto_Old.getId())){

                assuntoDAO.edit(assuntoParaAlterar);
            }
        }
    }
}

```

```

    }
}
public void delete(Assunto assunto ) throws Exception {

    assuntoDAO.delete(assunto);
}

public Collection select(Assunto assunto ) throws Exception {

    return assuntoDAO.pesquisarPorNome(assunto.getNome(),
AssuntoService.ORDENAR_POR_NOME);
}

public BaseDomain selectPorId(Assunto assunto ) throws Exception {
    return assuntoDAO.getDomain(assunto.getId());
}

public Collection getAll() throws Exception {
    return assuntoDAO.getAll( AssuntoService.ORDENAR_POR_NOME);
}

public AssuntoDAO getAssuntoDAO() {
    return assuntoDAO;
}

public void setAssuntoDAO(AssuntoDAO assuntoDAO) {
    this.assuntoDAO = assuntoDAO;
}

public Collection getQuantidadePorOutro() throws Exception {
    return outroAssuntoDAO.getQuantidadePorOutro();
}

public OutroAssuntoDAO getOutroAssuntoDAO() {
    return outroAssuntoDAO;
}
}

```

```

    public void setOutroAssuntoDAO(OutroAssuntoDAO outroAssuntoDAO) {
        this.outroAssuntoDAO = outroAssuntoDAO;
    }

    public void changeAssuntoPorOutroAssunto(OutroAssunto outroAssunto, Assunto
assuntoPadrao) throws Exception {

        outroAssuntoDAO.substituirOutrosPorDomain(outroAssunto.getNome(), assuntoPadr
ao);
    }
}

```

```

..\src\java\br\com\omd\ouvidoria\logica\service\impl\CaixaManifestacaoServiceImpl.java

```

```

/*

```

```

 * CaixaManifestacaoServiceImpl.java

```

```

 *

```

```

*/

```

```

package br.com.omd.ouvidoria.logica.service.impl;

```

```

import br.com.omd.ouvidoria.logica.domain.Manifestacao;

```

```

import br.com.omd.ouvidoria.logica.domain.Resposta;

```

```

import br.com.omd.ouvidoria.logica.domain.Unidade;

```

```

import br.com.omd.ouvidoria.logica.domain.Usuario;

```

```

import br.com.omd.ouvidoria.logica.service.CaixaManifestacaoService;

```

```

import br.com.omd.ouvidoria.persistencia.geral.dao.CaixaManifestacaoDAO;

```

```

import br.com.omd.ouvidoria.persistencia.geral.dao.DAO;

```

```

import br.com.omd.ouvidoria.util.Email;

```

```

import java.io.File;

```

```

import java.util.HashMap;

```

```

import java.util.LinkedHashMap;

```

```

import java.util.List;

```

```

public class CaixaManifestacaoServiceImpl extends ServiceImpl implements
CaixaManifestacaoService{

```

```
private DAO dao;

public CaixaManifestacaoServiceImpl() {
}

public DAO getDao() {
    return dao;
}

public void setDao(DAO dao) {
    this.dao = dao;
}

public List getNovas(Unidade unidade) throws Exception {
    return ((CaixaManifestacaoDAO)dao).getNovas(unidade);
}

public Manifestacao getNova(Long codigoManifestacao) throws Exception {
    return ((CaixaManifestacaoDAO)dao).getNova(codigoManifestacao);
}

public HashMap getDadosNova(Long idUnidade) throws Exception{
    return ((CaixaManifestacaoDAO)dao).getDadosNova(idUnidade);
}

public void atualizarNova(Manifestacao manifestacao) throws Exception {
    ((CaixaManifestacaoDAO)dao).atualizarNova(manifestacao);
}

public void encaminhar(Manifestacao manifestacao, File
arquivoNotificacaoEncaminhamento, Email email) throws Exception {
    ((CaixaManifestacaoDAO)dao).encaminhar(manifestacao,
arquivoNotificacaoEncaminhamento, email);
}
```

```

    public List getAguardandoResposta(Unidade unidadeParametro, Unidade
unidadeUsuario) throws Exception {
        return
((CaixaManifestacaoDAO)dao).getAguardandoResposta(unidadeParametro,
unidadeUsuario);
    }

    public Manifestacao getDetalhes(Long codigoManifestacao) throws Exception {
        return ((CaixaManifestacaoDAO)dao).getDetalhes(codigoManifestacao);
    }

    public List getUnidadesNivelAbaixoSemOutraSemNenhuma(Long idUnidade)
throws Exception {
        return
((CaixaManifestacaoDAO)dao).getUnidadesNivelAbaixoSemOutraSemNenhuma(idU
nidade);
    }

    public HashMap getDadosResponder(Long idUnidade) throws Exception {
        return ((CaixaManifestacaoDAO)dao).getDadosResponder(idUnidade);
    }

    public void cadastrarReposta(Manifestacao manifestacao, Resposta resposta)
throws Exception {
        ((CaixaManifestacaoDAO)dao).cadastrarReposta(manifestacao, resposta);
    }

    public Unidade getUnidade(Long idUnidade) throws Exception {
        return ((CaixaManifestacaoDAO)dao).getUnidade(idUnidade);
    }

    public List getRespondidas(Usuario usuario) throws Exception {
        return ((CaixaManifestacaoDAO)dao).getRespondidas(usuario);
    }

    public Manifestacao enviarResposta(Usuario usuario, Long codigoManifestacao,
Integer numeroRespostaSelecionada) throws Exception {

```

```

        return ((CaixaManifestacaoDAO)dao).enviarResposta(usuario,
codigoManifestacao, numeroRespostaSelecionada);
    }

    public List getEncerradas(Unidade unidadeParametro,Usuario usuario) throws
Exception {
        return ((CaixaManifestacaoDAO)dao).getEncerradas(unidadeParametro,
usuario);
    }

    public List getARevisar(Unidade unidadeParametro, Unidade unidadeUsuario)
throws Exception {
        return ((CaixaManifestacaoDAO)dao).getARevisar(unidadeParametro,
unidadeUsuario);
    }

    public List contaManifestacoesPorSituacao(Usuario usuario) throws Exception {
        return ((CaixaManifestacaoDAO)dao).contaManifestacoesPorSituacao(usuario);
    }

    public void setNova(Long idManifestacao, boolean nova) throws Exception{
        ((CaixaManifestacaoDAO)dao).setNova(idManifestacao, nova);
    }
}

..src\java\br\com\omd\ouvidoria\logica\service\impl\CargoServiceImpl.java
/*
 * CargoServiceImpl.java
 *
 */
package br.com.omd.ouvidoria.logica.service.impl;

import br.com.omd.ouvidoria.logica.domain.BaseDomain;
import br.com.omd.ouvidoria.logica.domain.Cargo;
import br.com.omd.ouvidoria.logica.dto.CargoDTO;
import br.com.omd.ouvidoria.logica.service.CargoService;

```

```

import br.com.ond.ouvidoria.persistencia.geral.dao.CargoDAO;
import java.util.ArrayList;
import java.util.Collection;
import org.apache.commons.beanutils.BeanUtils;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class CargoServiceImpl implements CargoService{

    private CargoDAO cargoDAO;

    /** Creates a new instance of CargoServiceImpl */
    public CargoServiceImpl() {

    }

    public void save(CargoDTO cargoDTO) throws Exception {
        Cargo cargo = new Cargo();
        BeanUtils.copyProperties(cargo, cargoDTO);
        cargoDAO.save(cargo);
    }

    public void edit(CargoDTO cargoDTO_Old,CargoDTO cargoDTO_New) throws
Exception {
        Cargo cargoPronto=new Cargo();

        Cargo cargoParaAlterar=(Cargo)selectPorId(cargoDTO_Old);

        if (cargoParaAlterar!=null){
            cargoParaAlterar.setNome(cargoDTO_New.getNome());

            if (cargoParaAlterar.getId().equals(cargoDTO_Old.getId())){

                cargoDAO.edit(cargoParaAlterar);
            }
        }
    }
}

```

```

        }
    }
}
public void delete(CargoDTO cargoDTO) throws Exception {
    Cargo cargo = new Cargo();
    BeanUtils.copyProperties(cargo, cargoDTO);
    cargoDAO.delete(cargo);
}

public Collection select(CargoDTO cargoDTO) throws Exception {
    return cargoDAO.pesquisarPorNome(cargoDTO.getNome(),
CargoService.ORDENAR_POR_NOME);
}

public BaseDomain selectPorId(CargoDTO cargoDTO) throws Exception {
    return cargoDAO.getDomain(cargoDTO.getId());
}

public Collection getAll() throws Exception {
    return cargoDAO.getAll( CargoService.ORDENAR_POR_NOME);
}

public CargoDAO getCargoDAO() {
    return cargoDAO;
}

public void setCargoDAO(CargoDAO cargoDAO) {
    this.cargoDAO = cargoDAO;
}
}

```

..\src\java\br\com\omd\ouvidoria\logica\service\impl\CidadeServiceImpl.java

/\*

\* CidadeServiceImpl.java

\*

```
*/
package br.com.omd.ouvidoria.logica.service.impl;

import br.com.omd.ouvidoria.logica.domain.Cidade;
import br.com.omd.ouvidoria.logica.domain.Estado;
import br.com.omd.ouvidoria.logica.domain.outros.OutraCidade;
import br.com.omd.ouvidoria.logica.service.CidadeService;
import br.com.omd.ouvidoria.logica.service.Service;
import br.com.omd.ouvidoria.persistencia.geral.dao.CidadeDAO;
import br.com.omd.ouvidoria.persistencia.geral.outros.dao.OutraCidadeDAO;
import java.util.Collection;
import java.util.List;

public class CidadeServiceImpl extends ServiceImpl implements CidadeService {

    private OutraCidadeDAO outraCidadeDAO;

    /** Creates a new instance of CargoServiceImpl */
    public CidadeServiceImpl() {

    }

    public Collection getAll() throws Exception {
        return ((CidadeDAO) dao).getAll(Service.ORDENAR_POR_NOME);
    }

    public List pesquisarSemOutra(Estado estado) throws Exception {
        return ((CidadeDAO) dao).pesquisarSemOutra(estado);
    }

    public List getEstados() throws Exception {
        return ((CidadeDAO) dao).getEstados();
    }
}
```

```
public List getEstadosComCidades() throws Exception {
    return ((CidadeDAO) dao).getEstadosComCidades();
}

public List getQuantidadePorOutroCidadeEstado() throws Exception {
    return getOutraCidadeDAO().getQuantidadePorOutroCidadeEstado();
}

public List pesquisarPorEstado(Long idEstado) throws Exception {
    return ((CidadeDAO) dao).pesquisarPorEstado(idEstado);
}

public Cidade save(Cidade cidade) throws Exception {
    return (Cidade) dao.save(cidade);
}

public List pesquisarCidades(Cidade cidade) throws Exception {
    return (List) ((CidadeDAO) dao).pesquisarCidades(cidade);
}

public void delete(Cidade cidade) throws Exception {
    ((CidadeDAO) dao).delete(cidade);
}

public OutraCidadeDAO getOutraCidadeDAO() {
    return outraCidadeDAO;
}

public void setOutraCidadeDAO(OutraCidadeDAO outraCidadeDAO) {
    this.outraCidadeDAO = outraCidadeDAO;
}

public void substituirOutraCidadePorCidade(OutraCidade outraCidade, Cidade
cidade) throws Exception {
    outraCidadeDAO.substituirOutraCidadePorCidade(outraCidade, cidade);
}
```

```
    }  
}  
  
..\src\java\br\com\omd\ouvidoria\logica\service\impl\EstadoServiceImpl.java  
/*  
 * EstadoServiceImpl.java  
 *  
 */  
package br.com.omd.ouvidoria.logica.service.impl;  
  
import br.com.omd.ouvidoria.logica.service.EstadoService;  
import br.com.omd.ouvidoria.persistencia.geral.dao.EstadoDAO;  
import java.util.Collection;  
  
public class EstadoServiceImpl implements EstadoService{  
  
    private EstadoDAO estadoDAO;  
  
    /** Creates a new instance of EstadoServiceImpl */  
    public EstadoServiceImpl() {  
  
    }  
  
    public Collection getAll() throws Exception {  
        return estadoDAO.getAll(null);  
    }  
  
    public EstadoDAO getEstadoDAO() {  
        return estadoDAO;  
    }  
  
    public void setEstadoDAO(EstadoDAO estadoDAO) {  
        this.estadoDAO = estadoDAO;  
    }  
}
```

```
}
```

```
..\src\java\br\com\omd\ouvidoria\logica\service\impl\FeriadoServiceImpl.java
```

```
/*
```

```
 * FeriadoServiceImpl.java
```

```
 *
```

```
*/
```

```
package br.com.omd.ouvidoria.logica.service.impl;
```

```
import br.com.omd.ouvidoria.logica.domain.BaseDomain;
```

```
import br.com.omd.ouvidoria.logica.domain.Feriado;
```

```
import br.com.omd.ouvidoria.logica.dto.FeriadoDTO;
```

```
import br.com.omd.ouvidoria.logica.service.FeriadoService;
```

```
import br.com.omd.ouvidoria.persistencia.geral.dao.FeriadoDAO;
```

```
import java.util.ArrayList;
```

```
import java.util.Collection;
```

```
import org.apache.commons.beanutils.BeanUtils;
```

```
import org.springframework.context.ApplicationContext;
```

```
import org.springframework.context.support.ClassPathXmlApplicationContext;
```

```
public class FeriadoServiceImpl implements FeriadoService{
```

```
    private FeriadoDAO feriadoDAO;
```

```
    /** Creates a new instance of FeriadoServiceImpl */
```

```
    public FeriadoServiceImpl() {
```

```
    }
```

```
    public void save(Feriado feriado ) throws Exception {
```

```
        feriadoDAO.save(feriado);
```

```
    }
```

```
    public void edit(Feriado feriado_Old,Feriado feriado_New) throws Exception {
```

```

Feriado feriadoPronto=new Feriado();
Feriado feriadoParaAlterar=(Feriado)selectPorId(feriado_Old);

if (feriadoParaAlterar!=null){
    feriadoParaAlterar.setData(feriado_New.getData());

    if (feriadoParaAlterar.getId().equals(feriado_Old.getId())){
        feriadoDAO.edit(feriadoParaAlterar);
    }
}

public void delete(Feriado feriado ) throws Exception {

    feriadoDAO.delete(feriado);
}

public Collection select(Feriado feriadoInicio, Feriado feriadoFim ) throws
Exception {
    return
feriadoDAO.pesquisarPorPeriodo(feriadoInicio.getData(),feriadoFim.getData(), "data
ASC");
}

public BaseDomain selectPorId(Feriado feriado ) throws Exception {
    return feriadoDAO.getDomain(feriado.getId());
}

public Collection getAll() throws Exception {
    return feriadoDAO.getAllSemException("data ASC");
}

public FeriadoDAO getFeriadoDAO() {
    return feriadoDAO;
}

```

```
    public void setFeriadoDAO(FeriadoDAO feriadoDAO) {  
        this.feriadoDAO = feriadoDAO;  
    }  
}
```

```
..\src\java\br\com\omd\ouvidoria\logica\service\impl\ManifestacaoServiceImpl.java  
/*
```

```
* ManifestacaoServiceImpl.java
```

```
*
```

```
*/
```

```
package br.com.omd.ouvidoria.logica.service.impl;
```

```
import br.com.omd.ouvidoria.logica.domain.Estado;  
import br.com.omd.ouvidoria.logica.domain.Manifestacao;  
import br.com.omd.ouvidoria.logica.dto.ConsultaManifestacaoDTO;  
import br.com.omd.ouvidoria.logica.service.ManifestacaoService;  
import br.com.omd.ouvidoria.persistencia.geral.dao.ManifestacaoDAO;  
import java.util.Date;  
import java.util.HashMap;  
import java.util.List;
```

```
public class ManifestacaoServiceImpl extends ServiceImpl implements  
ManifestacaoService{
```

```
    public ManifestacaoServiceImpl() {  
    }  
}
```

```
    public List getDadosOcorrencia() throws Exception {  
        return ((ManifestacaoDAO)dao).getDadosOcorrencia();  
    }  
}
```

```
    public HashMap getDadosCadastro() throws Exception {  
        return ((ManifestacaoDAO)dao).getDadosCadastro();  
    }  
}
```

```
public HashMap getDadosCadastroManifestacao() throws Exception {
    return ((ManifestacaoDAO)dao).getDadosCadastroManifestacao();
}

public Manifestacao salvarExternaAnonima(Manifestacao manifestacao) throws
Exception {
    return
((ManifestacaoDAO)dao).salvarExternaAnonima(this.incluiDatas(manifestacao));
}

public Manifestacao salvarExternaManifestante(Manifestacao manifestacao)
throws Exception {
    return
((ManifestacaoDAO)dao).salvarExternaManifestante(this.incluiDatas(manifestacao));
}

public List getEstados() throws Exception {
    return ((ManifestacaoDAO)dao).getEstados();
}

public List getCidadesPorEstado(Estado estado) throws Exception{
    return ((ManifestacaoDAO)dao).getCidadesPorEstado(estado);
}

public Manifestacao salvarInternaAnonima(Manifestacao manifestacao) throws
Exception {
    return
((ManifestacaoDAO)dao).salvarInternaAnonima(this.incluiDatas(manifestacao));
}

public Manifestacao salvarInternaManifestante(Manifestacao manifestacao)
throws Exception {
    return
((ManifestacaoDAO)dao).salvarInternaManifestante(this.incluiDatas(manifestacao));
}
```

```

private Manifestacao incluiDatas(Manifestacao manifestacao){
    Date dataAtual = new Date();
    manifestacao.setDataHoraRecebimento(dataAtual);
    manifestacao.setDataUltimaModificacao(dataAtual);
    return manifestacao;
}

public HashMap getDadosConsulta() throws Exception {
    return ((ManifestacaoDAO)dao).getDadosConsulta();
}

public List consultar(ConsultaManifestacaoDTO dto) throws Exception {
    return ((ManifestacaoDAO)dao).consultar(dto);
}

public Date getMenorDataRecebimento() throws Exception {
    return ((ManifestacaoDAO)dao).getMenorDataRecebimento();
}
}

..\src\java\br\com\omd\ouvidoria\logica\service\impl\NotificacaoServiceImpl.java
/*
 * NotificacaoServiceImpl.java
 *
 */
package br.com.omd.ouvidoria.logica.service.impl;

import br.com.omd.ouvidoria.logica.domain.Notificacao;
import br.com.omd.ouvidoria.logica.service.NotificacaoService;
import java.util.Collection;

public class NotificacaoServiceImpl extends ServiceImpl implements
NotificacaoService {

    /** Creates a new instance of NotificacaoServiceImpl */

```

```

public NotificacaoServiceImpl() {
}

public Collection getAll(String ordenacao) throws Exception {
    return dao.getAll(ORDENAR_POR_DESCRICAO);
}

public void atualizaHabilitado(Notificacao notificacao) throws Exception {
    boolean habilitado = notificacao.isHabilitado();
    notificacao = (Notificacao) dao.getDomain(notificacao.getId());
    notificacao.setHabilitado(habilitado);
    dao.save(notificacao);
}
}

..\src\java\br\com\omd\ouvidoria\logica\service\impl\OutraUnidadeServiceImpl.java
/*
 * OutraUnidadeServiceImpl.java
 *
 */
package br.com.omd.ouvidoria.logica.service.impl;

import br.com.omd.ouvidoria.logica.domain.Unidade;
import br.com.omd.ouvidoria.logica.domain.BaseDomain;
import br.com.omd.ouvidoria.logica.domain.outros.OutraUnidade;
import br.com.omd.ouvidoria.logica.dto.UnidadeDTO;
import br.com.omd.ouvidoria.logica.service.OutraUnidadeService;
import br.com.omd.ouvidoria.persistencia.geral.outros.dao.OutraUnidadeDAO;
import java.util.Collection;

public class OutraUnidadeServiceImpl implements OutraUnidadeService{

    private OutraUnidadeDAO outraUnidadeDAO;

```

```

/** Creates a new instance of UnidadeServiceImpl */
public OutraUnidadeServiceImpl() {

}

public void edit(OutraUnidade outraUnidade, Unidade unidadePadrao) throws
Exception {
    OutraUnidade unidadePronto=new OutraUnidade();
    OutraUnidade
    outraUnidadeParaAlterar=(OutraUnidade)selectPorId(outraUnidade);

    if (outraUnidadeParaAlterar!=null){
        if (outraUnidadeParaAlterar.getId().equals(outraUnidade.getId())){

outraUnidadeDAO.substituirOutrosPorDomain(outraUnidadeParaAlterar.getNome(),u
nidadePadrao);
        }
    }
}

public BaseDomain selectPorId(OutraUnidade outraUnidade) throws Exception {
    return outraUnidadeDAO.getDomain(outraUnidade.getId());
}

public Collection getAll() throws Exception {
    return outraUnidadeDAO.getAll(null);
}

public OutraUnidadeDAO getUnidadeDAO() {
    return outraUnidadeDAO;
}

public void setOutraUnidadeDAO(OutraUnidadeDAO outraUnidadeDAO) {
    this.outraUnidadeDAO = outraUnidadeDAO;
}
}

```

```
..\src\java\br\com\omd\ouvidoria\logica\service\impl\OutroAssuntoServiceImpl.java
```

```
/*
```

```
 * OutroAssuntoServiceImpl.java
```

```
 *
```

```
*/
```

```
package br.com.omd.ouvidoria.logica.service.impl;
```

```
import br.com.omd.ouvidoria.logica.domain.Assunto;
```

```
import br.com.omd.ouvidoria.logica.domain.BaseDomain;
```

```
import br.com.omd.ouvidoria.logica.domain.outros.OutroAssunto;
```

```
import br.com.omd.ouvidoria.logica.dto.AssuntoDTO;
```

```
import br.com.omd.ouvidoria.logica.service.OutroAssuntoService;
```

```
import br.com.omd.ouvidoria.persistencia.geral.outros.dao.OutroAssuntoDAO;
```

```
import java.util.Collection;
```

```
public class OutroAssuntoServiceImpl implements OutroAssuntoService{
```

```
    private OutroAssuntoDAO outroAssuntoDAO;
```

```
    /** Creates a new instance of AssuntoServiceImpl */
```

```
    public OutroAssuntoServiceImpl() {
```

```
    }
```

```
    public void edit(OutroAssunto outroAssunto, Assunto assuntoPadrao) throws  
    Exception {
```

```
        OutroAssunto assuntoPronto=new OutroAssunto();
```

```
        outroAssuntoDAO.substituirOutrosPorDomain(outroAssunto.getNome(), assuntoPadrao);
```

```
    }
```

```

public BaseDomain selectPorId(OutroAssunto outroAssunto) throws Exception {
    return outroAssuntoDAO.getDomain(outroAssunto.getId());
}

public Collection getAll() throws Exception {
    return outroAssuntoDAO.getAll(null);
}

public OutroAssuntoDAO getAssuntoDAO() {
    return outroAssuntoDAO;
}

public void setOutroAssuntoDAO(OutroAssuntoDAO outroAssuntoDAO) {
    this.outroAssuntoDAO = outroAssuntoDAO;
}
}

```

```

..\src\java\br\com\omd\ouvidoria\logica\service\impl\PerfilServiceImpl.java

```

```

/*

```

```

 * PerfilServiceImpl.java

```

```

 *

```

```

 */

```

```

package br.com.omd.ouvidoria.logica.service.impl;

```

```

import br.com.omd.ouvidoria.logica.service.PerfilService;

```

```

import br.com.omd.ouvidoria.persistencia.geral.dao.PerfilDAO;

```

```

import java.util.Collection;

```

```

public class PerfilServiceImpl implements PerfilService{

```

```

    private PerfilDAO perfilDAO;

```

```

    /** Creates a new instance of AssuntoServiceImpl */

```

```

    public PerfilServiceImpl() {

```

```

    }

    public Collection getAll() throws Exception {
        return perfilDAO.getAll(null);
    }

    public PerfilDAO getAssuntoDAO() {
        return perfilDAO;
    }

    public void setPerfilDAO(PerfilDAO perfilDAO) {
        this.perfilDAO = perfilDAO;
    }
}

..\src\java\br\com\omd\ouvidoria\logica\service\impl\PrazoServiceImpl.java
/*
 * PrazoServiceImpl.java
 *
 */
package br.com.omd.ouvidoria.logica.service.impl;

import br.com.omd.ouvidoria.logica.domain.BaseDomain;
import br.com.omd.ouvidoria.logica.domain.Prazo;
import br.com.omd.ouvidoria.logica.dto.PrazoDTO;
import br.com.omd.ouvidoria.logica.service.PrazoService;
import br.com.omd.ouvidoria.persistencia.geral.dao.PrazoDAO;
import java.util.ArrayList;
import java.util.Collection;
import org.apache.commons.beanutils.BeanUtils;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

```

```

public class PrazoServiceImpl implements PrazoService{

    private PrazoDAO prazoDAO;

    /** Creates a new instance of PrazoServiceImpl */
    public PrazoServiceImpl() {
//      ApplicationContext ctx = new
ClassPathXmlApplicationContext("applicationContext-dao.xml");
//      prazoDAO = (PrazoDAO)ctx.getBean("prazo");
    }

    public void save(PrazoDTO prazoDTO) throws Exception {

    }

    public void edit(PrazoDTO prazoDTO) throws Exception {
        Prazo prazoPronto=new Prazo();
        PrazoDTO prazoDTO_Old=new PrazoDTO();
        prazoDTO_Old.setId(prazoDTO.getId());

