UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE SISTEMAS DE INFORMAÇÃO

Avaliação do Framework Struts para Implementação de Aplicações Web usando Padrão Modelo-Visão-Controlador

Aluna: Alice Alves Corrêa
Orientador: Frank Siqueira

Trabalho de Conclusão de Curso
apresentado como parte dos requisitos para
obtenção do grau de Bacharel em Sistemas
de Informação

Florianópolis - SC

2004/2.

Avaliação do Framework Struts para Implementação de Aplicações Web usando Padrão Modelo-Visão-Controlador

Alice Alves Corrêa

Trabalho de Conclusão de Curso
apresentado como parte dos requisitos
para obtenção o grau de
Bacharel em Sistemas de Informação

Orientador
Frank Augusto de Siqueira

_____

Banca Examinadora

Ricardo Pereira e Silva

_____

João Bosco Mangueira Sobral

_____

Sumário

Lista de Figuras

Resumo

Este trabalho de conclusão de curso analisa os aspectos específicos do Framework Struts e propõe uma implementação para a sua utilização. O padrão Modelo-Visão-Controlador é detalhado e são relatadas as classes de extensão do Framework. São esclarecidos detalhes da configuração para implantação do Framework, que complementam as especificações das classes de extensão.

O trabalho propõe um modelo para a implementação de aplicações, incluindo a definição das atividades e das ferramentas utilizadas, bem como seus diagramas de modelagem. São relacionadas as capacidades e fraquezas do Framework, bem como aspectos relacionados a análise de custo de implantação para empresas.

1. Introdução

1.1. Descrição do Problema

A criação da *World Wide Web* (WWW) em 1993 permitiu acesso a conteúdos estáticos – textos, figuras, tabelas, etc. – estruturados na forma de páginas Web interconectadas por *links*. A popularização do acesso à Internet e a aceitação da Web por parte dos usuários como meio de difusão de informações fez com que a Web se tornasse cada vez mais ágil e dinâmica.

De modo a facilitar a geração e disponibilização de conteúdos cada vez mais dinâmicos e voltados aos interesses dos usuários, e visando a integração da Web com os sistemas computacionais pré-existentes, surgiram formas de disponibilizar conteúdos dinâmicos para o usuário. Com isto, o usuário deixou de ser um ente praticamente passivo no sistema - que simplesmente navegava pelos conteúdos - e passou a ser um ente ativo, influenciando a forma como os conteúdos que lhe são mostrados são gerados e atualizados.

A disponibilização de conteúdo dinâmico para os usuários da Web tornou-se possível inicialmente com a utilização de scripts escritos em alguma linguagem computacional, como C e Perl, através das quais era gerado o conteúdo a ser exibido por uma página Web com base em informações fornecidas pelo usuário ou armazenadas no servidor. No entanto, estes scripts passaram a se mostrar inadequados para lidar com a complexidade de novas aplicações centradas na Web, como sistemas de *home banking* e de comércio eletrônico. Como conseqüências destas limitações, foram criadas novas linguagens voltadas para a geração de conteúdo dinâmico para a Web, como PHP, ASP e JSP. Tais linguagens, apesar de especialmente voltadas para a Web, em geral resultam na criação de código extremamente difícil de manter e de reutilizar. Estas dificuldades se devem em grande parte à não existência de uma separação clara entre o código responsável pela apresentação dos dados, o código que coordena a execução de transações (ou seja, que implementa a lógica de negócios) e o código responsável por interagir com os bancos de dados.

O Jakarta Struts consiste em um arcabouço (*framework*) de código aberto (*open source*) patrocinado pela *Apache Software Foundation* que visa facilitar a manutenção de aplicações Web e a reutilização de seu código. Struts fornece um valioso suporte

para a criação de aplicações Web e está rapidamente se transformando em uma ferramenta bastante conhecida e utilizada pelos desenvolvedores deste tipo de aplicação.

Struts foi projetado com a intenção de ser um framework para a criação de Aplicações Web que facilmente separa a camada de apresentação das camadas de transações e de persistência. Sua implementação foi realizada na linguagem Java tendo como base o padrão de projeto (*design pattern*) Modelo-Visão-Controlador (MVC – *Model-View-Controller*).

O Framework Struts impõe a utilização de um conjunto de regras bem definidas para delimitar camadas de interface, lógica de negócios e persistência com o objetivo de integrá-las, deixando que o desenvolvedor tenha controle do fluxo da lógica e conseqüentemente permitindo o gerenciamento do problema.

1.1.1 Desenvolvimento para a Internet nas Empresas

Grande parte do desenvolvimento de software está migrando para o ambiente da Internet. Nas empresas, a primeira "aparição" na Internet acontecia por meio de páginas estáticas desenvolvidas pela equipe do departamento de marketing ao invés da equipe de desenvolvimento de sistemas. Com o tempo, a necessidade de conteúdo dinâmico, e de formulários para comunicação com clientes e fornecedores e a necessidade de integração com sistemas legados fez com que a equipe de desenvolvimento precisasse participar do processo.

Ao mesmo tempo, o desenvolvimento está sendo realizado com novas tecnologias e ferramentas de construção. Novas arquiteturas surgem e o ritmo de inovação é enorme.

O desenvolvimento para a Internet exige conhecimentos específicos e requer adaptação dos desenvolvedores. Entretanto o problema é que as empresas são incapazes de aprender com os erros do passado. Entre eles podemos citar: sistemas de complexidade crescente, sem arquitetura, sem documentação, construídos sem planejamento, analise e design, e que posteriormente terão um altíssimo custo de manutenção. (BELLOQUIM, 2004)

1.2 Objetivos do Trabalho

Este trabalho tem como principal objetivo avaliar a utilização do Struts para a implementação de uma aplicação Web segundo o Padrão Modelo-Visão-Controlador,

avaliando o desacoplamento entre a interface do usuário, lógica de negócios e camada de persistência.

De modo complementar, este trabalho objetiva ainda:

- Conhecer o framework Struts, seus componentes e a forma que eles interagem e as suas formas de utilização para desenvolvimento de aplicações Web;
- Avaliar a complexidade envolvida no desenvolvimento de aplicações Web utilizando Struts;
- Avaliar os requisitos para o desenvolvimento de aplicações Web utilizando Struts.
- Detectar eventuais limitações do framework Struts.

1.3 Justificativa

O desenvolvimento de aplicações centradas na Web tornou-se em pouco tempo um dos ramos da indústria de software mais ativos, devido ao grande número de empresas e instituições que disponibilizam serviços na Web. Muitas empresas – como as internacionalmente conhecidas Amazon, eBay e Yahoo, e as brasileiras Submarino, Grupos e Web Motors, para citar apenas algumas – surgiram a partir da possibilidade de explorar todo o potencial de vendas de produtos e serviços através da Internet. No entanto, algumas destas empresas ainda encontram dificuldades em obter lucros a partir dos negócios efetuados através da rede, em grande parte devido aos altos custos de desenvolvimento, operação, manutenção e atualização de seus Web Sites.

Em particular, os Web Sites desenvolvidos profissionalmente exigem uma grande robustez, confiabilidade e disponibilidade. A manutenção de uma página com muitas linhas de código acaba se tornando dispendiosa, pois é complicado verificar o fluxo dos dados bem como a estrutura do *layout* da página. O acoplamento entre páginas também pode ser considerado um fator preocupante, uma vez que se uma página for alterada, podem ser necessários ajustes na navegação de várias outras páginas. Conseqüentemente, essas exigências acabam por aumentar a complexidade do fluxo das informações, tornando a manutenção das páginas ainda mais difícil.

Dentre as principais razões para avaliarmos a utilização do Struts podemos destacar a possibilidade de redução dos custos envolvidos na disponibilização de aplicações que tem como base a *World Wide Web*. A simples redução destes custos

pode tornar um negócio viável, o que pode acabar por propiciar dividendos para uma empresa e gerar novos empregos.

Outro aspecto importante a ser destacado advém do fato de os desenvolvedores estarem mais interessados em criar interfaces e funcionalidades bem definidas e reusáveis do que em definir interfaces de usuário. Com a utilização do framework Struts, o trabalho pode ser dividido entre desenvolvedores e *designers*, evitando assim que o próprio programador seja responsável também pela definição do *layout* das páginas.

Com a utilização do Struts, o processo de internacionalização de páginas Web é simplificado enormemente devido à divisão existente entre a interface com o usuário e a lógica da aplicação e o acesso aos dados. Isto pode ser considerado um diferencial competitivo importante para os desenvolvedores de software para empresas com interesse no mercado global.

Pode-se citar como fator importante que contribui para estimular a utilização do *framework* o fato deste se tratar de um software livre e de código aberto (*open source*), o que além de promover a confiabilidade, acaba trazendo benefícios como robustez, segurança e velocidade de evolução além de escalabilidade, reutilização, interoperabilidade, integração e menor custo.

Adicionalmente, é importante ressaltar que a existência no *framework* Struts de mecanismos para validação de entradas na Web pode ser considerada como um importante complemento das tecnologias para desenvolvimento de aplicações Web.

1.4 Organização do Texto

Este documento é organizado da seguinte maneira:

- No capítulo 1 – Introdução – foram apresentadas as características do trabalho em linhas gerais;
- No capítulo 2 – Tecnologias para Desenvolvimento Web – serão descritas as tecnologias relacionadas ao contexto do trabalho;
- No capítulo 3 – Padrões de Projeto – são descritas as características do padrão de projetos Modelo-Visão-Controlador.
- No capítulo 4 – O Framework Struts – serão descritos detalhadamente os principais componentes do Framework.

- No capítulo 5 – Desenvolvimento de uma Aplicação usando Struts – será descrita a modelagem de uma aplicação que será utilizada para avaliar a adequação do framework Struts para desenvolvimento de aplicações Web;
- No capítulo 6 – Implementação – Serão apresentadas considerações sobre a implementação da aplicação, bem como uma amostra do funcionamento da mesma.
- No capítulo 7 – Análise do Framework – será feita a análise dos framework, mostrando os seus pontos positivos e negativos, e serão discutidas questões como custo, escalabilidade e desempenho.
- No capítulo 8 – Conclusão – são sumarizados os resultados e as contribuições deste trabalho.

2. Tecnologias para Desenvolvimento Web

Inicialmente a *World Wide Web* foi projetada para lidar com documentos estáticos. Páginas Web e figuras eram disponibilizadas pelos desenvolvedores, e todos os clientes Web, ao acessarem o servidor que continha esta página, tinham acesso ao mesmo conteúdo. Com o passar do tempo, a Web demonstrou ser uma ferramenta poderosa para oferecer conteúdo dinâmico, customizado de acordo com a identidade e as preferências do usuário. Diversas tecnologias foram propostas para este fim. Neste capítulo analisaremos as principais tecnologias destinadas ao desenvolvimento de aplicações Web.

2.1 Common Gateway Interface (CGI)

A *Common Gateway Interface* (CGI) surgiu com o propósito de permitir que os servidores interagissem com aplicações externas de uma maneira pela qual as páginas hipertexto não tivessem mais que ser exclusivamente estáticas.

Uma aplicação CGI pode retornar valores de um banco de dados e colocar esses resultados em uma tabela dentro de um documento, bem como dados informados em uma página de hipertexto podem ser inseridos em um banco de dados. Essa tecnologia mostrou infinitas possibilidades e iniciou de fato a popularização da Internet, permitindo uma interação dinâmica com os usuários. (JONHSON, STEARNS, SINGH et. al, 2002)

Apesar das aplicações CGI terem demonstrado a sua importância, estas possuíam sérias limitações em sua implementação. Para cada requisição de CGI enviada ao servidor, é criado um novo processo do sistema operacional. Uma vez que o processo termina a sua execução, sua saída deve ser enviada pelo servidor Web ao navegador (*browser*) da máquina cliente que efetuou a requisição. Este constante início e término de processos torna-se pesado e ineficiente. É possível imaginar como o tempo de resposta pode ser ruim se centenas de usuários efetuarem requisições simultâneas. Outra limitação do CGI é a difícil interligação com outros estágios de processamento de requisições, pois este está rodando em um processo separado no servidor web. Desta forma é difícil manipular fluxo de informações, autorizações e autenticação (*login*) de usuários.

Algumas alternativas ao CGI foram propostas, como a FastCGI, uma extensão independente de linguagem que não tem o mesmo modelo de processamento que o CGI padrão. É capaz de criar um único processo para cada programa FastCGI, permitindo

múltiplas requisições para o mesmo espaço de processo. Entretanto , quando vários clientes interagem simultaneamente, o programa necessita criar um gerenciador de processos para manipular cada requisição. Outro problema do FastCGI é que ele é portável apenas nas linguagens em que ele é escrito. Outras alternativas para o CGI incluem o mod_perl para servidores Apache, NSAPI para o Netscape e ISAPI para servidor web Microsoft IIS. Enquanto essas soluções oferecem sempre um melhor desempenho e escalabilidade, eles ainda possuem problemas de portabilidade. (JONHSON, STEARNS, SINGH et. al, 2002)

2.2 Tecnologias de Páginas de Servidor

Páginas de Servidor foram criadas para tornar as apresentações das páginas muito mais fáceis de serem criadas e mantidas, principalmente por não programadores. Podemos citar alguns exemplos de tecnologias: páginas ASP, páginas Coldfusion, páginas PHP, Include Server Side, templates Velocity e as JSPs. Todas elas têm o mesmo princípio (HUSTED, DUMOULIN, FRANCISCUS et. al, 2003):

- Uma pagina HTML é criada com trechos de código que serão executados no servidor.
- Quando uma solicitação para a página dinâmica é recebida, a página é usada para construir uma resposta dinâmica.
- A resposta é retornada como uma html padrão.

As principais vantagens do uso dessas tecnologias são as seguintes:

- A página do servidor não é um arquivo de programa compilado na aplicação básica.
- A sintaxe da página do servidor lembra uma página HTML convencional.
- O atendimento de cada resquisição não implica na criação de um novo processo no sistema operacional.

Apresentaremos a seguir as tecnologias de páginas do servidor que são mais amplamente utilizadas: *Hypertext Preprocessor* (PHP) e *Active Server Pages*(ASP).

2.2.1 Hypertext Preprocessor (PHP)

O *Hypertext Preprocessor* (PHP) é uma linguagem de script do lado do servidor. O PHP pode ser utilizado na maioria dos sistemas operacionais, incluindo o Linux,

variantes do Unix (Solaris, HP-UX e OpenBSD), Windows, Mac OS, Risc OS entre outros. O PHP é suportado também pela maioria dos servidores web atuais, incluindo Apache, Microsoft Internet Information e muitos outros.

Com o PHP é possível optar pelo sistema operacional e pelo servidor Web. Também é possível escolher entre utilizar a programação estruturada, a programação orientada a objetos ou uma mistura destas.

O PHP possui ainda suporte a uma ampla variedade de Bancos de Dados, o que se torna uma de suas principais vantagens (MANUAL PHP) .

2.2.2 Active Server Pages (ASP)

A *Active Server Pages*(ASP) é uma tecnologia da Microsoft para possibilitar a criação de páginas interativas na Web. Por usar controles ActiveX como seus componentes, a tecnologia ASP é basicamente restrita a plataformas baseadas em Microsoft Windows, pois componentes ActiveX são dependentes de plataforma. A plataforma ASP é disponibilizada para outros ambientes apenas por produtos terceirizados. As páginas ASP utilizam a linguagem VBScript ou JScript da Microsoft, e suportam o modelo COM.

A tecnologia ASP.NET, é baseada no Framework .NET da Microsoft e possui uma sintaxe muito similar ao ASP anteriormente mencionado. ASP .NET é um ambiente .NET compilado no qual é possível usar qualquer das linguagens suportadas: VB.NET, C#, JScript.NET e VBScipt.NET.(INTRODUCTION TO ASP. NET)

Para migrar de ASP para ASP.NET é necessário alterar as páginas existentes, uma vez que o ASP.NET é um novo modelo. A única vantagem nesse caso é que os desenvolvedores estarão mais familiarizados com a sintaxe.

2.3 Plataforma J2EE

J2EE é uma plataforma para executar aplicações Java do lado do servidor. Antes da criação desta plataforma, as aplicações Java eram escritas usando interfaces proprietárias. Como cada revendedor possuía a sua própria arquitetura, era necessário obter conhecimentos específicos para utilização e configuração. Os desenvolvedores precisavam aprender cada arquitetura utilizada, o que aumentava o tempo e o custo de desenvolvimento. Consequentemente a comunidade de desenvolvimento Java era fragmentada, o que tornava impossível o desenvolvimento de aplicações empresariais de qualidade.

A introdução da plataforma J2EE e a sua aceitação pelos revendedores, resultou na padronização das APIs (*Application Program Interface*) e diminuiu o tempo de aprendizado para aplicações Java para servidores. Revendedores (como por exemplo BEA e IBM), providenciaram implementações para essas interfaces criando seus servidores de aplicação, de acordo com as especificações J2EE. Assim foi possível concentrar esforços em implementar a lógica e a interface com o usuário, que eram mais importantes para a aplicação.(SHENOY e MALLYA, 2004)

A especificação J2EE define contêineres para o gerenciamento do ciclo de vida de componentes do servidor. Existem dois tipos de contêineres: Contêineres de servlets e contêineres de *Enterprise Java Beans* (EJBs).

2.3.1 Java Servlets

Enquanto a Linguagem Java encontrava-se em fase de grande crescimento, tanto em aumento do numero de APIs (*Application Program Interface*) disponíveis quanto em sua utilização por desenvolvedores, a tecnologia de Java Servlets foi criada. Esse advento abriu um mundo de possibilidades para os desenvolvedores.

Com a tecnologia de Java Servlets surgindo como uma alternativa eficiente para aplicações CGI, a arquitetura Web tornou-se atrativa para o desenvolvimento de aplicações distribuídas. Os servlets oferecem mecanismos adequados a qualquer servidor baseado em requisições e respostas, suportando os mecanismos básicos de conexão aos clientes, permitindo que o desenvolvimento se concentre na a extensão de serviços através de servlets.

Aplicações servlets não diferem das de CGI quanto à maneira pela qual interagem com o servidor. Assim o ciclo de vida de um Servlet se divide em quatro etapas como podemos observar na figura 2.1:

1. Cliente emite uma requisição ao servidor

2. O servidor chama o Servlet para que efetue a execução do serviço solicitado

3. O Servlet gera conteúdo dinâmico, respondendo á solicitação do cliente, podendo usar recursos disponíveis na plataforma Java.

4. o Servidor retorna o resultado gerado para o cliente como uma resposta http.

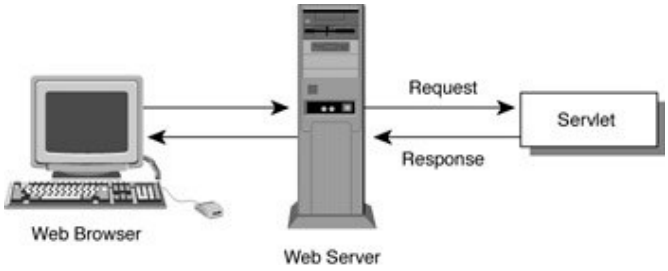

Figura 2.1 - Ciclo de vida de um servlet (GOODWILL, 2002)

Quando um servlet é carregado pela primeira vez na máquina virtual Java do servidor, seu método *init()* é invocado, preparando os requisitos para a execução de um serviço (instanciação de objetos, leitura de valor de contadores em arquivos e etc.) e promovendo a conexão com outros serviços (um banco de dados, por exemplo). Para liberar esses recursos utiliza-se o método *destroy()*, que é chamado quando o servidor concluir a sua atividade.

Além das vantagens já mencionadas quanto ao desempenho de execução de servlets em relação a CGI, é importante acrescentar que eles permitem a utilização de toda a plataforma baseada em Java, incluindo a vantagem da sua portabilidade.

2.3.2 Java Server Pages (JSP)

Java Server Pages é uma especificação que combina Java e HTML para prover conteúdo dinâmico em páginas Web. JSPs são mais convenientes de se escrever porque permitem que o código seja inserido diretamente em páginas HTML, em contraste com os servlets, nos quais todo o conteúdo em HTML precisa ser gerado a partir do código Java.

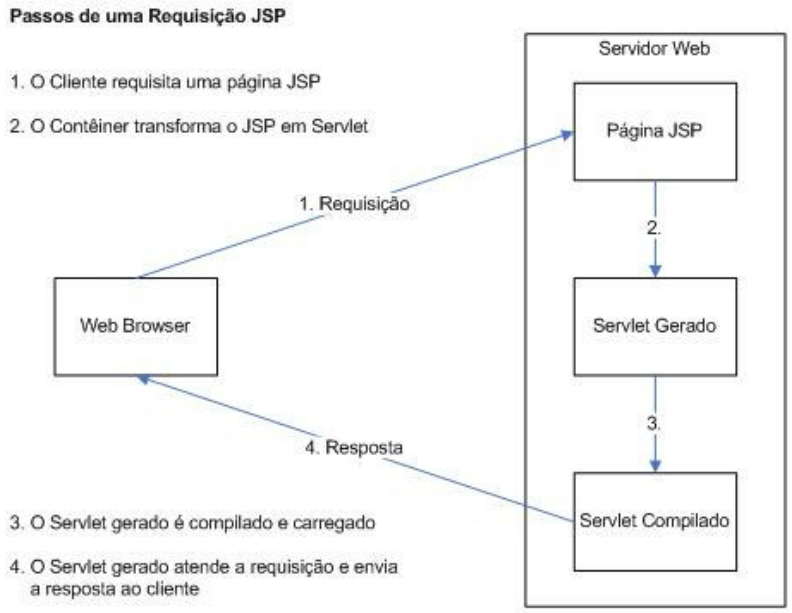

Figura 2.2 - Os passos de uma requisição JSP(GOODWILL, 2002)

A primeira vez que o arquivo é requisitado, ele é traduzido em um Servlet e compilado em um objeto que é carregado na memória conforme apresentado na figura 2.2. O servlet gerado atende a requisição e a saída é enviada ao cliente requisitante. Em todas as requisições seguintes o servidor irá averiguar se o código fonte da página mudou. Caso o código não tenha mudado, o servidor chama o objeto Java que já está compilado. Se o código fonte foi alterado, o contêiner irá recompilar novamente a página. (GOODWILL, 2002)

As JSPs fornecem abordagens distintas para a inserção do código que será usado no servidor. A primeira opção são os *scriptlets*, que são nada mais do que códigos Java colocados diretamente na página. Os *scriptlets* são relativamente fáceis de usar, rápidos e poderosos. A segunda opção são as tags JSP Customizadas, que tem sintaxe e formato parecido com as tags HTML; elas requerem um pouco mais esforço para serem escritas, entretanto são muito mais fáceis de usar e manter em longo prazo. (HUSTED, DUMOULIN, FRANCISCUS et. al, 2003)

2.3.2.1 Tags Customizadas

A tecnologia JSP permite que os desenvolvedores definam tags customizadas, que são tags de formatação que são substituídas por conteúdo dinâmico quando a página é apresentada. O conteúdo dinâmico é criado por uma classe gerenciadora a qual o programador cria e empacota em um arquivo de biblioteca. O programador define a

sintaxe para a tag e implementa o comportamento dela na classe gerenciadora. Designers podem então importar e usar as bibliotecas de tags assim como usam as outras tags de marcação (JONHSON, STEARNS, SINGH et. al, 2002).

Tags customizadas trouxeram muitos benefícios para as aplicações J2EE, entre eles:

- Tags Customizadas são reusáveis, sendo que *scriplets* geralmente não são.
- Bibliotecas de tags customizadas fornecem um alto nível de serviços para páginas JSP que são portáveis entre os contêineres JSP.
- Tags Customizadas facilitam a manutenção, uma vez que eles reduzem a quantidade de código repetida.
- Tags Customizadas ajudam os desenvolvedores a focar em suas tarefas principais. Os *designers* podem trabalhar exclusivamente com tags customizadas e formatação padrão, ao invés de varias tags em conjunto com elementos de *scriptlets*. Com isso os programadores podem se focar em desenvolver a lógica customizadas das tags.
- Tags Customizadas podem fornecer uma sintaxe intuitiva para a invocação das regras de negócio a não programadores.
- Tags Customizadas desacoplam a lógica de negócios e a apresentação de dados. Essa separação facilita a manutenção, esclarece a intenção do conteúdo de cada componente, e permite que programadores e designers trabalhem relativamente independentes um do outro.

2.3.2.2 Vantagens do JSP sobre os seus concorrentes

A tecnologia JSP possui várias vantagens sobre a maioria de seus concorrentes. Abaixo estão relacionadas algumas:

Vantagens sobre o PHP

PHP é uma linguagem gratuita e de código aberto que é parecida tanto com ASP quanto com JSP. A vantagem do JSP é que a parte dinâmica é escrita em Java, que é mais popular para os programadores, possuindo uma API abrangente para rede, acesso a banco de dados, objetos distribuídos, e outras facilidades. Enquanto isso o PHP requer a aprendizagem de toda uma nova linguagem.

Vantagens sobre o Active Server Pages (ASP)

ASP é uma tecnologia da concorrente Microsoft. JSP possui duas vantagens: A parte dinâmica é escrita em Java, e não em linguagens específica do ASP, sendo assim

mais robusta e mais apropriada para aplicações complexas que necessitem componentes reusáveis. A outra vantagem é que JSP é portável para outros sistemas operacionais e servidores Web, com a proposta de codificar uma vez e executar em qualquer ambiente.

Vantagens sobre o Java Servlets

Todos os documentos JSP são traduzidos automaticamente em servlets. Mas é mais conveniente escrever ou alterar tags HTML do que codificar instruções que geram o código da página. Com o auxilio do struts, podemos obrigar a separação da apresentação do conteúdo. Assim, podem-se colocar pessoas diferentes em tarefas diferentes: o *web designer* pode construir o HTML usando ferramentas específicas em visões em JSP e deixar espaço para os programadores de servlets inserirem conteúdo dinâmico.

Vantagens sobre o *Server Side Includes* (SSI)

SSI é uma tecnologia muito usada para inserir código externo dentro de uma página estática. JSP apresenta vantagens se comparada com esta abordagem porque você tem um rico conjunto de ferramentas para construir estas partes externas e tem mais opções com relação à resposta HTTP. Além disso, SSI é voltado apenas para inclusões simples, e não para programas que fazem conexões a bancos de dados, e outras atividades relacionadas.

Vantagens sobre o JavaScript

JavaScript, que é totalmente diferente da linguagem de programação Java, é geralmente usado para gerar HTML dinâmico no cliente, construir partes da página web à medida que o browser carrega o documento. Esta é uma capacidade útil, mas trata apenas situações onde a informação dinâmica está baseada no ambiente do cliente. Com a exceção de *cookies*, os dados de uma requisição HTTP não estão disponíveis para rotinas JavaScript do lado do cliente. E, uma vez que JavaScript necessita de rotinas para programação em rede, o código JavaScript no cliente não pode acessar recursos do lado do servidor como bancos de dados, catálogos, etc. JavaScript também pode ser usado no servidor, entretanto a linguagem Java é muito mais poderosa, flexível, confiável e portável.

3. Padrões de Projeto

Padrões de projeto (*design patterns*) são soluções já testadas que se mostraram eficientes para resolver problemas recorrentes, e que portanto devem ser seguidas para resolução destes problemas.

Com o passar do tempo, os padrões passaram a ser identificados e reutilizados em várias áreas, inclusive em desenvolvimento de software. O livro *Design Patterns: Elements of Reusable Object-Oriented Software* de Eric Gamma, Richard Helm, Ralph Johnson e John Vlissides, publicado em 1995, foi o primeiro a documentar padrões de projeto em software. A partir de então, padrões vêm sendo amplamente documentados e reutilizados. (HUSTED, DUMOULIN, FRANCISCUS et. al, 2003)

Padrões de projetos são guias, de forma que selecionar um padrão não é o mesmo que importar uma classe, pois é necessário implantar o padrão no contexto da sua aplicação. Os padrões de projeto não determinarão como a aplicação deve ser escrita, mas apenas como a aplicação pode ser implementada da melhor maneira depois que as características e requisitos estão determinados.

3.1 O Padrão MVC

O padrão MVC foi originado do SmallTalk, e consiste em três elementos: o Modelo, a Visão e o Controlador, conforme descrito abaixo (CAVANESS, 2002) :

- Modelo: Representa os objetos de dados. O modelo é o que está sendo manipulado e apresentado ao usuário.
- Visão: É a apresentação do modelo na tela. É o objeto que apresenta o estado dos objetos de dados.
- Controlador: Define a maneira que a interface com o usuário reage às entradas de um usuário. É o objeto que manipula o Modelo.

Os benefícios trazidos pela utilização do MVC são os seguintes (CAVANESS, 2002):

- Confiabilidade: As camadas de apresentação e de transação têm uma clara delimitação, o que permite mudar a aparência da aplicação sem recompilar o código do modelo ou do controlador. Isto permite que alterações na aparência não afetem a estabilidade do código responsável pela execução de transações.

- Reutilização elevada e adaptabilidade: O MVC permite o uso de múltiplos tipos de visão, todas acessando o mesmo código no servidor.
- Baixo custo de desenvolvimento e ciclo de vida minimizado: O MVC torna possível possuir programadores de lógica da aplicação e designers responsáveis por manter as interfaces com o usuário, aumentando assim a produtividade e a qualidade do produto final promovida pela especialização das atividades.
- Desenvolvimento rápido: o tempo de desenvolvimento pode ser reduzido substancialmente porque os programadores do controlador podem focar apenas em transações, e os designers podem manter seu foco apenas na apresentação.

3.1.1 Objetivos e Motivação do uso do MVC

O bem conhecido padrão de projeto Modelo-Visão-Controlador viabiliza uma maneira eficiente de projetar sistemas de software interativos. Também conhecido como apresentação, abstração e controle, sua idéia chave é separar as interfaces de usuário dos dados representados. O MVC clássico atualmente se aplica a um baixo nível de interação com o usuário, através de cliques em botões no teclado ou no mouse. No MVC, a visão mostra a informação para o usuário e em conjunto com o controlador - que processa as interações do usuário, compreende a interface de usuário da aplicação. O Modelo é a parte da aplicação que contém tanto a informação representada pela visão quanto a lógica que muda essa informação em resposta a interação do usuário. Esse padrão de projeto desacopla a informação da interface com o usuário. Aqui, a visão e o controlador são combinados em uma apresentação, os dados da aplicação são designados como o componente de abstração e o componente de controle é responsável pela comunicação entre a apresentação desacoplada e componentes de abstração. A utilização desse padrão de projeto torna as aplicações mais fáceis de serem desenvolvidas e mantidas desde que:

- A aparência possa ser mudada drasticamente sem mudar as estruturas de dados e a lógica do negócio.
- A aplicação possa facilmente manter diferentes interfaces, como múltiplas linguagens ou diferentes conjuntos de permissões de usuário.

O termo MVC também é utilizado para descrever o jeito que mudanças em larga escala em um Modelo são conduzidas por um Controlador, que é responsável não só por aceitar interações de processamentos, mas também pela lógica que muda o estado

inteiro de uma aplicação em resposta a um evento criado por uma interação com o usuário.

3.2 Padrões de Projeto aplicados a Desenvolvimento Web

Em seguida serão descritos os padrões de projeto utilizados em aplicações Web desenvolvidas em JSP.

3.2.1 Modelo 1

O Modelo 1 é a maneira mais fácil de desenvolver aplicações baseadas em JSP, No modelo 1, o browser acessa diretamente páginas JSP, o que implica que todo o gerenciamento de requisições seja feito diretamente por JSPs.

Nesse modelo, ao clicar em um link, uma página JSP é chamada diretamente. A figura 3.1 mostra o contêiner analisando a página JSP e executando o servlet resultante. O Código JSP contém código embutido para acessar os Java Beans do Modelo. O Java Beans do Modelo contém atributos para capturar os parâmetros passados pela requisição. Além disso ele contém lógica de negócios para acessar os dados necessários para a página. A página JSP é então mostrada em HTML usando Java Beans, podendo usar também classes auxiliares e outras tags.(SHENOY e MALLYA, 2004)

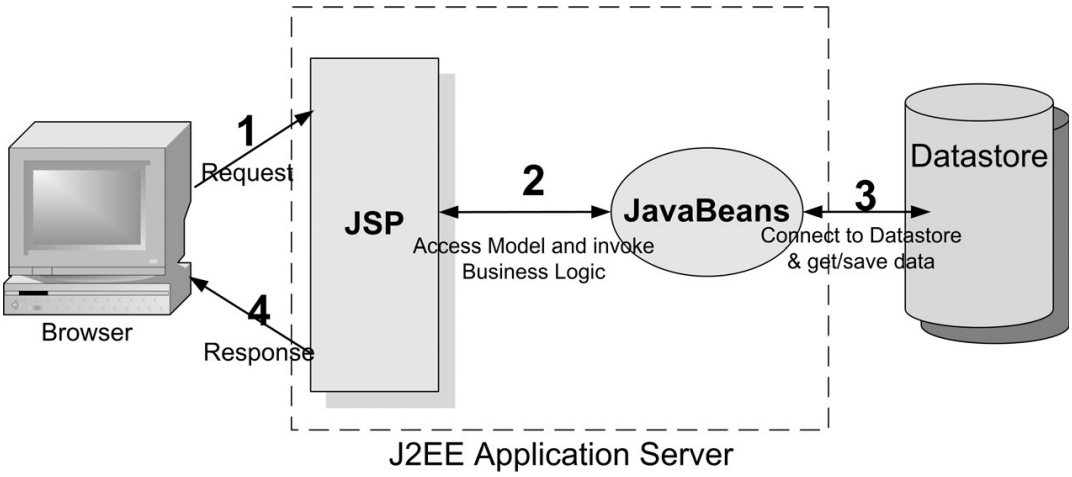

Figura 3.1 - Arquitetura do Modelo 1 (SHENOY e MALLYA, 2004)

3.2.1.1 Problemas na Arquitetura do Modelo 1

A arquitetura do Modelo1 é simples. Existe alguma separação do conteúdo (Java Beans do Modelo) e a apresentação. Essa separação é boa o suficiente para pequenas aplicações. Nessa arquitetura, a lógica de apresentação implica em uma considerável quantidade de código Java embutido na forma de *scriptlets*. Isso não é recomendável,

uma vez que a sua manutenção torna-se um pesadelo para os desenvolvedores. Em aplicações maiores, os JSPs são desenvolvidos e mantidos por *designers*. *Scriptlets* e formatação misturados revelam definições incompletas de regras e se tornam problemáticas.

O controle da aplicação é descentralizado, pois a próxima página a ser mostrada é determinada pela lógica embutida na página atual. O controle de navegação descentralizado pode trazer problemas quando os requisitos da aplicação mudam.

3.2.2 Modelo 2 – MVC

A principal diferença entre o Modelo 1 e o Modelo 2 é que no segundo, um controlador gerencia as requisições de um usuário, ao invés da própria página JSP. O controlador é implementado como um *servlet*. Os seguintes passos são executados quando um usuário envia a requisição de acordo com a figura 3.2:

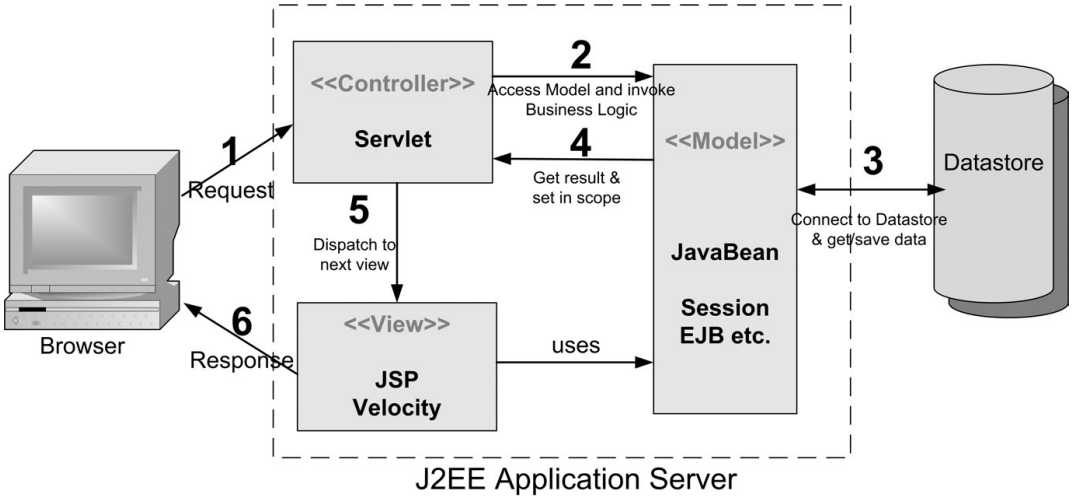

Figura 3.2 - Arquitetura do Modelo 2 MVC (SHENOY e MALLYA, 2004)

1 - O controlador gerencia a requisição do usuário.

2 - O controlador instancia o Java Beans baseado nos parâmetros de requisição.

3 - O Java Beans fala diretamente com a camada intermediária ou diretamente com o banco de dados para recuperar os dados solicitados.

4 - O Controlador coloca o Java Beans resultante no contexto determinado (requisição, sessão ou aplicação).

5 - O Controlador envia a requisição para a próxima visão, baseada na solicitação.

6 - A visão usa o Java Beans resultante para mostrar os dados.

Observe que não existe lógica de apresentação na JSP. A única função do JSP na Arquitetura do Modelo 2 é mostrar os dados do Java Beans em seu devido escopo.

3.2.2.1 Vantagens do Modelo 2

Como não existe lógica de apresentação no JSP praticamente não existem *scriptlets*, o que evita futuros transtornos. Apesar de desencorajar o uso de *scriptlets*, o Modelo 2 não evita que seja utilizado.

As aplicações baseadas na arquitetura do Modelo 2 são mais fáceis de manter e reusar, uma vez que as visões não referenciam umas as outras e que não existe lógica de apresentação nas visões. Também é permitido definir claramente as regras e responsabilidades em projetos maiores, possibilitando uma colaboração maior entre a equipe de desenvolvimento.

3.2.2.2 Problemas do MVC

Existe uma dificuldade em associar problemas do mundo real ao MVC. Em aplicações médias e grandes o controlador centralizado também é uma fraqueza. Se considerarmos uma pequena aplicação com 15 páginas e 5 submissões de formulário, o numero de requisições a serem tratadas são 75. A capacidade de centralização do controlador, que era uma das maiores vantagens do MVC acaba por torná-lo sobrecarregado. Para resolver essa questão existem duas opções: utilizar vários controladores, ou estender o MVC e utilizar um controlador configurável. Como o Struts utiliza apenas um controlador, o framework implementa o controlador configurável.

3.2.3 MVC com um Controlador Configurável

Quando uma aplicação fica maior torna-se inviável usar o MVC com um único controlador. Pode-se estendê-lo para lidar com as complexidades de aplicações do mundo real. Um mecanismo de extensão do MVC que é amplamente adotado é o Controlador configurável.

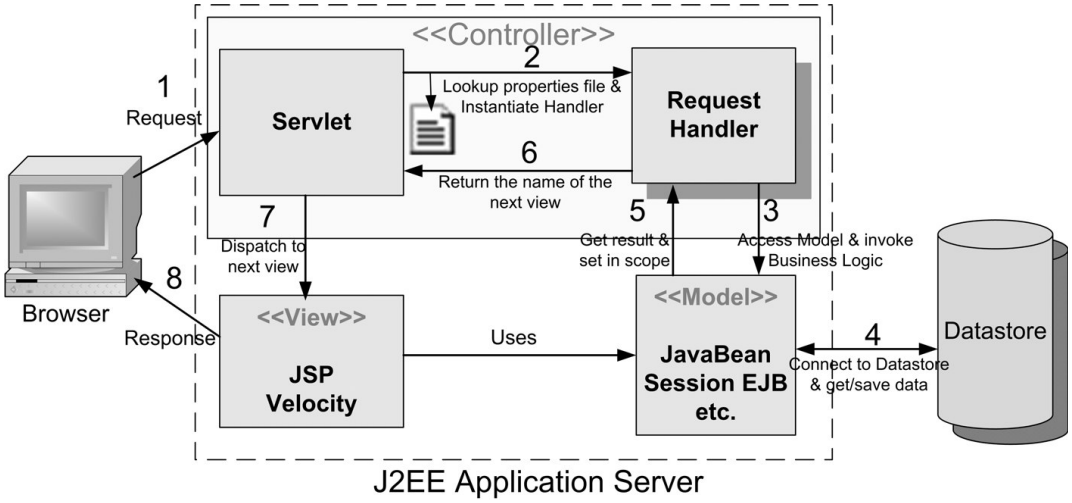

Figura 3.3 - MVC com controlador configurável(SHENOY e MALLYA, 2004)

Conforme podemos observar na figura 3.3, quando uma requisição http chega do cliente, o Servlet controlador procura no arquivo de configuração para decidir qual a classe que gerencia a requisição. Essa classe contém a informação sobre a lógica da apresentação para a requisição, incluindo a lógica de negócios, em outras palavras ela possui tudo o que é necessário para gerenciar a requisição. A única diferença do MVC propriamente dito é que o controlador configurável procura no arquivo antes de instanciar a classe, ao invés de chamá-la diretamente.(SHENOY e MALLYA, 2004)

4. O Framework Struts

4.1 Frameworks

Frameworks – ou "arcabouços" – são conjuntos de classes que podem ser facilmente reutilizados para construir aplicações. Para tal, basta adicionar as classes que implementam a lógica daquela aplicação em particular ao *framework* para ter uma aplicação pronta para ser utilizada.

Um *framework* fornece componentes genéricos que a aplicação deve estender para fornecer um conjunto de funcionalidades, ao contrário de bibliotecas de software que possuem funções ou rotinas que a aplicação pode invocar.

Os *frameworks* são voltados a um determinado tipo de aplicação, como por exemplo, aplicações financeiras ou gráficas, ou a um aspecto de sua funcionalidade, como por exemplo, interfaceamento com o usuário ou acesso a bancos de dados.

4.2 Definição do Framework Struts

O Projeto Jakarta Struts é um projeto de código aberto patrocinado pela *Apache Software Foudation*. O projeto Struts foi modelado com a intenção de ser um framework para a criação de Aplicações Web que facilmente separa a camada de apresentação das camadas de transações e de persistência. Desde sua concepção, Struts fornece um suporte bastante completo aos desenvolvedores de aplicações Web e está rapidamente se transformando em um framework de desenvolvimento dominante na comunidade de desenvolvedores Web (HUSTED, DUMOULIN, FRANCISCUS et. al, 2003).

O framework Struts foi criado por Craig R. McClanahan e doado para a *Apache Software Foundation* (ASF) no ano de 2000. O projeto tem muitos contribuidores, e é um dos projetos de maior popularidade da Apache (HUSTED, DUMOULIN, FRANCISCUS et. al, 2003), em conjunto com os projetos Ant, Log4j e Tomcat, que também serão utilizados nesse trabalho.

Embora o Framework Struts tenha sido construído com as tecnologias J2EE, ele não é parte da plataforma J2EE padrão. Portanto podemos considerá-lo complementar a todos os padrões J2EE (não existem características que se sobrepõe ou que são incompatíveis).(HUSLY e YU, 2002)

O Framework Struts é uma implementação do lado do servidor do MVC usando uma combinação de JSPs, JSP tags e Servlets Java. A organização de componentes de desenvolvimento para a Web pode ser classificada em antes e depois da utilização do framework, conforme ilustrado na figura 4.1. Antes do Struts podemos verificar a mistura entre HTML e Java, separada da camada de serviços. Com a utilização do framework podemos verificar uma maior modularização na distribuição dos elementos.(DAVIS, 2001)

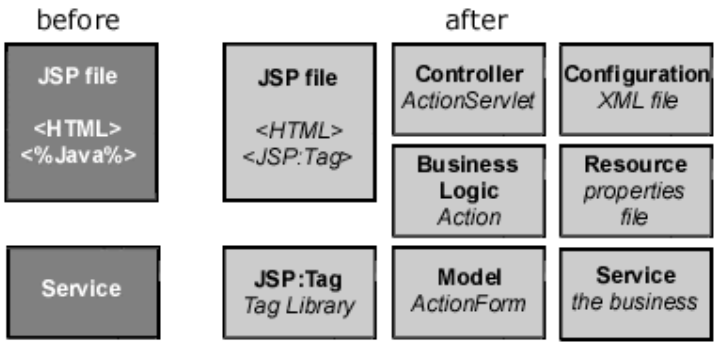

Figura 4.1 - Antes e Depois do Struts(DAVIS, 2001)

O desenvolvimento do Framework ainda se encontra em uma forma preliminar e está sofrendo uma grande quantidade de alterações. Esse aspecto se torna um sério problema, uma vez que não é possível ter certeza se, com as novas versões, as aplicações continuarão a funcionar normalmente ou terão que ser alteradas. Entretanto, ao separar a aplicação em componentes menores é possível obter maior reuso mesmo tendo problemas de compatibilidade com as tecnologias envolvidas.

O Framework possui esse nome em referência às estruturas que sustentam nossas construções civis. Essa é uma excelente analogia, pois ilustra o papel do Struts no desenvolvimento de aplicações Web. Quando estruturas físicas são construídas, os engenheiros utilizam estruturas para dar sustentação a cada andar de uma construção. De forma similar, os engenheiros de software utilizam o struts para dar sustentação a cada camada de uma aplicação (HUSTED, DUMOULIN, FRANCISCUS et. al, 2003).

4.3 Internacionalização

A internacionalização é o processo para construir uma aplicação de modo que ela possa ser adaptada a várias línguas e regiões sem alterações no código. A localização é o processo de adaptar o software a uma língua e a uma região específica,

adicionando componentes exclusivos do local, verificando se a composição das datas, números e moedas estão de acordo com os padrões para determinado local. Em geral, a parte mais demorada da localização é a tradução do texto. (HUSTED, DUMOULIN, FRANCISCUS et. al, 2003)

Embora seja fácil fazer com que componentes da apresentação usem caracteres codificados especialmente para cada linguagem, isso poderia tornar o texto difícil de ser alterado. Para reformular um rótulo ou uma mensagem, tanto o código da apresentação ou o código Java podem precisar ser atualizados por equipes de desenvolvimento diferentes. Isso implicaria que a mesma aplicação não pudesse ser reutilizada sem recompilar o código fonte.

Para evitar problemas, o framework struts permite que a definição de rótulos e mensagens em arquivos separados, chamados de pacote de recursos (*java.util.ResourceBundle*). Quando uma mensagem ou rótulo precisa ser escrito, a aplicação pode se referir à mensagem através de sua chave. O Framework recupera o texto para o rótulo durante a execução. O nome e localização de arquivos de imagens também podem ser lidos a partir de um pacote de recursos.

Muitas aplicações não são internacionalizadas, e torná-las assim pode ser muito custoso. Como struts tem suporte a internacionalização, a maior parte das aplicações Struts pode ser localizada sem muitos problemas. Se a aplicação for definida tendo elementos de internacionalização desde o início, a localização se resumirá em traduzir apenas um arquivo de mensagens.

A localização ajuda a manter um grupo de recursos comuns, o que pode tornar a aplicação mais fácil de ser mantida.

Assim como o Java, o Struts foi completamente internacionalizado. Esse é um dos motivos que torna o Struts tão popular no mundo todo. (HUSTED, DUMOULIN, FRANCISCUS et. al, 2003)

Um pacote de recursos (*java.util.ResourceBundle*) é uma coleção de objetos *Properties* (java.util.properties). Cada Objeto *Properties* é digitado para uma localidade, sendo que cada localidade é identificada pela região e pela língua. Se um objeto *Properties* não for encontrado para alguma localidade, o objeto mais próximo será retornado. Isso é possível porque para a localização é utilizado o princípio de herança.

No arquivo de configuração do Struts deve ser informado em qual pacote está o arquivo *Properties*:

```
<message-resources parameter="br.com.nota_fluxo.ApplicationResources" />
```

Para criarmos um arquivo *Properties* para a língua portuguesa no Brasil podemos colocar o nome conforme abaixo:

```
ApplicationResources_pt_BR.properties
```

Onde podemos observar o nome do arquivo com o código da língua seguido pelo código da localidade.

Um arquivo *Properties* possui pares de chaves-valor, sendo que a chave é um identificador para uma mensagem, valor é a mensagem propriamente dita.

```
empresa.cd_emp=Código da Empresa

empresa.nm_emp=Nome
```

Também podem existir mensagens que necessitem de informações em tempo de execução. Para isso podemos escrever mensagens como:

```
errors.required=O campo {0} deve ser preenchido.
```

Assim, quando quisermos utilizar a mensagem devemos passar o primeiro parâmetro, que será o nome do campo que é obrigatório tornando a informação muito mais clara para o usuário. No caso acima utilizamos um exemplo de uma mensagem do validador, bastando especificar na sua configuração o nome da chave que será usada no lugar do *{0}*.

4.4 O Modelo

O Modelo é comumente chamado de camada de negócios. O modelo representa os dados da lógica de negócios e deveria assemelhar-se com as entidades no mundo real e processos para a organização.

O modelo de componentes de uma aplicação deve ser o artefato mais valioso de uma organização(CAVANESS, 2002). É o que torna a aplicação diferente de todas as outras. (HUSTED, DUMOULIN, FRANCISCUS et. al, 2003)O Modelo inclui as entidades de negócios e as regras pertinentes a modificação dos dados.

A camada do modelo esconde os detalhes da implementação, o que facilita o gerenciamento de alterações, pois fornece uma interface simples e uniforme . Entretanto essa característica pode também ocultar, por exemplo, que várias consultas estão sendo executadas, ou que um objeto remoto está sendo acessado, o que faz com que o

desenvolvedor necessite tomar cuidado com a freqüência que esse método de negócios é chamado (HUSTED, DUMOULIN, FRANCISCUS et. al, 2003)

A maior parte dos desenvolvedores, ao escrever a sua primeira aplicação Struts, coloca o acesso ao banco de dados diretamente nas classes *Action*. Isso pode se tornar um problema a medida que a aplicação for crescendo, tornando-a difícil de expandir e manter. Desta forma os desenvolvedores são encorajados a enviar solicitações entre *Actions*. Quando as classes *Action* se tornam interdependentes, a aplicação pode se tornar confusa, o que acontece com o Modelo 1 de desenvolvimento para a internet que é baseado na tecnologia JSP.

Em uma aplicação mais complexa o modelo poderá utilizar *Entreprise Java Beans* (EJBs) (JONHSON, STEARNS, SINGH et. al, 2002). Essa tecnologia ainda está passando por algumas melhorias, entretanto ainda possui um impacto significante de desempenho se a camada Web usar beans de entidade diretamente como modelo, devido ao impacto causado pela sobrecarga de processamento para execução de chamadas remotas. Em vários casos, Java Beans são retornados dos beans de sessão e usados dentro da camada web. Esses Java Beans são usualmente referenciados como objetos de transferência de objetos ou de valores de objetos (o framework Struts permite usar os Java Beans para transferir os resultados da camada de lógica de negócios para a visão), que são usados dentro das visões para construir conteúdo dinâmico.

4.4.1 Tipos de Modelos

O termo ‘modelo’ tem vários significados diferentes. Em termos gerais, um modelo é uma representação de um aspecto da realidade. O principal propósito de criar um modelo é ajudar a entender, descrever, e simular como as coisas funcionam no mundo real.

No desenvolvimento de software, o termo ‘modelo’ é usado para indicar tanto a representação lógica de entidades do mundo real quanto a criação física de classes e interfaces que os programas podem usar. O primeiro passo, entretanto, sempre deve ser efetuar uma analise através do domínio do problema. Uma vez que os casos de uso estão completos, o próximo passo deveria ser desenvolver um modelo conceitual. (CAVANESS, 2002)

4.4.2 O Modelo Conceitual

Durante a análise do domínio do problema, o modelo conceitual deve ser desenvolvido baseado em entidades do mundo real dentro do espaço do problema. O modelo conceitual ilustra os conceitos, os relacionamentos e os atributos que pertencem a cada conceito. O comportamento atualmente não é representado nesse tipo de modelo.

O Modelo conceitual é desenvolvido de um conjunto de casos de uso para o sistema. O propósito de construir o modelo é ajudar a identificar as entidades que mais facilmente serão classes no estágio de projeto e ajudar a entender melhor o problema do domínio.

O modelo de projeto é um artefato do estágio de análise. Em grupos pequenos de desenvolvimento, ou pequenos projetos, é possível partir diretamente para a elaboração do modelo de projeto. O risco de fazer isso, entretanto, é que se pode deixar o estágio de análise sem um entendimento claro e conciso dos requisitos. Esse modelo inclui diagramas de interação, diagramas de classes e possivelmente diagramas de estado. (CAVANESS, 2002)

4.4.3 Persistência

Em geral, persistência significa que dados que são usados para alimentar uma aplicação, tanto por um usuário humano como por outros meios, devem existir durante o tempo de vida da aplicação. Apesar de a aplicação poder ser encerrada ou o computador poder ser desligado ou falhar, a informação deve permanecer. Isso é claramente importante para todas as organizações. Uma vez que os dados persistem, eles podem ser recuperados depois.

4.4.3.1 Armazenando objetos em um banco de dados

Objetos armazenam estados e comportamentos podem ser passados através de seus relacionamentos com outros objetos. O paradigma relacional é baseado no relacionamento de dados e junção de conjuntos de dados com campos identificadores semelhantes. Essencialmente o banco de dados relacional possui uma visão muito plana dos dados. Essas diferenças acarretam um desafio para os dois lados: objetos devem ser colocados de uma forma plana no banco de dados, mas os relacionamentos que os objetos possuem uns com os outros devem ser armazenados para que possam ser recuperados posteriormente. (CAVANESS, 2002)

4.5 A Visão

O framework usa os componentes de visão para adaptar conteúdo dinâmico para clientes Web. Baseados primariamente em Java Server Pages, os componentes fornecem suporte a internacionalização, aceitação de entradas de usuário, validação e tratamento de erros, fazendo com que fique mais fácil para o desenvolvedor focar nos requisitos do negócio.

Uma visão corresponde a uma representação do modelo do domínio em uma interface com o usuário. Existem muitas visões diferentes do mesmo modelo. Metaforicamente falando, a visão é uma janela que clientes podem usar para olhar o estado do modelo, e a perspectiva pode ser diferente dependendo em qual janela um cliente observa.

Nem sempre todos os dados são mostrados na visão, mas o usuário ainda está observando o mesmo modelo de negócios.

Os objetos não têm uma maneira comum de representação externa. Sua representação pode ser na forma de XML, XSLT, SOAP entre outros. No nosso caso, o formado HTML é gerado com o intuito de ser exibido em um navegador Web (CAVANESS, 2002).

4.5.1 Definição da visão

A visão dentro da camada web do MVC tipicamente consiste em páginas HTML e JSP, que são usadas para prover conteúdo estático, enquanto JSPs podem ser usados para conteúdo tanto estático quanto dinâmico. Apesar da maior parte do conteúdo dinâmico ser gerada na camada Web, algumas aplicações podem precisar de validações do lado do cliente, efetuadas com JavaScript. Esse fato não interfere ou infringe as regras do conceito MVC.

HTML e JSP não são as únicas opções para a visão. É possível suportar facilmente WML (*Wireless Markup Language* - linguagem de marcação baseada em XML projetada para uso em dispositivos sem fio) ao invés de HTML, por exemplo. Podem-se suportar múltiplas visões, para diferentes tipos de clientes, usando os mesmos componentes do modelo.

4.5.2 Visões dentro do Framework Struts

Geralmente as visões do struts são construídas usando páginas JSP. Outras abordagens e frameworks estão disponíveis para efetuar as mesmas funções, mas JSP é

mais amplamente usado dentro da comunidade de desenvolvedores Struts. Componentes adicionais podem ser usados por ou em conjunto com páginas JSP para adaptar as visões, incluindo (CAVANESS, 2002):

- Documentos HTML – Documentos que utilizam a Linguagem de marcação HTML.
- JSP *tag libraries* – Mecanismo pelo qual se pode definir novas tags a serem usadas nos JSP, de forma já padronizada e distribuída em pacotes.
- JavaScript e folhas de estilo – Linguagem JavaScript (executada no lado do cliente) e folhas de estilo que formatam as páginas.
- Arquivos multimídia – Imagens, Vídeos e Sons.
- Mensagens de *Resource Bundle* – Mensagens organizadas para a internacionalização de software.
- Classes *ActionForm* – Classe que contem elementos que serão usados nas telas de entrada de valores da aplicação.
- Usando Java Beans dentro de componentes de visão – Permite usar os JavaBeans para transferir os resultados da camada de lógica de negócios para a visão

4.5.3 Como o Struts usa JavaBeans

Na arquitetura MVC, são recebidas a notificações das alterações quando algo muda no modelo. Nas aplicações Web, essa notificação não é possível, ou ao menos é difícil de mostrar. Normalmente, o cliente emite uma requisição para o controlador atualizar a visão do estado do modelo. Em outras palavras, o cliente solicita a visão do modelo, ao invés do modelo enviar as mudanças para o cliente. (CAVANESS, 2002)

As JSPs envolvem duas abordagens, colocando códigos Java *scriptlets*, ou usando extensões da tag. As extensões de tag requerem mais esforço para escrever, entretanto depois de escritas são mais fáceis de usar e manter.

Embora as tags Struts sejam fáceis de usar, elas não são a única opção. A JSTL (*Java Server Pages Standard Libraries*) possui um conjunto de tags que acabam por sobrepor de forma flexível as tags Struts. Em breve espera-se que os contêineres sejam otimizados para JSTL, de forma que seja a tag escolhida para a maioria das aplicações. Embora elas não substituam as tags personalizadas, eliminam a necessidade de tags que fazem operações comuns. (HUSTED, DUMOULIN, FRANCISCUS et. al, 2003)

4.5.4 ActionForms

Quase todas as aplicações têm que aceitar entradas do usuário. Alguns exemplos de entradas são nomes de usuários e senhas, informações de cartão de crédito, entre outras. HTML fornece os componentes necessários para adaptar os campos de entrada em um navegador (*browser*), incluindo caixas de texto, botões convencionais, do tipo *radio* e do tipo *check*. Quando se está construindo esse tipo de páginas deve-se aninhar os componentes de entrada nos elementos HTML Forms.(CAVANESS, 2002)

A aplicação no servidor pode retornar valores que foram entrados, efetuar validação de entrada de dados, e então passar os dados para outro componente dentro da aplicação onde o processo de autenticação ocorre. Se os dados de entrada falham a regras de validação de entrada, a aplicação deve retornar para a localização anterior, mostrar novamente alguns ou todos os valores entrados e mostrar uma mensagem de erro indicando que o erro.

Fazer manualmente toda essa funcionalidade, retornando os valores, executando a validação, e mostrando as mensagens de erro em caso de falhas pode ser uma tarefa cansativa. Esse tipo de comportamento é efetuado em vários lugares por uma aplicação Web, e é ideal que seja gerenciado pelo framework e que o código possa ser reusado em várias aplicações.(CAVANESS, 2002)

A Classe *ActionForm* é usada para capturar entradas de dados de formulários HTML e transferir estes dados para a classe *Action*. Aplicações sempre precisam armazenar os dados de entrada temporariamente para que sejam mostrados novamente se um erro ocorrer. Dessa forma a classe *ActionForm* também age como um buffer para armazenar o estado dos dados que o usuário entrou enquanto estava sendo validado. O *ActionForm* também age como um *firewall* para a sua aplicação que ajuda a manter entradas suspeitas ou inválidas fora da camada de negócios. Ultimamente, quando dados são retornados da camada de apresentação, uma *ActionForm* particular pode ser populada e usada por uma página JSP para adaptar os campos de entradas para um formulário HTML. Isso permite maior consistência para os formulários HTML, uma vez que os dados são puxados da *ActionForm* e não de Java Beans diferentes.(CAVANESS, 2002)

4.5.5 Utilização da Classe DynaActionForm

Na versão 1.0 do Struts, era necessário que cada formulário nas páginas fosse associado a uma *ActionForm*. A versão 1.1 mudou isso com a introdução da subclasse

DynaActionForm, que seria o *ActionForm* dinâmico. A figura 4.2 ilustra a hierarquia das classes:

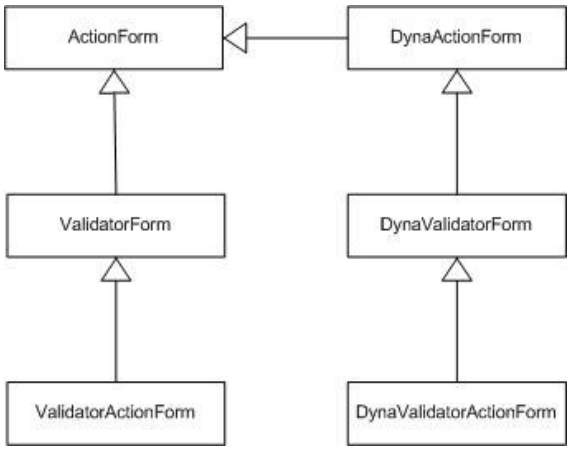

Figura 4.2 - Hierarquia de relacionamento dos forms de validação.(SHENOY e MALLYA, 2004)

Uma classe *DynaActionForm* é declarada da seguinte maneira no arquivo de configuração do Struts:

```
<form-bean name="empresaForm"
       type="org.apache.struts.action.DynaActionForm">
       <form-property name="cd_emp" type="java.lang.String "/>
       <form-property name="nm_emp" type="java.lang.String/>
</form-bean>
```

1 – Para a declaração, o tipo deve ser sempre *org.apache.struts.action.DynaActionForm*

2 – Um *ActionForm* convencional é desenvolvido em Java e declarado no arquivo de configuração do Struts enquanto a *DynaActionForm* não tem uma classe Java associada.

A Classe *DynaActionForm* é rápida e fácil, muito conveniente para um protótipo rápido, porém pode ser considerada sempre a melhor opção possuir *ActionForms* convencionais pelos seguintes motivos:

1 - O *DynaActionForm* aumenta o arquivo de configuração do Struts com a sua definição baseada em XML.

2 - Não existe checagem em tempo de compilação para os campos de formulário. Detecta-los em tempo de execução pode ser bastante trabalhoso.

3 - *ActionForms* podem ser organizados de uma maneira organizada dentro de pacotes (*packages*) ao invés da organização dentro do arquivo de configuração do Struts.

4 - *ActionForm* foram desenvolvidos para serem um tipo de barreira que isola e encapsula os parâmetros das requisições HTTP do uso direto em *Actions*. Com *DynaActionForm* o acesso a propriedades não é diferente de usar request.getParameter("..").

5 - A construção *DynaActionForm* em tempo de execução exige um maior poder de processamento, que pode ser evitado

6 - A economia de tempo obtida através do uso de um *DynaActionForm* é insignificante. Não consome tempo para que os ambientes de desenvolvimento atuais gerem métodos de acesso para os Atributos *ActionForm* (o tempo gasto com a geração de métodos de acesso é menor do que descobrir o erro em tempo de execução e ter que compilar o código novamente).

4.5.6 Uso de Tecnologia JSP para componentes da Visão

A maioria das aplicações web conta com navegadores padrão para exibir informações. Por sua vez, os navegadores padrões contam com a HTML como linguagem de formatação. Em geral as informações fornecidas por uma aplicação não necessitam ser exibidas de forma estática, precisando ter o seu conteúdo mostrado dinamicamente de acordo com a requisição do usuário. È possível mostrar conteúdo dinâmico usando Servlets, de forma que a implementação não pareça ser algo difícil de ser codificado. Poderíamos formatar em HTML de forma independente, da mesma maneira que enviaríamos o texto para exibição para o usuário ou para uma impressora. Páginas complexas se tornam inviáveis de serem escritas assim pois podem demorar para serem criadas e serem difíceis de manter. (HUSTED, DUMOULIN, FRANCISCUS et. al, 2003)

4.5.6.1 Bibliotecas de Tags

As Bibliotecas de tags Struts fornecem a maior parte das funcionalidades que as aplicações precisam para criar uma apresentação seguindo o padrão MVC.

O Struts possui as seguintes bibliotecas de tags incluídas na sua distribuição: *Html, Bean, Logic, Nested e Tiles*. As tags html estão definidas no arquivo *struts-*

html.tld, as tags Bean estão definidas em *struts-bean.tld* e assim por diante. (SHENOY e MALLYA, 2004)

Para usar as tags, elas devem ser incluídas no arquivo de configuração da aplicação (*web.xml*) conforme mostrado na seção 4.7 . Para cada página que for usar essas tags, também é necessário fazer a declaração das bibliotecas na página JSP da seguinte forma:

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

A biblioteca *struts-bean* contém tags úteis para acessar beans e as suas propriedades, bem como novos beans (baseados nesses acessos) que são acessíveis à pagina via variáveis de script e atributos de escopo. Mecanismos convenientes para criar novos beans baseados no valor de cookies, cabeçalhos e parâmetros também são fornecidos.(JAKARTA STRUTS FRAMEWORK)

A biblioteca *struts-html* contém tags usadas para criar entradas struts, bem como possui tag úteis para a criação de interfaces baseadas em HTML.

A biblioteca *struts-logic* contém tags que são úteis no gerenciamento da geração condicional de texto, incluindo suporte a coleções e gerenciamento de fluxo da aplicação.

A biblioteca *struts-tiles* contém tags necessárias para combinar vários componentes em uma visão final composta, usando o plug-in Tiles conforme descrito na seção 4.9.2.

A biblioteca *struts-nested* é uma extensão de outras bibliotecas de tags struts que permite o uso de beans aninhados.

4.5.6.2 Java Server Pages Standard Tag Library (JSTL)

Como uma alternativa para as tags JSP, a Java Server Pages Standard Tag Library (JSTL) apresenta um conjunto de tags que podem se sobrepor às tags do struts. Rapidamente podemos esperar que os contêineres sejam otimizados para o uso de JSTL, tonando-a a biblioteca de tags escolhida para a maior parte das aplicações. JSTL não dispensa a necessidade de tags personalizadas.

A maior parte das tags do struts foi desenvolvida para preencher uma lacuna, e não estão fortemente ligadas ao framework. Se JSTL já existisse, a maioria das tags do struts não teria sido desenvolvida. (HUSTED, DUMOULIN, FRANCISCUS et. al, 2003)

As tags struts são usadas em muitas aplicações, e não estão "competindo" com as tags JSTL, então podemos esperar que em futuras versões do framework tags struts sejam baseadas em JSTL.

4.5.6.3 Struts EL

Struts-EL é uma biblioteca que permite compatibilizar as tags Struts com as tags JSTL, através de um mecanismo para transformar as aplicações com tags struts existentes em aplicações utilizando JSTL de uma maneira não intrusiva (SHENOY e MALLYA, 2004).

Struts-EL torna fácil portar tags struts em tags Struts-EL, mas a real vantagem de usar o Struts-EL percebe-se quando partes que normalmente usariam *scriptlets* ficam complexas.

Nem todas as tags do Struts se tornaram portáveis ao Struts-EL. Onde já existem tags JSTL correspondentes, isto ocasionaria redundância. (SHENOY e MALLYA, 2004)

4.6 O Controlador

O Controlador é um Servlet Java, cujos componentes são responsáveis por detectar as entradas do usuário, possivelmente alterar o domínio do problema, e selecionar a próxima visão para o cliente. O Controlador ajuda a separar a apresentação do modelo. Essa separação nos dá muito mais liberdade para desenvolver uma grande variedade de apresentações baseadas em um único modelo.

O controlador em uma camada de aplicação Web efetua as seguintes tarefas:

1 - Intercepta as requisições HTTP de um cliente;

2 - Traduz cada requisição em uma operação específica de negócio;

3 - Invoca a própria operação de negócios ou a delega a um gerenciador;

4 - Ajuda a selecionar a próxima visão para mostrar para o cliente;

5 - Retorna a visão para o cliente.

Existe um ponto centralizado de controle da aplicação, uma vez que todas as requisições e respostas passam pelo Controlador. Essa característica ajuda quando funcionalidades são adicionadas. Código que normalmente teria que ser colocado em cada página JSP pode ser colocado no Servlet Controlador, que processa todas as

requisições. O controlador também ajuda a desacoplar os elementos de apresentação (visões) da lógica de negócios.

O uso do controlador provê um ponto centralizado de controle onde todas as requisições dos clientes são inicialmente processadas. O controle centralizado efetua duas requisições do MVC. Primeiro o controlador age como o mediador/tradutor entre as entradas e o modelo, provendo uma funcionalidade comum, como segurança, login e outros serviços importantes por trás de cada requisição do cliente. Segundo, a visão é desacoplada da lógica de negócios e de outros componentes de visão, uma vez que todas as requisições são filtradas através do controlador. Isso faz com que as aplicações se tornem muito mais flexíveis

O Framework Struts usa um Servlet para processar requisições que não chegaram. Entretanto, ele confia em vários outros componentes que são parte do domínio do controlador para ajudar a carregar essas responsabilidades.

4.6.1 O Mecanismo do Controlador

Entre várias outras vantagens, o padrão utilizado no controlador traz serviços como segurança, internacionalização e login controlado. Isso permite a aplicação consistente dessas funções por todas as requisições. Quando o comportamento desses serviços precisa de modificações, mudanças afetando potencialmente a aplicação toda precisam ser feitas somente em uma área relativamente pequena e isolada do programa.(CAVANESS, 2002)

4.6.2 A Classe ActionServlet

A classe *org.apache.struts.action.ActionServlet* age como um interceptor para a aplicação Struts. Todas as requisições da camada cliente passam através da *ActionServlet* antes de executar qualquer ação na aplicação.(SHENOY e MALLYA, 2004)

Na inicialização da aplicação, o *ActionServlet* lê os arquivos de configuração do Struts e os carrega na memória no método *init()*.

Através dos métodos *doGet* ou do *doPost* as requisições são interceptadas e gerenciadas adequadamente.

4.6.3 A classe RequestProcessor

Quando a classe *ActionServlet* é chamada, ela delega o gerenciamento da requisição a outra classe chamada *RequestProcessor* através do seu método *process()*.

A figura 4.3 mostra um fluxograma dos componentes do controlador do Struts gerenciando uma requisição http através do método *process()* da classe *RequestProcessor*.

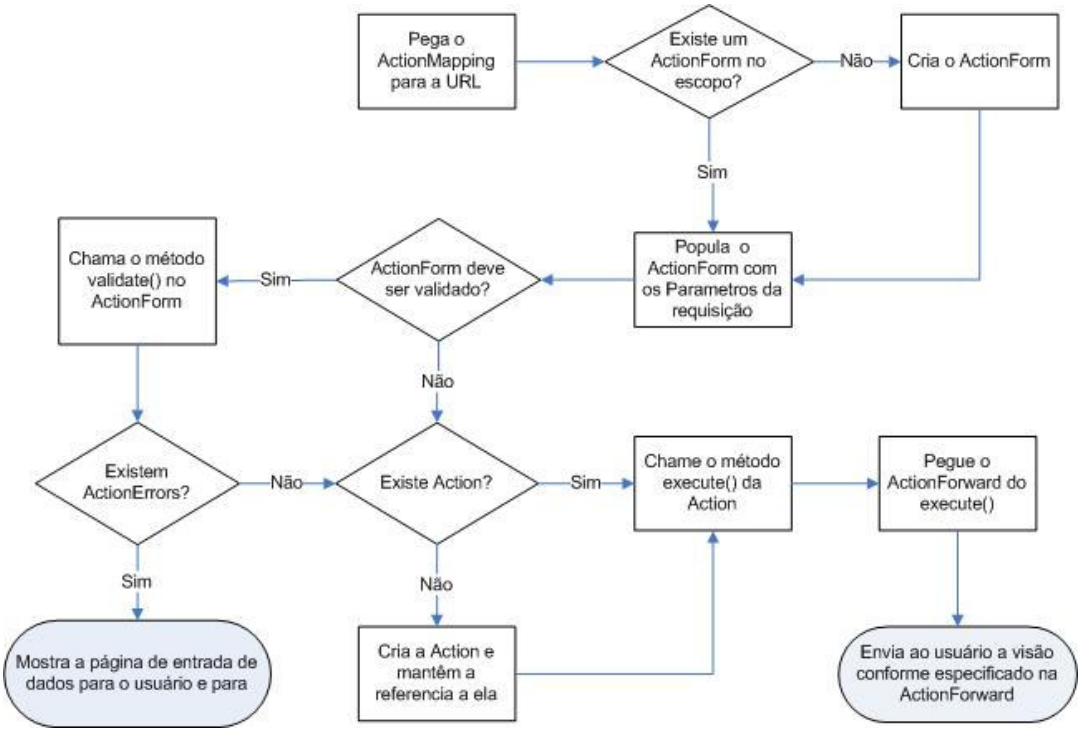

Figura 4.3 - Fluxograma do método *process()* da classe *RequestProcessor* (SHENOY e MALLYA, 2004)

As etapas do processamento de requisições são as seguintes:

1 - O RequestProcessor recupera o bloco para o XML selecionado a partir do arquivo de configuração do struts. Esse bloco é chamado de *ActionMapping* de acordo com a terminologia do struts. Realmente existe uma classe chamada *ActionMapping* no pacote org.apache.struts.action. Essa classe é responsável por manter o mapeamento entre uma URL e uma *Action*. A seguir observamos um exemplo de mapeamento.

```
<!-- ========================== Seção de Definição Action Mapping -->

    <action-mappings>
        <action
            attribute="empresaForm"
```

```
            name="empresaForm"
            scope="request"
            parameter="evento"
            path="/empresa"
            type="br.com.nota_fluxo.action.EmpresaAction"
            validate="false">
            <forward name="lista" path="/form/lista_empresa.jsp" />
            <forward name="editar" path="/form/editar_empresa.jsp" />
        </action>
    </action-mappings>
```

2 - O *RequestProcessor* procura pela URL '/empresa' no arquivo de configuração. O atributo 'type' determina qual classe *Action* deve ser instanciada. O bloco XML possui vários outros atributos. Esses atributos constituem nas propriedades JavaBeans da instância *ActionMapping* para o caminho '/empresa'. A classe *Action* é explicada nos passos a seguir. Por enquanto precisamos saber que a *Action* é a classe que contém a nossa lógica de negócios e que é chamada pelo struts.

Uma vez que cada requisição http é diferenciada da outra pelo seu caminho, existe um e somente um *ActionMapping* para cada caminho.

Outro atributo da *ActionMapping* é o 'name'. Esse é o nome lógico do *ActionForm* que será populado pelo *RequestProcessor*. Depois de selecionar o *ActionMapping*, o *ActionForm* é instanciado. Para tanto, é necessário informar o nome completo (pacote mais arquivo) usando para isso a declaração *FormBeans* como mostrado a seguir:

```
<form-beans>

  <form-bean
        name="empresaForm"
        type="br.com.nota_fluxo.form.EmpresaForm"/>

</form-beans>
```

Essa declaração associa um nome lógico ('empresaForm' no exemplo acima) a uma classe ('br.com.nota_fuxo.form.EmpresaForm' neste mesmo exemplo).

3 - O *RequestProcessor* instancia a EmpresaForm e a coloca em seu escopo determinado – tanto requisição como sessão, através do atributo 'scope' no mesmo *ActionMapping*.

4 - O *RequestProcessor* interage com os parâmetros da requisição http e popula as propriedades do EmpresaForm usando Introspecção Java (uma maneira de fazer reflexão através de Java Beans, usando os métodos de acesso).

5 - O *RequestProcessor* verifica o atributo 'validate' na *ActionMapping*. Se o atributo estiver como verdadeiro (*true*), o *RequestProcessor* chama o método *execute()* do formulário (neste caso, chamado 'EmpresaForm').

6 - O *RequestProcessor* instancia a classe *Action* especificada no *ActionMapping* e chama o método *execute()* na instância da classe *Action* correspondente ('EmpresaAction'). A assinatura do método *execute()* é como descrita abaixo:

```
public ActionForward execute(
                    ActionMapping mapping,
                    ActionForm form,
                    HttpServletRequest request,
                    HttpServletResponse response) throws Exception
```

A *ActionForm* também é disponível na instância *Action*. Foi para isso que a *ActionForm* foi criada, com o objetivo de manter e transferir dados dos parâmetros da requisição http para outros componentes do controlador, ao invés de ter que procurar por esses elementos cada vez que for usar na requisição http.

O *RequestProcessor* cria uma instância da *Action* (EmpresaAction) apenas se ela não existir. Existe apenas uma instância desta classe *Action* na aplicação.

O método *execute()* retorna a próxima visão a ser apresentada ao usuário. Sendo que a *ActionForward* é a classe que encapsula a informação sobre a próxima visão a ser apresentada.

4.6.4 A Classe Action

A classe *org.apache.struts.action.Action* é o coração do framework. Ela é a ponte entre a requisição de um cliente e a lógica dos negócios. Cada classe *Action* é destinada tipicamente para efetuar uma única operação de negócios do interesse do cliente. Uma única operação de negócios não significa que a *Action* não possa efetuar somente uma tarefa. Ao invés disso, a tarefa que ele efetua deve ser coesa e centrada em uma única unidade funcional. Em outras palavras, as tarefas efetuadas pela *Action* devem ser relacionadas a uma operação de negócios. Não deveria ser criado uma *Action* com funcionalidades de cesta de compras e responsabilidade de login e logout juntos, por exemplo. Essas áreas da aplicação não são fortemente relacionadas e não devem ser associadas.

Uma vez que a instância *Action* é determinada, o método *processActionPerform()* é chamado. O método *processActionPerform()* é responsável pela chamada do método *execute()* na instância da classe *Action*. Em versões anteriores

do Struts, a classe continha apenas o método *perform()* que está depreciado em favor do método *execute()* no Struts 1.1 visto que o método *perform()* suporta apenas as exceções *IOException* e *ServletException*. Devido à adição de tratamento de erros, o Framework precisa capturar todas as instâncias de subclasses de *java.lang.Exception* a partir da classe *Action*. (CAVANESS, 2002)

Embora o método *execute()* não seja abstrato, a implementação padrão retorna nulo, então é necessário herdar a classe e sobrescrever este método para utilizá-lo.

4.6.5 DispatchAction

A classe *DispatchAction* é uma subclasse bastante útil da *Action*, conforme mostrado na figura 4.4. Como cada classe *Action* é destinada tipicamente para efetuar uma única operação de negócios do interesse do cliente, seria necessário criar uma classe Action para cada operação. Essa abordagem é valida, entretanto não é adequada, uma vez que pode existir duplicidade de código entre essas *Actions*. (SHENOY e MALLYA, 2004)

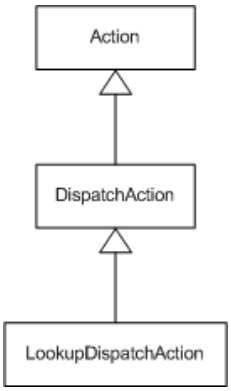

Figura 4.4 – Hierarquia de classes Action (JAKARTA STRUTS FRAMEWORK)

Assim a *DispatchAction* provê mecanismos para simplificar a execução de múltiplas operações, chamando-as pelo nome do método específico ao invés de sobrescrevermos o método *execute()*. Para tanto é necessário especificar no arquivo de configuração do struts o nome do parâmetro que irá apontar para o nome do método a ser utilizado. (SHENOY e MALLYA, 2004)

Para criarmos uma Classe *DispatchAction* devemos seguir os passos abaixo (SHENOY e MALLYA, 2004):

1- Criar uma Subclasse de *DispatchAction*

2- Identificar as ações relacionadas e criar um método para cada uma das ações lógicas.

3 - Identificar o parâmetro da requisição que irá identificar unicamente as ações.

4 - No arquivo de configuração do struts, ao definir uma *ActionMapping*, colocar o atributo *parameter* igual ao parâmetro escolhido no passo anterior.

5 - Colocar na JSP para que o parâmetro identificado no passo 3 tenha os valores da subclasse de *DispatchAction*.

4.6.6 A classe LookupDispatchAction

A classe *LookupDispatchAction* é uma subclasse da *DispatchAction*. A diferença entre elas é que a *LookupDispatchAction* procura no pacote de recursos para então pegar a chave para somente depois pegar o nome do método que foi estabelecido na subscrição do *getKeyMethodMap()*.

Para criarmos uma Classe *LookupDispatchAction* devemos seguir os passos abaixo (SHENOY e MALLYA, 2004) :

1- Criar uma Subclasse de *LookupDispatchAction*

2- Identificar as ações relacionadas e criar um método para cada uma das ações lógicas.

3 - Identificar o parâmetro da requisição que irá identificar unicamente as ações.

4 - No arquivo de configuração do struts, ao definir uma *ActionMapping*, colocar o atributo *parameter* igual ao parâmetro escolhido no passo anterior.

5 - Sobrescrever o método chamado *getKeyMethodMap()*, na subclasse de *LookupDispatchAction*. O método retorna um *java.util.Map*. As chaves usadas para preencher o *Map* devem ser as mesmas do pacote de recursos.

6 - Criar os botões na página JSP usando as mensagens do pacote de recursos de internacionalização no lugar dos seus nomes. Se a tag não for utilizada não existirão benefícios em usar a *LookupDispatchAction*.

A vantagem do uso da classe *LookupDispatchAction* é que nome do método não é determinado pela interface, mas sim pela chave no pacote de recursos. Uma vez que as chaves são sempre as mesmas, a aplicação segue os padrões de internacionalização, ao contrário da *DispatchAction*.

4.6.7 A Classe ActionForward

O método *execute()* retorna a próxima visão a ser mostrada. A *ActionForward* é a classe que encapsula a informação sobre a próxima visão a ser apresentada. O Struts estimula não colocar o nome da página JSP apontando diretamente para a próxima visão, mas sim usar um nome lógico. Essa associação entre o nome lógico e a página JSP física é encapsulada na instância *ActionForward* retornada do método *execute()*.(SHENOY e MALLYA, 2004)

A classe *ActionForward* é um invólucro para que exista o baixo acoplamento do recurso lógico com o recurso físico. O recurso físico é especificado somente no arquivo de configuração (bem como seus atributos de elemento forward: nome, caminho e redirecionamento) e não no próprio código. O despachante das requisições (*Request Dispatcher*) pode efetuar operações tanto para ir adiante (*forward*) quanto para redirecionar (*redirect*) um *ActionForward* dependendo do atributo *redirect* (CAVANESS, 2002) . Assim a diferença é que em um *forward* mantém a solicitação http do contexto, podendo apenas se usado dentro da própria aplicação, enquanto que um *redirect* efetua uma outra solicitação http, podendo estar em outra aplicação ou em outro lugar.

4.7 Configuração de uma Aplicação

Embora o arquivo de configuração de aplicações web, geralmente chamado de *web.xml*, possa configurar qualquer aplicação genérica, existem algumas opções que devem ser configuradas adequadamente para permitir a utilização do framework struts.(CAVANESS, 2002)

A seguir é apresentado um exemplo de arquivo de configuração:

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

<!DOCTYPE web-app
  PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"
  "http://java.sun.com/j2ee/dtds/web-app_2_2.dtd">

<web-app>
      <display-name>Notas Fiscais e Fluxo de Caixa</display-name>
      <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>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>1</load-on-startup>
        </servlet>

        <servlet-mapping>
            <servlet-name>action</servlet-name>
            <url-pattern>*.do</url-pattern>
        </servlet-mapping>

        <!-- Struts Tag Library -->
        <taglib>
            <taglib-uri>/WEB-INF/struts-bean.tld</taglib-uri>
            <taglib-location>/WEB-INF/struts-bean.tld</taglib-location>
        </taglib>

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

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

        <resource-ref>
            <description>Postgres Datasource Teste</description>
            <res-ref-name>jdbc/dspostgres</res-ref-name>
            <res-type>javax.sql.DataSource</res-type>
            <res-auth>Container</res-auth>
        </resource-ref>

</web-app>
```

O passo mais importante aqui é configurar o *ActionServlet*, que receberá todas as requisições para a aplicação. Podemos especificar qualquer nome lógico para o *ActionServlet*, que no exemplo acima tem o nome *Action*, mas temos que lembrar de usar o mesmo nome lógico para configurar o *servlet-mapping* posteriormente.

Carregar os objetos do arquivo de configuração do Struts pode se tornar uma tarefa que consome tempo, e para evitar isso podemos especificar o elemento 'load-on-startup' do servlet. Se o elemento 'load-on-startup' do servlet for igual a 1, esse carregamento é feito na inicialização do contêiner.

Como podemos observar, nos parâmetros especificados para o servlet existe um parâmetro 'config', que determina a localização do arquivo de configuração do struts,

frequentemente chamado de *struts-config.xml*. O parâmetro 'debug' define o nível de detalhe na depuração da execução; um número baixo implica menos detalhes. O parâmetro 'details' pode ser útil para identificar algum erro na carga do arquivo de configuração do struts para os objetos Java, sendo que um número baixo implica menos detalhes.

Depois de definir o servlet, precisamos definir que tipo de URL o servlet irá interceptar. O ideal é definir padrões que sejam caminhos ou sufixos, como no nosso exemplo, onde usamos o prefixo '.do'.

No bloco de recursos é criada uma referência para um recurso criado previamente, como por exemplo mecanismos de conectividade com um banco de dados JDBC (*Java Database Connectivity*).

4.7.1 Arquivo de Configuração do Struts

O Arquivo de configuração do struts é do tipo XML (*eXtensible Markup Language*) , geralmente chamado de *struts-config.xml*, e segue as regras estabelecidas em uma DTD (*Document Type Definition*) correspondente a sua versão, que mostra todos os elementos possíveis dentro de um arquivo de configuração.

Não seria interessante comentar todos os elementos, então observaremos apenas os cinco elementos mais importantes e os seus atributos mais relevantes.

Podemos destacar então:

1 - Seção de definição de *Form Beans*.

2 - Seção de definição de *Global Forwards*.

3 - Seção de definição de *Action Mappings*.

4 - Seção de definição do Controlador.

5 - Seção de definição do Recurso de Mensagens.

Podemos observar o seguinte 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>

<!-- ============================ Seção de Definição de Form Beans -->

    <form-beans>
```

```
      <form-bean
            name="empresaForm"
            type="br.com.nota_fluxo.form.EmpresaForm"/>

    </form-beans>

<!-- ========================== Seção de Definição Global Forward -->

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


<!-- ========================== Seção de Definição Action Mapping -->

    <action-mappings>
        <action
            attribute="empresaForm"
            name="empresaForm"
            scope="request"
            parameter="evento"
            path="/empresa"
            type="br.com.nota_fluxo.action.EmpresaAction"
            validate="false">
            <forward name="lista" path="/form/lista_empresa.jsp" />
            <forward name="editar" path="/form/editar_empresa.jsp" />
        </action>
    </action-mappings>


<!-- ========================== Seção de Definição do Controlador -->

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


<!-- ==================== Seção de Definição Recursos de Mensagens -->

      <message-resources
      parameter="br.com.nota_fluxo.ApplicationResources" />

</struts-config>
```

A seção de definição de *Form Beans* contém uma ou mais entradas para cada *ActionForm*. Cada *Form Bean* é identificado por um nome lógico que deve ser único, e para ser usado este deve possuir uma classe que herde a *ActionForm* e que possua nela os métodos de acesso aos seus atributos. Um fato interessante é que podemos usar várias vezes um mesmo *ActionForm*, uma vez que cada um possui um nome único associado. Essa característica é útil quando desejamos armazenar vários *Form Beans* de um mesmo tipo na mesma sessão do servlet.

A Seção de definição de *ActionMappings* contém o mapeamento URL (*Unified Resource Locator*) para uma classe *Action*, e também associa um Form Bean com um

caminho. O atributo tipo é o nome da classe *Action* associada. Cada classe *Action* deve ter um caminho único dentro do *ActionMapping*. Isso vem do fato que cada caminho URL deve ter um único gerenciador. Não há como associar várias *Actions* com o mesmo caminho. O atributo nome é o *Form Bean* associado a essa *Action*, que deve estar previamente declarado em sua respectiva seção.

Dentro da *ActionMapping* existem dois *forwards*. Esses *forwards* são locais, que só podem ser acessados dentro dessa *ActionMapping*. Por outro lado, os *forwards* definidos na seção *Global Forwards* são acessíveis de qualquer *ActionMapping*. Como é possível observar, um *forward* possui um nome e um caminho. O atributo nome é um identificador lógico, e o caminho é o recuso para o qual o controle será redirecionado.

A próxima seção é a de definição do Controlador. O Controlador é opcional, e se não for especificado será a classe *org.apache.action.RequestProcessor*. Existem casos em que pode ser necessário substituir ou herdar uma classe para que se tenha o um processador próprio especializado. Por exemplo, ao usarmos o plug-in Tiles temos que substituir o processador por *TilesRequestProcessor*.

A ultima seção de interesse imediato é a de Recurso de Mensagens; nesse elemento devemos informar a localização e o nome do arquivo em que se encontra o pacote de Mensagens utilizado nos recursos de internacionalização do Struts.

4.8 Tratamento de Exceções

O Tratamento de Exceções é uma parte crucial no desenvolvimento de aplicações para a Internet (SHENOY e MALLYA, 2004). Antes da versão 1.1 do Struts, o framework possuía um tratamento de exceções limitado. Entretanto, na versão 1.1 foi adicionado ao Struts um pequeno, porém eficiente framework de tratamento de exceções, que permite o tratamento declarativo e programático de erros.

O Tratamento de erros declarativo é o tratamento que deve ser acompanhado da declaração, em um arquivo de configuração, de quais exceções serão gerenciadas e como elas serão tratadas. Essa abordagem torna mais fácil a modificação da lógica de tratamento de exceções sem necessidade de recompilação de código(CAVANESS, 2002) .

Considere o método lista, de uma classe *LookupDispatchAction*:

```
public ActionForward lista
            (ActionMapping mapping
            ,ActionForm form
            ,HttpServletRequest request
            ,HttpServletResponse response) throws Exception{
```

Este possui na sua declaração a classe gerenciamento para exceções *Exception* (herdada por todos os tipos de exceção), o que significa que não é necessário manipular cada tipo de exceção, a menos que se queira que essas mensagens sejam apresentadas para o usuário de uma forma mais apropriada, como mostrá-la em uma pagina de erros da própria aplicação. (CAVANESS, 2002)

Pode ser interessante escolher onde poderemos visualizar essa exceção. Assim, podemos declará-la no arquivo de configuração do struts, sendo que toda a vez que ela ocorrer, a requisição será direcionada para uma página específica.

Se desejarmos que esse tratamento ocorra com as exceções SQL(*Structured Query Language*) de instruções que efetuam acesso a banco de dados basta declararmos a exceção dentro da *ActionMapping* ou de um *GlobalForward* da seguinte maneira:

```
<exception
      key="error.excecao_sql"
      path="/execaosql.jsp"
      type="java.sql.SQLException"/>
```

A chave indica a mensagem que deve ser apresentada a partir do Arquivo de recursos da aplicação.

Eliminar as redundâncias é um dos benefícios do tratamento declarativo.

O Tratamento Programático é o oposto do tratamento declarativo, envolvendo a maneira tradicional de escrever código específico para gerenciar as exceções, ao invés de simplesmente modificar um arquivo externo de configuração(CAVANESS, 2002) .

```
public ActionForward lista
            (ActionMapping mapping
            ,ActionForm form
            ,HttpServletRequest request
            ,HttpServletResponse response) throws Exception{

      ...
      ActionErrors errors = new ActionErrors();
      ...
      try{
            beanForm
                  .setEmpresa_lista
                        (beanSQL.select
                              (con
                              ,p_pagina
                              ,beanForm.getEmpresa_pesq()
                              ,beanForm.getNr_reg()
                              , "")
                        );
            beanForm.setContadoresPaginas
                  (beanSQL.getContaSelect()
                  ,beanSQL.getContaSelectAtual()
                  );
```

```
        } catch (Exception e) {
              errors.add(ActionErrors.GLOBAL_ERROR
              ,new ActionError("error.sql", e.toString()));
        }
        ...
        saveErrors(request, errors);
}
```

Frequentemente é necessário gerar um erro e mostrá-lo ao usuário. Entretanto, quando um usuário informa um registro que já existe por exemplo, não faz sentido mostrar uma mensagem de erro e redirecionar para outra página. Uma opção melhor é mostrar o erro e permitir que o usuário verifique o problema (SHENOY e MALLYA, 2004).

O único problema dessa abordagem é que ela leva a escrever o mesmo código redundante em quase todas as classes *Action*. (CAVANESS, 2002) .

4.9 Plug-ins do Struts

4.9.1 Validador

Um dos benefícios do Struts é a capacidade de validar dados. Caso qualquer validação falhe, a aplicação mostra o formulário HTML novamente para que os dados sejam corrigidos, permitindo que não seja necessário se preocupar com os mecanismos que capturam e mostram dados inválidos. O validador foi incorporado ao Struts pois geralmente é necessário fazer a mesma validação em vários campos, e porque não é bom que o código tenha que ser compilado ao colocar uma nova validação para um campo, bem como pode ser necessário validar campos no lado do cliente antes de enviar o formulário.

4.9.1.1 Distribuição das validações

A dificuldade para utilização do design pattern MVC para desenvolver aplicações baseadas na Web começa do fato que estas aplicações são intrinsecamente divididas ente o cliente e o servidor. A visão, sem dúvida, é mostrada no cliente, mas o modelo e o controlador podem teoricamente ser divididos entre o cliente e o servidor. O desenvolvedor é forçado a dividir a validação da aplicação Web no estágio da implementação sempre na fase inicial de projeto. Em contraponto, o MVC na sua concepção é independente de divisões. Em aplicações centralizadas, o modelo, a visão e o controlador são hospedados e executados em um único espaço de endereçamento no qual detalhes de particionamento não interessam. A independência de divisões de validação é uma das características do MVC, uma vez que as dependências de

localização não devem guiar decisões de arquitetura nem de projeto (LEFF e RAYFIELD, 2001).

Não há dúvida que os desenvolvedores podem simplesmente dividir a validação de uma aplicação (LEFF e RAYFIELD, 2001), decidindo que o método será executado no servidor e um outro método será no cliente. Uma vez que o particionamento foi efetuado, o MVC pode ser aplicado, em paralelo, na implementação das partes cliente e servidora da aplicação. O problema dessa abordagem advém do fato de ser sempre muito difícil fazer uma divisão correta no inicio da fase de projeto uma vez que essa decisão depende de requisitos que em geral mudam consideravelmente no curso do projeto. O problema ainda é dificultado pelo fato de depender de fatores de ambiente estáticos (poder de processamento de servidores) e dinâmicos (congestionamento da rede).

Na maioria das vezes o controle da validação não é uma característica independente e com grande capacidade de alteração. Ao invés de ser uma função apenas das características intrínsecas de aplicações Web, a divisão da validação depende de decisões de tecnologia que não possuem relacionamento com a aplicação. Dessa maneira, os desenvolvedores de aplicação ficam presos às suas tecnologias de implementação. Entretanto, muitos desenvolvedores consideram que applets são ruins (LEFF e RAYFIELD, 2001) porque são pesados e devido aos potenciais problemas de segurança. No entanto, também há desenvolvedores que afirmam (LEFF e RAYFIELD, 2001) que não é possível construir visões sofisticadas em HTML. Uma vez que as escolhas de tecnologias são feitas, elas determinam como a aplicação deve ter a sua validação dividida.

Em resumo, certamente aplicações Web podem utilizar o padrão de projeto MVC quando:

- A divisão das rotinas de validação é conhecida.

- A infra-estrutura da tecnologia é compatível com este particionamento.

Sem dúvidas, o que realmente necessitamos é evitar o particionamento enquanto implementamos a aplicação, uma vez que a decisão correta depende de fatores que não são necessariamente determinados até a fase de desenvolvimento atual.

Atrasar a divisão das rotinas de validação o maior tempo possível é bem mais atrativo por que o particionamento é planejado na maneira pela qual visões e a lógica de negócios são necessárias para os controladores (LEFF e RAYFIELD, 2001).

4.9.1.2 Validações no Cliente

As validações do lado do cliente são inseguras pois é fácil enviar uma página web e evitar qualquer script executado na página original. Assim, não podemos contar com as validações do lado do cliente, entretanto elas podem ser úteis, uma vez que evitam o envio imediato da página ao servidor, o que economiza tempo e largura de banda. Sendo assim, o ideal seria validar os dados tanto no servidor quanto no cliente usando o mesmo conjunto de regras. (HUSTED, DUMOULIN, FRANCISCUS et. al, 2003)

4.9.1.3 Características do Validador

O validador é configurado por arquivos XML, que geram regras de validação para campos em seu formulário.

Vantagens da utilização do Validador

- Algoritmos de validação javascript para tipos básicos como datas e inteiros são fornecidos, e se for necessário pode-se criar validações com suas próprias regras.
- Expressões regulares podem ser usadas para a validação de padrões como e-mail e CEP.
- Suporte a validação de acordo com a localidade do browser do cliente.
- Suporte para validação de múltiplas paginas.
- Único ponto de manutenção: validações do lado do cliente e do lado do servidor são geradas a partir da mesma configuração.
- Validações personalizadas podem ser definidas como expressões constantes ou no código Java.
- Manutenção: não é necessário alterar o código para aplicar a validação, é necessário apenas coloca-la no arquivo de configuração.

Como desvantagem da utilização do Validador podemos citar a falta de conversões e transformações nos dados.

É importante destacar que algumas validações só podem ser feitas no lado do servidor, como por exemplo, podemos ver se o nome do usuário e a senha obedecem o tamanho mínimo e máximo de caracteres, mas não podemos verificar se a senha e nome do usuário são válidos, pois para isso precisaríamos acessar os dados da aplicação.

4.9.1.4 Utilização do Validador

Para utilizar o validador é necessário especificar no arquivo de configuração do struts e indicar onde estão localizados os arquivos de configuração do validador:

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

No struts, as validações se localizam em dois arquivos XML – o *validation.xml* e o *validation-rules.xml*. O *validation-rules.xml* contém o conjunto de regras que estão prontas para serem utilizadas. O segundo arquivo – *validation.xml* é especifico da aplicação. Ele é o responsável por associar o arquivo que contém as regras com cada campo de um form do Struts. Por exemplo, existe uma regra chamada de *required*, que verifica se um campo foi preenchido. Com essa regra pode-se verificar se qualquer campo é nulo, apenas adicionando uma a seguinte declaração no *validation.xml*:

```
<form name="/empresa">
      <field property="cd_emp" depends="required">
      ..
      </field>
      <field ..
      </field>
      ..
</form>
```

O XML acima contém um bloco com um elemento <form> que procura um *ActionMapping* com o nome "/empresa". Todas as associações de regras para o form deste *ActionMapping* definido no arquivo de configuração do Struts, podem ser associadas com a regra *required*.

No arquivo *validation-rules.xml* podemos ver o código usado para o a validação da regra *required:*

```
<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.ActionErrors,
                  javax.servlet.http.HttpServletRequest"
                  msg="errors.required">
            </validator>
            <validator name="…">
            … …
            </validator>
      </global>
</form-validation>
```

Observamos que o validador define o nome da regra, o nome da classe que é usada, e os parâmetros que ela recebe para efetuar a validação. É necessário especificar também uma chave no arquivo de recursos do sistema para a mensagem de erro.

Assim é necessário colocar as mensagens de erro fornecidas pelo validador no arquivo de recursos do sistema. Este passo só precisa ser efetuado uma vez, e posteriormente apenas para cada regra adicionada.

```
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.
```

A primeira coluna corresponde aos valores das chaves, e a segunda coluna corresponde às mensagens do validador. Os números dentro de chaves correspondem aos parâmetros passados por cada campo validado, que deveria ser o nome do mesmo, especificado pelo usuário, e deveria estar especificado também no mesmo arquivo.

```
empresa.cd_emp=Código da Empresa
```

Podemos ver um arquivo de configuração um pouco mais completo, passando o nome do campo para a mensagem de erro:

```
<form-validation>
	<formset>
		<form name="/empresa">
			<field property="cd_emp"
				depends="required ">
				<arg0 key="empresa.cd_emp"/>
			</field>
		</form>
	</formset>
</form-validation>
```

Existem regras que podem depender de outras, por exemplo, para verificar se um valor é menor do que o outro, é necessário verificar se o campo está preenchido, se não estiver a verificação não é executada.

Para validações como de máscaras por exemplo, podemos efetuar validações utilizando expressões regulares. Como a maior parte das máscaras é utilizada mais de uma vez, pode ser interessante declará-la como uma variável global e referenciá-la posteriormente, como no exemplo abaixo.

Nesse exemplo, a expressão pode ser lida como “O campo pode ter qualquer número de caracteres desde que estejam no intervalo de “A” à “Z” e de “a” à “z”.

```
<form-validation>
	<global>
		<constant>
			<constant-name>mascara</constant-name>
			<constant-value>^[A-Za-z]*$</constant-value>
		</constant>
	</global>
	<formset>
		<form name="/empresa">
			<field property="nm_emp"
				depends="required,mask">
					<arg0 key="empresa.nm_emp"/>
					<var>
						<var-name>mask</var-name>
						<var-value>${mascara}</var-value>
					</var>
			</field>
		</form>
	</formset>
</form-validation>
```

Os passos necessários para a utilização do Validador Struts, usando o *ValidatorActionForm*, são os seguintes:

1 - Criar uma classe especifica da aplicação que herde a classe *ValidatorActionForm*

2 - Colocar no *validation.xml* o *ActionMapping* correspondente que usa a classe criada no passo anterior.

3 - Listar as regras para executar as dependências do campo.

4 - Para cada regra, adicionar a mensagem que será usada ao mostrar o erro.

5 - Para cada regra, fornecer os parâmetros usados.

6 – Se as regras de validação no *validation-rules.xml* não cumprem os requisitos, adicione novas regras.

É possível também, herdar a *ValidatorForm* ao invés da *ValidatorActionForm*, e pelo no *validation.xml* chamá-lo pelo nome do da classe ao invés do *ActionMapping*, entretanto a validação seria executada em todas as *ActionMapping* que a utilizassem a classe, e não para cada *ActionMapping* em separado, o que nem sempre é desejável. Em solicitações distintas podem ser necessárias validações específicas.

4.9.2 Tiles

A aparência de uma aplicação geralmente é o ultimo detalhe a ser aperfeiçoado e consequentemente mudará com freqüência. (HUSTED, DUMOULIN, FRANCISCUS et. al, 2003). Como qualquer outro componente, as páginas contêm muitos elementos em comum, como cabeçalhos, rodapés, menus entre outros. Esses elementos podem ser facilmente copiados de uma página para a outra. Entretanto essa solução é inaceitável até mesmo para as menores aplicações. Cada vez que um rodapé ou um cabeçalho for alterado, essas mudanças têm que ser aplicadas em cada página manualmente. Consequentemente, quaisquer alterações no layout da página terão que ser efetuadas em todas as páginas (SHENOY e MALLYA, 2004). Na figura 4.5 temos um exemplo de um layout de uma página da Internet.

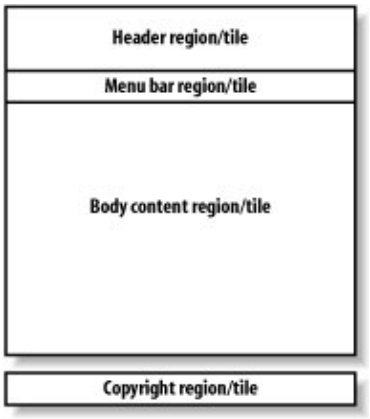

Figura 4.5 - Exemplo de layout de uma página da internet(CAVANESS, 2002)

O uso de *includes* de JSP é melhor do que a alternativa anterior, uma vez que evita a repetição de formatação. São criadas páginas para cada elemento que se deseja incluir. O cabeçalho e o rodapé são adicionados na página através da diretiva de inclusão. Quando o cabeçalho ou o rodapé mudarem isso afetará apenas os arquivos correspondentes. (SHENOY e MALLYA, 2004)

Entretanto a forma como os elementos são dispostos na página podem mudar a qualquer momento durante o desenvolvimento, implicando que de qualquer forma cada página JSP tenha que ser modificada de acordo com a nova organização proposta.

A abordagem de utilização de *includes* (inclusões) possui todo o layout baseado em tabelas HTML, sendo que todas as chamadas para o *include* devem ser colocadas em cada página.

O objetivo principal do Tiles é tirar o layout comum de páginas, de forma que possa ser reusado em várias JSPs.

No tiles, os layouts representam a estrutura da página como um todo. O Layout é simplesmente uma JSP. Podemos associar essa estrutura a um molde com encaixes. È possível colocar outras JSPs nesses encaixes de forma declarativa. Por exemplo, podemos criar um layout com cabeçalho, corpo e rodapé. No arquivo de configuração do tiles, determinamos quais páginas serão colocadas em cada encaixe. (SHENOY e MALLYA, 2004)

Em tempo de compilação, o framework tiles compõe a página usando o layout e preenchendo seus encaixes com cada JSP individualmente. Em essência o tiles é um framework que permite a apresentação conjunta de páginas a partir de componentes. Cada parte pode ser reusada livremente na aplicação. Isso reduz a quantidade de formatação necessária para manter e alterar a aparência de uma aplicação. O tiles usa uma biblioteca de tags para implementar os moldes.

Na abordagem de *includes*, todas as páginas embutidas (cabeçalho, rodapé etc.) fazem parte de um JSP principal. No Tiles, todas as páginas JSP fazem parte de um layout JSP antes de serem apresentadas. Essa abordagem reduz a redundância de HTML e proporciona o reuso máximo de lógica de formatação. (SHENOY e MALLYA, 2004)

Considerações:

1 - Embora o Tiles permita a construção de páginas de diversas maneiras, algumas podem não ser muito vantajosas sobre a abordagem de includes.

2 – Não é possível chamar diretamente o *layout* de páginas JSP a partir do browser. Só é possível acessar as definições do struts a partir do controlador e não por requisições externas.

4.10 Tecnologias Auxiliares

4.10.1 Log4j

O Log4j faz parte do Projeto Jakarta da *Apache Software Foundation*, e é uma ferramenta muito útil para depuração e tratamento de exceções. Essa tecnologia fornece um mecanismo para controlar declarativamente as mensagens de log, de forma que os códigos de testes e de produção sejam os mesmos, e a sobrecarga de processamento seja mínima quando o log estiver desativado. Ele é responsável por cuidar de qualquer tipo de mensagem, desde alertas a mensagens de erro. (SHENOY e MALLYA, 2004).

Vantagens sobre o gerenciamento de erros pelo contêiner:

- O comando para escrever a mensagem é sincronizado com as entradas e saídas do disco, tornando a aplicação mais lenta.

- Os erros do contêiner são gravados no console, entretanto procurar no console por uma exceção não é adequado em um sistema que estiver em produção.

- Nem sempre os erros do contêiner são guardados.

- Mesmo se o log do console for redirecionado para um arquivo de saída, ainda assim o arquivo pode ser sobrescrito quando o servidor for reiniciado.

- É mais vantajoso utilizar a ferramenta Log4j do que usar os comandos para escrever mensagens destinadas a depuração durante a fase de testes. Esta outra opção implica que seja necessário apagar as mensagens antes do programa ser utilizado no ambiente de produção, o que não garante que o código seja o mesmo código de testes.

4.10.2 Ant

O Ant é uma ferramenta para automação de construção de aplicações independente de plataforma. O Ant é um projeto de código aberto, que faz parte do projeto jakarta da *Apache Software Foundation*. Em poucas palavras o ANT pode, por exemplo, atualizar localização das classes, compilar o código separando os binários do código fonte em pastas distintas, gerar arquivos de distribuição, gerar documentação do projeto, efetuar testes de unidade no código, configurar e executar a aplicação.

A automação de tarefas não é novidade, entretanto o que faz o Ant ser poderoso é o fato de ser independente de sistema operacional. Além disso, é independente de ambiente de desenvolvimento sendo que a maior parte destes ambientes já trabalham com o Ant. Uma de suas principais vantagens é que seu arquivo de configuração em XML torna-o fácil de aprender. (CAVANESS, 2002)

5. Desenvolvimento de uma Aplicação usando Struts

Neste trabalho de conclusão de curso estaremos avaliando a adequação da utilização do Struts para desenvolvimento de uma aplicação Web.

O projeto será de uma Aplicação para Geração de notas fiscais com fluxo de caixa, projetado especificamente para pequenas empresas, que deverá ser capaz de emitir notas fiscais de acordo com os dados informados pelo usuário. A aplicação deverá permitir o lançamento de contas a pagar e a receber, enviando os seus lançamentos para uma conta caixa, que deve registrar todo o fluxo de dinheiro da empresa.

Esse projeto contará com suporte a múltiplas empresas, pois é muito comum empresas utilizarem o mesmo espaço físico e recursos técnicos e administrativos. Ao fazer o login, a aplicação já terá a informação de qual empresa é o usuário que está utilizando a aplicação, permitindo emissão de notas, pagamento e recebimento de contas apenas da respectiva empresa identificada.

5.1 Projeto UML

5.1.1 Casos de Uso

Neste capítulo utilizaremos descrições narrativas de processos do domínio, comumente chamados de casos de uso, para melhorar a compreensão sobre requisitos do problema.

Não existe um formato rígido para casos de uso, eles podem ser alterados para possibilitar uma maior clareza de comunicação. (LARMAN, 2000)

Usaremos casos de uso de alto nível para descrever sucintamente os eventos de um agente externo que usa um sistema para completar um processo, sendo útil principalmente na especificação inicial de requisitos e escopo do projeto.

Especificaremos também os casos de uso expandidos pois descrevem uma seqüência de eventos mais detalhada, para permitir uma compreensão mais completa de requisitos e processos, uma vez que casos de uso de alto nível podem ser vagos sobre decisões de projeto.

O diagrama da figura 5.1 mostra um conjunto de casos de usos para um sistema, a relação entre atores e casos de uso, para possibilitar uma rápida compreensão sobre os

atores e a maneira como eles interagem no sistema. Neste diagrama foram identificados três casos de uso principais: gerar nota fiscal, receber pagamento e pagar conta.

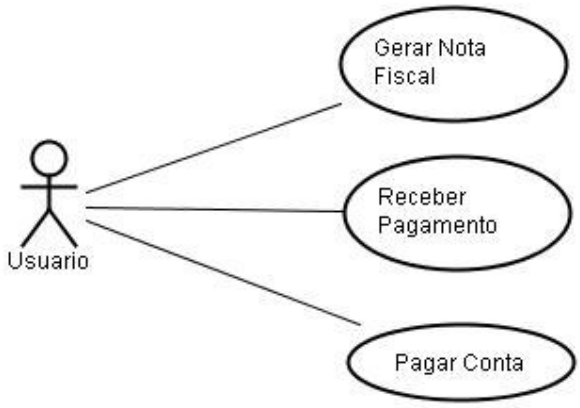

Figura 5.1 – Diagrama de Casos de Uso

Gerar Nota fiscal - Alto Nível

Caso de Uso	Gerar Nota Fiscal
Atores	Usuário, Cliente
Tipo	Primário
Descrição	Um usuário deseja emitir uma nota fiscal de acordo com os serviços prestados a um cliente. O usuário digita os dados da nota e a confirma. A nota é impressa e entregue ao cliente

Gerar Nota fiscal - Expandido

Caso de Uso	Gerar Nota Fiscal
Atores	Usuário, Cliente
Finalidade	Entregar uma nota fiscal para o cliente
Visão Geral	Um usuário deseja emitir uma nota fiscal de acordo com os serviços prestados a um cliente. O usuário digita os dados da nota e a confirma. A nota é impressa e entregue ao cliente
Tipo	Primário
Referências Cruzadas	

Seqüência Típica de Eventos

Ação do Ator	Resposta do Sistema
Este caso de uso inicia quando um usuário deseja emitir uma nota fiscal de acordo com os serviços prestados a um cliente.	
O usuário da Empresa informa o seu Cliente, a Data da Emissão da nota, a Data do Vencimento.	

O usuário preenche a nota informando o CNAE (Código Nacional de Atividade Econômica) e o Valor de cada serviço prestado pela empresa	Os dados do valor total da nota - Valor Bruto, Valor Líquido, IRRF Retido, IRRF a Pagar, COFINS Retido, PIS Retido, CSLL Retido, ISS Retido, ISS A pagar, Total Impostos Retidos, CST, CFPS,IRLP e Contribuição Social - são calculados e apresentados para o usuário A descrição de cada CNAE (Código Nacional de Atividade Econômica) é mostrada ao usuário
O usuário solicita a confirmação da exatidão dos dados contidos na nota	O sistema exibe uma mensagem comunicando que os dados não poderão mais ser alterados e que uma conta a receber será gerada para a nota O sistema disponibiliza a impressão da nota
O usuário solicita a impressão da nota	Exibe a nota na tela
A nota é impressa e entregue ao Cliente	

Recebimento do Pagamento de uma nota Fiscal - Alto Nível

Caso de Uso	Recebimento do Pagamento de uma nota Fiscal
Atores	Usuário, Cliente
Tipo	Primário
Descrição	O usuário deve informar o valor recebido pelo cliente e em que a que conta e nota fiscal o pagamento se destina

Recebimento do Pagamento de uma nota Fiscal - Expandido

Caso de Uso	Recebimento do Pagamento de uma nota Fiscal
Atores	Usuário, Cliente
Finalidade	Dar baixa de débitos referentes a serviços prestados
Visão Geral	O usuário deve informar o valor recebido pelo cliente e em que a que conta e nota fiscal o pagamento se destina
Tipo	Primário
Referências Cruzadas	

Seqüência Típica de Eventos

Ação do Ator	Resposta do Sistema
Este caso de uso inicia quando um cliente deseja pagar uma nota fiscal e solicita o recebimento a um usuário do sistema	
O Cliente entrega o valor ao usuário	
O usuário deve informar o valor recebido pelo cliente e em que a que conta e nota	O recebimento é registrado automaticamente no movimento de contas

fiscal o pagamento se destina.	

Pagamento de Contas - Alto Nível

Caso de Uso	Pagamento de Contas
Atores	Usuário, Cliente
Tipo	Primário
Visão Geral	O usuário deve informar o valor pago e a que a que conta o pagamento se destina

Pagamento de Contas - Expandido

Caso de Uso	Pagamento de Contas
Atores	Usuário, Cliente
Finalidade	Dar baixa de débitos referentes a serviços solicitados pela empresa
Visão Geral	O usuário deve informar o valor pago e a que a que conta o pagamento se destina
Tipo	Primário
Referências Cruzadas	

Seqüência Típica de Eventos

Ação do Ator	Resposta do Sistema
Este caso de uso inicia quando um usuário deseja pagar uma conta	
O usuário deve informar o valor, os acréscimos e os descontos, o valor pago e que tipo de contas a pagar se classifica e a que a que conta o pagamento se destina.	O valor total do que deve ser pago é apresentado. O pagamento é registrado automaticamente no movimento de contas

5.1.2 Contratos

Os contratos auxiliam na definição do comportamento do sistema e definem o efeito das operações (LARMAN, 2000). O comportamento descreve o que o sistema faz sem dizer como é feito. Para cada contrato é descrita informalmente a finalidade da operação, suas pré e pós-condições. As pré e pós-condições são orientadas a mudanças de estado e não a ações, pois devem ser declarações sobre os resultados e não sobre as ações a serem executadas.

Nome	entrarNota(data_emissao, data_vencimento, cliente)
Responsabilidades	Registrar a entrada de uma nota mostrando dados do cliente
Tipo	Sistema
Referências Cruzadas	Casos de Uso: Gerar Nota Fiscal
Notas	
Exceção	Se o Cliente for inválido mostrar erro
Saída	

Pré-Condições	O Cliente é conhecido do sistema
Pós-Condições	Se for nova nota, uma Nota foi criada (criação de instância) Nota.cliente recebeu o valor de cliente (modificação de atributo) Nota.dt_ref recebeu o valor de data_emissao (modificação de atributo) Nota.dt_vencto recebeu o valor de data_vencimento (modificação de atributo) A nota foi associada a um Cliente (formada uma associação)

Nome	entrarItem(cnae, valor)
Responsabilidades	Registrar a entrada de um item e acrescenta-lo em uma nota. Exibir a descrição do CNAE (Código Nacional de Atividade Econômica) e
Tipo	Sistema
Referências Cruzadas	Casos de Uso: Gerar Nota Fiscal
Notas	
Exceção	Se o Cliente for inválido mostrar erro
Saída	
Pré-Condições	A Nota já existe O CNAE (Código Nacional de Atividade Econômica) e é conhecido do sistema
Pós-Condições	Se for novo item, um Item foi criado (criação de instância) Item.cnae recebeu o valor de cnae(modificação de atributo) Item.valor recebeu o valor de valor (modificação de atributo) Item foi associado à Nota O item foi associado a um Cnae (formada uma associação)

Nome	confirmarNota ()
Responsabilidades	Confirmar entrada de uma nota, calculando os seus impostos, o valor líquido e o valor bruto geral. Coloca o valor no contas a receber.
Tipo	Sistema
Referências Cruzadas	Casos de Uso: Gerar Nota Fiscal
Notas	
Exceção	Se não existirem itens mostrar erro
Saída	
Pré-Condições	A Nota já existe Já existem itens associados à Nota
Pós-Condições	Conta_receber foi criada Conta_receber.valor recebeu o Nota.valor_bruto (modificação de atributo) Conta_receber.dt_vencto recebeu o Nota.dt_vencto (modificação de atributo) Conta_receber foi associada à Nota (formada uma associação)

Nome	solicitaImpressao()
Responsabilidades	Imprimir uma nota confirmada.
Tipo	Sistema
Referências	Casos de Uso: Gerar Nota Fiscal

Cruzadas	
Notas	
Exceção	Se a nota não estiver confirmada mostrar erro
Saída	
Pré-Condições	A Nota já existe Já existem itens associados à Nota
Pós-Condições	

Nome	receberPagamento(data, valor, conta)
Responsabilidades	Receber o pagamento de um cliente dando baixa na conta paga
Tipo	Sistema
Referências Cruzadas	Casos de Uso: Recebimento do Pagamento de uma nota Fiscal
Notas	
Exceção	
Saída	
Pré-Condições	O Conta_receber já existe
Pós-Condições	Conta_receber.dt_pagto recebeu data (modificação de atributo) Conta_receber.vl_recebido recebeu valor (modificação de atributo) Conta_receber.nr_conta recebeu conta (modificação de atributo) Um Movimento_contas foi criado Movimento_contas.nr_conta recebeu conta Movimento_contas.vl_movto recebeu valor

Nome	pagarConta (data_vencimento, data_pagamento, valor_pagar, valor_pago, conta)
Responsabilidades	Dar baixa de pagamento de contas da Empresa no sistema
Tipo	Sistema
Referências Cruzadas	Casos de Uso: Pagamento de Contas
Notas	
Exceção	Se a nota não estiver confirmada mostrar erro
Saída	
Pré-Condições	
Pós-Condições	Se for novo pagamento, uma Conta_pagar foi criada (criação de instância) Conta_pagar.dt_pagto recebeu data_pagamento (modificação de atributo) Conta_pagar.dt_vencto recebeu data_vencimento (modificação de atributo) Conta_pagar.vl_pagar recebeu valor_pagar (modificação de atributo) Conta_pagar.vl_pago recebeu valor_pago (modificação de atributo) Conta_pagar.nr_conta recebeu conta (modificação de atributo) Um Movimento_contas foi criado Movimento_contas.nr_conta recebeu conta Movimento_contas.vl_movto recebeu valor_pago

5.1.3 Classes

Com os casos de uso e os contratos especificados, pudemos distinguir as seguintes entidades:

- Empresa
- Cliente
- Cnae
- Contas
- Contas_pagar
- Contas_receber
- Movimento_contas
- Nota_fiscal
- Nota_item
- Ramo_atividade
- Tipo_contas_pagar
- Tipo_contas_receber

Pudemos assim definir as características de cada entidade:

- Empresa – Representa a prestadora de serviços, que irá fornecer a Nota Fiscal aos clientes.
- Cliente – Representa o freguês, a quem será entregue a Nota Fiscal Relativa aos serviços prestados.
- CNAE – Representa o Código Nacional de Atividade Econômica que deve ser usado para descrever itens de Notas fiscais.
- Contas – Representa as contas da empresa, incluindo o caixa(dinheiro que está na empresa) e as contas bancárias.
- Contas_pagar – Representa as contas que a empresa deve pagar.
- Contas_receber – Representa as contas que a empresa tem a receber.
- Movimento_contas – Representa a movimentação de contas da empresa, ou seja, se u fluxo de caixa, apresentando o que saiu e o que entrou na empresa.
- Nota_fiscal – Representa uma Nota Fiscal referente a serviços prestados, devendo possuir o valor dos impostos referentes.
- Nota_item – Representa os itens da nota, descritos pelo CNAE (Código Nacional de Atividade Econômica) com o respectivo valor de cada serviço.
- Ramo_atividade – Representa Ramos de Atividade onde está a empresa.

- Tipo_contas_pagar – Usado para informar se a conta a pagar é de água, luz, telefone e etc...
- Tipo_contas_receber – Usado para informar a origem de contas a receber.

As entidades e seus atributos foram representados na forma de classes, que estão representadas no diagrama de classes da Figura 5.2.

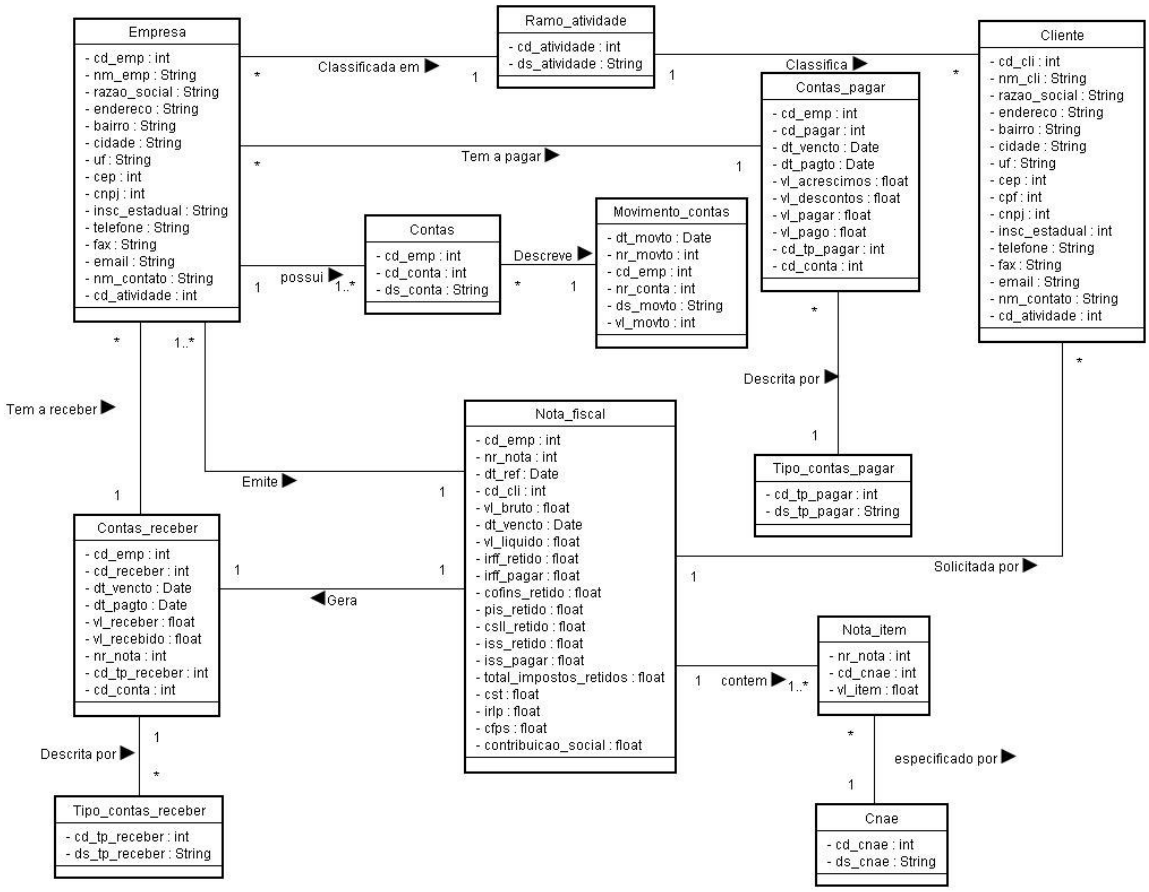

Figura 5.2 – Diagrama de Classes

As empresas possuem clientes em comum, evitando que seja necessário cadastrar os mesmos clientes para cada uma delas. Clientes podem ser tanto pessoas físicas quanto pessoas jurídicas, devendo assim ser especificado ou o Código de Pessoa Física ou o Código de Pessoa Jurídica.

A cada operação de confirmação de nota, pagamento ou recebimento de contas, é gerado um movimento de conta, com valor positivo se o for geração de notas e recebimento e negativo se for pagamento, para permitir um acompanhamento do fluxo de caixa das entradas e saídas da empresa.

5.1.4 Diagramas de Seqüência

Os diagramas de seqüência mostram como os objetos interagem com o uso de mensagens para realizar tarefas. Para tanto já devem estar definidos os casos de uso e contratos. O ponto de partida das interações é a satisfação das pós-condições dos contratos das operações. (LARMAN, 2000)

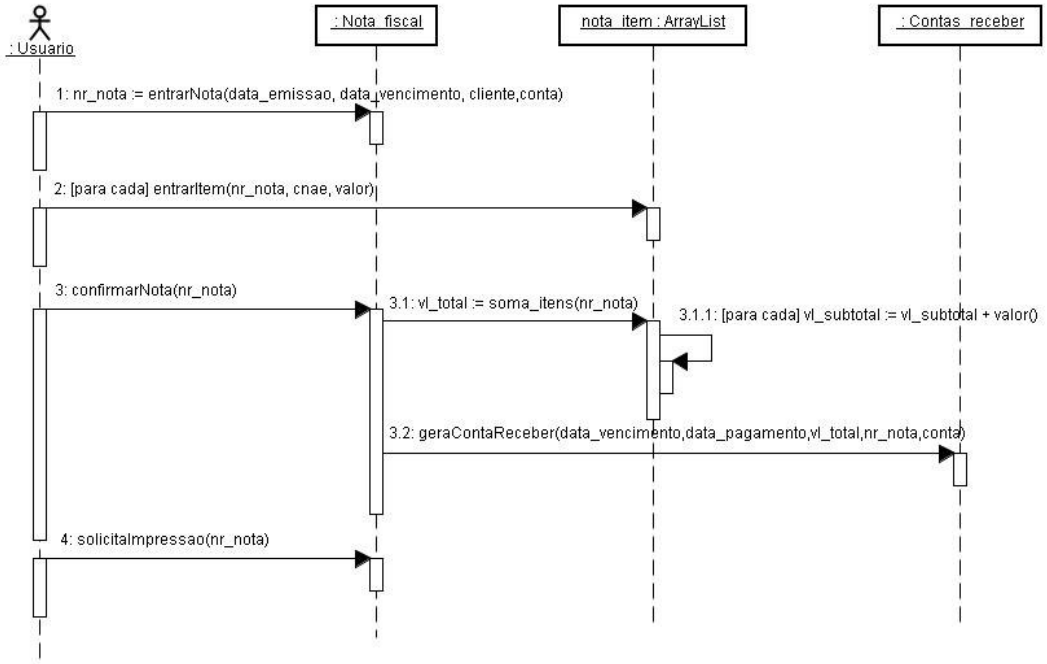

Figura 5.3 - Diagrama de Seqüência de Entrada de Notas

Na figura 5.3 podemos observar os procedimentos usados para efetuar uma geração de nota fiscal, sendo que devem ser informados os dados iniciais da nota como cliente a que se destina, data em que está sendo emitida e a data de seu vencimento. Posteriormente deve ser informado cada item da nota especificado por seu CNAE(Código Nacional de Atividade Econômica). A nota deve ser confirmada para que seus dados não possam mais ser alterados, gerando autormaticamente uma conta a receber para que possa finalmente ser impressa.

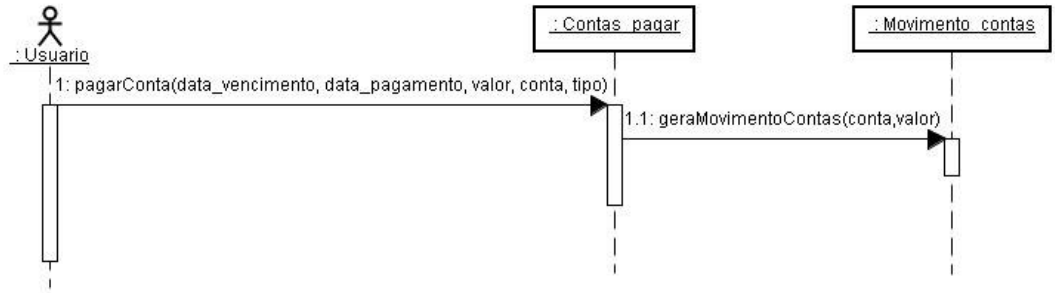

Figura 5.4 - Diagrama de Seqüência de Pagamento de Contas

Na figura 5.4 podemos observar o procedimento usado para efetuar o pagamento de uma conta, devendo ser informada a data do vencimento, a data em que está sendo paga, o valor e a que conta do movimento de contas está destinada. Cada pagamento de conta gera automaticamente um movimento de contas.

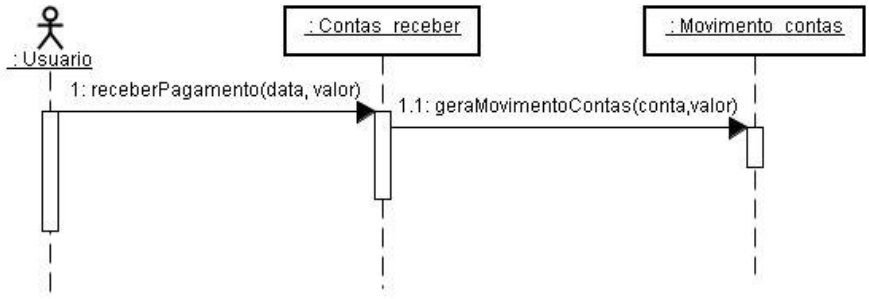

Figura 5.5 - Diagrama de Seqüência de Recebimento

Na figura 5.5 temos o procedimento para receber o pagamento de uma conta, devendo ser informada a data de seu recebimento, o valor recebido e a que conta do movimento de contas está destinada. Cada pagamento de conta gera automaticamente um movimento de contas.

5.1.5 Diagrama de Estado

Diagramas de Estado tem o propósito de mostrar os eventos dos casos de uso. O diagrama é composto por estados e transições. Estado é a condição de um objeto em certo momento no tempo. Transição é a relação entre dois estados, ocasionada por eventos.

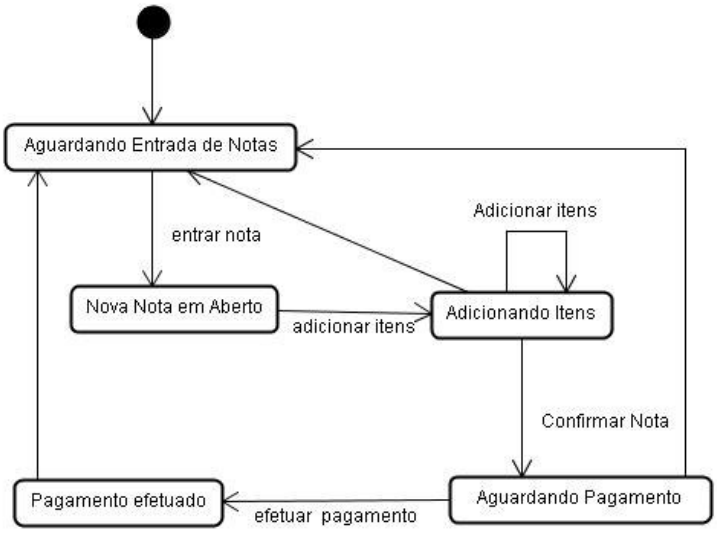

Figura 5.6 - Diagrama de Estados

O Diagrama de Estados apresentado na figura 5.6 ilustra todo o processo de geração de notas fiscais. Inicialmente, o sistema se encontra aguardando a entrada de novas notas. Para cada nota em aberto deve(m) ser inseridos o(s) item(ns) descrito(s) por seu(s) respectivo(s) CNAE(s) - Código Nacional de Atividade Econômica. Após a confirmação da nota é aguardado o pagamento, que pode ser efetuado posteriormente. Como podemos ver, em qualquer momento é permitido entrar novas notas, mesmo que ainda existam notas não confirmadas ou que os pagamentos não tenham sido efetuados.

6. Implementação

6.1 Metodologia

6.1.1 Definição de Atividades

As seguintes atividades serão realizadas durante a execução deste projeto:

- Delimitação do escopo da aplicação a ser implantada
- Modelagem da aplicação selecionada.
- Estudo bibliográfico
- Instalação e testes das ferramentas selecionadas
- Verificação das alternativas disponíveis para cada camada, suas vantagens e desvantagens.
- Estabelecimento padrões de apresentação e de organização de cada camada
- Funcionamento de plug-ins do Framework.
- Verificação do uso de Tecnologias correlacionadas (que fazem parte do mesmo projeto do Struts)
- Desenvolvimento das camadas da Aplicação
- Relato das dificuldades e limitações encontradas

6.1.2 Ferramentas Utilizadas

As seguintes ferramentas foram utilizadas durante a elaboração deste trabalho de conclusão de curso:

- Plataforma de desenvolvimento Java: j2sdk 1.4.2_04
- Ambiente de Desenvolvimento: Eclipse 2.1.3
- Servidor de Aplicações – Contêiner de Servlets: Tomcat 4.1.29
- Framework: Struts 1.2.4
- Banco de Dados: Postgres 7.3
- Ferramenta Interface de Banco de Dados: PgAdmin 3

- Ferramenta de Modelagem UML: Jude 1.3
- Ferramenta para Depuração: Log4j 1.2.8
- Ferramenta para construção da aplicação: Ant 1.5.3

Todas as ferramentas baseadas em Java utilizam o *Java 2 Software Development Kit Standard Edition* que fornece a máquina virtual Java, APIs (*Application Program Interfaces*) e outros componentes para desenvolvimento de aplicações escritas em Java, como compiladores e depuradores.

O Eclipse é uma plataforma robusta e de qualidade comercial para o desenvolvimento de ferramentas integradas. Essa ferramenta pode se tornar bastante poderosa pois permite o uso de plug-ins, que são extensões que podem ser colocadas para facilitar o processo de desenvolvimento (ECLIPSE PROJECT).

Em nosso trabalho utilizaremos o contêiner servlet Tomcat, que é usado na implementação oficial de referência para a tecnologia Java Servlets e Java Server Pages. O Tomcat é desenvolvido em um ambiente aberto e participativo, sob a licença da *Apache Software Foundation*.

O Banco de Dados utilizado é o Postgres, que foi escolhido por não possuir custo de licenças para o software, por ser portável a praticamente todas as plataformas Unix e na plataforma Windows através do Framework Cygwin.

A ferramenta auxiliar para desenvolvimento e gerenciamento do Banco de Dados Postgres é a PgAdmin, que possui código aberto e está disponível tanto para plataforma Linux quanto para Windows.

A ferramenta Jude para modelagem UML em Java é a única ferramenta utilizada que não possui código aberto, entretanto sua licença é gratuita na versão não comercial.

A ferramenta para depuração do código fonte utilizada foi o Log4j, e para a construção da aplicação foi o utilizado o ANT. Ambos foram descritos na seção 4.10 como tecnologias auxiliares do Struts.

6.1.3 Recursos Utilizados

Os seguintes recursos foram empregados para o desenvolvimento da aplicação:

- Software:
    - Todas as Ferramentas já mencionadas anteriormente

- Sistema Operacional: Windows ou Linux

- Hardware:
  - Computador com processador de no mínimo 1GHz, 256 MB de memória RAM ou mais, e disco de no mínimo 10GB (para o banco de dados).

Para execução da aplicação no servidor, são necessários os seguintes recursos:

- Software:
  - Plataforma de desenvolvimento Java: j2sdk 1.4.2_04
  - Servidor de Aplicações – Contêiner de Servlets: Tomcat 4.1.29
  - Banco de Dados: Postgres 7.3
  - Sistema Operacional: Windows ou Linux, sendo que as ferramentas mencionadas anteriormente devem ser as respectivas a cada plataforma, e que o Framework Struts e o Log4j já estão embutidos na aplicação.

- Hardware:
  - Computador com processador de no mínimo 1GHz, 256 MB de memória RAM ou mais, e disco de no mínimo 10GB (para o banco de dados).