        Prazo prazoParaAlterar=(Prazo)selectPorId(prazoDTO_Old);

        if (prazoParaAlterar!=null){
            if (prazoParaAlterar.getId().equals(prazoDTO.getId())){
                prazoParaAlterar.setPrazo(prazoDTO.getPrazo());
                prazoDAO.edit(prazoParaAlterar);
            }
        }
    }

    public void delete(PrazoDTO prazoDTO) throws Exception {

    }

    public Collection select(PrazoDTO prazoDTO) throws Exception {

```

```

        return null;
    }

    public BaseDomain selectPorId(PrazoDTO prazoDTO) throws Exception {
        return prazoDAO.getDomain(prazoDTO.getId());
    }

    public Collection getAll() throws Exception {
        return prazoDAO.getAll(null);
    }

    public PrazoDAO getPrazoDAO() {
        return prazoDAO;
    }

    public void setPrazoDAO(PrazoDAO prazoDAO) {
        this.prazoDAO = prazoDAO;
    }
}

..\src\java\br\com\omd\ouvidoria\logica\service\impl\RecebimentoServiceImpl.java
/*
 * RecebimentoServiceImpl.java
 *
 */
package br.com.omd.ouvidoria.logica.service.impl;

import br.com.omd.ouvidoria.logica.service.RecebimentoService;
import br.com.omd.ouvidoria.persistencia.geral.dao.RecebimentoDAO;
import java.util.Collection;

public class RecebimentoServiceImpl implements RecebimentoService{

    private RecebimentoDAO RecebimentoDAO;

```

```

/**
 * Creates a new instance of RecebimentoServiceImpl
 */
public RecebimentoServiceImpl() {
}

public Collection getAll() throws Exception {
    return RecebimentoDAO.getAll(null);
}

public RecebimentoDAO getRecebimentoDAO() {
    return RecebimentoDAO;
}

public void setRecebimentoDAO(RecebimentoDAO RecebimentoDAO) {
    this.RecebimentoDAO = RecebimentoDAO;
}
}

..\src\java\br\com\omd\ouvidoria\logica\service\impl\RelatorioServiceImpl.java
/*
 * RelatorioServiceImpl.java
 *
 */
package br.com.omd.ouvidoria.logica.service.impl;

import br.com.omd.ouvidoria.logica.domain.Assunto;
import br.com.omd.ouvidoria.logica.domain.Classificacao;
import br.com.omd.ouvidoria.logica.domain.Unidade;
import br.com.omd.ouvidoria.logica.domain.Usuario;
import br.com.omd.ouvidoria.logica.service.RelatorioService;
import br.com.omd.ouvidoria.persistencia.geral.dao.RelatorioDAO;
import br.com.omd.ouvidoria.util.Data;
import java.util.Collection;

```

```

import java.util.LinkedHashMap;

public class RelatorioServiceImpl implements RelatorioService{

    private RelatorioDAO relatorioDAO;

    /**
     * Creates a new instance of RelatorioServiceImpl
     */
    public RelatorioServiceImpl() {
    }

    public Collection getAll() throws Exception {
        return relatorioDAO.getAll(null);
    }

    public RelatorioDAO getRelatorioDAO() {
        return relatorioDAO;
    }

    public void setRelatorioDAO(RelatorioDAO relatorioDAO) {
        this.relatorioDAO = relatorioDAO;
    }

    public Collection getManifestacoes(Classificacao classificacao, Unidade unidade,
Assunto assunto, Data dataInicial, Data dataFinal) throws Exception {
        return relatorioDAO.getManifestacoes( classificacao, unidade, assunto,
dataInicial, dataFinal);
    }

    public Collection getClassificacao(Unidade unidade, Data dataInicial, Data
dataFinal) throws Exception {
        return relatorioDAO.getClassificacao( unidade, dataInicial, dataFinal);
    }

    public Collection getUnidades(Unidade unidade, Data dataInicial, Data dataFinal)
throws Exception {

```

```

        return relatorioDAO.getUnidades(unidade, dataInicial, dataFinal);
    }

    public LinkedHashMap getAssuntos(Unidade unidade, Data dataInicial, Data
dataFinal) throws Exception {
        return relatorioDAO.getAssuntos(unidade,dataInicial,dataFinal);
    }

    public Collection getUnidadesPorUsuario(Usuario user) throws Exception {
        return relatorioDAO.getUnidadesPorUsuario(user);
    }

    public Collection getUnidadesVinculadasPorUnidade(Unidade unit) throws
Exception {
        return relatorioDAO.getUnidadesVinculadasPorUnidade(unit);
    }
}

```

```

..\src\java\br\com\omd\ouvidoria\logica\service\impl\ServiceImpl.java

```

```

/*

```

```

 * ServiceImpl.java

```

```

 *

```

```

*/

```

```

package br.com.omd.ouvidoria.logica.service.impl;

```

```

import br.com.omd.ouvidoria.logica.domain.BaseDomain;

```

```

import br.com.omd.ouvidoria.logica.service.Service;

```

```

import br.com.omd.ouvidoria.persistencia.geral.dao.DAO;

```

```

import java.util.Collection;

```

```

public class ServiceImpl implements Service{

```

```

    protected DAO dao;

```

```

    /** Creates a new instance of ServiceImpl */

```

```

public ServiceImpl() {

}

public Collection getAll() throws Exception {
    return dao.getAll(null);
}

public DAO getDao() {
    return dao;
}

public void setDao(DAO dao) {
    this.dao = dao;
}
}

..\src\java\br\com\omd\ouvidoria\logica\service\impl\UnidadeServiceImpl.java
/*
 * UnidadeServiceImpl.java
 *
 */
package br.com.omd.ouvidoria.logica.service.impl;

import br.com.omd.ouvidoria.logica.domain.BaseDomain;
import br.com.omd.ouvidoria.logica.domain.Unidade;
import br.com.omd.ouvidoria.logica.domain.outros.OutraUnidade;
import br.com.omd.ouvidoria.logica.service.Service;
import br.com.omd.ouvidoria.logica.service.UnidadeService;
import br.com.omd.ouvidoria.persistencia.geral.dao.UnidadeDAO;
import br.com.omd.ouvidoria.persistencia.geral.outros.dao.OutraUnidadeDAO;
import java.util.Collection;

public class UnidadeServiceImpl implements UnidadeService{

```

```

private UnidadeDAO unidadeDAO;
private OutraUnidadeDAO outraUnidadeDAO;

/** Creates a new instance of UnidadeServiceImpl */
public UnidadeServiceImpl() {
}
public UnidadeDAO getUnidadeDAO() {
    return unidadeDAO;
}

public void setUnidadeDAO(UnidadeDAO unidadeDAO) {
    this.unidadeDAO = unidadeDAO;
}

public void save(Unidade unidade) throws Exception {
    getUnidadeDAO().save(unidade);
}

public void delete(Unidade unidade) throws Exception {
    getUnidadeDAO().delete(unidade);
}

public Collection select(Unidade unidade) throws Exception {
    return getUnidadeDAO().pesquisarPorNome(unidade.getNome(),
UnidadeService.ORDENAR_POR_NOME);
}

public BaseDomain selectPorId(Unidade unidade ) throws Exception {
    return unidadeDAO.getDomain(unidade.getId());
}

public BaseDomain selectOutraUnidadePorId(OutraUnidade outra_unidade )
throws Exception {
    return getUnidadeDAO().getDomain(outra_unidade.getId());
}

```

```
}  
public Collection getAll() throws Exception {  
    Collection colecao= getUnidadeDAO().getAll(Service.ORDENAR_POR_NOME);  
    return colecao;  
}
```

```
public void edit(Unidade unidade_Old, Unidade unidade_New) throws Exception {  
    Unidade unidadePronto=new Unidade();  
    Unidade unidadeParaAlterar=(Unidade)selectPorId(unidade_Old);
```

```
    if (unidadeParaAlterar!=null){  
        unidadeParaAlterar.setNome(unidade_New.getNome());  
        if(unidade_New.getSigla()!=null){  
            unidadeParaAlterar.setSigla(unidade_New.getSigla());  
        }  
    }
```

```
    if(unidade_New.getEndereco()!=null){  
        unidadeParaAlterar.setEndereco(unidade_New.getEndereco());  
    }else{  
        unidadeParaAlterar.setEndereco(null);  
    }  
}
```

```
    if(unidade_New.getBairro()!=null){  
        unidadeParaAlterar.setBairro(unidade_New.getBairro());  
    }else{  
        unidadeParaAlterar.setBairro(null);  
    }  
}
```

```
    if(unidade_New.getCidade()!=null){  
        unidadeParaAlterar.setCidade(unidade_New.getCidade());  
    }else{  
        unidadeParaAlterar.setCidade(null);  
    }  
}
```

```
if(unidade_New.getCep()!=null){
    unidadeParaAlterar.setCep(unidade_New.getCep());
}else{
    unidadeParaAlterar.setCep(null);
}
```

```
if(unidade_New.getDddTelefone1()!=null){
    unidadeParaAlterar.setDddTelefone1(unidade_New.getDddTelefone1());
}else{
    unidadeParaAlterar.setDddTelefone1(null);
}
```

```
if(unidade_New.getTelefone1()!=null){
    unidadeParaAlterar.setTelefone1(unidade_New.getTelefone1());
}else{
    unidadeParaAlterar.setTelefone1(null);
}
```

```
if(unidade_New.getDddTelefone2()!=null){
    unidadeParaAlterar.setDddTelefone2(unidade_New.getDddTelefone2());
}else{
    unidadeParaAlterar.setDddTelefone2(null);
}
```

```
if(unidade_New.getTelefone2()!=null){
    unidadeParaAlterar.setTelefone2(unidade_New.getTelefone2());
}else{
    unidadeParaAlterar.setTelefone2(null);
}
```

```
if(unidade_New.getDddFax()!=null){
    unidadeParaAlterar.setDddFax(unidade_New.getDddFax());
}else{
    unidadeParaAlterar.setDddFax(null);
}
```

```

    }

    if(unidade_New.getFax()!=null){
        unidadeParaAlterar.setFax(unidade_New.getFax());
    }else{
        unidadeParaAlterar.setFax(null);
    }

    if(unidade_New.getEmail()!=null){
        unidadeParaAlterar.setEmail(unidade_New.getEmail());
    }else{
        unidadeParaAlterar.setEmail(null);
    }

    if(unidade_New.getVinculo()!=null){
        unidadeParaAlterar.setVinculo(unidade_New.getVinculo());
    }

    if(unidade_New.getResponsavel()!=null){
        unidadeParaAlterar.setResponsavel(unidade_New.getResponsavel());
    }

    if (unidadeParaAlterar.getId().equals(unidade_Old.getId())){

        getUnidadeDAO().edit(unidadeParaAlterar);
    }
}

public Collection getQuantidadePorOutro() throws Exception {
    return getOutraUnidadeDAO().getQuantidadePorOutro();
}

public OutraUnidadeDAO getOutraUnidadeDAO() {

```

```

        return outraUnidadeDAO;
    }

    public void setOutraUnidadeDAO(OutraUnidadeDAO outraUnidadeDAO) {
        this.outraUnidadeDAO = outraUnidadeDAO;
    }

    public void changeUnidadePorOutraUnidade(OutraUnidade outraUnidade,
        Unidade unidadePadrao) throws Exception {

        outraUnidadeDAO.substituirOutrosPorDomain(outraUnidade.getNome(),unidadePadrao);
    }

    public Collection getAllSemOutraSemNenhuma() throws Exception {
        UnidadeDAO unidadeDAO=getUnidadeDAO();
        Collection unidades=unidadeDAO.getAllSemOutraSemNenhuma();
        return unidades;
    }
}

..\src\java\br\com\omd\ouvidoria\logica\service\impl\UsuarioServiceImpl.java
/*
 * UsuarioServiceImpl.java
 *
 */
package br.com.omd.ouvidoria.logica.service.impl;

import br.com.omd.ouvidoria.logica.domain.BaseDomain;
import br.com.omd.ouvidoria.logica.domain.Usuario;
import br.com.omd.ouvidoria.logica.dto.UsuarioDTO;
import br.com.omd.ouvidoria.logica.service.UsuarioService;
import br.com.omd.ouvidoria.persistencia.geral.dao.UsuarioDAO;
import java.util.Collection;

```

```

public class UsuarioServiceImpl implements UsuarioService {

    private UsuarioDAO usuarioDAO;

    /** Creates a new instance of UsuarioServiceImpl */
    public UsuarioServiceImpl() {
    }

    public Usuario autenticar(UsuarioDTO usuarioDTO) throws Exception {
        return getUsuarioDAO().autenticar(usuarioDTO);
    }

    public UsuarioDAO getUsuarioDAO() {
        return usuarioDAO;
    }

    public void setUsuarioDAO(UsuarioDAO usuarioDAO) {
        this.usuarioDAO = usuarioDAO;
    }

    public void save(Usuario usuario) throws Exception {
        usuarioDAO.save(usuario);
    }

    public void delete(Usuario usuario) throws Exception {
        usuarioDAO.delete(usuario);
    }

    public Collection select(Usuario usuario) throws Exception {
        return usuarioDAO.pesquisarPorNome(usuario.getNome(),
        UsuarioService.ORDENAR_POR_NOME);
    }

    public BaseDomain selectPorId(Usuario usuario ) throws Exception {
        return usuarioDAO.getDomain(usuario.getId());
    }
}

```

```
}
```

```
public Collection getAll() throws Exception {  
    return usuarioDAO.getAll(null);  
}
```

```
public void edit(Usuario usuario_Old, Usuario usuario_New) throws Exception {  
    Usuario usuarioPronto=new Usuario();  
    Usuario usuarioParaAlterar=(Usuario)selectPorId(usuario_Old);
```

```
    if (usuarioParaAlterar!=null){  
        if(usuario_New.getNome()!=null){  
            usuarioParaAlterar.setNome(usuario_New.getNome());  
        }  
        if(usuario_New.getLogin()!=null){  
            usuarioParaAlterar.setLogin(usuario_New.getLogin());  
        }  
        if(usuario_New.getLogin()!=null){  
            usuarioParaAlterar.setLogin(usuario_New.getLogin());  
        }  
        if(usuario_New.getApellido()!=null){  
            usuarioParaAlterar.setApellido(usuario_New.getApellido());  
        }  
        if(usuario_New.getSenha()!=null){  
            usuarioParaAlterar.setSenha(usuario_New.getSenha());  
        }else{  
            usuarioParaAlterar.setSenha(null);  
        }  
        if(usuario_New.getDddTelefone1()!=null){  
            usuarioParaAlterar.setDddTelefone1(usuario_New.getDddTelefone1());  
        }  
        if(usuario_New.getTelefone1()!=null){  
            usuarioParaAlterar.setTelefone1(usuario_New.getTelefone1());  
        }  
    }
```



```

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.DynaActionForm;

public class AssuntoAction extends BaseAction{

    public AssuntoAction() {
    }

    public ActionForward defaultMethod(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{
        return getAssuntos(mapping, form, request, response);
    }
    /**
     *@version 1.0
     *@method addAssunto
     *@author David Gerson Ramos de Andrade
     *@description action criado para inserção de novo assunto.
     */
    public ActionForward addAssunto(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{
        DynaActionForm dynaform=(DynaActionForm)form;
        try{
            String nomeNovoAssunto=(String)dynaform.get("assuntounovo");

```

```

        AssuntoService assuntoService =
(AssuntoService)getWebApplicationContext().getBean("assuntoService");
        Assunto assunto = new Assunto();
        assunto.setNome(nomeNovoAssunto);
        assuntoService.save(assunto);

        request.setAttribute("titulo", "Assunto Adicionado");
        request.setAttribute("mensagem", "Assunto adicionado com sucesso.");

        return mapping.findForward("resultadoSucesso");
    } catch(Exception e) {
        request.setAttribute("mensagem", e.getMessage());
        return mapping.findForward("erro");
    }
}
/**
 *@version 1.0
 *@method deleteAssunto
 *@author David Gerson Ramos de Andrade
 *@description action criado para remoção de novo assunto.
 */
public ActionForward deleteAssunto(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{

    DynaActionForm dynaform=(DynaActionForm)form;

    String idExcluirAssunto=(String)dynaform.get("idAssuntosRemoverCombo");
    String
senhaExclusao=Criptografia.retornaString(Criptografia.digest(((String)dynaform.get("
senha")).getBytes()));

```

```

HttpSession session = request.getSession(true);
Usuario user=(Usuario)session.getAttribute("usuario");
String destino= "resultadoSucesso";
UsuarioDTO userDTO=new UsuarioDTO();
userDTO.setLogin(user.getApellido());
userDTO.setSenha(senhaExclusao);

if (user != null) {
    if (user.getSenha().equals(senhaExclusao)) {
        try {

            UsuarioService usuarioService =
(UsuarioService)getWebApplicationContext().getBean("usuarioService");

            AssuntoService assuntoService =
(AssuntoService)getWebApplicationContext().getBean("assuntoService");
            Assunto assunto = new Assunto();
            assunto.setId(Long.valueOf(idExcluirAssunto));
            assuntoService.delete(assunto);

            request.setAttribute("titulo", "Assunto Excluído");
            request.setAttribute("mensagem", "Assunto excluído com sucesso.");

        } catch(Exception e) {
            request.setAttribute("mensagem", e.getMessage());
            destino = "erro";
        }
    } else {
        request.setAttribute("mensagem", "Senha inválida.");
        destino = "erro";
    }
} else {
    request.setAttribute("mensagem", "Usuário não logado.");
    destino = "erro";
}

```

```

    }
    return mapping.findForward(destino);
}
public ActionForward getAssunto(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{

    DynaActionForm dynaform=(DynaActionForm)form;
    try{
        String nomeAssunto=(String)dynaform.get("assuntoconsulta");

        AssuntoService assuntoService =
(AssuntoService)getWebApplicationContext().getBean("assuntoService");
        Assunto assunto = new Assunto();
        assunto.setNome(nomeAssunto);
        Collection listaAssuntosEncontrados=assuntoService.select(assunto);

        request.setAttribute("linhas",listaAssuntosEncontrados);

request.setAttribute("tamanhoLinhas",Integer.toString(listaAssuntosEncontrados.size
()));

        return mapping.findForward("resultadoConsulta");
    } catch(Exception e) {
        request.setAttribute("mensagem", e.getMessage());
        return mapping.findForward("erro");
    }
}

public ActionForward getAssuntos(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{
    DynaActionForm dynaform=(DynaActionForm)form;

```

```

try{

    AssuntoService assuntoService =
(AssuntoService)getWebApplicationContext().getBean("assuntoService");

request.setAttribute("listaOutrosAssuntos",assuntoService.getQuantidadePorOutro())
;

    request.setAttribute("listaAssuntos",assuntoService.getAll());

    return mapping.findForward("success");
} catch(Exception e) {
    request.setAttribute("mensagem", e.getMessage());
    return mapping.findForward("erro");
}
}

public ActionForward changeAssuntos(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{
DynaActionForm dynaform=(DynaActionForm)form;
try{
String idPadraoAssunto=(String)dynaform.get("idAssuntosChangeCombo");
String idOutroAssunto=(String)dynaform.get("idOutrosAssuntosCombo");

Assunto assuntoPadrao=new Assunto();
OutroAssunto outroAssunto=new OutroAssunto();

assuntoPadrao.setId(Long.valueOf(idPadraoAssunto));
outroAssunto.setNome(idOutroAssunto);

    AssuntoService assuntoService =
(AssuntoService)getWebApplicationContext().getBean("assuntoService");
    assuntoService.changeAssuntoPorOutroAssunto(outroAssunto,assuntoPadrao);

```

```

request.setAttribute("titulo", "Assunto Substituído");
request.setAttribute("mensagem", "Assunto substituído com sucesso.");

return mapping.findForward("resultadoSucesso");
} catch(Exception e) {
    request.setAttribute("mensagem", e.getMessage());
    return mapping.findForward("erro");
}
}
/**
 *@version 1.0
 *@method editAssunto
 *@author David Gerson Ramos de Andrade
 *@description action criado para alteração de assunto.
 */
public ActionForward editAssunto(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{
    DynaActionForm dynaform=(DynaActionForm)form;
    try{
        String nomeNovoAssunto=(String)dynaform.get("assuntoalterado");
        String idNovoAssunto=(String)dynaform.get("idAssuntosEditarCombo");

        AssuntoService assuntoService =
(AssuntoService)getWebApplicationContext().getBean("assuntoService");
        Assunto assunto_Old = new Assunto();
        Assunto assunto_New = new Assunto();
        assunto_New.setNome(nomeNovoAssunto);
        assunto_Old.setld(Long.valueOf(idNovoAssunto));
        assuntoService.edit(assunto_Old,assunto_New);

        request.setAttribute("titulo", "Assunto Alterado");
        request.setAttribute("mensagem", "Assunto alterado com sucesso.");
    }
}

```

```
        return mapping.findForward("resultadoSucesso");
    } catch(Exception e) {
        request.setAttribute("mensagem", e.getMessage());
        return mapping.findForward("erro");
    }
}
}
```

```
..\src\java\br\com\omd\ouvidoria\web\controle\action\BaseAction.java
```

```
/*
```

```
* BaseAction.java
```

```
*
```

```
*/
```

```
package br.com.omd.ouvidoria.web.controle.action;
```

```
import javax.servlet.http.HttpServletRequest;
```

```
import javax.servlet.http.HttpServletResponse;
```

```
import org.apache.struts.action.Action;
```

```
import org.apache.struts.action.ActionForm;
```

```
import org.apache.struts.action.ActionForward;
```

```
import org.apache.struts.action.ActionMapping;
```

```
import org.apache.struts.action.DynaActionForm;
```

```
import org.springframework.web.struts.DispatchActionSupport;
```

```
public abstract class BaseAction extends DispatchActionSupport {
```

```
    public ActionForward execute(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception {
        String method = request.getParameter("method");
```

```
    if (method == null || method.equals("")){
        method = "defaultMethod";
    }
    return dispatchMethod(mapping, form, request, response, method);
}
```

```
public abstract ActionForward defaultMethod(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception;
}
```

```
..\src\java\br\com\omd\ouvidoria\web\controle\action\CadastroExternoManifestacaoAction.java
```

```
package br.com.omd.ouvidoria.web.controle.action;
```

```
import br.com.omd.ouvidoria.logica.domain.Assunto;
import br.com.omd.ouvidoria.logica.domain.Cidade;
import br.com.omd.ouvidoria.logica.domain.Classificacao;
import br.com.omd.ouvidoria.logica.domain.Estado;
import br.com.omd.ouvidoria.logica.domain.Identificacao;
import br.com.omd.ouvidoria.logica.domain.Manifestacao;
import br.com.omd.ouvidoria.logica.domain.Manifestante;
import br.com.omd.ouvidoria.logica.domain.Modo;
import br.com.omd.ouvidoria.logica.domain.Unidade;
import br.com.omd.ouvidoria.logica.domain.outros.OutraCidade;
import br.com.omd.ouvidoria.logica.domain.outros.OutraUnidade;
import br.com.omd.ouvidoria.logica.domain.outros.OutroAssunto;
import br.com.omd.ouvidoria.logica.service.ManifestacaoService;
import java.text.SimpleDateFormat;
import java.util.HashMap;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
```

```

import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.DynaActionForm;

public class CadastroExternoManifestacaoAction extends BaseAction{

    public CadastroExternoManifestacaoAction() {
    }

    public ActionForward defaultMethod(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{
        return getDadosOcorrencia(mapping, form, request, response);
    }

    public ActionForward getDadosOcorrencia(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{
        DynaActionForm dynaform=(DynaActionForm)form;
        dynaform.initialize(mapping);
        ManifestacaoService manifestacaoService =
        (ManifestacaoService)getWebApplicationContext().getBean("manifestacaoService");
        request.setAttribute("unidades",manifestacaoService.getDadosOcorrencia());

        return mapping.findForward("dadosOcorrencia");
    }

    public ActionForward criaDadosOcorrencia(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{

```

```

DynaActionForm dynaform=(DynaActionForm)form;
SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyyhh:mm");

Manifestacao manifestacao = new Manifestacao();

manifestacao.setDataHoraOcorrencia(sdf.parse((String)dynaform.get("data")+ (String)
dynaform.get("hora")));
Unidade unidade = new Unidade();
String nomeUnidade = (String)dynaform.get("unidade");
unidade.setNome(nomeUnidade);
manifestacao.setUnidadeOcorrencia(unidade);

if (nomeUnidade.equals("Outra")){
    OutraUnidade outraUnidade = new OutraUnidade();
    outraUnidade.setNome((String)dynaform.get("outraUnidade"));
    manifestacao.setOutraUnidade(outraUnidade);
}

String envolvidos = dynaform.getString("envolvidos");
if (!envolvidos.equals("")){
    manifestacao.setEnvolvidos((String)dynaform.get("envolvidos"));
}
manifestacao.setDescricao((String)dynaform.get("descricao"));
dynaform.set("manifestacao", manifestacao);