A aplicação foi testada sob a plataforma Microsoft Windows, entretanto é teoricamente portável em plataformas Linux, uma vez que toda a aplicação é baseada em Java e o Contêiner de Servlets (Tomcat) e o Banco de dados (Postgres) estão disponíveis para a mesma.

6.2 Considerações sobre a Implementação

Nesse capítulo serão apresentadas as escolhas efetuadas na implementação do projeto, bem como a justificativa para cada uma delas.

6.2.1 Controlador

A classe *Action* possui apenas um método que é executado quando enviamos a página: o método *execute()*. É necessário tratar o fluxo das operações através de comparações ou de obrigar o usuário a criar uma classe *Action* para cada atividade desejada.

A classe *DispatchAction* é uma subclasse da *Action* que permite definir no arquivo de configuração do Struts um parâmetro. Nos botões ou *links* da interface com o usuário, esse parâmetro deve ser especificado e a classe *DispatchAction* deve possuir um método com o mesmo valor passado ao parâmetro.

Para o desenvolvimento da aplicação optamos por estender a classe *LookupDispatchAction* pois esta possui a mesma função da *DispatchAction* de separar os métodos, entretanto é mais aconselhada pois usa os recursos de internacionalização. Essa classe obriga o desenvolvedor a sobrescrever o método *getKeyMethodMap()*, que deve mapear as mensagens do arquivo de recursos de internacionalização com o nome do método a ser chamado. Nessa classe não podemos sobrescrever o método execute(), senão a classe funcionará da mesma forma que a sua superclasse *Action*. Se desejarmos executar uma operação padrão devemos sobrescrever o método *unspecified()*. No caso da nossa aplicação, desejamos que a operação padrão seja a lista com todos os objetos referentes ao *ActioForm* desta *LookupDispatchAction*. A figura 6.1 mostra a classe EmpresaForm herdando a *LookupDispatchAction*.

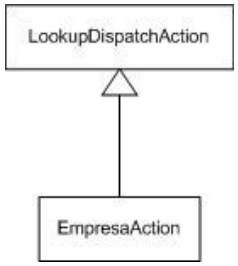

Figura 6.1 – Exemplo de herança da LookupDispatchAction

É importante observar que é no arquivo de configuração do struts é onde a visão e o controlador são acoplados, conforme observamos na declaração do *ActionMapping* abaixo:

```
<action-mappings>
    <action
        attribute="empresaForm"
        name="empresaForm"
        scope="request"
        parameter="evento"
        path="/empresa"
        type="br.com.nota_fluxo.action.EmpresaAction"
```

```
            validate="false">
            <forward name="lista" path="/form/lista_empresa.jsp" />
            <forward name="editar" path="/form/editar_empresa.jsp" />
        </action>
    </action-mappings>
```

Podemos observar que são especificados os Form Beans que já devem estar previamente declarados neste arquivo, o escopo (requisição, seção ou aplicação), o URL em que estará disponível e finalmente o nome da sublcasse de *Action* que deve ser executada. O campo relativo à validação está declarado como falso pois pretendemos fazer a validação apenas ao inserir e alterar, dentro das sublcasses de *Action*. Ainda no *ActionMapping* podemos explicitar as páginas que podem ser apresentadas ao usuário. Dentro da sublasse de *Action* é que será definido para qual das duas páginas a aplicação será direcionada.

6.2.2 Visão

Para os Form Beans que serão optamos por usar a classe *ValidatorActionForm* ao invés da classe *ActionForm*, pois desejamos utilizar o validador. Optamos ainda por não utilizar a classe *ValidatorForm*, pois não desejamos ter uma validação para cada Form Bean, que poderá ser usado por diversas *Action*, mas sim efetuar a validação em cada *Action* em separado. Optamos ainda por não usar as classes *DynaActionForm*, *DynaValidatorForm* e *DynaValidatorActionForm* porque os atributos que são declarados no arquivo de configuração do Struts não são verificados em tempo de compilação, portanto eventuais erros apareceriam apenas em tempo de execução conforme discutido na seção 4.5.5.

A nossa classe PadraoForm herda a *ValidatorActionForm*, e possui atributos para auxiliar a passagem de páginas, que devem ser comuns a todas as telas. A figura 6.2 mostra a herança destas respectivas classes conforme utilizado na aplicação.

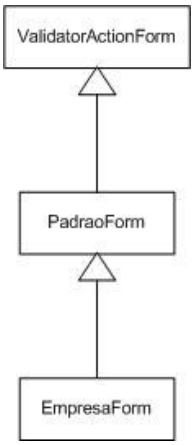

Figura 6.2 – Herança da classe ValidatorActionForm

Ainda na visão, só que na apresentação para o usuário escolhemos utilizar as bibliotecas de tags fornecidas pelo Struts, uma vez que as referências bibliográficas são bem mais detalhadas para essa opção.

6.2.3 Modelo

Para a camada do modelo, que representa as entidades usadas no sistema, criamos uma classe que mapeia os valores obtidos diretamente do banco de dados.

Existem métodos para quatro operações básicas do banco de dados: inserir, alterar, apagar e selecionar.

O método selecionar recebe o objeto *ActionForm* como parâmetro para executar a consulta, recebe também em que registro deve começar, a quantidade de registros selecionados e a conexão com o banco de dados. Um método é chamado para carregar variáveis desta classe, como numero total de registros relacionados e o número de registros apresentados. Também é chamado outro método encarregado de popular os atributos e retornar um conjunto de objetos do mesmo tipo do que foi inicialmente passado como parâmetro.

O método inserir recebe o objeto e a conexão com o banco, não sendo necessário o retorno de nenhum valor.

No método alterar o objeto, são passados a conexão e o número identificador da linha na tabela. O uso do identificador da linha simplifica a passagem de parâmetros quando são usadas chaves compostas.

No método apagar é passada a conexão e o número identificador da linha na tabela.

6.2.4 Opções gerais do Framework

Plug-ins

Optamos por não usar o plug-in Tiles, pois a aplicação não requer um grande número de telas, o que pode ser gerenciável ainda com *includes* JSP.

Conexão

Preferimos que o próprio contêiner servlet gerencie a conexão com o banco de dados, uma vez que o contêiner pode gerenciar melhor os recursos disponíveis para ele do que a funcionalidade de acesso a banco de dados embutida no Struts. A conexão com o banco será disponibilizada no escopo da sessão assim que o login for efetuado, para que não seja necessário estabelecer uma conexão a cada requisição, o que tornaria a aplicação bem mais lenta.

Exceções

Para o tratamento programático de exceções o Struts fornece um mecanismo para enviar mensagens ao usuário, conforme descrito na seção 4.8.

Para o tratamento efetivo de exceções criamos a classe ExcecaoGeral, que ao ser criada pode possuir um atributo do tipo *ActionError*. O uso apenas dos *ActionErrors* não interrompe o fluxo de execução do programa, o que é recomendado a maioria das vezes por questões de desempenho.

Tipos

Os tipos dos atributos foram alterados para *String*, para que pudessem ser suportados pelo *framework* Struts.

6.3 Funcionamento da aplicação

Para poder executar a aplicação, o usuário deve informar o nome do usuário e uma senha que sejam válidos no sistema. Para isso é criada uma conexão com o Banco de Dados, e caso os parâmetros informados estejam corretos, a conexão é disponibilizada para toda a seção do usuário. Podemos observar a tela de login na figura 6.3.

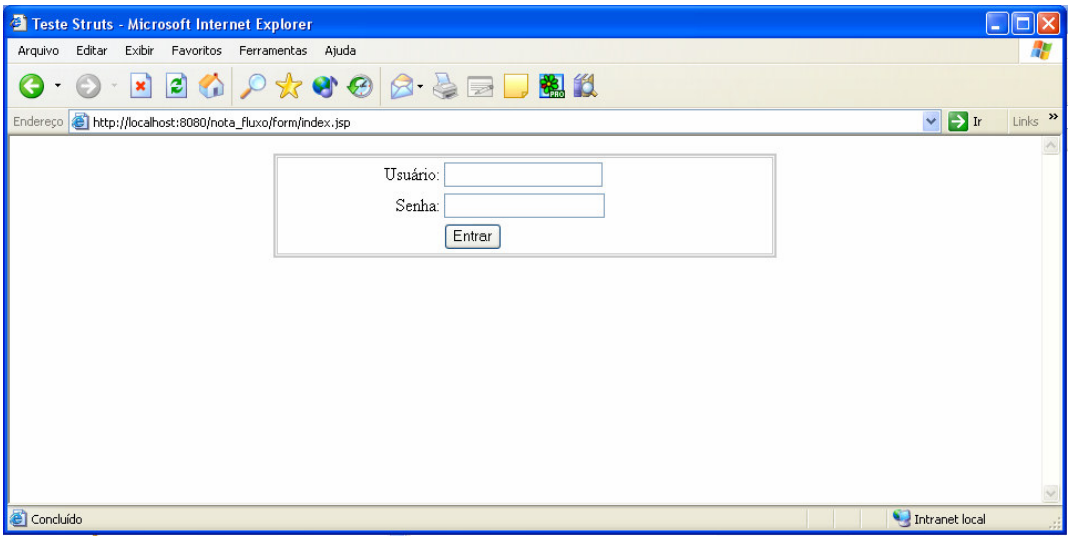

Figura 6.3 - Tela de Login

Todas telas seguirão um padrão estabelecido no desenvolvimento, no qual apenas os objetos e seus atributos serão diferentes. Assim, ao entrar em uma tela, será mostrada inicialmente a lista contendo os registros da entidade do banco de dados. É permitido que o usuário navegue entre os itens, indo para o próximo, o anterior, para o primeiro e para o ultimo registro da lista conforme mostrado na tela 6.4.

É desta tela que são chamadas todas as demais telas referentes a uma determinada entidade: consultar, inserir, alterar e apagar.

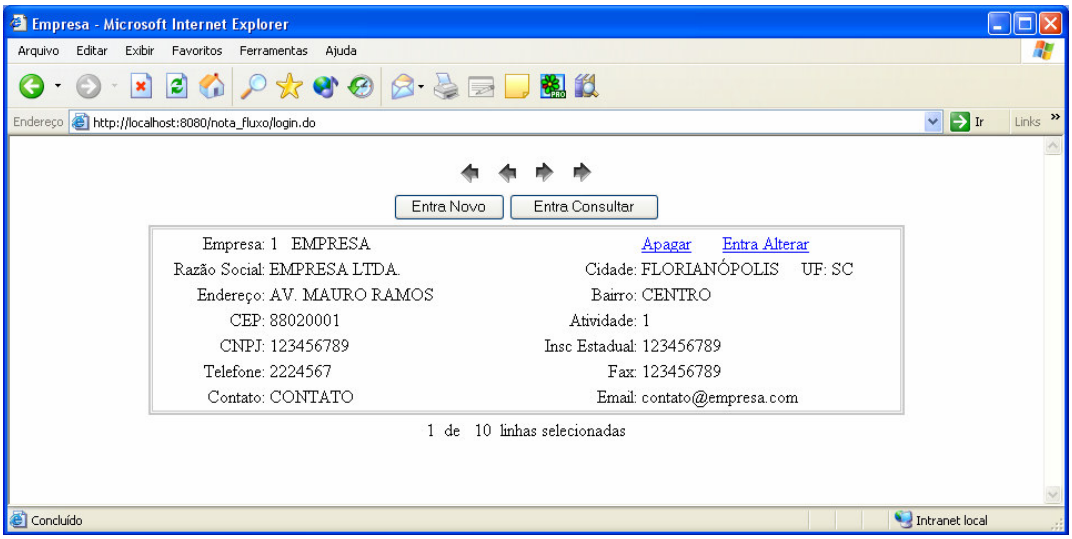

Figura 6.4 – Tela que apresenta lista os registros

Ao entrar na tela para entrar uma nova consulta deve ser apresentado o formulário com a ultima consulta executada no sistema, caso seja a primeira vez retornará em branco, conforme mostrado na figura 6.5. Ao preencher os campos, a página com a lista é retornada novamente, mas apenas com os registros que valores iguais aos da consulta. Se um campo nulo for informado na consulta, não são pesquisados os campos nulos, sendo retornado registros com todos os valores possíveis para este campo.

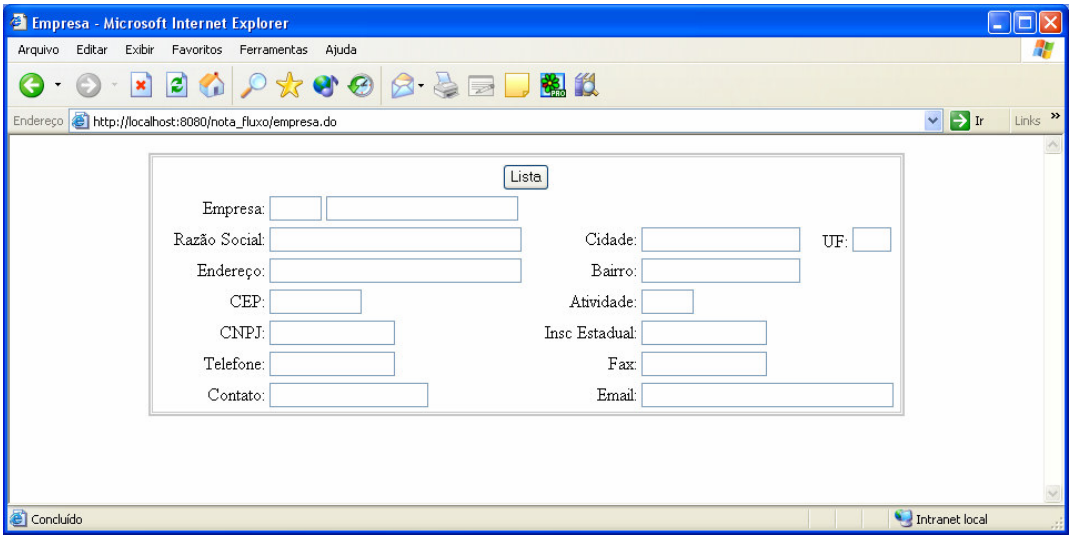

Figura 6.5 - Tela de consulta

Na tela para inserir um novo registro, é apresentado um formulário em branco, para que o usuário possa informar os dados solicitados. A inclusão só será confirmada ao clicar no botão 'Novo', conforme ilustrado na figura 6.6.

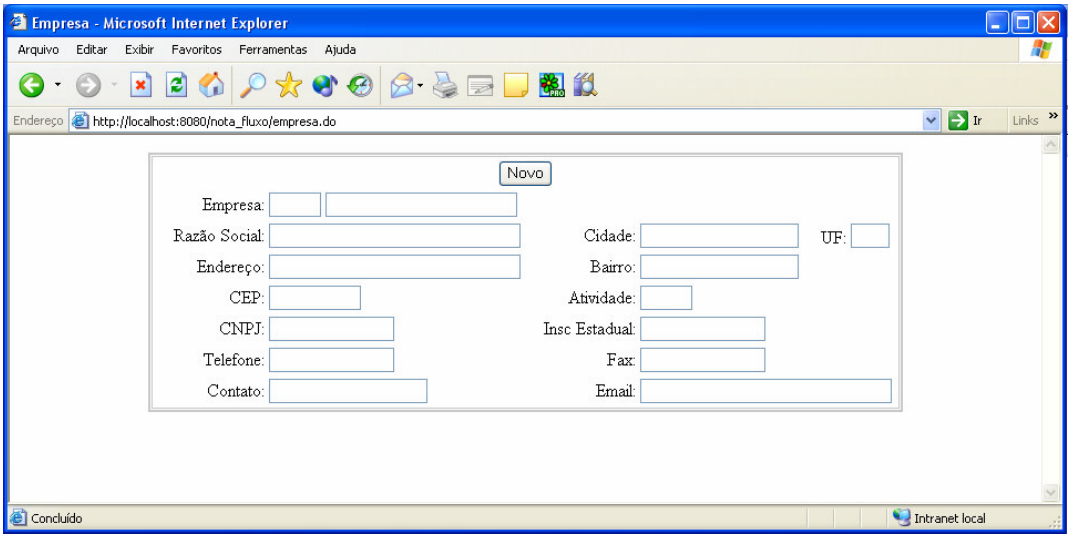

Figura 6.6 – Tela para inserir um novo registro

Na tela de alteração, os dados do registro listado atualmente são apresentados em um formulário para que seja possível modificar os seus valores, conforme mostrado na figura 6.7. A Alteração só será confirmada ao clicar no botão 'Alterar'.

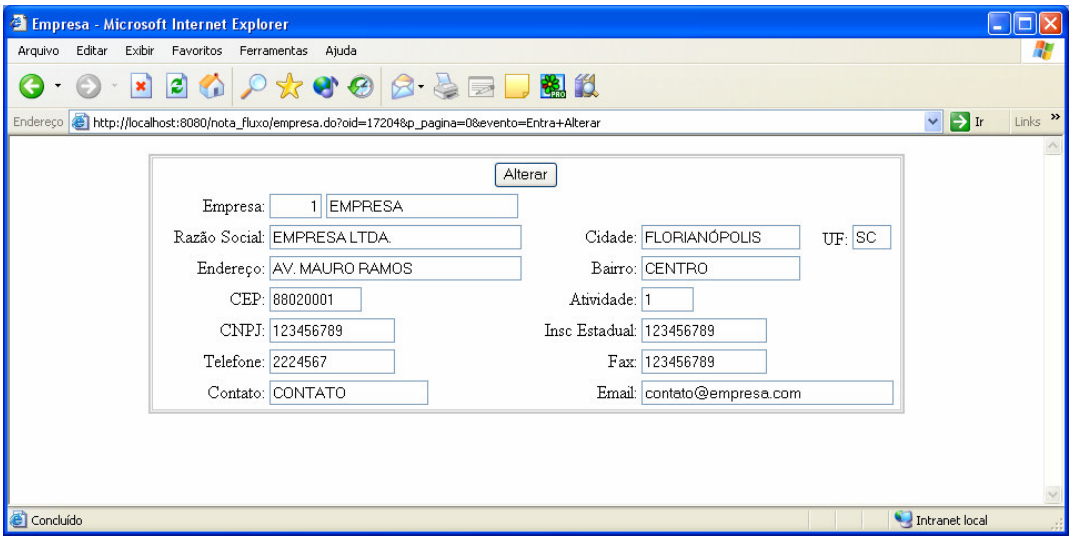

Figura 6.7 – Tela para alterar um registro existente

Ao excluir o registro atual, o usuário é questionado se realmente deseja apaga-lo, para evitar que registros sejam apagados acidentalmente, conforme ilustrado na figura 6.8.

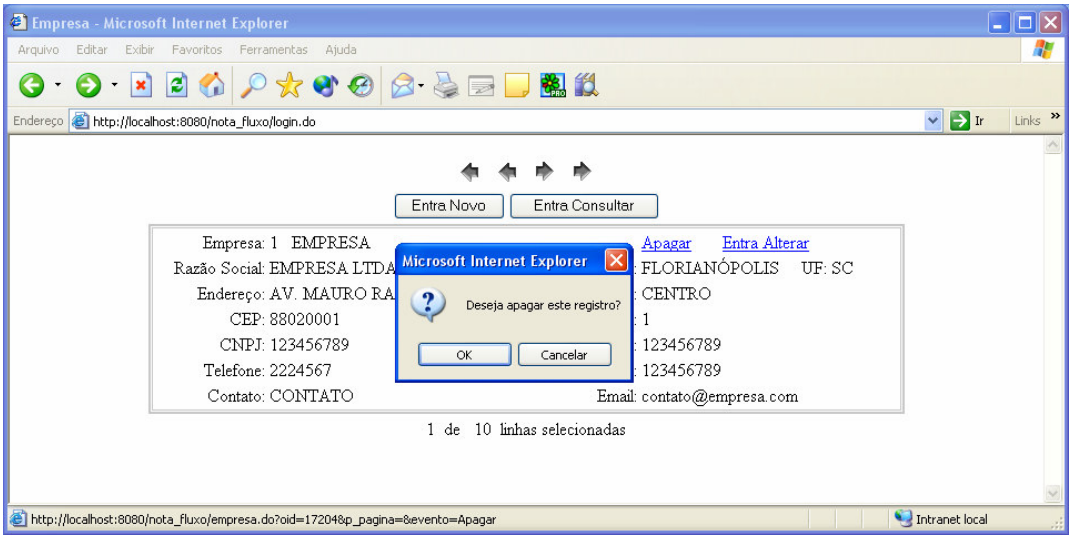

Figura 6.8 – Tela para confirmar a exclusão de um registro

7. Análise do Framework

7.1 Vantagens e Limitações do Framework Struts

Neste capítulo serão analisados os resultados obtidos com o desenvolvimento de uma aplicação Web, descrita nos capítulos anteriores, utilizando o Framework Struts, enfatizando as vantagens e limitações encontradas no framework, as mudanças ocorridas no processo de desenvolvimento, e as alterações no custo, no desempenho e na escalabilidade do produto final.

7.1.1 Pontos fracos do Struts

Ao longo do desenvolvimento da aplicação, as principais limitações encontradas no Framework Struts foram as seguintes (HUSTED, DUMOULIN, FRANCISCUS et. al, 2003):

- Inexistência de um modelo de eventos: O struts está intrinsecamente ligado ao modelo solicitação-resposta usado pelo protocolo HTTP, o que pode ser um fator limitante para desenvolvedores acostumados com eventos.
- Dificuldades para depuração: Não existe nenhum suporte para a depuração automática. Os desenvolvedores tem que fazer uso da criação de "pontos de interrupção" manuais escrevendo mensagens no registro do contêiner ou usando outra tecnologia como o log4j por exemplo.
- Modelo de Dados: O modelo de dados é deixado completamente em aberto para o desenvolvedor, não deixando nenhum tipo de recomendação nem possuindo um modelo de dados padrão.
- *ActionServlet*: Apenas um *ActionServlet* é usado em uma aplicação, o que pode ocasionar conflitos na configuração.
- Compreensão dos Componentes: É necessário o conhecimento dos componentes do Struts para que se consiga trabalhar com ele, incluindo o conhecimento das classes e de como elas interagem.
- Fornecedor sem Prioridade no Suporte: A *Apache Software Foundation* é mantida por voluntários, de forma que nem sempre é possível obter respostas e correções de forma imediata.
- Tamanho da Lista de E-mails: A lista da comunidade do Struts aumentou significativamente, e com isso pode ser difícil encontrar a informação desejada.

- Demora na atualização das versões do framework: As versões do Struts podem demorar a sair em comparação a outros produtos, sendo que muitos desenvolvedores têm que usar construções em fase de teste para poderem usar as últimas alterações.
- Limites da internacionalização: Os recursos para internacionalização funcionam adequadamente para mensagens de erro e rótulos, mas não são adequados para gerenciar blocos maiores de texto. Blocos maiores de texto necessitam de elementos de formatação (inicio de parágrafo, elementos em negrito ou itálico, fontes de tamanhos diferentes), o que é possível fazer, incluindo elementos de formatação HTML diretamente nas mensagens, entretanto não é uma solução adequada, pois mistura a formatação com o texto.
- Exceções do JSP: As exceções do JSP não são localizadas, de forma que aparecem sempre em inglês.

7.1.2 Pontos Fortes do Struts

Como pontos fortes do Struts para desenvolvimento de aplicações Web podemos destacar as seguintes características:

- Centrado no Modelo Solicitação-Resposta: O Struts é desenvolvido no modelo solicitação-resposta do protocolo HTTP, sendo bem familiar para desenvolvedores web.
- Registro Padrão: O Struts não necessita de outro pacote para ser instalado, configurado ou entendido no contêiner.
- Registro da depuração opcional: O Struts grava opcionalmente mensagens de status que podem ser úteis para depuração.
- Modelo Neutro: O Struts não é atrelado a nenhuma camada particular de persistência.
- Configuração centralizada: A configuração do Struts encapsula detalhes para uma aplicação ou módulo.
- Arquivo de Recursos específicos: O Struts permite o uso de um arquivo de recursos específico para cada local, tornando possível a internacionalização de aplicações.
- Simplicidade: O Struts possui relativamente poucas classes básicas para que os desenvolvedores aprendam.

- Código Aberto: O Código Fonte é aberto, fazendo com que todos possam saber o conteúdo do framework.
- Comunidade de Desenvolvedores: Muitas versões com correções fornecidas por desenvolvedores estão disponíveis.
- Comunidade de Revendedores: O Struts é incorporado em vários outros produtos.
- Versões estáveis: As versões formais do Struts ficam em um período de testes sem prazo definido, para assegurar que o produto tenha uma alta qualidade.
- Extensões Tag: O Struts inclui um conjunto de extensões de tags de uso geral, além das que usam recursos do framework, para evitar a utilização de *scriptlets.*
- Bem documentado: A documentação do Struts geralmente é bem detalhada, de forma que raramente será necessário verificar o código fonte da aplicação.
- Baseado em Padrões: O Framework implementa vários padrões em sua arquitetura, de forma a serem bem familiares para a maior parte dos desenvolvedores.
- Extensível: Todas as definições padrão podem ser configuradas. O desenvolvedor pode personalizar classes como *Action* e *ActionForm* bem como carregar as subclasses em seus lugares.

7.2 Custo de Implantação do Struts

O custo total de implantação do struts não envolve apenas dinheiro, mas também tempo e aprendizado. Iremos abordar quatro fatores: custo de licença, custo de treinamento, custo de suporte e de manutenção. (HUSLY e YU, 2002)

Custo de licença: É um software de código aberto, sob a licença da *Apache Software Foundation*, portanto é gratuito. Para revendedores independentes de software, o framework pode ser redistribuído sem o pagamento de *royalties* sob uma licença relativamente liberal (é necessário consultar advogados para obter mais detalhes).

Custo de treinamento: É um dos maiores interesses quando se está adotando um novo framework. É importante lembrar que o Struts é um framework construído sobre a plataforma J2EE. Portanto é necessário que o desenvolvedor tenha experiência em JSP, Servlets e sobre o conceito MVC para assimilar as principais características do Struts usando uma a duas semanas no mínimo. Para os desenvolvedores que não possuírem

esses conhecimentos prévios, logicamente levará mais tempo. Pode levar ainda mais tempo para que o aprendizado se torne produtivo, uma vez que aprender as características não é o mesmo que aprender o paradigma.

É interessante considerar o treinamento com instrutores para desenvolvedores que estarão expostos a uma plataforma totalmente nova. O custo em termos do dinheiro envolvido pode ser muito maior nesse caso. Entretanto antes de avaliar um custo de treinamento é importante verificar o nível do conhecimento dos desenvolvedores. O que é importante ressaltar com isso é que o custo com o treinamento com o Struts é relativamente menor do que com o treinamento em JSP e Servlets.

Custo de manutenção e suporte: O custo de suporte é difícil de mensurar, uma vez que como faz parte de uma comunidade de software aberto, um suporte excelente porém informal está disponível em listas de correspondência do Struts. Existe suporte pago de alguns vendedores como o Multitask (www.multitask.au). Existe outra opção de ter suporte diretamente de pessoas com grande conhecimento em Struts, uma vez que o seu código é disponível gratuitamente e pode ser modificado. Para projetos de desenvolvimento, uma regra que é comumente usada para avaliar do custo do suporte do Struts é igualar a metade do custo do suporte que um programador J2EE. (HUSLY e YU, 2002)

7.3 Análise de Escalabilidade e Desempenho

Escalabilidade geralmente significa escalabilidade em tempo de execução. Entretanto outro tipo de escalabilidade é igualmente importante: a escalabilidade do tamanho do projeto.

Escalabilidade em tempo de execução: O Struts compartilha muitas características do padrão Servlet. Os Servlets são muito leves, de forma que poucos recursos são necessários para inicializar e manter uma instância. Através da arquitetura Servlet, uma instância gerencia requisições simultâneas(GABHART, 2003). Assim as *Actions* são sem estado e são compartilhadas por várias *threads* (linhas de execução). Dessa forma para cada requisição é criada uma nova linha de execução e o método *service()* é executado em cada *thread* conforme mostrado na figura 7.1. Isso significa que o número de instâncias *Action* é constante e não aumentam com o volume de tráfego. (HUSLY e YU, 2002)

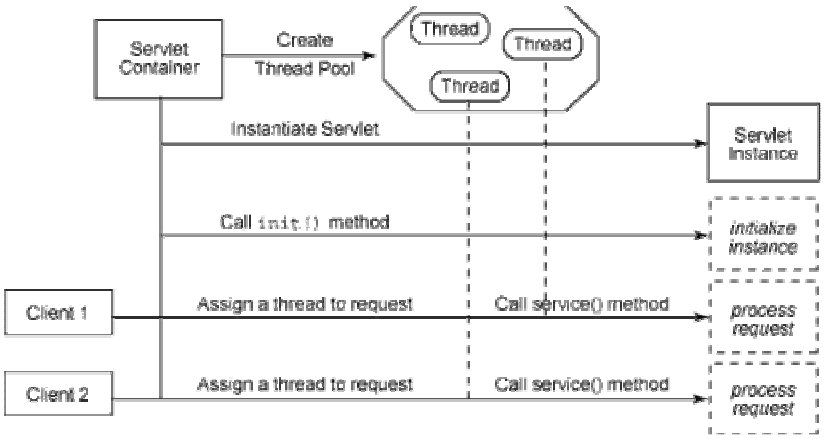

Figura 7.1 – O Modelo Servlet de linhas de execução (threads)(GABHART, 2003)

Cada escopo escolhido também possui implicações na escalabilidade e performance. Selecionar o escopo para um componente depende muito da forma como será utilizado.

Componentes com escopo de página (apenas para JSPs) ou requisição são os que menos causam problemas, pois os dados possuem um tempo de vida limitado.

O Escopo de aplicação é gerenciável se usado apenas para leitura de valores entre os Servlets de uma aplicação.

O estado no escopo de sessão é o que possui o maior impacto na escalabilidade e performance de uma aplicação Web. Para cada usuário conectado é acumulado o componente na sessão, de forma diferente ao escopo de aplicação que compartilha componentes com todos os usuários e Servlets. Também é importante lembrar que o escopo de sessão existe através das requisições, ao contrário do escopo de requisição, que descarta componentes quando a resposta é fornecida. (JONHSON, STEARNS, SINGH et. al, 2002)

Escalabilidade do tamanho do projeto: Esse aspecto está relacionado com a flexibilidade do framework em organizar o código. Struts introduz o suporte a sub-aplicações, o que permite o particionamento de uma aplicação em módulos independentes.

Desempenho: o Struts confia na introspecção Java Bean. Um dos aspectos relativo a desempenho que geralmente se comenta é a sobrecarga que o Struts ocasiona. Primeiramente deve estar claro que o que consome tempo na introspecção Java é a procura pelos métodos, e não a invocação deles. O Struts guarda essas informações, e elimina essa procura depois da inicialização. (HUSLY e YU, 2002)

8. Conclusão

O desenvolvimento de software para a Internet, desde o seu início, seguiu os mesmos caminhos que o desenvolvimento do software aplicativo percorreu há trinta anos atrás, cometendo os mesmos erros, gerando aplicações complexas e difíceis de manter. Infelizmente, vários dos preceitos pregados pela Engenharia de Software não foram incorporados ao desenvolvimento de aplicações Web.

É possível observar que desenvolver software para a Internet é muito mais do que se preocupar apenas com a apresentação da interface com o usuário. A interface com bancos de dados e o processamento da lógica de negócios precisam ser bem delimitados para permitir uma maior manutenibilidade e reuso de código das aplicações.

Com esse trabalho esperamos ter reafirmado a importância da existência de frameworks que auxiliem o desenvolvimento de aplicações para a web. Observamos a necessidade de efetuar o desacoplamento entre suas camadas para prover ganho no potencial de produtividade de desenvolvimento.

Uma das maiores vantagens de se usar um framework de desenvolvimento é a habilidade de estender e customizar as características de acordo com a necessidade da aplicação.

O framework Struts vem se firmando como um padrão no seu segmento, influenciado principalmente pela comunidade mundial de colaboradores que conferem estabilidade e robustez inestimáveis. A capacidade de ser utilizado em conjunto com várias tecnologias também pode ser citada como um fator importante para a sua popularidade.

Para conseguir utilizar toda a infra-estrutura oferecida pelo framework, é necessário que o desenvolvedor não conheça apenas o funcionamento e a forma de configuração do Struts, mas que conheça também Servlets e JSP. O Struts se torna uma opção para desenvolvimento bastante completa pois oferece recursos específicos para o desenvolvimento de aplicações Web, através de plug-ins como o Validator e o Tiles.

Durante a fase de desenvolvimento é interessante ainda a utilização de tecnologias auxiliares como Log4J e Ant.

Para um bom projeto, é necessário discutir as melhores práticas para as aplicações Struts, bem como é interessante aplicar padrões de projeto para assegurar a reusabilidade de componentes.

Para o desenvolvimento da aplicação o framework mostrou-se bastante flexível, entretanto a maleabilidade apresentada ainda permite que sejam cometidos os mesmos

erros que originam aplicações difíceis de manter, pois apesar de existir uma separação entre apresentação e lógica de negócios, a delimitação do modelo e da lógica pode não ser efetuada. Assim o framework define regras, mas não as impõe, o que pode ser necessário em algumas aplicações que lidam com problemas reais. No desenvolvimento da aplicação pudemos destacar também a confiabilidade e robustez fornecidas pelo framework.

Durante o período de implementação da aplicação, observamos a importância de se possuir um protótipo bem definido, que sirva de base para o desenvolvimento, de forma que estenda os componentes do Struts para prover as funcionalidades desejadas com a maior reutilização de código possível.

Entretanto devemos a cada projeto avaliar a real necessidade da implantação de frameworks de desenvolvimento como o Struts, uma vez que estes requerem conhecimento de padrões e consequentemente adiciona complexidade à aplicação. Em projetos de pouca complexidade que sejam desenvolvidos e mantidos por apenas um indivíduo, não se justificaria o investimento efetuado para o aprendizado do framework. Mesmo nestes casos, seria interessante aplicar padrões como o MVC mesmo que sem o auxílio do framework. Já para projetos de sistemas mais complexos, não resta dúvida que o framework Struts contribui consideravelmente para a melhoria da qualidade do software, auxiliando na divisão de tarefas entre desenvolvedores e designers, facilitando a evolução do sistema e aumentando a manutenibilidade.

Referências Bibliográficas

GOODWILL, James. Mastering Jakarta Struts. Wiley Publish Inc. 2002. Indianapolis, Indiana. USA.

CAVANESS, Chuck. Programing Jakarta Struts. O´Reilly & Associates Inc. 2002. Sebastopol, CA. USA.

HURBAIN. Isabelle. Jakarta Struts Beginner's Tutorial. Free Software Foundation.2002.

HUSTED Ted, DUMOULIN Cedric, FRANCISCUS George et al. Struts in Action. Manning Publications Co.2003. USA.

LEFF, Avraham, RAYFIELD, James T.. Web Application Development Using the Model/View/Controller Design Pattern. Fifth International Enterprise Distributed Object Computing Conference, 2001.

SHENOY Srikanth, MALLYA Nithin, Struts Survival Guide - Basics to Best Practices. Struts in Action. Object Source, 2004. USA.

Jakarta Struts Framework Disponível em: <http://struts.apache.org>. Acesso em 2 de dezembro de 2004.

Tomcat Project. Disponível em: <http://jakarta.apache.org/tomcat/index.html>. Acesso em 2 de dezembro de 2004.

BELLOQUIM, Átila. 2004. Desenvolvimento Tradicional x Desenvolvimento para a Internet. . Disponível em: <http://www.b4unews.com.br/conteudo.php?ct=noticias&idnoticia=426>. Acesso em 2 de novembro de 2004.

HUSLY, Harry, YU, Jonh. Issues in Struts Adoption. 2002. Disponível em: <http://www.scioworks.net/devnews/articles/struts_adoption_issues/>. Acesso em 10 de outubro de 2004.

JONHSON Mark, STEARNS Beth, SINGH Inderjeet et al. Desingin Enterprise Applications with the J2EE Platform. Addison-Wesley.2002 California USA.

DAVIS, Malcom. Struts, an open source MVC implementation.2001. Disponível em: <http://www-106.ibm.com/developerworks/java/library/j-struts/>. Acesso em 10

de outubro de 2004.

Eclipse project Faq. 2002. Disponível em: <http://www.eclipse.org/eclipse/faq/eclipse-faq.html>. Acesso em 10 de outubro de 2004.

Introduction to ASP. NET. Disponível em: <http://www.aspdotnetheaven.com/Tutorials/Introduction.asp>. Acesso em 10 de outubro de 2004.

Manual Php. 2004. Disponível em: <http://www.php.net/manual/pt_BR/intro-whatcando.php>. Acesso em 10 de outubro de 2004.

GABHART, Kyle. J2EE technologies for the stateless network. 2003. Disponível em: <http://www-106.ibm.com/developerworks/java/library/j-pj2ee1/>. Acesso em 2 de dezembro de 2004.

LARMAN, Craig. Utilizando UML e Padrões – Uma introdução à análise e ao projeto orientados a Objeto. Bookman. 2000. Porto Alegre.

Anexos

Cliente.java

```
package br.com.nota_fluxo.bean;

public class Cliente {
	private String cd_cli;
	private String nm_cli;
	private String razao_social;
	private String endereco;
	private String bairro;
	private String cidade;
	private String uf;
	private String cep;
	private String cpf;
	private String cnpj;
	private String insc_estadual;
	private String telefone;
	private String fax;
	private String email;
	private String nm_contato;
	private String cd_atividade;
	private String oid;
	/**
	 * @return
	 */
	public String getBairro() {
		return bairro;
	}

	/**
	 * @return
	 */
	public String getCd_atividade() {
		return cd_atividade;
	}

	/**
	 * @return
	 */
	public String getCd_cli() {
		return cd_cli;
	}

	/**
	 * @return
	 */
	public String getCep() {
		return cep;
	}

	/**
	 * @return
	 */
	public String getCidade() {
		return cidade;
	}

	/**
	 * @return
	 */
	public String getCnpj() {
		return cnpj;
	}

	/**
	 * @return
	 */
	public String getCpf() {
		return cpf;
	}

	/**
	 * @return
	 */
	public String getEmail() {
		return email;
	}

	/**
	 * @return
	 */
	public String getEndereco() {
```

```
        return endereco;
}

/**
 * @return
 */
public String getFax() {
        return fax;
}

/**
 * @return
 */
public String getInsc_estadual() {
        return insc_estadual;
}

/**
 * @return
 */
public String getNm_cli() {
        return nm_cli;
}

/**
 * @return
 */
public String getNm_contato() {
        return nm_contato;
}

/**
 * @return
 */
public String getRazao_social() {
        return razao_social;
}

/**
 * @return
 */
public String getTelefone() {
        return telefone;
}

/**
 * @return
 */
public String getUf() {
        return uf;
}

/**
 * @param string
 */
public void setBairro(String string) {
        bairro = string;
}

/**
 * @param string
 */
public void setCd_atividade(String string) {
        cd_atividade = string;
}

/**
 * @param string
 */
public void setCd_cli(String string) {
        cd_cli = string;
}

/**
 * @param string
 */
public void setCep(String string) {
        cep = string;
}

/**
 * @param string
 */
public void setCidade(String string) {
        cidade = string;
}

/**
```

```
     * @param string
     */
    public void setCnpj(String string) {
            cnpj = string;
    }

    /**
     * @param string
     */
    public void setCpf(String string) {
            cpf = string;
    }

    /**
     * @param string
     */
    public void setEmail(String string) {
            email = string;
    }

    /**
     * @param string
     */
    public void setEndereco(String string) {
            endereco = string;
    }

    /**
     * @param string
     */
    public void setFax(String string) {
            fax = string;
    }

    /**
     * @param string
     */
    public void setInsc_estadual(String string) {
            insc_estadual = string;
    }

    /**
     * @param string
     */
    public void setNm_cli(String string) {
            nm_cli = string;
    }

    /**
     * @param string
     */
    public void setNm_contato(String string) {
            nm_contato = string;
    }

    /**
     * @param string
     */
    public void setRazao_social(String string) {
            razao_social = string;
    }

    /**
     * @param string
     */
    public void setTelefone(String string) {
            telefone = string;
    }

    /**
     * @param string
     */
    public void setUf(String string) {
            uf = string;
    }

    /**
     * @return
     */
    public String getOid() {
            return oid;
    }

    /**
     * @param string
     */
    public void setOid(String string) {
            oid = string;
```

```
        }

}
```

Cnae.java

```
package br.com.nota_fluxo.bean;

public class Cnae {
        private String cd_cnae;
        private String ds_cnae;
        private String oid;
        /**
         * @return
         */
        public String getCd_cnae() {
                return cd_cnae;
        }

        /**
         * @return
         */
        public String getDs_cnae() {
                return ds_cnae;
        }

        /**
         * @param string
         */
        public void setCd_cnae(String string) {
                cd_cnae = string;
        }

        /**
         * @param string
         */
        public void setDs_cnae(String string) {
                ds_cnae = string;
        }

        /**
         * @return
         */
        public String getOid() {
                return oid;
        }

        /**
         * @param string
         */
        public void setOid(String string) {
                oid = string;
        }

}
```

Contas_pagar.java

```
package br.com.nota_fluxo.bean;

public class Contas_pagar {
        private String cd_emp;
        private String cd_pagar;
        private String dt_vencto;
        private String dt_pagto;
        private String vl_acrescimos;
        private String vl_descontos;
        private String vl_pagar;
        private String vl_pago;
        private String cd_tp_pagar;
        private String cd_conta;
        private String oid;
        /**
         * @return
         */
        public String getCd_conta() {
                return cd_conta;
        }

        /**
         * @return
         */
        public String getCd_emp() {
                return cd_emp;
        }

        /**
         * @return
```

```
 */
public String getCd_pagar() {
        return cd_pagar;
}

/**
 * @return
 */
public String getCd_tp_pagar() {
        return cd_tp_pagar;
}

/**
 * @return
 */
public String getDt_pagto() {
        return dt_pagto;
}

/**
 * @return
 */
public String getDt_vencto() {
        return dt_vencto;
}

/**
 * @return
 */
public String getVl_acrescimos() {
        return vl_acrescimos;
}

/**
 * @return
 */
public String getVl_descontos() {
        return vl_descontos;
}

/**
 * @return
 */
public String getVl_pagar() {
        return vl_pagar;
}

/**
 * @return
 */
public String getVl_pago() {
        return vl_pago;
}

/**
 * @param string
 */
public void setCd_conta(String string) {
        cd_conta = string;
}

/**
 * @param string
 */
public void setCd_emp(String string) {
        cd_emp = string;
}

/**
 * @param string
 */
public void setCd_pagar(String string) {
        cd_pagar = string;
}

/**
 * @param string
 */
public void setCd_tp_pagar(String string) {
        cd_tp_pagar = string;
}

/**
 * @param string
 */
public void setDt_pagto(String string) {
        dt_pagto = string;
}
```

```
        /**
         * @param string
         */
        public void setDt_vencto(String string) {
                dt_vencto = string;
        }

        /**
         * @param string
         */
        public void setVl_acrescimos(String string) {
                vl_acrescimos = string;
        }

        /**
         * @param string
         */
        public void setVl_descontos(String string) {
                vl_descontos = string;
        }

        /**
         * @param string
         */
        public void setVl_pagar(String string) {
                vl_pagar = string;
        }

        /**
         * @param string
         */
        public void setVl_pago(String string) {
                vl_pago = string;
        }

        /**
         * @return
         */
        public String getOid() {
                return oid;
        }

        /**
         * @param string
         */
        public void setOid(String string) {
                oid = string;
        }

}
```

Contas_receber.java

```
package br.com.nota_fluxo.bean;

public class Contas_receber {
        private String cd_emp;
        private String cd_receber;
        private String dt_vencto;
        private String dt_pagto;
        private String vl_receber;
        private String vl_recebido;
        private String nr_nota;
        private String cd_tp_receber;
        private String cd_conta;
        private String oid;
        /**
         * @return
         */
        public String getCd_conta() {
                return cd_conta;
        }

        /**
         * @return
         */
        public String getCd_emp() {
                return cd_emp;
        }

        /**
         * @return
         */
        public String getCd_receber() {
                return cd_receber;
        }
```

```
/**
 * @return
 */
public String getCd_tp_receber() {
        return cd_tp_receber;
}

/**
 * @return
 */
public String getDt_pagto() {
        return dt_pagto;
}

/**
 * @return
 */
public String getDt_vencto() {
        return dt_vencto;
}

/**
 * @return
 */
public String getNr_nota() {
        return nr_nota;
}

/**
 * @return
 */
public String getVl_receber() {
        return vl_receber;
}

/**
 * @return
 */
public String getVl_recebido() {
        return vl_recebido;
}

/**
 * @param string
 */
public void setCd_conta(String string) {
        cd_conta = string;
}

/**
 * @param string
 */
public void setCd_emp(String string) {
        cd_emp = string;
}

/**
 * @param string
 */
public void setCd_receber(String string) {
        cd_receber = string;
}

/**
 * @param string
 */
public void setCd_tp_receber(String string) {
        cd_tp_receber = string;
}

/**
 * @param string
 */
public void setDt_pagto(String string) {
        dt_pagto = string;
}

/**
 * @param string
 */
public void setDt_vencto(String string) {
        dt_vencto = string;
}

/**
 * @param string
 */
public void setNr_nota(String string) {
```

```
                nr_nota = string;
        }

        /**
         * @param string
         */
        public void setVl_receber(String string) {
                vl_receber = string;
        }

        /**
         * @param string
         */
        public void setVl_recebido(String string) {
                vl_recebido = string;
        }

        /**
         * @return
         */
        public String getOid() {
                return oid;
        }

        /**
         * @param string
         */
        public void setOid(String string) {
                oid = string;
        }

}
```

Contas.java

```
package br.com.nota_fluxo.bean;

public class Contas {
        private String cd_emp;
        private String cd_conta;
        private String ds_conta;
        private String oid;
        /**
         * @return
         */
        public String getCd_conta() {
                return cd_conta;
        }

        /**
         * @return
         */
        public String getCd_emp() {
                return cd_emp;
        }

        /**
         * @return
         */
        public String getDs_conta() {
                return ds_conta;
        }

        /**
         * @param string
         */
        public void setCd_conta(String string) {
                cd_conta = string;
        }

        /**
         * @param string
         */
        public void setCd_emp(String string) {
                cd_emp = string;
        }

        /**
         * @param string
         */
        public void setDs_conta(String string) {
                ds_conta = string;
        }

        /**
         * @return
         */
```

```
        public String getOid() {
                return oid;
        }

        /**
         * @param string
         */
        public void setOid(String string) {
                oid = string;
        }

}
```

Empresa.java

```
package br.com.nota_fluxo.bean;

public class Empresa {
        private String cd_emp;
        private String nm_emp;
        private String razao_social;
        private String cnpj;
        private String insc_estadual;
        private String endereco;
        private String bairro;
        private String cidade;
        private String uf;
        private String cep;
        private String fax;
        private String telefone;
        private String email;
        private String cd_atividade;
        private String nm_contato;
        private String oid;

        public String getBairro() {
                return bairro;
        }
        public String getCd_atividade() {
                return cd_atividade;
        }
        public String getCd_emp() {
                return cd_emp;
        }
        public String getCep() {
                return cep;
        }
        public String getCidade() {
                return cidade;
        }
        public String getCnpj() {
                return cnpj;
        }
        public String getEmail() {
                return email;
        }
        public String getEndereco() {
                return endereco;
        }
        public String getFax() {
                return fax;
        }
        public String getInsc_estadual() {
                return insc_estadual;
        }
        public String getNm_contato() {
                return nm_contato;
        }
        public String getNm_emp() {
                return nm_emp;
        }
        public String getRazao_social() {
                return razao_social;
        }
        public String getTelefone() {
                return telefone;
        }
        public String getUf() {
                return uf;
        }
        public void setBairro(String string) {
                bairro = string;
        }
        public void setCd_atividade(String string) {
                cd_atividade = string;
        }
        public void setCd_emp(String string) {
                cd_emp = string;
```

```
        }
        public void setCep(String string) {
                cep = string;
        }
        public void setCidade(String string) {
                cidade = string;
        }
        public void setCnpj(String string) {
                cnpj = string;
        }
        public void setEmail(String string) {
                email = string;
        }
        public void setEndereco(String string) {
                endereco = string;
        }
        public void setFax(String string) {
                fax = string;
        }
        public void setInsc_estadual(String string) {
                insc_estadual = string;
        }
        public void setNm_contato(String string) {
                nm_contato = string;
        }
        public void setNm_emp(String string) {
                nm_emp = string;
        }
        public void setRazao_social(String string) {
                razao_social = string;
        }
        public void setTelefone(String string) {
                telefone = string;
        }
        public void setUf(String string) {
                uf = string;
        }
        public String getOid() {
                return oid;
        }
        public void setOid(String string) {
                oid = string;
        }
}
```

Movimento_contas.java

```
package br.com.nota_fluxo.bean;

public class Movimento_contas {
        private String dt_movto;
        private String nr_movto;
        private String cd_emp;
        private String nr_conta;
        private String ds_movto;
        private String vl_movto;
        private String oid;
        /**
         * @return
         */
        public String getCd_emp() {
                return cd_emp;
        }

        /**
         * @return
         */
        public String getDs_movto() {
                return ds_movto;
        }

        /**
         * @return
         */
        public String getNr_conta() {
                return nr_conta;
        }

        /**
         * @return
         */
        public String getNr_movto() {
                return nr_movto;
        }

        /**
         * @return
         */
```

```
        public String getVl_movto() {
                return vl_movto;
        }

        /**
         * @param string
         */
        public void setCd_emp(String string) {
                cd_emp = string;
        }

        /**
         * @param string
         */
        public void setDs_movto(String string) {
                ds_movto = string;
        }

        /**
         * @param string
         */
        public void setNr_conta(String string) {
                nr_conta = string;
        }

        /**
         * @param string
         */
        public void setNr_movto(String string) {
                nr_movto = string;
        }

        /**
         * @param string
         */
        public void setVl_movto(String string) {
                vl_movto = string;
        }

        /**
         * @return
         */
        public String getOid() {
                return oid;
        }

        /**
         * @param string
         */
        public void setOid(String string) {
                oid = string;
        }

        /**
         * @return
         */
        public String getDt_movto() {
                return dt_movto;
        }

        /**
         * @param string
         */
        public void setDt_movto(String string) {
                dt_movto = string;
        }

}
```

Nota_fiscal.java

```
package br.com.nota_fluxo.bean;

public class Nota_fiscal {
        private String cd_emp;
        private String nr_nota;
        private String dt_ref;
        private String cd_cli;
        private String vl_bruto;
        private String dt_vencto;
        private String vl_liquido;
        private String irff_retido;
        private String irff_pagar;
        private String cofins_retido;
        private String pis_retido;
        private String csll_retido;
        private String iss_retido;
        private String iss_pagar;
```

```
private String total_impostos_retidos;
private String cst;
private String irlp;
private String cfps;
private String contribuicao_social;
private String cd_conta;
private String oid;
/**
 * @return
 */
public String getCd_cli() {
        return cd_cli;
}

/**
 * @return
 */
public String getCd_emp() {
        return cd_emp;
}

/**
 * @return
 */
public String getCfps() {
        return cfps;
}

/**
 * @return
 */
public String getCofins_retido() {
        return cofins_retido;
}

/**
 * @return
 */
public String getContribuicao_social() {
        return contribuicao_social;
}

/**
 * @return
 */
public String getCsll_retido() {
        return csll_retido;
}

/**
 * @return
 */
public String getCst() {
        return cst;
}

/**
 * @return
 */
public String getDt_ref() {
        return dt_ref;
}

/**
 * @return
 */
public String getDt_vencto() {
        return dt_vencto;
}

/**
 * @return
 */
public String getIrff_pagar() {
        return irff_pagar;
}

/**
 * @return
 */
public String getIrff_retido() {
        return irff_retido;
}

/**
 * @return
 */
public String getIrlp() {
```

```
        return irlp;
}

/**
 * @return
 */
public String getIss_pagar() {
        return iss_pagar;
}

/**
 * @return
 */
public String getIss_retido() {
        return iss_retido;
}

/**
 * @return
 */
public String getNr_nota() {
        return nr_nota;
}

/**
 * @return
 */
public String getPis_retido() {
        return pis_retido;
}

/**
 * @return
 */
public String getTotal_impostos_retidos() {
        return total_impostos_retidos;
}

/**
 * @return
 */
public String getVl_bruto() {
        return vl_bruto;
}

/**
 * @return
 */
public String getVl_liquido() {
        return vl_liquido;
}

/**
 * @param string
 */
public void setCd_cli(String string) {
        cd_cli = string;
}

/**
 * @param string
 */
public void setCd_emp(String string) {
        cd_emp = string;
}

/**
 * @param string
 */
public void setCfps(String string) {
        cfps = string;
}

/**
 * @param string
 */
public void setCofins_retido(String string) {
        cofins_retido = string;
}

/**
 * @param string
 */
public void setContribuicao_social(String string) {
        contribuicao_social = string;
}

/**
```

```
     * @param string
     */
    public void setCsll_retido(String string) {
            csll_retido = string;
    }

    /**
     * @param string
     */
    public void setCst(String string) {
            cst = string;
    }

    /**
     * @param string
     */
    public void setDt_ref(String string) {
            dt_ref = string;
    }

    /**
     * @param string
     */
    public void setDt_vencto(String string) {
            dt_vencto = string;
    }

    /**
     * @param string
     */
    public void setIrff_pagar(String string) {
            irff_pagar = string;
    }

    /**
     * @param string
     */
    public void setIrff_retido(String string) {
            irff_retido = string;
    }

    /**
     * @param string
     */
    public void setIrlp(String string) {
            irlp = string;
    }

    /**
     * @param string
     */
    public void setIss_pagar(String string) {
            iss_pagar = string;
    }

    /**
     * @param string
     */
    public void setIss_retido(String string) {
            iss_retido = string;
    }

    /**
     * @param string
     */
    public void setNr_nota(String string) {
            nr_nota = string;
    }

    /**
     * @param string
     */
    public void setPis_retido(String string) {
            pis_retido = string;
    }

    /**
     * @param string
     */
    public void setTotal_impostos_retidos(String string) {
            total_impostos_retidos = string;
    }

    /**
     * @param string
     */
    public void setVl_bruto(String string) {
            vl_bruto = string;
```

```
        }

        /**
         * @param string
         */
        public void setVl_liquido(String string) {
                vl_liquido = string;
        }

        /**
         * @return
         */
        public String getOid() {
                return oid;
        }

        /**
         * @param string
         */
        public void setOid(String string) {
                oid = string;
        }

        /**
         * @return
         */
        public String getCd_conta() {
                return cd_conta;
        }

        /**
         * @param string
         */
        public void setCd_conta(String string) {
                cd_conta = string;
        }

}
```

Nota_item.java

```
package br.com.nota_fluxo.bean;
public class Nota_item {
        private String cd_emp;
        private String nr_nota;
        private String cd_cnae;
        private String vl_item;
        private String oid;
        /**
         * @return
         */
        public String getCd_cnae() {
                return cd_cnae;
        }

        /**
         * @return
         */
        public String getNr_nota() {
                return nr_nota;
        }

        /**
         * @return
         */
        public String getVl_item() {
                return vl_item;
        }

        /**
         * @param string
         */
        public void setCd_cnae(String string) {
                cd_cnae = string;
        }

        /**
         * @param string
         */
        public void setNr_nota(String string) {
                nr_nota = string;
        }

        /**
         * @param string
         */
        public void setVl_item(String string) {
                vl_item = string;
```

```
        }

        /**
         * @return
         */
        public String getOid() {
                return oid;
        }

        /**
         * @param string
         */
        public void setOid(String string) {
                oid = string;
        }

        /**
         * @return
         */
        public String getCd_emp() {
                return cd_emp;
        }

        /**
         * @param string
         */
        public void setCd_emp(String string) {
                cd_emp = string;
        }

}
```

Ramo_Atividade.java

```
package br.com.nota_fluxo.bean;
public class Ramo_atividade {
        private String cd_atividade;
        private String ds_atividade;
        private String oid;
        /**
         * @return
         */
        public String getCd_atividade() {
                return cd_atividade;
        }

        /**
         * @return
         */
        public String getDs_atividade() {
                return ds_atividade;
        }

        /**
         * @param string
         */
        public void setCd_atividade(String string) {
                cd_atividade = string;
        }

        /**
         * @param string
         */
        public void setDs_atividade(String string) {
                ds_atividade = string;
        }

        /**
         * @return
         */
        public String getOid() {
                return oid;
        }

        /**
         * @param string
         */
        public void setOid(String string) {
                oid = string;
        }

}
```

Tipo_contas_pagar.java

```
package br.com.nota_fluxo.bean;
public class Tipo_contas_pagar {
        private String cd_tp_pagar;
```

```
        private String ds_tp_pagar;
        private String oid;
        /**
         * @return
         */
        public String getCd_tp_pagar() {
                return cd_tp_pagar;
        }

        /**
         * @return
         */
        public String getDs_tp_pagar() {
                return ds_tp_pagar;
        }

        /**
         * @param string
         */
        public void setCd_tp_pagar(String string) {
                cd_tp_pagar = string;
        }

        /**
         * @param string
         */
        public void setDs_tp_pagar(String string) {
                ds_tp_pagar = string;
        }

        /**
         * @return
         */
        public String getOid() {
                return oid;
        }

        /**
         * @param string
         */
        public void setOid(String string) {
                oid = string;
        }

}
```

Tipo_contas_receber.java

```
package br.com.nota_fluxo.bean;
public class Tipo_contas_receber {
        private String cd_tp_receber;
        private String ds_tp_receber;
        private String oid;
        /**
         * @return
         */
        public String getCd_tp_receber() {
                return cd_tp_receber;
        }

        /**
         * @return
         */
        public String getDs_tp_receber() {
                return ds_tp_receber;
        }

        /**
         * @param string
         */
        public void setCd_tp_receber(String string) {
                cd_tp_receber = string;
        }

        /**
         * @param string
         */
        public void setDs_tp_receber(String string) {
                ds_tp_receber = string;
        }

        /**
         * @return
         */
        public String getOid() {
                return oid;
        }
```

```
        /**
         * @param string
         */
        public void setOid(String string) {
                oid = string;
        }
}
```

Usuario.java

```
package br.com.nota_fluxo.bean;

public class Usuario {
        private String nm_usuario;
        private String ds_senha;
        private String cd_emp;
        private String oid;

        public String getDs_senha() {
                return ds_senha;
        }
        public String getNm_usuario() {
                return nm_usuario;
        }
        public void setDs_senha(String string) {
                ds_senha = string;
        }
        public void setNm_usuario(String string) {
                nm_usuario = string;
        }
        public String getCd_emp() {
                return cd_emp;
        }
        public void setCd_emp(String string) {
                cd_emp = string;
        }
        public String getOid() {
                return oid;
        }
        public void setOid(String string) {
                oid = string;
        }

}
```

ClienteForm.java

```
package br.com.nota_fluxo.form;

import java.io.Serializable;
import java.util.ArrayList;

import br.com.nota_fluxo.bean.Cliente;

public class ClienteForm extends PadraoForm implements Serializable {
        ArrayList cliente_lista = new ArrayList();
        Cliente cliente = new Cliente();
        Cliente cliente_pesq = new Cliente();

        public Cliente getCliente() {
                return cliente;
        }
        public ArrayList getCliente_lista() {
                return cliente_lista;
        }
        public Cliente getCliente_pesq() {
                return cliente_pesq;
        }
        public void setCliente(Cliente cliente) {
                this.cliente = cliente;
        }
        public void setCliente_lista(ArrayList list) {
                cliente_lista = list;
        }
        public void setCliente_pesq(Cliente cliente) {
                cliente_pesq = cliente;
        }

}
```

CnaeForm.java

```
package br.com.nota_fluxo.form;

import java.io.Serializable;
import java.util.ArrayList;

import br.com.nota_fluxo.bean.Cnae;
```

```
public class CnaeForm extends PadraoForm implements Serializable {
        ArrayList cnae_lista = new ArrayList();
        Cnae cnae = new Cnae();
        Cnae cnae_pesq = new Cnae();

        public Cnae getCnae() {
                return cnae;
        }
        public ArrayList getCnae_lista() {
                return cnae_lista;
        }
        public Cnae getCnae_pesq() {
                return cnae_pesq;
        }
        public void setCnae(Cnae cnae) {
                this.cnae = cnae;
        }
        public void setCnae_lista(ArrayList list) {
                cnae_lista = list;
        }
        public void setCnae_pesq(Cnae cnae) {
                cnae_pesq = cnae;
        }

}
```

Contas_pagarForm.java

```
package br.com.nota_fluxo.form;

import java.io.Serializable;
import java.util.ArrayList;

import br.com.nota_fluxo.bean.Contas_pagar;

public class Contas_pagarForm extends PadraoForm implements Serializable {
        ArrayList contas_pagar_lista = new ArrayList();
        Contas_pagar contas_pagar = new Contas_pagar();
        Contas_pagar contas_pagar_pesq = new Contas_pagar();

        public Contas_pagar getContas_pagar() {
                return contas_pagar;
        }
        public ArrayList getContas_pagar_lista() {
                return contas_pagar_lista;
        }
        public Contas_pagar getContas_pagar_pesq() {
                return contas_pagar_pesq;
        }
        public void setContas_pagar(Contas_pagar contas_pagar) {
                this.contas_pagar = contas_pagar;
        }
        public void setContas_pagar_lista(ArrayList list) {
                contas_pagar_lista = list;
        }
        public void setContas_pagar_pesq(Contas_pagar contas_pagar) {
                contas_pagar_pesq = contas_pagar;
        }

}
```

Contas_receberForm.java

```
package br.com.nota_fluxo.form;

import java.io.Serializable;
import java.util.ArrayList;

import br.com.nota_fluxo.bean.Contas_receber;

public class Contas_receberForm extends PadraoForm implements Serializable {
        ArrayList contas_receber_lista = new ArrayList();
        Contas_receber contas_receber = new Contas_receber();
        Contas_receber contas_receber_pesq = new Contas_receber();

        public Contas_receber getContas_receber() {
                return contas_receber;
        }
        public ArrayList getContas_receber_lista() {
                return contas_receber_lista;
        }
        public Contas_receber getContas_receber_pesq() {
                return contas_receber_pesq;
        }
        public void setContas_receber(Contas_receber contas_receber) {
                this.contas_receber = contas_receber;
        }
```

```
        public void setContas_receber_lista(ArrayList list) {
                contas_receber_lista = list;
        }
        public void setContas_receber_pesq(Contas_receber contas_receber) {
                contas_receber_pesq = contas_receber;
        }

}
```

ContasForm.java

```
package br.com.nota_fluxo.form;

import java.io.Serializable;
import java.util.ArrayList;

import br.com.nota_fluxo.bean.Contas;

public class ContasForm extends PadraoForm implements Serializable {
        ArrayList contas_lista = new ArrayList();
        Contas contas = new Contas();
        Contas contas_pesq = new Contas();

        public Contas getContas() {
                return contas;
        }
        public ArrayList getContas_lista() {
                return contas_lista;
        }
        public Contas getContas_pesq() {
                return contas_pesq;
        }
        public void setContas(Contas contas) {
                this.contas = contas;
        }
        public void setContas_lista(ArrayList list) {
                contas_lista = list;
        }
        public void setContas_pesq(Contas contas) {
                contas_pesq = contas;
        }

}
```

EmpresaForm.java

```
package br.com.nota_fluxo.form;

import java.io.Serializable;
import java.util.ArrayList;

import br.com.nota_fluxo.bean.Empresa;

public class EmpresaForm extends PadraoForm implements Serializable {
        ArrayList empresa_lista = new ArrayList();
        Empresa empresa = new Empresa();
        Empresa empresa_pesq = new Empresa();

        public Empresa getEmpresa() {
                return empresa;
        }
        public ArrayList getEmpresa_lista() {
                return empresa_lista;
        }
        public Empresa getEmpresa_pesq() {
                return empresa_pesq;
        }
        public void setEmpresa(Empresa empresa) {
                this.empresa = empresa;
        }
        public void setEmpresa_lista(ArrayList list) {
                empresa_lista = list;
        }
        public void setEmpresa_pesq(Empresa empresa) {
                empresa_pesq = empresa;
        }
}
```

Movimento_contasForm.java

```
package br.com.nota_fluxo.form;

import java.io.Serializable;
import java.util.ArrayList;

import br.com.nota_fluxo.bean.Movimento_contas;

public class Movimento_contasForm extends PadraoForm implements Serializable {
```

```
        ArrayList movimento_contas_lista = new ArrayList();
        Movimento_contas movimento_contas = new Movimento_contas();
        Movimento_contas movimento_contas_pesq = new Movimento_contas();

        public Movimento_contas getMovimento_contas() {
                return movimento_contas;
        }
        public ArrayList getMovimento_contas_lista() {
                return movimento_contas_lista;
        }
        public Movimento_contas getMovimento_contas_pesq() {
                return movimento_contas_pesq;
        }
        public void setMovimento_contas(Movimento_contas movimento_contas) {
                this.movimento_contas = movimento_contas;
        }
        public void setMovimento_contas_lista(ArrayList list) {
                movimento_contas_lista = list;
        }
        public void setMovimento_contas_pesq(Movimento_contas movimento_contas) {
                movimento_contas_pesq = movimento_contas;
        }

}
```

Nota_fiscalForm.java

```
package br.com.nota_fluxo.form;

import java.io.Serializable;
import java.util.ArrayList;

import br.com.nota_fluxo.bean.Nota_fiscal;

public class Nota_fiscalForm extends PadraoForm implements Serializable {
        ArrayList nota_fiscal_lista = new ArrayList();
        Nota_fiscal nota_fiscal = new Nota_fiscal();
        Nota_fiscal nota_fiscal_pesq = new Nota_fiscal();

        public Nota_fiscal getNota_fiscal() {
                return nota_fiscal;
        }
        public ArrayList getNota_fiscal_lista() {
                return nota_fiscal_lista;
        }
        public Nota_fiscal getNota_fiscal_pesq() {
                return nota_fiscal_pesq;
        }
        public void setNota_fiscal(Nota_fiscal nota_fiscal) {
                this.nota_fiscal = nota_fiscal;
        }
        public void setNota_fiscal_lista(ArrayList list) {
                nota_fiscal_lista = list;
        }
        public void setNota_fiscal_pesq(Nota_fiscal nota_fiscal) {
                nota_fiscal_pesq = nota_fiscal;
        }

}
```

Nota_itemForm.java

```
package br.com.nota_fluxo.form;

import java.io.Serializable;
import java.util.ArrayList;

import br.com.nota_fluxo.bean.Nota_item;

public class Nota_itemForm extends PadraoForm implements Serializable {
        ArrayList nota_item_lista = new ArrayList();
        Nota_item nota_item = new Nota_item();
        Nota_item nota_item_pesq = new Nota_item();

        public Nota_item getNota_item() {
                return nota_item;
        }
        public ArrayList getNota_item_lista() {
                return nota_item_lista;
        }
        public Nota_item getNota_item_pesq() {
                return nota_item_pesq;
        }
        public void setNota_item(Nota_item nota_item) {
                this.nota_item = nota_item;
        }
        public void setNota_item_lista(ArrayList list) {
                nota_item_lista = list;
```

```
	}
	public void setNota_item_pesq(Nota_item nota_item) {
		nota_item_pesq = nota_item;
	}

}
```

PadraoForm.java

```
package br.com.nota_fluxo.form;

import java.io.Serializable;

import org.apache.struts.validator.ValidatorActionForm;
import br.com.nota_fluxo.util.Util;

public class PadraoForm extends ValidatorActionForm implements Serializable {
	private int contador;
	private int contadorAtual;
	private String mostraAnteriores;
	private String mostraProximos;
	private String mostraConsulta;
	private int pagina_prox;
	private int pagina_ant;
	private int pagina_prim;
	private int pagina_ult;
	private int nr_reg = 10;

	public void setContadoresPaginas(int p_cont, int p_cont_atual){
		contador = p_cont;
		contadorAtual = p_cont_atual;
		if (contadorAtual <= nr_reg){
			mostraAnteriores = "N";
		} else {
			mostraAnteriores = "S";
		}
		if (contadorAtual < contador){
			mostraProximos = "S";
		} else {
			mostraProximos = "N";
		}
		pagina_prox = contadorAtual;
		if (contadorAtual == contador){
			if ((contadorAtual%nr_reg)==0 ){
				pagina_ant  = contadorAtual - (2*nr_reg) - (contadorAtual%nr_reg);
			} else {
				pagina_ant  = contadorAtual - nr_reg - (contadorAtual%nr_reg);
			}
		} else {
			pagina_ant  = contadorAtual - (2*nr_reg) - (contadorAtual%nr_reg);
		}
		pagina_prim = 0;
		if (contadorAtual == contador){
			if ((contador%nr_reg)==0 ){
				pagina_ult = contador - nr_reg;
			} else {
				pagina_ult = contador - (contador%nr_reg) ;
			}
		} else {
			if ((contador%nr_reg)==0 ){
				pagina_ult = contador - nr_reg;
			} else {
				pagina_ult = contador - (contador%nr_reg) ;
			}
		}
	}

	public int retornaPagina(String p_nr_atual){
		int v_nr_pag = 0;
		if (Util.trataNulo(p_nr_atual)!= null){
			v_nr_pag = Integer.parseInt(p_nr_atual);
			v_nr_pag = v_nr_pag - (v_nr_pag%nr_reg);
		}
		return v_nr_pag;
	}
	public int getContador() {
		return contador;
	}
	public int getContadorAtual() {
		return contadorAtual;
	}
	public String getMostraAnteriores() {
		return mostraAnteriores;
	}
	public String getMostraConsulta() {
		return mostraConsulta;
	}
	public String getMostraProximos() {
```

```
                return mostraProximos;
        }
        public int getNr_reg() {
                return nr_reg;
        }
        public int getPagina_ant() {
                return pagina_ant;
        }
        public int getPagina_prim() {
                return pagina_prim;
        }
        public int getPagina_prox() {
                return pagina_prox;
        }
        public int getPagina_ult() {
                return pagina_ult;
        }
        public void setContador(int i) {
                contador = i;
        }
        public void setContadorAtual(int i) {
                contadorAtual = i;
        }
        public void setMostraAnteriores(String string) {
                mostraAnteriores = string;
        }
        public void setMostraConsulta(String string) {
                mostraConsulta = string;
        }
        public void setMostraProximos(String string) {
                mostraProximos = string;
        }
        public void setNr_reg(int i) {
                nr_reg = i;
        }
        public void setPagina_ant(int i) {
                pagina_ant = i;
        }
        public void setPagina_prim(int i) {
                pagina_prim = i;
        }
        public void setPagina_prox(int i) {
                pagina_prox = i;
        }
        public void setPagina_ult(int i) {
                pagina_ult = i;
        }
}
```

Ramo_AtividadeForm.java

```
package br.com.nota_fluxo.form;

import java.io.Serializable;
import java.util.ArrayList;

import br.com.nota_fluxo.bean.Ramo_atividade;

public class Ramo_atividadeForm extends PadraoForm implements Serializable {
        ArrayList ramo_atividade_lista = new ArrayList();
        Ramo_atividade ramo_atividade = new Ramo_atividade();
        Ramo_atividade ramo_atividade_pesq = new Ramo_atividade();

        public Ramo_atividade getRamo_atividade() {
                return ramo_atividade;
        }
        public ArrayList getRamo_atividade_lista() {
                return ramo_atividade_lista;
        }
        public Ramo_atividade getRamo_atividade_pesq() {
                return ramo_atividade_pesq;
        }
        public void setRamo_atividade(Ramo_atividade ramo_atividade) {
                this.ramo_atividade = ramo_atividade;
        }
        public void setRamo_atividade_lista(ArrayList list) {
                ramo_atividade_lista = list;
        }
        public void setRamo_atividade_pesq(Ramo_atividade ramo_atividade) {
                ramo_atividade_pesq = ramo_atividade;
        }

}
```

Tipo_contas_pagarForm.java

```
package br.com.nota_fluxo.form;
```

```
import java.io.Serializable;
import java.util.ArrayList;

import br.com.nota_fluxo.bean.Tipo_contas_pagar;

public class Tipo_contas_pagarForm extends PadraoForm implements Serializable {
        ArrayList tipo_contas_pagar_lista = new ArrayList();
        Tipo_contas_pagar tipo_contas_pagar = new Tipo_contas_pagar();
        Tipo_contas_pagar tipo_contas_pagar_pesq = new Tipo_contas_pagar();

        public Tipo_contas_pagar getTipo_contas_pagar() {
                return tipo_contas_pagar;
        }
        public ArrayList getTipo_contas_pagar_lista() {
                return tipo_contas_pagar_lista;
        }
        public Tipo_contas_pagar getTipo_contas_pagar_pesq() {
                return tipo_contas_pagar_pesq;
        }
        public void setTipo_contas_pagar(Tipo_contas_pagar tipo_contas_pagar) {
                this.tipo_contas_pagar = tipo_contas_pagar;
        }
        public void setTipo_contas_pagar_lista(ArrayList list) {
                tipo_contas_pagar_lista = list;
        }
        public void setTipo_contas_pagar_pesq(Tipo_contas_pagar tipo_contas_pagar) {
                tipo_contas_pagar_pesq = tipo_contas_pagar;
        }

}
```

Tipo_contas_receberForm.java

```
package br.com.nota_fluxo.form;

import java.io.Serializable;
import java.util.ArrayList;

import br.com.nota_fluxo.bean.Tipo_contas_receber;

public class Tipo_contas_receberForm extends PadraoForm implements Serializable {
        ArrayList tipo_contas_receber_lista = new ArrayList();
        Tipo_contas_receber tipo_contas_receber = new Tipo_contas_receber();
        Tipo_contas_receber tipo_contas_receber_pesq = new Tipo_contas_receber();

        public Tipo_contas_receber getTipo_contas_receber() {
                return tipo_contas_receber;
        }
        public ArrayList getTipo_contas_receber_lista() {
                return tipo_contas_receber_lista;
        }
        public Tipo_contas_receber getTipo_contas_receber_pesq() {
                return tipo_contas_receber_pesq;
        }
        public void setTipo_contas_receber(Tipo_contas_receber tipo_contas_receber) {
                this.tipo_contas_receber = tipo_contas_receber;
        }
        public void setTipo_contas_receber_lista(ArrayList list) {
                tipo_contas_receber_lista = list;
        }
        public void setTipo_contas_receber_pesq(Tipo_contas_receber tipo_contas_receber) {
                tipo_contas_receber_pesq = tipo_contas_receber;
        }

}
```

UsuárioForm.java

```
package br.com.nota_fluxo.form;

import java.io.Serializable;
import java.util.ArrayList;

import br.com.nota_fluxo.bean.Usuario;

public class UsuarioForm extends PadraoForm implements Serializable {
        ArrayList usuario_lista = new ArrayList();
        Usuario usuario = new Usuario();
        Usuario usuario_pesq = new Usuario();

        public Usuario getUsuario() {
                return usuario;
        }
        public ArrayList getUsuario_lista() {
                return usuario_lista;
        }
        public Usuario getUsuario_pesq() {
                return usuario_pesq;
```

```
        }
        public void setUsuario(Usuario usuario) {
                this.usuario = usuario;
        }
        public void setUsuario_lista(ArrayList list) {
                usuario_lista = list;
        }
        public void setUsuario_pesq(Usuario usuario) {
                usuario_pesq = usuario;
        }
}
```

ClienteSQL.java

```
package br.com.nota_fluxo.sql;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

import org.apache.log4j.Category;

import br.com.nota_fluxo.bean.Cliente;
import br.com.nota_fluxo.util.Util;

public class ClienteSQL {
        private int contaSelect;
        private int contaSelectAtual;
        private int nr_reg;
        private String nr_atual;

        Category cat = Category.getInstance(ClienteSQL.class.getName());

        private ArrayList carregaObjetos(ResultSet rs, String p_oid)
                throws SQLException {
                ArrayList retorno = new ArrayList();
                int nr_reg = 0;
                while (rs.next()) {
                        if (Util.trataNulo(p_oid) != null) {
                                if (Integer.parseInt(p_oid) ==
Integer.parseInt(rs.getString("oid"))) {
                                        nr_atual = nr_reg + "";
                                }
                        } else {
                                Cliente cliente = new Cliente();
                                cliente.setCd_cli(rs.getString("cd_cli"));
                                cliente.setNm_cli(rs.getString("nm_cli"));
                                cliente.setRazao_social(rs.getString("razao_social"));
                                cliente.setEndereco(rs.getString("endereco"));
                                cliente.setBairro(rs.getString("bairro"));
                                cliente.setCidade(rs.getString("cidade"));
                                cliente.setUf(rs.getString("uf"));
                                cliente.setCep(rs.getString("cep"));
                                cliente.setCnpj(rs.getString("cnpj"));
                                cliente.setInsc_estadual(rs.getString("insc_estadual"));
                                cliente.setTelefone(rs.getString("telefone"));
                                cliente.setFax(rs.getString("fax"));
                                cliente.setEmail(rs.getString("email"));
                                cliente.setNm_contato(rs.getString("nm_contato"));
                                cliente.setCd_atividade(rs.getString("cd_atividade"));
                                cliente.setOid(rs.getString("oid"));
                                retorno.add(cliente);
                        }
                        nr_reg++;
                }
                return retorno;
        }

        public ArrayList select(
                Connection con,
                int p_pagina,
                Cliente obj,
                int p_nr_reg,
                String p_oid,
                String p_mostra_contadores)
                throws SQLException {
                ArrayList retorno = new ArrayList();
                String sql =
                "select "
                + "  cd_cli"
                + ", nm_cli"
                + ", razao_social"
                + ", endereco"
                + ", bairro"
                + ", cidade"
                + ", uf"
```

```
                + ", cep"
                + ", cnpj"
                + ", insc_estadual"
                + ", telefone"
                + ", fax"
                + ", email"
                + ", nm_contato"
                + ", cd_atividade "
                + ", oid "
                + "from t_cliente "
                + "where "
                + "       cd_atividade                              = coalesce(?, cd_atividade) "
                + "and coalesce(nm_contato,'-1')= coalesce(?, coalesce(nm_contato, '-1')) "
                + "and coalesce(email, '-1')      = coalesce(?, coalesce(email, '-1')) "
                + "and coalesce(fax, '-1')                = coalesce(?, coalesce(fax, '-1')) "
                + "and coalesce(telefone, '-1')   = coalesce(?, coalesce(telefone, '-1')) "
                + "and coalesce(insc_estadual,'-1')= coalesce(?, coalesce(insc_estadual, '-1')) "
                + "and coalesce(cnpj, -1)         = coalesce(?, coalesce(cnpj, -1)) "
                + "and coalesce(cep, -1)          = coalesce(?, coalesce(cep, -1)) "
                + "and uf                                                 = coalesce(?, uf) "
                + "and cidade                                     = coalesce(?, cidade) "
                + "and coalesce(bairro, '-1')     = coalesce(?, coalesce(bairro, '-1')) "
                + "and endereco                                   = coalesce(?, endereco) "
                + "and razao_social                               = coalesce(?, razao_social) "
                + "and nm_cli                                     = coalesce(?, nm_cli) ";
                if (Util.trataNulo(obj.getCd_cli()) == null) {
                        sql = sql + "and cd_cli           = coalesce(?, cd_cli)";
                } else {
                        sql = sql + "and cd_cli           = ? ";
                }

                if (Util.trataNulo(obj.getOid()) == null) {
                        sql = sql + "and oid                      = coalesce(?, oid) ";
                } else {
                        sql = sql + "and oid                      = ? ";
                }

                sql = sql + "order by 1 " + "offset ? " + "limit ? ";
                PreparedStatement pstmt = con.prepareStatement(sql);
                // Coloca os parâmetros
                pstmt.setObject(1, Util.trataNulo(obj.getCd_atividade()));
                pstmt.setObject(2, Util.trataNulo(obj.getNm_contato()));
                pstmt.setObject(3, Util.trataNulo(obj.getEmail()));
                pstmt.setObject(4, Util.trataNulo(obj.getFax()));
                pstmt.setObject(5, Util.trataNulo(obj.getTelefone()));
                pstmt.setObject(6, Util.trataNulo(obj.getInsc_estadual()));
                pstmt.setObject(7, Util.trataNulo(obj.getCnpj()));
                pstmt.setObject(8, Util.trataNulo(obj.getCep()));
                pstmt.setObject(9, Util.trataNulo(obj.getUf()));
                pstmt.setObject(10, Util.trataNulo(obj.getCidade()));
                pstmt.setObject(11, Util.trataNulo(obj.getBairro()));
                pstmt.setObject(12, Util.trataNulo(obj.getEndereco()));
                pstmt.setObject(13, Util.trataNulo(obj.getRazao_social()));
                pstmt.setObject(14, Util.trataNulo(obj.getNm_cli()));
                pstmt.setObject(15, Util.trataNulo(obj.getCd_cli()));
                pstmt.setObject(16, Util.trataNulo(obj.getOid()));
                ResultSet rs = coloca_contadores(pstmt, 17, p_pagina, p_nr_reg, p_oid,
p_mostra_contadores);
                return this.carregaObjetos(rs, p_oid);
        }

        public void insert(Cliente obj, Connection con) throws SQLException {
                String sql =
                        " insert into t_cliente "
                                + "( cd_cli,nm_cli,razao_social,endereco,bairro"
                                + ",cidade,uf,cep,cnpj,insc_estadual"
                                + ",telefone,fax,email,nm_contato,cd_atividade ) "
                                + " values "
                                + "(?,?,?,?,?"
                                + ",?,?,?,?,?"
                                + ",?,?,?,?,?) ";
                PreparedStatement pstmt = con.prepareStatement(sql);
                //
                pstmt.setObject(1, Util.trataNulo(obj.getCd_cli()));
                pstmt.setObject(2, Util.trataNulo(obj.getNm_cli()));
                pstmt.setObject(3, Util.trataNulo(obj.getRazao_social()));
                pstmt.setObject(4, Util.trataNulo(obj.getEndereco()));
                pstmt.setObject(5, Util.trataNulo(obj.getBairro()));
                pstmt.setObject(6, Util.trataNulo(obj.getCidade()));
                pstmt.setObject(7, Util.trataNulo(obj.getUf()));
                pstmt.setObject(8, Util.trataNulo(obj.getCep()));
                pstmt.setObject(9, Util.trataNulo(obj.getCnpj()));
                pstmt.setObject(10, Util.trataNulo(obj.getInsc_estadual()));
                pstmt.setObject(11, Util.trataNulo(obj.getTelefone()));
                pstmt.setObject(12, Util.trataNulo(obj.getFax()));
                pstmt.setObject(13, Util.trataNulo(obj.getEmail()));
                pstmt.setObject(14, Util.trataNulo(obj.getNm_contato()));
                pstmt.setObject(15, Util.trataNulo(obj.getCd_atividade()));
```

```
        //
        pstmt.executeUpdate();
        pstmt.close();
}

public void update(Cliente obj, Connection con) throws SQLException {
        String sql =
                " update t_cliente "
                        + " set "
                        + " nm_cli = ?"
                        + ",razao_social = ?"
                        + ",endereco = ?"
                        + ",bairro = ?"
                        + ",cidade = ?"
                        + ",uf = ?"
                        + ",cep = ?"
                        + ",cnpj = ?"
                        + ",insc_estadual = ?"
                        + ",telefone = ?"
                        + ",fax = ?"
                        + ",email = ?"
                        + ",nm_contato = ?"
                        + ",cd_atividade = ? "
                        + " where oid = ? ";
        PreparedStatement pstmt = con.prepareStatement(sql);
        pstmt.setObject(1, Util.trataNulo(obj.getNm_cli()));
        pstmt.setObject(2, Util.trataNulo(obj.getRazao_social()));
        pstmt.setObject(3, Util.trataNulo(obj.getEndereco()));
        pstmt.setObject(4, Util.trataNulo(obj.getBairro()));
        pstmt.setObject(5, Util.trataNulo(obj.getCidade()));
        pstmt.setObject(6, Util.trataNulo(obj.getUf()));
        pstmt.setObject(7, Util.trataNulo(obj.getCep()));
        pstmt.setObject(8, Util.trataNulo(obj.getCnpj()));
        pstmt.setObject(9, Util.trataNulo(obj.getInsc_estadual()));
        pstmt.setObject(10, Util.trataNulo(obj.getTelefone()));
        pstmt.setObject(11, Util.trataNulo(obj.getFax()));
        pstmt.setObject(12, Util.trataNulo(obj.getEmail()));
        pstmt.setObject(13, Util.trataNulo(obj.getNm_contato()));
        pstmt.setObject(14, Util.trataNulo(obj.getCd_atividade()));
        pstmt.setObject(15, Util.trataNulo(obj.getOid()));
        pstmt.executeUpdate();
        pstmt.close();
}

public void delete(String p_oid, Connection con) throws SQLException {
        String sql = "delete from t_cliente where oid = ?";
        PreparedStatement pstmt = con.prepareStatement(sql);
        pstmt.setObject(1, Util.trataNulo(p_oid));
        pstmt.executeUpdate();
        pstmt.close();
}

public ResultSet coloca_contadores(
        PreparedStatement pstmt,
        int p_posicao_offset,
        int p_pagina,
        int p_nr_reg,
        String p_oid,
        String p_mostra_contadores)
        throws SQLException {
        //Carregar o Contador total
        int v_posicao_limit = p_posicao_offset + 1;
        ArrayList retorno = new ArrayList();
        ResultSet rs;
        if (Util.trataNulo(p_mostra_contadores)==null){
                pstmt.setInt(p_posicao_offset, 0);
                pstmt.setInt(v_posicao_limit, 2147483647);
                rs = pstmt.executeQuery();
                rs.last();
                contaSelect = rs.getRow();
        }
        nr_reg = p_nr_reg;
        if (p_pagina > contaSelect) {
                p_pagina = contaSelect;
        }
        pstmt.setInt(p_posicao_offset, p_pagina);
        pstmt.setInt(v_posicao_limit, nr_reg);
        rs = pstmt.executeQuery();
        rs.last();
        contaSelectAtual = rs.getRow() + p_pagina;
        rs.beforeFirst();
        return rs;
}
public int getContaSelect() {
        return contaSelect;
}
public int getContaSelectAtual() {
        return contaSelectAtual;
```

```
	}
	public String getNr_atual() {
		return nr_atual;
	}
	public int getNr_reg() {
		return nr_reg;
	}
	public void setContaSelect(int i) {
		contaSelect = i;
	}
	public void setContaSelectAtual(int i) {
		contaSelectAtual = i;
	}
	public void setNr_atual(String string) {
		nr_atual = string;
	}
	public void setNr_reg(int i) {
		nr_reg = i;
	}
}
```

CnaeSQL.java

```
package br.com.nota_fluxo.sql;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

import org.apache.log4j.Category;

import br.com.nota_fluxo.bean.Cnae;
import br.com.nota_fluxo.util.Util;

public class CnaeSQL {
	private int contaSelect;
	private int contaSelectAtual;
	private int nr_reg;
	private String nr_atual;

	Category cat = Category.getInstance(CnaeSQL.class.getName());

	private ArrayList carregaObjetos(ResultSet rs, String p_oid)
		throws SQLException {
		ArrayList retorno = new ArrayList();
		int nr_reg = 0;
		while (rs.next()) {
			if (Util.trataNulo(p_oid) != null) {
				if (Integer.parseInt(p_oid) ==
Integer.parseInt(rs.getString("oid"))) {
					nr_atual = nr_reg + "";
				}
			} else {
				Cnae cnae = new Cnae();
				cnae.setCd_cnae(rs.getString("cd_cnae"));
				cnae.setDs_cnae(rs.getString("ds_cnae"));
				cnae.setOid(rs.getString("oid"));
				retorno.add(cnae);
			}
			nr_reg++;
		}
		return retorno;
	}

	public ArrayList select(
		Connection con,
		int p_pagina,
		Cnae obj,
		int p_nr_reg,
		String p_oid,
		String p_mostra_contadores)
		throws SQLException {
		ArrayList retorno = new ArrayList();
		String sql =
		"select "
		+ "  cd_cnae"
		+ ", ds_cnae"
		+ ", oid "
		+ "from t_cnae "
		+ "where ds_cnae                                    = coalesce(?, ds_cnae) ";
		if (Util.trataNulo(obj.getCd_cnae()) == null) {
			sql = sql + "and cd_cnae          = coalesce(?, cd_cnae)";
		} else {
			sql = sql + "and cd_cnae          = ? ";
		}
```

```
                if (Util.trataNulo(obj.getOid()) == null) {
                        sql = sql + "and oid                        = coalesce(?, oid) ";
                } else {
                        sql = sql + "and oid                        = ? ";
                }

                sql = sql + "order by 1 " + "offset ? " + "limit ? ";
                PreparedStatement pstmt = con.prepareStatement(sql);
                // Coloca os parâmetros
                pstmt.setObject(1, Util.trataNulo(obj.getDs_cnae()));
                pstmt.setObject(2, Util.trataNulo(obj.getCd_cnae()));
                pstmt.setObject(3, Util.trataNulo(obj.getOid()));
                ResultSet rs = coloca_contadores(pstmt, 4, p_pagina, p_nr_reg, p_oid,
p_mostra_contadores);
                return this.carregaObjetos(rs, p_oid);
        }

        public void insert(Cnae obj, Connection con) throws SQLException {
                String sql =
                        " insert into t_cnae "
                                + "( cd_cnae,ds_cnae ) "
                                + " values "
                                + "(?,?) ";
                PreparedStatement pstmt = con.prepareStatement(sql);
                //
                pstmt.setObject(1, Util.trataNulo(obj.getCd_cnae()));
                pstmt.setObject(2, Util.trataNulo(obj.getDs_cnae()));
                //
                pstmt.executeUpdate();
                pstmt.close();
        }

        public void update(Cnae obj, Connection con) throws SQLException {
                String sql =
                        " update t_cnae "
                                + " set "
                                + " ds_cnae = ?"
                                + " where oid = ? ";
                PreparedStatement pstmt = con.prepareStatement(sql);
                pstmt.setObject(1, Util.trataNulo(obj.getDs_cnae()));
                pstmt.setObject(2, Util.trataNulo(obj.getOid()));
                pstmt.executeUpdate();
                pstmt.close();
        }

        public void delete(String p_oid, Connection con) throws SQLException {
                String sql = "delete from t_cnae where oid = ?";
                PreparedStatement pstmt = con.prepareStatement(sql);
                pstmt.setObject(1, Util.trataNulo(p_oid));
                pstmt.executeUpdate();
                pstmt.close();
        }

        public ResultSet coloca_contadores(
                PreparedStatement pstmt,
                int p_posicao_offset,
                int p_pagina,
                int p_nr_reg,
                String p_oid,
                String p_mostra_contadores)
                throws SQLException {
                //Carregar o Contador total
                int v_posicao_limit = p_posicao_offset + 1;
                ArrayList retorno = new ArrayList();
                ResultSet rs;
                if (Util.trataNulo(p_mostra_contadores)==null){
                        pstmt.setInt(p_posicao_offset, 0);
                        pstmt.setInt(v_posicao_limit, 2147483647);
                        rs = pstmt.executeQuery();
                        rs.last();
                        contaSelect = rs.getRow();
                }
                nr_reg = p_nr_reg;
                if (p_pagina > contaSelect) {
                        p_pagina = contaSelect;
                }
                pstmt.setInt(p_posicao_offset, p_pagina);
                pstmt.setInt(v_posicao_limit, nr_reg);
                rs = pstmt.executeQuery();
                rs.last();
                contaSelectAtual = rs.getRow() + p_pagina;
                rs.beforeFirst();
                return rs;
        }
        public int getContaSelect() {
                return contaSelect;
        }
        public int getContaSelectAtual() {
```

```
                return contaSelectAtual;
        }
        public String getNr_atual() {
                return nr_atual;
        }
        public int getNr_reg() {
                return nr_reg;
        }
        public void setContaSelect(int i) {
                contaSelect = i;
        }
        public void setContaSelectAtual(int i) {
                contaSelectAtual = i;
        }
        public void setNr_atual(String string) {
                nr_atual = string;
        }
        public void setNr_reg(int i) {
                nr_reg = i;
        }
}
```

Contas_pagarSQL.java

```
package br.com.nota_fluxo.sql;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

import org.apache.log4j.Category;

import br.com.nota_fluxo.bean.Contas_pagar;
import br.com.nota_fluxo.util.Util;

public class Contas_pagarSQL {
        private int contaSelect;
        private int contaSelectAtual;
        private int nr_reg;
        private String nr_atual;

        Category cat = Category.getInstance(Contas_pagarSQL.class.getName());

        private ArrayList carregaObjetos(ResultSet rs, String p_oid)
                throws SQLException {
                ArrayList retorno = new ArrayList();
                int nr_reg = 0;
                while (rs.next()) {
                        if (Util.trataNulo(p_oid) != null) {
                                if (Integer.parseInt(p_oid) ==
Integer.parseInt(rs.getString("oid"))) {
                                        nr_atual = nr_reg + "";
                                }
                        } else {
                                Contas_pagar contas_pagar = new Contas_pagar();
                                contas_pagar.setCd_emp(rs.getString("cd_emp"));
                                contas_pagar.setCd_pagar(rs.getString("cd_pagar"));
                                contas_pagar.setDt_vencto(rs.getString("dt_vencto"));
                                contas_pagar.setDt_pagto(rs.getString("dt_pagto"));
                                contas_pagar.setVl_acrescimos(rs.getString("vl_acrescimos"));
                                contas_pagar.setVl_descontos(rs.getString("vl_descontos"));
                                contas_pagar.setVl_pagar(rs.getString("vl_pagar"));
                                contas_pagar.setVl_pago(rs.getString("vl_pago"));
                                contas_pagar.setCd_tp_pagar(rs.getString("cd_tp_pagar"));
                                contas_pagar.setCd_conta(rs.getString("cd_conta"));
                                contas_pagar.setOid(rs.getString("oid"));
                                retorno.add(contas_pagar);
                        }
                        nr_reg++;
                }
                return retorno;
        }

        public ArrayList select(
                Connection con,
                int p_pagina,
                Contas_pagar obj,
                int p_nr_reg,
                String p_oid,
                String p_mostra_contadores)
                throws SQLException {
                ArrayList retorno = new ArrayList();
                String sql =
                "select "+
                "  cd_emp" +
                ",cd_pagar" +
```

```
                ",to_char(dt_vencto, 'dd/mm/yyyy') as dt_vencto " +
                ",to_char(dt_pagto, 'dd/mm/yyyy') as dt_pagto" +
                ",vl_acrescimos" +
                ",vl_descontos" +
                ",vl_pagar" +
                ",vl_pago" +
                ",cd_tp_pagar" +
                ",cd_conta "+
                ", oid " +
                "from t_contas_pagar " +
                "where cd_conta                                 = coalesce(?, cd_conta) "+
                "and cd_tp_pagar                        = coalesce(?, cd_tp_pagar) "+
                "and coalesce(vl_pago, -1)              = coalesce(?, coalesce(vl_pago, -1)) "+
                "and coalesce(vl_pagar, -1)             = coalesce(?, coalesce(vl_pagar, -1)) "+
                "and coalesce(vl_descontos,-1)    = coalesce(?, coalesce(vl_descontos,-1)) "+
                "and coalesce(vl_acrescimos, -1)= coalesce(?, coalesce(vl_acrescimos, -1)) "+
                "and coalesce(dt_pagto, to_date('0000','yyyy'))    = coalesce(to_date(?,
'dd/mm/yyyy'), coalesce(dt_pagto, to_date('0000','yyyy'))) "+
                "and dt_vencto                                  = coalesce(to_date(?,
'dd/mm/yyyy'), dt_vencto) ";
                if (Util.trataNulo(obj.getCd_pagar()) == null) {
                        sql = sql + "and cd_pagar         = coalesce(?, cd_pagar)";
                } else {
                        sql = sql + "and cd_pagar         = ? ";
                }
                if (Util.trataNulo(obj.getCd_emp()) == null) {
                        sql = sql + "and cd_emp           = coalesce(?, cd_emp)";
                } else {
                        sql = sql + "and cd_emp           = ? ";
                }

                if (Util.trataNulo(obj.getOid()) == null) {
                        sql = sql + "and oid                      = coalesce(?, oid) ";
                } else {
                        sql = sql + "and oid                      = ? ";
                }

                sql = sql + "order by 1 " + "offset ? " + "limit ? ";
                PreparedStatement pstmt = con.prepareStatement(sql);
                // Coloca os parâmetros
                pstmt.setObject(1, Util.trataNulo(obj.getCd_conta()));
                pstmt.setObject(2, Util.trataNulo(obj.getCd_tp_pagar()));
                pstmt.setObject(3, Util.trataNulo(obj.getVl_pago()));
                pstmt.setObject(4, Util.trataNulo(obj.getVl_pagar()));
                pstmt.setObject(5, Util.trataNulo(obj.getVl_descontos()));
                pstmt.setObject(6, Util.trataNulo(obj.getVl_acrescimos()));
                pstmt.setObject(7, Util.trataNulo(obj.getDt_pagto()));
                pstmt.setObject(8, Util.trataNulo(obj.getDt_vencto()));
                pstmt.setObject(9, Util.trataNulo(obj.getCd_pagar()));
                pstmt.setObject(10, Util.trataNulo(obj.getCd_emp()));
                pstmt.setObject(11, Util.trataNulo(obj.getOid()));
                ResultSet rs = coloca_contadores(pstmt, 12, p_pagina, p_nr_reg, p_oid,
p_mostra_contadores);
                return this.carregaObjetos(rs, p_oid);
        }

        public void insert(Contas_pagar obj, Connection con) throws SQLException {
                String sql =
                        " insert into t_contas_pagar "
                                + "( cd_emp,cd_pagar,dt_vencto,dt_pagto,vl_acrescimos"
                                + ",vl_descontos,vl_pagar,vl_pago,cd_tp_pagar,cd_conta ) "
                                + " values "
                                + "(?,?,to_date(?,'dd/mm/yyyy'),to_date(?,'dd/mm/yyyy'),?"
                                + ",?,?,?,?,?) ";
                PreparedStatement pstmt = con.prepareStatement(sql);
                //
                pstmt.setObject(1, Util.trataNulo(obj.getCd_emp()));
                pstmt.setObject(2, Util.trataNulo(obj.getCd_pagar()));
                pstmt.setObject(3, Util.trataNulo(obj.getDt_vencto()));
                pstmt.setObject(4, Util.trataNulo(obj.getDt_pagto()));
                pstmt.setObject(5, Util.trataNulo(obj.getVl_acrescimos()));
                pstmt.setObject(6, Util.trataNulo(obj.getVl_descontos()));
                pstmt.setObject(7, Util.trataNulo(obj.getVl_pagar()));
                pstmt.setObject(8, Util.trataNulo(obj.getVl_pago()));
                pstmt.setObject(9, Util.trataNulo(obj.getCd_tp_pagar()));
                pstmt.setObject(10, Util.trataNulo(obj.getCd_conta()));
                //
                pstmt.executeUpdate();
                pstmt.close();
        }

        public void update(Contas_pagar obj, Connection con) throws SQLException {
                String sql =
                        " update t_contas_pagar "
                                + " set "
                                + " dt_vencto = to_date(?, 'dd/mm/yyyy')"
                                + ",dt_pagto = to_date(?, 'dd/mm/yyyy')"
                                + ",vl_acrescimos = ?"
```

```
                                + ",vl_descontos = ?"
                                + ",vl_pagar = ?"
                                + ",vl_pago = ?"
                                + ",cd_tp_pagar = ?"
                                + ",cd_conta = ? "
                                + " where oid = ? ";
                PreparedStatement pstmt = con.prepareStatement(sql);
                pstmt.setObject(1, Util.trataNulo(obj.getDt_vencto()));
                pstmt.setObject(2, Util.trataNulo(obj.getDt_pagto()));
                pstmt.setObject(3, Util.trataNulo(obj.getVl_acrescimos()));
                pstmt.setObject(4, Util.trataNulo(obj.getVl_descontos()));
                pstmt.setObject(5, Util.trataNulo(obj.getVl_pagar()));
                pstmt.setObject(6, Util.trataNulo(obj.getVl_pago()));
                pstmt.setObject(7, Util.trataNulo(obj.getCd_tp_pagar()));
                pstmt.setObject(8, Util.trataNulo(obj.getCd_conta()));
                pstmt.setObject(9, Util.trataNulo(obj.getOid()));
                pstmt.executeUpdate();
                pstmt.close();
        }

        public void delete(String p_oid, Connection con) throws SQLException {
                String sql = "delete from t_contas_pagar where oid = ?";
                PreparedStatement pstmt = con.prepareStatement(sql);
                pstmt.setObject(1, Util.trataNulo(p_oid));
                pstmt.executeUpdate();
                pstmt.close();
        }

        public ResultSet coloca_contadores(
                PreparedStatement pstmt,
                int p_posicao_offset,
                int p_pagina,
                int p_nr_reg,
                String p_oid,
                String p_mostra_contadores)
                throws SQLException {
                //Carregar o Contador total
                int v_posicao_limit = p_posicao_offset + 1;
                ArrayList retorno = new ArrayList();
                ResultSet rs;
                if (Util.trataNulo(p_mostra_contadores)==null){
                        pstmt.setInt(p_posicao_offset, 0);
                        pstmt.setInt(v_posicao_limit, 2147483647);
                        rs = pstmt.executeQuery();
                        rs.last();
                        contaSelect = rs.getRow();
                }
                nr_reg = p_nr_reg;
                if (p_pagina > contaSelect) {
                        p_pagina = contaSelect;
                }
                pstmt.setInt(p_posicao_offset, p_pagina);
                pstmt.setInt(v_posicao_limit, nr_reg);
                rs = pstmt.executeQuery();
                rs.last();
                contaSelectAtual = rs.getRow() + p_pagina;
                rs.beforeFirst();
                return rs;
        }
        public int getContaSelect() {
                return contaSelect;
        }
        public int getContaSelectAtual() {
                return contaSelectAtual;
        }
        public String getNr_atual() {
                return nr_atual;
        }
        public int getNr_reg() {
                return nr_reg;
        }
        public void setContaSelect(int i) {
                contaSelect = i;
        }
        public void setContaSelectAtual(int i) {
                contaSelectAtual = i;
        }
        public void setNr_atual(String string) {
                nr_atual = string;
        }
        public void setNr_reg(int i) {
                nr_reg = i;
        }
}
```

Contas_receberSQL.java

```
package br.com.nota_fluxo.sql;
```

```
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

import org.apache.log4j.Category;

import br.com.nota_fluxo.bean.Contas_receber;
import br.com.nota_fluxo.util.Util;

public class Contas_receberSQL {
        private int contaSelect;
        private int contaSelectAtual;
        private int nr_reg;
        private String nr_atual;

        Category cat = Category.getInstance(Contas_receberSQL.class.getName());

        private ArrayList carregaObjetos(ResultSet rs, String p_oid)
                throws SQLException {
                ArrayList retorno = new ArrayList();
                int nr_reg = 0;
                while (rs.next()) {
                        if (Util.trataNulo(p_oid) != null) {
                                if (Integer.parseInt(p_oid) ==
Integer.parseInt(rs.getString("oid"))) {
                                        nr_atual = nr_reg + "";
                                }
                        } else {
                                Contas_receber contas_receber = new Contas_receber();
                                contas_receber.setCd_emp(rs.getString("cd_emp"));
                                contas_receber.setCd_receber(rs.getString("cd_receber"));
                                contas_receber.setDt_vencto(rs.getString("dt_vencto"));
                                contas_receber.setDt_pagto(rs.getString("dt_pagto"));
                                contas_receber.setVl_receber(rs.getString("vl_receber"));
                                contas_receber.setVl_recebido(rs.getString("vl_recebido"));
                                contas_receber.setNr_nota(rs.getString("nr_nota"));
                                contas_receber.setCd_tp_receber(rs.getString("cd_tp_receber"));
                                contas_receber.setCd_conta(rs.getString("cd_conta"));
                                contas_receber.setOid(rs.getString("oid"));
                                retorno.add(contas_receber);
                        }
                        nr_reg++;
                }
                return retorno;
        }

        public ArrayList select(
                Connection con,
                int p_pagina,
                Contas_receber obj,
                int p_nr_reg,
                String p_oid,
                String p_mostra_contadores)
                throws SQLException {
                ArrayList retorno = new ArrayList();
                String sql =
                "select "
                        + " cd_emp"
                        + ",cd_receber"
                        + ",to_char(dt_vencto,'dd/mm/yyyy') as dt_vencto"
                        + ",to_char(dt_pagto,'dd/mm/yyyy') as dt_pagto"
                        + ",vl_receber"
                        + ",vl_recebido"
                        + ",nr_nota"
                        + ",cd_tp_receber"
                        + ",cd_conta "
                        + ", oid "
                        + "from t_contas_receber "
                        + "where "
                        + "         cd_conta              = coalesce(?, cd_conta) "
                        + "and cd_tp_receber= coalesce(?, cd_tp_receber) "
                        + "and nr_nota           = coalesce(?, nr_nota) "
                        + "and coalesce(vl_recebido, -1)= coalesce(?, coalesce(vl_recebido, -1)) "
                        + "and coalesce(vl_receber, -1)   = coalesce(?, coalesce(vl_receber, -1)) "
                        + "and coalesce(dt_pagto, to_date('0000','yyyy'))  = coalesce(to_date(?,
'dd/mm/yyyy'), coalesce(dt_pagto, to_date('0000','yyyy'))) "
                        + "and dt_vencto                                = coalesce(to_date(?,
'dd/mm/yyyy'), dt_vencto) ";
                if (Util.trataNulo(obj.getCd_receber()) == null) {
                        sql = sql+ "and cd_receber               = coalesce(?, cd_receber)";
                } else {
                        sql = sql + "and cd_receber              = ? ";
                }
                if (Util.trataNulo(obj.getCd_emp()) == null) {
                        sql = sql+ "and cd_emp           = coalesce(?, cd_emp)";
```

```
                } else {
                        sql = sql + "and cd_emp           = ? ";
                }
                if (Util.trataNulo(obj.getOid()) == null) {
                        sql = sql + "and oid                      = coalesce(?, oid) ";
                } else {
                        sql = sql + "and oid                      = ? ";
                }

                sql = sql + "order by cd_emp desc, cd_receber desc " + "offset ? " + "limit ? ";
                PreparedStatement pstmt = con.prepareStatement(sql);
                // Coloca os parâmetros
                pstmt.setObject(1, Util.trataNulo(obj.getCd_conta()));
                pstmt.setObject(2, Util.trataNulo(obj.getCd_tp_receber()));
                pstmt.setObject(3, Util.trataNulo(obj.getNr_nota()));
                pstmt.setObject(4, Util.trataNulo(obj.getVl_recebido()));
                pstmt.setObject(5, Util.trataNulo(obj.getVl_receber()));
                pstmt.setObject(6, Util.trataNulo(obj.getDt_pagto()));
                pstmt.setObject(7, Util.trataNulo(obj.getDt_vencto()));
                pstmt.setObject(8, Util.trataNulo(obj.getCd_receber()));
                pstmt.setObject(9, Util.trataNulo(obj.getCd_emp()));
                pstmt.setObject(10, Util.trataNulo(obj.getOid()));
                ResultSet rs = coloca_contadores(pstmt, 11, p_pagina, p_nr_reg, p_oid,
p_mostra_contadores);
                return this.carregaObjetos(rs, p_oid);
        }

        public void insert(Contas_receber obj, Connection con) throws SQLException {
                String sql =
                        " insert into t_contas_receber "
                                + "( cd_emp,cd_receber,dt_vencto,dt_pagto"
                                + ",vl_receber,vl_recebido,nr_nota,cd_tp_receber,cd_conta ) "
                                + " values "
                                + "(?,?,to_date(?, 'dd/mm/yyyy'),to_date(?, 'dd/mm/yyyy')"
                                + ",?,?,?,?,?) ";
                PreparedStatement pstmt = con.prepareStatement(sql);
                //
                pstmt.setObject(1, Util.trataNulo(obj.getCd_emp()));
                pstmt.setObject(2, Util.trataNulo(obj.getCd_receber()));
                pstmt.setObject(3, Util.trataNulo(obj.getDt_vencto()));
                pstmt.setObject(4, Util.trataNulo(obj.getDt_pagto()));
                pstmt.setObject(5, Util.trataNulo(obj.getVl_receber()));
                pstmt.setObject(6, Util.trataNulo(obj.getVl_recebido()));
                pstmt.setObject(7, Util.trataNulo(obj.getNr_nota()));
                pstmt.setObject(8, Util.trataNulo(obj.getCd_tp_receber()));
                pstmt.setObject(9, Util.trataNulo(obj.getCd_conta()));
                //
                pstmt.executeUpdate();
                pstmt.close();
        }

        public void update(Contas_receber obj, Connection con) throws SQLException {
                String sql =
                        " update t_contas_receber "
                                + " set "
                                + " dt_vencto = to_date(?, 'dd/mm/yyyy')"
                                + ",dt_pagto = to_date(?, 'dd/mm/yyyy')"
                                + ",vl_receber = ?"
                                + ",vl_recebido = ?"
                                + ",nr_nota = ?"
                                + ",cd_tp_receber = ?"
                                + ",cd_conta = ? "
                                + " where oid = ? ";
                PreparedStatement pstmt = con.prepareStatement(sql);
                pstmt.setObject(1, Util.trataNulo(obj.getDt_vencto()));
                pstmt.setObject(2, Util.trataNulo(obj.getDt_pagto()));
                pstmt.setObject(3, Util.trataNulo(obj.getVl_receber()));
                pstmt.setObject(4, Util.trataNulo(obj.getVl_recebido()));
                pstmt.setObject(5, Util.trataNulo(obj.getNr_nota()));
                pstmt.setObject(6, Util.trataNulo(obj.getCd_tp_receber()));
                pstmt.setObject(7, Util.trataNulo(obj.getCd_conta()));
                pstmt.setObject(8, Util.trataNulo(obj.getOid()));
                pstmt.executeUpdate();
                pstmt.close();
        }

        public void delete(String p_oid, Connection con) throws SQLException {
                String sql = "delete from t_contas_receber where oid = ?";
                PreparedStatement pstmt = con.prepareStatement(sql);
                pstmt.setObject(1, Util.trataNulo(p_oid));
                pstmt.executeUpdate();
                pstmt.close();
        }

        public ResultSet coloca_contadores(
                PreparedStatement pstmt,
                int p_posicao_offset,
                int p_pagina,
```

```
                int p_nr_reg,
                String p_oid,
                String p_mostra_contadores)
                throws SQLException {
                //Carregar o Contador total
                int v_posicao_limit = p_posicao_offset + 1;
                ArrayList retorno = new ArrayList();
                ResultSet rs;
                if (Util.trataNulo(p_mostra_contadores)==null){
                        pstmt.setInt(p_posicao_offset, 0);
                        pstmt.setInt(v_posicao_limit, 2147483647);
                        rs = pstmt.executeQuery();
                        rs.last();
                        contaSelect = rs.getRow();
                }
                nr_reg = p_nr_reg;
                if (p_pagina > contaSelect) {
                        p_pagina = contaSelect;
                }
                pstmt.setInt(p_posicao_offset, p_pagina);
                pstmt.setInt(v_posicao_limit, nr_reg);
                rs = pstmt.executeQuery();
                rs.last();
                contaSelectAtual = rs.getRow() + p_pagina;
                rs.beforeFirst();
                return rs;
        }
        public int getContaSelect() {
                return contaSelect;
        }
        public int getContaSelectAtual() {
                return contaSelectAtual;
        }
        public String getNr_atual() {
                return nr_atual;
        }
        public int getNr_reg() {
                return nr_reg;
        }
        public void setContaSelect(int i) {
                contaSelect = i;
        }
        public void setContaSelectAtual(int i) {
                contaSelectAtual = i;
        }
        public void setNr_atual(String string) {
                nr_atual = string;
        }
        public void setNr_reg(int i) {
                nr_reg = i;
        }
}
```

ContasSQL.java

```
package br.com.nota_fluxo.sql;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

import org.apache.log4j.Category;

import br.com.nota_fluxo.bean.Contas;
import br.com.nota_fluxo.util.Util;

public class ContasSQL {
        private int contaSelect;
        private int contaSelectAtual;
        private int nr_reg;
        private String nr_atual;

        Category cat = Category.getInstance(ContasSQL.class.getName());

        private ArrayList carregaObjetos(ResultSet rs, String p_oid)
                throws SQLException {
                ArrayList retorno = new ArrayList();
                int nr_reg = 0;
                while (rs.next()) {
                        if (Util.trataNulo(p_oid) != null) {
                                if (Integer.parseInt(p_oid) ==
Integer.parseInt(rs.getString("oid"))) {
                                        nr_atual = nr_reg + "";
                                }
                        } else {
                                Contas contas = new Contas();
```

```
                                contas.setCd_emp(rs.getString("cd_emp"));
                                contas.setCd_conta(rs.getString("cd_conta"));
                                contas.setDs_conta(rs.getString("ds_conta"));
                                contas.setOid(rs.getString("oid"));
                                retorno.add(contas);
                        }
                        nr_reg++;
                }
                return retorno;
        }

        public ArrayList select(
                Connection con,
                int p_pagina,
                Contas obj,
                int p_nr_reg,
                String p_oid,
                String p_mostra_contadores)
                throws SQLException {
                ArrayList retorno = new ArrayList();
                String sql =
                "select "
                        + " cd_emp"
                        + ",cd_conta"
                        + ",ds_conta"
                        + ", oid "
                        + "from t_contas "
                        + "where ds_conta                                                = coalesce(?,
ds_conta) ";
                if (Util.trataNulo(obj.getCd_conta()) == null) {
                        sql = sql + "and cd_conta         = coalesce(?, cd_conta)";
                } else {
                        sql = sql + "and cd_conta         = ? ";
                }
                if (Util.trataNulo(obj.getCd_emp()) == null) {
                        sql = sql + "and cd_emp           = coalesce(?, cd_emp)";
                } else {
                        sql = sql + "and cd_emp           = ? ";
                }

                if (Util.trataNulo(obj.getOid()) == null) {
                        sql = sql + "and oid                        = coalesce(?, oid) ";
                } else {
                        sql = sql + "and oid                        = ? ";
                }

                sql = sql + "order by 1 " + "offset ? " + "limit ? ";
                PreparedStatement pstmt = con.prepareStatement(sql);
                // Coloca os parâmetros
                pstmt.setObject(1, Util.trataNulo(obj.getDs_conta()));
                pstmt.setObject(2, Util.trataNulo(obj.getCd_conta()));
                pstmt.setObject(3, Util.trataNulo(obj.getCd_emp()));
                pstmt.setObject(4, Util.trataNulo(obj.getOid()));
                ResultSet rs = coloca_contadores(pstmt, 5, p_pagina, p_nr_reg, p_oid,
p_mostra_contadores);
                return this.carregaObjetos(rs, p_oid);
        }

        public void insert(Contas obj, Connection con) throws SQLException {
                String sql =
                        " insert into t_contas "
                                + "( cd_emp,cd_conta,ds_conta ) "
                                + " values "
                                + "(?,?,?) ";
                PreparedStatement pstmt = con.prepareStatement(sql);
                //
                pstmt.setObject(1, Util.trataNulo(obj.getCd_emp()));
                pstmt.setObject(2, Util.trataNulo(obj.getCd_conta()));
                pstmt.setObject(3, Util.trataNulo(obj.getDs_conta()));
                //
                pstmt.executeUpdate();
                pstmt.close();
        }

        public void update(Contas obj, Connection con) throws SQLException {
                String sql =
                        " update t_contas "
                                + " set "
                                + " ds_conta = ?"
                                + " where oid = ? ";
                PreparedStatement pstmt = con.prepareStatement(sql);
                pstmt.setObject(1, Util.trataNulo(obj.getDs_conta()));
                pstmt.setObject(2, Util.trataNulo(obj.getOid()));
                pstmt.executeUpdate();
                pstmt.close();
        }

        public void delete(String p_oid, Connection con) throws SQLException {
```

```
                String sql = "delete from t_contas where oid = ?";
                PreparedStatement pstmt = con.prepareStatement(sql);
                pstmt.setObject(1, Util.trataNulo(p_oid));
                pstmt.executeUpdate();
                pstmt.close();
        }

        public ResultSet coloca_contadores(
                PreparedStatement pstmt,
                int p_posicao_offset,
                int p_pagina,
                int p_nr_reg,
                String p_oid,
                String p_mostra_contadores)
                throws SQLException {
                //Carregar o Contador total
                int v_posicao_limit = p_posicao_offset + 1;
                ArrayList retorno = new ArrayList();
                ResultSet rs;
                if (Util.trataNulo(p_mostra_contadores)==null){
                        pstmt.setInt(p_posicao_offset, 0);
                        pstmt.setInt(v_posicao_limit, 2147483647);
                        rs = pstmt.executeQuery();
                        rs.last();
                        contaSelect = rs.getRow();
                }
                nr_reg = p_nr_reg;
                if (p_pagina > contaSelect) {
                        p_pagina = contaSelect;
                }
                pstmt.setInt(p_posicao_offset, p_pagina);
                pstmt.setInt(v_posicao_limit, nr_reg);
                rs = pstmt.executeQuery();
                rs.last();
                contaSelectAtual = rs.getRow() + p_pagina;
                rs.beforeFirst();
                return rs;
        }
        public int getContaSelect() {
                return contaSelect;
        }
        public int getContaSelectAtual() {
                return contaSelectAtual;
        }
        public String getNr_atual() {
                return nr_atual;
        }
        public int getNr_reg() {
                return nr_reg;
        }
        public void setContaSelect(int i) {
                contaSelect = i;
        }
        public void setContaSelectAtual(int i) {
                contaSelectAtual = i;
        }
        public void setNr_atual(String string) {
                nr_atual = string;
        }
        public void setNr_reg(int i) {
                nr_reg = i;
        }
}
```

EmpresaSQL.java

```
package br.com.nota_fluxo.sql;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

import org.apache.log4j.Category;

import br.com.nota_fluxo.bean.Empresa;
import br.com.nota_fluxo.util.Util;

public class EmpresaSQL {
        private int contaSelect;
        private int contaSelectAtual;
        private int nr_reg;
        private String nr_atual;

        Category cat = Category.getInstance(EmpresaSQL.class.getName());

        private ArrayList carregaObjetos(ResultSet rs, String p_oid)
```

```
                throws SQLException {
                ArrayList retorno = new ArrayList();
                int nr_reg = 0;
                while (rs.next()) {
                        if (Util.trataNulo(p_oid) != null) {
                                if(Integer.parseInt(p_oid)==Integer.parseInt(rs.getString("oid")))
{
                                        nr_atual = nr_reg + "";
                                }
                        } else {
                                Empresa empresa = new Empresa();
                                empresa.setCd_emp(rs.getString("cd_emp"));
                                empresa.setNm_emp(rs.getString("nm_emp"));
                                empresa.setRazao_social(rs.getString("razao_social"));
                                empresa.setEndereco(rs.getString("endereco"));
                                empresa.setBairro(rs.getString("bairro"));
                                empresa.setCidade(rs.getString("cidade"));
                                empresa.setUf(rs.getString("uf"));
                                empresa.setCep(rs.getString("cep"));
                                empresa.setCnpj(rs.getString("cnpj"));
                                empresa.setInsc_estadual(rs.getString("insc_estadual"));
                                empresa.setTelefone(rs.getString("telefone"));
                                empresa.setFax(rs.getString("fax"));
                                empresa.setEmail(rs.getString("email"));
                                empresa.setNm_contato(rs.getString("nm_contato"));
                                empresa.setCd_atividade(rs.getString("cd_atividade"));
                                empresa.setOid(rs.getString("oid"));
                                retorno.add(empresa);
                        }
                        nr_reg++;
                }
                return retorno;
        }

        public ArrayList select(
                Connection con,
                int p_pagina,
                Empresa obj,
                int p_nr_reg,
                String p_oid,
                String p_mostra_contadores)
                throws SQLException {
                ArrayList retorno = new ArrayList();
                String sql =
                "select "
                + "  cd_emp"
                + ", nm_emp"
                + ", razao_social"
                + ", endereco"
                + ", bairro"
                + ", cidade"
                + ", uf"
                + ", cep"
                + ", cnpj"
                + ", insc_estadual"
                + ", telefone"
                + ", fax"
                + ", email"
                + ", nm_contato"
                + ", cd_atividade "
                + ", oid "
                + "from t_empresa "
                + "where "
                + "      cd_atividade                              = coalesce(?,cd_atividade) "
                + "and coalesce(nm_contato,'-1')= coalesce(?,coalesce(nm_contato, '-1'))"
                + "and coalesce(email, '-1')      = coalesce(?,coalesce(email, '-1')) "
                + "and coalesce(fax, '-1')                = coalesce(?,coalesce(fax, '-1')) "
                + "and coalesce(telefone, '-1')   = coalesce(?,coalesce(telefone, '-1')) "
                + "and coalesce(insc_estadual,'-1')= coalesce(?,coalesce(insc_estadual,'-1'))"
                + "and coalesce(cnpj, -1)         = coalesce(?,coalesce(cnpj, -1)) "
                + "and coalesce(cep, -1)          = coalesce(?,coalesce(cep, -1)) "
                + "and uf                                           = coalesce(?,uf) "
                + "and cidade                                       = coalesce(?,cidade) "
                + "and coalesce(bairro, '-1')     = coalesce(?,coalesce(bairro, '-1')) "
                + "and endereco                                     = coalesce(?,endereco) "
                + "and razao_social                                 = coalesce(?,razao_social) "
                + "and nm_emp                                       = coalesce(?,nm_emp) ";
                if (Util.trataNulo(obj.getCd_emp()) == null) {
                        sql = sql + "and cd_emp           = coalesce(?, cd_emp)";
                } else {
                        sql = sql + "and cd_emp           = ? ";
                }

                if (Util.trataNulo(obj.getOid()) == null) {
                        sql = sql + "and oid                        = coalesce(?, oid) ";
                } else {
                        sql = sql + "and oid                        = ? ";
                }
```

```
        sql = sql + "order by 1 " + "offset ? " + "limit ? ";
        PreparedStatement pstmt = con.prepareStatement(sql);
        // Coloca os parâmetros
        pstmt.setObject(1, Util.trataNulo(obj.getCd_atividade()));
        pstmt.setObject(2, Util.trataNulo(obj.getNm_contato()));
        pstmt.setObject(3, Util.trataNulo(obj.getEmail()));
        pstmt.setObject(4, Util.trataNulo(obj.getFax()));
        pstmt.setObject(5, Util.trataNulo(obj.getTelefone()));
        pstmt.setObject(6, Util.trataNulo(obj.getInsc_estadual()));
        pstmt.setObject(7, Util.trataNulo(obj.getCnpj()));
        pstmt.setObject(8, Util.trataNulo(obj.getCep()));
        pstmt.setObject(9, Util.trataNulo(obj.getUf()));
        pstmt.setObject(10, Util.trataNulo(obj.getCidade()));
        pstmt.setObject(11, Util.trataNulo(obj.getBairro()));
        pstmt.setObject(12, Util.trataNulo(obj.getEndereco()));
        pstmt.setObject(13, Util.trataNulo(obj.getRazao_social()));
        pstmt.setObject(14, Util.trataNulo(obj.getNm_emp()));
        pstmt.setObject(15, Util.trataNulo(obj.getCd_emp()));
        pstmt.setObject(16, Util.trataNulo(obj.getOid()));
        ResultSet rs=coloca_contadores(pstmt,17,p_pagina,p_nr_reg

,p_oid,p_mostra_contadores);
        return this.carregaObjetos(rs, p_oid);
}

public void insert(Empresa obj, Connection con) throws SQLException {
        String sql =
                " insert into t_empresa "
                        + "( cd_emp,nm_emp,razao_social,endereco,bairro"
                        + ",cidade,uf,cep,cnpj,insc_estadual"
                        + ",telefone,fax,email,nm_contato,cd_atividade ) "
                        + " values "
                        + "(?,?,?,?,?"
                        + ",?,?,?,?,?"
                        + ",?,?,?,?,?) ";
        PreparedStatement pstmt = con.prepareStatement(sql);
        //
        pstmt.setObject(1, Util.trataNulo(obj.getCd_emp()));
        pstmt.setObject(2, Util.trataNulo(obj.getNm_emp()));
        pstmt.setObject(3, Util.trataNulo(obj.getRazao_social()));
        pstmt.setObject(4, Util.trataNulo(obj.getEndereco()));
        pstmt.setObject(5, Util.trataNulo(obj.getBairro()));
        pstmt.setObject(6, Util.trataNulo(obj.getCidade()));
        pstmt.setObject(7, Util.trataNulo(obj.getUf()));
        pstmt.setObject(8, Util.trataNulo(obj.getCep()));
        pstmt.setObject(9, Util.trataNulo(obj.getCnpj()));
        pstmt.setObject(10, Util.trataNulo(obj.getInsc_estadual()));
        pstmt.setObject(11, Util.trataNulo(obj.getTelefone()));
        pstmt.setObject(12, Util.trataNulo(obj.getFax()));
        pstmt.setObject(13, Util.trataNulo(obj.getEmail()));
        pstmt.setObject(14, Util.trataNulo(obj.getNm_contato()));
        pstmt.setObject(15, Util.trataNulo(obj.getCd_atividade()));
        //
        pstmt.executeUpdate();
        pstmt.close();
}

public void update(Empresa obj, Connection con) throws SQLException {
        String sql =
                " update t_empresa "
                        + " set "
                        + " nm_emp = ?"
                        + ",razao_social = ?"
                        + ",endereco = ?"
                        + ",bairro = ?"
                        + ",cidade = ?"
                        + ",uf = ?"
                        + ",cep = ?"
                        + ",cnpj = ?"
                        + ",insc_estadual = ?"
                        + ",telefone = ?"
                        + ",fax = ?"
                        + ",email = ?"
                        + ",nm_contato = ?"
                        + ",cd_atividade = ? "
                        + " where oid = ? ";
        PreparedStatement pstmt = con.prepareStatement(sql);
        pstmt.setObject(1, Util.trataNulo(obj.getNm_emp()));
        pstmt.setObject(2, Util.trataNulo(obj.getRazao_social()));
        pstmt.setObject(3, Util.trataNulo(obj.getEndereco()));
        pstmt.setObject(4, Util.trataNulo(obj.getBairro()));
        pstmt.setObject(5, Util.trataNulo(obj.getCidade()));
        pstmt.setObject(6, Util.trataNulo(obj.getUf()));
        pstmt.setObject(7, Util.trataNulo(obj.getCep()));
        pstmt.setObject(8, Util.trataNulo(obj.getCnpj()));
        pstmt.setObject(9, Util.trataNulo(obj.getInsc_estadual()));
        pstmt.setObject(10, Util.trataNulo(obj.getTelefone()));
```

```
                pstmt.setObject(11, Util.trataNulo(obj.getFax()));
                pstmt.setObject(12, Util.trataNulo(obj.getEmail()));
                pstmt.setObject(13, Util.trataNulo(obj.getNm_contato()));
                pstmt.setObject(14, Util.trataNulo(obj.getCd_atividade()));
                pstmt.setObject(15, Util.trataNulo(obj.getOid()));
                pstmt.executeUpdate();
                pstmt.close();
        }

        public void delete(String p_oid, Connection con) throws SQLException {
                String sql = "delete from t_empresa where oid = ?";
                PreparedStatement pstmt = con.prepareStatement(sql);
                pstmt.setObject(1, Util.trataNulo(p_oid));
                pstmt.executeUpdate();
                pstmt.close();
        }

        public ResultSet coloca_contadores(
                PreparedStatement pstmt,
                int p_posicao_offset,
                int p_pagina,
                int p_nr_reg,
                String p_oid,
                String p_mostra_contadores)
                throws SQLException {
                //Carregar o Contador total
                int v_posicao_limit = p_posicao_offset + 1;
                ArrayList retorno = new ArrayList();
                ResultSet rs;
                if (Util.trataNulo(p_mostra_contadores)==null){
                        pstmt.setInt(p_posicao_offset, 0);
                        pstmt.setInt(v_posicao_limit, 2147483647);
                        rs = pstmt.executeQuery();
                        rs.last();
                        contaSelect = rs.getRow();
                }
                nr_reg = p_nr_reg;
                if (p_pagina > contaSelect) {
                        p_pagina = contaSelect;
                }
                pstmt.setInt(p_posicao_offset, p_pagina);
                pstmt.setInt(v_posicao_limit, nr_reg);
                rs = pstmt.executeQuery();
                rs.last();
                contaSelectAtual = rs.getRow() + p_pagina;
                rs.beforeFirst();
                return rs;
        }
        public int getContaSelect() {
                return contaSelect;
        }
        public int getContaSelectAtual() {
                return contaSelectAtual;
        }
        public String getNr_atual() {
                return nr_atual;
        }
        public int getNr_reg() {
                return nr_reg;
        }
        public void setContaSelect(int i) {
                contaSelect = i;
        }
        public void setContaSelectAtual(int i) {
                contaSelectAtual = i;
        }
        public void setNr_atual(String string) {
                nr_atual = string;
        }
        public void setNr_reg(int i) {
                nr_reg = i;
        }
}
```

Movimento_contasSQL.java

```
package br.com.nota_fluxo.sql;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

import org.apache.log4j.Category;

import br.com.nota_fluxo.bean.Movimento_contas;
import br.com.nota_fluxo.util.Util;
```

```
public class Movimento_contasSQL {
        private int contaSelect;
        private int contaSelectAtual;
        private int nr_reg;
        private String nr_atual;

        Category cat = Category.getInstance(Movimento_contasSQL.class.getName());

        private ArrayList carregaObjetos(ResultSet rs, String p_oid)
                throws SQLException {
                ArrayList retorno = new ArrayList();
                int nr_reg = 0;
                while (rs.next()) {
                        if (Util.trataNulo(p_oid) != null) {
                                if (Integer.parseInt(p_oid) ==
Integer.parseInt(rs.getString("oid"))) {
                                        nr_atual = nr_reg + "";
                                }
                        } else {
                                Movimento_contas movimento_contas = new Movimento_contas();
                                movimento_contas.setDt_movto(rs.getString("dt_movto"));
                                movimento_contas.setNr_movto(rs.getString("nr_movto"));
                                movimento_contas.setCd_emp(rs.getString("cd_emp"));
                                movimento_contas.setNr_conta(rs.getString("nr_conta"));
                                movimento_contas.setDs_movto(rs.getString("ds_movto"));
                                movimento_contas.setVl_movto(rs.getString("vl_movto"));
                                movimento_contas.setOid(rs.getString("oid"));
                                retorno.add(movimento_contas);
                        }
                        nr_reg++;
                }
                return retorno;
        }

        public ArrayList select(
                Connection con,
                int p_pagina,
                Movimento_contas obj,
                int p_nr_reg,
                String p_oid,
                String p_mostra_contadores)
                throws SQLException {
                ArrayList retorno = new ArrayList();
                String sql =
                "select "
                        + "  to_char(dt_movto, 'dd/mm/yyyy') as dt_movto"
                        + ",nr_movto"
                        + ",cd_emp"
                        + ",nr_conta"
                        + ",ds_movto"
                        + ",vl_movto"
                        + ", oid "
                        + "from t_movimento_contas "
                        + "where "
                        + "             vl_movto= coalesce(?, vl_movto) "
                        + "and ds_movto  = coalesce(?, ds_movto) "
                        + "and nr_conta  = coalesce(?, nr_conta) "
                        + "and cd_emp    = coalesce(?, cd_emp) ";
                if (Util.trataNulo(obj.getNr_movto()) == null) {
                        sql = sql + "and nr_movto         = coalesce(?, nr_movto)";
                } else {
                        sql = sql + "and nr_movto         = ? ";
                }
                if (Util.trataNulo(obj.getDt_movto()) == null) {
                        sql = sql + "and dt_movto         = coalesce(to_date(?, 'dd/mm/yyyy'),
dt_movto)";
                } else {
                        sql = sql + "and dt_movto         = ? ";
                }
                if (Util.trataNulo(obj.getOid()) == null) {
                        sql = sql + "and oid                      = coalesce(?, oid) ";
                } else {
                        sql = sql + "and oid                      = ? ";
                }

                sql = sql + "order by 1 " + "offset ? " + "limit ? ";
                PreparedStatement pstmt = con.prepareStatement(sql);
                // Coloca os parâmetros
                pstmt.setObject(1, Util.trataNulo(obj.getVl_movto()));
                pstmt.setObject(2, Util.trataNulo(obj.getDs_movto()));
                pstmt.setObject(3, Util.trataNulo(obj.getNr_conta()));
                pstmt.setObject(4, Util.trataNulo(obj.getCd_emp()));
                pstmt.setObject(5, Util.trataNulo(obj.getNr_movto()));
                pstmt.setObject(6, Util.trataNulo(obj.getDt_movto()));
                pstmt.setObject(7, Util.trataNulo(obj.getOid()));
                ResultSet rs = coloca_contadores(pstmt, 8, p_pagina, p_nr_reg, p_oid,
p_mostra_contadores);
```

```
        return this.carregaObjetos(rs, p_oid);
}

public void insert(Movimento_contas obj, Connection con) throws SQLException {
        String sql =
                " insert into t_movimento_contas "
                        + "( dt_movto,nr_movto,cd_emp,nr_conta,ds_movto,vl_movto ) "
                        + " values "
                        + "(to_date(?, 'dd/mm/yyyy'),?,?,?,?,?) ";
        PreparedStatement pstmt = con.prepareStatement(sql);
        //
        pstmt.setObject(1, Util.trataNulo(obj.getDt_movto()));
        pstmt.setObject(2, Util.trataNulo(obj.getNr_movto()));
        pstmt.setObject(3, Util.trataNulo(obj.getCd_emp()));
        pstmt.setObject(4, Util.trataNulo(obj.getNr_conta()));
        pstmt.setObject(5, Util.trataNulo(obj.getDs_movto()));
        pstmt.setObject(6, Util.trataNulo(obj.getVl_movto()));
        //
        pstmt.executeUpdate();
        pstmt.close();
}

public void update(Movimento_contas obj, Connection con) throws SQLException {
        String sql =
                " update t_movimento_contas "
                        + " set "
                        + " cd_emp = ?"
                        + ",nr_conta = ?"
                        + ",ds_movto = ?"
                        + ",vl_movto = ?"
                        + " where oid = ? ";
        PreparedStatement pstmt = con.prepareStatement(sql);
        pstmt.setObject(1, Util.trataNulo(obj.getCd_emp()));
        pstmt.setObject(2, Util.trataNulo(obj.getNr_conta()));
        pstmt.setObject(3, Util.trataNulo(obj.getDs_movto()));
        pstmt.setObject(4, Util.trataNulo(obj.getVl_movto()));
        pstmt.setObject(5, Util.trataNulo(obj.getOid()));
        pstmt.executeUpdate();
        pstmt.close();
}

public void delete(String p_oid, Connection con) throws SQLException {
        String sql = "delete from t_movimento_contas where oid = ?";
        PreparedStatement pstmt = con.prepareStatement(sql);
        pstmt.setObject(1, Util.trataNulo(p_oid));
        pstmt.executeUpdate();
        pstmt.close();
}

public ResultSet coloca_contadores(
        PreparedStatement pstmt,
        int p_posicao_offset,
        int p_pagina,
        int p_nr_reg,
        String p_oid,
        String p_mostra_contadores)
        throws SQLException {
        //Carregar o Contador total
        int v_posicao_limit = p_posicao_offset + 1;
        ArrayList retorno = new ArrayList();
        ResultSet rs;
        if (Util.trataNulo(p_mostra_contadores)==null){
                pstmt.setInt(p_posicao_offset, 0);
                pstmt.setInt(v_posicao_limit, 2147483647);
                rs = pstmt.executeQuery();
                rs.last();
                contaSelect = rs.getRow();
        }
        nr_reg = p_nr_reg;
        if (p_pagina > contaSelect) {
                p_pagina = contaSelect;
        }
        pstmt.setInt(p_posicao_offset, p_pagina);
        pstmt.setInt(v_posicao_limit, nr_reg);
        rs = pstmt.executeQuery();
        rs.last();
        contaSelectAtual = rs.getRow() + p_pagina;
        rs.beforeFirst();
        return rs;
}
public int getContaSelect() {
        return contaSelect;
}
public int getContaSelectAtual() {
        return contaSelectAtual;
}
public String getNr_atual() {
        return nr_atual;
```

```
        }
        public int getNr_reg() {
                return nr_reg;
        }
        public void setContaSelect(int i) {
                contaSelect = i;
        }
        public void setContaSelectAtual(int i) {
                contaSelectAtual = i;
        }
        public void setNr_atual(String string) {
                nr_atual = string;
        }
        public void setNr_reg(int i) {
                nr_reg = i;
        }
}
```