//Prepara passo 2
ManifestacaoService manifestacaoService =
(ManifestacaoService)getWebApplicationContext().getBean("manifestacaoService");
HashMap dadosCadastro = manifestacaoService.getDadosCadastro();
request.setAttribute("identificacoes", dadosCadastro.get("identificacoes"));
request.setAttribute("modosRetorno", dadosCadastro.get("modosRetorno"));
request.setAttribute("classificacoes", dadosCadastro.get("classificacoes"));
request.setAttribute("assuntos", dadosCadastro.get("assuntos"));

return mapping.findForward("dadosCadastro");

```

```
}
```

```
public ActionForward criaDadosCadastro(ActionMapping mapping,  
    ActionForm form,  
    HttpServletRequest request,  
    HttpServletResponse response) throws Exception{  
    DynaActionForm dynaform=(DynaActionForm)form;  
    Manifestacao manifestacao = (Manifestacao) dynaform.get("manifestacao");  
    if (manifestacao.getDescricao() == null){  
        return this.getDadosOcorrencia(mapping, form, request, response);  
    } else {  
        Identificacao identificacao = new Identificacao();  
        identificacao.setNome((String)dynaform.get("identificacao"));  
        manifestacao.setIdentificacao(identificacao);  
  
        Classificacao classificacao = new Classificacao();  
        classificacao.setId((Long)dynaform.get("classificacao"));  
        manifestacao.setClassificacao(classificacao);  
  
        Assunto assunto = new Assunto();  
        String nomeAssunto = (String)dynaform.get("assunto");  
        assunto.setNome(nomeAssunto);  
        manifestacao.setAssunto(assunto);  
        if (nomeAssunto.equals("Outro")){  
            OutroAssunto outroAssunto = new OutroAssunto();  
            outroAssunto.setNome((String)dynaform.get("outroAssunto"));  
            manifestacao.setOutroAssunto(outroAssunto);  
        }  
        String paginaDestino;  
        if (identificacao.getNome().equals("Anônima")) {  
            ManifestacaoService manifestacaoService =  
(ManifestacaoService)getWebApplicationContext().getBean("manifestacaoService");  
            request.setAttribute("manifestacao",  
manifestacaoService.salvarExternaAnonima(manifestacao));
```

```

        dynaform.initialize(mapping);
        paginaDestino = "manifestacaoCadastrada";
    } else {
        Modo modoRetorno = new Modo();
        modoRetorno.setNome((String)dynaform.get("modoRetorno"));
        manifestacao.setModoRetorno(modoRetorno);
        //Prepara passo 3
        if (modoRetorno.getNome().equals("Correspondência")){
            //Consultar Estado e Cidade
            ManifestacaoService manifestacaoService =
(ManifestacaoService)getWebApplicationContext().getBean("manifestacaoService");
            request.setAttribute("estados",manifestacaoService.getEstados());
        }
        HttpSession session = request.getSession(false);
        session.setAttribute("manifestacao", manifestacao);
        paginaDestino = "cadastroManifestante";
    }
    return mapping.findForward(paginaDestino);
}
}

```

```

public ActionForward getCidades(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{
    DynaActionForm dynaform=(DynaActionForm)form;
    request.setAttribute("nome", (String)dynaform.get("nome"));
    request.setAttribute("endereco", (String)dynaform.get("endereco"));
    request.setAttribute("bairro", (String)dynaform.get("bairro"));
    request.setAttribute("cep", (String)dynaform.get("cep"));
    request.setAttribute("dddTelefone1", (String) dynaform.get("dddTelefone1"));
    request.setAttribute("telefone1", (String) dynaform.get("telefone1"));
    request.setAttribute("dddTelefone2", (String) dynaform.get("dddTelefone2"));
    request.setAttribute("telefone2", (String) dynaform.get("telefone2"));
}

```

```

Estado estado = new Estado();
estado.setId((Long)dynaform.get("estado"));
ManifestacaoService manifestacaoService =
(ManifestacaoService)getWebApplicationContext().getBean("manifestacaoService");
request.setAttribute("estados",manifestacaoService.getEstados());
request.setAttribute("cidades",
manifestacaoService.getCidadesPorEstado(estado));
request.setAttribute("idEstado", estado.getId());
return mapping.findForward("cadastroManifestante");
}

```

```

public ActionForward criaManifestante(ActionMapping mapping,
ActionForm form,
HttpServletRequest request,
HttpServletResponse response) throws Exception{
DynaActionForm dynaform=(DynaActionForm)form;
HttpSession session = request.getSession(false);
Manifestacao manifestacao = (Manifestacao)
session.getAttribute("manifestacao");
if (manifestacao == null){
return this.getDadosOcorrencia(mapping, form, request, response);
} else {
Manifestante manifestante = new Manifestante();
manifestante.setNome((String) dynaform.get("nome"));

String fax = (String) dynaform.get("fax");
if (!fax.equals("")){
manifestante.setDddFax((String) dynaform.get("dddFax"));
manifestante.setFax(fax);
}

String email = (String) dynaform.get("email");
if (!email.equals("")){
manifestante.setEmail(email);
}
}
}

```

```
String endereco = (String) dynaform.get("endereco");
if (!endereco.equals("")){
    manifestante.setEndereco(endereco);
    manifestante.setBairro((String) dynaform.get("bairro"));
    manifestante.setEndereco((String) dynaform.get("endereco"));

    Cidade cidade = new Cidade();
    cidade.setNome((String) dynaform.get("cidade"));
    manifestante.setCidade(cidade);

    if (cidade.getNome().equals("Outra")){
        OutraCidade outraCidade = new OutraCidade();
        outraCidade.setNome((String) dynaform.get("outraCidade"));

        Estado estado = new Estado();
        estado.setId((Long)dynaform.get("estado"));
        outraCidade.setEstado(estado);

        manifestante.setOutraCidade(outraCidade);
    }
    manifestante.setCep((String) dynaform.get("cep"));
}

String telefone1 = (String) dynaform.get("telefone1");
if (!telefone1.equals("")){
    manifestante.setDddTelefone1((String) dynaform.get("dddTelefone1"));
    manifestante.setTelefone1(telefone1);
}

String telefone2 = (String) dynaform.get("telefone2");
if (!telefone2.equals("")){
    manifestante.setDddTelefone2((String) dynaform.get("dddTelefone2"));
    manifestante.setTelefone2(telefone2);
}
```

```

        dynaform.initialize(mapping);
        manifestacao.setManifestante(manifestante);
        ManifestacaoService manifestacaoService =
(ManifestacaoService)getWebApplicationContext().getBean("manifestacaoService");
        session.removeAttribute("manifestacao");
        request.setAttribute("manifestacao",
manifestacaoService.salvarExternaManifestante(manifestacao));
        return mapping.findForward("manifestacaoCadastrada");
    }
}
}

```

..\src\java\br\com\omd\ouvidoria\web\controle\action\CadastroManifestacaoAction.java

```

package br.com.omd.ouvidoria.web.controle.action;

import br.com.omd.ouvidoria.logica.domain.Assunto;
import br.com.omd.ouvidoria.logica.domain.Cidade;
import br.com.omd.ouvidoria.logica.domain.Classificacao;
import br.com.omd.ouvidoria.logica.domain.Estado;
import br.com.omd.ouvidoria.logica.domain.Identificacao;
import br.com.omd.ouvidoria.logica.domain.Manifestacao;
import br.com.omd.ouvidoria.logica.domain.Manifestante;
import br.com.omd.ouvidoria.logica.domain.Modo;
import br.com.omd.ouvidoria.logica.domain.Unidade;
import br.com.omd.ouvidoria.logica.domain.outros.OutraCidade;
import br.com.omd.ouvidoria.logica.domain.outros.OutraUnidade;
import br.com.omd.ouvidoria.logica.domain.outros.OutroAssunto;
import br.com.omd.ouvidoria.logica.service.ManifestacaoService;
import java.text.SimpleDateFormat;
import java.util.HashMap;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import org.apache.struts.action.ActionForm;

```

```
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.DynaActionForm;

public class CadastroManifestacaoAction extends BaseAction{

    public CadastroManifestacaoAction() {
    }

    public ActionForward defaultMethod(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{
        return getDadosCadastroManifestacao(mapping, form, request, response);
    }

    public ActionForward getDadosCadastroManifestacao(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{
        ManifestacaoService manifestacaoService =
        (ManifestacaoService)getWebApplicationContext().getBean("manifestacaoService");
        HashMap dadosCadastro =
        manifestacaoService.getDadosCadastroManifestacao();
        request.setAttribute("unidades",dadosCadastro.get("unidades"));
        request.setAttribute("classificacoes",dadosCadastro.get("classificacoes"));
        request.setAttribute("modosEntrada",dadosCadastro.get("modosEntrada"));
        request.setAttribute("identificacoes",dadosCadastro.get("identificacoes"));
        request.setAttribute("modosRetorno",dadosCadastro.get("modosRetorno"));
        request.setAttribute("assuntos",dadosCadastro.get("assuntos"));

        return mapping.findForward("cadastroManifestacao");
    }
}
```

```

public ActionForward criaManifestacao(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{
    DynaActionForm dynaform=(DynaActionForm)form;
    SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyyhh:mm");

    Manifestacao manifestacao = new Manifestacao();

    manifestacao.setDataHoraOcorrencia(sdf.parse((String)dynaform.get("data")+
    (String)dynaform.get("hora")));
    Unidade unidade = new Unidade();
    String nomeUnidade = (String)dynaform.get("unidade");
    unidade.setNome(nomeUnidade);
    manifestacao.setUnidadeOcorrencia(unidade);

    if (nomeUnidade.equals("Outra")){
        OutraUnidade outraUnidade = new OutraUnidade();
        outraUnidade.setNome((String)dynaform.get("outraUnidade"));
        manifestacao.setOutraUnidade(outraUnidade);
    }
    String envolvidos = dynaform.getString("envolvidos");
    if (!envolvidos.equals("")){
        manifestacao.setEnvolvidos((String)dynaform.get("envolvidos"));
    }
    manifestacao.setDescricao((String)dynaform.get("descricao"));

    Identificacao identificacao = new Identificacao();
    identificacao.setNome((String)dynaform.get("identificacao"));
    manifestacao.setIdentificacao(identificacao);

    Classificacao classificacao = new Classificacao();
    classificacao.setId((Long)dynaform.get("classificacao"));
    manifestacao.setClassificacao(classificacao);

```

```

Modo modoEntrada = new Modo();
modoEntrada.setId((Long)dynaform.get("modoEntrada"));
manifestacao.setModoEntrada(modoEntrada);

Assunto assunto = new Assunto();
String nomeAssunto = (String)dynaform.get("assunto");
assunto.setNome(nomeAssunto);
manifestacao.setAssunto(assunto);
if (nomeAssunto.equals("Outro")){
    OutroAssunto outroAssunto = new OutroAssunto();
    outroAssunto.setNome((String)dynaform.get("outroAssunto"));
    manifestacao.setOutroAssunto(outroAssunto);
}

String paginaDestino;
if (identificacao.getNome().equals("Anônima")) {
    ManifestacaoService manifestacaoService =
(ManifestacaoService)getWebApplicationContext().getBean("manifestacaoService");
    request.setAttribute("manifestacao",
manifestacaoService.salvarInternaAnonima(manifestacao));
    dynaform.initialize(mapping);
    paginaDestino = "manifestacaoCadastrada";
} else {
    Modo modoRetorno = new Modo();
    modoRetorno.setNome((String)dynaform.get("modoRetorno"));
    manifestacao.setModoRetorno(modoRetorno);
    //Prepara passo 3
    if (modoRetorno.getNome().equals("Correspondência")){
        //Consultar Estado e Cidade
        ManifestacaoService manifestacaoService =
(ManifestacaoService)getWebApplicationContext().getBean("manifestacaoService");
        request.setAttribute("estados",manifestacaoService.getEstados());
    }
    HttpSession session = request.getSession(false);

```

```
        session.setAttribute("manifestacao", manifestacao);
        paginaDestino = "cadastroManifestante";
    }
    return mapping.findForward(paginaDestino);
}
```

```
public ActionForward getCidades(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{
    DynaActionForm dynaform=(DynaActionForm)form;
    request.setAttribute("nome", (String)dynaform.get("nome"));
    request.setAttribute("endereco", (String)dynaform.get("endereco"));
    request.setAttribute("bairro", (String)dynaform.get("bairro"));
    request.setAttribute("cep", (String)dynaform.get("cep"));
    request.setAttribute("dddTelefone1", (String) dynaform.get("dddTelefone1"));
    request.setAttribute("telefone1", (String) dynaform.get("telefone1"));
    request.setAttribute("dddTelefone2", (String) dynaform.get("dddTelefone2"));
    request.setAttribute("telefone2", (String) dynaform.get("telefone2"));

    Estado estado = new Estado();
    estado.setId((Long)dynaform.get("estado"));
    ManifestacaoService manifestacaoService =
    (ManifestacaoService)getWebApplicationContext().getBean("manifestacaoService");
    request.setAttribute("estados", manifestacaoService.getEstados());
    request.setAttribute("cidades",
    manifestacaoService.getCidadesPorEstado(estado));
    request.setAttribute("idEstado", estado.getId());
    return mapping.findForward("cadastroManifestante");
}
```

```
public ActionForward criaManifestante(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{
```

```
DynaActionForm dynaform=(DynaActionForm)form;
HttpSession session = request.getSession(false);
Manifestacao manifestacao = (Manifestacao)
session.getAttribute("manifestacao");
if (manifestacao == null){
    return this.getDadosCadastroManifestacao(mapping, form, request,
response);
} else {
    Manifestante manifestante = new Manifestante();
    manifestante.setNome((String) dynaform.get("nome"));

    String fax = (String) dynaform.get("fax");
    if (!fax.equals("")){
        manifestante.setDddFax((String) dynaform.get("dddFax"));
        manifestante.setFax(fax);
    }

    String email = (String) dynaform.get("email");
    if (!email.equals("")){
        manifestante.setEmail(email);
    }

    String endereco = (String) dynaform.get("endereco");
    if (!endereco.equals("")){
        manifestante.setEndereco(endereco);
        manifestante.setBairro((String) dynaform.get("bairro"));
        manifestante.setEndereco((String) dynaform.get("endereco"));

        Cidade cidade = new Cidade();
        cidade.setNome((String) dynaform.get("cidade"));
        manifestante.setCidade(cidade);

        if (cidade.getNome().equals("Outra")){
            OutraCidade outraCidade = new OutraCidade();
            outraCidade.setNome((String) dynaform.get("outraCidade"));
        }
    }
}
```

```

        Estado estado = new Estado();
        estado.setId((Long)dynaform.get("estado"));
        outraCidade.setEstado(estado);

        manifestante.setOutraCidade(outraCidade);
    }
    manifestante.setCep((String) dynaform.get("cep"));
}

String telefone1 = (String) dynaform.get("telefone1");
if (!telefone1.equals("")){
    manifestante.setDddTelefone1((String) dynaform.get("dddTelefone1"));
    manifestante.setTelefone1(telefone1);
}
String telefone2 = (String) dynaform.get("telefone2");
if (!telefone2.equals("")){
    manifestante.setDddTelefone2((String) dynaform.get("dddTelefone2"));
    manifestante.setTelefone2(telefone2);
}
dynaform.initialize(mapping);
manifestacao.setManifestante(manifestante);
ManifestacaoService manifestacaoService =
(ManifestacaoService)getWebApplicationContext().getBean("manifestacaoService");
session.removeAttribute("manifestacao");
request.setAttribute("manifestacao",
manifestacaoService.salvarInternaManifestante(manifestacao));
return mapping.findForward("manifestacaoCadastrada");
}
}
}

```

```

..\src\java\br\com\omd\ouvidoria\web\controle\action\CargoAction.java
package br.com.omd.ouvidoria.web.controle.action;

```

```
import br.com.ond.ouvidoria.logica.domain.Usuario;
import br.com.ond.ouvidoria.logica.dto.CargoDTO;
import br.com.ond.ouvidoria.logica.dto.UsuarioDTO;
import br.com.ond.ouvidoria.logica.service.CargoService;
import br.com.ond.ouvidoria.logica.service.UsuarioService;
import br.com.ond.ouvidoria.util.Criptografia;
import java.util.Collection;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.DynaActionForm;
```

```
public class CargoAction extends BaseAction{
```

```
    public CargoAction() {
    }
}
```

```
    public ActionForward defaultMethod(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{
        return getCargos(mapping, form, request, response);
    }
}
```

```
/**
```

```
 *@version 1.0
```

```
 *@method addCargo
```

```
 *@author David Gerson Ramos de Andrade
```

```
 *@description action criado para inserção de novo cargo.
```

```
 */
```

```

public ActionForward addCargo(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{
    DynaActionForm dynaform=(DynaActionForm)form;
    try{
        String nomeNovoCargo=(String)dynaform.get("nome");

        CargoService cargoService =
(CargoService)getWebApplicationContext().getBean("cargoService");
        CargoDTO cargoDTO = new CargoDTO();
        cargoDTO.setNome(nomeNovoCargo);
        cargoService.save(cargoDTO);

        request.setAttribute("titulo", "Cargo Adicionado");
        request.setAttribute("mensagem", "Cargo adicionado com sucesso.");

        return mapping.findForward("resultadoSucesso");
    } catch(Exception e) {
        request.setAttribute("mensagem", e.getMessage());
        return mapping.findForward("erro");
    }
}
/**
 *@version 1.0
 *@method deleteCargo
 *@author David Gerson Ramos de Andrade
 *@description action criado para remoção de novo cargo.
 */
public ActionForward deleteCargo(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{

```

```

DynaActionForm dynaform=(DynaActionForm)form;

String idExcluirCargo=(String)dynaform.get("idCargosRemoverCombo");
String
senhaExclusao=Criptografia.retornaString(Criptografia.digest(((String)dynaform.get("
senha")).getBytes()));

HttpSession session = request.getSession(true);
Usuario user=(Usuario)session.getAttribute("usuario");
String destino= "resultadoSucesso";
UsuarioDTO userDTO=new UsuarioDTO();
userDTO.setLogin(user.getApelido());
userDTO.setSenha(senhaExclusao);

if (user != null) {
    if (user.getSenha().equals(senhaExclusao)) {
        try {
            CargoService cargoService =
(CargoService)getWebApplicationContext().getBean("cargoService");
            CargoDTO cargoDTO = new CargoDTO();
            cargoDTO.setId(Long.valueOf(idExcluirCargo));

            cargoService.delete(cargoDTO);

            request.setAttribute("titulo", "Cargo Excluído");
            request.setAttribute("mensagem", "Cargo excluído com sucesso.");

        } catch(Exception e) {
            request.setAttribute("mensagem", e.getMessage());
            destino = "erro";
        }
    } else {
        request.setAttribute("mensagem", "Senha inválida.");
        destino = "erro";
    }
}

```

```

    }
} else {
    request.setAttribute("mensagem", "Usuário não logado.");
    destino = "erro";
}
return mapping.findForward(destino);
}
public ActionForward getCargo(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{

    DynaActionForm dynaform=(DynaActionForm)form;
    try{
        String nomeCargo=(String)dynaform.get("cargoesconsulta");

        CargoService cargoService =
(CargoService)getWebApplicationContext().getBean("cargoService");
        CargoDTO cargoDTO = new CargoDTO();
        cargoDTO.setNome(nomeCargo);
        Collection listaCargosEncontrados=cargoService.select(cargoDTO);

        request.setAttribute("linhas",listaCargosEncontrados);

request.setAttribute("tamanhoLinhas",Integer.toString(listaCargosEncontrados.size())
);

        return mapping.findForward("resultadoConsulta");
    } catch(Exception e) {
        request.setAttribute("mensagem", e.getMessage());
        return mapping.findForward("erro");
    }
}
public ActionForward getCargos(ActionMapping mapping,
    ActionForm form,

```

```

        HttpServletRequest request,
        HttpServletResponse response) throws Exception{
    DynaActionForm dynaform=(DynaActionForm)form;
    try{
        CargoService cargoService =
(CargoService)getWebApplicationContext().getBean("cargoService");
        request.setAttribute("listaCargos",cargoService.getAll());

        return mapping.findForward("success");
    } catch(Exception e) {
        request.setAttribute("mensagem", e.getMessage());
        return mapping.findForward("erro");
    }
}
/**
 *@version 1.0
 *@method editCargo
 *@author David Gerson Ramos de Andrade
 *@description action criado para alteração de cargo.
 */
public ActionForward editCargo(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{
    DynaActionForm dynaform=(DynaActionForm)form;
    try{
        String nomeNovoCargo=(String)dynaform.get("cargoesedit");
        String idNovoCargo=(String)dynaform.get("idCargoeseditarCombo");

        CargoService cargoService =
(CargoService)getWebApplicationContext().getBean("cargoService");
        CargoDTO cargoDTO_Old = new CargoDTO();
        CargoDTO cargoDTO_New = new CargoDTO();
        cargoDTO_New.setNome(nomeNovoCargo);

```

```

        cargoDTO_Old.setId(Long.valueOf(idNovoCargo));
        cargoService.edit(cargoDTO_Old,cargoDTO_New);

        request.setAttribute("titulo", "Cargo Alterado");
        request.setAttribute("mensagem", "Cargo alterado com sucesso.");

        return mapping.findForward("resultadoSucesso");
    } catch(Exception e) {
        request.setAttribute("mensagem", e.getMessage());
        return mapping.findForward("erro");
    }
}
}
}

```

```

..\src\java\br\com\omd\ouvidoria\web\controle\action\CidadeAction.java
package br.com.omd.ouvidoria.web.controle.action;

```

```

import br.com.omd.ouvidoria.logica.domain.Cidade;
import br.com.omd.ouvidoria.logica.domain.Estado;
import br.com.omd.ouvidoria.logica.domain.Usuario;
import br.com.omd.ouvidoria.logica.domain.outros.OutraCidade;
import br.com.omd.ouvidoria.logica.service.CidadeService;
import br.com.omd.ouvidoria.util.Criptografia;
import java.util.List;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.DynaActionForm;

```

```

public class CidadeAction extends BaseAction{

```

```

public CidadeAction() {
}

public ActionForward defaultMethod(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{
    return montaDadosTela(mapping, form, request, response);
}

public ActionForward getCidades(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{
    DynaActionForm dynaform=(DynaActionForm)form;

    Long idEstado = (Long) dynaform.get("estado");

    CidadeService cidadeService =
(CidadeService)getWebApplicationContext().getBean("cidadeService");
    request.setAttribute("estados",cidadeService.getEstados());
    request.setAttribute("cidades",cidadeService.pesquisarPorEstado(idEstado));

request.setAttribute("estadosComCidades",cidadeService.getEstadosComCidades())
;
    request.setAttribute("idEstado", idEstado);

request.setAttribute("outrasCidades",cidadeService.getQuantidadePorOutroCidadeE
stado());

    return mapping.findForward("cidades");
}

public ActionForward montaDadosTela(ActionMapping mapping,
    ActionForm form,

```

```

        HttpServletRequest request,
        HttpServletResponse response) throws Exception{
        DynaActionForm dynaform=(DynaActionForm)form;

        CidadeService cidadeService =
(CidadeService)getWebApplicationContext().getBean("cidadeService");
        request.setAttribute("estados",cidadeService.getEstados());

request.setAttribute("estadosComCidades",cidadeService.getEstadosComCidades())
;

request.setAttribute("outrasCidades",cidadeService.getQuantidadePorOutroCidadeE
stado());

        return mapping.findForward("cidades");
    }

public ActionForward criaCidade(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{
        DynaActionForm dynaform=(DynaActionForm)form;

        String destino= "success";

        try {

            CidadeService cidadeService =
(CidadeService)getWebApplicationContext().getBean("cidadeService");
            Cidade cidade = new Cidade();
            cidade.setNome((String) dynaform.get("cidade"));

            if (cidade.getNome().equalsIgnoreCase("Outra")) {
                request.setAttribute("mensagem", "Cidade já cadastrada.");
                destino = "erro";
            } else {

```

```

        Estado estado = new Estado();
        estado.setld((Long) dynaform.get("estado"));
        cidade.setEstado(estado);

        cidadeService.save(cidade);

        request.setAttribute("titulo", "Cidade Cadastrada");
        request.setAttribute("mensagem", "Cidade cadastrada com sucesso.");
    }

} catch(Exception e) {
    request.setAttribute("mensagem", e.getMessage());
    destino = "erro";
}

return mapping.findForward(destino);
}

public ActionForward editaCidade(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{
    DynaActionForm dynaform=(DynaActionForm)form;

    String destino= "success";

    try {

        CidadeService cidadeService =
(CidadeService)getWebApplicationContext().getBean("cidadeService");
        Cidade cidade = new Cidade();
        cidade.setNome((String) dynaform.get("cidade"));
        cidade.setld((Long) dynaform.get("id"));
    }
}

```

```

    if (cidade.getNome().equalsIgnoreCase("Outra")) {
        request.setAttribute("mensagem", "Cidade já cadastrada.");
        destino = "erro";
    } else {
        Estado estado = new Estado();
        estado.setId((Long) dynaform.get("estado"));
        cidade.setEstado(estado);

        cidadeService.save(cidade);

        request.setAttribute("titulo", "Cidade Atualizada");
        request.setAttribute("mensagem", "Cidade atualizada com sucesso.");
    }

} catch(Exception e) {
    request.setAttribute("mensagem", e.getMessage());
    destino = "erro";
}

return mapping.findForward(destino);
}

public ActionForward consultaCidade(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{
    DynaActionForm dynaform=(DynaActionForm)form;

    Long idEstado = (Long) dynaform.get("estado");
    String nomeCidade = (String) dynaform.get("cidade");

    Cidade cidade = new Cidade();
    cidade.setNome(nomeCidade);

```

```

if (idEstado.longValue() > 0) {
    Estado estado = new Estado();
    estado.setId(idEstado);
    cidade.setEstado(estados);
}
String destino = "resultadoConsulta";
try {
    CidadeService cidadeService =
(CidadeService)getWebApplicationContext().getBean("cidadeService");
    List cidades = cidadeService.pesquisarCidades(cidade);
    request.setAttribute("cidades", cidades);
    request.setAttribute("numLinhas", "" + cidades.size());
} catch (Exception e){
    destino = "erro";
    request.setAttribute("mensagem",e.getMessage());
}

return mapping.findForward(destino);
}

```

```

public ActionForward excluiCidade(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{
    DynaActionForm dynaform=(DynaActionForm)form;