Nota_fiscalSQL.java

```
package br.com.nota_fluxo.sql;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

import org.apache.log4j.Category;

import br.com.nota_fluxo.bean.Nota_fiscal;
import br.com.nota_fluxo.util.Util;

public class Nota_fiscalSQL {
        private int contaSelect;
        private int contaSelectAtual;
        private int nr_reg;
        private String nr_atual;

        Category cat = Category.getInstance(Nota_fiscalSQL.class.getName());

        private ArrayList carregaObjetos(ResultSet rs, String p_oid)
                throws SQLException {
                ArrayList retorno = new ArrayList();
                int nr_reg = 0;
                while (rs.next()) {
                        if (Util.trataNulo(p_oid) != null) {
                                if (Integer.parseInt(p_oid) ==
Integer.parseInt(rs.getString("oid"))) {
                                        nr_atual = nr_reg + "";
                                }
                        } else {
                                Nota_fiscal nota_fiscal = new Nota_fiscal();
                                nota_fiscal.setCd_emp(rs.getString("cd_emp"));
                                nota_fiscal.setNr_nota(rs.getString("nr_nota"));
                                nota_fiscal.setDt_ref(rs.getString("dt_ref"));
                                nota_fiscal.setCd_cli(rs.getString("cd_cli"));
                                nota_fiscal.setVl_bruto(rs.getString("vl_bruto"));
                                nota_fiscal.setDt_vencto(rs.getString("dt_vencto"));
                                nota_fiscal.setVl_liquido(rs.getString("vl_liquido"));
                                nota_fiscal.setIrff_retido(rs.getString("irff_retido"));
                                nota_fiscal.setIrff_pagar(rs.getString("irff_pagar"));
                                nota_fiscal.setCofins_retido(rs.getString("cofins_retido"));
                                nota_fiscal.setPis_retido(rs.getString("pis_retido"));
                                nota_fiscal.setCsll_retido(rs.getString("csll_retido"));
                                nota_fiscal.setIss_retido(rs.getString("iss_retido"));
                                nota_fiscal.setIrff_pagar(rs.getString("iss_pagar"));

        nota_fiscal.setTotal_impostos_retidos(rs.getString("total_impostos_retidos"));
                                nota_fiscal.setCst(rs.getString("cst"));
                                nota_fiscal.setIrlp(rs.getString("Irlp"));
                                nota_fiscal.setCfps(rs.getString("cfps"));

        nota_fiscal.setContribuicao_social(rs.getString("contribuicao_social"));
                                nota_fiscal.setCd_conta(rs.getString("cd_conta"));
                                nota_fiscal.setOid(rs.getString("oid"));
                                retorno.add(nota_fiscal);
                        }
                        nr_reg++;
                }
                return retorno;
        }

        public ArrayList select(
                Connection con,
                int p_pagina,
                Nota_fiscal obj,
                int p_nr_reg,
```

```
                String p_oid,
                String p_mostra_contadores)
                throws SQLException {
                ArrayList retorno = new ArrayList();
                String sql =
                "select "
                        + "  cd_emp"
                        + ",nr_nota"
                        + ",to_char(dt_ref, 'dd/mm/yyyy') as dt_ref"
                        + ",cd_cli"
                        + ",vl_bruto"
                        + ",to_char(dt_vencto, 'dd/mm/yyyy') as dt_vencto"
                        + ",vl_liquido"
                        + ",irff_retido"
                        + ",irff_pagar"
                        + ",cofins_retido"
                        + ",pis_retido"
                        + ",csll_retido"
                        + ",iss_retido"
                        + ",iss_pagar"
                        + ",total_impostos_retidos"
                        + ",cst"
                        + ",Irlp"
                        + ",cfps"
                        + ",contribuicao_social"
                        + ",cd_conta"
                        + ", oid "
                        + "from t_nota_fiscal "
                        + "where "
                        + "         coalesce(cd_conta, -1)         = coalesce(?, coalesce(cd_conta,
-1)) "
                        + "and coalesce(contribuicao_social, -1)= coalesce(?,
coalesce(contribuicao_social, -1)) "
                        + "and coalesce(cfps, -1)                  = coalesce(?, coalesce(cfps, -1))
"
                        + "and coalesce(irlp, -1)                  = coalesce(?, coalesce(irlp, -1))
"
                        + "and coalesce(cst, -1)                   = coalesce(?, coalesce(cst, -1))
"
                        + "and coalesce(total_impostos_retidos, -1)= coalesce(?,
coalesce(total_impostos_retidos, -1)) "
                        + "and coalesce(iss_pagar, -1)             = coalesce(?, coalesce(iss_pagar,
-1)) "
                        + "and coalesce(iss_retido, -1)            = coalesce(?,
coalesce(iss_retido, -1)) "
                        + "and coalesce(csll_retido, -1)  = coalesce(?, coalesce(csll_retido, -1)) "
                        + "and coalesce(pis_retido, -1)            = coalesce(?,
coalesce(pis_retido, -1)) "
                        + "and coalesce(cofins_retido, -1)         = coalesce(?,
coalesce(cofins_retido, -1)) "
                        + "and coalesce(irff_pagar, -1)            = coalesce(?,
coalesce(irff_pagar, -1)) "
                        + "and coalesce(irff_retido, -1)  = coalesce(?, coalesce(irff_retido, -1)) "
                        + "and vl_liquido                          = coalesce(?, vl_liquido) "
                        + "and coalesce(dt_vencto, to_date('0000','yyyy')) = coalesce(to_date(?,
'dd/mm/yyyy'), coalesce(dt_vencto, to_date('0000','yyyy'))) "
                        + "and vl_bruto                            = coalesce(?, vl_bruto) "
                        + "and cd_cli                              = coalesce(?, cd_cli) "
                        + "and dt_ref                              = coalesce(to_date(?,
'dd/mm/yyyy'), dt_ref) ";
                if (Util.trataNulo(obj.getNr_nota()) == null) {
                        sql = sql + "and nr_nota          = coalesce(?, nr_nota)";
                } else {
                        sql = sql + "and nr_nota          = ? ";
                }
                if (Util.trataNulo(obj.getCd_emp()) == null) {
                        sql = sql + "and cd_emp           = coalesce(?, cd_emp)";
                } else {
                        sql = sql + "and cd_emp           = ? ";
                }

                if (Util.trataNulo(obj.getOid()) == null) {
                        sql = sql + "and oid                       = coalesce(?, oid) ";
                } else {
                        sql = sql + "and oid                       = ? ";
                }

                sql = sql + "order by 1 " + "offset ? " + "limit ? ";
                PreparedStatement pstmt = con.prepareStatement(sql);
                // Coloca os parâmetros
                pstmt.setObject(1, Util.trataNulo(obj.getCd_conta()));
                pstmt.setObject(2, Util.trataNulo(obj.getContribuicao_social()));
                pstmt.setObject(3, Util.trataNulo(obj.getCfps()));
                pstmt.setObject(4, Util.trataNulo(obj.getIrlp()));
                pstmt.setObject(5, Util.trataNulo(obj.getCst()));
                pstmt.setObject(6, Util.trataNulo(obj.getTotal_impostos_retidos()));
                pstmt.setObject(7, Util.trataNulo(obj.getIss_pagar()));
                pstmt.setObject(8, Util.trataNulo(obj.getIss_retido()));
```

```
                pstmt.setObject(9, Util.trataNulo(obj.getCsll_retido()));
                pstmt.setObject(10, Util.trataNulo(obj.getPis_retido()));
                pstmt.setObject(11, Util.trataNulo(obj.getCofins_retido()));
                pstmt.setObject(12, Util.trataNulo(obj.getIrff_pagar()));
                pstmt.setObject(13, Util.trataNulo(obj.getIrff_retido()));
                pstmt.setObject(14, Util.trataNulo(obj.getVl_liquido()));
                pstmt.setObject(15, Util.trataNulo(obj.getDt_vencto()));
                pstmt.setObject(16, Util.trataNulo(obj.getVl_bruto()));
                pstmt.setObject(17, Util.trataNulo(obj.getCd_cli()));
                pstmt.setObject(18, Util.trataNulo(obj.getDt_ref()));
                pstmt.setObject(19, Util.trataNulo(obj.getNr_nota()));
                pstmt.setObject(20, Util.trataNulo(obj.getCd_emp()));
                pstmt.setObject(21, Util.trataNulo(obj.getOid()));
                ResultSet rs = coloca_contadores(pstmt, 22, p_pagina, p_nr_reg, p_oid,
p_mostra_contadores);
                return this.carregaObjetos(rs, p_oid);
        }

        public void insert(Nota_fiscal obj, Connection con) throws SQLException {
                String sql =
                        " insert into t_nota_fiscal "
                                + "( cd_emp,nr_nota,dt_ref,cd_cli,vl_bruto"
                                + ",dt_vencto,vl_liquido,irff_retido,irff_pagar,cofins_retido"
                                +
",pis_retido,csll_retido,iss_retido,iss_pagar,total_impostos_retidos"
                                + ",cst,Irlp,cfps,contribuicao_social,cd_conta ) "
                                + " values "
                                + "(?,?,to_date(?, 'dd/mm/yyyy'),?,?"
                                + ",to_date(?, 'dd/mm/yyyy'),?,?,?,?"
                                + ",?,?,?,?,?,?"
                                + ",?,?,?,?,?) ";
                PreparedStatement pstmt = con.prepareStatement(sql);
                //
                pstmt.setObject(1, Util.trataNulo(obj.getCd_emp()));
                pstmt.setObject(2, Util.trataNulo(obj.getNr_nota()));
                pstmt.setObject(3, Util.trataNulo(obj.getDt_ref()));
                pstmt.setObject(4, Util.trataNulo(obj.getCd_cli()));
                pstmt.setObject(5, Util.trataNulo(obj.getVl_bruto()));
                pstmt.setObject(6, Util.trataNulo(obj.getDt_vencto()));
                pstmt.setObject(7, Util.trataNulo(obj.getVl_liquido()));
                pstmt.setObject(8, Util.trataNulo(obj.getIrff_retido()));
                pstmt.setObject(9, Util.trataNulo(obj.getIrff_pagar()));
                pstmt.setObject(10, Util.trataNulo(obj.getCofins_retido()));
                pstmt.setObject(11, Util.trataNulo(obj.getPis_retido()));
                pstmt.setObject(12, Util.trataNulo(obj.getCsll_retido()));
                pstmt.setObject(13, Util.trataNulo(obj.getIss_retido()));
                pstmt.setObject(14, Util.trataNulo(obj.getIss_pagar()));
                pstmt.setObject(15, Util.trataNulo(obj.getTotal_impostos_retidos()));
                pstmt.setObject(16, Util.trataNulo(obj.getCst()));
                pstmt.setObject(17, Util.trataNulo(obj.getIrlp()));
                pstmt.setObject(18, Util.trataNulo(obj.getCfps()));
                pstmt.setObject(19, Util.trataNulo(obj.getContribuicao_social()));
                pstmt.setObject(20, Util.trataNulo(obj.getCd_conta()));
                //
                pstmt.executeUpdate();
                pstmt.close();
        }

        public void update(Nota_fiscal obj, Connection con) throws SQLException {
                String sql =
                        " update t_nota_fiscal "
                                + " set "
                                + " dt_ref = to_date(?, 'dd/mm/yyyy')"
                                + ",cd_cli = ?"
                                + ",vl_bruto = ?"
                                + ",dt_vencto = to_date(?, 'dd/mm/yyyy')"
                                + ",vl_liquido = ?"
                                + ",irff_retido = ?"
                                + ",irff_pagar = ?"
                                + ",cofins_retido = ?"
                                + ",pis_retido = ?"
                                + ",csll_retido = ?"
                                + ",iss_retido = ?"
                                + ",iss_pagar = ?"
                                + ",total_impostos_retidos = ?"
                                + ",cst = ?"
                                + ",Irlp = ?"
                                + ",cfps = ?"
                                + ",contribuicao_social = ?"
                                + ",cd_conta = ?"
                                + " where oid = ? ";
                PreparedStatement pstmt = con.prepareStatement(sql);
                pstmt.setObject(1, Util.trataNulo(obj.getDt_ref()));
                pstmt.setObject(2, Util.trataNulo(obj.getCd_cli()));
                pstmt.setObject(3, Util.trataNulo(obj.getVl_bruto()));
                pstmt.setObject(4, Util.trataNulo(obj.getDt_vencto()));
                pstmt.setObject(5, Util.trataNulo(obj.getVl_liquido()));
                pstmt.setObject(6, Util.trataNulo(obj.getIrff_retido()));
```

```
                pstmt.setObject(7, Util.trataNulo(obj.getIrff_pagar()));
                pstmt.setObject(8, Util.trataNulo(obj.getCofins_retido()));
                pstmt.setObject(9, Util.trataNulo(obj.getPis_retido()));
                pstmt.setObject(10, Util.trataNulo(obj.getCsll_retido()));
                pstmt.setObject(11, Util.trataNulo(obj.getIss_retido()));
                pstmt.setObject(12, Util.trataNulo(obj.getIss_pagar()));
                pstmt.setObject(13, Util.trataNulo(obj.getTotal_impostos_retidos()));
                pstmt.setObject(14, Util.trataNulo(obj.getCst()));
                pstmt.setObject(15, Util.trataNulo(obj.getIrlp()));
                pstmt.setObject(16, Util.trataNulo(obj.getCfps()));
                pstmt.setObject(17, Util.trataNulo(obj.getContribuicao_social()));
                pstmt.setObject(18, Util.trataNulo(obj.getCd_conta()));
                pstmt.setObject(19, Util.trataNulo(obj.getOid()));
                pstmt.executeUpdate();
                pstmt.close();
        }

        public void delete(String p_oid, Connection con) throws SQLException {
                String sql = "delete from t_nota_fiscal where oid = ?";
                PreparedStatement pstmt = con.prepareStatement(sql);
                pstmt.setObject(1, Util.trataNulo(p_oid));
                pstmt.executeUpdate();
                pstmt.close();
        }

        public ResultSet coloca_contadores(
                PreparedStatement pstmt,
                int p_posicao_offset,
                int p_pagina,
                int p_nr_reg,
                String p_oid,
                String p_mostra_contadores)
                throws SQLException {
                //Carregar o Contador total
                int v_posicao_limit = p_posicao_offset + 1;
                ArrayList retorno = new ArrayList();
                ResultSet rs;
                if (Util.trataNulo(p_mostra_contadores)==null){
                        pstmt.setInt(p_posicao_offset, 0);
                        pstmt.setInt(v_posicao_limit, 2147483647);
                        rs = pstmt.executeQuery();
                        rs.last();
                        contaSelect = rs.getRow();
                }
                nr_reg = p_nr_reg;
                if (p_pagina > contaSelect) {
                        p_pagina = contaSelect;
                }
                pstmt.setInt(p_posicao_offset, p_pagina);
                pstmt.setInt(v_posicao_limit, nr_reg);
                rs = pstmt.executeQuery();
                rs.last();
                contaSelectAtual = rs.getRow() + p_pagina;
                rs.beforeFirst();
                return rs;
        }
        public int getContaSelect() {
                return contaSelect;
        }
        public int getContaSelectAtual() {
                return contaSelectAtual;
        }
        public String getNr_atual() {
                return nr_atual;
        }
        public int getNr_reg() {
                return nr_reg;
        }
        public void setContaSelect(int i) {
                contaSelect = i;
        }
        public void setContaSelectAtual(int i) {
                contaSelectAtual = i;
        }
        public void setNr_atual(String string) {
                nr_atual = string;
        }
        public void setNr_reg(int i) {
                nr_reg = i;
        }
}
```

Nota_itemSQL.java

```
package br.com.nota_fluxo.sql;

import java.sql.Connection;
import java.sql.PreparedStatement;
```

```
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

import org.apache.log4j.Category;

import br.com.nota_fluxo.bean.Nota_item;
import br.com.nota_fluxo.util.Util;

public class Nota_itemSQL {
        private int contaSelect;
        private int contaSelectAtual;
        private int nr_reg;
        private String nr_atual;

        Category cat = Category.getInstance(Nota_itemSQL.class.getName());

        private ArrayList carregaObjetos(ResultSet rs, String p_oid)
                throws SQLException {
                ArrayList retorno = new ArrayList();
                int nr_reg = 0;
                while (rs.next()) {
                        if (Util.trataNulo(p_oid) != null) {
                                if (Integer.parseInt(p_oid) ==
Integer.parseInt(rs.getString("oid"))) {
                                        nr_atual = nr_reg + "";
                                }
                        } else {
                                Nota_item nota_item = new Nota_item();
                                nota_item.setCd_emp(rs.getString("cd_emp"));
                                nota_item.setNr_nota(rs.getString("nr_nota"));
                                nota_item.setCd_cnae(rs.getString("cd_cnae"));
                                nota_item.setVl_item(rs.getString("vl_item"));
                                nota_item.setOid(rs.getString("oid"));
                                retorno.add(nota_item);
                        }
                        nr_reg++;
                }
                return retorno;
        }

        public ArrayList select(
                Connection con,
                int p_pagina,
                Nota_item obj,
                int p_nr_reg,
                String p_oid,
                String p_mostra_contadores)
                throws SQLException {
                ArrayList retorno = new ArrayList();
                String sql =
                "select "
                        + "  cd_emp"
                        + ",nr_nota"
                        + ",cd_cnae"
                        + ",vl_item "
                        + ", oid "
                        + "from t_nota_item "
                        + "where vl_item                                    = coalesce(?, vl_item) ";
                if (Util.trataNulo(obj.getCd_cnae()) == null) {
                        sql = sql + "and cd_cnae          = coalesce(?, cd_cnae)";
                } else {
                        sql = sql + "and cd_cnae          = ? ";
                }
                if (Util.trataNulo(obj.getNr_nota()) == null) {
                        sql = sql + "and nr_nota          = coalesce(?, nr_nota)";
                } else {
                        sql = sql + "and nr_nota          = ? ";
                }
                if (Util.trataNulo(obj.getCd_emp()) == null) {
                        sql = sql + "and cd_emp           = coalesce(?, cd_emp)";
                } else {
                        sql = sql + "and cd_emp           = ? ";
                }

                if (Util.trataNulo(obj.getOid()) == null) {
                        sql = sql + "and oid                        = coalesce(?, oid) ";
                } else {
                        sql = sql + "and oid                        = ? ";
                }

                sql = sql + "order by 1 " + "offset ? " + "limit ? ";
                PreparedStatement pstmt = con.prepareStatement(sql);
                // Coloca os parâmetros
                pstmt.setObject(1, Util.trataNulo(obj.getVl_item()));
                pstmt.setObject(2, Util.trataNulo(obj.getCd_cnae()));
                pstmt.setObject(3, Util.trataNulo(obj.getNr_nota()));
                pstmt.setObject(4, Util.trataNulo(obj.getCd_emp()));
```

```
                pstmt.setObject(5, Util.trataNulo(obj.getOid()));
                ResultSet rs = coloca_contadores(pstmt, 6, p_pagina, p_nr_reg, p_oid,
p_mostra_contadores);
                return this.carregaObjetos(rs, p_oid);
        }

        public void insert(Nota_item obj, Connection con) throws SQLException {
                String sql =
                        " insert into t_nota_item "
                                + "( cd_emp,nr_nota,cd_cnae,vl_item ) "
                                + " values "
                                + "(?,?,?,?) ";
                PreparedStatement pstmt = con.prepareStatement(sql);
                //
                pstmt.setObject(1, Util.trataNulo(obj.getCd_emp()));
                pstmt.setObject(2, Util.trataNulo(obj.getNr_nota()));
                pstmt.setObject(3, Util.trataNulo(obj.getCd_cnae()));
                pstmt.setObject(4, Util.trataNulo(obj.getVl_item()));
                //
                pstmt.executeUpdate();
                pstmt.close();
        }

        public void update(Nota_item obj, Connection con) throws SQLException {
                String sql =
                        " update t_nota_item "
                                + " set "
                                + " vl_item = ?"
                                + " where oid = ? ";
                PreparedStatement pstmt = con.prepareStatement(sql);
                pstmt.setObject(1, Util.trataNulo(obj.getVl_item()));
                pstmt.setObject(2, Util.trataNulo(obj.getOid()));
                pstmt.executeUpdate();
                pstmt.close();
        }

        public void delete(String p_oid, Connection con) throws SQLException {
                String sql = "delete from t_nota_item where oid = ?";
                PreparedStatement pstmt = con.prepareStatement(sql);
                pstmt.setObject(1, Util.trataNulo(p_oid));
                pstmt.executeUpdate();
                pstmt.close();
        }

        public ResultSet coloca_contadores(
                PreparedStatement pstmt,
                int p_posicao_offset,
                int p_pagina,
                int p_nr_reg,
                String p_oid,
                String p_mostra_contadores)
                throws SQLException {
                //Carregar o Contador total
                int v_posicao_limit = p_posicao_offset + 1;
                ArrayList retorno = new ArrayList();
                ResultSet rs;
                if (Util.trataNulo(p_mostra_contadores)==null){
                        pstmt.setInt(p_posicao_offset, 0);
                        pstmt.setInt(v_posicao_limit, 2147483647);
                        rs = pstmt.executeQuery();
                        rs.last();
                        contaSelect = rs.getRow();
                }
                nr_reg = p_nr_reg;
                if (p_pagina > contaSelect) {
                        p_pagina = contaSelect;
                }
                pstmt.setInt(p_posicao_offset, p_pagina);
                pstmt.setInt(v_posicao_limit, nr_reg);
                rs = pstmt.executeQuery();
                rs.last();
                contaSelectAtual = rs.getRow() + p_pagina;
                rs.beforeFirst();
                return rs;
        }
        public int getContaSelect() {
                return contaSelect;
        }
        public int getContaSelectAtual() {
                return contaSelectAtual;
        }
        public String getNr_atual() {
                return nr_atual;
        }
        public int getNr_reg() {
                return nr_reg;
        }
        public void setContaSelect(int i) {
```

```
                contaSelect = i;
        }
        public void setContaSelectAtual(int i) {
                contaSelectAtual = i;
        }
        public void setNr_atual(String string) {
                nr_atual = string;
        }
        public void setNr_reg(int i) {
                nr_reg = i;
        }
}
```

Ramo_AtividadeSQL.java

```
package br.com.nota_fluxo.sql;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

import org.apache.log4j.Category;

import br.com.nota_fluxo.bean.Ramo_atividade;
import br.com.nota_fluxo.util.Util;

public class Ramo_atividadeSQL {
        private int contaSelect;
        private int contaSelectAtual;
        private int nr_reg;
        private String nr_atual;

        Category cat = Category.getInstance(Ramo_atividadeSQL.class.getName());

        private ArrayList carregaObjetos(ResultSet rs, String p_oid)
                throws SQLException {
                ArrayList retorno = new ArrayList();
                int nr_reg = 0;
                while (rs.next()) {
                        if (Util.trataNulo(p_oid) != null) {
                                if (Integer.parseInt(p_oid) ==
Integer.parseInt(rs.getString("oid"))) {
                                        nr_atual = nr_reg + "";
                                }
                        } else {
                                Ramo_atividade ramo_atividade = new Ramo_atividade();
                                ramo_atividade.setCd_atividade(rs.getString("cd_atividade"));
                                ramo_atividade.setDs_atividade(rs.getString("ds_atividade"));
                                ramo_atividade.setOid(rs.getString("oid"));
                                retorno.add(ramo_atividade);
                        }
                        nr_reg++;
                }
                return retorno;
        }

        public ArrayList select(
                Connection con,
                int p_pagina,
                Ramo_atividade obj,
                int p_nr_reg,
                String p_oid,
                String p_mostra_contadores)
                throws SQLException {
                ArrayList retorno = new ArrayList();
                String sql =
                "select "
                + "  cd_atividade" +
                        ",ds_atividade"
                + ", oid "
                + "from t_ramo_atividade "
                + "where ds_atividade                                        = coalesce(?,
ds_atividade) ";
                if (Util.trataNulo(obj.getCd_atividade()) == null) {
                        sql = sql + "and cd_atividade              = coalesce(?, cd_atividade)";
                } else {
                        sql = sql + "and cd_atividade              = ? ";
                }

                if (Util.trataNulo(obj.getOid()) == null) {
                        sql = sql + "and oid                       = coalesce(?, oid) ";
                } else {
                        sql = sql + "and oid                       = ? ";
                }

                sql = sql + "order by 1 " + "offset ? " + "limit ? ";
```

```
                PreparedStatement pstmt = con.prepareStatement(sql);
                // Coloca os parâmetros
                pstmt.setObject(1, Util.trataNulo(obj.getDs_atividade()));
                pstmt.setObject(2, Util.trataNulo(obj.getCd_atividade()));
                pstmt.setObject(3, Util.trataNulo(obj.getOid()));
                ResultSet rs = coloca_contadores(pstmt, 4, p_pagina, p_nr_reg, p_oid,
p_mostra_contadores);
                return this.carregaObjetos(rs, p_oid);
        }

        public void insert(Ramo_atividade obj, Connection con) throws SQLException {
                String sql =
                        " insert into t_ramo_atividade "
                                + "( cd_atividade,ds_atividade ) "
                                + " values "
                                + "(?,?) ";
                PreparedStatement pstmt = con.prepareStatement(sql);
                //
                pstmt.setObject(1, Util.trataNulo(obj.getCd_atividade()));
                pstmt.setObject(2, Util.trataNulo(obj.getDs_atividade()));
                //
                pstmt.executeUpdate();
                pstmt.close();
        }

        public void update(Ramo_atividade obj, Connection con) throws SQLException {
                String sql =
                        " update t_ramo_atividade "
                                + " set "
                                + " ds_atividade = ?"
                                + " where oid = ? ";
                PreparedStatement pstmt = con.prepareStatement(sql);
                pstmt.setObject(1, Util.trataNulo(obj.getDs_atividade()));
                pstmt.setObject(2, Util.trataNulo(obj.getOid()));
                pstmt.executeUpdate();
                pstmt.close();
        }

        public void delete(String p_oid, Connection con) throws SQLException {
                String sql = "delete from t_ramo_atividade where oid = ?";
                PreparedStatement pstmt = con.prepareStatement(sql);
                pstmt.setObject(1, Util.trataNulo(p_oid));
                pstmt.executeUpdate();
                pstmt.close();
        }

        public ResultSet coloca_contadores(
                PreparedStatement pstmt,
                int p_posicao_offset,
                int p_pagina,
                int p_nr_reg,
                String p_oid,
                String p_mostra_contadores)
                throws SQLException {
                //Carregar o Contador total
                int v_posicao_limit = p_posicao_offset + 1;
                ArrayList retorno = new ArrayList();
                ResultSet rs;
                if (Util.trataNulo(p_mostra_contadores)==null){
                        pstmt.setInt(p_posicao_offset, 0);
                        pstmt.setInt(v_posicao_limit, 2147483647);
                        rs = pstmt.executeQuery();
                        rs.last();
                        contaSelect = rs.getRow();
                }
                nr_reg = p_nr_reg;
                if (p_pagina > contaSelect) {
                        p_pagina = contaSelect;
                }
                pstmt.setInt(p_posicao_offset, p_pagina);
                pstmt.setInt(v_posicao_limit, nr_reg);
                rs = pstmt.executeQuery();
                rs.last();
                contaSelectAtual = rs.getRow() + p_pagina;
                rs.beforeFirst();
                return rs;
        }
        public int getContaSelect() {
                return contaSelect;
        }
        public int getContaSelectAtual() {
                return contaSelectAtual;
        }
        public String getNr_atual() {
                return nr_atual;
        }
        public int getNr_reg() {
                return nr_reg;
```

```
        }
        public void setContaSelect(int i) {
                contaSelect = i;
        }
        public void setContaSelectAtual(int i) {
                contaSelectAtual = i;
        }
        public void setNr_atual(String string) {
                nr_atual = string;
        }
        public void setNr_reg(int i) {
                nr_reg = i;
        }
}
```

Tipo_contas_pagarSQL.java

```
package br.com.nota_fluxo.sql;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

import org.apache.log4j.Category;

import br.com.nota_fluxo.bean.Tipo_contas_pagar;
import br.com.nota_fluxo.util.Util;

public class Tipo_contas_pagarSQL {
        private int contaSelect;
        private int contaSelectAtual;
        private int nr_reg;
        private String nr_atual;

        Category cat = Category.getInstance(Tipo_contas_pagarSQL.class.getName());

        private ArrayList carregaObjetos(ResultSet rs, String p_oid)
                throws SQLException {
                ArrayList retorno = new ArrayList();
                int nr_reg = 0;
                while (rs.next()) {
                        if (Util.trataNulo(p_oid) != null) {
                                if (Integer.parseInt(p_oid) ==
Integer.parseInt(rs.getString("oid"))) {
                                        nr_atual = nr_reg + "";
                                }
                        } else {
                                Tipo_contas_pagar tipo_contas_pagar = new Tipo_contas_pagar();
                                tipo_contas_pagar.setCd_tp_pagar(rs.getString("cd_tp_pagar"));
                                tipo_contas_pagar.setDs_tp_pagar(rs.getString("ds_tp_pagar"));
                                tipo_contas_pagar.setOid(rs.getString("oid"));
                                retorno.add(tipo_contas_pagar);
                        }
                        nr_reg++;
                }
                return retorno;
        }

        public ArrayList select(
                Connection con,
                int p_pagina,
                Tipo_contas_pagar obj,
                int p_nr_reg,
                String p_oid,
                String p_mostra_contadores)
                throws SQLException {
                ArrayList retorno = new ArrayList();
                String sql =
                "select "
                + "  cd_tp_pagar" +
                        ",ds_tp_pagar"
                + ", oid "
                + "from t_tipo_contas_pagar "
                + "where ds_tp_pagar                                    = coalesce(?,
ds_tp_pagar) ";
                if (Util.trataNulo(obj.getDs_tp_pagar()) == null) {
                        sql = sql + "and cd_tp_pagar              = coalesce(?, cd_tp_pagar)";
                } else {
                        sql = sql + "and cd_tp_pagar              = ? ";
                }

                if (Util.trataNulo(obj.getOid()) == null) {
                        sql = sql + "and oid                      = coalesce(?, oid) ";
                } else {
                        sql = sql + "and oid                      = ? ";
                }
```

```
                sql = sql + "order by 1 " + "offset ? " + "limit ? ";
                PreparedStatement pstmt = con.prepareStatement(sql);
                // Coloca os parâmetros
                pstmt.setObject(1, Util.trataNulo(obj.getDs_tp_pagar()));
                pstmt.setObject(2, Util.trataNulo(obj.getCd_tp_pagar()));
                pstmt.setObject(3, Util.trataNulo(obj.getOid()));
                ResultSet rs = coloca_contadores(pstmt, 4, p_pagina, p_nr_reg, p_oid,
p_mostra_contadores);
                return this.carregaObjetos(rs, p_oid);
        }

        public void insert(Tipo_contas_pagar obj, Connection con) throws SQLException {
                String sql =
                        " insert into t_tipo_contas_pagar "
                                + "( cd_tp_pagar,ds_tp_pagar ) "
                                + " values "
                                + "(?,?) ";
                PreparedStatement pstmt = con.prepareStatement(sql);
                //
                pstmt.setObject(1, Util.trataNulo(obj.getCd_tp_pagar()));
                pstmt.setObject(2, Util.trataNulo(obj.getDs_tp_pagar()));
                //
                pstmt.executeUpdate();
                pstmt.close();
        }

        public void update(Tipo_contas_pagar obj, Connection con) throws SQLException {
                String sql =
                        " update t_tipo_contas_pagar "
                                + " set "
                                + " ds_tp_pagar = ?"
                                + " where oid = ? ";
                PreparedStatement pstmt = con.prepareStatement(sql);
                pstmt.setObject(1, Util.trataNulo(obj.getDs_tp_pagar()));
                pstmt.setObject(2, Util.trataNulo(obj.getOid()));
                pstmt.executeUpdate();
                pstmt.close();
        }

        public void delete(String p_oid, Connection con) throws SQLException {
                String sql = "delete from t_tipo_contas_pagar where oid = ?";
                PreparedStatement pstmt = con.prepareStatement(sql);
                pstmt.setObject(1, Util.trataNulo(p_oid));
                pstmt.executeUpdate();
                pstmt.close();
        }

        public ResultSet coloca_contadores(
                PreparedStatement pstmt,
                int p_posicao_offset,
                int p_pagina,
                int p_nr_reg,
                String p_oid,
                String p_mostra_contadores)
                throws SQLException {
                //Carregar o Contador total
                int v_posicao_limit = p_posicao_offset + 1;
                ArrayList retorno = new ArrayList();
                ResultSet rs;
                if (Util.trataNulo(p_mostra_contadores)==null){
                        pstmt.setInt(p_posicao_offset, 0);
                        pstmt.setInt(v_posicao_limit, 2147483647);
                        rs = pstmt.executeQuery();
                        rs.last();
                        contaSelect = rs.getRow();
                }
                nr_reg = p_nr_reg;
                if (p_pagina > contaSelect) {
                        p_pagina = contaSelect;
                }
                pstmt.setInt(p_posicao_offset, p_pagina);
                pstmt.setInt(v_posicao_limit, nr_reg);
                rs = pstmt.executeQuery();
                rs.last();
                contaSelectAtual = rs.getRow() + p_pagina;
                rs.beforeFirst();
                return rs;
        }
        public int getContaSelect() {
                return contaSelect;
        }
        public int getContaSelectAtual() {
                return contaSelectAtual;
        }
        public String getNr_atual() {
                return nr_atual;
        }
```

```
        public int getNr_reg() {
                return nr_reg;
        }
        public void setContaSelect(int i) {
                contaSelect = i;
        }
        public void setContaSelectAtual(int i) {
                contaSelectAtual = i;
        }
        public void setNr_atual(String string) {
                nr_atual = string;
        }
        public void setNr_reg(int i) {
                nr_reg = i;
        }
}
```

Tipo_contas_receberSQL.java

```
package br.com.nota_fluxo.sql;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

import org.apache.log4j.Category;

import br.com.nota_fluxo.bean.Tipo_contas_receber;
import br.com.nota_fluxo.util.Util;

public class Tipo_contas_receberSQL {
        private int contaSelect;
        private int contaSelectAtual;
        private int nr_reg;
        private String nr_atual;

        Category cat = Category.getInstance(Tipo_contas_receberSQL.class.getName());

        private ArrayList carregaObjetos(ResultSet rs, String p_oid)
                throws SQLException {
                ArrayList retorno = new ArrayList();
                int nr_reg = 0;
                while (rs.next()) {
                        if (Util.trataNulo(p_oid) != null) {
                                if (Integer.parseInt(p_oid) ==
Integer.parseInt(rs.getString("oid"))) {
                                        nr_atual = nr_reg + "";
                                }
                        } else {
                                Tipo_contas_receber tipo_contas_receber = new
Tipo_contas_receber();

        tipo_contas_receber.setCd_tp_receber(rs.getString("cd_tp_receber"));

        tipo_contas_receber.setDs_tp_receber(rs.getString("ds_tp_receber"));
                                tipo_contas_receber.setOid(rs.getString("oid"));
                                retorno.add(tipo_contas_receber);
                        }
                        nr_reg++;
                }
                return retorno;
        }

        public ArrayList select(
                Connection con,
                int p_pagina,
                Tipo_contas_receber obj,
                int p_nr_reg,
                String p_oid,
                String p_mostra_contadores)
                throws SQLException {
                ArrayList retorno = new ArrayList();
                String sql =
                "select "
                + "  cd_tp_receber" +
                        ",ds_tp_receber"
                + ", oid "
                + "from t_tipo_contas_receber "
                + "where ds_tp_receber                                  = coalesce(?,
ds_tp_receber) ";
                if (Util.trataNulo(obj.getDs_tp_receber()) == null) {
                        sql = sql + "and cd_tp_receber            = coalesce(?, cd_tp_receber)";
                } else {
                        sql = sql + "and cd_tp_receber            = ? ";
                }
```

```
                if (Util.trataNulo(obj.getOid()) == null) {
                        sql = sql + "and oid                        = coalesce(?, oid) ";
                } else {
                        sql = sql + "and oid                        = ? ";
                }

                sql = sql + "order by 1 " + "offset ? " + "limit ? ";
                PreparedStatement pstmt = con.prepareStatement(sql);
                // Coloca os parâmetros
                pstmt.setObject(1, Util.trataNulo(obj.getDs_tp_receber()));
                pstmt.setObject(2, Util.trataNulo(obj.getCd_tp_receber()));
                pstmt.setObject(3, Util.trataNulo(obj.getOid()));
                ResultSet rs = coloca_contadores(pstmt, 4, p_pagina, p_nr_reg, p_oid,
p_mostra_contadores);
                return this.carregaObjetos(rs, p_oid);
        }

        public void insert(Tipo_contas_receber obj, Connection con) throws SQLException {
                String sql =
                        " insert into t_tipo_contas_receber "
                                + "( cd_tp_receber,ds_tp_receber ) "
                                + " values "
                                + "(?,?) ";
                PreparedStatement pstmt = con.prepareStatement(sql);
                //
                pstmt.setObject(1, Util.trataNulo(obj.getCd_tp_receber()));
                pstmt.setObject(2, Util.trataNulo(obj.getDs_tp_receber()));
                //
                pstmt.executeUpdate();
                pstmt.close();
        }

        public void update(Tipo_contas_receber obj, Connection con) throws SQLException {
                String sql =
                        " update t_tipo_contas_receber "
                                + " set "
                                + " ds_tp_receber = ?"
                                + " where oid = ? ";
                PreparedStatement pstmt = con.prepareStatement(sql);
                pstmt.setObject(1, Util.trataNulo(obj.getDs_tp_receber()));
                pstmt.setObject(2, Util.trataNulo(obj.getOid()));
                pstmt.executeUpdate();
                pstmt.close();
        }

        public void delete(String p_oid, Connection con) throws SQLException {
                String sql = "delete from t_tipo_contas_receber where oid = ?";
                PreparedStatement pstmt = con.prepareStatement(sql);
                pstmt.setObject(1, Util.trataNulo(p_oid));
                pstmt.executeUpdate();
                pstmt.close();
        }

        public ResultSet coloca_contadores(
                PreparedStatement pstmt,
                int p_posicao_offset,
                int p_pagina,
                int p_nr_reg,
                String p_oid,
                String p_mostra_contadores)
                throws SQLException {
                //Carregar o Contador total
                int v_posicao_limit = p_posicao_offset + 1;
                ArrayList retorno = new ArrayList();
                ResultSet rs;
                if (Util.trataNulo(p_mostra_contadores)==null){
                        pstmt.setInt(p_posicao_offset, 0);
                        pstmt.setInt(v_posicao_limit, 2147483647);
                        rs = pstmt.executeQuery();
                        rs.last();
                        contaSelect = rs.getRow();
                }
                nr_reg = p_nr_reg;
                if (p_pagina > contaSelect) {
                        p_pagina = contaSelect;
                }
                pstmt.setInt(p_posicao_offset, p_pagina);
                pstmt.setInt(v_posicao_limit, nr_reg);
                rs = pstmt.executeQuery();
                rs.last();
                contaSelectAtual = rs.getRow() + p_pagina;
                rs.beforeFirst();
                return rs;
        }
        public int getContaSelect() {
                return contaSelect;
        }
        public int getContaSelectAtual() {
```

```
                return contaSelectAtual;
        }
        public String getNr_atual() {
                return nr_atual;
        }
        public int getNr_reg() {
                return nr_reg;
        }
        public void setContaSelect(int i) {
                contaSelect = i;
        }
        public void setContaSelectAtual(int i) {
                contaSelectAtual = i;
        }
        public void setNr_atual(String string) {
                nr_atual = string;
        }
        public void setNr_reg(int i) {
                nr_reg = i;
        }
}
```

UsuárioSQL.java

```
package br.com.nota_fluxo.sql;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

import org.apache.log4j.Category;

import br.com.nota_fluxo.bean.Usuario;
import br.com.nota_fluxo.util.Util;

public class UsuarioSQL {
        private int contaSelect;
        private int contaSelectAtual;
        private int nr_reg;
        private String nr_atual;

        Category cat = Category.getInstance(UsuarioSQL.class.getName());

        private ArrayList carregaObjetos(ResultSet rs, String p_oid)
                throws SQLException {
                ArrayList retorno = new ArrayList();
                int nr_reg = 0;
                while (rs.next()) {
                        if (Util.trataNulo(p_oid) != null) {
                                if (Integer.parseInt(p_oid) ==
Integer.parseInt(rs.getString("oid"))) {
                                        nr_atual = nr_reg + "";
                                }
                        } else {
                                Usuario usuario = new Usuario();
                                usuario.setNm_usuario(rs.getString("nm_usuario"));
                                usuario.setDs_senha(rs.getString("ds_senha"));
                                usuario.setCd_emp(rs.getString("cd_emp"));
                                usuario.setOid(rs.getString("oid"));
                                retorno.add(usuario);
                        }
                        nr_reg++;
                }
                return retorno;
        }

        public ArrayList select(
                Connection con,
                int p_pagina,
                Usuario obj,
                int p_nr_reg,
                String p_oid,
                String p_mostra_contadores)
                throws SQLException {
                ArrayList retorno = new ArrayList();
                String sql =
                "select "
                        + " nm_usuario "
                        + ",ds_senha "
                        + ",cd_emp "
                        + ", oid "
                        + "from t_usuario "
                        + "where cd_emp  = coalesce(?, cd_emp) "
                        + "and ds_senha  = coalesce(?, ds_senha) ";
                if (Util.trataNulo(obj.getNm_usuario()) == null) {
                        sql = sql + "and nm_usuario               = coalesce(?, nm_usuario)";
```

```
                } else {
                        sql = sql + "and nm_usuario                = ? ";
                }

                if (Util.trataNulo(obj.getOid()) == null) {
                        sql = sql + "and oid                       = coalesce(?, oid) ";
                } else {
                        sql = sql + "and oid                       = ? ";
                }

                sql = sql + "order by 1 " + "offset ? " + "limit ? ";
                PreparedStatement pstmt = con.prepareStatement(sql);
                // Coloca os parâmetros
                pstmt.setObject(1, Util.trataNulo(obj.getCd_emp()));
                pstmt.setObject(2, Util.trataNulo(obj.getDs_senha()));
                pstmt.setObject(3, Util.trataNulo(obj.getNm_usuario()));
                pstmt.setObject(4, Util.trataNulo(obj.getOid()));
                ResultSet rs = coloca_contadores(pstmt, 5, p_pagina, p_nr_reg, p_oid,
p_mostra_contadores);
                return this.carregaObjetos(rs, p_oid);
        }

        public void insert(Usuario obj, Connection con) throws SQLException {
                String sql =
                        " insert into t_usuario "
                                + "( nm_usuario,ds_senha,cd_emp ) "
                                + " values "
                                + "(?,?,?) ";
                PreparedStatement pstmt = con.prepareStatement(sql);
                //
                pstmt.setObject(1, Util.trataNulo(obj.getNm_usuario()));
                pstmt.setObject(2, Util.trataNulo(obj.getDs_senha()));
                pstmt.setObject(3, Util.trataNulo(obj.getCd_emp()));
                //
                pstmt.executeUpdate();
                pstmt.close();
        }

        public void update(Usuario obj, Connection con) throws SQLException {
                String sql =
                        " update t_usuario "
                                + " set "
                                + " ds_senha = ?"
                                + " where oid = ? ";
                PreparedStatement pstmt = con.prepareStatement(sql);
                pstmt.setObject(1, Util.trataNulo(obj.getDs_senha()));
                pstmt.setObject(2, Util.trataNulo(obj.getOid()));
                pstmt.executeUpdate();
                pstmt.close();
        }

        public void delete(String p_oid, Connection con) throws SQLException {
                String sql = "delete from t_usuario where oid = ?";
                PreparedStatement pstmt = con.prepareStatement(sql);
                pstmt.setObject(1, Util.trataNulo(p_oid));
                pstmt.executeUpdate();
                pstmt.close();
        }

        public ResultSet coloca_contadores(
                PreparedStatement pstmt,
                int p_posicao_offset,
                int p_pagina,
                int p_nr_reg,
                String p_oid,
                String p_mostra_contadores)
                throws SQLException {
                //Carregar o Contador total
                int v_posicao_limit = p_posicao_offset + 1;
                ArrayList retorno = new ArrayList();
                ResultSet rs;
                if (Util.trataNulo(p_mostra_contadores)==null){
                        pstmt.setInt(p_posicao_offset, 0);
                        pstmt.setInt(v_posicao_limit, 2147483647);
                        rs = pstmt.executeQuery();
                        rs.last();
                        contaSelect = rs.getRow();
                }
                nr_reg = p_nr_reg;
                if (p_pagina > contaSelect) {
                        p_pagina = contaSelect;
                }
                pstmt.setInt(p_posicao_offset, p_pagina);
                pstmt.setInt(v_posicao_limit, nr_reg);
                rs = pstmt.executeQuery();
                rs.last();
                contaSelectAtual = rs.getRow() + p_pagina;
                rs.beforeFirst();
```

```
                return rs;
        }
        public int getContaSelect() {
                return contaSelect;
        }
        public int getContaSelectAtual() {
                return contaSelectAtual;
        }
        public String getNr_atual() {
                return nr_atual;
        }
        public int getNr_reg() {
                return nr_reg;
        }
        public void setContaSelect(int i) {
                contaSelect = i;
        }
        public void setContaSelectAtual(int i) {
                contaSelectAtual = i;
        }
        public void setNr_atual(String string) {
                nr_atual = string;
        }
        public void setNr_reg(int i) {
                nr_reg = i;
        }
}
```

ClienteAction.java

```
package br.com.nota_fluxo.action;

import java.sql.Connection;
import java.util.HashMap;
import java.util.Map;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.log4j.Category;

import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;
import org.apache.struts.action.ActionMessages;
import org.apache.struts.actions.LookupDispatchAction;

import br.com.nota_fluxo.bean.Cliente;
import br.com.nota_fluxo.form.ClienteForm;
import br.com.nota_fluxo.sql.ClienteSQL;
import br.com.nota_fluxo.util.Util;

public class ClienteAction extends LookupDispatchAction {

        public Category cat = Category.getInstance(ClienteAction.class.getName());

        protected Map getKeyMethodMap(){
                Map map = new HashMap();
                map.put("nota_fluxo.entra_inserir","entra_inserir");
                map.put("nota_fluxo.entra_consultar","entra_consultar");
                map.put("nota_fluxo.entra_alterar","entra_alterar");
                map.put("nota_fluxo.inserir","inserir");
                map.put("nota_fluxo.apagar","apagar");
                map.put("nota_fluxo.alterar","alterar");
                map.put("nota_fluxo.lista","lista");
                map.put("nota_fluxo.login.valida","lista");
                return map;
        }

        public ActionForward unspecified(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                return (lista(mapping, form, request, response));
        }

        public ActionForward lista(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                ActionMessages msgs = new ActionMessages();

                Connection con =
```

```
                (Connection) request.getSession().getAttribute("conexao");

        //UsuarioForm usuario_form =
        //      (UsuarioForm) request.getSession().getAttribute("usuarioForm");
        //cat.warn("cliente "+usuario_form.getUsuario().getCd_emp());

        ClienteForm beanForm = (ClienteForm) form;
        ClienteSQL beanSQL = new ClienteSQL();
        try {
                beanForm.setNr_reg(1); //Numero de registros da lista
                beanForm.setMostraConsulta("N");

                int p_pagina = 0;
                if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                        p_pagina = 0;
                } else {
                        p_pagina =
                                (int) Integer.parseInt(request.getParameter("p_pagina"));
                }
                beanForm.setCliente_lista(
                        beanSQL.select(
                                con,
                                p_pagina,
                                beanForm.getCliente_pesq(),
                                beanForm.getNr_reg(),
                                "", ""));
                beanForm.setContadoresPaginas(
                        beanSQL.getContaSelect(),
                        beanSQL.getContaSelectAtual());
        } catch (Exception e) {
                msgs.add(
                        ActionMessages.GLOBAL_MESSAGE,
                        new ActionMessage("error.sql", e));
        }
        if (!(msgs.isEmpty())) {
                saveMessages(request, msgs);
        }
        return (mapping.findForward("lista"));
}

public ActionForward entra_consultar(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) {
        ClienteForm beanForm = (ClienteForm) form;
        beanForm.setMostraConsulta("S");
        return (mapping.findForward("editar"));
}

public ActionForward inserir(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
        ActionMessages msgs = new ActionMessages();
        Connection con =
                (Connection) request.getSession().getAttribute("conexao");
        ClienteForm beanForm = (ClienteForm) form;
        ClienteSQL beanSQL = new ClienteSQL();
        msgs = beanForm.validate(mapping, request);
        if (msgs.isEmpty()) {
                try {
                        int p_pagina = 0;

                        if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                                p_pagina = 0;
                        } else {
                                p_pagina =
                                        (int) Integer.parseInt(
                                                request.getParameter("p_pagina"));
                        }
                        String oid = "-1";
                        if (Util.trataNulo(request.getParameter("oid")) != null) {
                                oid = request.getParameter("oid");
                        }

                        Cliente v_cliente = new Cliente();
                        v_cliente.setCd_cli(beanForm.getCliente().getCd_cli());
                        beanForm.setCliente_lista(
                                beanSQL.select(con, 0, v_cliente, 1, "", "N"));
                        if (beanForm.getCliente_lista().size() > 0) {
                                throw new ExcecaoGeral(
                                        new ActionMessage("nota_fluxo.inserir.erro"));
                        } else {
                                beanSQL.insert(beanForm.getCliente(), con);
                                con.commit();
```

```
                                        beanForm.setCliente_lista(

        beanSQL.select(con,0,beanForm.getCliente(),1,"","N"));
                                        beanForm.setCliente_pesq(new Cliente());
                                        beanForm.setCliente_lista(
                                                beanSQL.select(
                                                        con,
                                                        0,
                                                        beanForm.getCliente_pesq(),
                                                        2000000000,
                                                        ((Cliente)
beanForm.getCliente_lista().get(0))
                                                                .getOid(), "N"));
                                        //
                                        beanForm.setNr_reg(1);
                                        p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
                                        beanForm.setCliente_lista(
                                                beanSQL.select(
                                                        con,
                                                        p_pagina,
                                                        beanForm.getCliente_pesq(),
                                                        beanForm.getNr_reg(),
                                                        "", ""));
                                        beanForm.setContadoresPaginas(
                                                beanSQL.getContaSelect(),
                                                beanSQL.getContaSelectAtual());
                                }
                        } catch (ExcecaoGeral e) {
                                msgs.add(ActionMessages.GLOBAL_MESSAGE, e.getAction_message());
                        } catch (Exception e) {
                                msgs.add(
                                        ActionMessages.GLOBAL_MESSAGE,
                                        new ActionMessage("error.sql", e.toString()));
                        }
                }
                if (!(msgs.isEmpty())) {
                        con.rollback();
                        saveMessages(request, msgs);
                }
                return (mapping.findForward("lista"));
        }

        public ActionForward entra_inserir(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                ClienteForm beanForm = (ClienteForm) form;
                beanForm.setCliente(new Cliente());
                beanForm.setMostraConsulta("N");
                return (mapping.findForward("editar"));
        }

        public ActionForward apagar(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                ActionMessages msgs = new ActionMessages();
                Connection con =
                        (Connection) request.getSession().getAttribute("conexao");
                ClienteForm beanForm = (ClienteForm) form;
                ClienteSQL beanSQL = new ClienteSQL();
                try {
                        int p_pagina = 0;
                        if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                                p_pagina = 0;
                        } else {
                                p_pagina =
                                        (int) Integer.parseInt(request.getParameter("p_pagina"));
                        }

                        String oid = "-1";
                        if (Util.trataNulo(request.getParameter("oid")) != null) {
                                oid = request.getParameter("oid");
                        }

                        Cliente v_cliente = new Cliente();
                        v_cliente.setOid(oid);
                        v_cliente =
                                (Cliente) beanSQL.select(con, 0, v_cliente, 1, "","N").get(0);

                        beanForm.setCliente_lista(
                                beanSQL.select(
                                        con,
                                        0,
```

```
                                beanForm.getCliente_pesq(),
                                2000000000,
                                oid, ""));
                        p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());

                        beanSQL.delete(oid, con);
                        con.commit();

                        request.setAttribute("oid", "");
                } catch (Exception e) {
                        msgs.add(
                                ActionMessages.GLOBAL_MESSAGE,
                                new ActionMessage("error.sql", e.toString()));
                }
                if (!(msgs.isEmpty())) {
                        con.rollback();
                        saveMessages(request, msgs);
                }
                return (lista(mapping, beanForm, request, response));
        }

        public ActionForward entra_alterar(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                ActionMessages msgs = new ActionMessages();
                Connection con =
                        (Connection) request.getSession().getAttribute("conexao");
                ClienteForm beanForm = (ClienteForm) form;
                try {
                        ClienteSQL beanSQL = new ClienteSQL();

                        String oid = "-1";
                        if (Util.trataNulo(request.getParameter("oid")) != null) {
                                oid = request.getParameter("oid");
                        }

                        Cliente v_cliente = new Cliente();
                        v_cliente.setOid(oid);
                        beanForm.setCliente_lista(beanSQL.select(con,0,v_cliente,1,"","N"));
                        beanForm.setCliente((Cliente) beanForm.getCliente_lista().get(0));
                } catch (Exception e) {
                        msgs.add(
                                ActionMessages.GLOBAL_MESSAGE,
                                new ActionMessage("error.sql", e.toString()));
                }
                if (!(msgs.isEmpty())) {
                        con.rollback();
                        saveMessages(request, msgs);
                }
                return (mapping.findForward("editar"));
        }

        public ActionForward alterar(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                ActionMessages msgs = new ActionMessages();
                Connection con =
                        (Connection) request.getSession().getAttribute("conexao");
                ClienteForm beanForm = (ClienteForm) form;
                ClienteSQL beanSQL = new ClienteSQL();
                msgs = beanForm.validate(mapping, request);
                if (msgs.isEmpty()) {
                        try {
                                int p_pagina = 0;
                                if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                                        p_pagina = 0;
                                } else {
                                        p_pagina =
                                                (int) Integer.parseInt(
                                                        request.getParameter("p_pagina"));
                                }

                                String oid = "-1";
                                if (Util.trataNulo(request.getParameter("oid")) != null) {
                                        oid = request.getParameter("oid");
                                }

                                beanSQL.update(beanForm.getCliente(), con);
                                con.commit();

                                beanForm.setCliente_lista(
                                        beanSQL.select(con, 0, beanForm.getCliente(), 1, "", "N"));
```

```
                                if (beanForm.getCliente_lista().size() == 0) {
                                        throw new ExcecaoGeral(
                                                new ActionMessage("nota_fluxo.alterar.erro"));
                                }

                                beanForm.setCliente_lista(
                                        beanSQL.select(
                                                con,
                                                0,
                                                beanForm.getCliente_pesq(),
                                                2000000000,
                                                beanForm.getCliente().getOid(), "N"));


                                //
                                beanForm.setNr_reg(1);
                                p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
                                beanForm.setCliente_lista(
                                        beanSQL.select(
                                                con,
                                                p_pagina,
                                                beanForm.getCliente_pesq(),
                                                beanForm.getNr_reg(),
                                                "",""));
                                beanForm.setContadoresPaginas(
                                        beanSQL.getContaSelect(),
                                        beanSQL.getContaSelectAtual());


                        } catch (ExcecaoGeral e) {
                                msgs.add(ActionMessages.GLOBAL_MESSAGE, e.getAction_message());
                        } catch (Exception e) {
                                msgs.add(
                                        ActionMessages.GLOBAL_MESSAGE,
                                        new ActionMessage("error.sql", e.toString()));
                        }
                }
                if (!(msgs.isEmpty())) {
                        con.rollback();
                        saveMessages(request, msgs);
                }
                return (mapping.findForward("lista"));
        }
}
```

CnaeAction.java

```
package br.com.nota_fluxo.action;

import java.sql.Connection;
import java.util.HashMap;
import java.util.Map;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.log4j.Category;

import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;
import org.apache.struts.action.ActionMessages;
import org.apache.struts.actions.LookupDispatchAction;

import br.com.nota_fluxo.bean.Cnae;
import br.com.nota_fluxo.form.CnaeForm;
import br.com.nota_fluxo.sql.CnaeSQL;
import br.com.nota_fluxo.util.Util;

public class CnaeAction extends LookupDispatchAction {

        public Category cat = Category.getInstance(CnaeAction.class.getName());

        protected Map getKeyMethodMap(){
                Map map = new HashMap();
                map.put("nota_fluxo.entra_inserir","entra_inserir");
                map.put("nota_fluxo.entra_consultar","entra_consultar");
                map.put("nota_fluxo.entra_alterar","entra_alterar");
                map.put("nota_fluxo.inserir","inserir");
                map.put("nota_fluxo.apagar","apagar");
                map.put("nota_fluxo.alterar","alterar");
                map.put("nota_fluxo.lista","lista");
                map.put("nota_fluxo.login.valida","lista");
                return map;
        }

        public ActionForward unspecified(
```

```
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
        return (lista(mapping, form, request, response));
}

public ActionForward lista(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
        ActionMessages msgs = new ActionMessages();

        Connection con =
                (Connection) request.getSession().getAttribute("conexao");

        //UsuarioForm usuario_form =
        //      (UsuarioForm) request.getSession().getAttribute("usuarioForm");
        //cat.warn("cnae "+usuario_form.getUsuario().getCd_emp());

        CnaeForm beanForm = (CnaeForm) form;
        CnaeSQL beanSQL = new CnaeSQL();
        try {
                beanForm.setNr_reg(10); //Numero de registros da lista
                beanForm.setMostraConsulta("N");

                int p_pagina = 0;
                if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                        p_pagina = 0;
                } else {
                        p_pagina =
                                (int) Integer.parseInt(request.getParameter("p_pagina"));
                }
                beanForm.setCnae_lista(
                        beanSQL.select(
                                con,
                                p_pagina,
                                beanForm.getCnae_pesq(),
                                beanForm.getNr_reg(),
                                "", ""));
                beanForm.setContadoresPaginas(
                        beanSQL.getContaSelect(),
                        beanSQL.getContaSelectAtual());
        } catch (Exception e) {
                msgs.add(
                        ActionMessages.GLOBAL_MESSAGE,
                        new ActionMessage("error.sql", e));
        }
        if (!(msgs.isEmpty())) {
                saveMessages(request, msgs);
        }
        return (mapping.findForward("lista"));
}

public ActionForward entra_consultar(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) {
        CnaeForm beanForm = (CnaeForm) form;
        beanForm.setMostraConsulta("S");
        return (mapping.findForward("editar"));
}

public ActionForward inserir(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
        ActionMessages msgs = new ActionMessages();
        Connection con =
                (Connection) request.getSession().getAttribute("conexao");
        CnaeForm beanForm = (CnaeForm) form;
        CnaeSQL beanSQL = new CnaeSQL();
        msgs = beanForm.validate(mapping, request);
        if (msgs.isEmpty()) {
                try {
                        int p_pagina = 0;

                        if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                                p_pagina = 0;
                        } else {
                                p_pagina =
                                        (int) Integer.parseInt(
```

```
                                        request.getParameter("p_pagina"));
                        }
                        String oid = "-1";
                        if (Util.trataNulo(request.getParameter("oid")) != null) {
                                oid = request.getParameter("oid");
                        }

                        Cnae v_cnae = new Cnae();
                        v_cnae.setCd_cnae(beanForm.getCnae().getCd_cnae());
                        beanForm.setCnae_lista(
                                beanSQL.select(con, 0, v_cnae, 1, "", "N"));
                        if (beanForm.getCnae_lista().size() > 0) {
                                throw new ExcecaoGeral(
                                        new ActionMessage("nota_fluxo.inserir.erro"));
                        } else {
                                beanSQL.insert(beanForm.getCnae(), con);
                                con.commit();
                                beanForm.setCnae_lista(

beanSQL.select(con,0,beanForm.getCnae(),1,"","N"));
                                beanForm.setCnae_pesq(new Cnae());
                                beanForm.setCnae_lista(
                                        beanSQL.select(
                                                con,
                                                0,
                                                beanForm.getCnae_pesq(),
                                                2000000000,
                                                ((Cnae) beanForm.getCnae_lista().get(0))
                                                        .getOid(), "N"));
                                //
                                beanForm.setNr_reg(10);
                                p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
                                beanForm.setCnae_lista(
                                        beanSQL.select(
                                                con,
                                                p_pagina,
                                                beanForm.getCnae_pesq(),
                                                beanForm.getNr_reg(),
                                                "", ""));
                                beanForm.setContadoresPaginas(
                                        beanSQL.getContaSelect(),
                                        beanSQL.getContaSelectAtual());
                        }
                } catch (ExcecaoGeral e) {
                        msgs.add(ActionMessages.GLOBAL_MESSAGE, e.getAction_message());
                } catch (Exception e) {
                        msgs.add(
                                ActionMessages.GLOBAL_MESSAGE,
                                new ActionMessage("error.sql", e.toString()));
                }
        }
        if (!(msgs.isEmpty())) {
                con.rollback();
                saveMessages(request, msgs);
        }
        return (mapping.findForward("lista"));
}

public ActionForward entra_inserir(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
        CnaeForm beanForm = (CnaeForm) form;
        beanForm.setCnae(new Cnae());
        beanForm.setMostraConsulta("N");
        return (mapping.findForward("editar"));
}

public ActionForward apagar(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
        ActionMessages msgs = new ActionMessages();
        Connection con =
                (Connection) request.getSession().getAttribute("conexao");
        CnaeForm beanForm = (CnaeForm) form;
        CnaeSQL beanSQL = new CnaeSQL();
        try {
                int p_pagina = 0;
                if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                        p_pagina = 0;
                } else {
                        p_pagina =
                                (int) Integer.parseInt(request.getParameter("p_pagina"));
```

```
                }

                String oid = "-1";
                if (Util.trataNulo(request.getParameter("oid")) != null) {
                        oid = request.getParameter("oid");
                }

                Cnae v_cnae = new Cnae();
                v_cnae.setOid(oid);
                v_cnae =
                        (Cnae) beanSQL.select(con, 0, v_cnae, 1, "","N").get(0);

                beanForm.setCnae_lista(
                        beanSQL.select(
                                con,
                                0,
                                beanForm.getCnae_pesq(),
                                2000000000,
                                oid, ""));
                p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());

                beanSQL.delete(oid, con);
                con.commit();

                request.setAttribute("oid", "");
        } catch (Exception e) {
                msgs.add(
                        ActionMessages.GLOBAL_MESSAGE,
                        new ActionMessage("error.sql", e.toString()));
        }
        if (!(msgs.isEmpty())) {
                con.rollback();
                saveMessages(request, msgs);
        }
        return (lista(mapping, beanForm, request, response));
}

public ActionForward entra_alterar(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
        ActionMessages msgs = new ActionMessages();
        Connection con =
                (Connection) request.getSession().getAttribute("conexao");
        CnaeForm beanForm = (CnaeForm) form;
        try {
                CnaeSQL beanSQL = new CnaeSQL();

                String oid = "-1";
                if (Util.trataNulo(request.getParameter("oid")) != null) {
                        oid = request.getParameter("oid");
                }

                Cnae v_cnae = new Cnae();
                v_cnae.setOid(oid);
                beanForm.setCnae_lista(beanSQL.select(con,0,v_cnae,1,"","N"));
                beanForm.setCnae((Cnae) beanForm.getCnae_lista().get(0));
        } catch (Exception e) {
                msgs.add(
                        ActionMessages.GLOBAL_MESSAGE,
                        new ActionMessage("error.sql", e.toString()));
        }
        if (!(msgs.isEmpty())) {
                con.rollback();
                saveMessages(request, msgs);
        }
        return (mapping.findForward("editar"));
}

public ActionForward alterar(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
        ActionMessages msgs = new ActionMessages();
        Connection con =
                (Connection) request.getSession().getAttribute("conexao");
        CnaeForm beanForm = (CnaeForm) form;
        CnaeSQL beanSQL = new CnaeSQL();
        msgs = beanForm.validate(mapping, request);
        if (msgs.isEmpty()) {
                try {
                        int p_pagina = 0;
                        if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                                p_pagina = 0;
```

```
                                } else {
                                        p_pagina =
                                                (int) Integer.parseInt(
                                                        request.getParameter("p_pagina"));
                                }

                                String oid = "-1";
                                if (Util.trataNulo(request.getParameter("oid")) != null) {
                                        oid = request.getParameter("oid");
                                }

                                beanSQL.update(beanForm.getCnae(), con);
                                con.commit();

                                beanForm.setCnae_lista(
                                        beanSQL.select(con, 0, beanForm.getCnae(), 1, "", "N"));
                                if (beanForm.getCnae_lista().size() == 0) {
                                        throw new ExcecaoGeral(
                                                new ActionMessage("nota_fluxo.alterar.erro"));
                                }

                                beanForm.setCnae_lista(
                                        beanSQL.select(
                                                con,
                                                0,
                                                beanForm.getCnae_pesq(),
                                                2000000000,
                                                beanForm.getCnae().getOid(), "N"));


                                //
                                beanForm.setNr_reg(10);
                                p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
                                beanForm.setCnae_lista(
                                        beanSQL.select(
                                                con,
                                                p_pagina,
                                                beanForm.getCnae_pesq(),
                                                beanForm.getNr_reg(),
                                                "",""));
                                beanForm.setContadoresPaginas(
                                        beanSQL.getContaSelect(),
                                        beanSQL.getContaSelectAtual());


                        } catch (ExcecaoGeral e) {
                                msgs.add(ActionMessages.GLOBAL_MESSAGE, e.getAction_message());
                        } catch (Exception e) {
                                msgs.add(
                                        ActionMessages.GLOBAL_MESSAGE,
                                        new ActionMessage("error.sql", e.toString()));
                        }
                }
                if (!(msgs.isEmpty())) {
                        con.rollback();
                        saveMessages(request, msgs);
                }
                return (mapping.findForward("lista"));
        }
}
```