    Usuario usuario = (Usuario)request.getSession(false).getAttribute("usuario");
    String senha =
Criptografia.retornaString(Criptografia.digest(((String)dynaform.get("senhaConferenci
a")).getBytes()));
    String destino= "success";

    if (usuario != null) {
        if (usuario.getSenha().equals(senha)) {
            try {

```

```

        CidadeService cidadeService =
(CidadeService)getWebApplicationContext().getBean("cidadeService");
        Cidade cidade = new Cidade();
        cidade.setNome((String) dynaform.get("cidade"));
        cidade.setId((Long) dynaform.get("id"));

        Estado estado = new Estado();
        estado.setId((Long) dynaform.get("estado"));
        cidade.setEstado(estado);

        cidadeService.delete(cidade);

        request.setAttribute("titulo", "Cidade Excluída");
        request.setAttribute("mensagem", "Cidade excluída com sucesso.");

    } catch(Exception e) {
        request.setAttribute("mensagem", e.getMessage());
        destino = "erro";
    }
    } else {
        request.setAttribute("mensagem", "Senha inválida.");
        destino = "erro";
    }
    } else {
        request.setAttribute("mensagem", "Usuário não logado.");
        destino = "erro";
    }
    }

    return mapping.findForward(destino);
}

public ActionForward outraCidade(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,

```

```
    HttpServletResponse response) throws Exception{
    DynaActionForm dynaform=(DynaActionForm)form;

    String destino= "success";

    try {

        CidadeService cidadeService =
(CidadeService)getWebApplicationContext().getBean("cidadeService");
        Cidade cidade = new Cidade();
        cidade.setNome((String) dynaform.get("cidade"));
        cidade.setId((Long) dynaform.get("id"));

        Estado estado = new Estado();
        estado.setId((Long) dynaform.get("estado"));
        cidade.setEstado(estado);

        String nomeOutraCidade = (String) dynaform.get("outraCidade");
        Long idOutroEstado =
Long.valueOf(nomeOutraCidade.substring(nomeOutraCidade.indexOf("|")+1));
        nomeOutraCidade = nomeOutraCidade.substring(0,
nomeOutraCidade.indexOf("|"));

        OutraCidade outraCidade = new OutraCidade();
        outraCidade.setNome(nomeOutraCidade);

        Estado outroEstado = new Estado();
        outroEstado.setId(idOutroEstado);
        outraCidade.setEstado(outroEstado);

        cidadeService.substituirOutraCidadePorCidade(outraCidade, cidade);

        request.setAttribute("titulo", "Outra(s) Cidade(s) Substituída(s)");
        request.setAttribute("mensagem", "Outra(s) Cidade(s) substituída(s) com
sucesso.");
    }
}
```

```

    } catch(Exception e) {
        request.setAttribute("mensagem", e.getMessage());
        destino = "erro";
    }

    return mapping.findForward(destino);
}

}

..\src\java\br\com\omd\ouvidoria\web\controle\action\ConsultaAction.java
/*
 * ConsultaAction.java
 *
 */

package br.com.omd.ouvidoria.web.controle.action;

import br.com.omd.ouvidoria.logica.dto.ConsultaManifestacaoDTO;
import br.com.omd.ouvidoria.logica.service.CidadeService;
import br.com.omd.ouvidoria.logica.service.ManifestacaoService;
import java.text.SimpleDateFormat;
import java.util.HashMap;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.DynaActionForm;

public class ConsultaAction extends BaseAction {

    /** Creates a new instance of ConsultaAction */

```

```
public ConsultaAction() {  
}
```

```
public ActionForward defaultMethod(ActionMapping mapping,  
    ActionForm form,  
    HttpServletRequest request,  
    HttpServletResponse response) throws Exception{  
    return montaDadosTela(mapping, form, request, response);  
}
```

```
public ActionForward montaDadosTela(ActionMapping mapping,  
    ActionForm form,  
    HttpServletRequest request,  
    HttpServletResponse response) throws Exception{  
    DynaActionForm dynaform=(DynaActionForm)form;
```

```
    ManifestacaoService manifestacaoService =  
(ManifestacaoService)getWebApplicationContext().getBean("manifestacaoService");  
    HashMap dadosTela = manifestacaoService.getDadosConsulta();
```

```
    request.setAttribute("estados", dadosTela.get("estados"));  
    request.setAttribute("assuntos", dadosTela.get("assuntos"));  
    request.setAttribute("classificacoes", dadosTela.get("classificacoes"));  
    request.setAttribute("unidades", dadosTela.get("unidades"));
```

```
    return mapping.findForward("consulta");  
}
```

```
public ActionForward getCidades(ActionMapping mapping,  
    ActionForm form,  
    HttpServletRequest request,  
    HttpServletResponse response) throws Exception{  
    DynaActionForm dynaform=(DynaActionForm)form;
```

```
    String idEstado = dynaform.getString("idEstado");
```

```

ManifestacaoService manifestacaoService =
(ManifestacaoService)getWebApplicationContext().getBean("manifestacaoService");
    CidadeService cidadeService =
(CidadeService)getWebApplicationContext().getBean("cidadeService");
    HashMap dadosTela = manifestacaoService.getDadosConsulta();
    if (!idEstado.equals("")) {
        request.setAttribute("cidades",
cidadeService.pesquisarPorEstado(Long.valueOf(idEstado)));
    }

```

```

//Seta valores selecionados na tela

```

```

request.setAttribute("codigo", (String) dynaform.get("codigo"));
request.setAttribute("nome", (String) dynaform.get("nome"));
request.setAttribute("bairro", (String) dynaform.get("bairro"));
request.setAttribute("idEstado", (String) dynaform.get("idEstado"));
request.setAttribute("idCidade", (String) dynaform.get("idCidade"));
request.setAttribute("cep", (String) dynaform.get("cep"));
request.setAttribute("datainicio", (String) dynaform.get("datainicio"));
request.setAttribute("datafim", (String) dynaform.get("datafim"));
request.setAttribute("idAssunto", (String) dynaform.get("idAssunto"));
request.setAttribute("idClassificacao", (String) dynaform.get("idClassificacao"));
request.setAttribute("idUnidade", (String) dynaform.get("idUnidade"));

```

```

//Popula combos

```

```

request.setAttribute("estados", dadosTela.get("estados"));
request.setAttribute("assuntos", dadosTela.get("assuntos"));
request.setAttribute("classificacoes", dadosTela.get("classificacoes"));
request.setAttribute("unidades", dadosTela.get("unidades"));

```

```

return mapping.findForward("consulta");

```

```

}

```

```

public ActionForward resultadoConsulta(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{

```

```

DynaActionForm dynaform=(DynaActionForm)form;

ConsultaManifestacaoDTO dto = new ConsultaManifestacaoDTO();
String aux = "";

//    dto.setId(Long.valueOf(dynaform.getString("id")));
    dto.setNomeManifestante(dynaform.getString("nome"));
    dto.setBairro(dynaform.getString("bairro"));

    aux = dynaform.getString("idEstado");
    if (!aux.equals(""))
        dto.setIdEstado(Long.valueOf(aux));

    aux = dynaform.getString("idCidade");
    if (!aux.equals(""))
        dto.setIdCidade(Long.valueOf(aux));

    dto.setCep(dynaform.getString("cep"));

    aux = dynaform.getString("datainicio");
    if (!aux.equals("")) {
        SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");
        dto.setDataInicio(sdf.parse(dynaform.getString("datainicio")));
    }

    aux = dynaform.getString("datafim");
    if (!aux.equals("")) {
        SimpleDateFormat sdfHora = new
SimpleDateFormat("dd/MM/yyyyhh:mm:ss");
        dto.setDataFim(sdfHora.parse(dynaform.getString("datafim")+"23:59:59"));
    }

    aux = dynaform.getString("idAssunto");
    if (!aux.equals(""))
        dto.setIdAssunto(Long.valueOf(aux));

```

```

        aux = dynaform.getString("idClassificacao");
        if (!aux.equals(""))
            dto.setIdClassificacao(Long.valueOf(aux));

        aux = dynaform.getString("idUnidade");
        if (!aux.equals(""))
            dto.setIdUnidade(Long.valueOf(aux));

        ManifestacaoService manifestacaoService =
(ManifestacaoService)getWebApplicationContext().getBean("manifestacaoService");
        request.setAttribute("resultado", manifestacaoService.consultar(dto));

        return mapping.findForward("resultadoConsulta");
    }
}

```

```

..\src\java\br\com\omd\ouvidoria\web\controle\action\FeriadoAction.java
package br.com.omd.ouvidoria.web.controle.action;

```

```

import br.com.omd.ouvidoria.logica.domain.Feriado;
import br.com.omd.ouvidoria.logica.domain.Usuario;
import br.com.omd.ouvidoria.logica.dto.UsuarioDTO;
import br.com.omd.ouvidoria.logica.service.FeriadoService;
import br.com.omd.ouvidoria.logica.service.UsuarioService;
import br.com.omd.ouvidoria.util.Criptografia;
import java.text.SimpleDateFormat;
import java.util.Collection;
import java.util.Date;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

```

```

import org.apache.struts.action.DynaActionForm;

public class FeriadoAction extends BaseAction{

    public FeriadoAction() {
    }

    public ActionForward defaultMethod(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{
        return getFeriados(mapping, form, request, response);
    }
    /**
    *@version 1.0
    *@method addFeriado
    *@author David Gerson Ramos de Andrade
    *@description action criado para inserção de novo feriado.
    */
    public ActionForward addFeriado(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{

        DynaActionForm dynaform=(DynaActionForm)form;
        try{
            String novoFeriado=(String)dynaform.get("feriadonaovo");

            FeriadoService feriadoService =
(FeriadoService)getWebApplicationContext().getBean("feriadoService");
            Feriado feriado = new Feriado();
            Date dataFeriado = new

```

```

SimpleDateFormat("dd/MM/yyyy").parse(novoFeriado);

        feriado.setData(dataFeriado);

        feriadoService.save(feriado);

        request.setAttribute("titulo", "Feriado Adicionado");
        request.setAttribute("mensagem", "Feriado adicionado com sucesso.");

        return mapping.findForward("resultadoSucesso");
    } catch (Exception e) {
        request.setAttribute("mensagem", e.getMessage());
        return mapping.findForward("erro");
    }
}
/**
 * @version 1.0
 * @method deleteFeriado
 * @author David Gerson Ramos de Andrade
 * @description action criado para remoção de novo feriado.
 */
public ActionForward deleteFeriado(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{

    DynaActionForm dynaform=(DynaActionForm)form;

    String idExcluirFeriado=(String)dynaform.get("idFeriadosRemoverCombo");
    String
senhaExclusao=Criptografia.retornaString(Criptografia.digest(((String)dynaform.get("
senha")).getBytes()));

    HttpSession session = request.getSession(true);

```

```

    Usuario user=(Usuario)session.getAttribute("usuario");

    UsuarioDTO userDTO=new UsuarioDTO();
    userDTO.setLogin(user.getApellido());
    userDTO.setSenha(senhaExclusao);

    try {

        UsuarioService usuarioService =
        (UsuarioService)getWebApplicationContext().getBean("usuarioService");
        Usuario usuario =usuarioService.autenticar(userDTO);

        FeriadoService feriadoService =
        (FeriadoService)getWebApplicationContext().getBean("feriadoService");
        Feriado feriado = new Feriado();

        feriado.setId(Long.valueOf(idExcluirFeriado));

        feriadoService.delete(feriado);

        request.setAttribute("titulo", "Feriado Excluído");
        request.setAttribute("mensagem", "Feriado excluído com sucesso.");

        return mapping.findForward("resultadoSucesso");
    } catch (Exception e ){
        request.setAttribute("erro", new Boolean(true));
        return mapping.findForward("errologin");
    }
}

public ActionForward getFeriadoPorPeriodo(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{

```

```

DynaActionForm dynaform=(DynaActionForm)form;
try{
    String inicioPeriodoFeriado=(String)dynaform.get("datainicio");
    String fimPeriodoFeriado=(String)dynaform.get("datafim");

    FeriadoService feriadoService =
(FeriadoService)getWebApplicationContext().getBean("feriadoService");

    Feriado feriadoInicial = new Feriado();
    if (!(inicioPeriodoFeriado.equals(""))){
        Date dataInicioFeriado = new
SimpleDateFormat("dd/MM/yyyy").parse(inicioPeriodoFeriado);
        feriadoInicial.setData(dataInicioFeriado);
    }
    Feriado feriadoFinal = new Feriado();
    if (!(fimPeriodoFeriado.equals(""))){
        Date dataFimFeriado = new
SimpleDateFormat("dd/MM/yyyy").parse(fimPeriodoFeriado);
        feriadoFinal.setData(dataFimFeriado);
    }
    Collection
listaFeriadosEncontrados=feriadoService.select(feriadoInicial,feriadoFinal);

    request.setAttribute("linhas",listaFeriadosEncontrados);

request.setAttribute("tamanhoLinhas",Integer.toString(listaFeriadosEncontrados.size(
)));

    return mapping.findForward("resultadoConsulta");
} catch(Exception e) {
    request.setAttribute("mensagem", e.getMessage());
    return mapping.findForward("erro");
}
}

```

```

public ActionForward getFeriados(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{
    DynaActionForm dynaform=(DynaActionForm)form;
    try{
        FeriadoService feriadoService =
(FeriadoService)getWebApplicationContext().getBean("feriadoService");
        request.setAttribute("feriados",feriadoService.getAll());

        return mapping.findForward("success");
    } catch(Exception e) {
        request.setAttribute("mensagem", e.getMessage());
        return mapping.findForward("erro");
    }
}
/**
 *@version 1.0
 *@method editFeriado
 *@author David Gerson Ramos de Andrade
 *@description action criado para alteração de feriado.
 */
public ActionForward editFeriado(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{
    DynaActionForm dynaform=(DynaActionForm)form;
    try{
        String dataNovaFeriado=(String)dynaform.get("feriadoalterado");
        String idOldFeriado=(String)dynaform.get("idFeriadosEditarCombo");

        FeriadoService feriadoService =
(FeriadoService)getWebApplicationContext().getBean("feriadoService");
        Feriado feriado_Old = new Feriado();

```

```

        Feriado feriado_New = new Feriado();

        Date dataInicio = new
SimpleDateFormat("dd/MM/yyyy").parse(dataNovaFeriado);

        feriado_New.setData(dataInicio);

        feriado_Old.setId(Long.valueOf(idOldFeriado));

        feriadoService.edit(feriado_Old,feriado_New);

        request.setAttribute("titulo", "Feriado Alterado");
        request.setAttribute("mensagem", "Feriado alterado com sucesso.");

        return mapping.findForward("resultadoSucesso");
    } catch(Exception e) {
        request.setAttribute("mensagem", e.getMessage());
        return mapping.findForward("erro");
    }
}
}
}

```

```

..\src\java\br\com\omd\ouvidoria\web\controle\action>LoginAction.java
package br.com.omd.ouvidoria.web.controle.action;

```

```

import br.com.omd.ouvidoria.logica.domain.Usuario;
import br.com.omd.ouvidoria.logica.dto.UsuarioDTO;
import br.com.omd.ouvidoria.logica.service.UsuarioService;
import br.com.omd.ouvidoria.util.Criptografia;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

```

```

import org.apache.struts.action.DynaActionForm;

public class LoginAction extends BaseAction {

    public LoginAction() {
    }

    public ActionForward defaultMethod(ActionMapping mapping, ActionForm form,
    HttpServletRequest request, HttpServletResponse response) throws Exception {
        DynaActionForm dynaform=(DynaActionForm)form;
        String login = (String)dynaform.get("login");
        String senha =
        Criptografia.retornaString(Criptografia.digest(((String)dynaform.get("senha")).getBytes()));

        UsuarioDTO usuarioDTO = new UsuarioDTO();

        usuarioDTO.setLogin(login);
        usuarioDTO.setSenha(senha);

        UsuarioService usuarioService =
        (UsuarioService)getWebApplicationContext().getBean("usuarioService");
        String destino;
        try {
            Usuario usuario = usuarioService.autenticar(usuarioDTO);
            HttpSession session = request.getSession(true);
            session.setAttribute("usuario", usuario);
            destino = "success";
        } catch (Exception e ){
            e.printStackTrace();
            destino = "errologin";
            request.setAttribute("erro", new Boolean(true));
        }

        return mapping.findForward(destino);
    }
}

```

```
}
```

```
..\src\java\br\com\omd\ouvidoria\web\controle\action\MenuAction.java  
package br.com.omd.ouvidoria.web.controle.action;
```

```
import br.com.omd.ouvidoria.logica.domain.Unidade;  
import br.com.omd.ouvidoria.logica.domain.Usuario;  
import br.com.omd.ouvidoria.logica.service.CaixaManifestacaoService;  
import br.com.omd.ouvidoria.logica.service.RelatorioService;  
import br.com.omd.ouvidoria.logica.service.UnidadeService;  
import java.util.Collection;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import javax.servlet.http.HttpSession;  
import org.apache.struts.action.ActionForm;  
import org.apache.struts.action.ActionForward;  
import org.apache.struts.action.ActionMapping;  
import org.apache.struts.action.DynaActionForm;
```

```
public class MenuAction extends BaseAction{
```

```
    public MenuAction() {  
    }  
}
```

```
    public ActionForward defaultMethod(ActionMapping mapping,  
        ActionForm form,  
        HttpServletRequest request,  
        HttpServletResponse response) throws Exception{  
        DynaActionForm dynaform=(DynaActionForm)form;  
  
        HttpSession session = request.getSession(true);  
        Usuario user=(Usuario)session.getAttribute("usuario");
```

```
CaixaManifestacaoService caixaManifestacaoService =
(CaixaManifestacaoService)getWebApplicationContext().getBean("caixaManifestacaoService");
```

```
request.setAttribute("manifestacoesPorSituacao",caixaManifestacaoService.contaManifestacoesPorSituacao(user));
```

```
if (user.getPerfil().getId().longValue() == 1) {
    Collection colecao = caixaManifestacaoService.getAguardandoResposta(null, null);
```

```
request.setAttribute("aguardandoResposta", colecao);
request.setAttribute("tamanhoAguardandoResposta",new Integer(colecao.size()));
```

```
colecao = caixaManifestacaoService.getARevisar(null, null);
request.setAttribute("aRevisar", colecao);
request.setAttribute("tamanhoARevisar",new Integer(colecao.size()));
```

```
colecao = caixaManifestacaoService.getEncerradas(null, user);
request.setAttribute("encerradas", colecao);
request.setAttribute("tamanhoEncerradas",new Integer(colecao.size()));
```

```
}
```

```
RelatorioService relatorioService =
(RelatorioService)getWebApplicationContext().getBean("relatorioService");
Collection relatorios = relatorioService.getAll();
request.setAttribute("relatorios",relatorios);
request.setAttribute("tamanhoRelatorios",new Integer(relatorios.size()));
```

```
Collection unidades=relatorioService.getUnidadesPorUsuario(user);
request.setAttribute("unidades",unidades);
```

```
String unidVinculada=(String)dynaform.get("unidade");
if(!(unidVinculada.equals(""))){
    Unidade unit=new Unidade();
    unit.setId(Long.valueOf(unidVinculada));
```

```

        Collection
        unidadesVinculadas=relatorioService.getUnidadesVinculadasPorUnidade(unit);
        request.setAttribute("unidadesVinculadas",unidadesVinculadas);
    }

    return mapping.findForward("success");
}
}
}

```

```

..\src\java\br\com\omd\ouvidoria\web\controle\action\notificacaoAction.java

```

```

/*

```

```

 * NotificacaoAction.java

```

```

 *

```

```

*/

```

```

package br.com.omd.ouvidoria.web.controle.action;

```

```

import br.com.omd.ouvidoria.logica.domain.Notificacao;

```

```

import br.com.omd.ouvidoria.logica.service.NotificacaoService;

```

```

import javax.servlet.http.HttpServletRequest;

```

```

import javax.servlet.http.HttpServletResponse;

```

```

import org.apache.struts.action.ActionForm;

```

```

import org.apache.struts.action.ActionForward;

```

```

import org.apache.struts.action.ActionMapping;

```

```

public class NotificacaoAction extends BaseAction {

```

```

    /** Creates a new instance of NotificacaoAction */

```

```

    public NotificacaoAction() {

```

```

    }

```

```

    public ActionForward defaultMethod(ActionMapping mapping,

```

```

        ActionForm form,

```

```

        HttpServletRequest request,

```

```

        HttpServletResponse response) throws Exception {

```

```
    NotificacaoService notificacaoService =
    (NotificacaoService)getWebApplicationContext().getBean("notificacaoService");
    request.setAttribute("notificacoes",
    notificacaoService.getAll(notificacaoService.ORDENAR_POR_DESCRICAO));
    return mapping.findForward("inicial");
}
```

```
public ActionForward atualizaHabilitado(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception {

    String destino = "success";

    try {
        NotificacaoService notificacaoService =
        (NotificacaoService)getWebApplicationContext().getBean("notificacaoService");

        Notificacao notificacao = new Notificacao();
        notificacao.setId(new Long(1));
        if (request.getParameter("1") != null) {
            notificacao.setHabilitado(true);
        } else {
            notificacao.setHabilitado(false);
        }

        notificacaoService.atualizaHabilitado(notificacao);

        request.setAttribute("titulo", "Notificação Atualizada");
        request.setAttribute("mensagem", "Notificação atualizada com sucesso.");

    } catch(Exception e) {
        request.setAttribute("mensagem", e.getMessage());
        destino = "erro";
    }
}
```

```

    }

    return mapping.findForward(destino);
}
}

..\src\java\br\com\omd\ouvidoria\web\controle\action\PrazoAction.java
package br.com.omd.ouvidoria.web.controle.action;

import br.com.omd.ouvidoria.logica.dto.PrazoDTO;
import br.com.omd.ouvidoria.logica.service.PrazoService;
import java.util.Collection;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.DynaActionForm;

public class PrazoAction extends BaseAction{

    public PrazoAction() {
    }

    public ActionForward defaultMethod(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{
        return getPrazos(mapping, form, request, response);
    }

    public ActionForward getPrazos(ActionMapping mapping,

```

```

        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{
    DynaActionForm dynaform=(DynaActionForm)form;

    PrazoService prazoService =
(PrazoService)getWebApplicationContext().getBean("prazoService");
    request.setAttribute("prazos",prazoService.getAll());

    return mapping.findForward("success");
}
/**
 *@version 1.0
 *@method editPrazo
 *@author David Gerson Ramos de Andrade
 *@description action criado para alteração de prazo.
 */
public ActionForward editPrazo(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{
    DynaActionForm dynaform=(DynaActionForm)form;
    try{
        String novoPrazo=(String)dynaform.get("prazoedit");
        String idNovoPrazo=(String)dynaform.get("idPrazo");

        PrazoService prazoService =
(PrazoService)getWebApplicationContext().getBean("prazoService");
        PrazoDTO prazoDTO = new PrazoDTO();

        prazoDTO.setPrazo(Integer.parseInt(novoPrazo));
        prazoDTO.setId(Long.valueOf(idNovoPrazo));
        prazoService.edit(prazoDTO);
    }
}

```

```
        request.setAttribute("titulo", "Prazo Alterado");
        request.setAttribute("mensagem", "Prazo alterado com sucesso.");

        return mapping.findForward("resultadoSucesso");
    } catch(Exception e) {
        request.setAttribute("mensagem", e.getMessage());
        return mapping.findForward("erro");
    }
}
}
```

```
..\src\java\br\com\omd\ouvidoria\web\controle\action\UnidadeAction.java
package br.com.omd.ouvidoria.web.controle.action;
```

```
import br.com.omd.ouvidoria.logica.domain.Cidade;
import br.com.omd.ouvidoria.logica.domain.Estado;
import br.com.omd.ouvidoria.logica.domain.Unidade;
import br.com.omd.ouvidoria.logica.domain.Usuario;
import br.com.omd.ouvidoria.logica.domain.outros.OutraUnidade;
import br.com.omd.ouvidoria.logica.dto.UsuarioDTO;
import br.com.omd.ouvidoria.logica.service.CidadeService;
import br.com.omd.ouvidoria.logica.service.OutraUnidadeService;
import br.com.omd.ouvidoria.logica.service.UnidadeService;
import br.com.omd.ouvidoria.logica.service.UsuarioService;
import br.com.omd.ouvidoria.util.Criptografia;
import java.util.ArrayList;
import java.util.Collection;
import java.util.Iterator;
import java.util.List;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
```

```

import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.DynaActionForm;

public class UnidadeAction extends BaseAction{

    public UnidadeAction() {
    }

    public ActionForward defaultMethod(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{
        return getUnidades(mapping, form, request, response);
    }

    /**
     *@version 1.0
     *@method addUnidade
     *@author David Gerson Ramos de Andrade
     *@description action criado para inserção de novo unidade.
     */
    public ActionForward addUnidade(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{
        DynaActionForm dynaform=(DynaActionForm)form;

        String nomeNovoUnidade=(String)dynaform.get("nomeUnidade");
        String siglaNovoUnidade=(String)dynaform.get("siglaUnidade");
        String
idComboRepresentantesNovoUnidade=(String)dynaform.get("idComboRepresentant
es");
        String
idComboVinculosNovoUnidade=(String)dynaform.get("idComboVinculos");

```

```

String idComboCidadeNovoUnidade=(String)dynaform.get("idComboCidades");
String idComboEstadoNovoUnidade=(String)dynaform.get("idComboEstados");
String enderecoNovoUnidade=(String)dynaform.get("enderecoUnidade");
String bairroNovoUnidade=(String)dynaform.get("bairroUnidade");
String cepNovoUnidade=(String)dynaform.get("cepUnidade");
String dddUmNovoUnidade=(String)dynaform.get("dddUmUnidade");
String telUmNovoUnidade=(String)dynaform.get("telUmUnidade");
String dddDoisNovoUnidade=(String)dynaform.get("dddDoisUnidade");
String telDoisNovoUnidade=(String)dynaform.get("telDoisUnidade");
String dddNovoUnidade=(String)dynaform.get("dddFaxUnidade");
String faxNovoUnidade=(String)dynaform.get("faxUnidade");
String emailNovoUnidade=(String)dynaform.get("emailUnidade");

try {
    UnidadeService unidadeService =
(UnidadeService)getWebApplicationContext().getBean("unidadeService");
    Unidade unidade = new Unidade();
    unidade.setNome(nomeNovoUnidade);
    unidade.setSigla(siglaNovoUnidade);
    Usuario responsavel=new Usuario();
    Long idResponsavel=Long.valueOf(idComboRepresentantesNovoUnidade);
    responsavel.setld(idResponsavel);
    unidade.setResponsavel(responsavel);
    Unidade vinculo=new Unidade();
    Long idVinculo=Long.valueOf(idComboVinculosNovoUnidade);
    vinculo.setld(idVinculo);
    unidade.setVinculo(vinculo);

    unidade.setBairro(bairroNovoUnidade);

    /*Cidade e estado*/
    if (!idComboEstadoNovoUnidade.equals("0")){
        if (!(idComboCidadeNovoUnidade.equals(""))){
            Cidade cidadeUnidade=new Cidade();

```

```

        Long idCidade=Long.valueOf(idComboCidadeNovoUnidade);
        cidadeUnidade.setId(idCidade);
        unidade.setCidade(cidadeUnidade);
    }
}else{
    unidade.setCidade(null);
}
unidade.setCep(cepNovoUnidade);
unidade.setDddTelefone1(dddUmNovoUnidade);
unidade.setTelefone1(telUmNovoUnidade);
unidade.setDddTelefone2(dddDoisNovoUnidade);
unidade.setTelefone2(telDoisNovoUnidade);
unidade.setDddFax(dddNovoUnidade);
unidade.setFax(faxNovoUnidade);
unidade.setEmail(emailNovoUnidade);

unidade.setEndereco(enderecoNovoUnidade);
unidadeService.save(unidade);

request.setAttribute("titulo", "Unidade Cadastrada");
request.setAttribute("mensagem", "Unidade cadastrada com sucesso.");

return mapping.findForward("resultadoSucesso");
} catch(Exception e) {
    request.setAttribute("mensagem", e.getMessage());
    return mapping.findForward("erro");
}
}

/**
 *@version 1.0
 *@method deleteUnidade
 *@author David Gerson Ramos de Andrade
 *@description action criado para remoção de novo unidade.

```

```

*/
public ActionForward deleteUnidade(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{

    DynaActionForm dynaform=(DynaActionForm)form;

    String idExcluirUnidade=(String)dynaform.get("idComboUnidadeExcluir");
    String
senhaExclusao=Criptografia.retornaString(Criptografia.digest(((String)dynaform.get("
senha")).getBytes()));

    HttpSession session = request.getSession(true);
    Usuario user=(Usuario)session.getAttribute("usuario");
    String destino= "resultadoSucesso";
    UsuarioDTO userDTO=new UsuarioDTO();
    userDTO.setLogin(user.getApelido());
    userDTO.setSenha(senhaExclusao);

    if (user != null) {
        if (user.getSenha().equals(senhaExclusao)) {
            try {
                UsuarioService usuarioService =
(UsuarioService)getWebApplicationContext().getBean("usuarioService");

                UnidadeService unidadeService =
(UnidadeService)getWebApplicationContext().getBean("unidadeService");
                Unidade unidadeDelete = new Unidade();

                unidadeDelete.setld(Long.valueOf(idExcluirUnidade));

                unidadeService.delete(unidadeDelete);

```

```

        request.setAttribute("titulo", "Unidade Excluída");
        request.setAttribute("mensagem", "Unidade excluída com sucesso.");

    } catch(Exception e) {
        request.setAttribute("mensagem", e.getMessage());
        destino = "erro";
    }
} else {
    request.setAttribute("mensagem", "Senha inválida.");
    destino = "erro";
}
} else {
    request.setAttribute("mensagem", "Usuário não logado.");
    destino = "erro";
}
return mapping.findForward(destino);
}

```

```

public ActionForward exhibeUnidade(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{
    try{
        Long idLong=Long.valueOf((String)request.getParameter("id"));

        UnidadeService unidadeService =
(UnidadeService)getWebApplicationContext().getBean("unidadeService");
        Unidade unidade = new Unidade();
        unidade.setId(idLong);
        Unidade unidadeEncontrada=(Unidade)unidadeService.selectPorId(unidade);

        request.setAttribute("unidadeNome",unidadeEncontrada.getNome());
        request.setAttribute("unidadeSigla",unidadeEncontrada.getSigla());
        request.setAttribute("unidadeEndereco",unidadeEncontrada.getEndereco());
        request.setAttribute("unidadeBairro",unidadeEncontrada.getBairro());
    }
}

```

```

        request.setAttribute("unidadeCEP",unidadeEncontrada.getCep());
        if(unidadeEncontrada.getResponsavel()!=null){

request.setAttribute("unidadeRepresentante",unidadeEncontrada.getResponsavel().g
etNome());
        }
        if(unidadeEncontrada.getVinculo()!=null){

request.setAttribute("unidadeVínculo",unidadeEncontrada.getVinculo().getNome());
        }
        request.setAttribute("unidadeTelefone1" ,
unidadeEncontrada.getTelefone1());
        request.setAttribute("unidadeTelefone2", unidadeEncontrada.getTelefone2());
        request.setAttribute("unidadeFax",unidadeEncontrada.getDddFax());
        if( unidadeEncontrada.getCidade()!=null){
            request.setAttribute("unidadeCidade",
unidadeEncontrada.getCidade().getNome());
            request.setAttribute("unidadeEstado",
unidadeEncontrada.getCidade().getEstado().getSigla());
        }
        request.setAttribute("unidadeEmail" ,unidadeEncontrada.getEmail());
        return mapping.findForward("exibicao");

    } catch(Exception e) {
        request.setAttribute("mensagem", e.getMessage());
        return mapping.findForward("erro");
    }
}

public ActionForward getUnidade(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{

    DynaActionForm dynaform=(DynaActionForm)form;
    try{

```

```

String nomeUnidade=(String)dynaform.get("consultaUnidade");

UnidadeService unidadeService =
(UnidadeService)getWebApplicationContext().getBean("unidadeService");
Unidade unidade = new Unidade();
unidade.setNome(nomeUnidade);

List listaUnidadesEncontrados=(List) unidadeService.select(unidade);

request.setAttribute("listaUnidadesEncontrados",listaUnidadesEncontrados);

request.setAttribute("tamanhoLinhas",Integer.toString(listaUnidadesEncontrados.size
()));

return mapping.findForward("resultadoConsulta");

} catch(Exception e) {
request.setAttribute("mensagem", e.getMessage());
return mapping.findForward("erro");
}
}

public ActionForward getUnidades(ActionMapping mapping,
ActionForm form,
HttpServletRequest request,
HttpServletResponse response) throws Exception{
DynaActionForm dynaform=(DynaActionForm)form;
try{
UsuarioService usuarioService =
(UsuarioService)getWebApplicationContext().getBean("usuarioService");
request.setAttribute("listaRepresentantes",usuarioService.getAll());

UnidadeService unidadeService =
(UnidadeService)getWebApplicationContext().getBean("unidadeService");
List vinculoSemPrimeiro=(List) unidadeService.getAll();
List vinculoSemOutrosNenhum=new ArrayList();

```

```

        Iterator it2 =vinculoSemPrimeiro.iterator();
        while (it2.hasNext()){
            Unidade unit=(Unidade)it2.next();
            if (!(unit.getNome().equals("Outra"))) &&
(!unit.getNome().equals("Nenhuma"))){
                vinculoSemOutrosNenhum.add(unit);
            }
        }
        request.setAttribute("listaVinculos",vinculoSemOutrosNenhum);

        CidadeService cidadeService =
(CidadeService)getWebApplicationContext().getBean("cidadeService");

        request.setAttribute("listaEstados",cidadeService.getEstadosComCidades());

        request.setAttribute("listaOutrasUnidades",
unidadeService.getQuantidadePorOutro());

        List unidadeSemPrimeiro=(List) unidadeService.getAll();
        List unidadesSemOutrosNenhum=new ArrayList();
        Iterator it =unidadeSemPrimeiro.iterator();
        while (it.hasNext()){
            Unidade unit=(Unidade)it.next();
            if (!(unit.getNome().equals("Outra"))) &&
(!unit.getNome().equals("Nenhuma"))){
                unidadesSemOutrosNenhum.add(unit);
            }
        }

        request.setAttribute("listaUnidades",unidadesSemOutrosNenhum);

        return mapping.findForward("success");
    } catch(Exception e) {
        request.setAttribute("mensagem", e.getMessage());
    }

```

```

        return mapping.findForward("erro");
    }
}

public ActionForward changeUnidades(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{
    DynaActionForm dynaform=(DynaActionForm)form;
    try{
        String idPadraoUnidade=(String)dynaform.get("idComboUnidadesTroca");
        String idOutraUnidade=(String)dynaform.get("idComboOutrasUnidades");

        Unidade unidadePadrao=new Unidade();
        OutraUnidade outraUnidade=new OutraUnidade();

        unidadePadrao.setId(Long.valueOf(idPadraoUnidade));
        outraUnidade.setNome(idOutraUnidade);

        UnidadeService unidadeService =
(UnidadeService)getWebApplicationContext().getBean("unidadeService");

        unidadeService.changeUnidadePorOutraUnidade(outraUnidade,unidadePadrao);

        request.setAttribute("titulo", "Unidade Trocada");
        request.setAttribute("mensagem", "Unidade trocada com sucesso.");

        return mapping.findForward("resultadoSucesso");

    } catch(Exception e) {
        request.setAttribute("mensagem", e.getMessage());
        return mapping.findForward("erro");
    }
}

```

```

/**
 *@version 1.0
 *@method editUnidade
 *@author David Gerson Ramos de Andrade
 *@description action criado para alteração de unidade.
 */
public ActionForward editUnidade(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{
    DynaActionForm dynaform=(DynaActionForm)form;

    try{

        String nomeEdicaoUnidade=(String)dynaform.get("nomeUnidade");
        String siglaEdicaoUnidade=(String)dynaform.get("siglaUnidade");
        String
idComboRepresentantesEdicaoUnidade=(String)dynaform.get("idComboRepresenta
ntes");
        String
idComboVinculosEdicaoUnidade=(String)dynaform.get("idComboVinculos");
        String enderecoEdicaoUnidade=(String)dynaform.get("enderecoUnidade");
        String bairroEdicaoUnidade=(String)dynaform.get("bairroUnidade");
        String
idComboCidadeEdicaoUnidade=(String)dynaform.get("idComboCidades");
        String
idComboEstadoEdicaoUnidade=(String)dynaform.get("idComboEstados");
        String cepEdicaoUnidade=(String)dynaform.get("cepUnidade");
        String dddUmEdicaoUnidade=(String)dynaform.get("dddUmUnidade");
        String telUmEdicaoUnidade=(String)dynaform.get("telUmUnidade");
        String dddDoisEdicaoUnidade=(String)dynaform.get("dddDoisUnidade");
        String telDoisEdicaoUnidade=(String)dynaform.get("telDoisUnidade");
        String dddEdicaoUnidade=(String)dynaform.get("dddFaxUnidade");
        String faxEdicaoUnidade=(String)dynaform.get("faxUnidade");
        String emailEdicaoUnidade=(String)dynaform.get("emailUnidade");

```

```
String idNovoUnidade=(String)dynaform.get("unidadeIdEditar");
```

```
UnidadeService unidadeService =  
(UnidadeService)getWebApplicationContext().getBean("unidadeService");  
Unidade unidade_Old = new Unidade();  
Unidade unidade_New = new Unidade();  
if(!nomeEdicaoUnidade.equals(""))  
    unidade_New.setNome(nomeEdicaoUnidade);  
if(!siglaEdicaoUnidade.equals("")){  
    unidade_New.setSigla(siglaEdicaoUnidade);  
}  
Usuario responsavel=new Usuario();  
if(!idComboRepresentantesEdicaoUnidade.equals("")){  
    Long  
idResponsavel=Long.valueOf(idComboRepresentantesEdicaoUnidade);  
    responsavel.setId(idResponsavel);  
    unidade_New.setResponsavel(responsavel);  
}  
Unidade vinculo=new Unidade();  
  
if((!idComboVinculosEdicaoUnidade.equals(""))&&!(idComboVinculosEdicaoUnidade  
.equals("0"))){  
    Long idVinculo=Long.valueOf(idComboVinculosEdicaoUnidade);  
    vinculo.setId(idVinculo);  
    unidade_New.setVinculo(vinculo);  
}  
if(!enderecoEdicaoUnidade.equals("")){  
    unidade_New.setEndereco(enderecoEdicaoUnidade);  
}else{  
    unidade_New.setEndereco(null);  
}  
if(!bairroEdicaoUnidade.equals("")){  
    unidade_New.setBairro(bairroEdicaoUnidade);  
}else{  
    unidade_New.setBairro(null);
```

```

}
Cidade cidadeUnidade=new Cidade();
if (!lidComboEstadoEdicaoUnidade.equals("0")){
    if(!lidComboEstadoEdicaoUnidade.equals("")){
        Estado estadoUnidade=new Estado();
        Long idEstado=Long.valueOf(idComboEstadoEdicaoUnidade);
        estadoUnidade.setId(idEstado);
        cidadeUnidade.setEstado(estadoUnidade);
    }
    if(!lidComboCidadeEdicaoUnidade.equals("")){
        Long idCidade=Long.valueOf(idComboCidadeEdicaoUnidade);
        cidadeUnidade.setId(idCidade);
        unidade_New.setCidade(cidadeUnidade);
    }
}
}else {
    unidade_New.setCidade(null);
}
}
if(!lcepEdicaoUnidade.equals("")){
    unidade_New.setCep(cepEdicaoUnidade);
}
else{
    unidade_New.setCep(null);
}
}
if(!ldddUmEdicaoUnidade.equals("")){
    unidade_New.setDddTelefone1(dddUmEdicaoUnidade);
}
else{
    unidade_New.setDddTelefone1(null);
}
}
if(!ltelUmEdicaoUnidade.equals("")){
    unidade_New.setTelefone1(telUmEdicaoUnidade);
}
else{
    unidade_New.setTelefone1(null);
}
}
if(!ldddDoisEdicaoUnidade.equals("")){
    unidade_New.setDddTelefone2(dddDoisEdicaoUnidade);
}
}

```

```

}else{
    unidade_New.setDddTelefone2(null);
}
if(!telDoisEdicaoUnidade.equals("")){
    unidade_New.setTelefone2(telDoisEdicaoUnidade);
}else{
    unidade_New.setTelefone2(null);
}
if(!dddEdicaoUnidade.equals("")){
    unidade_New.setDddFax(dddEdicaoUnidade);
}else{
    unidade_New.setDddFax(null);
}
if(!faxEdicaoUnidade.equals("")){
    unidade_New.setFax(faxEdicaoUnidade);
}else{
    unidade_New.setFax(null);
}
if(!emailEdicaoUnidade.equals("")){
    unidade_New.setEmail(emailEdicaoUnidade);
}else{
    unidade_New.setEmail(null);
}

unidade_Old.setId(Long.valueOf(idNovoUnidade));

unidadeService.edit(unidade_Old,unidade_New);

request.setAttribute("titulo", "Unidade Alterada");
request.setAttribute("mensagem", "Unidade alterada com sucesso.");
System.out.print("fim edição");
return mapping.findForward("resultadoSucesso");
} catch(Exception e) {
    request.setAttribute("mensagem", e.getMessage());
}

```

```
        return mapping.findForward("erro");
    }
}
```

```
public ActionForward exhibeUnidadeEdicao(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{

    DynaActionForm dynaform=(DynaActionForm)form;
    try{
        String idUnidadeString=(String)dynaform.get("unidadeIdEditar");
        request.setAttribute("unidadeIdEditar",idUnidadeString);
        Long idLong=Long.valueOf(idUnidadeString);

        UnidadeService unidadeService =
(UnidadeService)getWebApplicationContext().getBean("unidadeService");
        Unidade unidade = new Unidade();
        unidade.setId(idLong);
        Unidade unidadeEncontrada=(Unidade)unidadeService.selectPorId(unidade);

        request.setAttribute("unidadeEncontrada",unidadeEncontrada);

        if(unidadeEncontrada.getCidade()!=null){
            request.setAttribute("idEstado",
unidadeEncontrada.getCidade().getEstado().getId());
            request.setAttribute("idCidade", unidadeEncontrada.getCidade().getId());
        }
        //Preenchendo as listas
        UsuarioService usuarioService =
(UsuarioService)getWebApplicationContext().getBean("usuarioService");
        Collection users=usuarioService.getAll();
        if(unidadeEncontrada.getResponsavel()!=null){
            users.remove(unidadeEncontrada.getResponsavel());
        }
    }
}
```

```

    }
    request.setAttribute("listaRepresentantes",users);

    List vinculoSemPrimeiro=(List) unidadeService.getAll();
    List vinculoSemOutrosNenhum=new ArrayList();
    Iterator it2 =vinculoSemPrimeiro.iterator();
    while (it2.hasNext()){
        Unidade unit=(Unidade)it2.next();
        if (!(unit.getNome().equals("Outra"))) &&
(!unit.getNome().equals("Nenhuma"))){
            vinculoSemOutrosNenhum.add(unit);
        }
    }
    vinculoSemOutrosNenhum.remove(unidadeEncontrada);
    request.setAttribute("listaVinculos",vinculoSemOutrosNenhum);

    CidadeService cidadeService =
(CidadeService)getWebApplicationContext().getBean("cidadeService");

    Collection states=cidadeService.getEstadosComCidades();
    if(unidadeEncontrada.getCidade()!=null){
        states.remove(unidadeEncontrada.getCidade().getEstado());
    }

    if(unidadeEncontrada.getCidade()!=null){
        Estado estado=new Estado();
        estado.setId(unidadeEncontrada.getCidade().getEstado().getId());
        Collection citys=cidadeService.pesquisarSemOutra(estado);
        citys.remove(unidadeEncontrada.getCidade());
        request.setAttribute("listaCidades",citys);
    }
    if(unidadeEncontrada.getResponsavel()!=null){
        request.setAttribute("idComboRepresentantes",

```

```

unidadeEncontrada.getResponsavel().getId());
    }
    if(unidadeEncontrada.getVinculo()!=null){
        request.setAttribute("idComboVinculos",
unidadeEncontrada.getVinculo().getId());
    }

    request.setAttribute("listaEstados",states);
    //Fim preenchendo as listas

    request.setAttribute("nomeUnidade",unidadeEncontrada.getNome());
    request.setAttribute("siglaUnidade",unidadeEncontrada.getSigla());
    request.setAttribute("enderecoUnidade",unidadeEncontrada.getEndereco());
    request.setAttribute("bairroUnidade",unidadeEncontrada.getBairro());
    request.setAttribute("cepUnidade",unidadeEncontrada.getCep());

request.setAttribute("dddTelefone1Unidade",unidadeEncontrada.getDddTelefone1());
    request.setAttribute("telefone1Unidade",unidadeEncontrada.getTelefone1());

request.setAttribute("dddTelefone2Unidade",unidadeEncontrada.getDddTelefone2());
    request.setAttribute("telefone2Unidade",unidadeEncontrada.getTelefone2());
    request.setAttribute("dddFaxUnidade",unidadeEncontrada.getDddFax());
    request.setAttribute("faxUnidade",unidadeEncontrada.getFax());
    request.setAttribute("emailUnidade",unidadeEncontrada.getEmail());

    return mapping.findForward("edicao");
} catch(Exception e) {
    request.setAttribute("mensagem", e.getMessage());
    return mapping.findForward("erro");
}
}

public ActionForward getCidadesOnAddUsuario(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,

```

```

        HttpServletResponse response) throws Exception{
        DynaActionForm dynaform=(DynaActionForm)form;
        Collection unidade=(Collection)request.getAttribute("unidadeExibirEdicao");

        String estadold= (String) dynaform.get("idComboEstados");

        String representantesId= (String) dynaform.get("idComboRepresentantes");
        String vinculold= (String) dynaform.get("idComboVinculos");
        String nomeUnidade= (String) dynaform.get("nomeUnidade");
        String siglaUnidade= (String) dynaform.get( "siglaUnidade");
        String enderecoUnidade= (String) dynaform.get("enderecoUnidade");
        String bairroUnidade= (String) dynaform.get( "bairroUnidade");
        String cepUnidade= (String) dynaform.get("cepUnidade");
        String dddUmUnidade= (String) dynaform.get( "dddUmUnidade");
        String telefoneUmUnidade= (String) dynaform.get( "telUmUnidade");
        String dddDoisUnidade= (String) dynaform.get("dddDoisUnidade");
        String telefoneDoisUnidade= (String) dynaform.get("telDoisUnidade");
        String dddFaxUnidade= (String) dynaform.get("dddFaxUnidade");
        String faxUnidade= (String) dynaform.get( "faxUnidade");
        String emailUnidade= (String) dynaform.get("emailUnidade");
        Unidade
        unidadeEncontrada=(Unidade)request.getAttribute("unidadeEncontrada");

        Long idEstado =Long.valueOf(estadold);

        CidadeService cidadeService =
        (CidadeService)getWebApplicationContext().getBean("cidadeService");
        request.setAttribute("listaEstados",cidadeService.getEstadosComCidades());
        if(!(estadold.equals("0"))){
            Estado estado=new Estado();
            estado.setId(idEstado);

request.setAttribute("listaCidades",cidadeService.pesquisarSemOutra(estado));
        }
        request.setAttribute("idEstado", idEstado);

```

```

        request.setAttribute("idComboRepresentantes", representantesId);
        UsuarioService usuarioService =
        (UsuarioService)getWebApplicationContext().getBean("usuarioService");
        request.setAttribute("listaRepresentantes", usuarioService.getAll());

        request.setAttribute("idComboRepresentantes", representantesId);

        request.setAttribute("idComboVinculos", vinculoid);
        UnidadeService unidadeService =
        (UnidadeService)getWebApplicationContext().getBean("unidadeService");

        request.setAttribute("idComboVinculos", vinculoid);

        List vinculoSemPrimeiro=(List) unidadeService.getAll();
        List vinculoSemOutrosNenhum=new ArrayList();
        Iterator it2 =vinculoSemPrimeiro.iterator();
        while (it2.hasNext()){
            Unidade unit=(Unidade)it2.next();
            if ((!(unit.getNome().equals("Outra"))) &&
            (!(unit.getNome().equals("Nenhuma")))){
                vinculoSemOutrosNenhum.add(unit);
            }
        }
        request.setAttribute("listaVinculos",vinculoSemOutrosNenhum);

        request.setAttribute("listaOutrasUnidades",
        unidadeService.getQuantidadePorOutro());

        List unidadeSemPrimeiro=(List) unidadeService.getAll();
        List unidadesSemOutrosNenhum=new ArrayList();
        Iterator it =unidadeSemPrimeiro.iterator();
        while (it.hasNext()){
            Unidade unit=(Unidade)it.next();
            if ((!(unit.getNome().equals("Outra"))) &&
            (!(unit.getNome().equals("Nenhuma")))){

```

```

        unidadesSemOutrosNenhum.add(unit);
    }
}

request.setAttribute("listaUnidades",unidadesSemOutrosNenhum);

request.setAttribute("unidadeEncontrada",unidadeEncontrada);
request.setAttribute("unidadeExibirEdicao",unidade);
request.setAttribute("nomeUnidade",nomeUnidade);
request.setAttribute("siglaUnidade",siglaUnidade);
request.setAttribute("enderecoUnidade",enderecoUnidade);
request.setAttribute("bairroUnidade",bairroUnidade);
request.setAttribute("cepUnidade",cepUnidade);
request.setAttribute("dddUmUnidade",dddUmUnidade);
request.setAttribute("telUmUnidade",telefoneUmUnidade);
request.setAttribute("dddDoisUnidade",dddDoisUnidade);
request.setAttribute("telDoisUnidade",telefoneDoisUnidade);
request.setAttribute("dddFaxUnidade",dddFaxUnidade);
request.setAttribute("faxUnidade",faxUnidade);
request.setAttribute("emailUnidade",emailUnidade);

return mapping.findForward("success");
}

public ActionForward getCidadesOnEditUsuario(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{
    DynaActionForm dynaform=(DynaActionForm)form;

```