Contas_pagarAction.java

```
package br.com.nota_fluxo.action;

import java.sql.Connection;
import java.util.HashMap;
import java.util.Map;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.log4j.Category;

import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;
import org.apache.struts.action.ActionMessages;
import org.apache.struts.actions.LookupDispatchAction;

import br.com.nota_fluxo.bean.Contas_pagar;
import br.com.nota_fluxo.form.Contas_pagarForm;
import br.com.nota_fluxo.sql.Contas_pagarSQL;
import br.com.nota_fluxo.util.Util;

public class Contas_pagarAction extends LookupDispatchAction {
```

```
public Category cat = Category.getInstance(Contas_pagarAction.class.getName());

protected Map getKeyMethodMap(){
        Map map = new HashMap();
        map.put("nota_fluxo.entra_inserir","entra_inserir");
        map.put("nota_fluxo.entra_consultar","entra_consultar");
        map.put("nota_fluxo.entra_alterar","entra_alterar");
        map.put("nota_fluxo.inserir","inserir");
        map.put("nota_fluxo.apagar","apagar");
        map.put("nota_fluxo.alterar","alterar");
        map.put("nota_fluxo.lista","lista");
        map.put("nota_fluxo.login.valida","lista");
        return map;
}

public ActionForward unspecified(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
        return (lista(mapping, form, request, response));
}

public ActionForward lista(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
        ActionMessages msgs = new ActionMessages();

        Connection con =
                (Connection) request.getSession().getAttribute("conexao");

        //UsuarioForm usuario_form =
        //      (UsuarioForm) request.getSession().getAttribute("usuarioForm");
        //cat.warn("contas_pagar "+usuario_form.getUsuario().getCd_emp());

        Contas_pagarForm beanForm = (Contas_pagarForm) form;
        Contas_pagarSQL beanSQL = new Contas_pagarSQL();
        try {
                beanForm.setNr_reg(1); //Numero de registros da lista
                beanForm.setMostraConsulta("N");

                int p_pagina = 0;
                if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                        p_pagina = 0;
                } else {
                        p_pagina =
                                (int) Integer.parseInt(request.getParameter("p_pagina"));
                }
                beanForm.setContas_pagar_lista(
                        beanSQL.select(
                                con,
                                p_pagina,
                                beanForm.getContas_pagar_pesq(),
                                beanForm.getNr_reg(),
                                "", ""));
                beanForm.setContadoresPaginas(
                        beanSQL.getContaSelect(),
                        beanSQL.getContaSelectAtual());
        } catch (Exception e) {
                msgs.add(
                        ActionMessages.GLOBAL_MESSAGE,
                        new ActionMessage("error.sql", e));
        }
        if (!(msgs.isEmpty())) {
                saveMessages(request, msgs);
        }
        return (mapping.findForward("lista"));
}

public ActionForward entra_consultar(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) {
        Contas_pagarForm beanForm = (Contas_pagarForm) form;
        beanForm.setMostraConsulta("S");
        return (mapping.findForward("editar"));
}

public ActionForward inserir(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
```

```
                throws Exception {
                ActionMessages msgs = new ActionMessages();
                Connection con =
                        (Connection) request.getSession().getAttribute("conexao");
                Contas_pagarForm beanForm = (Contas_pagarForm) form;
                Contas_pagarSQL beanSQL = new Contas_pagarSQL();
                msgs = beanForm.validate(mapping, request);
                if (msgs.isEmpty()) {
                        try {
                                int p_pagina = 0;

                                if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                                        p_pagina = 0;
                                } else {
                                        p_pagina =
                                                (int) Integer.parseInt(
                                                        request.getParameter("p_pagina"));
                                }
                                String oid = "-1";
                                if (Util.trataNulo(request.getParameter("oid")) != null) {
                                        oid = request.getParameter("oid");
                                }

                                Contas_pagar v_contas_pagar = new Contas_pagar();
                                v_contas_pagar.setCd_emp(beanForm.getContas_pagar().getCd_emp());

        v_contas_pagar.setCd_pagar(beanForm.getContas_pagar().getCd_pagar());
                                beanForm.setContas_pagar_lista(
                                        beanSQL.select(con, 0, v_contas_pagar, 1, "", "N"));
                                if (beanForm.getContas_pagar_lista().size() > 0) {
                                        throw new ExcecaoGeral(
                                                new ActionMessage("nota_fluxo.inserir.erro"));
                                } else {
                                        beanSQL.insert(beanForm.getContas_pagar(), con);
                                        con.commit();
                                        beanForm.setContas_pagar_lista(

        beanSQL.select(con,0,beanForm.getContas_pagar(),1,"","N"));
                                        beanForm.setContas_pagar_pesq(new Contas_pagar());
                                        beanForm.setContas_pagar_lista(
                                                beanSQL.select(
                                                        con,
                                                        0,
                                                        beanForm.getContas_pagar_pesq(),
                                                        2000000000,
                                                        ((Contas_pagar)
beanForm.getContas_pagar_lista().get(0))
                                                                        .getOid(), "N"));
                                        //
                                        beanForm.setNr_reg(1);
                                        p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
                                        beanForm.setContas_pagar_lista(
                                                beanSQL.select(
                                                        con,
                                                        p_pagina,
                                                        beanForm.getContas_pagar_pesq(),
                                                        beanForm.getNr_reg(),
                                                        "", ""));
                                        beanForm.setContadoresPaginas(
                                                beanSQL.getContaSelect(),
                                                beanSQL.getContaSelectAtual());
                                }
                        } catch (ExcecaoGeral e) {
                                msgs.add(ActionMessages.GLOBAL_MESSAGE, e.getAction_message());
                        } catch (Exception e) {
                                msgs.add(
                                        ActionMessages.GLOBAL_MESSAGE,
                                        new ActionMessage("error.sql", e.toString()));
                        }
                }
                if (!(msgs.isEmpty())) {
                        con.rollback();
                        saveMessages(request, msgs);
                }
                return (mapping.findForward("lista"));
        }

        public ActionForward entra_inserir(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                Contas_pagarForm beanForm = (Contas_pagarForm) form;
                beanForm.setContas_pagar(new Contas_pagar());
                beanForm.setMostraConsulta("N");
                return (mapping.findForward("editar"));
        }
```

```
	public ActionForward apagar(
			ActionMapping mapping,
			ActionForm form,
			HttpServletRequest request,
			HttpServletResponse response)
			throws Exception {
			ActionMessages msgs = new ActionMessages();
			Connection con =
				(Connection) request.getSession().getAttribute("conexao");
			Contas_pagarForm beanForm = (Contas_pagarForm) form;
			Contas_pagarSQL beanSQL = new Contas_pagarSQL();
			try {
				int p_pagina = 0;
				if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
					p_pagina = 0;
				} else {
					p_pagina =
						(int) Integer.parseInt(request.getParameter("p_pagina"));
				}

				String oid = "-1";
				if (Util.trataNulo(request.getParameter("oid")) != null) {
					oid = request.getParameter("oid");
				}

				Contas_pagar v_contas_pagar = new Contas_pagar();
				v_contas_pagar.setOid(oid);
				v_contas_pagar =
					(Contas_pagar) beanSQL.select(con, 0, v_contas_pagar, 1,
"","N").get(0);

				beanForm.setContas_pagar_lista(
					beanSQL.select(
						con,
						0,
						beanForm.getContas_pagar_pesq(),
						2000000000,
						oid, ""));
				p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());

				beanSQL.delete(oid, con);
				con.commit();

				request.setAttribute("oid", "");
			} catch (Exception e) {
				msgs.add(
					ActionMessages.GLOBAL_MESSAGE,
					new ActionMessage("error.sql", e.toString()));
			}
			if (!(msgs.isEmpty())) {
				con.rollback();
				saveMessages(request, msgs);
			}
			return (lista(mapping, beanForm, request, response));
	}

	public ActionForward entra_alterar(
			ActionMapping mapping,
			ActionForm form,
			HttpServletRequest request,
			HttpServletResponse response)
			throws Exception {
			ActionMessages msgs = new ActionMessages();
			Connection con =
				(Connection) request.getSession().getAttribute("conexao");
			Contas_pagarForm beanForm = (Contas_pagarForm) form;
			try {
				Contas_pagarSQL beanSQL = new Contas_pagarSQL();

				String oid = "-1";
				if (Util.trataNulo(request.getParameter("oid")) != null) {
					oid = request.getParameter("oid");
				}

				Contas_pagar v_contas_pagar = new Contas_pagar();
				v_contas_pagar.setOid(oid);

	beanForm.setContas_pagar_lista(beanSQL.select(con,0,v_contas_pagar,1,"","N"));
				beanForm.setContas_pagar((Contas_pagar)
beanForm.getContas_pagar_lista().get(0));
			} catch (Exception e) {
				msgs.add(
					ActionMessages.GLOBAL_MESSAGE,
					new ActionMessage("error.sql", e.toString()));
			}
			if (!(msgs.isEmpty())) {
				con.rollback();
```

```
                        saveMessages(request, msgs);
                }
                return (mapping.findForward("editar"));
        }

        public ActionForward alterar(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                ActionMessages msgs = new ActionMessages();
                Connection con =
                        (Connection) request.getSession().getAttribute("conexao");
                Contas_pagarForm beanForm = (Contas_pagarForm) form;
                Contas_pagarSQL beanSQL = new Contas_pagarSQL();
                msgs = beanForm.validate(mapping, request);
                if (msgs.isEmpty()) {
                        try {
                                int p_pagina = 0;
                                if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                                        p_pagina = 0;
                                } else {
                                        p_pagina =
                                                (int) Integer.parseInt(
                                                        request.getParameter("p_pagina"));
                                }

                                String oid = "-1";
                                if (Util.trataNulo(request.getParameter("oid")) != null) {
                                        oid = request.getParameter("oid");
                                }

                                beanSQL.update(beanForm.getContas_pagar(), con);
                                con.commit();

                                beanForm.setContas_pagar_lista(
                                        beanSQL.select(con, 0, beanForm.getContas_pagar(), 1, "",
"N"));
                                if (beanForm.getContas_pagar_lista().size() == 0) {
                                        throw new ExcecaoGeral(
                                                new ActionMessage("nota_fluxo.alterar.erro"));
                                }

                                beanForm.setContas_pagar_lista(
                                        beanSQL.select(
                                                con,
                                                0,
                                                beanForm.getContas_pagar_pesq(),
                                                2000000000,
                                                beanForm.getContas_pagar().getOid(), "N"));


                                //
                                beanForm.setNr_reg(1);
                                p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
                                beanForm.setContas_pagar_lista(
                                        beanSQL.select(
                                                con,
                                                p_pagina,
                                                beanForm.getContas_pagar_pesq(),
                                                beanForm.getNr_reg(),
                                                "",""));
                                beanForm.setContadoresPaginas(
                                        beanSQL.getContaSelect(),
                                        beanSQL.getContaSelectAtual());


                        } catch (ExcecaoGeral e) {
                                msgs.add(ActionMessages.GLOBAL_MESSAGE, e.getAction_message());
                        } catch (Exception e) {
                                msgs.add(
                                        ActionMessages.GLOBAL_MESSAGE,
                                        new ActionMessage("error.sql", e.toString()));
                        }
                }
                if (!(msgs.isEmpty())) {
                        con.rollback();
                        saveMessages(request, msgs);
                }
                return (mapping.findForward("lista"));
        }
}
```

Contas_receberAction.java

```
package br.com.nota_fluxo.action;
```

```
import java.sql.Connection;
import java.util.HashMap;
import java.util.Map;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.log4j.Category;

import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;
import org.apache.struts.action.ActionMessages;
import org.apache.struts.actions.LookupDispatchAction;

import br.com.nota_fluxo.bean.Contas_receber;
import br.com.nota_fluxo.form.Contas_receberForm;
import br.com.nota_fluxo.sql.Contas_receberSQL;
import br.com.nota_fluxo.util.Util;

public class Contas_receberAction extends LookupDispatchAction {

	public Category cat = Category.getInstance(Contas_receberAction.class.getName());

	protected Map getKeyMethodMap(){
		Map map = new HashMap();
		map.put("nota_fluxo.entra_inserir","entra_inserir");
		map.put("nota_fluxo.entra_consultar","entra_consultar");
		map.put("nota_fluxo.entra_alterar","entra_alterar");
		map.put("nota_fluxo.inserir","inserir");
		map.put("nota_fluxo.areceber","areceber");
		map.put("nota_fluxo.alterar","alterar");
		map.put("nota_fluxo.lista","lista");
		map.put("nota_fluxo.login.valida","lista");
		return map;
	}

	public ActionForward unspecified(
		ActionMapping mapping,
		ActionForm form,
		HttpServletRequest request,
		HttpServletResponse response)
		throws Exception {
		return (lista(mapping, form, request, response));
	}

	public ActionForward lista(
		ActionMapping mapping,
		ActionForm form,
		HttpServletRequest request,
		HttpServletResponse response)
		throws Exception {
		ActionMessages msgs = new ActionMessages();

		Connection con =
			(Connection) request.getSession().getAttribute("conexao");

		//UsuarioForm usuario_form =
		//	(UsuarioForm) request.getSession().getAttribute("usuarioForm");
		//cat.warn("contas_receber "+usuario_form.getUsuario().getCd_emp());

		Contas_receberForm beanForm = (Contas_receberForm) form;
		Contas_receberSQL beanSQL = new Contas_receberSQL();
		try {
			beanForm.setNr_reg(1); //Numero de registros da lista
			beanForm.setMostraConsulta("N");

			int p_pagina = 0;
			if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
				p_pagina = 0;
			} else {
				p_pagina =
					(int) Integer.parseInt(request.getParameter("p_pagina"));
			}
			beanForm.setContas_receber_lista(
				beanSQL.select(
					con,
					p_pagina,
					beanForm.getContas_receber_pesq(),
					beanForm.getNr_reg(),
					"", ""));
			beanForm.setContadoresPaginas(
				beanSQL.getContaSelect(),
				beanSQL.getContaSelectAtual());
		} catch (Exception e) {
			msgs.add(
				ActionMessages.GLOBAL_MESSAGE,
```

```
                                new ActionMessage("error.sql", e));
                        }
                        if (!(msgs.isEmpty())) {
                                saveMessages(request, msgs);
                        }
                        return (mapping.findForward("lista"));
                }

                public ActionForward entra_consultar(
                        ActionMapping mapping,
                        ActionForm form,
                        HttpServletRequest request,
                        HttpServletResponse response) {
                        Contas_receberForm beanForm = (Contas_receberForm) form;
                        beanForm.setMostraConsulta("S");
                        return (mapping.findForward("editar"));
                }

                public ActionForward inserir(
                        ActionMapping mapping,
                        ActionForm form,
                        HttpServletRequest request,
                        HttpServletResponse response)
                        throws Exception {
                        ActionMessages msgs = new ActionMessages();
                        Connection con =
                                (Connection) request.getSession().getAttribute("conexao");
                        Contas_receberForm beanForm = (Contas_receberForm) form;
                        Contas_receberSQL beanSQL = new Contas_receberSQL();
                        msgs = beanForm.validate(mapping, request);
                        if (msgs.isEmpty()) {
                                try {
                                        int p_pagina = 0;

                                        if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                                                p_pagina = 0;
                                        } else {
                                                p_pagina =
                                                        (int) Integer.parseInt(
                                                                request.getParameter("p_pagina"));
                                        }
                                        String oid = "-1";
                                        if (Util.trataNulo(request.getParameter("oid")) != null) {
                                                oid = request.getParameter("oid");
                                        }

                                        Contas_receber v_contas_receber = new Contas_receber();

                v_contas_receber.setCd_emp(beanForm.getContas_receber().getCd_emp());

                v_contas_receber.setCd_receber(beanForm.getContas_receber().getCd_receber());
                                        beanForm.setContas_receber_lista(
                                                beanSQL.select(con, 0, v_contas_receber, 1, "", "N"));
                                        if (beanForm.getContas_receber_lista().size() > 0) {
                                                throw new ExcecaoGeral(
                                                        new ActionMessage("nota_fluxo.inserir.erro"));
                                        } else {
                                                beanSQL.insert(beanForm.getContas_receber(), con);
                                                con.commit();
                                                beanForm.setContas_receber_lista(

                beanSQL.select(con,0,beanForm.getContas_receber(),1,"","N"));
                                                beanForm.setContas_receber_pesq(new Contas_receber());
                                                beanForm.setContas_receber_lista(
                                                        beanSQL.select(
                                                                con,
                                                                0,
                                                                beanForm.getContas_receber_pesq(),
                                                                2000000000,
                                                                ((Contas_receber)
beanForm.getContas_receber_lista().get(0))
                                                                        .getOid(), "N"));
                                                //
                                                beanForm.setNr_reg(1);
                                                p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
                                                beanForm.setContas_receber_lista(
                                                        beanSQL.select(
                                                                con,
                                                                p_pagina,
                                                                beanForm.getContas_receber_pesq(),
                                                                beanForm.getNr_reg(),
                                                                "", ""));
                                                beanForm.setContadoresPaginas(
                                                        beanSQL.getContaSelect(),
                                                        beanSQL.getContaSelectAtual());
                                        }
                                } catch (ExcecaoGeral e) {
                                        msgs.add(ActionMessages.GLOBAL_MESSAGE, e.getAction_message());
```

```
                        } catch (Exception e) {
                                msgs.add(
                                        ActionMessages.GLOBAL_MESSAGE,
                                        new ActionMessage("error.sql", e.toString()));
                        }
                }
                if (!(msgs.isEmpty())) {
                        con.rollback();
                        saveMessages(request, msgs);
                }
                return (mapping.findForward("lista"));
        }

        public ActionForward entra_inserir(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                Contas_receberForm beanForm = (Contas_receberForm) form;
                beanForm.setContas_receber(new Contas_receber());
                beanForm.setMostraConsulta("N");
                return (mapping.findForward("editar"));
        }

        public ActionForward areceber(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                ActionMessages msgs = new ActionMessages();
                Connection con =
                        (Connection) request.getSession().getAttribute("conexao");
                Contas_receberForm beanForm = (Contas_receberForm) form;
                Contas_receberSQL beanSQL = new Contas_receberSQL();
                try {
                        int p_pagina = 0;
                        if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                                p_pagina = 0;
                        } else {
                                p_pagina =
                                        (int) Integer.parseInt(request.getParameter("p_pagina"));
                        }

                        String oid = "-1";
                        if (Util.trataNulo(request.getParameter("oid")) != null) {
                                oid = request.getParameter("oid");
                        }

                        Contas_receber v_contas_receber = new Contas_receber();
                        v_contas_receber.setOid(oid);
                        v_contas_receber =
                                (Contas_receber) beanSQL.select(con, 0, v_contas_receber, 1,
"","N").get(0);

                        beanForm.setContas_receber_lista(
                                beanSQL.select(
                                        con,
                                        0,
                                        beanForm.getContas_receber_pesq(),
                                        2000000000,
                                        oid, ""));
                        p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());

                        beanSQL.delete(oid, con);
                        con.commit();

                        request.setAttribute("oid", "");
                } catch (Exception e) {
                        msgs.add(
                                ActionMessages.GLOBAL_MESSAGE,
                                new ActionMessage("error.sql", e.toString()));
                }
                if (!(msgs.isEmpty())) {
                        con.rollback();
                        saveMessages(request, msgs);
                }
                return (lista(mapping, beanForm, request, response));
        }

        public ActionForward entra_alterar(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                ActionMessages msgs = new ActionMessages();
```

```
                Connection con =
                        (Connection) request.getSession().getAttribute("conexao");
                Contas_receberForm beanForm = (Contas_receberForm) form;
                try {
                        Contas_receberSQL beanSQL = new Contas_receberSQL();

                        String oid = "-1";
                        if (Util.trataNulo(request.getParameter("oid")) != null) {
                                oid = request.getParameter("oid");
                        }

                        Contas_receber v_contas_receber = new Contas_receber();
                        v_contas_receber.setOid(oid);

        beanForm.setContas_receber_lista(beanSQL.select(con,0,v_contas_receber,1,"","N"));
                        beanForm.setContas_receber((Contas_receber)
beanForm.getContas_receber_lista().get(0));
                } catch (Exception e) {
                        msgs.add(
                                ActionMessages.GLOBAL_MESSAGE,
                                new ActionMessage("error.sql", e.toString()));
                }
                if (!(msgs.isEmpty())) {
                        con.rollback();
                        saveMessages(request, msgs);
                }
                return (mapping.findForward("editar"));
        }

        public ActionForward alterar(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                ActionMessages msgs = new ActionMessages();
                Connection con =
                        (Connection) request.getSession().getAttribute("conexao");
                Contas_receberForm beanForm = (Contas_receberForm) form;
                Contas_receberSQL beanSQL = new Contas_receberSQL();
                msgs = beanForm.validate(mapping, request);
                if (msgs.isEmpty()) {
                        try {
                                int p_pagina = 0;
                                if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                                        p_pagina = 0;
                                } else {
                                        p_pagina =
                                                (int) Integer.parseInt(
                                                        request.getParameter("p_pagina"));
                                }

                                String oid = "-1";
                                if (Util.trataNulo(request.getParameter("oid")) != null) {
                                        oid = request.getParameter("oid");
                                }

                                beanSQL.update(beanForm.getContas_receber(), con);
                                con.commit();

                                beanForm.setContas_receber_lista(
                                        beanSQL.select(con, 0, beanForm.getContas_receber(), 1, "",
"N"));
                                if (beanForm.getContas_receber_lista().size() == 0) {
                                        throw new ExcecaoGeral(
                                                new ActionMessage("nota_fluxo.alterar.erro"));
                                }

                                beanForm.setContas_receber_lista(
                                        beanSQL.select(
                                                con,
                                                0,
                                                beanForm.getContas_receber_pesq(),
                                                2000000000,
                                                beanForm.getContas_receber().getOid(), "N"));


                                //
                                beanForm.setNr_reg(1);
                                p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
                                beanForm.setContas_receber_lista(
                                        beanSQL.select(
                                                con,
                                                p_pagina,
                                                beanForm.getContas_receber_pesq(),
                                                beanForm.getNr_reg(),
                                                "",""));
                                beanForm.setContadoresPaginas(
```

```
                                        beanSQL.getContaSelect(),
                                        beanSQL.getContaSelectAtual());


                        } catch (ExcecaoGeral e) {
                                msgs.add(ActionMessages.GLOBAL_MESSAGE, e.getAction_message());
                        } catch (Exception e) {
                                msgs.add(
                                        ActionMessages.GLOBAL_MESSAGE,
                                        new ActionMessage("error.sql", e.toString()));
                        }
                }
                if (!(msgs.isEmpty())) {
                        con.rollback();
                        saveMessages(request, msgs);
                }
                return (mapping.findForward("lista"));
        }
}
```

ContasAction.java

```
package br.com.nota_fluxo.action;

import java.sql.Connection;
import java.util.HashMap;
import java.util.Map;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.log4j.Category;

import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;
import org.apache.struts.action.ActionMessages;
import org.apache.struts.actions.LookupDispatchAction;

import br.com.nota_fluxo.bean.Contas;
import br.com.nota_fluxo.form.ContasForm;
import br.com.nota_fluxo.sql.ContasSQL;
import br.com.nota_fluxo.util.Util;

public class ContasAction extends LookupDispatchAction {

        public Category cat = Category.getInstance(ContasAction.class.getName());

        protected Map getKeyMethodMap(){
                Map map = new HashMap();
                map.put("nota_fluxo.entra_inserir","entra_inserir");
                map.put("nota_fluxo.entra_consultar","entra_consultar");
                map.put("nota_fluxo.entra_alterar","entra_alterar");
                map.put("nota_fluxo.inserir","inserir");
                map.put("nota_fluxo.apagar","apagar");
                map.put("nota_fluxo.alterar","alterar");
                map.put("nota_fluxo.lista","lista");
                map.put("nota_fluxo.login.valida","lista");
                return map;
        }

        public ActionForward unspecified(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                return (lista(mapping, form, request, response));
        }

        public ActionForward lista(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                ActionMessages msgs = new ActionMessages();

                Connection con =
                        (Connection) request.getSession().getAttribute("conexao");

                //UsuarioForm usuario_form =
                //      (UsuarioForm) request.getSession().getAttribute("usuarioForm");
                //cat.warn("contas "+usuario_form.getUsuario().getCd_emp());

                ContasForm beanForm = (ContasForm) form;
                ContasSQL beanSQL = new ContasSQL();
```

```
        try {
                beanForm.setNr_reg(10); //Numero de registros da lista
                beanForm.setMostraConsulta("N");

                int p_pagina = 0;
                if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                        p_pagina = 0;
                } else {
                        p_pagina =
                                (int) Integer.parseInt(request.getParameter("p_pagina"));
                }
                beanForm.setContas_lista(
                        beanSQL.select(
                                con,
                                p_pagina,
                                beanForm.getContas_pesq(),
                                beanForm.getNr_reg(),
                                "", ""));
                beanForm.setContadoresPaginas(
                        beanSQL.getContaSelect(),
                        beanSQL.getContaSelectAtual());
        } catch (Exception e) {
                msgs.add(
                        ActionMessages.GLOBAL_MESSAGE,
                        new ActionMessage("error.sql", e));
        }
        if (!(msgs.isEmpty())) {
                saveMessages(request, msgs);
        }
        return (mapping.findForward("lista"));
}

public ActionForward entra_consultar(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) {
        ContasForm beanForm = (ContasForm) form;
        beanForm.setMostraConsulta("S");
        return (mapping.findForward("editar"));
}

public ActionForward inserir(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
        ActionMessages msgs = new ActionMessages();
        Connection con =
                (Connection) request.getSession().getAttribute("conexao");
        ContasForm beanForm = (ContasForm) form;
        ContasSQL beanSQL = new ContasSQL();
        msgs = beanForm.validate(mapping, request);
        if (msgs.isEmpty()) {
                try {
                        int p_pagina = 0;

                        if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                                p_pagina = 0;
                        } else {
                                p_pagina =
                                        (int) Integer.parseInt(
                                                request.getParameter("p_pagina"));
                        }
                        String oid = "-1";
                        if (Util.trataNulo(request.getParameter("oid")) != null) {
                                oid = request.getParameter("oid");
                        }

                        Contas v_contas = new Contas();
                        v_contas.setCd_emp(beanForm.getContas().getCd_emp());
                        v_contas.setCd_conta(beanForm.getContas().getCd_conta());
                        beanForm.setContas_lista(
                                beanSQL.select(con, 0, v_contas, 1, "", "N"));
                        if (beanForm.getContas_lista().size() > 0) {
                                throw new ExcecaoGeral(
                                        new ActionMessage("nota_fluxo.inserir.erro"));
                        } else {
                                beanSQL.insert(beanForm.getContas(), con);
                                con.commit();
                                beanForm.setContas_lista(

beanSQL.select(con,0,beanForm.getContas(),1,"","N"));
                                beanForm.setContas_pesq(new Contas());
                                beanForm.setContas_lista(
                                        beanSQL.select(
                                                con,
```

```
                                                0,
                                                beanForm.getContas_pesq(),
                                                2000000000,
                                                ((Contas)
beanForm.getContas_lista().get(0))
                                                        .getOid(), "N"));
                                        //
                                        beanForm.setNr_reg(10);
                                        p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
                                        beanForm.setContas_lista(
                                                beanSQL.select(
                                                        con,
                                                        p_pagina,
                                                        beanForm.getContas_pesq(),
                                                        beanForm.getNr_reg(),
                                                        "", ""));
                                        beanForm.setContadoresPaginas(
                                                beanSQL.getContaSelect(),
                                                beanSQL.getContaSelectAtual());
                                }
                        } catch (ExcecaoGeral e) {
                                msgs.add(ActionMessages.GLOBAL_MESSAGE, e.getAction_message());
                        } catch (Exception e) {
                                msgs.add(
                                        ActionMessages.GLOBAL_MESSAGE,
                                        new ActionMessage("error.sql", e.toString()));
                        }
                }
                if (!(msgs.isEmpty())) {
                        con.rollback();
                        saveMessages(request, msgs);
                }
                return (mapping.findForward("lista"));
        }

        public ActionForward entra_inserir(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                ContasForm beanForm = (ContasForm) form;
                beanForm.setContas(new Contas());
                beanForm.setMostraConsulta("N");
                return (mapping.findForward("editar"));
        }

        public ActionForward apagar(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                ActionMessages msgs = new ActionMessages();
                Connection con =
                        (Connection) request.getSession().getAttribute("conexao");
                ContasForm beanForm = (ContasForm) form;
                ContasSQL beanSQL = new ContasSQL();
                try {
                        int p_pagina = 0;
                        if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                                p_pagina = 0;
                        } else {
                                p_pagina =
                                        (int) Integer.parseInt(request.getParameter("p_pagina"));
                        }

                        String oid = "-1";
                        if (Util.trataNulo(request.getParameter("oid")) != null) {
                                oid = request.getParameter("oid");
                        }

                        Contas v_contas = new Contas();
                        v_contas.setOid(oid);
                        v_contas =
                                (Contas) beanSQL.select(con, 0, v_contas, 1, "","N").get(0);

                        beanForm.setContas_lista(
                                beanSQL.select(
                                        con,
                                        0,
                                        beanForm.getContas_pesq(),
                                        2000000000,
                                        oid, ""));
                        p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());

                        beanSQL.delete(oid, con);
                        con.commit();
```

```
                request.setAttribute("oid", "");
        } catch (Exception e) {
                msgs.add(
                        ActionMessages.GLOBAL_MESSAGE,
                        new ActionMessage("error.sql", e.toString()));
        }
        if (!(msgs.isEmpty())) {
                con.rollback();
                saveMessages(request, msgs);
        }
        return (lista(mapping, beanForm, request, response));
}

public ActionForward entra_alterar(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
        ActionMessages msgs = new ActionMessages();
        Connection con =
                (Connection) request.getSession().getAttribute("conexao");
        ContasForm beanForm = (ContasForm) form;
        try {
                ContasSQL beanSQL = new ContasSQL();

                String oid = "-1";
                if (Util.trataNulo(request.getParameter("oid")) != null) {
                        oid = request.getParameter("oid");
                }

                Contas v_contas = new Contas();
                v_contas.setOid(oid);
                beanForm.setContas_lista(beanSQL.select(con,0,v_contas,1,"","N"));
                beanForm.setContas((Contas) beanForm.getContas_lista().get(0));
        } catch (Exception e) {
                msgs.add(
                        ActionMessages.GLOBAL_MESSAGE,
                        new ActionMessage("error.sql", e.toString()));
        }
        if (!(msgs.isEmpty())) {
                con.rollback();
                saveMessages(request, msgs);
        }
        return (mapping.findForward("editar"));
}

public ActionForward alterar(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
        ActionMessages msgs = new ActionMessages();
        Connection con =
                (Connection) request.getSession().getAttribute("conexao");
        ContasForm beanForm = (ContasForm) form;
        ContasSQL beanSQL = new ContasSQL();
        msgs = beanForm.validate(mapping, request);
        if (msgs.isEmpty()) {
                try {
                        int p_pagina = 0;
                        if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                                p_pagina = 0;
                        } else {
                                p_pagina =
                                        (int) Integer.parseInt(
                                                request.getParameter("p_pagina"));
                        }

                        String oid = "-1";
                        if (Util.trataNulo(request.getParameter("oid")) != null) {
                                oid = request.getParameter("oid");
                        }

                        beanSQL.update(beanForm.getContas(), con);
                        con.commit();

                        beanForm.setContas_lista(
                                beanSQL.select(con, 0, beanForm.getContas(), 1, "", "N"));
                        if (beanForm.getContas_lista().size() == 0) {
                                throw new ExcecaoGeral(
                                        new ActionMessage("nota_fluxo.alterar.erro"));
                        }

                        beanForm.setContas_lista(
                                beanSQL.select(
```

```
                                                con,
                                                0,
                                                beanForm.getContas_pesq(),
                                                2000000000,
                                                beanForm.getContas().getOid(), "N"));


                                        //
                                        beanForm.setNr_reg(10);
                                        p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
                                        beanForm.setContas_lista(
                                                beanSQL.select(
                                                        con,
                                                        p_pagina,
                                                        beanForm.getContas_pesq(),
                                                        beanForm.getNr_reg(),
                                                        "",""));
                                        beanForm.setContadoresPaginas(
                                                beanSQL.getContaSelect(),
                                                beanSQL.getContaSelectAtual());


                                } catch (ExcecaoGeral e) {
                                        msgs.add(ActionMessages.GLOBAL_MESSAGE, e.getAction_message());
                                } catch (Exception e) {
                                        msgs.add(
                                                ActionMessages.GLOBAL_MESSAGE,
                                                new ActionMessage("error.sql", e.toString()));
                                }
                        }
                        if (!(msgs.isEmpty())) {
                                con.rollback();
                                saveMessages(request, msgs);
                        }
                        return (mapping.findForward("lista"));
                }
        }
```

EmpresaAction.java

```
package br.com.nota_fluxo.action;

import java.sql.Connection;
import java.util.HashMap;
import java.util.Map;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.log4j.Category;

import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;
import org.apache.struts.action.ActionMessages;
import org.apache.struts.actions.LookupDispatchAction;

import br.com.nota_fluxo.bean.Empresa;
import br.com.nota_fluxo.form.EmpresaForm;
import br.com.nota_fluxo.sql.EmpresaSQL;
import br.com.nota_fluxo.util.Util;

public class EmpresaAction extends LookupDispatchAction {

        public Category cat = Category.getInstance(EmpresaAction.class.getName());

        protected Map getKeyMethodMap(){
                Map map = new HashMap();
                map.put("nota_fluxo.entra_inserir","entra_inserir");
                map.put("nota_fluxo.entra_consultar","entra_consultar");
                map.put("nota_fluxo.entra_alterar","entra_alterar");
                map.put("nota_fluxo.inserir","inserir");
                map.put("nota_fluxo.apagar","apagar");
                map.put("nota_fluxo.alterar","alterar");
                map.put("nota_fluxo.lista","lista");
                map.put("nota_fluxo.login.valida","lista");
                return map;
        }

        public ActionForward unspecified(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                return (lista(mapping, form, request, response));
        }
```

```
public ActionForward lista(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
        ActionMessages msgs = new ActionMessages();

        Connection con =
                (Connection) request.getSession().getAttribute("conexao");

        //UsuarioForm usuario_form =
        //      (UsuarioForm) request.getSession().getAttribute("usuarioForm");
        //cat.warn("empresa "+usuario_form.getUsuario().getCd_emp());

        EmpresaForm beanForm = (EmpresaForm) form;
        EmpresaSQL beanSQL = new EmpresaSQL();
        try {
                beanForm.setNr_reg(1); //Numero de registros da lista
                beanForm.setMostraConsulta("N");

                int p_pagina = 0;
                if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                        p_pagina = 0;
                } else {
                        p_pagina =
                                (int) Integer.parseInt(request.getParameter("p_pagina"));
                }
                beanForm.setEmpresa_lista(
                        beanSQL.select(
                                con,
                                p_pagina,
                                beanForm.getEmpresa_pesq(),
                                beanForm.getNr_reg(),
                                "", ""));
                beanForm.setContadoresPaginas(
                        beanSQL.getContaSelect(),
                        beanSQL.getContaSelectAtual());
        } catch (Exception e) {
                msgs.add(
                        ActionMessages.GLOBAL_MESSAGE,
                        new ActionMessage("error.sql", e));
        }
        if (!(msgs.isEmpty())) {
                saveMessages(request, msgs);
        }
        return (mapping.findForward("lista"));
}

public ActionForward entra_consultar(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) {
        EmpresaForm beanForm = (EmpresaForm) form;
        beanForm.setMostraConsulta("S");
        return (mapping.findForward("editar"));
}

public ActionForward inserir(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
        ActionMessages msgs = new ActionMessages();
        Connection con =
                (Connection) request.getSession().getAttribute("conexao");
        EmpresaForm beanForm = (EmpresaForm) form;
        EmpresaSQL beanSQL = new EmpresaSQL();
        msgs = beanForm.validate(mapping, request);
        if (msgs.isEmpty()) {
                try {
                        int p_pagina = 0;

                        if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                                p_pagina = 0;
                        } else {
                                p_pagina =
                                        (int) Integer.parseInt(
                                                request.getParameter("p_pagina"));
                        }
                        String oid = "-1";
                        if (Util.trataNulo(request.getParameter("oid")) != null) {
                                oid = request.getParameter("oid");
                        }
```

```
                                Empresa v_empresa = new Empresa();
                                v_empresa.setCd_emp(beanForm.getEmpresa().getCd_emp());
                                beanForm.setEmpresa_lista(
                                        beanSQL.select(con, 0, v_empresa, 1, "", "N"));
                                if (beanForm.getEmpresa_lista().size() > 0) {
                                        throw new ExcecaoGeral(
                                                new ActionMessage("nota_fluxo.inserir.erro"));
                                } else {
                                        beanSQL.insert(beanForm.getEmpresa(), con);
                                        con.commit();
                                        beanForm.setEmpresa_lista(

        beanSQL.select(con,0,beanForm.getEmpresa(),1,"","N"));
                                        beanForm.setEmpresa_pesq(new Empresa());
                                        beanForm.setEmpresa_lista(
                                                beanSQL.select(
                                                        con,
                                                        0,
                                                        beanForm.getEmpresa_pesq(),
                                                        2000000000,
                                                        ((Empresa)
beanForm.getEmpresa_lista().get(0))
                                                                .getOid(), "N"));
                                        //
                                        beanForm.setNr_reg(1);
                                        p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
                                        beanForm.setEmpresa_lista(
                                                beanSQL.select(
                                                        con,
                                                        p_pagina,
                                                        beanForm.getEmpresa_pesq(),
                                                        beanForm.getNr_reg(),
                                                        "", ""));
                                        beanForm.setContadoresPaginas(
                                                beanSQL.getContaSelect(),
                                                beanSQL.getContaSelectAtual());
                                }
                        } catch (ExcecaoGeral e) {
                                msgs.add(ActionMessages.GLOBAL_MESSAGE, e.getAction_message());
                        } catch (Exception e) {
                                msgs.add(
                                        ActionMessages.GLOBAL_MESSAGE,
                                        new ActionMessage("error.sql", e.toString()));
                        }
                }
                if (!(msgs.isEmpty())) {
                        con.rollback();
                        saveMessages(request, msgs);
                }
                return (mapping.findForward("lista"));
        }

        public ActionForward entra_inserir(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                EmpresaForm beanForm = (EmpresaForm) form;
                beanForm.setEmpresa(new Empresa());
                beanForm.setMostraConsulta("N");
                return (mapping.findForward("editar"));
        }

        public ActionForward apagar(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                ActionMessages msgs = new ActionMessages();
                Connection con =
                        (Connection) request.getSession().getAttribute("conexao");
                EmpresaForm beanForm = (EmpresaForm) form;
                EmpresaSQL beanSQL = new EmpresaSQL();
                try {
                        int p_pagina = 0;
                        if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                                p_pagina = 0;
                        } else {
                                p_pagina =
                                        (int) Integer.parseInt(request.getParameter("p_pagina"));
                        }

                        String oid = "-1";
                        if (Util.trataNulo(request.getParameter("oid")) != null) {
                                oid = request.getParameter("oid");
                        }
```

```
                Empresa v_empresa = new Empresa();
                v_empresa.setOid(oid);
                v_empresa =
                        (Empresa) beanSQL.select(con, 0, v_empresa, 1, "","N").get(0);

                beanForm.setEmpresa_lista(
                        beanSQL.select(
                                con,
                                0,
                                beanForm.getEmpresa_pesq(),
                                2000000000,
                                oid, ""));
                p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());

                beanSQL.delete(oid, con);
                con.commit();

                request.setAttribute("oid", "");
        } catch (Exception e) {
                msgs.add(
                        ActionMessages.GLOBAL_MESSAGE,
                        new ActionMessage("error.sql", e.toString()));
        }
        if (!(msgs.isEmpty())) {
                con.rollback();
                saveMessages(request, msgs);
        }
        return (lista(mapping, beanForm, request, response));
}

public ActionForward entra_alterar(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
        ActionMessages msgs = new ActionMessages();
        Connection con =
                (Connection) request.getSession().getAttribute("conexao");
        EmpresaForm beanForm = (EmpresaForm) form;
        try {
                EmpresaSQL beanSQL = new EmpresaSQL();

                String oid = "-1";
                if (Util.trataNulo(request.getParameter("oid")) != null) {
                        oid = request.getParameter("oid");
                }

                Empresa v_empresa = new Empresa();
                v_empresa.setOid(oid);
                beanForm.setEmpresa_lista(beanSQL.select(con,0,v_empresa,1,"","N"));
                beanForm.setEmpresa((Empresa) beanForm.getEmpresa_lista().get(0));
        } catch (Exception e) {
                msgs.add(
                        ActionMessages.GLOBAL_MESSAGE,
                        new ActionMessage("error.sql", e.toString()));
        }
        if (!(msgs.isEmpty())) {
                con.rollback();
                saveMessages(request, msgs);
        }
        return (mapping.findForward("editar"));
}

public ActionForward alterar(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
        ActionMessages msgs = new ActionMessages();
        Connection con =
                (Connection) request.getSession().getAttribute("conexao");
        EmpresaForm beanForm = (EmpresaForm) form;
        EmpresaSQL beanSQL = new EmpresaSQL();
        msgs = beanForm.validate(mapping, request);
        if (msgs.isEmpty()) {
                try {
                        int p_pagina = 0;
                        if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                                p_pagina = 0;
                        } else {
                                p_pagina =
                                        (int) Integer.parseInt(
                                                request.getParameter("p_pagina"));
                        }
```

```
                                String oid = "-1";
                                if (Util.trataNulo(request.getParameter("oid")) != null) {
                                        oid = request.getParameter("oid");
                                }

                                beanSQL.update(beanForm.getEmpresa(), con);
                                con.commit();

                                beanForm.setEmpresa_lista(
                                        beanSQL.select(con, 0, beanForm.getEmpresa(), 1, "", "N"));
                                if (beanForm.getEmpresa_lista().size() == 0) {
                                        throw new ExcecaoGeral(
                                                new ActionMessage("nota_fluxo.alterar.erro"));
                                }

                                beanForm.setEmpresa_lista(
                                        beanSQL.select(
                                                con,
                                                0,
                                                beanForm.getEmpresa_pesq(),
                                                2000000000,
                                                beanForm.getEmpresa().getOid(), "N"));


                                //
                                beanForm.setNr_reg(1);
                                p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
                                beanForm.setEmpresa_lista(
                                        beanSQL.select(
                                                con,
                                                p_pagina,
                                                beanForm.getEmpresa_pesq(),
                                                beanForm.getNr_reg(),
                                                "",""));
                                beanForm.setContadoresPaginas(
                                        beanSQL.getContaSelect(),
                                        beanSQL.getContaSelectAtual());


                        } catch (ExcecaoGeral e) {
                                msgs.add(ActionMessages.GLOBAL_MESSAGE, e.getAction_message());
                        } catch (Exception e) {
                                msgs.add(
                                        ActionMessages.GLOBAL_MESSAGE,
                                        new ActionMessage("error.sql", e.toString()));
                        }
                }
                if (!(msgs.isEmpty())) {
                        con.rollback();
                        saveMessages(request, msgs);
                }
                return (mapping.findForward("lista"));
        }
}
```

Movimento_contasAction.java

```
package br.com.nota_fluxo.action;

import java.sql.Connection;
import java.util.HashMap;
import java.util.Map;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.log4j.Category;

import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;
import org.apache.struts.action.ActionMessages;
import org.apache.struts.actions.LookupDispatchAction;

import br.com.nota_fluxo.bean.Movimento_contas;
import br.com.nota_fluxo.form.Movimento_contasForm;
import br.com.nota_fluxo.sql.Movimento_contasSQL;
import br.com.nota_fluxo.util.Util;

public class Movimento_contasAction extends LookupDispatchAction {

        public Category cat = Category.getInstance(Movimento_contasAction.class.getName());

        protected Map getKeyMethodMap(){
                Map map = new HashMap();
                map.put("nota_fluxo.entra_inserir","entra_inserir");
                map.put("nota_fluxo.entra_consultar","entra_consultar");
```

```
        map.put("nota_fluxo.entra_alterar","entra_alterar");
        map.put("nota_fluxo.inserir","inserir");
        map.put("nota_fluxo.apagar","apagar");
        map.put("nota_fluxo.alterar","alterar");
        map.put("nota_fluxo.lista","lista");
        map.put("nota_fluxo.login.valida","lista");
        return map;
}

public ActionForward unspecified(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
        return (lista(mapping, form, request, response));
}

public ActionForward lista(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
        ActionMessages msgs = new ActionMessages();

        Connection con =
                (Connection) request.getSession().getAttribute("conexao");

        //UsuarioForm usuario_form =
        //       (UsuarioForm) request.getSession().getAttribute("usuarioForm");
        //cat.warn("movimento_contas "+usuario_form.getUsuario().getCd_emp());

        Movimento_contasForm beanForm = (Movimento_contasForm) form;
        Movimento_contasSQL beanSQL = new Movimento_contasSQL();
        try {
                beanForm.setNr_reg(10); //Numero de registros da lista
                beanForm.setMostraConsulta("N");

                int p_pagina = 0;
                if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                        p_pagina = 0;
                } else {
                        p_pagina =
                                (int) Integer.parseInt(request.getParameter("p_pagina"));
                }
                beanForm.setMovimento_contas_lista(
                        beanSQL.select(
                                con,
                                p_pagina,
                                beanForm.getMovimento_contas_pesq(),
                                beanForm.getNr_reg(),
                                "", ""));
                beanForm.setContadoresPaginas(
                        beanSQL.getContaSelect(),
                        beanSQL.getContaSelectAtual());
        } catch (Exception e) {
                msgs.add(
                        ActionMessages.GLOBAL_MESSAGE,
                        new ActionMessage("error.sql", e));
        }
        if (!(msgs.isEmpty())) {
                saveMessages(request, msgs);
        }
        return (mapping.findForward("lista"));
}

public ActionForward entra_consultar(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) {
        Movimento_contasForm beanForm = (Movimento_contasForm) form;
        beanForm.setMostraConsulta("S");
        return (mapping.findForward("editar"));
}

public ActionForward inserir(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
        ActionMessages msgs = new ActionMessages();
        Connection con =
                (Connection) request.getSession().getAttribute("conexao");
        Movimento_contasForm beanForm = (Movimento_contasForm) form;
        Movimento_contasSQL beanSQL = new Movimento_contasSQL();
```

```
                msgs = beanForm.validate(mapping, request);
                if (msgs.isEmpty()) {
                        try {
                                int p_pagina = 0;

                                if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                                        p_pagina = 0;
                                } else {
                                        p_pagina =
                                                (int) Integer.parseInt(
                                                        request.getParameter("p_pagina"));
                                }
                                String oid = "-1";
                                if (Util.trataNulo(request.getParameter("oid")) != null) {
                                        oid = request.getParameter("oid");
                                }

                                Movimento_contas v_movimento_contas = new Movimento_contas();

        v_movimento_contas.setDt_movto(beanForm.getMovimento_contas().getDt_movto());

        v_movimento_contas.setNr_movto(beanForm.getMovimento_contas().getNr_movto());
                                beanForm.setMovimento_contas_lista(
                                        beanSQL.select(con, 0, v_movimento_contas, 1, "", "N"));
                                if (beanForm.getMovimento_contas_lista().size() > 0) {
                                        throw new ExcecaoGeral(
                                                new ActionMessage("nota_fluxo.inserir.erro"));
                                } else {
                                        beanSQL.insert(beanForm.getMovimento_contas(), con);
                                        con.commit();
                                        beanForm.setMovimento_contas_lista(

        beanSQL.select(con,0,beanForm.getMovimento_contas(),1,"","N"));
                                        beanForm.setMovimento_contas_pesq(new Movimento_contas());
                                        beanForm.setMovimento_contas_lista(
                                                beanSQL.select(
                                                        con,
                                                        0,
                                                        beanForm.getMovimento_contas_pesq(),
                                                        2000000000,
                                                        ((Movimento_contas)
beanForm.getMovimento_contas_lista().get(0))
                                                                        .getOid(), "N"));
                                        //
                                        beanForm.setNr_reg(10);
                                        p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
                                        beanForm.setMovimento_contas_lista(
                                                beanSQL.select(
                                                        con,
                                                        p_pagina,
                                                        beanForm.getMovimento_contas_pesq(),
                                                        beanForm.getNr_reg(),
                                                        "", ""));
                                        beanForm.setContadoresPaginas(
                                                beanSQL.getContaSelect(),
                                                beanSQL.getContaSelectAtual());
                                }
                        } catch (ExcecaoGeral e) {
                                msgs.add(ActionMessages.GLOBAL_MESSAGE, e.getAction_message());
                        } catch (Exception e) {
                                msgs.add(
                                        ActionMessages.GLOBAL_MESSAGE,
                                        new ActionMessage("error.sql", e.toString()));
                        }
                }
                if (!(msgs.isEmpty())) {
                        con.rollback();
                        saveMessages(request, msgs);
                }
                return (mapping.findForward("lista"));
        }

        public ActionForward entra_inserir(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                Movimento_contasForm beanForm = (Movimento_contasForm) form;
                beanForm.setMovimento_contas(new Movimento_contas());
                beanForm.setMostraConsulta("N");
                return (mapping.findForward("editar"));
        }

        public ActionForward apagar(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
```

```
                HttpServletResponse response)
                throws Exception {
                ActionMessages msgs = new ActionMessages();
                Connection con =
                        (Connection) request.getSession().getAttribute("conexao");
                Movimento_contasForm beanForm = (Movimento_contasForm) form;
                Movimento_contasSQL beanSQL = new Movimento_contasSQL();
                try {
                        int p_pagina = 0;
                        if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                                p_pagina = 0;
                        } else {
                                p_pagina =
                                        (int) Integer.parseInt(request.getParameter("p_pagina"));
                        }

                        String oid = "-1";
                        if (Util.trataNulo(request.getParameter("oid")) != null) {
                                oid = request.getParameter("oid");
                        }

                        Movimento_contas v_movimento_contas = new Movimento_contas();
                        v_movimento_contas.setOid(oid);
                        v_movimento_contas =
                                (Movimento_contas) beanSQL.select(con, 0, v_movimento_contas, 1,
"","N").get(0);

                        beanForm.setMovimento_contas_lista(
                                beanSQL.select(
                                        con,
                                        0,
                                        beanForm.getMovimento_contas_pesq(),
                                        2000000000,
                                        oid, ""));
                        p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());

                        beanSQL.delete(oid, con);
                        con.commit();

                        request.setAttribute("oid", "");
                } catch (Exception e) {
                        msgs.add(
                                ActionMessages.GLOBAL_MESSAGE,
                                new ActionMessage("error.sql", e.toString()));
                }
                if (!(msgs.isEmpty())) {
                        con.rollback();
                        saveMessages(request, msgs);
                }
                return (lista(mapping, beanForm, request, response));
        }

        public ActionForward entra_alterar(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                ActionMessages msgs = new ActionMessages();
                Connection con =
                        (Connection) request.getSession().getAttribute("conexao");
                Movimento_contasForm beanForm = (Movimento_contasForm) form;
                try {
                        Movimento_contasSQL beanSQL = new Movimento_contasSQL();

                        String oid = "-1";
                        if (Util.trataNulo(request.getParameter("oid")) != null) {
                                oid = request.getParameter("oid");
                        }

                        Movimento_contas v_movimento_contas = new Movimento_contas();
                        v_movimento_contas.setOid(oid);

        beanForm.setMovimento_contas_lista(beanSQL.select(con,0,v_movimento_contas,1,"","N"));
                        beanForm.setMovimento_contas((Movimento_contas)
beanForm.getMovimento_contas_lista().get(0));
                } catch (Exception e) {
                        msgs.add(
                                ActionMessages.GLOBAL_MESSAGE,
                                new ActionMessage("error.sql", e.toString()));
                }
                if (!(msgs.isEmpty())) {
                        con.rollback();
                        saveMessages(request, msgs);
                }
                return (mapping.findForward("editar"));
        }
```

```
        public ActionForward alterar(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                ActionMessages msgs = new ActionMessages();
                Connection con =
                        (Connection) request.getSession().getAttribute("conexao");
                Movimento_contasForm beanForm = (Movimento_contasForm) form;
                Movimento_contasSQL beanSQL = new Movimento_contasSQL();
                msgs = beanForm.validate(mapping, request);
                if (msgs.isEmpty()) {
                        try {
                                int p_pagina = 0;
                                if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                                        p_pagina = 0;
                                } else {
                                        p_pagina =
                                                (int) Integer.parseInt(
                                                        request.getParameter("p_pagina"));
                                }

                                String oid = "-1";
                                if (Util.trataNulo(request.getParameter("oid")) != null) {
                                        oid = request.getParameter("oid");
                                }

                                beanSQL.update(beanForm.getMovimento_contas(), con);
                                con.commit();

                                beanForm.setMovimento_contas_lista(
                                        beanSQL.select(con, 0, beanForm.getMovimento_contas(), 1,
""", "N"));
                                if (beanForm.getMovimento_contas_lista().size() == 0) {
                                        throw new ExcecaoGeral(
                                                new ActionMessage("nota_fluxo.alterar.erro"));
                                }

                                beanForm.setMovimento_contas_lista(
                                        beanSQL.select(
                                                con,
                                                0,
                                                beanForm.getMovimento_contas_pesq(),
                                                2000000000,
                                                beanForm.getMovimento_contas().getOid(), "N"));


                                //
                                beanForm.setNr_reg(10);
                                p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
                                beanForm.setMovimento_contas_lista(
                                        beanSQL.select(
                                                con,
                                                p_pagina,
                                                beanForm.getMovimento_contas_pesq(),
                                                beanForm.getNr_reg(),
                                                "",""));
                                beanForm.setContadoresPaginas(
                                        beanSQL.getContaSelect(),
                                        beanSQL.getContaSelectAtual());


                        } catch (ExcecaoGeral e) {
                                msgs.add(ActionMessages.GLOBAL_MESSAGE, e.getAction_message());
                        } catch (Exception e) {
                                msgs.add(
                                        ActionMessages.GLOBAL_MESSAGE,
                                        new ActionMessage("error.sql", e.toString()));
                        }
                }
                if (!(msgs.isEmpty())) {
                        con.rollback();
                        saveMessages(request, msgs);
                }
                return (mapping.findForward("lista"));
        }
}
```

Nota_fiscalAction.java

```
package br.com.nota_fluxo.action;

import java.sql.Connection;
import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;
```

```
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.log4j.Category;

import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;
import org.apache.struts.action.ActionMessages;
import org.apache.struts.actions.LookupDispatchAction;

import br.com.nota_fluxo.bean.Contas_receber;
import br.com.nota_fluxo.bean.Nota_fiscal;
import br.com.nota_fluxo.bean.Nota_item;
import br.com.nota_fluxo.form.Contas_receberForm;
import br.com.nota_fluxo.form.Nota_fiscalForm;
import br.com.nota_fluxo.form.Nota_itemForm;
import br.com.nota_fluxo.sql.Contas_receberSQL;
import br.com.nota_fluxo.sql.Nota_fiscalSQL;
import br.com.nota_fluxo.sql.Nota_itemSQL;
import br.com.nota_fluxo.util.Util;

public class Nota_fiscalAction extends LookupDispatchAction {

        public Category cat = Category.getInstance(Nota_fiscalAction.class.getName());

        protected Map getKeyMethodMap(){
                Map map = new HashMap();
                map.put("nota_fluxo.entra_inserir","entra_inserir");
                map.put("nota_fluxo.entra_consultar","entra_consultar");
                map.put("nota_fluxo.entra_alterar","entra_alterar");
                map.put("nota_fluxo.inserir","inserir");
                map.put("nota_fluxo.apagar","apagar");
                map.put("nota_fluxo.alterar","alterar");
                map.put("nota_fluxo.lista","lista");
                map.put("nota_fluxo.login.valida","lista");
                map.put("nota_fluxo.confirmar","confirmar_nota");
                return map;
        }

        public ActionForward unspecified(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                return (lista(mapping, form, request, response));
        }

        public ActionForward lista(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                ActionMessages msgs = new ActionMessages();

                Connection con =
                        (Connection) request.getSession().getAttribute("conexao");

                //UsuarioForm usuario_form =
                //      (UsuarioForm) request.getSession().getAttribute("usuarioForm");
                //cat.warn("nota_fiscal "+usuario_form.getUsuario().getCd_emp());

                Nota_fiscalForm beanForm = (Nota_fiscalForm) form;
                Nota_fiscalSQL beanSQL = new Nota_fiscalSQL();
                try {
                        beanForm.setNr_reg(1); //Numero de registros da lista
                        beanForm.setMostraConsulta("N");

                        int p_pagina = 0;
                        if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                                p_pagina = 0;
                        } else {
                                p_pagina =
                                        (int) Integer.parseInt(request.getParameter("p_pagina"));
                        }
                        beanForm.setNota_fiscal_lista(
                                beanSQL.select(
                                        con,
                                        p_pagina,
                                        beanForm.getNota_fiscal_pesq(),
                                        beanForm.getNr_reg(),
                                        "", ""));
                        beanForm.setContadoresPaginas(
                                beanSQL.getContaSelect(),
                                beanSQL.getContaSelectAtual());
```

```
                } catch (Exception e) {
                        msgs.add(
                                ActionMessages.GLOBAL_MESSAGE,
                                new ActionMessage("error.sql", e));
                }
                if (!(msgs.isEmpty())) {
                        saveMessages(request, msgs);
                }
                return (mapping.findForward("lista"));
        }

        public ActionForward entra_consultar(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response) {
                Nota_fiscalForm beanForm = (Nota_fiscalForm) form;
                beanForm.setMostraConsulta("S");
                return (mapping.findForward("editar"));
        }

        public ActionForward inserir(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                ActionMessages msgs = new ActionMessages();
                Connection con =
                        (Connection) request.getSession().getAttribute("conexao");
                Nota_fiscalForm beanForm = (Nota_fiscalForm) form;
                Nota_fiscalSQL beanSQL = new Nota_fiscalSQL();
                msgs = beanForm.validate(mapping, request);
                if (msgs.isEmpty()) {
                        try {
                                int p_pagina = 0;

                                if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                                        p_pagina = 0;
                                } else {
                                        p_pagina =
                                                (int) Integer.parseInt(
                                                        request.getParameter("p_pagina"));
                                }
                                String oid = "-1";
                                if (Util.trataNulo(request.getParameter("oid")) != null) {
                                        oid = request.getParameter("oid");
                                }

                                Nota_fiscal v_nota_fiscal = new Nota_fiscal();
                                v_nota_fiscal.setCd_emp(beanForm.getNota_fiscal().getCd_emp());
                                v_nota_fiscal.setNr_nota(beanForm.getNota_fiscal().getNr_nota());
                                beanForm.setNota_fiscal_lista(
                                        beanSQL.select(con, 0, v_nota_fiscal, 1, "", "N"));
                                if (beanForm.getNota_fiscal_lista().size() > 0) {
                                        throw new ExcecaoGeral(
                                                new ActionMessage("nota_fluxo.inserir.erro"));
                                } else {
                                        beanSQL.insert(beanForm.getNota_fiscal(), con);
                                        con.commit();
                                        beanForm.setNota_fiscal_lista(

        beanSQL.select(con,0,beanForm.getNota_fiscal(),1,"","N"));
                                        beanForm.setNota_fiscal_pesq(new Nota_fiscal());
                                        beanForm.setNota_fiscal_lista(
                                                beanSQL.select(
                                                        con,
                                                        0,
                                                        beanForm.getNota_fiscal_pesq(),
                                                        2000000000,
                                                        ((Nota_fiscal)
beanForm.getNota_fiscal_lista().get(0))
                                                                .getOid(), "N"));
                                        //
                                        beanForm.setNr_reg(1);
                                        p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
                                        beanForm.setNota_fiscal_lista(
                                                beanSQL.select(
                                                        con,
                                                        p_pagina,
                                                        beanForm.getNota_fiscal_pesq(),
                                                        beanForm.getNr_reg(),
                                                        "", ""));
                                        beanForm.setContadoresPaginas(
                                                beanSQL.getContaSelect(),
                                                beanSQL.getContaSelectAtual());
                                }
                        } catch (ExcecaoGeral e) {
```

```
                        msgs.add(ActionMessages.GLOBAL_MESSAGE, e.getAction_message());
                } catch (Exception e) {
                        msgs.add(
                                ActionMessages.GLOBAL_MESSAGE,
                                new ActionMessage("error.sql", e.toString()));
                }
        }
        if (!(msgs.isEmpty())) {
                con.rollback();
                saveMessages(request, msgs);
        }
        return (mapping.findForward("lista"));
}

public ActionForward entra_inserir(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
        Nota_fiscalForm beanForm = (Nota_fiscalForm) form;
        beanForm.setNota_fiscal(new Nota_fiscal());
        beanForm.setMostraConsulta("N");
        return (mapping.findForward("editar"));
}

public ActionForward apagar(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
        ActionMessages msgs = new ActionMessages();
        Connection con =
                (Connection) request.getSession().getAttribute("conexao");
        Nota_fiscalForm beanForm = (Nota_fiscalForm) form;
        Nota_fiscalSQL beanSQL = new Nota_fiscalSQL();
        try {
                int p_pagina = 0;
                if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                        p_pagina = 0;
                } else {
                        p_pagina =
                                (int) Integer.parseInt(request.getParameter("p_pagina"));
                }

                String oid = "-1";
                if (Util.trataNulo(request.getParameter("oid")) != null) {
                        oid = request.getParameter("oid");
                }

                Nota_fiscal v_nota_fiscal = new Nota_fiscal();
                v_nota_fiscal.setOid(oid);
                v_nota_fiscal =
                        (Nota_fiscal) beanSQL.select(con, 0, v_nota_fiscal, 1,
"","N").get(0);

                beanForm.setNota_fiscal_lista(
                        beanSQL.select(
                                con,
                                0,
                                beanForm.getNota_fiscal_pesq(),
                                2000000000,
                                oid, ""));
                p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());

                beanSQL.delete(oid, con);
                con.commit();

                request.setAttribute("oid", "");
        } catch (Exception e) {
                msgs.add(
                        ActionMessages.GLOBAL_MESSAGE,
                        new ActionMessage("error.sql", e.toString()));
        }
        if (!(msgs.isEmpty())) {
                con.rollback();
                saveMessages(request, msgs);
        }
        return (lista(mapping, beanForm, request, response));
}

public ActionForward entra_alterar(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
```

```
                ActionMessages msgs = new ActionMessages();
                Connection con =
                        (Connection) request.getSession().getAttribute("conexao");
                Nota_fiscalForm beanForm = (Nota_fiscalForm) form;
                try {
                        Nota_fiscalSQL beanSQL = new Nota_fiscalSQL();

                        String oid = "-1";
                        if (Util.trataNulo(request.getParameter("oid")) != null) {
                                oid = request.getParameter("oid");
                        }

                        Nota_fiscal v_nota_fiscal = new Nota_fiscal();
                        v_nota_fiscal.setOid(oid);

        beanForm.setNota_fiscal_lista(beanSQL.select(con,0,v_nota_fiscal,1,"","N"));
                        beanForm.setNota_fiscal((Nota_fiscal)
beanForm.getNota_fiscal_lista().get(0));
                } catch (Exception e) {
                        msgs.add(
                                ActionMessages.GLOBAL_MESSAGE,
                                new ActionMessage("error.sql", e.toString()));
                }
                if (!(msgs.isEmpty())) {
                        con.rollback();
                        saveMessages(request, msgs);
                }
                return (mapping.findForward("editar"));
        }

        public ActionForward alterar(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                ActionMessages msgs = new ActionMessages();
                Connection con =
                        (Connection) request.getSession().getAttribute("conexao");
                Nota_fiscalForm beanForm = (Nota_fiscalForm) form;
                Nota_fiscalSQL beanSQL = new Nota_fiscalSQL();
                msgs = beanForm.validate(mapping, request);
                if (msgs.isEmpty()) {
                        try {
                                int p_pagina = 0;
                                if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                                        p_pagina = 0;
                                } else {
                                        p_pagina =
                                                (int) Integer.parseInt(
                                                        request.getParameter("p_pagina"));
                                }

                                String oid = "-1";
                                if (Util.trataNulo(request.getParameter("oid")) != null) {
                                        oid = request.getParameter("oid");
                                }

                                beanSQL.update(beanForm.getNota_fiscal(), con);
                                con.commit();

                                beanForm.setNota_fiscal_lista(
                                        beanSQL.select(con, 0, beanForm.getNota_fiscal(), 1, "",
"N"));
                                if (beanForm.getNota_fiscal_lista().size() == 0) {
                                        throw new ExcecaoGeral(
                                                new ActionMessage("nota_fluxo.alterar.erro"));
                                }

                                beanForm.setNota_fiscal_lista(
                                        beanSQL.select(
                                                con,
                                                0,
                                                beanForm.getNota_fiscal_pesq(),
                                                2000000000,
                                                beanForm.getNota_fiscal().getOid(), "N"));


                                //
                                beanForm.setNr_reg(1);
                                p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
                                beanForm.setNota_fiscal_lista(
                                        beanSQL.select(
                                                con,
                                                p_pagina,
                                                beanForm.getNota_fiscal_pesq(),
                                                beanForm.getNr_reg(),
                                                "",""));
```

```
                                beanForm.setContadoresPaginas(
                                        beanSQL.getContaSelect(),
                                        beanSQL.getContaSelectAtual());


                        } catch (ExcecaoGeral e) {
                                msgs.add(ActionMessages.GLOBAL_MESSAGE, e.getAction_message());
                        } catch (Exception e) {
                                msgs.add(
                                        ActionMessages.GLOBAL_MESSAGE,
                                        new ActionMessage("error.sql", e.toString()));
                        }
                }
                if (!(msgs.isEmpty())) {
                        con.rollback();
                        saveMessages(request, msgs);
                }
                return (mapping.findForward("lista"));
        }

        public ActionForward confirmar_nota(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                ActionMessages msgs = new ActionMessages();
                Connection con =
                        (Connection) request.getSession().getAttribute("conexao");
                Nota_fiscalForm beanForm = (Nota_fiscalForm) form;
                Nota_fiscalSQL beanSQL = new Nota_fiscalSQL();
                msgs = beanForm.validate(mapping, request);
                if (msgs.isEmpty()) {
                        try {
                                int p_pagina = 0;
                                if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                                        p_pagina = 0;
                                } else {
                                        p_pagina =
                                                (int) Integer.parseInt(
                                                        request.getParameter("p_pagina"));
                                }

                                String oid = "-1";
                                if (Util.trataNulo(request.getParameter("oid")) != null) {
                                        oid = request.getParameter("oid");
                                }
                                //selecionar o q estiver com esse oid



                                beanForm.setNota_fiscal_lista(
                                        beanSQL.select(con, 0, beanForm.getNota_fiscal(), 1, "",
"N"));
                                if (beanForm.getNota_fiscal_lista().size() == 0) {
                                        throw new ExcecaoGeral(
                                                new ActionMessage("nota_fluxo.alterar.erro"));
                                }


        beanForm.setNota_fiscal((Nota_fiscal)beanForm.getNota_fiscal_lista().get(0));

                                Nota_itemForm v_nota_item_form = new Nota_itemForm();
                                v_nota_item_form.getNota_item().setCd_emp(
                                        ((Nota_fiscal)
beanForm.getNota_fiscal_lista().get(0)).getCd_emp());
                                v_nota_item_form.getNota_item().setNr_nota(
                                        ((Nota_fiscal)
beanForm.getNota_fiscal_lista().get(0)).getNr_nota());
                                Nota_itemSQL nota_itemSQL = new Nota_itemSQL();

                                v_nota_item_form.setNota_item_lista(
                                        nota_itemSQL.select(
                                                con,
                                                0,
                                                v_nota_item_form.getNota_item(),
                                                200000,
                                                "",
                                                "N"));

                                Iterator iter = v_nota_item_form.getNota_item_lista().iterator();
                                Nota_item v_nota_item = new Nota_item();
                                float total = 0;
                                while (iter.hasNext()){
                                        v_nota_item = (Nota_item)iter.next();
                                        total = total + Float.parseFloat(v_nota_item.getVl_item());
                                        //v_nota_item.getNota_item_lista()
```

```
                                }
                                cat.warn("total: "+total);

                                beanForm.getNota_fiscal().setVl_liquido(total+"");
                                beanForm.getNota_fiscal().setVl_bruto(total+"");

                                beanSQL.update(beanForm.getNota_fiscal(), con);


                                beanForm.setNota_fiscal_lista(
                                        beanSQL.select(
                                                con,
                                                0,
                                                beanForm.getNota_fiscal_pesq(),
                                                2000000000,
                                                beanForm.getNota_fiscal().getOid(), "N"));


                                //
                                beanForm.setNr_reg(1);
                                p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
                                beanForm.setNota_fiscal_lista(
                                        beanSQL.select(
                                                con,
                                                p_pagina,
                                                beanForm.getNota_fiscal_pesq(),
                                                beanForm.getNr_reg(),
                                                "",""));
                                beanForm.setContadoresPaginas(
                                        beanSQL.getContaSelect(),
                                        beanSQL.getContaSelectAtual());

                                //Cria conta a receber
                                Contas_receberForm contas_receber_form = new Contas_receberForm();
                                Contas_receber contas_receber = new Contas_receber();
                                Contas_receberSQL contas_receberSQL = new Contas_receberSQL();

                                String aux_cd_receber = "";
                                contas_receber.setCd_emp(beanForm.getNota_fiscal().getCd_emp());
                                contas_receber =
                                        (Contas_receber) contas_receberSQL
                                                .select(con, 0, contas_receber, 1, "",
"N").get(0);
                                aux_cd_receber = contas_receber.getCd_receber();
                                if  (Util.trataNulo(aux_cd_receber)==null){
                                        aux_cd_receber = "0";
                                } else {
                                        aux_cd_receber = (Integer.parseInt(aux_cd_receber) + 1)+"";
                                }
                                contas_receber = new Contas_receber();
                                contas_receber.setCd_emp(beanForm.getNota_fiscal().getCd_emp());
                                contas_receber.setCd_receber(aux_cd_receber);

        contas_receber.setDt_vencto(beanForm.getNota_fiscal().getDt_vencto());
                                contas_receber.setDt_pagto(null);
                                contas_receber.setVl_receber(total+"");
                                contas_receber.setVl_recebido(null);
                                contas_receber.setNr_nota(beanForm.getNota_fiscal().getNr_nota());
                                contas_receber.setCd_tp_receber("1");//fixo para notas

        contas_receber.setCd_conta(beanForm.getNota_fiscal().getCd_conta());

                                contas_receberSQL.insert(contas_receber,con);

                                con.commit();
                        } catch (ExcecaoGeral e) {
                                msgs.add(ActionMessages.GLOBAL_MESSAGE, e.getAction_message());
                        } catch (Exception e) {
                                msgs.add(
                                        ActionMessages.GLOBAL_MESSAGE,
                                        new ActionMessage("error.sql", e.toString()));
                        }
                }
                if (!(msgs.isEmpty())) {
                        con.rollback();
                        saveMessages(request, msgs);
                }
                return (mapping.findForward("lista"));
        }
}
```