```
String estadoId= (String) dynaform.get("idComboEstados");
String representantesId= (String) dynaform.get("idComboRepresentantes");
String vinculoid= (String) dynaform.get("idComboVinculos");
String nomeUnidade= (String) dynaform.get("nomeUnidade");
String siglaUnidade= (String) dynaform.get( "siglaUnidade");
String enderecoUnidade= (String) dynaform.get("enderecoUnidade");
String bairroUnidade= (String) dynaform.get( "bairroUnidade");
String cepUnidade= (String) dynaform.get("cepUnidade");
String dddUmUnidade= (String) dynaform.get( "dddUmUnidade");
String telefoneUmUnidade= (String) dynaform.get( "telUmUnidade");
String dddDoisUnidade= (String) dynaform.get("dddDoisUnidade");
String telefoneDoisUnidade= (String) dynaform.get("telDoisUnidade");
String dddFaxUnidade= (String) dynaform.get("dddFaxUnidade");
String faxUnidade= (String) dynaform.get( "faxUnidade");
String emailUnidade= (String) dynaform.get("emailUnidade");
```

```
String idUnidadeString=(String)dynaform.get("unidadeIdEditar");
request.setAttribute("unidadeIdEditar",idUnidadeString);
```

```
Long idEstado =Long.valueOf(estadoId);
CidadeService cidadeService =
(CidadeService)getWebApplicationContext().getBean("cidadeService");
request.setAttribute("listaEstados",cidadeService.getEstadosComCidades());
if(!(estadoId.equals("0"))){
    Estado estado=new Estado();
    estado.setId(idEstado);

request.setAttribute("listaCidades",cidadeService.pesquisarSemOutra(estado));
}
request.setAttribute("idEstado", idEstado);
```

```
request.setAttribute("idComboRepresentantes", representantesId);
UsuarioService usuarioService =
(UsuarioService)getWebApplicationContext().getBean("usuarioService");
```

```

request.setAttribute("listaRepresentantes",usuarioService.getAll());

request.setAttribute("idComboVinculos", vinculoid);
UnidadeService unidadeService =
(UnidadeService)getWebApplicationContext().getBean("unidadeService");

List vinculoSemPrimeiro=(List) unidadeService.getAll();
List vinculoSemOutrosNenhum=new ArrayList();
Iterator it2 =vinculoSemPrimeiro.iterator();
while (it2.hasNext()){
    Unidade unit=(Unidade)it2.next();
    if (!(unit.getNome().equals("Outra"))) &&
(!unit.getNome().equals("Nenhuma"))){
        vinculoSemOutrosNenhum.add(unit);
    }
}
request.setAttribute("listaVinculos",vinculoSemOutrosNenhum);

request.setAttribute("nomeUnidade",nomeUnidade);
request.setAttribute("siglaUnidade",siglaUnidade);
request.setAttribute("enderecoUnidade",enderecoUnidade);
request.setAttribute("bairroUnidade",bairroUnidade);
request.setAttribute("cepUnidade",cepUnidade);
request.setAttribute("dddTelefone1Unidade",dddUmUnidade);
request.setAttribute("telefone1Unidade",telefoneUmUnidade);
request.setAttribute("dddTelefone2Unidade",dddDoisUnidade);
request.setAttribute("telefone2Unidade",telefoneDoisUnidade);
request.setAttribute("dddFaxUnidade",dddFaxUnidade);
request.setAttribute("faxUnidade",faxUnidade);
request.setAttribute("emailUnidade",emailUnidade);

return mapping.findForward("edicao");

```

```
}  
}
```

```
..\src\java\br\com\omd\ouvidoria\web\controle\action\UsuarioAction.java  
package br.com.omd.ouvidoria.web.controle.action;
```

```
import br.com.omd.ouvidoria.logica.domain.Cargo;  
import br.com.omd.ouvidoria.logica.domain.Perfil;  
import br.com.omd.ouvidoria.logica.domain.Telefone;  
import br.com.omd.ouvidoria.logica.domain.Unidade;  
import br.com.omd.ouvidoria.logica.domain.Usuario;  
import br.com.omd.ouvidoria.logica.dto.UsuarioDTO;  
import br.com.omd.ouvidoria.logica.service.CargoService;  
import br.com.omd.ouvidoria.logica.service.PerfilService;  
import br.com.omd.ouvidoria.logica.service.UnidadeService;  
import br.com.omd.ouvidoria.logica.service.UsuarioService;  
import br.com.omd.ouvidoria.util.Criptografia;  
import java.util.ArrayList;  
import java.util.Collection;  
import java.util.Iterator;  
import java.util.List;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import javax.servlet.http.HttpSession;  
import org.apache.struts.action.ActionForm;  
import org.apache.struts.action.ActionForward;  
import org.apache.struts.action.ActionMapping;  
import org.apache.struts.action.DynaActionForm;
```

```
public class UsuarioAction extends BaseAction{
```

```
    public UsuarioAction() {  
    }  
}
```

```

public ActionForward defaultMethod(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{
    return getUsuarios(mapping, form, request, response);
}
/**
 *@version 1.0
 *@method addUsuario
 *@author David Gerson Ramos de Andrade
 *@description action criado para inserção de novo usuario.
 */
public ActionForward addUsuario(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{
    DynaActionForm dynaform=(DynaActionForm)form;
    try{
        String nomeNovoUsuario=(String)dynaform.get("nomeCadastroUsuario");
        String loginNovoUsuario=(String)dynaform.get("loginCadastroUsuario");
        String apelidoNovoUsuario=(String)dynaform.get("apelidoCadastroUsuario");
        String senhaNovoUsuario=(String)dynaform.get("senhaCadastroUsuario");
        String idPerfilNovoUsuario=(String)dynaform.get("idComboUsuarioPerfil");
        String idCargoNovoUsuario=(String)dynaform.get("idComboUsuarioCargo");
        String
idUnidadeNovoUsuario=(String)dynaform.get("idComboUsuarioUnidade");
        String dddNovoUsuario=(String)dynaform.get("dddCadastroUsuario");
        String
telefoneNovoUsuario=(String)dynaform.get("telefoneCadastroUsuario");
        String emailNovoUsuario=(String)dynaform.get("emailCadastroUsuario");

        UsuarioService usuarioService =
(UsuarioService)getWebApplicationContext().getBean("usuarioService");

```

```

        Usuario usuario = new Usuario();
        usuario.setNome(nomeNovoUsuario);
        usuario.setLogin(loginNovoUsuario);
        if (apelidoNovoUsuario!=""){
            usuario.setApelido(apelidoNovoUsuario);
        } else{
            if(nomeNovoUsuario.indexOf(" ")!=-1){
                String
                apelidoNewUser=nomeNovoUsuario.substring(0,nomeNovoUsuario.indexOf(" "));
                usuario.setApelido(apelidoNewUser);
            }else{
                usuario.setApelido(nomeNovoUsuario);
            }
        }

        String
        senhaCadastroNovoUsuario=Criptografia.retornaString(Criptografia.digest((senhaNovoUsuario).getBytes()));
        usuario.setSenha(senhaCadastroNovoUsuario);
        Perfil perfilUsuario=new Perfil();
        Long idperfil=Long.valueOf(idPerfilNovoUsuario);
        perfilUsuario.setld(idperfil);
        usuario.setPerfil(perfilUsuario);
        if(!idCargoNovoUsuario.equals("0")){
            Cargo cargoUsuario=new Cargo();
            Long idcargo=Long.valueOf(idCargoNovoUsuario);
            cargoUsuario.setld(idcargo);
            usuario.setCargo(cargoUsuario);
        }else{
            usuario.setCargo(null);
        }
        Unidade unidadeUsuario=new Unidade();
        Long idunidade=Long.valueOf(idUnidadeNovoUsuario);
        unidadeUsuario.setld(idunidade);
        usuario.setUnidade(unidadeUsuario);

```

```

        usuario.setDddTelefone1(dddNovoUsuario);
        usuario.setTelefone1(telefoneNovoUsuario);
        usuario.setEmail(emailNovoUsuario);

        usuarioService.save(usuario);

        request.setAttribute("titulo", "Usuario Cadastrado");
        request.setAttribute("mensagem", "Usuario cadastrado com sucesso.");

        return mapping.findForward("resultadoSucesso");
    } catch (Exception e) {
        request.setAttribute("mensagem", e.getMessage());
        return mapping.findForward("erro");
    }
}
/**
 * @version 1.0
 * @method deleteUsuario
 * @author David Gerson Ramos de Andrade
 * @description action criado para remoção de novo usuario.
 */
public ActionForward deleteUsuario(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{

    DynaActionForm dynaform=(DynaActionForm)form;

    String idExcluirUsuario=(String)dynaform.get("idComboUsuarioExcluir");

    HttpSession session = request.getSession(true);
    Usuario user=(Usuario)session.getAttribute("usuario");

    Usuario usuario = (Usuario)request.getSession(false).getAttribute("usuario");

```

```

String senhaExclusao =
Criptografia.retornaString(Criptografia.digest(((String)dynaform.get("senhaConferenci
a")).getBytes()));
String destino= "resultadoSucesso";
UsuarioDTO userDTO=new UsuarioDTO();
userDTO.setLogin(user.getApelido());
userDTO.setSenha(senhaExclusao);

if (usuario != null) {
    if (user.getSenha().equals(senhaExclusao)) {
        try {
            UsuarioService usuarioService =
(UsuarioService)getWebApplicationContext().getBean("usuarioService");

            Usuario usuarioDelete = new Usuario();

            usuarioDelete.setId(Long.valueOf(idExcluirUsuario));

            usuarioService.delete(usuarioDelete);

            request.setAttribute("titulo", "Usuário Excluído");
            request.setAttribute("mensagem", "Usuário excluído com sucesso.");

        } catch(Exception e) {
            request.setAttribute("mensagem", e.getMessage());
            destino = "erro";
        }
    } else {
        request.setAttribute("mensagem", "Senha inválida.");
        destino = "erro";
    }
} else {
    request.setAttribute("mensagem", "Usuário não logado.");
    destino = "erro";
}
}

```

```

        return mapping.findForward(destino);
    }
    public ActionForward getUsuario(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{

        DynaActionForm dynaform=(DynaActionForm)form;
        try{
            String nomeUsuario=(String)dynaform.get("consultaUsuario");

            UsuarioService usuarioService =
(UusuarioService)getWebApplicationContext().getBean("usuarioService");
            Usuario usuario = new Usuario();
            usuario.setNome(nomeUsuario);
            Collection listaUsuariosEncontrados=usuarioService.select(usuario);

            request.setAttribute("listaUsuariosEncontrados",listaUsuariosEncontrados);

request.setAttribute("tamanhoLinhas",Integer.toString(listaUsuariosEncontrados.size(
)));

            return mapping.findForward("resultadoConsulta");
        } catch(Exception e) {
            request.setAttribute("mensagem", e.getMessage());
            return mapping.findForward("erro");
        }
    }
    public ActionForward getUsuarios(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{
        DynaActionForm dynaform=(DynaActionForm)form;
        try{
            UsuarioService usuarioService =
(UusuarioService)getWebApplicationContext().getBean("usuarioService");

```

```

        request.setAttribute("listaUsuarios",usuarioService.getAll());

        CargoService cargoService =
(CargoService)getWebApplicationContext().getBean("cargoService");
        request.setAttribute("listaCargos",cargoService.getAll());

        PerfilService perfilService =
(PerfilService)getWebApplicationContext().getBean("perfilService");
        request.setAttribute("listaPerfis",perfilService.getAll());

        UnidadeService unidadeService =
(UnidadeService)getWebApplicationContext().getBean("unidadeService");

        List unidadeGeral=(List) unidadeService.getAll();
        List unidadeSemOutrosNenhum=new ArrayList();
        Iterator it2 =unidadeGeral.iterator();
        while (it2.hasNext()){
            Unidade unit=(Unidade)it2.next();
            if (!(unit.getNome().equals("Outra"))) &&
(!unit.getNome().equals("Nenhuma"))){
                unidadeSemOutrosNenhum.add(unit);
            }
        }

        request.setAttribute("listaUnidades",unidadeSemOutrosNenhum);

        return mapping.findForward("success");

    } catch(Exception e) {
        request.setAttribute("mensagem", e.getMessage());
        return mapping.findForward("erro");
    }
}
/**
 *@version 1.0

```

```

    *@method editUsuario
    *@author David Gerson Ramos de Andrade
    *@description action criado para alteração de usuario.
    */
    public ActionForward editUsuario(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{
        DynaActionForm dynaform=(DynaActionForm)form;

        try{
            String nomeEdicaoUsuario=(String)dynaform.get("nomeEdicaoUsuario");
            String usuariold=(String)dynaform.get("usuarioldEditar");
            String loginEdicaoUsuario=(String)dynaform.get("loginEdicaoUsuario");
            String apelidoEdicaoUsuario=(String)dynaform.get("apelidoEdicaoUsuario");
            String senhaEdicaoUsuario=(String)dynaform.get("senhaEdicaoUsuario");
            String idPerfilEdicaoUsuario=(String)dynaform.get("idComboUsuarioPerfil");
            String idCargoEdicaoUsuario=(String)dynaform.get("idComboUsuarioCargo");
            String
            idUnidadeEdicaoUsuario=(String)dynaform.get("idComboUsuarioUnidade");
            String dddEdicaoUsuario=(String)dynaform.get("dddEdicaoUsuario");
            String telefoneEdicaoUsuario=(String)dynaform.get("telefoneEdicaoUsuario");
            String emailEdicaoUsuario=(String)dynaform.get("emailEdicaoUsuario");

            UsuarioService usuarioService =
            (UsuarioService)getWebApplicationContext().getBean("usuarioService");
            Usuario usuario_Old = new Usuario();
            Long idOldUser=Long.valueOf(usuariold);
            usuario_Old.setId(idOldUser);
            Usuario usuario_New = new Usuario();

            usuario_New.setNome(nomeEdicaoUsuario);
            usuario_New.setLogin(loginEdicaoUsuario);
            if (apelidoEdicaoUsuario!=""){

```

```

        usuario_New.setApelido(apelidoEdicaoUsuario);
    } else{
        if(nomeEdicaoUsuario.indexOf(" ")!=-1){
            String
apelidoNewUser=nomeEdicaoUsuario.substring(0,nomeEdicaoUsuario.indexOf(" "));
            usuario_New.setApelido(apelidoNewUser);
        }else{
            usuario_New.setApelido(nomeEdicaoUsuario);
        }
    }
    String senhaEdicaoNovoUsuario=null;
    if (!(senhaEdicaoUsuario.equals(""))){

senhaEdicaoNovoUsuario=Criptografia.retornaString(Criptografia.digest((senhaEdica
oUsuario).getBytes()));
    }
    usuario_New.setSenha(senhaEdicaoNovoUsuario);
    Perfil perfilUsuario=new Perfil();
    Long idperfil=Long.valueOf(idPerfilEdicaoUsuario);
    perfilUsuario.setld(idperfil);
    usuario_New.setPerfil(perfilUsuario);
    Cargo cargoUsuario=new Cargo();
    if(!idCargoEdicaoUsuario.equals("0")){
        Long idcargo=Long.valueOf(idCargoEdicaoUsuario);
        cargoUsuario.setld(idcargo);
        usuario_New.setCargo(cargoUsuario);
    }else{
        usuario_New.setCargo(null);
    }
    Unidade unidadeUsuario=new Unidade();
    Long idunidade=Long.valueOf(idUnidadeEdicaoUsuario);
    unidadeUsuario.setld(idunidade);
    usuario_New.setUnidade(unidadeUsuario);

usuario_New.setDddTelefone1(dddEdicaoUsuario);

```

```

        usuario_New.setTelefone1(telefoneEdicaoUsuario);
        usuario_New.setEmail(emailEdicaoUsuario);

        usuarioService.edit(usuario_Old,usuario_New);

        request.setAttribute("titulo", "Usuario Alterado");
        request.setAttribute("mensagem", "Usuario alterado com sucesso.");

        return mapping.findForward("resultadoSucesso");
    } catch(Exception e) {
        request.setAttribute("mensagem", e.getMessage());
        return mapping.findForward("erro");
    }
}

public ActionForward exhibeUsuario(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{

    Long idLong=Long.valueOf((String)request.getParameter("id"));

    UsuarioService usuarioService =
    (UsuarioService)getWebApplicationContext().getBean("usuarioService");
    Usuario usuario = new Usuario();
    usuario.setId(idLong);
    Usuario us=(Usuario)usuarioService.selectPorId(usuario);
    request.setAttribute("idUsuario",us.getId().toString());
    request.setAttribute("nomeUsuario",us.getNome());
    request.setAttribute("loginUsuario",us.getLogin());
    request.setAttribute("apelidoUsuario",us.getApelido());
    if(us.getCargo()!=null){
        request.setAttribute( "cargoUsuario",us.getCargo().getNome());
    }
}

```

```

        if(us.getTelefone1()==null) {
            request.setAttribute( "telefoneUsuario","");
        }else{
            request.setAttribute(
"telefoneUsuario",us.getDddTelefone1()+us.getTelefone1());
        }
        request.setAttribute( "emailUsuario",us.getEmail());
        request.setAttribute( "perfilUsuario",us.getPerfil().getNome());
        request.setAttribute( "unidadeUsuario",us.getUnidade().getNome());

        return mapping.findForward("exibicao");
    }
    public ActionForward exhibeUsuarioEdicao(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{

        DynaActionForm dynaform=(DynaActionForm)form;
        try{
            String idUsuarioString=(String)dynaform.get("usuarioldEditar");

            Long idLong=Long.valueOf(idUsuarioString);

            UsuarioService usuarioService =
(UusuarioService)getWebApplicationContext().getBean("usuarioService");
            Usuario usuario = new Usuario();
            usuario.setId(idLong);
            Usuario usuarioEncontrado=(Usuario)usuarioService.selectPorId(usuario);

            request.setAttribute("usuarioNome",usuarioEncontrado.getNome());

            request.setAttribute("usuarioDddTelefone1",usuarioEncontrado.getDddTelefone1());
            request.setAttribute("usuarioTelefone1",usuarioEncontrado.getTelefone1());
            request.setAttribute("usuarioEmail",usuarioEncontrado.getEmail());
            request.setAttribute("usuarioLogin",usuarioEncontrado.getLogin());

```

```
request.setAttribute("usuarioApellido",usuarioEncontrado.getApellido());
request.setAttribute("usuarioPerfil",usuarioEncontrado.getPerfil());
request.setAttribute("usuarioUnidade",usuarioEncontrado.getUnidade());
request.setAttribute("usuarioCargo",usuarioEncontrado.getCargo());
```

```
//Preenchendo as listas
```

```
    CargoService cargoService =
(CargoService)getWebApplicationContext().getBean("cargoService");
    Collection cargosList=cargoService.getAll();
    if(usuarioEncontrado.getCargo()!=null){
        cargosList.remove(usuarioEncontrado.getCargo());
    }
    request.setAttribute("listaCargos",cargosList);
```

```
    PerfilService perfilService =
(PerfilService)getWebApplicationContext().getBean("perfilService");
    Collection perfilList=perfilService.getAll();
    if(usuarioEncontrado.getPerfil()!=null){
        perfilList.remove(usuarioEncontrado.getPerfil());
    }
    request.setAttribute("listaPerfis",perfilList);
```

```
    UnidadeService unidadeService =
(UnidadeService)getWebApplicationContext().getBean("unidadeService");
```

```
    List unidadeGeral=(List) unidadeService.getAll();
    List unidadeSemOutrosNenhum=new ArrayList();
    Iterator it2 =unidadeGeral.iterator();
    while (it2.hasNext()){
        Unidade unit=(Unidade)it2.next();
        if (!(unit.getNome().equals("Outra"))) &&
(!unit.getNome().equals("Nenhuma"))){
            unidadeSemOutrosNenhum.add(unit);
        }
    }
}
```

```

        if(usuarioEncontrado.getUnidade()!=null){
            unidadeSemOutrosNenhum.remove(usuarioEncontrado.getUnidade());
        }
        request.setAttribute("listaUnidades",unidadeSemOutrosNenhum);
        //Fim preenchendo as listas

        return mapping.findForward("edicao");
    } catch(Exception e) {
        request.setAttribute("mensagem", e.getMessage());
        return mapping.findForward("erro");
    }
}
}
}

```

..\src\java\br\com\omd\ouvidoria\web\controle\action\caixa\AguardandoRespostaAction.java

```
package br.com.omd.ouvidoria.web.controle.action.caixa;
```

```

import br.com.omd.ouvidoria.logica.domain.Unidade;
import br.com.omd.ouvidoria.logica.domain.Usuario;
import br.com.omd.ouvidoria.logica.service.CaixaManifestacaoService;
import br.com.omd.ouvidoria.web.controle.action.*;
import java.util.HashMap;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.DynaActionForm;

```

```
public class AguardandoRespostaAction extends BaseAction{
```

```

    public AguardandoRespostaAction() {
    }

```

```

public ActionForward defaultMethod(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{
    return getAguardandoResposta(mapping, form, request, response);
}

public ActionForward getAguardandoResposta(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{
    DynaActionForm dynaform=(DynaActionForm)form;
    String destino = "aguardandoResposta";
    Usuario usuario = (Usuario)request.getSession(false).getAttribute("usuario");
    if (usuario != null){
        CaixaManifestacaoService caixaManifestacaoService =
(CaixaManifestacaoService)getWebApplicationContext().getBean("caixaManifestaca
oService");
        Unidade unidadeUsuario = null;
        Unidade unidadeParametro = null;
        long idPerfil = usuario.getPerfil().getId().longValue();
        if (idPerfil == 3){
            unidadeUsuario = usuario.getUnidade();
        } else {
            if (idPerfil == 1){
                String idUnidade = request.getParameter("idUnidade");
                if (idUnidade != null){
                    unidadeParametro = new Unidade();
                    unidadeParametro.setId(Long.valueOf(idUnidade));
                    request.setAttribute("unidadeParametro",
caixaManifestacaoService.getUnidade(unidadeParametro.getId()));
                    unidadeUsuario = usuario.getUnidade();
                }
            }
        }
    }
}

```

```

    }
}
try {
    request.setAttribute("aguardandoResposta",
caixaManifestacaoService.getAguardandoResposta(unidadeParametro,
unidadeUsuario));
    } catch (Exception e){
        destino = "erro";
        request.setAttribute("mensagem",e.getMessage());
    }
}
return mapping.findForward(destino);
}

```

```

public ActionForward getNova(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{
    Usuario usuario = (Usuario)request.getSession(false).getAttribute("usuario");
    if (usuario == null){
        return null;
    }
    DynaActionForm dynaform=(DynaActionForm)form;
    CaixaManifestacaoService caixaManifestacaoService =
(CaixaManifestacaoService)getWebApplicationContext().getBean("caixaManifestaca
oService");

    request.setAttribute("manifestacao",caixaManifestacaoService.getNova((Long)dynafo
rm.get("id")));

    HashMap dadosCadastro =
caixaManifestacaoService.getDadosNova(usuario.getUnidade().getId());
    request.setAttribute("unidades",dadosCadastro.get("unidades"));

    request.setAttribute("unidadesNivel1SemOutraSemNenhuma",dadosCadastro.get("u
nidadesNivel1SemOutraSemNenhuma"));
    request.setAttribute("classificacoes",dadosCadastro.get("classificacoes"));
    request.setAttribute("assuntos",dadosCadastro.get("assuntos"));
}

```

```
        return mapping.findForward("nova");
    }
}
```

```
..\src\java\br\com\omd\ouvidoria\web\controle\action\caixa\ARevisarAction.java
package br.com.omd.ouvidoria.web.controle.action.caixa;
```

```
import br.com.omd.ouvidoria.logica.domain.Unidade;
import br.com.omd.ouvidoria.logica.domain.Usuario;
import br.com.omd.ouvidoria.logica.service.CaixaManifestacaoService;
import br.com.omd.ouvidoria.web.controle.action.*;
import java.util.HashMap;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.DynaActionForm;
```

```
public class ARevisarAction extends BaseAction{
```

```
    public ARevisarAction() {
    }
}
```

```
    public ActionForward defaultMethod(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{
        String destino = "arevisar";
        Usuario usuario = (Usuario)request.getSession(false).getAttribute("usuario");
        if (usuario != null){
            CaixaManifestacaoService caixaManifestacaoService =
            (CaixaManifestacaoService)getWebApplicationContext().getBean("caixaManifestaca
```

```

oService");
    Unidade unidadeUsuario = null;
    Unidade unidadeParametro = null;
    String idUnidade = request.getParameter("idUnidade");
    if (idUnidade != null){
        unidadeParametro = new Unidade();
        unidadeParametro.setId(Long.valueOf(idUnidade));
        request.setAttribute("unidadeParametro",
caixaManifestacaoService.getUnidade(unidadeParametro.getId()));
    } else {
        if (usuario.getPerfil().getId().longValue() == 3){
            unidadeUsuario = usuario.getUnidade();
        }
    }
    try {
        request.setAttribute("arevisar",
caixaManifestacaoService.getAREvisar(unidadeParametro, unidadeUsuario));
    } catch (Exception e){
        destino = "erro";
        request.setAttribute("mensagem",e.getMessage());
    }
}
return mapping.findForward(destino);
}
}

```

```

..\src\java\br\com\omd\ouvidoria\web\controle\action\caixa\CaixaManifestacoesAction
.java

```

```

package br.com.omd.ouvidoria.web.controle.action.caixa;

```

```

import br.com.omd.ouvidoria.logica.domain.Usuario;
import br.com.omd.ouvidoria.logica.service.CaixaManifestacaoService;
import br.com.omd.ouvidoria.web.controle.action.*;
import javax.servlet.http.HttpServletRequest;

```



```

import br.com.ond.ouvidoria.logica.service.CaixaManifestacaoService;
import br.com.ond.ouvidoria.web.controle.action.*;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.DynaActionForm;

public class DetalhesAction extends BaseAction{

    public DetalhesAction() {
    }

    public ActionForward defaultMethod(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{
    try{
        DynaActionForm dynaform=(DynaActionForm)form;
        CaixaManifestacaoService caixaManifestacaoService =
(CaixaManifestacaoService)getWebApplicationContext().getBean("caixaManifestaca
oService");

request.setAttribute("manifestacao",caixaManifestacaoService.getDetalhes((Long)dy
naform.get("codigo")));
        return mapping.findForward("detalhes");
    } catch(Exception e) {
        request.setAttribute("mensagem", e.getMessage());
        return mapping.findForward("erro");
    }
    }

}

```

```
..\src\java\br\com\omd\ouvidoria\web\controle\action\caixa\EncaminharAction.java  
package br.com.omd.ouvidoria.web.controle.action.caixa;
```

```
import br.com.omd.ouvidoria.logica.domain.Assunto;  
import br.com.omd.ouvidoria.logica.domain.Ciclo;  
import br.com.omd.ouvidoria.logica.domain.Classificacao;  
import br.com.omd.ouvidoria.logica.domain.Encaminhamento;  
import br.com.omd.ouvidoria.logica.domain.Manifestacao;  
import br.com.omd.ouvidoria.logica.domain.Unidade;  
import br.com.omd.ouvidoria.logica.domain.Usuario;  
import br.com.omd.ouvidoria.logica.domain.outros.OutraUnidade;  
import br.com.omd.ouvidoria.logica.domain.outros.OutroAssunto;  
import br.com.omd.ouvidoria.logica.service.CaixaManifestacaoService;  
import br.com.omd.ouvidoria.util.Email;  
import br.com.omd.ouvidoria.util.ExceptionEmail;  
import br.com.omd.ouvidoria.web.controle.action.*;  
import java.io.File;  
import java.util.ArrayList;  
import javax.mail.MessagingException;  
import javax.servlet.ServletContext;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import org.apache.struts.action.ActionForm;  
import org.apache.struts.action.ActionForward;  
import org.apache.struts.action.ActionMapping;  
import org.apache.struts.action.DynaActionForm;
```

```
public class EncaminharAction extends BaseAction{
```

```
    public EncaminharAction() {  
    }  
}
```

```
    public ActionForward defaultMethod(ActionMapping mapping,
```

```

        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{
    return montaTelaEncaminhamento(mapping,form, request, response);
}

public ActionForward montaTelaEncaminhamento(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{
    this.montaTela(form, request);
    return mapping.findForward("encaminhar");
}

public ActionForward montaTelaComentario(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{
    this.montaTela(form, request);
    return mapping.findForward("comentar");
}

public void montaTela(ActionForm form,
        HttpServletRequest request) throws Exception{
    Usuario usuario = (Usuario)request.getSession(false).getAttribute("usuario");
    if (usuario != null){
        CaixaManifestacaoService caixaManifestacaoService =
(CaixaManifestacaoService)getWebApplicationContext().getBean("caixaManifestacaoService");

request.setAttribute("manifestacao",caixaManifestacaoService.getDetalhes(((Long)((DynaActionForm)form).get("codigo"))));

request.setAttribute("unidadesNivelAbaixoSemOutraSemNenhuma",caixaManifestacaoService.getUnidadesNivelAbaixoSemOutraSemNenhuma(usuario.getUnidade().getId()));
    }
}

```

```
public ActionForward encaminhar(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{
    DynaActionForm dynaform=(DynaActionForm)form;
    Manifestacao manifestacao = new Manifestacao();
    manifestacao.setCodigo((Long)dynaform.get("codigo"));

    return realizaEncaminhamento(mapping, dynaform,request, manifestacao);
}
```

```
public ActionForward encaminhaNova(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{
    DynaActionForm dynaform=(DynaActionForm)form;
    Manifestacao manifestacao = new Manifestacao();
    manifestacao.setCodigo((Long)dynaform.get("codigo"));
    Unidade unidade = new Unidade();
    String nomeUnidade = dynaform.getString("unidade");
    unidade.setNome(nomeUnidade);
    manifestacao.setUnidadeOcorrencia(unidade);

    if (nomeUnidade.equals("Outra")){
        OutraUnidade outraUnidade = new OutraUnidade();
        outraUnidade.setNome(dynaform.getString("outraUnidade"));
        manifestacao.setOutraUnidade(outraUnidade);
    }

    manifestacao.setEnvolvidos(dynaform.getString("envolvidos"));
    manifestacao.setDescricao(dynaform.getString("descricao"));

    Classificacao classificacao = new Classificacao();
```

```

classificacao.setId((Long)dynaform.get("classificacao"));
manifestacao.setClassificacao(classificacao);

Assunto assunto = new Assunto();
String nomeAssunto = (String)dynaform.get("assunto");
assunto.setNome(nomeAssunto);
manifestacao.setAssunto(assunto);
if (nomeAssunto.equals("Outro")){
    OutroAssunto outroAssunto = new OutroAssunto();
    outroAssunto.setNome(dynaform.getString("outroAssunto"));
    manifestacao.setOutroAssunto(outroAssunto);
}
Long idUnidadeEncaminhar = (Long)dynaform.get("unidadeEncaminhar");
CaixaManifestacaoService caixaManifestacaoService =
(CaixaManifestacaoService)getWebApplicationContext().getBean("caixaManifestacaoService");
manifestacao.setNova(false);
caixaManifestacaoService.atualizarNova(manifestacao);
if (idUnidadeEncaminhar.longValue() == 0) {
    request.setAttribute("titulo", "Manifestação Atualizada");
    request.setAttribute("texto", "Manifestação atualizada");
    request.setAttribute("manifestacao", manifestacao);
    return mapping.findForward("resultado");
} else {
    return realizaEncaminhamento(mapping, dynaform,request, manifestacao);
}
}

public ActionForward realizaEncaminhamento(ActionMapping mapping,
DynaActionForm dynaform, HttpServletRequest request,
Manifestacao manifestacao) throws Exception{
Encaminhamento encaminhamento = new Encaminhamento();
Usuario usuario = (Usuario)request.getSession(false).getAttribute("usuario");
encaminhamento.setUsuario(usuario);
encaminhamento.setCargo(usuario.getCargo());

```

```

encaminhamento.setOrigem(usuario.getUnidade());

Unidade unidadeEncaminhar = new Unidade();
unidadeEncaminhar.setId((Long)dynaform.get("unidadeEncaminhar"));
encaminhamento.setDestino(unidadeEncaminhar);
String comentario = dynaform.getString("comentario");
if (!comentario.equals("")){
    encaminhamento.setTexto(dynaform.getString("comentario"));
}

Ciclo ciclo = new Ciclo();
ciclo.setEncaminhamentos(new ArrayList());
ciclo.getEncaminhamentos().add(encaminhamento);
ciclo.setManifestacao(manifestacao);

manifestacao.setCiclos(new ArrayList());
manifestacao.getCiclos().add(ciclo);

ServletContext context = getServletContext();
File arquivoNotificacaoEncaminhamento = new
File(context.getRealPath("docs/notificacao_encaminhamento.txt"));

Email email = new Email();
email.setServidor(context.getInitParameter("servidorEmail"));
email.setEmissor(context.getInitParameter("apelidoEmail") + " <" +
    context.getInitParameter("email") + ">");
email.setUsuario(context.getInitParameter("usuarioEmail"));
email.setSenha(context.getInitParameter("senhaEmail"));
email.setAssunto("Nova Manifestação");

CaixaManifestacaoService caixaManifestacaoService =
(CaixaManifestacaoService)getWebApplicationContext().getBean("caixaManifestaca
oService");
try {
    caixaManifestacaoService.encaminhar(manifestacao,
arquivoNotificacaoEncaminhamento, email);
}

```

```
    } catch (ExceptionEmail exceptionEmail) {
        request.setAttribute("exceptionEmail", exceptionEmail);
    }
    request.setAttribute("titulo", "Encaminhamento Enviado");
    request.setAttribute("texto", "Encaminhamento da manifestação realizado");
    request.setAttribute("manifestacao", manifestacao);
    return mapping.findForward("resultado");
}

}
```

```
..\src\java\br\com\omd\ouvidoria\web\controle\action\caixa\EncerradasAction.java
package br.com.omd.ouvidoria.web.controle.action.caixa;
```

```
import br.com.omd.ouvidoria.logica.domain.Unidade;
import br.com.omd.ouvidoria.logica.domain.Usuario;
import br.com.omd.ouvidoria.logica.service.CaixaManifestacaoService;
import br.com.omd.ouvidoria.web.controle.action.*;
import java.util.HashMap;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.DynaActionForm;
```

```
public class EncerradasAction extends BaseAction{
```

```
    public EncerradasAction() {
    }
}
```

```
    public ActionForward defaultMethod(ActionMapping mapping,
```

```

        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{
    return montaTela(mapping, form, request, response);
}

```

```

public ActionForward montaTela(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{

    DynaActionForm dynaform=(DynaActionForm)form;
    String destino = "encerradas";
    Usuario usuario = (Usuario)request.getSession(false).getAttribute("usuario");
    if (usuario != null){
        CaixaManifestacaoService caixaManifestacaoService =
(CaixaManifestacaoService)getWebApplicationContext().getBean("caixaManifestaca
oService");

        Unidade unidadeParametro = null;
        String idUnidade = request.getParameter("idUnidade");
        if (idUnidade != null){
            unidadeParametro = new Unidade();
            unidadeParametro.setId(Long.valueOf(idUnidade));
            request.setAttribute("unidadeParametro",
caixaManifestacaoService.getUnidade(unidadeParametro.getId()));
        }
        try {
            request.setAttribute("encerradas",
caixaManifestacaoService.getEncerradas(unidadeParametro, usuario));
        } catch (Exception e){
            destino = "erro";
            request.setAttribute("mensagem",e.getMessage());
        }
    }
    return mapping.findForward(destino);
}

```

```
}  
}
```

```
..\src\java\br\com\omd\ouvidoria\web\controle\action\caixa\EnviarRespostaAction.java
```

```
package br.com.omd.ouvidoria.web.controle.action.caixa;
```

```
import br.com.omd.ouvidoria.logica.domain.Envio;  
import br.com.omd.ouvidoria.logica.domain.Manifestacao;  
import br.com.omd.ouvidoria.logica.domain.Manifestante;  
import br.com.omd.ouvidoria.logica.domain.Usuario;  
import br.com.omd.ouvidoria.logica.service.CaixaManifestacaoService;  
import br.com.omd.ouvidoria.util.Email;  
import br.com.omd.ouvidoria.util.ExceptionEmail;  
import br.com.omd.ouvidoria.web.controle.action.*;  
import com.sun.rsa.sign.s;  
import java.io.RandomAccessFile;  
import java.text.NumberFormat;  
import java.text.SimpleDateFormat;  
import javax.mail.MessagingException;  
import javax.servlet.ServletContext;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import org.apache.struts.action.ActionForm;  
import org.apache.struts.action.ActionForward;  
import org.apache.struts.action.ActionMapping;  
import org.apache.struts.action.DynaActionForm;
```

```
public class EnviarRespostaAction extends BaseAction{
```

```
    public EnviarRespostaAction() {  
    }
```

```
    public ActionForward defaultMethod(ActionMapping mapping,
```

```

        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{
        DynaActionForm dynaform=(DynaActionForm)form;
        CaixaManifestacaoService caixaManifestacaoService =
        (CaixaManifestacaoService)getWebApplicationContext().getBean("caixaManifestacaoService");

        request.setAttribute("manifestacao",caixaManifestacaoService.getDetalhes((Long)dynaform.get("codigo")));
        return mapping.findForward("enviar");
    }

```

```

public ActionForward enviarResposta(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{
        Usuario usuario = (Usuario)request.getSession(false).getAttribute("usuario");
        String destino = "resultado";
        if (usuario != null){
            DynaActionForm dynaform=(DynaActionForm)form;
            CaixaManifestacaoService caixaManifestacaoService =
            (CaixaManifestacaoService)getWebApplicationContext().getBean("caixaManifestacaoService");

            Manifestacao manifestacao =
            caixaManifestacaoService.enviarResposta(usuario, (Long)dynaform.get("codigo"),
            (Integer)dynaform.get("numeroRespostaSelecionada"));

            request.setAttribute("manifestacao",manifestacao);
            request.setAttribute("titulo", "Manifestação Enviada");
            request.setAttribute("texto", "Manifestação enviada");
            if (!manifestacao.getIdentificacao().getNome().equals("Anônima")){
                String nomeModoRetorno = manifestacao.getModoRetorno().getNome();
                boolean retornoEmail = nomeModoRetorno.equals("E-mail");

                ServletContext context = getServletContext();
                RandomAccessFile arquivo = null;

```

```

        if (retornoEmail) {
            arquivo = new
RandomAccessFile(context.getRealPath("docs/resposta.txt"), "r");
        } else {
            arquivo = new
RandomAccessFile(context.getRealPath("docs/resposta.rtf"), "r");
        }
        StringBuffer sbArquivo = new StringBuffer();
        String linha = arquivo.readLine();
        while (linha != null) {
            sbArquivo.append(linha);
            if (retornoEmail) {
                sbArquivo.append("\n");
            }
            linha = arquivo.readLine();
        }
        arquivo.close();

        StringBuffer sbLinha = new StringBuffer();
        StringBuffer sbLinha2 = new StringBuffer();
        StringBuffer sbLinha3 = new StringBuffer();
        String resposta = sbArquivo.toString();

        Manifestante manifestante = manifestacao.getManifestante();
        if (!retornoEmail) {
            if (nomeModoRetorno.equals("Correspondência")) {
                sbLinha.append(manifestante.getEndereco());
                sbLinha.append(" - ");
                sbLinha.append(manifestante.getBairro());

                sbLinha2.append(manifestante.getCep());
                sbLinha2.append(" - ");
                if (manifestante.getOutraCidade() != null) {
                    sbLinha2.append(manifestante.getOutraCidade().getNome());
                } else {

```

```
        sbLinha2.append(manifestante.getCidade().getNome());
    }
    sbLinha2.append(" - ");
    sbLinha2.append(manifestante.getCidade().getEstado().getNome());

    if (manifestante.getTelefone1() != null) {
        sbLinha3.append("Telefone 1: ");
        sbLinha3.append(manifestante.getDddTelefone1());
        sbLinha3.append(" ");
        sbLinha3.append(manifestante.getTelefone1());

        if (manifestante.getTelefone2() != null) {
            sbLinha3.append("    Telefone 2: ");
            sbLinha3.append(manifestante.getDddTelefone2());
            sbLinha3.append(" ");
            sbLinha3.append(manifestante.getTelefone2());
        }
    }
} else {
    if (nomeModoRetorno.equals("Fax")) {
        sbLinha.append("Fax: ");
        sbLinha.append(manifestante.getDddFax());
        sbLinha.append(" ");
        sbLinha.append(manifestante.getFax());

        if (manifestante.getTelefone1() != null) {
            sbLinha2.append("Telefone 1: ");
            sbLinha2.append(manifestante.getDddTelefone1());
            sbLinha2.append(" ");
            sbLinha2.append(manifestante.getTelefone1());

            if (manifestante.getTelefone2() != null) {
                sbLinha3.append("Telefone 2: ");
                sbLinha3.append(manifestante.getDddTelefone2());
```

```

        sbLinha3.append(" ");
        sbLinha3.append(manifestante.getTelefone2());
    }
}
} else {
    sbLinha.append("Telefone 1: ");
    sbLinha.append(manifestante.getDddTelefone1());
    sbLinha.append(" ");
    sbLinha.append(manifestante.getTelefone1());

    if (manifestante.getTelefone2() != null) {
        sbLinha2.append("Telefone 2: ");
        sbLinha2.append(manifestante.getDddTelefone2());
        sbLinha2.append(" ");
        sbLinha2.append(manifestante.getTelefone2());
    }
}
}
}
    resposta = resposta.replaceFirst("=dadosManifestante=",
sbLinha.toString());
    resposta = resposta.replaceFirst("=dadosManifestante2=",
sbLinha2.toString());
    resposta = resposta.replaceFirst("=dadosManifestante3=",
sbLinha3.toString());
}
//Formatar id
NumberFormat nf = NumberFormat.getInstance();
nf.setGroupingUsed(true);
String idFormatado = nf.format(manifestacao.getCodigo());
resposta = resposta.replaceAll("=idManifestacao=", idFormatado);
resposta = resposta.replaceAll("=manifestante=", manifestante.getNome());
SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");
resposta = resposta.replaceAll("=dataCadastro=",
sdf.format(manifestacao.getDataHoraRecebimento()));

resposta = resposta.replaceAll("=classificacao=",

```

```

manifestacao.getClassificacao().getNome());
    if (manifestacao.getOutroAssunto() != null) {
        resposta = resposta.replaceAll("=assunto=",
            manifestacao.getAssunto().getNome() + " - " +
manifestacao.getOutroAssunto());
    } else {
        resposta = resposta.replaceAll("=assunto=",
manifestacao.getAssunto().getNome());
    }
    resposta = resposta.replaceAll("=dataOcorrencia=",
sdf.format(manifestacao.getDataHoraOcorrencia()));
    if (manifestacao.getOutraUnidade() != null) {
        resposta = resposta.replaceAll("=unidadeOcorrencia=",
manifestacao.getUnidadeOcorrencia().getNome() + " - " +
manifestacao.getOutraUnidade().getNome());
    } else {
        resposta = resposta.replaceAll("=unidadeOcorrencia=",
manifestacao.getUnidadeOcorrencia().getNome());
    }
    resposta = resposta.replaceAll("=envolvidos=",
manifestacao.getEnvolvidos());
    resposta = resposta.replaceAll("=descricao=",
manifestacao.getDescricao());

    Envio envio = (Envio)manifestacao.getEnvios().get(0);
    resposta = resposta.replaceAll("=dataEnvio=",
sdf.format(envio.getDataHora()));
    resposta = resposta.replaceAll("=responsavelResposta=",
envio.getResponsavelResposta());
    resposta = resposta.replaceAll("=unidadeResposta=",
envio.getUnidadeResposta().getNome());
    resposta = resposta.replaceAll("=respostaFinal=",
envio.getRespostaFinal());

    if (retornoEmail) {
        Email email = new Email();
        email.setServidor(context.getInitParameter("servidorEmail"));
        email.setEmissor(context.getInitParameter("apelidoEmail") + " <" +

```



```

import br.com.ond.ouvidoria.web.controle.action.*;
import java.util.HashMap;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.DynaActionForm;

public class NovasAction extends BaseAction{

    public NovasAction() {
    }

    public ActionForward defaultMethod(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{
        return getNovas(mapping, form, request, response);
    }

    public ActionForward getNovas(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{
        String destino = "novas";
        Usuario usuario = (Usuario)request.getSession(false).getAttribute("usuario");
        if (usuario != null){
            try {
                CaixaManifestacaoService caixaManifestacaoService =
(CaixaManifestacaoService)getWebApplicationContext().getBean("caixaManifestaca
oService");

request.setAttribute("novas",caixaManifestacaoService.getNovas(usuario.getUnidade
()));

```

```

    } catch (Exception e){
        destino = "erro";
        request.setAttribute("mensagem",e.getMessage());
    }
}
return mapping.findForward(destino);

}

public ActionForward getNova(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{
    Usuario usuario = (Usuario)request.getSession(false).getAttribute("usuario");
    if (usuario == null){
        return null;
    }
    DynaActionForm dynaform=(DynaActionForm)form;
    CaixaManifestacaoService caixaManifestacaoService =
(CaixaManifestacaoService)getWebApplicationContext().getBean("caixaManifestacaoService");
    Manifestacao manifestacao =
caixaManifestacaoService.getNova((Long)dynaform.get("codigo"));
    request.setAttribute("manifestacao",manifestacao);
    if (usuario.getPerfil().getId().longValue() == 3){
        caixaManifestacaoService.setNova(manifestacao.getId(), false);
    }
    HashMap dadosCadastro =
caixaManifestacaoService.getDadosNova(usuario.getUnidade().getId());
    request.setAttribute("unidades",dadosCadastro.get("unidades"));

request.setAttribute("unidadesNivel1SemOutraSemNenhuma",dadosCadastro.get("unidadesNivel1SemOutraSemNenhuma"));
    request.setAttribute("classificacoes",dadosCadastro.get("classificacoes"));
    request.setAttribute("assuntos",dadosCadastro.get("assuntos"));
    return mapping.findForward("nova");
}

```

```

    }

}

..\src\java\br\com\omd\ouvidoria\web\controle\action\caixa\ResponderAction.java
package br.com.omd.ouvidoria.web.controle.action.caixa;

import br.com.omd.ouvidoria.logica.domain.Cargo;
import br.com.omd.ouvidoria.logica.domain.Manifestacao;
import br.com.omd.ouvidoria.logica.domain.Resposta;
import br.com.omd.ouvidoria.logica.domain.Unidade;
import br.com.omd.ouvidoria.logica.domain.Usuario;
import br.com.omd.ouvidoria.logica.service.CaixaManifestacaoService;
import br.com.omd.ouvidoria.web.controle.action.*;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.HashMap;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.DynaActionForm;

public class ResponderAction extends BaseAction{

    public ResponderAction() {
    }

    public ActionForward defaultMethod(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{

```

```

        return montaTela(mapping,form, request, response);
    }
    public ActionForward montaTela(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{
        Usuario usuario = (Usuario)request.getSession(false).getAttribute("usuario");
        if (usuario == null){
            return null;
        }
        DynaActionForm dynaform=(DynaActionForm)form;
        CaixaManifestacaoService caixaManifestacaoService =
        (CaixaManifestacaoService)getWebApplicationContext().getBean("caixaManifestacaoService");

        request.setAttribute("manifestacao",caixaManifestacaoService.getDetalhes(((Long)dynaform.get("codigo"))));
        HashMap dadosResponder =
        caixaManifestacaoService.getDadosResponder(usuario.getUnidade().getId());

        request.setAttribute("unidadeAtualENivelAbaixoSemOutraSemNenhuma",dadosResponder.get("unidadeAtualENivelAbaixoSemOutraSemNenhuma"));
        request.setAttribute("cargos",dadosResponder.get("cargos"));
        return mapping.findForward("responder");
    }

    public ActionForward cadastrarResposta(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{
        Usuario usuario = (Usuario)request.getSession(false).getAttribute("usuario");
        if (usuario == null){
            return null;
        }
        DynaActionForm dynaform=(DynaActionForm)form;

        Manifestacao manifestacao = new Manifestacao();

```

```
manifestacao.setCodigo((Long)dynaform.get("codigo"));
Resposta resposta = new Resposta();
resposta.setCargo(usuario.getCargo());
resposta.setDataHoraAndamento(new Date());
resposta.setOrigem(usuario.getUnidade());
resposta.setUsuario(usuario);
```

```
SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");
resposta.setDataHoraResposta(sdf.parse(dynaform.getString("data")));
Cargo cargoResponsavel = new Cargo();
cargoResponsavel.setId((Long)dynaform.get("cargo"));
resposta.setCargoResponsavel(cargoResponsavel);
resposta.setResponsavelResposta(dynaform.getString("responsavel"));
resposta.setTexto(dynaform.getString("resposta"));
Unidade unidadeResposta = new Unidade();
unidadeResposta.setId((Long)dynaform.get("unidade"));
resposta.setUnidadeResposta(unidadeResposta);
```

```
CaixaManifestacaoService caixaManifestacaoService =
(CaixaManifestacaoService)getWebApplicationContext().getBean("caixaManifestacaoService");
```

```
caixaManifestacaoService.cadastrarResposta(manifestacao, resposta);
```

```
request.setAttribute("titulo", "Resposta Cadastrada");
request.setAttribute("texto", "Resposta cadastrada");
request.setAttribute("manifestacao", manifestacao);
return mapping.findForward("resultado");
```

```
}
```

```
}
```

```
..\src\java\br\com\omd\ouvidoria\web\controle\action\caixa\RespondidasAction.java
package br.com.omd.ouvidoria.web.controle.action.caixa;
```

```
import br.com.omd.ouvidoria.logica.domain.Unidade;
```

```

import br.com.ond.ouvidoria.logica.domain.Usuario;
import br.com.ond.ouvidoria.logica.service.CaixaManifestacaoService;
import br.com.ond.ouvidoria.web.controle.action.*;
import java.util.HashMap;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.DynaActionForm;

public class RespondidasAction extends BaseAction{

    public RespondidasAction() {
    }

    public ActionForward defaultMethod(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{
        return montaTela(mapping, form, request, response);
    }

    public ActionForward montaTela(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{
        DynaActionForm dynaform=(DynaActionForm)form;
        String destino = "respondidas";
        Usuario usuario = (Usuario)request.getSession(false).getAttribute("usuario");

        if (usuario != null){
            CaixaManifestacaoService caixaManifestacaoService =
(CaixaManifestacaoService)getWebApplicationContext().getBean("caixaManifestaca

```

```

oService");
    try {
        request.setAttribute("respondidas",
caixaManifestacaoService.getRespondidas(usuario));
    } catch (Exception e){
        destino = "erro";
        request.setAttribute("mensagem",e.getMessage());
    }
}
return mapping.findForward(destino);
}

```

```

public ActionForward getNova(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{
    Usuario usuario = (Usuario)request.getSession(false).getAttribute("usuario");
    if (usuario == null){
        return null;
    }
    DynaActionForm dynaform=(DynaActionForm)form;
    CaixaManifestacaoService caixaManifestacaoService =
(CaixaManifestacaoService)getWebApplicationContext().getBean("caixaManifestaca
oService");

    request.setAttribute("manifestacao",caixaManifestacaoService.getNova((Long)dynafo
rm.get("id")));