Nota_itemAction.java

```
package br.com.nota_fluxo.action;

import java.sql.Connection;
import java.util.HashMap;
import java.util.Map;
```

```
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.log4j.Category;

import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;
import org.apache.struts.action.ActionMessages;
import org.apache.struts.actions.LookupDispatchAction;

import br.com.nota_fluxo.bean.Nota_item;
import br.com.nota_fluxo.form.Nota_itemForm;
import br.com.nota_fluxo.sql.Nota_itemSQL;
import br.com.nota_fluxo.util.Util;

public class Nota_itemAction extends LookupDispatchAction {

	public Category cat = Category.getInstance(Nota_itemAction.class.getName());

	protected Map getKeyMethodMap(){
		Map map = new HashMap();
		map.put("nota_fluxo.entra_inserir","entra_inserir");
		map.put("nota_fluxo.entra_consultar","entra_consultar");
		map.put("nota_fluxo.entra_alterar","entra_alterar");
		map.put("nota_fluxo.inserir","inserir");
		map.put("nota_fluxo.apagar","apagar");
		map.put("nota_fluxo.alterar","alterar");
		map.put("nota_fluxo.lista","lista");
		map.put("nota_fluxo.login.valida","lista");
		return map;
	}

	public ActionForward unspecified(
		ActionMapping mapping,
		ActionForm form,
		HttpServletRequest request,
		HttpServletResponse response)
		throws Exception {
		return (lista(mapping, form, request, response));
	}

	public ActionForward lista(
		ActionMapping mapping,
		ActionForm form,
		HttpServletRequest request,
		HttpServletResponse response)
		throws Exception {
		ActionMessages msgs = new ActionMessages();

		Connection con =
			(Connection) request.getSession().getAttribute("conexao");

		//UsuarioForm usuario_form =
		//	(UsuarioForm) request.getSession().getAttribute("usuarioForm");
		//cat.warn("nota_item "+usuario_form.getUsuario().getCd_emp());

		Nota_itemForm beanForm = (Nota_itemForm) form;
		Nota_itemSQL beanSQL = new Nota_itemSQL();
		try {
			beanForm.setNr_reg(10); //Numero de registros da lista
			beanForm.setMostraConsulta("N");

			int p_pagina = 0;
			if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
				p_pagina = 0;
			} else {
				p_pagina =
					(int) Integer.parseInt(request.getParameter("p_pagina"));
			}
			beanForm.setNota_item_lista(
				beanSQL.select(
					con,
					p_pagina,
					beanForm.getNota_item_pesq(),
					beanForm.getNr_reg(),
					"", ""));
			beanForm.setContadoresPaginas(
				beanSQL.getContaSelect(),
				beanSQL.getContaSelectAtual());
		} catch (Exception e) {
			msgs.add(
				ActionMessages.GLOBAL_MESSAGE,
				new ActionMessage("error.sql", e));
		}
		if (!(msgs.isEmpty())) {
```

```
                        saveMessages(request, msgs);
                }
                return (mapping.findForward("lista"));
        }

        public ActionForward entra_consultar(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response) {
                Nota_itemForm beanForm = (Nota_itemForm) form;
                beanForm.setMostraConsulta("S");
                return (mapping.findForward("editar"));
        }

        public ActionForward inserir(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                ActionMessages msgs = new ActionMessages();
                Connection con =
                        (Connection) request.getSession().getAttribute("conexao");
                Nota_itemForm beanForm = (Nota_itemForm) form;
                Nota_itemSQL beanSQL = new Nota_itemSQL();
                msgs = beanForm.validate(mapping, request);
                if (msgs.isEmpty()) {
                        try {
                                int p_pagina = 0;

                                if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                                        p_pagina = 0;
                                } else {
                                        p_pagina =
                                                (int) Integer.parseInt(
                                                        request.getParameter("p_pagina"));
                                }
                                String oid = "-1";
                                if (Util.trataNulo(request.getParameter("oid")) != null) {
                                        oid = request.getParameter("oid");
                                }

                                Nota_item v_nota_item = new Nota_item();
                                v_nota_item.setCd_emp(beanForm.getNota_item().getCd_emp());
                                v_nota_item.setNr_nota(beanForm.getNota_item().getNr_nota());
                                v_nota_item.setCd_cnae(beanForm.getNota_item().getCd_cnae());
                                beanForm.setNota_item_lista(
                                        beanSQL.select(con, 0, v_nota_item, 1, "", "N"));
                                if (beanForm.getNota_item_lista().size() > 0) {
                                        throw new ExcecaoGeral(
                                                new ActionMessage("nota_fluxo.inserir.erro"));
                                } else {
                                        beanSQL.insert(beanForm.getNota_item(), con);
                                        con.commit();
                                        beanForm.setNota_item_lista(

        beanSQL.select(con,0,beanForm.getNota_item(),1,"","N"));
                                        beanForm.setNota_item_pesq(new Nota_item());
                                        beanForm.setNota_item_lista(
                                                beanSQL.select(
                                                        con,
                                                        0,
                                                        beanForm.getNota_item_pesq(),
                                                        2000000000,
                                                        ((Nota_item)
beanForm.getNota_item_lista().get(0))
                                                                        .getOid(), "N"));
                                        //
                                        beanForm.setNr_reg(10);
                                        p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
                                        beanForm.setNota_item_lista(
                                                beanSQL.select(
                                                        con,
                                                        p_pagina,
                                                        beanForm.getNota_item_pesq(),
                                                        beanForm.getNr_reg(),
                                                        "", ""));
                                        beanForm.setContadoresPaginas(
                                                beanSQL.getContaSelect(),
                                                beanSQL.getContaSelectAtual());
                                }
                        } catch (ExcecaoGeral e) {
                                msgs.add(ActionMessages.GLOBAL_MESSAGE, e.getAction_message());
                        } catch (Exception e) {
                                msgs.add(
                                        ActionMessages.GLOBAL_MESSAGE,
                                        new ActionMessage("error.sql", e.toString()));
```

```
                }
        }
        if (!(msgs.isEmpty())) {
                con.rollback();
                saveMessages(request, msgs);
        }
        return (mapping.findForward("lista"));
}

public ActionForward entra_inserir(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
        Nota_itemForm beanForm = (Nota_itemForm) form;
        //beanForm.setNota_item(new Nota_item());
        beanForm.setMostraConsulta("N");
        return (mapping.findForward("editar"));
}

public ActionForward apagar(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
        ActionMessages msgs = new ActionMessages();
        Connection con =
                (Connection) request.getSession().getAttribute("conexao");
        Nota_itemForm beanForm = (Nota_itemForm) form;
        Nota_itemSQL beanSQL = new Nota_itemSQL();
        try {
                int p_pagina = 0;
                if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                        p_pagina = 0;
                } else {
                        p_pagina =
                                (int) Integer.parseInt(request.getParameter("p_pagina"));
                }

                String oid = "-1";
                if (Util.trataNulo(request.getParameter("oid")) != null) {
                        oid = request.getParameter("oid");
                }

                Nota_item v_nota_item = new Nota_item();
                v_nota_item.setOid(oid);
                v_nota_item =
                         (Nota_item) beanSQL.select(con, 0, v_nota_item, 1, "","N").get(0);

                beanForm.setNota_item_lista(
                        beanSQL.select(
                                con,
                                0,
                                beanForm.getNota_item_pesq(),
                                2000000000,
                                oid, ""));
                p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());

                beanSQL.delete(oid, con);
                con.commit();

                request.setAttribute("oid", "");
        } catch (Exception e) {
                msgs.add(
                        ActionMessages.GLOBAL_MESSAGE,
                        new ActionMessage("error.sql", e.toString()));
        }
        if (!(msgs.isEmpty())) {
                con.rollback();
                saveMessages(request, msgs);
        }
        return (lista(mapping, beanForm, request, response));
}

public ActionForward entra_alterar(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
        ActionMessages msgs = new ActionMessages();
        Connection con =
                (Connection) request.getSession().getAttribute("conexao");
        Nota_itemForm beanForm = (Nota_itemForm) form;
        try {
                Nota_itemSQL beanSQL = new Nota_itemSQL();
```

```
                        String oid = "-1";
                        if (Util.trataNulo(request.getParameter("oid")) != null) {
                                oid = request.getParameter("oid");
                        }

                        Nota_item v_nota_item = new Nota_item();
                        v_nota_item.setOid(oid);
                        beanForm.setNota_item_lista(beanSQL.select(con,0,v_nota_item,1,"","N"));
                        beanForm.setNota_item((Nota_item) beanForm.getNota_item_lista().get(0));
                } catch (Exception e) {
                        msgs.add(
                                ActionMessages.GLOBAL_MESSAGE,
                                new ActionMessage("error.sql", e.toString()));
                }
                if (!(msgs.isEmpty())) {
                        con.rollback();
                        saveMessages(request, msgs);
                }
                return (mapping.findForward("editar"));
        }

        public ActionForward alterar(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                ActionMessages msgs = new ActionMessages();
                Connection con =
                        (Connection) request.getSession().getAttribute("conexao");
                Nota_itemForm beanForm = (Nota_itemForm) form;
                Nota_itemSQL beanSQL = new Nota_itemSQL();
                msgs = beanForm.validate(mapping, request);
                if (msgs.isEmpty()) {
                        try {
                                int p_pagina = 0;
                                if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                                        p_pagina = 0;
                                } else {
                                        p_pagina =
                                                (int) Integer.parseInt(
                                                        request.getParameter("p_pagina"));
                                }

                                String oid = "-1";
                                if (Util.trataNulo(request.getParameter("oid")) != null) {
                                        oid = request.getParameter("oid");
                                }

                                beanSQL.update(beanForm.getNota_item(), con);
                                con.commit();

                                beanForm.setNota_item_lista(
                                        beanSQL.select(con, 0, beanForm.getNota_item(), 1, "",
"N"));
                                if (beanForm.getNota_item_lista().size() == 0) {
                                        throw new ExcecaoGeral(
                                                new ActionMessage("nota_fluxo.alterar.erro"));
                                }

                                beanForm.setNota_item_lista(
                                        beanSQL.select(
                                                con,
                                                0,
                                                beanForm.getNota_item_pesq(),
                                                2000000000,
                                                beanForm.getNota_item().getOid(), "N"));


                                //
                                beanForm.setNr_reg(10);
                                p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
                                beanForm.setNota_item_lista(
                                        beanSQL.select(
                                                con,
                                                p_pagina,
                                                beanForm.getNota_item_pesq(),
                                                beanForm.getNr_reg(),
                                                "",""));
                                beanForm.setContadoresPaginas(
                                        beanSQL.getContaSelect(),
                                        beanSQL.getContaSelectAtual());


                        } catch (ExcecaoGeral e) {
                                msgs.add(ActionMessages.GLOBAL_MESSAGE, e.getAction_message());
                        } catch (Exception e) {
```

```
                                msgs.add(
                                        ActionMessages.GLOBAL_MESSAGE,
                                        new ActionMessage("error.sql", e.toString()));
                        }
                }
                if (!(msgs.isEmpty())) {
                        con.rollback();
                        saveMessages(request, msgs);
                }
                return (mapping.findForward("lista"));
        }
}
```

Ramo_AtividadeAction.java

```
package br.com.nota_fluxo.action;

import java.sql.Connection;
import java.util.HashMap;
import java.util.Map;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.log4j.Category;

import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;
import org.apache.struts.action.ActionMessages;
import org.apache.struts.actions.LookupDispatchAction;

import br.com.nota_fluxo.bean.Ramo_atividade;
import br.com.nota_fluxo.form.Ramo_atividadeForm;
import br.com.nota_fluxo.sql.Ramo_atividadeSQL;
import br.com.nota_fluxo.util.Util;

public class Ramo_atividadeAction extends LookupDispatchAction {

        public Category cat = Category.getInstance(Ramo_atividadeAction.class.getName());

        protected Map getKeyMethodMap(){
                Map map = new HashMap();
                map.put("nota_fluxo.entra_inserir","entra_inserir");
                map.put("nota_fluxo.entra_consultar","entra_consultar");
                map.put("nota_fluxo.entra_alterar","entra_alterar");
                map.put("nota_fluxo.inserir","inserir");
                map.put("nota_fluxo.apagar","apagar");
                map.put("nota_fluxo.alterar","alterar");
                map.put("nota_fluxo.lista","lista");
                map.put("nota_fluxo.login.valida","lista");
                return map;
        }

        public ActionForward unspecified(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                return (lista(mapping, form, request, response));
        }

        public ActionForward lista(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                ActionMessages msgs = new ActionMessages();

                Connection con =
                        (Connection) request.getSession().getAttribute("conexao");

                //UsuarioForm usuario_form =
                //      (UsuarioForm) request.getSession().getAttribute("usuarioForm");
                //cat.warn("ramo_atividade "+usuario_form.getUsuario().getCd_emp());

                Ramo_atividadeForm beanForm = (Ramo_atividadeForm) form;
                Ramo_atividadeSQL beanSQL = new Ramo_atividadeSQL();
                try {
                        beanForm.setNr_reg(10); //Numero de registros da lista
                        beanForm.setMostraConsulta("N");

                        int p_pagina = 0;
                        if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                                p_pagina = 0;
```

```
                        } else {
                                p_pagina =
                                        (int) Integer.parseInt(request.getParameter("p_pagina"));
                        }
                        beanForm.setRamo_atividade_lista(
                                beanSQL.select(
                                        con,
                                        p_pagina,
                                        beanForm.getRamo_atividade_pesq(),
                                        beanForm.getNr_reg(),
                                        "", ""));
                        beanForm.setContadoresPaginas(
                                beanSQL.getContaSelect(),
                                beanSQL.getContaSelectAtual());
                } catch (Exception e) {
                        msgs.add(
                                ActionMessages.GLOBAL_MESSAGE,
                                new ActionMessage("error.sql", e));
                }
                if (!(msgs.isEmpty())) {
                        saveMessages(request, msgs);
                }
                return (mapping.findForward("lista"));
        }

        public ActionForward entra_consultar(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response) {
                Ramo_atividadeForm beanForm = (Ramo_atividadeForm) form;
                beanForm.setMostraConsulta("S");
                return (mapping.findForward("editar"));
        }

        public ActionForward inserir(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                ActionMessages msgs = new ActionMessages();
                Connection con =
                        (Connection) request.getSession().getAttribute("conexao");
                Ramo_atividadeForm beanForm = (Ramo_atividadeForm) form;
                Ramo_atividadeSQL beanSQL = new Ramo_atividadeSQL();
                msgs = beanForm.validate(mapping, request);
                if (msgs.isEmpty()) {
                        try {
                                int p_pagina = 0;

                                if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                                        p_pagina = 0;
                                } else {
                                        p_pagina =
                                                (int) Integer.parseInt(
                                                        request.getParameter("p_pagina"));
                                }
                                String oid = "-1";
                                if (Util.trataNulo(request.getParameter("oid")) != null) {
                                        oid = request.getParameter("oid");
                                }

                                Ramo_atividade v_ramo_atividade = new Ramo_atividade();

        v_ramo_atividade.setCd_atividade(beanForm.getRamo_atividade().getCd_atividade());
                                beanForm.setRamo_atividade_lista(
                                        beanSQL.select(con, 0, v_ramo_atividade, 1, "", "N"));
                                if (beanForm.getRamo_atividade_lista().size() > 0) {
                                        throw new ExcecaoGeral(
                                                new ActionMessage("nota_fluxo.inserir.erro"));
                                } else {
                                        beanSQL.insert(beanForm.getRamo_atividade(), con);
                                        con.commit();
                                        beanForm.setRamo_atividade_lista(

        beanSQL.select(con,0,beanForm.getRamo_atividade(),1,"","N"));
                                        beanForm.setRamo_atividade_pesq(new Ramo_atividade());
                                        beanForm.setRamo_atividade_lista(
                                                beanSQL.select(
                                                        con,
                                                        0,
                                                        beanForm.getRamo_atividade_pesq(),
                                                        2000000000,
                                                        ((Ramo_atividade)
beanForm.getRamo_atividade_lista().get(0))
                                                                .getOid(), "N"));
                                        //
```

```
                                        beanForm.setNr_reg(10);
                                        p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
                                        beanForm.setRamo_atividade_lista(
                                                beanSQL.select(
                                                        con,
                                                        p_pagina,
                                                        beanForm.getRamo_atividade_pesq(),
                                                        beanForm.getNr_reg(),
                                                        "", ""));
                                        beanForm.setContadoresPaginas(
                                                beanSQL.getContaSelect(),
                                                beanSQL.getContaSelectAtual());
                                }
                        } catch (ExcecaoGeral e) {
                                msgs.add(ActionMessages.GLOBAL_MESSAGE, e.getAction_message());
                        } catch (Exception e) {
                                msgs.add(
                                        ActionMessages.GLOBAL_MESSAGE,
                                        new ActionMessage("error.sql", e.toString()));
                        }
                }
                if (!(msgs.isEmpty())) {
                        con.rollback();
                        saveMessages(request, msgs);
                }
                return (mapping.findForward("lista"));
        }

        public ActionForward entra_inserir(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                Ramo_atividadeForm beanForm = (Ramo_atividadeForm) form;
                beanForm.setRamo_atividade(new Ramo_atividade());
                beanForm.setMostraConsulta("N");
                return (mapping.findForward("editar"));
        }

        public ActionForward apagar(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                ActionMessages msgs = new ActionMessages();
                Connection con =
                        (Connection) request.getSession().getAttribute("conexao");
                Ramo_atividadeForm beanForm = (Ramo_atividadeForm) form;
                Ramo_atividadeSQL beanSQL = new Ramo_atividadeSQL();
                try {
                        int p_pagina = 0;
                        if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                                p_pagina = 0;
                        } else {
                                p_pagina =
                                        (int) Integer.parseInt(request.getParameter("p_pagina"));
                        }

                        String oid = "-1";
                        if (Util.trataNulo(request.getParameter("oid")) != null) {
                                oid = request.getParameter("oid");
                        }

                        Ramo_atividade v_ramo_atividade = new Ramo_atividade();
                        v_ramo_atividade.setOid(oid);
                        v_ramo_atividade =
                                (Ramo_atividade) beanSQL.select(con, 0, v_ramo_atividade, 1,
"","N").get(0);

                        beanForm.setRamo_atividade_lista(
                                beanSQL.select(
                                        con,
                                        0,
                                        beanForm.getRamo_atividade_pesq(),
                                        2000000000,
                                        oid, ""));
                        p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());

                        beanSQL.delete(oid, con);
                        con.commit();

                        request.setAttribute("oid", "");
                } catch (Exception e) {
                        msgs.add(
                                ActionMessages.GLOBAL_MESSAGE,
                                new ActionMessage("error.sql", e.toString()));
```

```
                }
                if (!(msgs.isEmpty())) {
                        con.rollback();
                        saveMessages(request, msgs);
                }
                return (lista(mapping, beanForm, request, response));
        }

        public ActionForward entra_alterar(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                ActionMessages msgs = new ActionMessages();
                Connection con =
                        (Connection) request.getSession().getAttribute("conexao");
                Ramo_atividadeForm beanForm = (Ramo_atividadeForm) form;
                try {
                        Ramo_atividadeSQL beanSQL = new Ramo_atividadeSQL();

                        String oid = "-1";
                        if (Util.trataNulo(request.getParameter("oid")) != null) {
                                oid = request.getParameter("oid");
                        }

                        Ramo_atividade v_ramo_atividade = new Ramo_atividade();
                        v_ramo_atividade.setOid(oid);

        beanForm.setRamo_atividade_lista(beanSQL.select(con,0,v_ramo_atividade,1,"","N"));
                        beanForm.setRamo_atividade((Ramo_atividade)
beanForm.getRamo_atividade_lista().get(0));
                } catch (Exception e) {
                        msgs.add(
                                ActionMessages.GLOBAL_MESSAGE,
                                new ActionMessage("error.sql", e.toString()));
                }
                if (!(msgs.isEmpty())) {
                        con.rollback();
                        saveMessages(request, msgs);
                }
                return (mapping.findForward("editar"));
        }

        public ActionForward alterar(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                ActionMessages msgs = new ActionMessages();
                Connection con =
                        (Connection) request.getSession().getAttribute("conexao");
                Ramo_atividadeForm beanForm = (Ramo_atividadeForm) form;
                Ramo_atividadeSQL beanSQL = new Ramo_atividadeSQL();
                msgs = beanForm.validate(mapping, request);
                if (msgs.isEmpty()) {
                        try {
                                int p_pagina = 0;
                                if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                                        p_pagina = 0;
                                } else {
                                        p_pagina =
                                                (int) Integer.parseInt(
                                                        request.getParameter("p_pagina"));
                                }

                                String oid = "-1";
                                if (Util.trataNulo(request.getParameter("oid")) != null) {
                                        oid = request.getParameter("oid");
                                }

                                beanSQL.update(beanForm.getRamo_atividade(), con);
                                con.commit();

                                beanForm.setRamo_atividade_lista(
                                        beanSQL.select(con, 0, beanForm.getRamo_atividade(), 1, "",
"N"));
                                if (beanForm.getRamo_atividade_lista().size() == 0) {
                                        throw new ExcecaoGeral(
                                                new ActionMessage("nota_fluxo.alterar.erro"));
                                }

                                beanForm.setRamo_atividade_lista(
                                        beanSQL.select(
                                                con,
                                                0,
                                                beanForm.getRamo_atividade_pesq(),
```

```
                                                2000000000,
                                                beanForm.getRamo_atividade().getOid(), "N"));

                                        //
                                        beanForm.setNr_reg(10);
                                        p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
                                        beanForm.setRamo_atividade_lista(
                                                beanSQL.select(
                                                        con,
                                                        p_pagina,
                                                        beanForm.getRamo_atividade_pesq(),
                                                        beanForm.getNr_reg(),
                                                        "",""));
                                        beanForm.setContadoresPaginas(
                                                beanSQL.getContaSelect(),
                                                beanSQL.getContaSelectAtual());


                                } catch (ExcecaoGeral e) {
                                        msgs.add(ActionMessages.GLOBAL_MESSAGE, e.getAction_message());
                                } catch (Exception e) {
                                        msgs.add(
                                                ActionMessages.GLOBAL_MESSAGE,
                                                new ActionMessage("error.sql", e.toString()));
                                }
                        }
                        if (!(msgs.isEmpty())) {
                                con.rollback();
                                saveMessages(request, msgs);
                        }
                        return (mapping.findForward("lista"));
                }
        }
```

Tipo_contas_pagarAction.java

```
package br.com.nota_fluxo.action;

import java.sql.Connection;
import java.util.HashMap;
import java.util.Map;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.log4j.Category;

import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;
import org.apache.struts.action.ActionMessages;
import org.apache.struts.actions.LookupDispatchAction;

import br.com.nota_fluxo.bean.Tipo_contas_pagar;
import br.com.nota_fluxo.form.Tipo_contas_pagarForm;
import br.com.nota_fluxo.sql.Tipo_contas_pagarSQL;
import br.com.nota_fluxo.util.Util;

public class Tipo_contas_pagarAction extends LookupDispatchAction {

        public Category cat = Category.getInstance(Tipo_contas_pagarAction.class.getName());

        protected Map getKeyMethodMap(){
                Map map = new HashMap();
                map.put("nota_fluxo.entra_inserir","entra_inserir");
                map.put("nota_fluxo.entra_consultar","entra_consultar");
                map.put("nota_fluxo.entra_alterar","entra_alterar");
                map.put("nota_fluxo.inserir","inserir");
                map.put("nota_fluxo.apagar","apagar");
                map.put("nota_fluxo.alterar","alterar");
                map.put("nota_fluxo.lista","lista");
                map.put("nota_fluxo.login.valida","lista");
                return map;
        }

        public ActionForward unspecified(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                return (lista(mapping, form, request, response));
        }

        public ActionForward lista(
                ActionMapping mapping,
```

```
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
        ActionMessages msgs = new ActionMessages();

        Connection con =
                (Connection) request.getSession().getAttribute("conexao");

        //UsuarioForm usuario_form =
        //      (UsuarioForm) request.getSession().getAttribute("usuarioForm");
        //cat.warn("tipo_contas_pagar "+usuario_form.getUsuario().getCd_emp());

        Tipo_contas_pagarForm beanForm = (Tipo_contas_pagarForm) form;
        Tipo_contas_pagarSQL beanSQL = new Tipo_contas_pagarSQL();
        try {
                beanForm.setNr_reg(10); //Numero de registros da lista
                beanForm.setMostraConsulta("N");

                int p_pagina = 0;
                if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                        p_pagina = 0;
                } else {
                        p_pagina =
                                (int) Integer.parseInt(request.getParameter("p_pagina"));
                }
                beanForm.setTipo_contas_pagar_lista(
                        beanSQL.select(
                                con,
                                p_pagina,
                                beanForm.getTipo_contas_pagar_pesq(),
                                beanForm.getNr_reg(),
                                "", ""));
                beanForm.setContadoresPaginas(
                        beanSQL.getContaSelect(),
                        beanSQL.getContaSelectAtual());
        } catch (Exception e) {
                msgs.add(
                        ActionMessages.GLOBAL_MESSAGE,
                        new ActionMessage("error.sql", e));
        }
        if (!(msgs.isEmpty())) {
                saveMessages(request, msgs);
        }
        return (mapping.findForward("lista"));
}

public ActionForward entra_consultar(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) {
        Tipo_contas_pagarForm beanForm = (Tipo_contas_pagarForm) form;
        beanForm.setMostraConsulta("S");
        return (mapping.findForward("editar"));
}

public ActionForward inserir(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
        ActionMessages msgs = new ActionMessages();
        Connection con =
                (Connection) request.getSession().getAttribute("conexao");
        Tipo_contas_pagarForm beanForm = (Tipo_contas_pagarForm) form;
        Tipo_contas_pagarSQL beanSQL = new Tipo_contas_pagarSQL();
        msgs = beanForm.validate(mapping, request);
        if (msgs.isEmpty()) {
                try {
                        int p_pagina = 0;

                        if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                                p_pagina = 0;
                        } else {
                                p_pagina =
                                        (int) Integer.parseInt(
                                                request.getParameter("p_pagina"));
                        }
                        String oid = "-1";
                        if (Util.trataNulo(request.getParameter("oid")) != null) {
                                oid = request.getParameter("oid");
                        }

                        Tipo_contas_pagar v_tipo_contas_pagar = new Tipo_contas_pagar();

v_tipo_contas_pagar.setCd_tp_pagar(beanForm.getTipo_contas_pagar().getCd_tp_pagar());
```

```
                                beanForm.setTipo_contas_pagar_lista(
                                        beanSQL.select(con, 0, v_tipo_contas_pagar, 1, "", "N"));
                                if (beanForm.getTipo_contas_pagar_lista().size() > 0) {
                                        throw new ExcecaoGeral(
                                                new ActionMessage("nota_fluxo.inserir.erro"));
                                } else {
                                        beanSQL.insert(beanForm.getTipo_contas_pagar(), con);
                                        con.commit();
                                        beanForm.setTipo_contas_pagar_lista(

        beanSQL.select(con,0,beanForm.getTipo_contas_pagar(),1,"","N"));
                                        beanForm.setTipo_contas_pagar_pesq(new
Tipo_contas_pagar());
                                        beanForm.setTipo_contas_pagar_lista(
                                                beanSQL.select(
                                                        con,
                                                        0,
                                                        beanForm.getTipo_contas_pagar_pesq(),
                                                        2000000000,
                                                        ((Tipo_contas_pagar)
beanForm.getTipo_contas_pagar_lista().get(0))
                                                                .getOid(), "N"));
                                        //
                                        beanForm.setNr_reg(10);
                                        p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
                                        beanForm.setTipo_contas_pagar_lista(
                                                beanSQL.select(
                                                        con,
                                                        p_pagina,
                                                        beanForm.getTipo_contas_pagar_pesq(),
                                                        beanForm.getNr_reg(),
                                                        "", ""));
                                        beanForm.setContadoresPaginas(
                                                beanSQL.getContaSelect(),
                                                beanSQL.getContaSelectAtual());
                                }
                        } catch (ExcecaoGeral e) {
                                msgs.add(ActionMessages.GLOBAL_MESSAGE, e.getAction_message());
                        } catch (Exception e) {
                                msgs.add(
                                        ActionMessages.GLOBAL_MESSAGE,
                                        new ActionMessage("error.sql", e.toString()));
                        }
                }
                if (!(msgs.isEmpty())) {
                        con.rollback();
                        saveMessages(request, msgs);
                }
                return (mapping.findForward("lista"));
        }

        public ActionForward entra_inserir(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                Tipo_contas_pagarForm beanForm = (Tipo_contas_pagarForm) form;
                beanForm.setTipo_contas_pagar(new Tipo_contas_pagar());
                beanForm.setMostraConsulta("N");
                return (mapping.findForward("editar"));
        }

        public ActionForward apagar(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                ActionMessages msgs = new ActionMessages();
                Connection con =
                        (Connection) request.getSession().getAttribute("conexao");
                Tipo_contas_pagarForm beanForm = (Tipo_contas_pagarForm) form;
                Tipo_contas_pagarSQL beanSQL = new Tipo_contas_pagarSQL();
                try {
                        int p_pagina = 0;
                        if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                                p_pagina = 0;
                        } else {
                                p_pagina =
                                        (int) Integer.parseInt(request.getParameter("p_pagina"));
                        }

                        String oid = "-1";
                        if (Util.trataNulo(request.getParameter("oid")) != null) {
                                oid = request.getParameter("oid");
                        }
```

```
                        Tipo_contas_pagar v_tipo_contas_pagar = new Tipo_contas_pagar();
                        v_tipo_contas_pagar.setOid(oid);
                        v_tipo_contas_pagar =
                                (Tipo_contas_pagar) beanSQL.select(con, 0, v_tipo_contas_pagar, 1,
"","N").get(0);

                        beanForm.setTipo_contas_pagar_lista(
                                beanSQL.select(
                                        con,
                                        0,
                                        beanForm.getTipo_contas_pagar_pesq(),
                                        2000000000,
                                        oid, ""));
                        p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());

                        beanSQL.delete(oid, con);
                        con.commit();

                        request.setAttribute("oid", "");
                } catch (Exception e) {
                        msgs.add(
                                ActionMessages.GLOBAL_MESSAGE,
                                new ActionMessage("error.sql", e.toString()));
                }
                if (!(msgs.isEmpty())) {
                        con.rollback();
                        saveMessages(request, msgs);
                }
                return (lista(mapping, beanForm, request, response));
        }

        public ActionForward entra_alterar(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                ActionMessages msgs = new ActionMessages();
                Connection con =
                        (Connection) request.getSession().getAttribute("conexao");
                Tipo_contas_pagarForm beanForm = (Tipo_contas_pagarForm) form;
                try {
                        Tipo_contas_pagarSQL beanSQL = new Tipo_contas_pagarSQL();

                        String oid = "-1";
                        if (Util.trataNulo(request.getParameter("oid")) != null) {
                                oid = request.getParameter("oid");
                        }

                        Tipo_contas_pagar v_tipo_contas_pagar = new Tipo_contas_pagar();
                        v_tipo_contas_pagar.setOid(oid);

        beanForm.setTipo_contas_pagar_lista(beanSQL.select(con,0,v_tipo_contas_pagar,1,"","N"));
                        beanForm.setTipo_contas_pagar((Tipo_contas_pagar)
beanForm.getTipo_contas_pagar_lista().get(0));
                } catch (Exception e) {
                        msgs.add(
                                ActionMessages.GLOBAL_MESSAGE,
                                new ActionMessage("error.sql", e.toString()));
                }
                if (!(msgs.isEmpty())) {
                        con.rollback();
                        saveMessages(request, msgs);
                }
                return (mapping.findForward("editar"));
        }

        public ActionForward alterar(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                ActionMessages msgs = new ActionMessages();
                Connection con =
                        (Connection) request.getSession().getAttribute("conexao");
                Tipo_contas_pagarForm beanForm = (Tipo_contas_pagarForm) form;
                Tipo_contas_pagarSQL beanSQL = new Tipo_contas_pagarSQL();
                msgs = beanForm.validate(mapping, request);
                if (msgs.isEmpty()) {
                        try {
                                int p_pagina = 0;
                                if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                                        p_pagina = 0;
                                } else {
                                        p_pagina =
                                                (int) Integer.parseInt(
                                                        request.getParameter("p_pagina"));
```

```
                                }

                                String oid = "-1";
                                if (Util.trataNulo(request.getParameter("oid")) != null) {
                                        oid = request.getParameter("oid");
                                }

                                beanSQL.update(beanForm.getTipo_contas_pagar(), con);
                                con.commit();

                                beanForm.setTipo_contas_pagar_lista(
                                        beanSQL.select(con, 0, beanForm.getTipo_contas_pagar(), 1,
"", "N"));
                                if (beanForm.getTipo_contas_pagar_lista().size() == 0) {
                                        throw new ExcecaoGeral(
                                                new ActionMessage("nota_fluxo.alterar.erro"));
                                }

                                beanForm.setTipo_contas_pagar_lista(
                                        beanSQL.select(
                                                con,
                                                0,
                                                beanForm.getTipo_contas_pagar_pesq(),
                                                2000000000,
                                                beanForm.getTipo_contas_pagar().getOid(), "N"));


                                //
                                beanForm.setNr_reg(10);
                                p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
                                beanForm.setTipo_contas_pagar_lista(
                                        beanSQL.select(
                                                con,
                                                p_pagina,
                                                beanForm.getTipo_contas_pagar_pesq(),
                                                beanForm.getNr_reg(),
                                                "",""));
                                beanForm.setContadoresPaginas(
                                        beanSQL.getContaSelect(),
                                        beanSQL.getContaSelectAtual());


                        } catch (ExcecaoGeral e) {
                                msgs.add(ActionMessages.GLOBAL_MESSAGE, e.getAction_message());
                        } catch (Exception e) {
                                msgs.add(
                                        ActionMessages.GLOBAL_MESSAGE,
                                        new ActionMessage("error.sql", e.toString()));
                        }
                }
                if (!(msgs.isEmpty())) {
                        con.rollback();
                        saveMessages(request, msgs);
                }
                return (mapping.findForward("lista"));
        }
}
```

Tipo_contas_receberAction.java

```
package br.com.nota_fluxo.action;

import java.sql.Connection;
import java.util.HashMap;
import java.util.Map;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.log4j.Category;

import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;
import org.apache.struts.action.ActionMessages;
import org.apache.struts.actions.LookupDispatchAction;

import br.com.nota_fluxo.bean.Tipo_contas_receber;
import br.com.nota_fluxo.form.Tipo_contas_receberForm;
import br.com.nota_fluxo.sql.Tipo_contas_receberSQL;
import br.com.nota_fluxo.util.Util;

public class Tipo_contas_receberAction extends LookupDispatchAction {

        public Category cat = Category.getInstance(Tipo_contas_receberAction.class.getName());

        protected Map getKeyMethodMap(){
```

```
        Map map = new HashMap();
        map.put("nota_fluxo.entra_inserir","entra_inserir");
        map.put("nota_fluxo.entra_consultar","entra_consultar");
        map.put("nota_fluxo.entra_alterar","entra_alterar");
        map.put("nota_fluxo.inserir","inserir");
        map.put("nota_fluxo.apagar","apagar");
        map.put("nota_fluxo.alterar","alterar");
        map.put("nota_fluxo.lista","lista");
        map.put("nota_fluxo.login.valida","lista");
        return map;
}

public ActionForward unspecified(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
        return (lista(mapping, form, request, response));
}

public ActionForward lista(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
        ActionMessages msgs = new ActionMessages();

        Connection con =
                (Connection) request.getSession().getAttribute("conexao");

        //UsuarioForm usuario_form =
        //      (UsuarioForm) request.getSession().getAttribute("usuarioForm");
        //cat.warn("tipo_contas_receber "+usuario_form.getUsuario().getCd_emp());

        Tipo_contas_receberForm beanForm = (Tipo_contas_receberForm) form;
        Tipo_contas_receberSQL beanSQL = new Tipo_contas_receberSQL();
        try {
                beanForm.setNr_reg(10); //Numero de registros da lista
                beanForm.setMostraConsulta("N");

                int p_pagina = 0;
                if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                        p_pagina = 0;
                } else {
                        p_pagina =
                                (int) Integer.parseInt(request.getParameter("p_pagina"));
                }
                beanForm.setTipo_contas_receber_lista(
                        beanSQL.select(
                                con,
                                p_pagina,
                                beanForm.getTipo_contas_receber_pesq(),
                                beanForm.getNr_reg(),
                                "", ""));
                beanForm.setContadoresPaginas(
                        beanSQL.getContaSelect(),
                        beanSQL.getContaSelectAtual());
        } catch (Exception e) {
                msgs.add(
                        ActionMessages.GLOBAL_MESSAGE,
                        new ActionMessage("error.sql", e));
        }
        if (!(msgs.isEmpty())) {
                saveMessages(request, msgs);
        }
        return (mapping.findForward("lista"));
}

public ActionForward entra_consultar(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) {
        Tipo_contas_receberForm beanForm = (Tipo_contas_receberForm) form;
        beanForm.setMostraConsulta("S");
        return (mapping.findForward("editar"));
}

public ActionForward inserir(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
        ActionMessages msgs = new ActionMessages();
        Connection con =
```

```
                        (Connection) request.getSession().getAttribute("conexao");
                Tipo_contas_receberForm beanForm = (Tipo_contas_receberForm) form;
                Tipo_contas_receberSQL beanSQL = new Tipo_contas_receberSQL();
                msgs = beanForm.validate(mapping, request);
                if (msgs.isEmpty()) {
                        try {
                                int p_pagina = 0;

                                if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                                        p_pagina = 0;
                                } else {
                                        p_pagina =
                                                (int) Integer.parseInt(
                                                        request.getParameter("p_pagina"));
                                }
                                String oid = "-1";
                                if (Util.trataNulo(request.getParameter("oid")) != null) {
                                        oid = request.getParameter("oid");
                                }

                                Tipo_contas_receber v_tipo_contas_receber = new
Tipo_contas_receber();

        v_tipo_contas_receber.setCd_tp_receber(beanForm.getTipo_contas_receber().getCd_tp_receber())
;
                                beanForm.setTipo_contas_receber_lista(
                                        beanSQL.select(con, 0, v_tipo_contas_receber, 1, "", "N"));
                                if (beanForm.getTipo_contas_receber_lista().size() > 0) {
                                        throw new ExcecaoGeral(
                                                new ActionMessage("nota_fluxo.inserir.erro"));
                                } else {
                                        beanSQL.insert(beanForm.getTipo_contas_receber(), con);
                                        con.commit();
                                        beanForm.setTipo_contas_receber_lista(

        beanSQL.select(con,0,beanForm.getTipo_contas_receber(),1,"","N"));
                                        beanForm.setTipo_contas_receber_pesq(new
Tipo_contas_receber());
                                        beanForm.setTipo_contas_receber_lista(
                                                beanSQL.select(
                                                        con,
                                                        0,
                                                        beanForm.getTipo_contas_receber_pesq(),
                                                        2000000000,
                                                        ((Tipo_contas_receber)
beanForm.getTipo_contas_receber_lista().get(0))
                                                                .getOid(), "N"));
                                        //
                                        beanForm.setNr_reg(10);
                                        p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
                                        beanForm.setTipo_contas_receber_lista(
                                                beanSQL.select(
                                                        con,
                                                        p_pagina,
                                                        beanForm.getTipo_contas_receber_pesq(),
                                                        beanForm.getNr_reg(),
                                                        "", ""));
                                        beanForm.setContadoresPaginas(
                                                beanSQL.getContaSelect(),
                                                beanSQL.getContaSelectAtual());
                                }
                        } catch (ExcecaoGeral e) {
                                msgs.add(ActionMessages.GLOBAL_MESSAGE, e.getAction_message());
                        } catch (Exception e) {
                                msgs.add(
                                        ActionMessages.GLOBAL_MESSAGE,
                                        new ActionMessage("error.sql", e.toString()));
                        }
                }
                if (!(msgs.isEmpty())) {
                        con.rollback();
                        saveMessages(request, msgs);
                }
                return (mapping.findForward("lista"));
        }

        public ActionForward entra_inserir(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                Tipo_contas_receberForm beanForm = (Tipo_contas_receberForm) form;
                beanForm.setTipo_contas_receber(new Tipo_contas_receber());
                beanForm.setMostraConsulta("N");
                return (mapping.findForward("editar"));
        }
```

```
        public ActionForward apagar(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                ActionMessages msgs = new ActionMessages();
                Connection con =
                        (Connection) request.getSession().getAttribute("conexao");
                Tipo_contas_receberForm beanForm = (Tipo_contas_receberForm) form;
                Tipo_contas_receberSQL beanSQL = new Tipo_contas_receberSQL();
                try {
                        int p_pagina = 0;
                        if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                                p_pagina = 0;
                        } else {
                                p_pagina =
                                        (int) Integer.parseInt(request.getParameter("p_pagina"));
                        }

                        String oid = "-1";
                        if (Util.trataNulo(request.getParameter("oid")) != null) {
                                oid = request.getParameter("oid");
                        }

                        Tipo_contas_receber v_tipo_contas_receber = new Tipo_contas_receber();
                        v_tipo_contas_receber.setOid(oid);
                        v_tipo_contas_receber =
                                (Tipo_contas_receber) beanSQL.select(con, 0, v_tipo_contas_receber,
1, "","N").get(0);

                        beanForm.setTipo_contas_receber_lista(
                                beanSQL.select(
                                        con,
                                        0,
                                        beanForm.getTipo_contas_receber_pesq(),
                                        2000000000,
                                        oid, ""));
                        p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());

                        beanSQL.delete(oid, con);
                        con.commit();

                        request.setAttribute("oid", "");
                } catch (Exception e) {
                        msgs.add(
                                ActionMessages.GLOBAL_MESSAGE,
                                new ActionMessage("error.sql", e.toString()));
                }
                if (!(msgs.isEmpty())) {
                        con.rollback();
                        saveMessages(request, msgs);
                }
                return (lista(mapping, beanForm, request, response));
        }

        public ActionForward entra_alterar(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                ActionMessages msgs = new ActionMessages();
                Connection con =
                        (Connection) request.getSession().getAttribute("conexao");
                Tipo_contas_receberForm beanForm = (Tipo_contas_receberForm) form;
                try {
                        Tipo_contas_receberSQL beanSQL = new Tipo_contas_receberSQL();

                        String oid = "-1";
                        if (Util.trataNulo(request.getParameter("oid")) != null) {
                                oid = request.getParameter("oid");
                        }

                        Tipo_contas_receber v_tipo_contas_receber = new Tipo_contas_receber();
                        v_tipo_contas_receber.setOid(oid);

        beanForm.setTipo_contas_receber_lista(beanSQL.select(con,0,v_tipo_contas_receber,1,"","N"));
                        beanForm.setTipo_contas_receber((Tipo_contas_receber)
beanForm.getTipo_contas_receber_lista().get(0));
                } catch (Exception e) {
                        msgs.add(
                                ActionMessages.GLOBAL_MESSAGE,
                                new ActionMessage("error.sql", e.toString()));
                }
                if (!(msgs.isEmpty())) {
                        con.rollback();
                        saveMessages(request, msgs);
```

```
                }
                return (mapping.findForward("editar"));
        }

        public ActionForward alterar(
                ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
                throws Exception {
                ActionMessages msgs = new ActionMessages();
                Connection con =
                        (Connection) request.getSession().getAttribute("conexao");
                Tipo_contas_receberForm beanForm = (Tipo_contas_receberForm) form;
                Tipo_contas_receberSQL beanSQL = new Tipo_contas_receberSQL();
                msgs = beanForm.validate(mapping, request);
                if (msgs.isEmpty()) {
                        try {
                                int p_pagina = 0;
                                if (Util.trataNulo(request.getParameter("p_pagina")) == null) {
                                        p_pagina = 0;
                                } else {
                                        p_pagina =
                                                (int) Integer.parseInt(
                                                        request.getParameter("p_pagina"));
                                }

                                String oid = "-1";
                                if (Util.trataNulo(request.getParameter("oid")) != null) {
                                        oid = request.getParameter("oid");
                                }

                                beanSQL.update(beanForm.getTipo_contas_receber(), con);
                                con.commit();

                                beanForm.setTipo_contas_receber_lista(
                                        beanSQL.select(con, 0, beanForm.getTipo_contas_receber(),
1, "", "N"));
                                if (beanForm.getTipo_contas_receber_lista().size() == 0) {
                                        throw new ExcecaoGeral(
                                                new ActionMessage("nota_fluxo.alterar.erro"));
                                }

                                beanForm.setTipo_contas_receber_lista(
                                        beanSQL.select(
                                                con,
                                                0,
                                                beanForm.getTipo_contas_receber_pesq(),
                                                2000000000,
                                                beanForm.getTipo_contas_receber().getOid(), "N"));


                                //
                                beanForm.setNr_reg(10);
                                p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
                                beanForm.setTipo_contas_receber_lista(
                                        beanSQL.select(
                                                con,
                                                p_pagina,
                                                beanForm.getTipo_contas_receber_pesq(),
                                                beanForm.getNr_reg(),
                                                "",""));
                                beanForm.setContadoresPaginas(
                                        beanSQL.getContaSelect(),
                                        beanSQL.getContaSelectAtual());


                        } catch (ExcecaoGeral e) {
                                msgs.add(ActionMessages.GLOBAL_MESSAGE, e.getAction_message());
                        } catch (Exception e) {
                                msgs.add(
                                        ActionMessages.GLOBAL_MESSAGE,
                                        new ActionMessage("error.sql", e.toString()));
                        }
                }
                if (!(msgs.isEmpty())) {
                        con.rollback();
                        saveMessages(request, msgs);
                }
                return (mapping.findForward("lista"));
        }
}
```

LoginAction.java

```
package br.com.nota_fluxo.action;

import java.sql.Connection;
```

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

import javax.naming.Context;
import javax.naming.InitialContext;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.sql.DataSource;

import org.apache.log4j.Category;

import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;
import org.apache.struts.action.ActionMessages;
import org.apache.struts.actions.LookupDispatchAction;

import br.com.nota_fluxo.bean.Usuario;
import br.com.nota_fluxo.form.UsuarioForm;
import br.com.nota_fluxo.sql.UsuarioSQL;
import br.com.nota_fluxo.util.Util;

public class LoginAction extends LookupDispatchAction {

	public Category cat = Category.getInstance(LoginAction.class.getName());

	protected Map getKeyMethodMap() {
		Map map = new HashMap();
		map.put("nota_fluxo.login.valida", "valida");
		return map;
	}

	public ActionForward unspecified(
		ActionMapping mapping,
		ActionForm form,
		HttpServletRequest request,
		HttpServletResponse response)
		throws Exception {
		return (mapping.findForward("editar"));
	}

	public ActionForward valida(
		ActionMapping mapping,
		ActionForm form,
		HttpServletRequest request,
		HttpServletResponse response)
		throws Exception {

		ActionMessages msgs = new ActionMessages();
		try {
			Context initContext;
			initContext = new InitialContext();
			Context envContext = (Context) initContext.lookup("java:/comp/env");
			DataSource ds = (DataSource) envContext.lookup("jdbc/dspostgres");
			Connection con = ds.getConnection();
			con.setAutoCommit(false);
			request.getSession().setAttribute("conexao", con);

			UsuarioForm beanForm = (UsuarioForm) form;
			if ((Util.trataNulo(beanForm.getUsuario().getNm_usuario())==null)
				||(Util.trataNulo(beanForm.getUsuario().getDs_senha())==null)){
					throw new ExcecaoGeral(
						new ActionMessage("nota_fluxo.login.preencha"));
			} else {
				UsuarioSQL beanSQL = new UsuarioSQL();
				Usuario v_usuario = new Usuario();
				v_usuario.setNm_usuario(beanForm.getUsuario().getNm_usuario());
				v_usuario.setDs_senha(beanForm.getUsuario().getDs_senha());
				if (beanSQL.select(con, 0, v_usuario,1,"","N").size()==1){
					beanForm.setUsuario((Usuario)beanSQL.select(con, 0,
v_usuario,1,"","N").get(0));
					return (mapping.findForward("lista"));
				} else {
					throw new ExcecaoGeral(
						new ActionMessage("nota_fluxo.login.erro"));
				}

			}
		} catch (ExcecaoGeral e) {
			msgs.add(ActionMessages.GLOBAL_MESSAGE, e.getAction_message());
			saveMessages(request, msgs);
			return (mapping.findForward("editar"));
		} catch (Exception e) {
			msgs.add(ActionMessages.GLOBAL_MESSAGE,new ActionMessage("error.sql", e));
			saveMessages(request, msgs);
			return (mapping.findForward("editar"));
		}
```

```
        }
}
```

ExcecaoGeral.java

```
package br.com.nota_fluxo.action;

import org.apache.struts.action.ActionMessage;;

public class ExcecaoGeral extends Exception {
        private ActionMessage action_message;

        public ExcecaoGeral(){
                super();
        }
        public ExcecaoGeral(ActionMessage p_action_message){
                setAction_message(p_action_message);
        }
        public ActionMessage getAction_message() {
                return action_message;
        }
        public void setAction_message(ActionMessage message) {
                action_message = message;
        }
}
```

Útil.java

```
package br.com.nota_fluxo.util;

public class Util {

        public static String trataNulo (String p_string){
                if      ((p_string==null)                       ||(p_string.equals("null"))
                        ||(p_string.equals(""))   ||(p_string==="")){
                        p_string = null;
                }
                return p_string;
        }
}
```

editar_cliente.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<html>
<head>
<title>Cliente</title>
</head>

<body bgcolor="#ffffff">
<html:form action="/cliente">
<input type="hidden" name="action">
<table width="75%" border="1" bgcolor="#c8c8c8" bordercolor="#ffffff" align="center">
        <tr>
                <td>
                        <table width="100%" border="0" bgcolor="#ffffff">
                          <tr>
                        <logic:notEqual name="clienteForm" property="mostraConsulta" value="S">
                                <html:hidden name="clienteForm" property="cliente.oid"/>
                            <td colspan="5" align="center">
                                        <logic:present name="clienteForm" property="cliente.oid">
                                                <html:submit property="evento">
                                                        <bean:message key="nota_fluxo.alterar"/>
                                                </html:submit>
                                        </logic:present>
                                        <logic:notPresent name="clienteForm"
property="cliente.oid">
                                                <html:submit property="evento">
                                                        <bean:message key="nota_fluxo.inserir"/>
                                                </html:submit>
                                        </logic:notPresent>
                            </td>
                          </tr>

                          <tr>
                                <td class="TxtGeral" width="15%" align="right">Cliente:
                                <td class="TxtGeral" width="35%">
                                        <logic:present name="clienteForm" property="cliente.oid">
                                                <html:text maxlength="4" size="4" style="text-
align: right" name="clienteForm" readonly="true" property="cliente.cd_cli" />
                                        </logic:present>
                                        <logic:notPresent name="clienteForm"
property="cliente.oid">
                                            <html:text maxlength="4" size="4" style="text-align:
right" name="clienteForm" property="cliente.cd_cli" />
                                        </logic:notPresent>
```

```
                                    <html:text maxlength="30" size="25" name="clienteForm"
property="cliente.nm_cli"  />
                                <td class="TxtGeral" width="15%" align="right">
                                <td class="TxtGeral" width="35%">
                              <tr>
                                <td class="TxtGeral" width="15%" align="right">Razão Social:
                                <td class="TxtGeral" width="35%"><html:text maxlength="30" size="34"
name="clienteForm" property="cliente.razao_social"  />
                                <td class="TxtGeral" width="15%" align="right">Cidade:
                                <td class="TxtGeral" width="35%"><html:text maxlength="30" size="20"
name="clienteForm" property="cliente.cidade"  />     UF: <html:text maxlength="2"
size="2" name="clienteForm" property="cliente.uf"  />
                              <tr>
                                <td class="TxtGeral" width="15%" align="right">Endereço:
                                <td class="TxtGeral" width="35%"><html:text maxlength="30" size="34"
name="clienteForm" property="cliente.endereco"  />
                                <td class="TxtGeral" width="15%" align="right">Bairro:
                                <td class="TxtGeral" width="35%"><html:text maxlength="30" size="20"
name="clienteForm" property="cliente.bairro"  />
                              <tr>
                                <td class="TxtGeral" width="15%" align="right">CEP:
                                <td class="TxtGeral" width="35%"><html:text maxlength="8"  size="10"
name="clienteForm" property="cliente.cep"  />
                                <td class="TxtGeral" width="15%" align="right">Atividade:
                                <td class="TxtGeral" width="35%"><html:text maxlength="4"  size="4"
name="clienteForm" property="cliente.cd_atividade"  />
                              <tr>
                                <td class="TxtGeral" width="15%" align="right">CNPJ:
                                <td class="TxtGeral" width="35%"><html:text maxlength="16" size="15"
name="clienteForm" property="cliente.cnpj"  />
                                <td class="TxtGeral" width="15%" align="right">Insc Estadual:
                                <td class="TxtGeral" width="35%"><html:text maxlength="10" size="15"
name="clienteForm" property="cliente.insc_estadual"  />
                              <tr>
                                <td class="TxtGeral" width="15%" align="right">Telefone:
                                <td class="TxtGeral" width="35%"><html:text maxlength="10" size="15"
name="clienteForm" property="cliente.telefone"  />
                                <td class="TxtGeral" width="15%" align="right">Fax:
                                <td class="TxtGeral" width="35%"><html:text maxlength="10" size="15"
name="clienteForm" property="cliente.fax"  />
                              <tr>
                                <td class="TxtGeral" width="15%" align="right">Contato:
                                <td class="TxtGeral" width="35%"><html:text maxlength="30" size="20"
name="clienteForm" property="cliente.nm_contato"  />
                                <td class="TxtGeral" width="15%" align="right">Email:
                                <td class="TxtGeral" width="35%"><html:text maxlength="30" size="34"
name="clienteForm" property="cliente.email"  />

                                    <html:hidden name="clienteForm" property="cliente_pesq.cd_cli"/>
                                    <html:hidden name="clienteForm" property="cliente_pesq.nm_cli"/>
                                    <html:hidden name="clienteForm"
property="cliente_pesq.razao_social"/>
                                    <html:hidden name="clienteForm" property="cliente_pesq.endereco"/>
                                    <html:hidden name="clienteForm" property="cliente_pesq.bairro"/>
                                    <html:hidden name="clienteForm" property="cliente_pesq.cidade"/>
                                    <html:hidden name="clienteForm" property="cliente_pesq.uf"/>
                                    <html:hidden name="clienteForm" property="cliente_pesq.cep"/>
                                    <html:hidden name="clienteForm" property="cliente_pesq.cnpj"/>
                                    <html:hidden name="clienteForm"
property="cliente_pesq.insc_estadual"/>
                                    <html:hidden name="clienteForm" property="cliente_pesq.telefone"/>
                                    <html:hidden name="clienteForm" property="cliente_pesq.fax"/>
                                    <html:hidden name="clienteForm" property="cliente_pesq.email"/>
                                    <html:hidden name="clienteForm"
property="cliente_pesq.nm_contato"/>
                                    <html:hidden name="clienteForm"
property="cliente_pesq.cd_atividade"/>
                        </logic:notEqual>

                        <logic:equal name="clienteForm" property="mostraConsulta" value="S">
                                <tr>
                                    <td colspan="5" align="center">
                                            <html:submit property="evento">
                                                <bean:message key="nota_fluxo.lista"/>
                                            </html:submit>
                                    </td>
                                </tr>
                            <tr>
                                <td class="TxtGeral" width="15%" align="right">Cliente:
                                <td class="TxtGeral" width="35%">
                                        <html:text maxlength="4" size="4" style="text-align: right"
name="clienteForm" property="cliente_pesq.cd_emp" />
                                        <html:text maxlength="30" size="25" name="clienteForm"
property="cliente_pesq.nm_emp"  />
                                <td class="TxtGeral" width="15%" align="right">
                                <td class="TxtGeral" width="35%">
                              <tr>
                                <td class="TxtGeral" width="15%" align="right">Razão Social:
```

```
                            <td class="TxtGeral" width="35%"><html:text maxlength="30" size="34"
name="clienteForm" property="cliente_pesq.razao_social"  />
                            <td class="TxtGeral" width="15%" align="right">Cidade:
                            <td class="TxtGeral" width="35%"><html:text maxlength="30" size="20"
name="clienteForm" property="cliente_pesq.cidade"  />     UF: <html:text maxlength="2"
size="2" name="clienteForm" property="cliente_pesq.uf"  />
                          <tr>
                            <td class="TxtGeral" width="15%" align="right">Endereço:
                            <td class="TxtGeral" width="35%"><html:text maxlength="30" size="34"
name="clienteForm" property="cliente_pesq.endereco"  />
                            <td class="TxtGeral" width="15%" align="right">Bairro:
                            <td class="TxtGeral" width="35%"><html:text maxlength="30" size="20"
name="clienteForm" property="cliente_pesq.bairro"  />
                          <tr>
                            <td class="TxtGeral" width="15%" align="right">CEP:
                            <td class="TxtGeral" width="35%"><html:text maxlength="8"  size="10"
name="clienteForm" property="cliente_pesq.cep"  />
                            <td class="TxtGeral" width="15%" align="right">Atividade:
                            <td class="TxtGeral" width="35%"><html:text maxlength="4"  size="4"
name="clienteForm" property="cliente_pesq.cd_atividade"  />
                          <tr>
                            <td class="TxtGeral" width="15%" align="right">CNPJ:
                            <td class="TxtGeral" width="35%"><html:text maxlength="16" size="15"
name="clienteForm" property="cliente_pesq.cnpj"  />
                            <td class="TxtGeral" width="15%" align="right">Insc Estadual:
                            <td class="TxtGeral" width="35%"><html:text maxlength="10" size="15"
name="clienteForm" property="cliente_pesq.insc_estadual"  />
                          <tr>
                            <td class="TxtGeral" width="15%" align="right">Telefone:
                            <td class="TxtGeral" width="35%"><html:text maxlength="10" size="15"
name="clienteForm" property="cliente_pesq.telefone"  />
                            <td class="TxtGeral" width="15%" align="right">Fax:
                            <td class="TxtGeral" width="35%"><html:text maxlength="10" size="15"
name="clienteForm" property="cliente_pesq.fax"  />
                          <tr>
                            <td class="TxtGeral" width="15%" align="right">Contato:
                            <td class="TxtGeral" width="35%"><html:text maxlength="30" size="20"
name="clienteForm" property="cliente_pesq.nm_contato"  />
                            <td class="TxtGeral" width="15%" align="right">Email:
                            <td class="TxtGeral" width="35%"><html:text maxlength="30" size="34"
name="clienteForm" property="cliente_pesq.email"  />
                        </logic:equal>
                        </table>
                </td>
        </tr>
</table>

<html:messages id="msg" message="true">
        <bean:write name="msg"/><br>
</html:messages>

</html:form>
</body>
</html>
```

editar_cnae.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<html>
<head>
<title>Cnae</title>
</head>

<body bgcolor="#ffffff">
<html:form action="/cnae">
<input type="hidden" name="action">
<table width="75%" border="1" bgcolor="#c8c8c8" bordercolor="#ffffff" align="center">
        <tr>
                <td>
                        <table width="100%" border="0" bgcolor="#ffffff">
                          <tr>
                        <logic:notEqual name="cnaeForm" property="mostraConsulta" value="S">
                                <html:hidden name="cnaeForm" property="cnae.oid"/>
                            <td colspan="5" align="center">
                                        <logic:present name="cnaeForm" property="cnae.oid">
                                                <html:submit property="evento">
                                                        <bean:message key="nota_fluxo.alterar"/>
                                                </html:submit>
                                        </logic:present>
                                        <logic:notPresent name="cnaeForm" property="cnae.oid">
                                                <html:submit property="evento">
                                                        <bean:message key="nota_fluxo.inserir"/>
                                                </html:submit>
                                        </logic:notPresent>
                            </td>
                          </tr>
```

```
                        <tr>
                                <td  align="center">Cnae:
                                        <logic:present name="cnaeForm" property="cnae.oid">
                                                <html:text maxlength="4" size="4" style="text-
align: right" name="cnaeForm" readonly="true" property="cnae.cd_cnae" />
                                        </logic:present>
                                        <logic:notPresent name="cnaeForm" property="cnae.oid">
                                            <html:text maxlength="4" size="4" style="text-align:
right" name="cnaeForm" property="cnae.cd_cnae" />
                                        </logic:notPresent>
                                        Descrição:<html:text maxlength="30" size="20"
name="cnaeForm" property="cnae.ds_cnae" />

                                <html:hidden name="cnaeForm" property="cnae_pesq.cd_cnae"/>
                                <html:hidden name="cnaeForm" property="cnae_pesq.ds_cnae"/>
                        </logic:notEqual>

                        <logic:equal name="cnaeForm" property="mostraConsulta" value="S">
                                <tr>
                                    <td colspan="5" align="center">
                                                <html:submit property="evento">
                                                        <bean:message key="nota_fluxo.lista"/>
                                                </html:submit>
                                        </td>
                                </tr>
                          <tr>
                            <td align="center">
                                Código:<html:text maxlength="30" size="34" name="cnaeForm"
property="cnae_pesq.cd_cnae" />
                                        Descrição: <html:text maxlength="30" size="20" name="cnaeForm"
property="cnae_pesq.ds_cnae" />
                        </logic:equal>
                        </table>
                </td>
        </tr>
</table>

<html:messages id="msg" message="true">
        <bean:write name="msg"/><br>
</html:messages>