    HashMap dadosCadastro =
caixaManifestacaoService.getDadosNova(usuario.getUnidade().getId());
    request.setAttribute("unidades",dadosCadastro.get("unidades"));

    request.setAttribute("unidadesNivel1SemOutraSemNenhuma",dadosCadastro.get("u
nidadesNivel1SemOutraSemNenhuma"));
    request.setAttribute("classificacoes",dadosCadastro.get("classificacoes"));
    request.setAttribute("assuntos",dadosCadastro.get("assuntos"));
    return mapping.findForward("nova");
}

```

```
}
```

```
..\src\java\br\com\omd\ouvidoria\web\controle\action\relatorio\RelatorioAction.java  
package br.com.omd.ouvidoria.web.controle.action.relatorio;
```

```
import br.com.omd.ouvidoria.logica.domain.Manifestacao;  
import br.com.omd.ouvidoria.logica.service.ManifestacaoService;  
import br.com.omd.ouvidoria.logica.service.RelatorioService;  
import br.com.omd.ouvidoria.util.Data;  
import br.com.omd.ouvidoria.web.controle.action.BaseAction;  
import java.util.Collection;  
import java.util.Date;  
import java.util.Iterator;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import javax.servlet.http.HttpSession;  
import org.apache.struts.action.ActionForm;  
import org.apache.struts.action.ActionForward;  
import org.apache.struts.action.ActionMapping;  
import org.apache.struts.action.DynaActionForm;
```

```
public class RelatorioAction extends BaseAction{
```

```
    public RelatorioAction() {  
    }  
}
```

```
    public ActionForward defaultMethod(ActionMapping mapping,  
        ActionForm form,  
        HttpServletRequest request,
```

```

        HttpServletResponse response) throws Exception{

    DynaActionForm dynaform=(DynaActionForm)form;

    String datainicio = (String)dynaform.get("dataInicial");

    if (datainicio.equals("")) {
        ManifestacaoService manifestacaoService =
(ManifestacaoService)getWebApplicationContext().getBean("manifestacaoService");
        Data dataLimite = new
Data(manifestacaoService.getMenorDataRecebimento());
        request.setAttribute("inicio", dataLimite.getData());
    }

    String datafim = (String)dynaform.get("dataFinal");
    if (datafim.equals("")) {
        Date date_atual=new Date(System.currentTimeMillis());
        Data dataLimite = new Data(date_atual);
        request.setAttribute("fim", dataLimite.getData());
    }

    return mapping.findForward("success");
}

public ActionForward definePeriodo(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{
    String paginaDestino = "erro";
    try {
        //inclui Manifestação na sessão
        HttpSession session = request.getSession(false);
        if (session == null) {
            response.sendRedirect("../sair.do");
        }
    }
}

```

```

}
String data = request.getParameter("dataInicial");
session.setAttribute("inicio", data);
String mensagem = null;
if (!data.equals("")) {
    try {
        data = this.converteData2(data);
        session.setAttribute("dataInicial", data);
    } catch (StringIndexOutOfBoundsException s ) {
        session.setAttribute("dataInicial", null);
        mensagem = "Formato da Data Inicial inválido, a Data Inicial deve obedecer o formato dd/mm/aaaa.";
        request.setAttribute("mensagem", mensagem);
    }
} else {
    session.setAttribute("dataInicial", null);
}
if (mensagem == null) {
    data = request.getParameter("dataFinal");
    session.setAttribute("fim", data);
    if (!data.equals("")) {
        try {
            data = this.converteData2(data);
            session.setAttribute("dataFinal", data);
        } catch (StringIndexOutOfBoundsException s ) {
            session.setAttribute("dataFinal", null);
            mensagem = "Formato da Data Inicial inválido, a Data Final deve obedecer o formato dd/mm/aaaa.";
            request.setAttribute("mensagem", mensagem);
        }
    } else {
        session.setAttribute("dataFinal", null);
    }
}
if (mensagem == null) {
    paginaDestino = "resultado";
}

```

```

        }
    }
} catch (Exception e) {
    request.setAttribute("mensagem", e.getMessage());
}

return mapping.findForward(paginaDestino);
}

```

```

public String converteData2(String d) throws Exception {
    String temp = "";
    try {
        temp = d.substring(6,10);
        temp = temp + "-" + d.substring(3,5);
        temp = temp + "-" + d.substring(0,2);
    } catch (Exception e) {
        throw e;
    }
    return temp;
}

```

```

}

```

```

..\src\java\br\com\omd\ouvidoria\web\controle\action\relatorio\RelatorioAssuntoAction
.java

```

```

package br.com.omd.ouvidoria.web.controle.action.relatorio;

```

```

import br.com.omd.ouvidoria.logica.domain.Classificacao;
import br.com.omd.ouvidoria.logica.domain.Unidade;
import br.com.omd.ouvidoria.logica.service.ManifestacaoService;
import br.com.omd.ouvidoria.logica.service.RelatorioService;
import br.com.omd.ouvidoria.logica.service.UnidadeService;
import br.com.omd.ouvidoria.util.Data;
import br.com.omd.ouvidoria.web.controle.action.BaseAction;
import java.util.Collection;

```

```
import java.util.Date;
import java.util.LinkedHashMap;
import java.util.Set;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.DynaActionForm;
```

```
public class RelatorioAssuntoAction extends BaseAction{
```

```
    public RelatorioAssuntoAction() {
    }
}
```

```
    public ActionForward defaultMethod(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{
```

```
        DynaActionForm dynaform=(DynaActionForm)form;
        LinkedHashMap manifestacoesPorAssunto=null;
        String usuarioUnidadeId=null;
```

```
        usuarioUnidadeId=request.getParameter("unidade");
        String dataInicialForm=(String)dynaform.get("dataInicial");
        String dataFinalForm=(String)dynaform.get("dataFinal");
        Unidade unidade=null;
```

```
        UnidadeService unidadeService =
(UnidadeService)getWebApplicationContext().getBean("unidadeService");
```

```

if(usuarioUnidadeId!=null){
    unidade=new Unidade();
    unidade.setId(Long.valueOf(usuarioUnidadeId));
    request.setAttribute("unidadeParametro",
unidadeService.selectPorId(unidade));
}

```

```

RelatorioService relatorioService =
(RelatorioService)getWebApplicationContext().getBean("relatorioService");

```

```

Data dataIni=null;
Data dataFim=null;
if (!dataInicialForm.equals("")){
    String di=dataInicialForm.substring(0,2);
    String mi=dataInicialForm.substring(3,5);
    String ai=dataInicialForm.substring(6,10);
    dataIni=new
Data(Integer.parseInt(di),Integer.parseInt(mi),Integer.parseInt(ai));
} else{
    ManifestacaoService manifestacaoService =
(ManifestacaoService)getWebApplicationContext().getBean("manifestacaoService");
    dataIni = new Data(manifestacaoService.getMenorDataRecebimento());
    request.setAttribute("inicio", dataIni.getData());
}
if(!dataFinalForm.equals("")){
    String df=dataFinalForm.substring(0,2);
    String mf=dataFinalForm.substring(3,5);
    String af=dataFinalForm.substring(6,10);
    dataFim=new
Data(Integer.parseInt(df),Integer.parseInt(mf),Integer.parseInt(af));
} else{
    dataFim=new Data(new Date(System.currentTimeMillis()));
    request.setAttribute("fim", dataFim.getData());
}

```

```

manifestacoesPorAssunto=relatorioService.getAssuntos(unidade,dataIni,dataFim);
    if (manifestacoesPorAssunto!=null){
        Set chavesClassificacao=manifestacoesPorAssunto.keySet();

request.setAttribute("chavesListasManifestacaoPorAssunto",chavesClassificacao);
    }
    request.setAttribute("listasManifestacaoPorAssunto",
manifestacoesPorAssunto);
    if(manifestacoesPorAssunto!=null){
        request.setAttribute("tamanholistasManifestacaoPorAssunto",new
Integer(manifestacoesPorAssunto.size()));
    } else{
        request.setAttribute("tamanholistasManifestacaoPorAssunto",new Integer(0));
    }

    return mapping.findForward("success");
}
}

```

```

..\src\java\br\com\omd\ouvidoria\web\controle\action\relatorio\RelatorioClassificacao
Action.java

```

```

package br.com.omd.ouvidoria.web.controle.action.relatorio;

```

```

import br.com.omd.ouvidoria.logica.domain.Unidade;
import br.com.omd.ouvidoria.logica.service.ManifestacaoService;
import br.com.omd.ouvidoria.logica.service.RelatorioService;
import br.com.omd.ouvidoria.logica.service.UnidadeService;
import br.com.omd.ouvidoria.util.Data;
import br.com.omd.ouvidoria.web.controle.action.BaseAction;
import java.util.Collection;
import java.util.Date;
import java.util.LinkedHashMap;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

```

```
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.DynaActionForm;
```

```
public class RelatorioClassificacaoAction extends BaseAction{
```

```
    public RelatorioClassificacaoAction() {
    }
}
```

```
    public ActionForward defaultMethod(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{
```

```
        DynaActionForm dynaform=(DynaActionForm)form;
        Collection manifestacoesPorClassificacao=null;
```

```
        String usuarioUnidadeId=null;
        usuarioUnidadeId=request.getParameter("unidade");
```

```
        String dataInicialForm=(String)dynaform.get("dataInicial");
        String dataFinalForm=(String)dynaform.get("dataFinal");
        Unidade unidade=null;
        UnidadeService unidadeService =
        (UnidadeService)getWebApplicationContext().getBean("unidadeService");
```

```
        if(usuarioUnidadeId!=null){
            unidade=new Unidade();
```

```
        unidade.setId(Long.valueOf(usuarioUnidadeId));
        request.setAttribute("unidadeParametro",
unidadeService.selectPorId(unidade));
    }
```

```
RelatorioService relatorioService =
(RelatorioService)getWebApplicationContext().getBean("relatorioService");
```

```
Data dataIni=null;
Data dataFim=null;
if (!dataInicialForm.equals("")){
    String di=dataInicialForm.substring(0,2);
    String mi=dataInicialForm.substring(3,5);
    String ai=dataInicialForm.substring(6,10);
    dataIni=new
Data(Integer.parseInt(di),Integer.parseInt(mi),Integer.parseInt(ai));
} else{
    ManifestacaoService manifestacaoService =
(ManifestacaoService)getWebApplicationContext().getBean("manifestacaoService");
    dataIni = new Data(manifestacaoService.getMenorDataRecebimento());
    request.setAttribute("inicio", dataIni.getData());
}
if(!dataFinalForm.equals("")){
    String df=dataFinalForm.substring(0,2);
    String mf=dataFinalForm.substring(3,5);
    String af=dataFinalForm.substring(6,10);
    dataFim=new
Data(Integer.parseInt(df),Integer.parseInt(mf),Integer.parseInt(af));
} else{
    dataFim=new Data(new Date(System.currentTimeMillis()));
    request.setAttribute("fim", dataFim.getData());
}
```

```
manifestacoesPorClassificacao=relatorioService.getClassificacao(unidade,dataIni,da
taFim);
```

```
        request.setAttribute("listasManifestacaoPorClassificacao",
manifestacoesPorClassificacao);
        if(manifestacoesPorClassificacao!=null){
            request.setAttribute("tamanholistasManifestacaoPorClassificacao",new
Integer(manifestacoesPorClassificacao.size()));
        }else{
            request.setAttribute("tamanholistasManifestacaoPorClassificacao",new
Integer(0));
        }
        return mapping.findForward("success");
    }
}
}
```

```
..\src\java\br\com\omd\ouvidoria\web\controle\action\relatorio\RelatorioManifestacao
Action.java
```

```
package br.com.omd.ouvidoria.web.controle.action.relatorio;
```

```
import br.com.omd.ouvidoria.logica.domain.Assunto;
import br.com.omd.ouvidoria.logica.domain.Classificacao;
import br.com.omd.ouvidoria.logica.domain.Unidade;
import br.com.omd.ouvidoria.logica.dto.ConsultaManifestacaoDTO;
import br.com.omd.ouvidoria.logica.service.AssuntoService;
import br.com.omd.ouvidoria.logica.service.ManifestacaoService;
import br.com.omd.ouvidoria.logica.service.RelatorioService;
import br.com.omd.ouvidoria.logica.service.UnidadeService;
import br.com.omd.ouvidoria.util.Data;
import br.com.omd.ouvidoria.web.controle.action.BaseAction;
import java.util.Collection;
import java.util.Date;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import org.apache.struts.action.ActionForm;
```

```
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.DynaActionForm;
```

```
public class RelatorioManifestacaoAction extends BaseAction{
```

```
    public RelatorioManifestacaoAction() {
    }
```

```
    public ActionForward defaultMethod(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{
```

```
        DynaActionForm dynaform=(DynaActionForm)form;
        Collection manifestacoes=null;
        String usuarioUnidadeId=null;
        String usuarioAssuntoId=null;
        String classificacaoAssuntoId=null;
        Unidade unidade=null;
        Classificacao classificacao=null;
        Assunto assunto=null;
        Data dataIni=null;
        Data dataFim=null;
```

```
        String unidadeId=request.getParameter("unidade");
        String assuntoId=request.getParameter("assunto");
        String classificacaoId=request.getParameter("classificacao");
```

```
        String dataInicialForm=(String)dynaform.get("dataInicial");
```

```

String dataFinalForm=(String)dynaform.get("dataFinal");

if(unidadeId!=null){
    unidade=new Unidade();
    unidade.setId(Long.valueOf(unidadeId));
}

if(classificacaoId!=null){
    classificacao=new Classificacao();
    classificacao.setId(Long.valueOf(classificacaoId));
}

if(assuntoId!=null){
    assunto=new Assunto();
    assunto.setId(Long.valueOf(assuntoId));
}

if (!dataInicialForm.equals("")){
    String di=dataInicialForm.substring(0,2);
    String mi=dataInicialForm.substring(3,5);
    String ai=dataInicialForm.substring(6,10);
    dataIni=new
Data(Integer.parseInt(di),Integer.parseInt(mi),Integer.parseInt(ai));
} else{
    ManifestacaoService manifestacaoService =
(ManifestacaoService)getWebApplicationContext().getBean("manifestacaoService");
    dataIni = new Data(manifestacaoService.getMenorDataRecebimento());
    request.setAttribute("inicio", dataIni.getData());
}
if(!dataFinalForm.equals("")){
    String df=dataFinalForm.substring(0,2);
    String mf=dataFinalForm.substring(3,5);

```

```
        String af=dataFinalForm.substring(6,10);
        dataFim=new
Data(Integer.parseInt(df),Integer.parseInt(mf),Integer.parseInt(af));
    }else{
        dataFim=new Data(new Date(System.currentTimeMillis()));
        request.setAttribute("fim", dataFim.getData());
    }
}
```

```
RelatorioService relatorioService =
(RelatorioService)getWebApplicationContext().getBean("relatorioService");
```

```
manifestacoes=relatorioService.getManifestacoes(classificacao,unidade,assunto,dat
alni,dataFim);
```

```
        request.setAttribute("resultado", manifestacoes);
        return mapping.findForward("success");
    }
}
}
```

```
..\src\java\br\com\omd\ouvidoria\web\controle\action\relatorio\RelatorioUnidadeAction
.java
```

```
package br.com.omd.ouvidoria.web.controle.action.relatorio;
```

```
import br.com.omd.ouvidoria.logica.domain.Unidade;
import br.com.omd.ouvidoria.logica.service.ManifestacaoService;
import br.com.omd.ouvidoria.logica.service.RelatorioService;
import br.com.omd.ouvidoria.logica.service.UnidadeService;
import br.com.omd.ouvidoria.util.Data;
import br.com.omd.ouvidoria.web.controle.action.BaseAction;
import java.util.Collection;
import java.util.Date;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
```

```

import org.apache.log4j.varia.NullAppender;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.DynaActionForm;

public class RelatorioUnidadeAction extends BaseAction{

    public RelatorioUnidadeAction() {
    }

    public ActionForward defaultMethod(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception{

        DynaActionForm dynaform=(DynaActionForm)form;
        Collection listasUnidades=null;
        String usuarioUnidadeId=null;

        usuarioUnidadeId=request.getParameter("unidade");
        String dataInicialForm=(String)dynaform.get("dataInicial");
        String dataFinalForm=(String)dynaform.get("dataFinal");

        Unidade unidade=null;
        UnidadeService unidadeService =
(UnidadeService)getWebApplicationContext().getBean("unidadeService");
        if(usuarioUnidadeId!=null){
            unidade=new Unidade();
            unidade.setId(Long.valueOf(usuarioUnidadeId));
            request.setAttribute("unidadeParametro",

```

```

unidadeService.selectPorId(unidade));
    }

    RelatorioService relatorioService =
(RelatorioService)getWebApplicationContext().getBean("relatorioService");

    Data dataIni=null;
    Data dataFim=null;
    if (!dataInicialForm.equals("")){
        String di=dataInicialForm.substring(0,2);
        String mi=dataInicialForm.substring(3,5);
        String ai=dataInicialForm.substring(6,10);
        dataIni=new
Data(Integer.parseInt(di),Integer.parseInt(mi),Integer.parseInt(ai));
    } else{
        ManifestacaoService manifestacaoService =
(ManifestacaoService)getWebApplicationContext().getBean("manifestacaoService");
        dataIni = new Data(manifestacaoService.getMenorDataRecebimento());
        request.setAttribute("inicio", dataIni.getData());
    }
    if(!dataFinalForm.equals("")){
        String df=dataFinalForm.substring(0,2);
        String mf=dataFinalForm.substring(3,5);
        String af=dataFinalForm.substring(6,10);
        dataFim=new
Data(Integer.parseInt(df),Integer.parseInt(mf),Integer.parseInt(af));
    }else{
        dataFim=new Data(new Date(System.currentTimeMillis()));
        request.setAttribute("fim", dataFim.getData());
    }

    String unidade_id=(String)request.getParameter("unidade");
    Unidade unidadePesquisa=null;
    if (unidade_id!=null){
        unidadePesquisa=new Unidade();

```

```

        unidadePesquisa.setId(Long.valueOf(unidade_id));
    }

listasUnidades=relatorioService.getUnidades(unidadePesquisa,dataIni,dataFim);

    request.setAttribute("listasUnidades", listasUnidades);
    if(listasUnidades!=null){
        request.setAttribute("tamanholistasUnidades",new Integer(
listasUnidades.size()));
    }else{
        request.setAttribute("tamanholistasUnidades",new Integer(0));
    }

    return mapping.findForward("success");
}
}

```

```

}

```

```

..\src\java\br\com\omd\ouvidoria\web\menu\ContaManifestacoes.java
package br.com.omd.ouvidoria.web.menu;

```

```

import java.sql.*;
public abstract class ContaManifestacoes
{
    protected Connection conexao;
    protected int unidade;

    /**
     * Constructor for objects of class ContaManifestacoes
     */
    public ContaManifestacoes() {
    }
    public void setConexao(Connection conexao) {

```

```

        this.conexao = conexao;
    }
    public void setUnidade(int unidade) {
        this.unidade = unidade;
    }
    public ContaManifestacoes(Connection conexao, int unidade){
        this.conexao = conexao;
        this.unidade = unidade;
    }
    public abstract int getNovas() throws SQLException;
    public abstract int getRespondidas() throws SQLException;
    public abstract int getAguardandoResposta() throws SQLException;
    public abstract int getaRevisar() throws SQLException;
    public abstract int getEncerradas() throws SQLException;
    protected int conta(PreparedStatement pstmt) throws SQLException{
        int quantidade = 0;
        try {
            ResultSet rs = pstmt.executeQuery();
            if (rs.next()){
                quantidade = rs.getInt("Qtde");
            }
        } catch(SQLException e) {
            throw e;
        }
        return quantidade;
    }
}

```

```

..\src\java\br\com\omd\ouvidoria\web\menu\ContaManifestacoesOuvidor.java
package br.com.omd.ouvidoria.web.menu;

```

```

import java.sql.*;
public class ContaManifestacoesOuvidor extends ContaManifestacoes {

```

```

public ContaManifestacoesOuvidor(Connection conexao, int unidade){
    super(conexao,unidade);
}
public ContaManifestacoesOuvidor(){
    super();
}
public int getNovas() throws SQLException {
    int quantidade = 0;
    try {
        PreparedStatement pstmt = conexao.prepareStatement("SELECT
count(Distinct(ID)) as Qtde " +
                "FROM Manifestacoes WHERE Nova='s' and
UnidadeResponsavel = ?");
        pstmt.setInt(1, unidade);
        return this.conta(pstmt);
    } catch(SQLException e) {
        throw e;
    }
}

public int getRespondidas() throws SQLException {
    int quantidade = 0;
    try {
        PreparedStatement pstmt = conexao.prepareStatement("SELECT
count(Distinct(ID)) as Qtde " +
                "FROM Manifestacoes WHERE Envio = 0 and ((Comentario
= 0 and " +
                "CadastroResposta1 <> 0) or CadastroResposta2 <> 0 or " +
                "CadastroResposta3 <> 0)");
        return this.conta(pstmt);
    } catch(SQLException e) {
        throw e;
    }
}

```

```

public int getAguardandoResposta() throws SQLException {
    try {
        PreparedStatement pstmt = conexao.prepareStatement("SELECT
count(Distinct(ID)) as Qtde " +
                "FROM Manifestacoes WHERE CadastroResposta1 = 0 and
not (Nova='s' and UnidadeResponsavel=2)");
        return this.conta(pstmt);
    } catch(SQLException e) {
        throw e;
    }
}

public int getaRevisar() throws SQLException {
    try {
        PreparedStatement pstmt = conexao.prepareStatement("SELECT
count(Distinct(ID)) as Qtde " +
                "FROM Manifestacoes WHERE CadastroResposta1 <> 0
and " +
                "Comentario <> 0 and CadastroResposta2 = 0 and Envio =
0");
        return this.conta(pstmt);
    } catch(SQLException e) {
        throw e;
    }
}

public int getEncerradas() throws SQLException {
    try {
        PreparedStatement pstmt = conexao.prepareStatement("SELECT
count(Distinct(ID)) as Qtde " +
                "FROM Manifestacoes WHERE Envio <> 0");
        return this.conta(pstmt);
    } catch(SQLException e) {
        throw e;
    }
}
}

```

```
..\src\java\br\com\omd\ouvidoria\web\menu\ContaManifestacoesRepresentante.java  
package br.com.omd.ouvidoria.web.menu;
```

```
import java.sql.*;  
public class ContaManifestacoesRepresentante extends ContaManifestacoes {  
    protected int usuario;  
    public ContaManifestacoesRepresentante(Connection conexao, int unidade, int  
usuario){  
        super(conexao,unidade);  
        this.usuario = usuario;  
    }  
    public ContaManifestacoesRepresentante(){  
    }  
    public void setUsuario(int usuario) {  
        this.usuario = usuario;  
    }  
    public int getNovas() throws SQLException {  
        int quantidade = 0;  
        try {  
            PreparedStatement pstmt = conexao.prepareStatement("SELECT  
count(Distinct(ID)) as Qtde " +  
                "FROM Manifestacoes WHERE Nova='s' and  
UnidadeResponsavel = ?");  
            pstmt.setInt(1, unidade);  
            return this.conta(pstmt);  
        } catch(SQLException e) {  
            throw e;  
        }  
    }  
    public int getaRevisar() throws SQLException {  
        try {  
            PreparedStatement pstmt = conexao.prepareStatement("SELECT  
count(Distinct(ID)) as Qtde " +  
                "FROM Manifestacoes WHERE CadastroResposta1 <> 0 " +  
                "and Comentario <> 0 and CadastroResposta2 = 0 and
```

```

Envio=0 and " +
        "UnidadeResponsavel = ? ");
        pstmt.setInt(1, unidade);
        return this.conta(pstmt);
    } catch(SQLException e) {
        throw e;
    }
}

public int getAguardandoResposta() throws SQLException {
    try {
        PreparedStatement pstmt = conexao.prepareStatement("SELECT
count(Distinct(ID)) as Qtde " +
        "FROM Manifestacoes WHERE Nova='n' and
CadastroResposta1 = 0 and " +
        "UnidadeResponsavel = ?");
        pstmt.setInt(1, unidade);
        return this.conta(pstmt);
    } catch(SQLException e) {
        throw e;
    }
}

public int getRespondidas() throws SQLException {
    int quantidade = 0;
    try {
        PreparedStatement pstmt = conexao.prepareStatement("SELECT
count(Distinct(m.ID)) as Qtde " +
        "FROM Manifestacoes m LEFT JOIN CadastrosResposta c
ON " +
        "m.CadastroResposta1 = c.ID OR m.CadastroResposta2 =
c.ID WHERE " +
        "Envio = 0 and c.usuario = ?");
        pstmt.setInt(1, usuario);
        return this.conta(pstmt);
    } catch(SQLException e) {
        throw e;
    }
}

```

```

    }
    public int getEncerradas() throws SQLException {
        try {
            PreparedStatement pstmt = conexao.prepareStatement("SELECT
count(Distinct(m.ID)) as Qtde " +
                "FROM Manifestacoes m JOIN Envios e ON m.envio = e.id "
+
                "JOIN Unidades u ON u.id = e.unidadeResposta JOIN
Unidades u2 ON u.vinculo = u2.id " +
                "WHERE e.unidadeResposta = ? or u.vinculo = ?");
            pstmt.setInt(1, unidade);
            pstmt.setInt(2, unidade);
            return this.conta(pstmt);
        } catch(SQLException e) {
            throw e;
        }
    }
}

```