</html:form>
</body>
</html>
```

editar_contas_pagar.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<html>
<head>
<title>Contas_pagar</title>
</head>

<body bgcolor="#ffffff">
<html:form action="/contas_pagar">
<input type="hidden" name="action">
<table width="75%" border="1" bgcolor="#c8c8c8" bordercolor="#ffffff" align="center">
        <tr>
                <td>
                        <table width="100%" border="0" bgcolor="#ffffff">
                          <tr>
                        <logic:notEqual name="contas_pagarForm" property="mostraConsulta"
value="S">
                                <html:hidden name="contas_pagarForm" property="contas_pagar.oid"/>
                           <td colspan="5" align="center">
                                        <logic:present name="contas_pagarForm"
property="contas_pagar.oid">
                                                <html:submit property="evento">
                                                        <bean:message key="nota_fluxo.alterar"/>
                                                </html:submit>
                                        </logic:present>
                                        <logic:notPresent name="contas_pagarForm"
property="contas_pagar.oid">
                                                <html:submit property="evento">
                                                        <bean:message key="nota_fluxo.inserir"/>
                                                </html:submit>
                                        </logic:notPresent>
                           </td>
                          </tr>

                          <tr>
                                <td width="15%" align="right">Empresa:
                                <td width="35%">
                                        <logic:present name="contas_pagarForm"
property="contas_pagar.oid">
```

```
                                        <html:text maxlength="4" size="4" style="text-
align: right" name="contas_pagarForm" readonly="true" property="contas_pagar.cd_emp" />
                                    </logic:present>
                                    <logic:notPresent name="contas_pagarForm"
property="contas_pagar.oid">
                                        <html:text maxlength="4" size="4" style="text-align:
right" name="contas_pagarForm" property="contas_pagar.cd_emp" />
                                    </logic:notPresent>
                                    Nr
                                    <logic:present name="contas_pagarForm"
property="contas_pagar.oid">
                                        <html:text maxlength="4" size="4" style="text-
align: right" name="contas_pagarForm" readonly="true" property="contas_pagar.cd_pagar" />
                                    </logic:present>
                                    <logic:notPresent name="contas_pagarForm"
property="contas_pagar.oid">
                                        <html:text maxlength="4" size="4" style="text-align:
right" name="contas_pagarForm" property="contas_pagar.cd_pagar" />
                                    </logic:notPresent>
                            <td width="15%" align="right">
                            <td width="35%">
                          <tr>
                            <td width="15%" align="right">Dt Vencimento:
                            <td width="35%"><html:text maxlength="10" size="10"
name="contas_pagarForm" property="contas_pagar.dt_vencto" />
                            <td width="15%" align="right">Dt Pagamento:
                            <td width="35%"><html:text maxlength="10" size="10"
name="contas_pagarForm" property="contas_pagar.dt_pagto" />
                          <tr>
                            <td width="15%" align="right">Vl Acrescimos:
                            <td width="35%"><html:text maxlength="8" size="8"
name="contas_pagarForm" property="contas_pagar.vl_acrescimos" />
                            <td width="15%" align="right">Vl Descontos:
                            <td width="35%"><html:text maxlength="8" size="8"
name="contas_pagarForm" property="contas_pagar.vl_descontos" />
                          <tr>
                            <td width="15%" align="right">Vl a Pagar:
                            <td width="35%"><html:text maxlength="8"  size="10"
name="contas_pagarForm" property="contas_pagar.vl_pagar" />
                            <td width="15%" align="right">Vl Pago:
                            <td width="35%"><html:text maxlength="8"  size="8"
name="contas_pagarForm" property="contas_pagar.vl_pago" />
                          <tr>
                            <td width="15%" align="right">Tipo:
                            <td width="35%"><html:text maxlength="4" size="4"
name="contas_pagarForm" property="contas_pagar.cd_tp_pagar" />
                            <td width="15%" align="right">Conta:
                            <td width="35%"><html:text maxlength="4" size="4"
name="contas_pagarForm" property="contas_pagar.cd_conta" />

                                <html:hidden name="contas_pagarForm"
property="contas_pagar_pesq.cd_emp"/>
                                <html:hidden name="contas_pagarForm"
property="contas_pagar_pesq.cd_pagar"/>
                                <html:hidden name="contas_pagarForm"
property="contas_pagar_pesq.dt_vencto"/>
                                <html:hidden name="contas_pagarForm"
property="contas_pagar_pesq.dt_pagto"/>
                                <html:hidden name="contas_pagarForm"
property="contas_pagar_pesq.vl_acrescimos"/>
                                <html:hidden name="contas_pagarForm"
property="contas_pagar_pesq.vl_descontos"/>
                                <html:hidden name="contas_pagarForm"
property="contas_pagar_pesq.vl_pagar"/>
                                <html:hidden name="contas_pagarForm"
property="contas_pagar_pesq.vl_pago"/>
                                <html:hidden name="contas_pagarForm"
property="contas_pagar_pesq.cd_tp_pagar"/>
                                <html:hidden name="contas_pagarForm"
property="contas_pagar_pesq.cd_conta"/>
                        </logic:notEqual>

                        <logic:equal name="contas_pagarForm" property="mostraConsulta" value="S">
                                <tr>
                                    <td colspan="5" align="center">
                                        <html:submit property="evento">
                                            <bean:message key="nota_fluxo.lista"/>
                                        </html:submit>
                                    </td>
                                </tr>
                            <tr>
                                <td width="15%" align="right">Empresa:
                                <td width="35%">
                                    <html:text maxlength="4" size="4" style="text-align: right"
name="contas_pagarForm" property="contas_pagar_pesq.cd_emp" />
                                    Nr
                                    <html:text maxlength="4" size="4" name="contas_pagarForm"
property="contas_pagar_pesq.cd_conta" />
```

```
                              <td width="15%" align="right">
                              <td width="35%">
                            <tr>
                              <td width="15%" align="right">Dt Vencimento:
                              <td width="35%"><html:text maxlength="10" size="10"
name="contas_pagarForm" property="contas_pagar_pesq.dt_vencto" />
                              <td width="15%" align="right">Dt_pagamento:
                              <td width="35%"><html:text maxlength="10" size="10"
name="contas_pagarForm" property="contas_pagar_pesq.dt_pagto" />
                            <tr>
                              <td width="15%" align="right">Vl Acrescimos:
                              <td width="35%"><html:text maxlength="8"  size="8"
name="contas_pagarForm" property="contas_pagar_pesq.vl_acrescimos" />
                              <td width="15%" align="right">Vl Descontos:
                              <td width="35%"><html:text maxlength="8"  size="8"
name="contas_pagarForm" property="contas_pagar_pesq.vl_descontos" />
                            <tr>
                              <td width="15%" align="right">Vl a Pagar:
                              <td width="35%"><html:text maxlength="8" size="8"
name="contas_pagarForm" property="contas_pagar_pesq.vl_pagar" />
                              <td width="15%" align="right">Vl Pato:
                              <td width="35%"><html:text maxlength="8" size="8"
name="contas_pagarForm" property="contas_pagar_pesq.vl_pago" />
                            <tr>
                              <td width="15%" align="right">Tipo:
                              <td width="35%"><html:text maxlength="4" size="4"
name="contas_pagarForm" property="contas_pagar_pesq.cd_tp_pagar" />
                              <td width="15%" align="right">Conta:
                              <td width="35%"><html:text maxlength="4" size="4"
name="contas_pagarForm" property="contas_pagar_pesq.cd_conta" />
                           </logic:equal>
                           </table>
                  </td>
          </tr>
</table>

<html:messages id="msg" message="true">
        <bean:write name="msg"/><br>
</html:messages>

</html:form>
</body>
</html>

editar_contas_receber.jsp

<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<html>
<head>
<title>Contas_receber</title>
</head>

<body bgcolor="#ffffff">
<html:form action="/contas_receber">
<input type="hidden" name="action">
<table width="75%" border="1" bgcolor="#c8c8c8" bordercolor="#ffffff" align="center">
         <tr>
                  <td>
                           <table width="100%" border="0" bgcolor="#ffffff">
                             <tr>
                           <logic:notEqual name="contas_receberForm" property="mostraConsulta"
value="S">
                                    <html:hidden name="contas_receberForm"
property="contas_receber.oid"/>
                              <td colspan="5" align="center">
                                       <logic:present name="contas_receberForm"
property="contas_receber.oid">
                                                <html:submit property="evento">
                                                        <bean:message key="nota_fluxo.alterar"/>
                                                </html:submit>
                                       </logic:present>
                                       <logic:notPresent name="contas_receberForm"
property="contas_receber.oid">
                                                <html:submit property="evento">
                                                        <bean:message key="nota_fluxo.inserir"/>
                                                </html:submit>
                                       </logic:notPresent>
                              </td>
                            </tr>

                            <tr>
                                  <td width="15%" align="right">Empresa:
                                  <td width="35%">
                                          <logic:present name="contas_receberForm"
property="contas_receber.oid">
```

```
                                        <html:text maxlength="4" size="4" style="text-
align: right" name="contas_receberForm" readonly="true" property="contas_receber.cd_emp" />
                                    </logic:present>
                                    <logic:notPresent name="contas_receberForm"
property="contas_receber.oid">
                                        <html:text maxlength="4" size="4" style="text-align:
right" name="contas_receberForm" property="contas_receber.cd_emp" />
                                    </logic:notPresent>
                                    Nr
                                    <logic:present name="contas_receberForm"
property="contas_receber.oid">
                                        <html:text maxlength="4" size="4" style="text-
align: right" name="contas_receberForm" readonly="true" property="contas_receber.cd_receber" />
                                    </logic:present>
                                    <logic:notPresent name="contas_receberForm"
property="contas_receber.oid">
                                        <html:text maxlength="4" size="4" style="text-align:
right" name="contas_receberForm" property="contas_receber.cd_receber" />
                                    </logic:notPresent>
                            <td width="15%" align="right">
                            <td width="35%">
                          <tr>
                            <td width="15%" align="right">Dt Vencimento:
                            <td width="35%"><html:text maxlength="10" size="10"
name="contas_receberForm" property="contas_receber.dt_vencto" />
                            <td width="15%" align="right">Dt Pagamento:
                            <td width="35%"><html:text maxlength="10" size="10"
name="contas_receberForm" property="contas_receber.dt_pagto" />
                           <tr>
                            <td width="15%" align="right">Vl a Receber:
                            <td width="35%"><html:text maxlength="8"  size="10"
name="contas_receberForm" property="contas_receber.vl_receber" />
                            <td width="15%" align="right">Vl Recebido:
                            <td width="35%"><html:text maxlength="8"  size="8"
name="contas_receberForm" property="contas_receber.vl_recebido" />
                           <tr>
                            <td width="15%" align="right">Tipo:
                            <td width="35%"><html:text maxlength="4" size="4"
name="contas_receberForm" property="contas_receber.cd_tp_receber" />
                            <td width="15%" align="right">Conta:
                            <td width="35%"><html:text maxlength="4" size="4"
name="contas_receberForm" property="contas_receber.cd_conta" />
                           <tr>
                            <td width="15%" align="right">Nota:
                            <td width="35%"><html:text maxlength="4" size="4"
name="contas_receberForm" property="contas_receber.nr_nota" />
                            <td width="15%" align="right">
                            <td width="35%">

                                <html:hidden name="contas_receberForm"
property="contas_receber_pesq.cd_emp"/>
                                <html:hidden name="contas_receberForm"
property="contas_receber_pesq.cd_receber"/>
                                <html:hidden name="contas_receberForm"
property="contas_receber_pesq.dt_vencto"/>
                                <html:hidden name="contas_receberForm"
property="contas_receber_pesq.dt_pagto"/>
                                <html:hidden name="contas_receberForm"
property="contas_receber_pesq.vl_receber"/>
                                <html:hidden name="contas_receberForm"
property="contas_receber_pesq.vl_recebido"/>
                                <html:hidden name="contas_receberForm"
property="contas_receber_pesq.cd_tp_receber"/>
                                <html:hidden name="contas_receberForm"
property="contas_receber_pesq.cd_conta"/>
                                <html:hidden name="contas_receberForm"
property="contas_receber_pesq.nr_nota"/>
                        </logic:notEqual>

                        <logic:equal name="contas_receberForm" property="mostraConsulta" value="S">
                                <tr>
                                    <td colspan="5" align="center">
                                        <html:submit property="evento">
                                            <bean:message key="nota_fluxo.lista"/>
                                        </html:submit>
                                    </td>
                                </tr>
                          <tr>
                                <td width="15%" align="right">Empresa:
                                <td width="35%">
                                    <html:text maxlength="4" size="4" style="text-align: right"
name="contas_receberForm" property="contas_receber_pesq.cd_emp" />
                                    Nr
                                    <html:text maxlength="4" size="4" name="contas_receberForm"
property="contas_receber_pesq.cd_conta" />
                            <td width="15%" align="right">
                            <td width="35%">
                          <tr>
```

```
                            <td width="15%" align="right">Dt Vencimento:
                            <td width="35%"><html:text maxlength="10" size="10"
name="contas_receberForm" property="contas_receber_pesq.dt_vencto" />
                            <td width="15%" align="right">Dt_pagamento:
                            <td width="35%"><html:text maxlength="10" size="10"
name="contas_receberForm" property="contas_receber_pesq.dt_pagto" />
                          <tr>
                            <td width="15%" align="right">Vl a Receber:
                            <td width="35%"><html:text maxlength="8" size="8"
name="contas_receberForm" property="contas_receber_pesq.vl_receber" />
                            <td width="15%" align="right">Vl Recebido:
                            <td width="35%"><html:text maxlength="8" size="8"
name="contas_receberForm" property="contas_receber_pesq.vl_recebido" />
                          <tr>
                            <td width="15%" align="right">Tipo:
                            <td width="35%"><html:text maxlength="4" size="4"
name="contas_receberForm" property="contas_receber_pesq.cd_tp_receber" />
                            <td width="15%" align="right">Conta:
                            <td width="35%"><html:text maxlength="4" size="4"
name="contas_receberForm" property="contas_receber_pesq.cd_conta" />
                          <tr>
                            <td width="15%" align="right">Nota:
                            <td width="35%"><html:text maxlength="4" size="4"
name="contas_receberForm" property="contas_receber_pesq.nr_nota" />
                            <td width="15%" align="right">
                            <td width="35%">
                        </logic:equal>
                        </table>
                </td>
        </tr>
</table>

<html:messages id="msg" message="true">
        <bean:write name="msg"/><br>
</html:messages>

</html:form>
</body>
</html>
```

editar_contas.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<html>
<head>
<title>Contas</title>
</head>

<body bgcolor="#ffffff">
<html:form action="/contas">
<input type="hidden" name="action">
<table width="75%" border="1" bgcolor="#c8c8c8" bordercolor="#ffffff" align="center">
        <tr>
                <td>
                        <table width="100%" border="0" bgcolor="#ffffff">
                          <tr>
                        <logic:notEqual name="contasForm" property="mostraConsulta" value="S">
                                <html:hidden name="contasForm" property="contas.oid"/>
                            <td colspan="5" align="center">
                                        <logic:present name="contasForm" property="contas.oid">
                                                <html:submit property="evento">
                                                        <bean:message key="nota_fluxo.alterar"/>
                                                </html:submit>
                                        </logic:present>
                                        <logic:notPresent name="contasForm" property="contas.oid">
                                                <html:submit property="evento">
                                                        <bean:message key="nota_fluxo.inserir"/>
                                                </html:submit>
                                        </logic:notPresent>
                            </td>
                          </tr>

                          <tr>
                                <td  align="center">
                                        Empresa:
                                        <logic:present name="contasForm" property="contas.oid">
                                                <html:text maxlength="4" size="4" style="text-
align: right" name="contasForm" readonly="true" property="contas.cd_emp" />
                                        </logic:present>
                                        <logic:notPresent name="contasForm" property="contas.oid">
                                            <html:text maxlength="4" size="4" style="text-align:
right" name="contasForm" property="contas.cd_emp" />
                                        </logic:notPresent>
                                        Conta:
                                        <logic:present name="contasForm" property="contas.oid">
```

```
                                                <html:text maxlength="4" size="4" style="text-
align: right" name="contasForm" readonly="true" property="contas.cd_conta" />
                                        </logic:present>
                                        <logic:notPresent name="contasForm" property="contas.oid">
                                            <html:text maxlength="4" size="4" style="text-align:
right" name="contasForm" property="contas.cd_conta" />
                                        </logic:notPresent>
                                        Descrição:<html:text maxlength="30" size="20"
name="contasForm" property="contas.ds_conta" />

                                <html:hidden name="contasForm" property="contas_pesq.cd_emp"/>
                                <html:hidden name="contasForm" property="contas_pesq.cd_conta"/>
                                <html:hidden name="contasForm" property="contas_pesq.ds_conta"/>
                        </logic:notEqual>

                        <logic:equal name="contasForm" property="mostraConsulta" value="S">
                                <tr>
                                    <td colspan="5" align="center">
                                                <html:submit property="evento">
                                                        <bean:message key="nota_fluxo.lista"/>
                                                </html:submit>
                                        </td>
                                </tr>
                          <tr>
                            <td align="center">
                                Empresa:<html:text maxlength="4" size="4" name="contasForm"
property="contas_pesq.cd_emp" />
                                Código:<html:text maxlength="4" size="4" name="contasForm"
property="contas_pesq.cd_conta" />
                                        Descrição: <html:text maxlength="30" size="20"
name="contasForm" property="contas_pesq.ds_conta" />
                        </logic:equal>
                        </table>
                </td>
        </tr>
</table>

<html:messages id="msg" message="true">
        <bean:write name="msg"/><br>
</html:messages>

</html:form>
</body>
</html>
```

editar_empresa.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<html>
<head>
<title>Empresa</title>
</head>

<body bgcolor="#ffffff">
<html:form action="/empresa">
<input type="hidden" name="action">
<table width="75%" border="1" bgcolor="#c8c8c8" bordercolor="#ffffff" align="center">
        <tr>
                <td>
                        <table width="100%" border="0" bgcolor="#ffffff">
                          <tr>
                        <logic:notEqual name="empresaForm" property="mostraConsulta" value="S">
                                <html:hidden name="empresaForm" property="empresa.oid"/>
                            <td colspan="5" align="center">
                                        <logic:present name="empresaForm" property="empresa.oid">
                                                <html:submit property="evento">
                                                        <bean:message key="nota_fluxo.alterar"/>
                                                </html:submit>
                                        </logic:present>
                                        <logic:notPresent name="empresaForm"
property="empresa.oid">
                                                <html:submit property="evento">
                                                        <bean:message key="nota_fluxo.inserir"/>
                                                </html:submit>
                                        </logic:notPresent>
                            </td>
                          </tr>

                          <tr>
                                <td width="15%" align="right">Empresa:
                                <td width="35%">
                                        <logic:present name="empresaForm" property="empresa.oid">
                                                <html:text maxlength="4" size="4" style="text-
align: right" name="empresaForm" readonly="true" property="empresa.cd_emp" />
                                        </logic:present>
```

```
                                    <logic:notPresent name="empresaForm"
property="empresa.oid">
                                        <html:text maxlength="4" size="4" style="text-align:
right" name="empresaForm" property="empresa.cd_emp" />
                                    </logic:notPresent>
                                    <html:text maxlength="30" size="25" name="empresaForm"
property="empresa.nm_emp" />
                            <td width="15%" align="right">
                            <td width="35%">
                          <tr>
                            <td width="15%" align="right">Razão Social:
                            <td width="35%"><html:text maxlength="30" size="34" name="empresaForm"
property="empresa.razao_social" />
                            <td width="15%" align="right">Cidade:
                            <td width="35%"><html:text maxlength="30" size="20" name="empresaForm"
property="empresa.cidade" />     UF: <html:text maxlength="2" size="2" name="empresaForm"
property="empresa.uf" />
                          <tr>
                            <td width="15%" align="right">Endereço:
                            <td width="35%"><html:text maxlength="30" size="34" name="empresaForm"
property="empresa.endereco" />
                            <td width="15%" align="right">Bairro:
                            <td width="35%"><html:text maxlength="30" size="20" name="empresaForm"
property="empresa.bairro" />
                          <tr>
                            <td width="15%" align="right">CEP:
                            <td width="35%"><html:text maxlength="8"  size="10" name="empresaForm"
property="empresa.cep" />
                            <td width="15%" align="right">Atividade:
                            <td width="35%"><html:text maxlength="4"  size="4" name="empresaForm"
property="empresa.cd_atividade" />
                          <tr>
                            <td width="15%" align="right">CNPJ:
                            <td width="35%"><html:text maxlength="16" size="15" name="empresaForm"
property="empresa.cnpj" />
                            <td width="15%" align="right">Insc Estadual:
                            <td width="35%"><html:text maxlength="10" size="15" name="empresaForm"
property="empresa.insc_estadual" />
                          <tr>
                            <td width="15%" align="right">Telefone:
                            <td width="35%"><html:text maxlength="10" size="15" name="empresaForm"
property="empresa.telefone" />
                            <td width="15%" align="right">Fax:
                            <td width="35%"><html:text maxlength="10" size="15" name="empresaForm"
property="empresa.fax" />
                          <tr>
                            <td width="15%" align="right">Contato:
                            <td width="35%"><html:text maxlength="30" size="20" name="empresaForm"
property="empresa.nm_contato" />
                            <td width="15%" align="right">Email:
                            <td width="35%"><html:text maxlength="30" size="34" name="empresaForm"
property="empresa.email" />

                                <html:hidden name="empresaForm" property="empresa_pesq.cd_emp"/>
                                <html:hidden name="empresaForm" property="empresa_pesq.nm_emp"/>
                                <html:hidden name="empresaForm"
property="empresa_pesq.razao_social"/>
                                <html:hidden name="empresaForm" property="empresa_pesq.endereco"/>
                                <html:hidden name="empresaForm" property="empresa_pesq.bairro"/>
                                <html:hidden name="empresaForm" property="empresa_pesq.cidade"/>
                                <html:hidden name="empresaForm" property="empresa_pesq.uf"/>
                                <html:hidden name="empresaForm" property="empresa_pesq.cep"/>
                                <html:hidden name="empresaForm" property="empresa_pesq.cnpj"/>
                                <html:hidden name="empresaForm"
property="empresa_pesq.insc_estadual"/>
                                <html:hidden name="empresaForm" property="empresa_pesq.telefone"/>
                                <html:hidden name="empresaForm" property="empresa_pesq.fax"/>
                                <html:hidden name="empresaForm" property="empresa_pesq.email"/>
                                <html:hidden name="empresaForm"
property="empresa_pesq.nm_contato"/>
                                <html:hidden name="empresaForm"
property="empresa_pesq.cd_atividade"/>
                        </logic:notEqual>

                        <logic:equal name="empresaForm" property="mostraConsulta" value="S">
                                <tr>
                                    <td colspan="5" align="center">
                                                <html:submit property="evento">
                                                        <bean:message key="nota_fluxo.lista"/>
                                                </html:submit>
                                    </td>
                                </tr>
                            <tr>
                                <td width="15%" align="right">Empresa:
                                <td width="35%">
                                        <html:text maxlength="4" size="4" style="text-align: right"
name="empresaForm" property="empresa_pesq.cd_emp" />
```

```
                                        <html:text maxlength="30" size="25" name="empresaForm"
property="empresa_pesq.nm_emp" />
                              <td width="15%" align="right">
                              <td width="35%">
                            <tr>
                              <td width="15%" align="right">Razão Social:
                              <td width="35%"><html:text maxlength="30" size="34" name="empresaForm"
property="empresa_pesq.razao_social" />
                              <td width="15%" align="right">Cidade:
                              <td width="35%"><html:text maxlength="30" size="20" name="empresaForm"
property="empresa_pesq.cidade" />     UF: <html:text maxlength="2" size="2"
name="empresaForm" property="empresa_pesq.uf" />
                            <tr>
                              <td width="15%" align="right">Endereço:
                              <td width="35%"><html:text maxlength="30" size="34" name="empresaForm"
property="empresa_pesq.endereco" />
                              <td width="15%" align="right">Bairro:
                              <td width="35%"><html:text maxlength="30" size="20" name="empresaForm"
property="empresa_pesq.bairro" />
                            <tr>
                              <td width="15%" align="right">CEP:
                              <td width="35%"><html:text maxlength="8"  size="10" name="empresaForm"
property="empresa_pesq.cep" />
                              <td width="15%" align="right">Atividade:
                              <td width="35%"><html:text maxlength="4"  size="4" name="empresaForm"
property="empresa_pesq.cd_atividade" />
                            <tr>
                              <td width="15%" align="right">CNPJ:
                              <td width="35%"><html:text maxlength="16" size="15" name="empresaForm"
property="empresa_pesq.cnpj" />
                              <td width="15%" align="right">Insc Estadual:
                              <td width="35%"><html:text maxlength="10" size="15" name="empresaForm"
property="empresa_pesq.insc_estadual" />
                            <tr>
                              <td width="15%" align="right">Telefone:
                              <td width="35%"><html:text maxlength="10" size="15" name="empresaForm"
property="empresa_pesq.telefone" />
                              <td width="15%" align="right">Fax:
                              <td width="35%"><html:text maxlength="10" size="15" name="empresaForm"
property="empresa_pesq.fax" />
                            <tr>
                              <td width="15%" align="right">Contato:
                              <td width="35%"><html:text maxlength="30" size="20" name="empresaForm"
property="empresa_pesq.nm_contato" />
                              <td width="15%" align="right">Email:
                              <td width="35%"><html:text maxlength="30" size="34" name="empresaForm"
property="empresa_pesq.email" />
                          </logic:equal>
                          </table>
                  </td>
          </tr>
</table>

<html:messages id="msg" message="true">
        <bean:write name="msg"/><br>
</html:messages>

</html:form>
</body>
</html>
```

editar_movimento_contas.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<html>
<head>
<title>Movimento_contas</title>
</head>

<body bgcolor="#ffffff">
<html:form action="/movimento_contas">
<input type="hidden" name="action">
<table width="95%" border="1" bgcolor="#c8c8c8" bordercolor="#ffffff" align="center">
        <tr>
                <td>
                        <table width="100%" border="0" bgcolor="#ffffff">
                          <tr>
                        <logic:notEqual name="movimento_contasForm" property="mostraConsulta"
value="S">
                                <html:hidden name="movimento_contasForm"
property="movimento_contas.oid"/>
                              <td colspan="5" align="center">
                                        <logic:present name="movimento_contasForm"
property="movimento_contas.oid">
                                                <html:submit property="evento">
                                                        <bean:message key="nota_fluxo.alterar"/>
```

```
                                        </html:submit>
                                    </logic:present>
                                    <logic:notPresent name="movimento_contasForm"
property="movimento_contas.oid">
                                        <html:submit property="evento">
                                            <bean:message key="nota_fluxo.inserir"/>
                                        </html:submit>
                                    </logic:notPresent>
                                </td>
                            </tr>

                            <tr>
                                <td  align="center">Data:
                                    <logic:present name="movimento_contasForm"
property="movimento_contas.oid">
                                        <html:text maxlength="10" size="10" style="text-
align: right" name="movimento_contasForm" readonly="true" property="movimento_contas.dt_movto" />
                                    </logic:present>
                                    <logic:notPresent name="movimento_contasForm"
property="movimento_contas.oid">
                                        <html:text maxlength="10" size="10" style="text-align:
right" name="movimento_contasForm" property="movimento_contas.dt_movto" />
                                    </logic:notPresent>
                                    Nr:
                                    <logic:present name="movimento_contasForm"
property="movimento_contas.oid">
                                        <html:text maxlength="4" size="4" style="text-
align: right" name="movimento_contasForm" readonly="true" property="movimento_contas.nr_movto" />
                                    </logic:present>
                                    <logic:notPresent name="movimento_contasForm"
property="movimento_contas.oid">
                                        <html:text maxlength="4" size="4" style="text-align:
right" name="movimento_contasForm" property="movimento_contas.nr_movto" />
                                    </logic:notPresent>
                                    Emp:<html:text maxlength="4" size="4"
name="movimento_contasForm" property="movimento_contas.cd_emp" />
                                    Conta:<html:text maxlength="4" size="4"
name="movimento_contasForm" property="movimento_contas.nr_conta" />
                                    Descrição:<html:text maxlength="30" size="20"
name="movimento_contasForm" property="movimento_contas.ds_movto" />
                                    Valor:<html:text maxlength="10" size="10"
name="movimento_contasForm" property="movimento_contas.vl_movto" />
                                <html:hidden name="movimento_contasForm"
property="movimento_contas_pesq.dt_movto"/>
                                <html:hidden name="movimento_contasForm"
property="movimento_contas_pesq.nr_movto"/>
                                <html:hidden name="movimento_contasForm"
property="movimento_contas_pesq.cd_emp"/>
                                <html:hidden name="movimento_contasForm"
property="movimento_contas_pesq.nr_conta"/>
                                <html:hidden name="movimento_contasForm"
property="movimento_contas_pesq.ds_movto"/>
                                <html:hidden name="movimento_contasForm"
property="movimento_contas_pesq.vl_movto"/>
                            </logic:notEqual>

                            <logic:equal name="movimento_contasForm" property="mostraConsulta"
value="S">
                                <tr>
                                    <td colspan="5" align="center">
                                        <html:submit property="evento">
                                            <bean:message key="nota_fluxo.lista"/>
                                        </html:submit>
                                    </td>
                                </tr>
                              <tr>
                                <td align="center">
                                    Data:<html:text maxlength="10" size="10"
name="movimento_contasForm" property="movimento_contas_pesq.dt_movto" />
                                        Nr: <html:text maxlength="4" size="4"
name="movimento_contasForm" property="movimento_contas_pesq.nr_movto" />
                                        Emp: <html:text maxlength="4" size="4"
name="movimento_contasForm" property="movimento_contas_pesq.cd_emp" />
                                        Conta: <html:text maxlength="4" size="4"
name="movimento_contasForm" property="movimento_contas_pesq.nr_conta" />
                                        Descrição: <html:text maxlength="30" size="20"
name="movimento_contasForm" property="movimento_contas_pesq.ds_movto" />
                                        Valor: <html:text maxlength="10" size="10"
name="movimento_contasForm" property="movimento_contas_pesq.vl_movto" />
                            </logic:equal>
                            </table>
                    </td>
            </tr>
    </table>

    <html:messages id="msg" message="true">
            <bean:write name="msg"/><br>
    </html:messages>
```

```
</html:form>
</body>
</html>
```

editar_nota_fiscal.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<html>
<head>
<title>Nota_fiscal</title>
</head>

<body bgcolor="#ffffff">
<html:form action="/nota_fiscal">
<input type="hidden" name="action">
<table width="75%" border="1" bgcolor="#c8c8c8" bordercolor="#ffffff" align="center">
        <tr>
                <td>
                        <table width="100%" border="0" bgcolor="#ffffff">
                          <tr>
                        <logic:notEqual name="nota_fiscalForm" property="mostraConsulta" value="S">
                                <html:hidden name="nota_fiscalForm" property="nota_fiscal.oid"/>
                            <td colspan="5" align="center">
                                        <logic:present name="nota_fiscalForm"
property="nota_fiscal.oid">
                                                <html:submit property="evento">
                                                        <bean:message key="nota_fluxo.alterar"/>
                                                </html:submit>
                                        </logic:present>
                                        <logic:notPresent name="nota_fiscalForm"
property="nota_fiscal.oid">
                                                <html:submit property="evento">
                                                        <bean:message key="nota_fluxo.inserir"/>
                                                </html:submit>
                                        </logic:notPresent>
                            </td>
                          </tr>

                          <tr>
                                <td width="15%" align="right">Empresa:
                                <td width="35%">
                                        <logic:present name="nota_fiscalForm"
property="nota_fiscal.oid">
                                                <html:text maxlength="4" size="4" style="text-
align: right" name="nota_fiscalForm" readonly="true" property="nota_fiscal.cd_emp" />
                                        </logic:present>
                                        <logic:notPresent name="nota_fiscalForm"
property="nota_fiscal.oid">
                                            <html:text maxlength="4" size="4" style="text-align:
right" name="nota_fiscalForm" property="nota_fiscal.cd_emp" />
                                        </logic:notPresent>
                                        Nr:
                                        <logic:present name="nota_fiscalForm"
property="nota_fiscal.oid">
                                                <html:text maxlength="4" size="4" style="text-
align: right" name="nota_fiscalForm" readonly="true" property="nota_fiscal.nr_nota" />
                                        </logic:present>
                                        <logic:notPresent name="nota_fiscalForm"
property="nota_fiscal.oid">
                                            <html:text maxlength="4" size="4" style="text-align:
right" name="nota_fiscalForm" property="nota_fiscal.nr_nota" />
                                        </logic:notPresent>
                            <td width="15%" align="right">
                            <td width="35%">
                          <tr>
                            <td width="15%" align="right">Data:
                            <td width="35%"><html:text maxlength="10" size="10"
name="nota_fiscalForm" property="nota_fiscal.dt_ref" />
                            <td width="15%" align="right">Cliente:
                            <td width="35%"><html:text maxlength="8" size="8"
name="nota_fiscalForm" property="nota_fiscal.cd_cli" />
                          <tr>
                            <td width="15%" align="right">Dt Vencimento:
                            <td width="35%"><html:text maxlength="10" size="10"
name="nota_fiscalForm" property="nota_fiscal.dt_vencto" />
                            <td width="15%" align="right">Vl Bruto:
                            <td width="35%"><html:text maxlength="8" size="8"
name="nota_fiscalForm" property="nota_fiscal.vl_bruto" />
                          <tr>
                            <td width="15%" align="right">Vl Liquido:
                            <td width="35%"><html:text maxlength="8"  size="8"
name="nota_fiscalForm" property="nota_fiscal.vl_liquido" />
                            <td width="15%" align="right">Conta:
                            <td width="35%"><html:text maxlength="8"  size="8"
name="nota_fiscalForm" property="nota_fiscal.cd_conta" />
```

```
                                <html:hidden name="nota_fiscalForm"
property="nota_fiscal_pesq.cd_emp"/>
                                <html:hidden name="nota_fiscalForm"
property="nota_fiscal_pesq.nr_nota"/>
                                <html:hidden name="nota_fiscalForm"
property="nota_fiscal_pesq.dt_ref"/>
                                <html:hidden name="nota_fiscalForm"
property="nota_fiscal_pesq.cd_cli"/>
                                <html:hidden name="nota_fiscalForm"
property="nota_fiscal_pesq.vl_bruto"/>
                                <html:hidden name="nota_fiscalForm"
property="nota_fiscal_pesq.dt_vencto"/>
                                <html:hidden name="nota_fiscalForm"
property="nota_fiscal_pesq.vl_liquido"/>
                                <html:hidden name="nota_fiscalForm"
property="nota_fiscal_pesq.cd_conta"/>
                        </logic:notEqual>

                        <logic:equal name="nota_fiscalForm" property="mostraConsulta" value="S">
                                <tr>
                                    <td colspan="5" align="center">
                                                <html:submit property="evento">
                                                        <bean:message key="nota_fluxo.lista"/>
                                                </html:submit>
                                    </td>
                                </tr>
                          <tr>
                                <td width="15%" align="right">Empresa:
                                <td width="35%">
                                        <html:text maxlength="4" size="4" style="text-align: right"
name="nota_fiscalForm" property="nota_fiscal_pesq.cd_emp" />
                                        Nr
                                        <html:text maxlength="4" size="4"name="nota_fiscalForm"
property="nota_fiscal_pesq.nr_nota" />
                            <td width="15%" align="right">
                            <td width="35%">
                          <tr>
                            <td width="15%" align="right">Data:
                            <td width="35%"><html:text maxlength="10" size="10"
name="nota_fiscalForm" property="nota_fiscal_pesq.dt_ref" />
                            <td width="15%" align="right">Cliente:
                            <td width="35%"><html:text maxlength="8" size="8"
name="nota_fiscalForm" property="nota_fiscal_pesq.cd_cli" />
                          <tr>
                            <td width="15%" align="right">Dt Vencimento:
                            <td width="35%"><html:text maxlength="10" size="10"
name="nota_fiscalForm" property="nota_fiscal_pesq.dt_vencto" />
                            <td width="15%" align="right">Vl Bruto:
                            <td width="35%"><html:text maxlength="8" size="8"
name="nota_fiscalForm" property="nota_fiscal_pesq.vl_bruto" />
                          <tr>
                            <td width="15%" align="right">Vl Liquido:
                            <td width="35%"><html:text maxlength="8"  size="8"
name="nota_fiscalForm" property="nota_fiscal_pesq.vl_liquido" />
                            <td width="15%" align="right">Conta:
                            <td width="35%"><html:text maxlength="8"  size="8"
name="nota_fiscalForm" property="nota_fiscal_pesq.cd_conta" />
                        </logic:equal>
                        </table>
                </td>
        </tr>
</table>

<html:messages id="msg" message="true">
        <bean:write name="msg"/><br>
</html:messages>

</html:form>
</body>
</html>
```

editar_nota_item.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<html>
<head>
<title>Nota_item</title>
</head>

<body bgcolor="#ffffff">
<html:form action="/nota_item">
<input type="hidden" name="action">
<table width="75%" border="1" bgcolor="#c8c8c8" bordercolor="#ffffff" align="center">
        <tr>
                <td>
```

```
                        <table width="100%" border="0" bgcolor="#ffffff">
                          <tr>
                        <logic:notEqual name="nota_itemForm" property="mostraConsulta" value="S">
                                <html:hidden name="nota_itemForm" property="nota_item.oid"/>
                            <td colspan="5" align="center">
                                        <logic:present name="nota_itemForm"
property="nota_item.oid">
                                                <html:submit property="evento">
                                                        <bean:message key="nota_fluxo.alterar"/>
                                                </html:submit>
                                        </logic:present>
                                        <logic:notPresent name="nota_itemForm"
property="nota_item.oid">
                                                <html:submit property="evento">
                                                        <bean:message key="nota_fluxo.inserir"/>
                                                </html:submit>
                                        </logic:notPresent>
                            </td>
                          </tr>

                          <tr>
                                <td  align="center">
                                        Empresa:
                                        <logic:present name="nota_itemForm"
property="nota_item.oid">
                                                <html:text maxlength="4" size="4" style="text-
align: right" name="nota_itemForm" readonly="true" property="nota_item.cd_emp" />
                                        </logic:present>
                                        <logic:notPresent name="nota_itemForm"
property="nota_item.oid">
                                            <html:text maxlength="4" size="4" style="text-align:
right" name="nota_itemForm" property="nota_item.cd_emp" />
                                        </logic:notPresent>
                                        Nota:
                                        <logic:present name="nota_itemForm"
property="nota_item.oid">
                                                <html:text maxlength="10" size="10" style="text-
align: right" name="nota_itemForm" readonly="true" property="nota_item.nr_nota" />
                                        </logic:present>
                                        <logic:notPresent name="nota_itemForm"
property="nota_item.oid">
                                            <html:text maxlength="10" size="10" style="text-align:
right" name="nota_itemForm" property="nota_item.nr_nota" />
                                        </logic:notPresent>
                                        Cnae:
                                        <logic:present name="nota_itemForm"
property="nota_item.oid">
                                                <html:text maxlength="10" size="10" style="text-
align: right" name="nota_itemForm" readonly="true" property="nota_item.cd_cnae" />
                                        </logic:present>
                                        <logic:notPresent name="nota_itemForm"
property="nota_item.oid">
                                            <html:text maxlength="10" size="10" style="text-align:
right" name="nota_itemForm" property="nota_item.cd_cnae" />
                                        </logic:notPresent>
                                        Valor:<html:text maxlength="8" size="8"
name="nota_itemForm" property="nota_item.vl_item" />

                                <html:hidden name="nota_itemForm" property="nota_item.cd_emp"/>
                                <html:hidden name="nota_itemForm" property="nota_item.nr_nota"/>
                                <html:hidden name="nota_itemForm"
property="nota_item_pesq.cd_emp"/>
                                <html:hidden name="nota_itemForm"
property="nota_item_pesq.nr_nota"/>
                                <html:hidden name="nota_itemForm"
property="nota_item_pesq.cd_cnae"/>
                                <html:hidden name="nota_itemForm"
property="nota_item_pesq.vl_item"/>
                        </logic:notEqual>

                        <logic:equal name="nota_itemForm" property="mostraConsulta" value="S">
                                <tr>
                                    <td colspan="5" align="center">
                                                <html:submit property="evento">
                                                        <bean:message key="nota_fluxo.lista"/>
                                                </html:submit>
                                    </td>
                                </tr>
                          <tr>
                            <td align="center">
                                Empresa:<html:text maxlength="4" size="4" name="nota_itemForm"
property="nota_item_pesq.cd_emp" />
                                    Nota: <html:text maxlength="10" size="10" name="nota_itemForm"
property="nota_item_pesq.nr_nota" />
                                    Cnae: <html:text maxlength="10" size="10" name="nota_itemForm"
property="nota_item_pesq.cd_cnae" />
                                    Valor: <html:text maxlength="8" size="8" name="nota_itemForm"
property="nota_item_pesq.vl_item" />
```

```
                                <html:hidden name="nota_itemForm" property="nota_item.cd_emp"/>
                                <html:hidden name="nota_itemForm" property="nota_item.nr_nota"/>
                        </logic:equal>
                        </table>
                </td>
        </tr>
</table>

<html:messages id="msg" message="true">
        <bean:write name="msg"/><br>
</html:messages>

</html:form>
</body>
</html>
```

editar_ramo_Atividade.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<html>
<head>
<title>Ramo_atividade</title>
</head>

<body bgcolor="#ffffff">
<html:form action="/ramo_atividade">
<input type="hidden" name="action">
<table width="75%" border="1" bgcolor="#c8c8c8" bordercolor="#ffffff" align="center">
        <tr>
                <td>
                        <table width="100%" border="0" bgcolor="#ffffff">
                          <tr>
                        <logic:notEqual name="ramo_atividadeForm" property="mostraConsulta"
value="S">
                                <html:hidden name="ramo_atividadeForm"
property="ramo_atividade.oid"/>
                            <td colspan="5" align="center">
                                        <logic:present name="ramo_atividadeForm"
property="ramo_atividade.oid">
                                                <html:submit property="evento">
                                                        <bean:message key="nota_fluxo.alterar"/>
                                                </html:submit>
                                        </logic:present>
                                        <logic:notPresent name="ramo_atividadeForm"
property="ramo_atividade.oid">
                                                <html:submit property="evento">
                                                        <bean:message key="nota_fluxo.inserir"/>
                                                </html:submit>
                                        </logic:notPresent>
                            </td>
                          </tr>

                          <tr>
                                <td  align="center">Ramo_atividade:
                                        <logic:present name="ramo_atividadeForm"
property="ramo_atividade.oid">
                                                <html:text maxlength="4" size="4" style="text-
align: right" name="ramo_atividadeForm" readonly="true" property="ramo_atividade.cd_atividade" />
                                        </logic:present>
                                        <logic:notPresent name="ramo_atividadeForm"
property="ramo_atividade.oid">
                                            <html:text maxlength="4" size="4" style="text-align:
right" name="ramo_atividadeForm" property="ramo_atividade.cd_atividade" />
                                        </logic:notPresent>
                                        Descrição:<html:text maxlength="30" size="20"
name="ramo_atividadeForm" property="ramo_atividade.ds_atividade" />

                                <html:hidden name="ramo_atividadeForm"
property="ramo_atividade_pesq.cd_atividade"/>
                                <html:hidden name="ramo_atividadeForm"
property="ramo_atividade_pesq.ds_atividade"/>
                        </logic:notEqual>

                        <logic:equal name="ramo_atividadeForm" property="mostraConsulta" value="S">
                                <tr>
                                    <td colspan="5" align="center">
                                                <html:submit property="evento">
                                                        <bean:message key="nota_fluxo.lista"/>
                                                </html:submit>
                                        </td>
                                </tr>
                          <tr>
                            <td align="center">
                                Código:<html:text maxlength="30" size="34"
name="ramo_atividadeForm" property="ramo_atividade_pesq.cd_atividade" />
```

```
                                    Descrição: <html:text maxlength="30" size="20"
name="ramo_atividadeForm" property="ramo_atividade_pesq.ds_atividade" />
                        </logic:equal>
                        </table>
                </td>
        </tr>
</table>

<html:messages id="msg" message="true">
        <bean:write name="msg"/><br>
</html:messages>

</html:form>
</body>
</html>
```

editar_tipo_contas_pagar.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<html>
<head>
<title>Tipo_contas_pagar</title>
</head>

<body bgcolor="#ffffff">
<html:form action="/tipo_contas_pagar">
<input type="hidden" name="action">
<table width="75%" border="1" bgcolor="#c8c8c8" bordercolor="#ffffff" align="center">
        <tr>
                <td>
                        <table width="100%" border="0" bgcolor="#ffffff">
                          <tr>
                        <logic:notEqual name="tipo_contas_pagarForm" property="mostraConsulta"
value="S">
                                <html:hidden name="tipo_contas_pagarForm"
property="tipo_contas_pagar.oid"/>
                            <td colspan="5" align="center">
                                        <logic:present name="tipo_contas_pagarForm"
property="tipo_contas_pagar.oid">
                                                <html:submit property="evento">
                                                        <bean:message key="nota_fluxo.alterar"/>
                                                </html:submit>
                                        </logic:present>
                                        <logic:notPresent name="tipo_contas_pagarForm"
property="tipo_contas_pagar.oid">
                                                <html:submit property="evento">
                                                        <bean:message key="nota_fluxo.inserir"/>
                                                </html:submit>
                                        </logic:notPresent>
                            </td>
                          </tr>

                          <tr>
                                <td  align="center">Tipo_contas_pagar:
                                        <logic:present name="tipo_contas_pagarForm"
property="tipo_contas_pagar.oid">
                                                <html:text maxlength="4" size="4" style="text-
align: right" name="tipo_contas_pagarForm" readonly="true" property="tipo_contas_pagar.cd_tp_pagar"
/>
                                        </logic:present>
                                        <logic:notPresent name="tipo_contas_pagarForm"
property="tipo_contas_pagar.oid">
                                            <html:text maxlength="4" size="4" style="text-align:
right" name="tipo_contas_pagarForm" property="tipo_contas_pagar.cd_tp_pagar" />
                                        </logic:notPresent>
                                        Descrição:<html:text maxlength="30" size="20"
name="tipo_contas_pagarForm" property="tipo_contas_pagar.ds_tp_pagar" />

                                <html:hidden name="tipo_contas_pagarForm"
property="tipo_contas_pagar_pesq.cd_tp_pagar"/>
                                <html:hidden name="tipo_contas_pagarForm"
property="tipo_contas_pagar_pesq.ds_tp_pagar"/>
                        </logic:notEqual>

                        <logic:equal name="tipo_contas_pagarForm" property="mostraConsulta"
value="S">
                                <tr>
                                    <td colspan="5" align="center">
                                        <html:submit property="evento">
                                                <bean:message key="nota_fluxo.lista"/>
                                        </html:submit>
                                    </td>
                                </tr>
                          <tr>
                            <td align="center">
```

```
                                Código:<html:text maxlength="30" size="34"
name="tipo_contas_pagarForm" property="tipo_contas_pagar_pesq.cd_tp_pagar" />
                                        Descrição: <html:text maxlength="30" size="20"
name="tipo_contas_pagarForm" property="tipo_contas_pagar_pesq.ds_tp_pagar" />
                        </logic:equal>
                        </table>
                </td>
        </tr>
</table>

<html:messages id="msg" message="true">
        <bean:write name="msg"/><br>
</html:messages>

</html:form>
</body>
</html>
```

editar_tipo_contas_receber.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<html>
<head>
<title>Tipo_contas_receber</title>
</head>

<body bgcolor="#ffffff">
<html:form action="/tipo_contas_receber">
<input type="hidden" name="action">
<table width="75%" border="1" bgcolor="#c8c8c8" bordercolor="#ffffff" align="center">
        <tr>
                <td>
                        <table width="100%" border="0" bgcolor="#ffffff">
                          <tr>
                        <logic:notEqual name="tipo_contas_receberForm" property="mostraConsulta"
value="S">
                                <html:hidden name="tipo_contas_receberForm"
property="tipo_contas_receber.oid"/>
                            <td colspan="5" align="center">
                                <logic:present name="tipo_contas_receberForm"
property="tipo_contas_receber.oid">
                                        <html:submit property="evento">
                                                <bean:message key="nota_fluxo.alterar"/>
                                        </html:submit>
                                </logic:present>
                                <logic:notPresent name="tipo_contas_receberForm"
property="tipo_contas_receber.oid">
                                        <html:submit property="evento">
                                                <bean:message key="nota_fluxo.inserir"/>
                                        </html:submit>
                                </logic:notPresent>
                              </td>
                            </tr>

                            <tr>
                                  <td  align="center">Tipo_contas_receber:
                                <logic:present name="tipo_contas_receberForm"
property="tipo_contas_receber.oid">
                                        <html:text maxlength="4" size="4" style="text-
align: right" name="tipo_contas_receberForm" readonly="true"
property="tipo_contas_receber.cd_tp_receber" />
                                </logic:present>
                                <logic:notPresent name="tipo_contas_receberForm"
property="tipo_contas_receber.oid">
                                    <html:text maxlength="4" size="4" style="text-align:
right" name="tipo_contas_receberForm" property="tipo_contas_receber.cd_tp_receber" />
                                </logic:notPresent>
                                Descrição:<html:text maxlength="30" size="20"
name="tipo_contas_receberForm" property="tipo_contas_receber.ds_tp_receber" />

                                <html:hidden name="tipo_contas_receberForm"
property="tipo_contas_receber_pesq.cd_tp_receber"/>
                                <html:hidden name="tipo_contas_receberForm"
property="tipo_contas_receber_pesq.ds_tp_receber"/>
                        </logic:notEqual>

                        <logic:equal name="tipo_contas_receberForm" property="mostraConsulta"
value="S">
                                <tr>
                                    <td colspan="5" align="center">
                                        <html:submit property="evento">
                                                <bean:message key="nota_fluxo.lista"/>
                                        </html:submit>
                                    </td>
                                </tr>
                          <tr>
```

```
                            <td align="center">
                                Código:<html:text maxlength="30" size="34"
name="tipo_contas_receberForm" property="tipo_contas_receber_pesq.cd_tp_receber" />
                                    Descrição: <html:text maxlength="30" size="20"
name="tipo_contas_receberForm" property="tipo_contas_receber_pesq.ds_tp_receber" />
                        </logic:equal>
                        </table>
                </td>
        </tr>
</table>

<html:messages id="msg" message="true">
        <bean:write name="msg"/><br>
</html:messages>

</html:form>
</body>
</html>
```

lista_cliente.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<html>
<head>
<title>Cliente</title>
<script>
        function confirma(){
                if (confirm("Deseja apagar este registro?") == true) {
                        return true;
                } else {
                        return false;
                }
        }
</script>
</head>

<body bgcolor="#ffffff">
<html:form action="/cliente">

<input type="hidden" name="oid">
<input type="hidden" name="p_pagina">

<jsp:include page="barra.jsp">
        <jsp:param name="p_nm_form" value="clienteForm" />
</jsp:include>

<table width="75%" border="1" bgcolor="#c8c8c8" bordercolor="#ffffff" align="center">
        <tr>
                <td>
                        <table width="100%" border="0" bgcolor="ffffff" bordercolor="#000000">
                        <logic:iterate id="listaBeans" name="clienteForm" property="cliente_lista">
                          <tr>
                                <td width="15%" align="right">Cliente:
                                <td width="35%"><bean:write name="listaBeans" property="cd_cli"/>
                                           <bean:write name="listaBeans"
property="nm_cli"/>
                            <td width="15%" align="right">
                            <%java.util.HashMap params;%>
                            <td width="35%">
                                        <bean:define id="new_oid" name="listaBeans"
property="oid"/>
<%
                                    params = new java.util.HashMap();
                                        params.put("evento","Apagar");
                                        params.put("oid",new_oid);
                                        params.put("p_pagina",request.getParameter("p_pagina"));
                                        request.setAttribute("parametros", params);
%>                                      <html:link page="/cliente.do" name="parametros"
onclick="return confirma()">
                                                <bean:message key="nota_fluxo.apagar"/>
                                        </html:link>
                                             
                                        <bean:define id="new_oid" name="listaBeans"
property="oid"/>
<%
                                    params = new java.util.HashMap();
                                        params.put("evento","Entra Alterar");
                                        params.put("oid",new_oid);
                                        params.put("p_pagina",request.getParameter("p_pagina"));
                                        request.setAttribute("parametros", params);
%>                                      <html:link page="/cliente.do" name="parametros" >
                                                <bean:message key="nota_fluxo.entra_alterar"/>
                                        </html:link>

                            <tr>
                              <td width="15%" align="right">Razão Social:
```

```
                              <td width="35%"><bean:write name="listaBeans" property="razao_social"/>
                              <td width="15%" align="right">Cidade:
                              <td width="35%"><bean:write name="listaBeans" property="cidade"/>
                                      UF: <bean:write name="listaBeans" property="uf"/>
                            <tr>
                              <td width="15%" align="right">Endereço:
                              <td width="35%"><bean:write name="listaBeans" property="endereco"/>
                              <td width="15%" align="right">Bairro:
                              <td width="35%"><bean:write name="listaBeans" property="bairro"/>
                            <tr>
                              <td width="15%" align="right">CEP:
                              <td width="35%"><bean:write name="listaBeans" property="cep"/>
                              <td width="15%" align="right">Atividade:
                              <td width="35%"><bean:write name="listaBeans" property="cd_atividade"/>
                            <tr>
                              <td width="15%" align="right">CNPJ:
                              <td width="35%"><bean:write name="listaBeans" property="cnpj"/>
                              <td width="15%" align="right">Insc Estadual:
                              <td width="35%"><bean:write name="listaBeans"
property="insc_estadual"/>
                            <tr>
                              <td width="15%" align="right">Telefone:
                              <td width="35%"><bean:write name="listaBeans" property="telefone"/>
                              <td width="15%" align="right">Fax:
                              <td width="35%"><bean:write name="listaBeans" property="fax"/>
                            <tr>
                              <td width="15%" align="right">Contato:
                              <td width="35%"><bean:write name="listaBeans" property="nm_contato"/>
                              <td width="15%" align="right">Email:
                              <td width="35%"><bean:write name="listaBeans" property="email"/>
                          </logic:iterate>
                          </table>
                  </td>
          </tr>
</table>

<html:hidden name="clienteForm" property="cliente_pesq.cd_cli"/>
<html:hidden name="clienteForm" property="cliente_pesq.nm_cli"/>
<html:hidden name="clienteForm" property="cliente_pesq.razao_social"/>
<html:hidden name="clienteForm" property="cliente_pesq.endereco"/>
<html:hidden name="clienteForm" property="cliente_pesq.bairro"/>
<html:hidden name="clienteForm" property="cliente_pesq.cidade"/>
<html:hidden name="clienteForm" property="cliente_pesq.uf"/>
<html:hidden name="clienteForm" property="cliente_pesq.cep"/>
<html:hidden name="clienteForm" property="cliente_pesq.cnpj"/>
<html:hidden name="clienteForm" property="cliente_pesq.insc_estadual"/>
<html:hidden name="clienteForm" property="cliente_pesq.telefone"/>
<html:hidden name="clienteForm" property="cliente_pesq.fax"/>
<html:hidden name="clienteForm" property="cliente_pesq.email"/>
<html:hidden name="clienteForm" property="cliente_pesq.nm_contato"/>
<html:hidden name="clienteForm" property="cliente_pesq.cd_atividade"/>

<jsp:include page="contadores.jsp">
        <jsp:param name="p_nm_form" value="clienteForm" />
</jsp:include>

<html:messages id="msg" message="true">
        <bean:write name="msg"/><br>
</html:messages>

</html:form>


</body>
</html>
```

lista_cnae.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<html>
<head>
<title>Cnae</title>
<script>
        function confirma(){
                if (confirm("Deseja apagar este registro?") == true) {
                        return true;
                } else {
                        return false;
                }
        }
</script>
</head>

<body bgcolor="#ffffff">
<html:form action="/cnae">
```

```
<input type="hidden" name="oid">
<input type="hidden" name="p_pagina">

<jsp:include page="barra.jsp">
        <jsp:param name="p_nm_form" value="cnaeForm" />
</jsp:include>

<table width="75%" border="1" bgcolor="#c8c8c8" bordercolor="#ffffff" align="center">
        <tr>
                <td>
                        <table width="100%" border="0" bgcolor="ffffff" bordercolor="#000000">
                          <tr>
                            <td align="right">Código
                            <td class="TxtGeral">Descrição
                        <logic:iterate id="listaBeans" name="cnaeForm" property="cnae_lista">
                          <tr>
                            <td width="10%" align="right"><bean:write name="listaBeans"
property="cd_cnae"/>
                            <td width="60%"><bean:write name="listaBeans" property="ds_cnae"/>
                            <%java.util.HashMap params;%>
                            <td width="35%">
                                        <bean:define id="new_oid" name="listaBeans"
property="oid"/>
<%
                                    params = new java.util.HashMap();
                                        params.put("evento","Apagar");
                                        params.put("oid",new_oid);
                                        params.put("p_pagina",request.getParameter("p_pagina"));
                                        request.setAttribute("parametros", params);
%>                                      <html:link page="/cnae.do" name="parametros"
onclick="return confirma()">
                                                <bean:message key="nota_fluxo.apagar"/>
                                        </html:link>
                                             
                                        <bean:define id="new_oid" name="listaBeans"
property="oid"/>
<%
                                    params = new java.util.HashMap();
                                        params.put("evento","Entra Alterar");
                                        params.put("oid",new_oid);
                                        params.put("p_pagina",request.getParameter("p_pagina"));
                                        request.setAttribute("parametros", params);
%>                                      <html:link page="/cnae.do" name="parametros" >
                                                <bean:message key="nota_fluxo.entra_alterar"/>
                                        </html:link>

                        </logic:iterate>
                        </table>
                </td>
        </tr>
</table>

<html:hidden name="cnaeForm" property="cnae_pesq.cd_cnae"/>
<html:hidden name="cnaeForm" property="cnae_pesq.ds_cnae"/>

<jsp:include page="contadores.jsp">
        <jsp:param name="p_nm_form" value="cnaeForm" />
</jsp:include>

<html:messages id="msg" message="true">
        <bean:write name="msg"/><br>
</html:messages>

</html:form>


</body>
</html>
```

lista_contas_pagar.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<html>
<head>
<title>Contas_pagar</title>
<script>
        function confirma(){
                if (confirm("Deseja apagar este registro?") == true) {
                        return true;
                } else {
                        return false;
                }
        }
</script>
</head>
```

```
<body bgcolor="#ffffff">
<html:form action="/contas_pagar">

<input type="hidden" name="oid">
<input type="hidden" name="p_pagina">

<jsp:include page="barra.jsp">
        <jsp:param name="p_nm_form" value="contas_pagarForm" />
</jsp:include>

<table width="75%" border="1" bgcolor="#c8c8c8" bordercolor="#ffffff" align="center">
        <tr>
                <td>
                        <table width="100%" border="0" bgcolor="ffffff" bordercolor="#000000">
                        <logic:iterate id="listaBeans" name="contas_pagarForm"
property="contas_pagar_lista">
                          <tr>
                                <td width="15%" align="right">Empresa:
                                <td width="35%"><bean:write name="listaBeans" property="cd_emp"/>
                                          Nr.  <bean:write name="listaBeans"
property="cd_pagar"/>
                            <td width="15%" align="right">
                            <%java.util.HashMap params;%>
                            <td width="35%">
                                        <bean:define id="new_oid" name="listaBeans"
property="oid"/>
<%
                                    params = new java.util.HashMap();
                                        params.put("evento","Apagar");
                                        params.put("oid",new_oid);
                                        params.put("p_pagina",request.getParameter("p_pagina"));
                                        request.setAttribute("parametros", params);
%>                                      <html:link page="/contas_pagar.do" name="parametros"
onclick="return confirma()">
                                                <bean:message key="nota_fluxo.apagar"/>
                                        </html:link>
                                             
                                        <bean:define id="new_oid" name="listaBeans"
property="oid"/>
<%
                                    params = new java.util.HashMap();
                                        params.put("evento","Entra Alterar");
                                        params.put("oid",new_oid);
                                        params.put("p_pagina",request.getParameter("p_pagina"));
                                        request.setAttribute("parametros", params);
%>                                      <html:link page="/contas_pagar.do" name="parametros" >
                                                <bean:message key="nota_fluxo.entra_alterar"/>
                                        </html:link>

                            <tr>
                              <td width="15%" align="right">Dt Vencimento:
                              <td width="35%"><bean:write name="listaBeans" property="dt_vencto"/>
                              <td width="15%" align="right">Dt Pagamento:
                              <td width="35%"><bean:write name="listaBeans" property="dt_pagto"/>
                            <tr>
                              <td width="15%" align="right">Vl Acréscimos:
                              <td width="35%"><bean:write name="listaBeans"
property="vl_acrescimos"/>
                              <td width="15%" align="right">Descontos:
                              <td width="35%"><bean:write name="listaBeans" property="vl_descontos"/>
                            <tr>
                              <td width="15%" align="right">Vl a Pagar:
                              <td width="35%"><bean:write name="listaBeans" property="vl_pagar"/>
                              <td width="15%" align="right">Vl Pago:
                              <td width="35%"><bean:write name="listaBeans" property="vl_pago"/>
                            <tr>
                              <td width="15%" align="right">Tipo:
                              <td width="35%"><bean:write name="listaBeans" property="cd_tp_pagar"/>
                              <td width="15%" align="right">Conta:
                              <td width="35%"><bean:write name="listaBeans" property="cd_conta"/>
                        </logic:iterate>
                        </table>
                </td>
        </tr>
</table>

<html:hidden name="contas_pagarForm" property="contas_pagar_pesq.cd_emp"/>
<html:hidden name="contas_pagarForm" property="contas_pagar_pesq.cd_pagar"/>
<html:hidden name="contas_pagarForm" property="contas_pagar_pesq.dt_vencto"/>
<html:hidden name="contas_pagarForm" property="contas_pagar_pesq.dt_pagto"/>
<html:hidden name="contas_pagarForm" property="contas_pagar_pesq.vl_acrescimos"/>
<html:hidden name="contas_pagarForm" property="contas_pagar_pesq.vl_descontos"/>
<html:hidden name="contas_pagarForm" property="contas_pagar_pesq.vl_pagar"/>
<html:hidden name="contas_pagarForm" property="contas_pagar_pesq.vl_pago"/>
<html:hidden name="contas_pagarForm" property="contas_pagar_pesq.cd_tp_pagar"/>
<html:hidden name="contas_pagarForm" property="contas_pagar_pesq.cd_conta"/>

<jsp:include page="contadores.jsp">
```

```
        <jsp:param name="p_nm_form" value="contas_pagarForm" />
</jsp:include>

<html:messages id="msg" message="true">
        <bean:write name="msg"/><br>
</html:messages>

</html:form>


</body>
</html>
```

lista_contas_receber.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<html>
<head>
<title>Contas_receber</title>
<script>
        function confirma(){
                if (confirm("Deseja areceber este registro?") == true) {
                        return true;
                } else {
                        return false;
                }
        }
</script>
</head>

<body bgcolor="#ffffff">
<html:form action="/contas_receber">

<input type="hidden" name="oid">
<input type="hidden" name="p_pagina">

<jsp:include page="barra.jsp">
        <jsp:param name="p_nm_form" value="contas_receberForm" />
</jsp:include>

<table width="75%" border="1" bgcolor="#c8c8c8" bordercolor="#ffffff" align="center">
        <tr>
                <td>
                        <table width="100%" border="0" bgcolor="ffffff" bordercolor="#000000">
                        <logic:iterate id="listaBeans" name="contas_receberForm"
property="contas_receber_lista">
                          <tr>
                                <td width="15%" align="right">Empresa:
                                <td width="35%"><bean:write name="listaBeans" property="cd_emp"/>
                                          Nr.  <bean:write name="listaBeans"
property="cd_receber"/>
                            <td width="15%" align="right">
                            <%java.util.HashMap params;%>
                            <td width="35%">
                                        <bean:define id="new_oid" name="listaBeans"
property="oid"/>
<%
                                    params = new java.util.HashMap();
                                        params.put("evento","Areceber");
                                        params.put("oid",new_oid);
                                        params.put("p_pagina",request.getParameter("p_pagina"));
                                        request.setAttribute("parametros", params);
%>                                      <html:link page="/contas_receber.do" name="parametros"
onclick="return confirma()">
                                                <bean:message key="nota_fluxo.apagar"/>
                                        </html:link>
                                             
                                        <bean:define id="new_oid" name="listaBeans"
property="oid"/>
<%
                                    params = new java.util.HashMap();
                                        params.put("evento","Entra Alterar");
                                        params.put("oid",new_oid);
                                        params.put("p_pagina",request.getParameter("p_pagina"));
                                        request.setAttribute("parametros", params);
%>                                      <html:link page="/contas_receber.do" name="parametros" >
                                                <bean:message key="nota_fluxo.entra_alterar"/>
                                        </html:link>

                            <tr>
                              <td width="15%" align="right">Dt Vencimento:
                              <td width="35%"><bean:write name="listaBeans" property="dt_vencto"/>
                              <td width="15%" align="right">Dt Pagamento:
                              <td width="35%"><bean:write name="listaBeans" property="dt_pagto"/>
                            <tr>
                              <td width="15%" align="right">Vl a Receber:
```

```
                              <td width="35%"><bean:write name="listaBeans" property="vl_receber"/>
                              <td width="15%" align="right">Vl Recebido:
                              <td width="35%"><bean:write name="listaBeans" property="vl_recebido"/>
                            <tr>
                              <td width="15%" align="right">Tipo:
                              <td width="35%"><bean:write name="listaBeans"
property="cd_tp_receber"/>
                              <td width="15%" align="right">Conta:
                              <td width="35%"><bean:write name="listaBeans" property="cd_conta"/>
                            <tr>
                              <td width="15%" align="right">Nota:
                              <td width="35%"><bean:write name="listaBeans" property="nr_nota"/>
                              <td width="15%" align="right">
                              <td width="35%">
                          </logic:iterate>
                          </table>
                </td>
        </tr>
</table>

<html:hidden name="contas_receberForm" property="contas_receber_pesq.cd_emp"/>
<html:hidden name="contas_receberForm" property="contas_receber_pesq.cd_receber"/>
<html:hidden name="contas_receberForm" property="contas_receber_pesq.dt_vencto"/>
<html:hidden name="contas_receberForm" property="contas_receber_pesq.dt_pagto"/>
<html:hidden name="contas_receberForm" property="contas_receber_pesq.vl_receber"/>
<html:hidden name="contas_receberForm" property="contas_receber_pesq.vl_recebido"/>
<html:hidden name="contas_receberForm" property="contas_receber_pesq.cd_tp_receber"/>
<html:hidden name="contas_receberForm" property="contas_receber_pesq.cd_conta"/>
<html:hidden name="contas_receberForm" property="contas_receber_pesq.nr_nota"/>

<jsp:include page="contadores.jsp">
        <jsp:param name="p_nm_form" value="contas_receberForm" />
</jsp:include>

<html:messages id="msg" message="true">
        <bean:write name="msg"/><br>
</html:messages>

</html:form>


</body>
</html>
```

lista_contas.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<html>
<head>
<title>Contas</title>
<script>
        function confirma(){
                if (confirm("Deseja apagar este registro?") == true) {
                        return true;
                } else {
                        return false;
                }
        }
</script>
</head>

<body bgcolor="#ffffff">
<html:form action="/contas">

<input type="hidden" name="oid">
<input type="hidden" name="p_pagina">

<jsp:include page="barra.jsp">
        <jsp:param name="p_nm_form" value="contasForm" />
</jsp:include>

<table width="75%" border="1" bgcolor="#c8c8c8" bordercolor="#ffffff" align="center">
        <tr>
                <td>
                        <table width="100%" border="0" bgcolor="ffffff" bordercolor="#000000">
                          <tr>
                            <td align="right">Emp
                            <td align="right">Código
                            <td class="TxtGeral">Descrição
                        <logic:iterate id="listaBeans" name="contasForm" property="contas_lista">
                          <tr>
                            <td width="10%" align="right"><bean:write name="listaBeans"
property="cd_emp"/>
                            <td width="10%" align="right"><bean:write name="listaBeans"
property="cd_conta"/>
                            <td width="40%"><bean:write name="listaBeans" property="ds_conta"/>
```

```
                                <%java.util.HashMap params;%>
                                <td width="40%">
                                        <bean:define id="new_oid" name="listaBeans"
property="oid"/>
<%
                                params = new java.util.HashMap();
                                        params.put("evento","Apagar");
                                        params.put("oid",new_oid);
                                        params.put("p_pagina",request.getParameter("p_pagina"));
                                        request.setAttribute("parametros", params);
%>                                      <html:link page="/contas.do" name="parametros"
onclick="return confirma()">
                                                <bean:message key="nota_fluxo.apagar"/>
                                        </html:link>
                                             
                                        <bean:define id="new_oid" name="listaBeans"
property="oid"/>
<%
                                params = new java.util.HashMap();
                                        params.put("evento","Entra Alterar");
                                        params.put("oid",new_oid);
                                        params.put("p_pagina",request.getParameter("p_pagina"));
                                        request.setAttribute("parametros", params);
%>                                      <html:link page="/contas.do" name="parametros" >
                                                <bean:message key="nota_fluxo.entra_alterar"/>
                                        </html:link>

                        </logic:iterate>
                        </table>
                </td>
        </tr>
</table>

<html:hidden name="contasForm" property="contas_pesq.cd_emp"/>
<html:hidden name="contasForm" property="contas_pesq.cd_conta"/>
<html:hidden name="contasForm" property="contas_pesq.ds_conta"/>

<jsp:include page="contadores.jsp">
        <jsp:param name="p_nm_form" value="contasForm" />
</jsp:include>

<html:messages id="msg" message="true">
        <bean:write name="msg"/><br>
</html:messages>

</html:form>


</body>
</html>
```

lista_empresa.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<html>
<head>
<title>Empresa</title>
<script>
        function confirma(){
                if (confirm("Deseja apagar este registro?") == true) {
                        return true;
                } else {
                        return false;
                }
        }
</script>
</head>

<body bgcolor="#ffffff">
<html:form action="/empresa">

<input type="hidden" name="oid">
<input type="hidden" name="p_pagina">

<jsp:include page="barra.jsp">
        <jsp:param name="p_nm_form" value="empresaForm" />
</jsp:include>

<table width="75%" border="1" bgcolor="#c8c8c8" bordercolor="#ffffff" align="center">
        <tr>
                <td>
                        <table width="100%" border="0" bgcolor="ffffff" bordercolor="#000000">
                        <logic:iterate id="listaBeans" name="empresaForm" property="empresa_lista">
                          <tr>
                                <td width="15%" align="right">Empresa:
                                <td width="35%"><bean:write name="listaBeans" property="cd_emp"/>
```

```
                                           <bean:write name="listaBeans"
property="nm_emp"/>
                                <td width="15%" align="right">
                                <%java.util.HashMap params;%>
                                <td width="35%">
                                        <bean:define id="new_oid" name="listaBeans"
property="oid"/>
<%
                                    params = new java.util.HashMap();
                                        params.put("evento","Apagar");
                                        params.put("oid",new_oid);
                                        params.put("p_pagina",request.getParameter("p_pagina"));
                                        request.setAttribute("parametros", params);
%>                                      <html:link page="/empresa.do" name="parametros"
onclick="return confirma()">
                                                <bean:message key="nota_fluxo.apagar"/>
                                        </html:link>
                                             
                                        <bean:define id="new_oid" name="listaBeans"
property="oid"/>
<%
                                    params = new java.util.HashMap();
                                        params.put("evento","Entra Alterar");
                                        params.put("oid",new_oid);
                                        params.put("p_pagina",request.getParameter("p_pagina"));
                                        request.setAttribute("parametros", params);
%>                                      <html:link page="/empresa.do" name="parametros" >
                                                <bean:message key="nota_fluxo.entra_alterar"/>
                                        </html:link>

                            <tr>
                              <td width="15%" align="right">Razão Social:
                              <td width="35%"><bean:write name="listaBeans" property="razao_social"/>
                              <td width="15%" align="right">Cidade:
                              <td width="35%"><bean:write name="listaBeans" property="cidade"/>
                                      UF: <bean:write name="listaBeans" property="uf"/>
                            <tr>
                              <td width="15%" align="right">Endereço:
                              <td width="35%"><bean:write name="listaBeans" property="endereco"/>
                              <td width="15%" align="right">Bairro:
                              <td width="35%"><bean:write name="listaBeans" property="bairro"/>
                            <tr>
                              <td width="15%" align="right">CEP:
                              <td width="35%"><bean:write name="listaBeans" property="cep"/>
                              <td width="15%" align="right">Atividade:
                              <td width="35%"><bean:write name="listaBeans" property="cd_atividade"/>
                            <tr>
                              <td width="15%" align="right">CNPJ:
                              <td width="35%"><bean:write name="listaBeans" property="cnpj"/>
                              <td width="15%" align="right">Insc Estadual:
                              <td width="35%"><bean:write name="listaBeans"
property="insc_estadual"/>
                            <tr>
                              <td width="15%" align="right">Telefone:
                              <td width="35%"><bean:write name="listaBeans" property="telefone"/>
                              <td width="15%" align="right">Fax:
                              <td width="35%"><bean:write name="listaBeans" property="fax"/>
                            <tr>
                              <td width="15%" align="right">Contato:
                              <td width="35%"><bean:write name="listaBeans" property="nm_contato"/>
                              <td width="15%" align="right">Email:
                              <td width="35%"><bean:write name="listaBeans" property="email"/>
                        </logic:iterate>
                        </table>
                </td>
        </tr>
</table>

<html:hidden name="empresaForm" property="empresa_pesq.cd_emp"/>
<html:hidden name="empresaForm" property="empresa_pesq.nm_emp"/>
<html:hidden name="empresaForm" property="empresa_pesq.razao_social"/>
<html:hidden name="empresaForm" property="empresa_pesq.endereco"/>
<html:hidden name="empresaForm" property="empresa_pesq.bairro"/>
<html:hidden name="empresaForm" property="empresa_pesq.cidade"/>
<html:hidden name="empresaForm" property="empresa_pesq.uf"/>
<html:hidden name="empresaForm" property="empresa_pesq.cep"/>
<html:hidden name="empresaForm" property="empresa_pesq.cnpj"/>
<html:hidden name="empresaForm" property="empresa_pesq.insc_estadual"/>
<html:hidden name="empresaForm" property="empresa_pesq.telefone"/>
<html:hidden name="empresaForm" property="empresa_pesq.fax"/>
<html:hidden name="empresaForm" property="empresa_pesq.email"/>
<html:hidden name="empresaForm" property="empresa_pesq.nm_contato"/>
<html:hidden name="empresaForm" property="empresa_pesq.cd_atividade"/>

<jsp:include page="contadores.jsp">
        <jsp:param name="p_nm_form" value="empresaForm" />
</jsp:include>
```

```
<html:messages id="msg" message="true">
        <bean:write name="msg"/><br>
</html:messages>

</html:form>


</body>
</html>
```

lista_movimento_contas.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<html>
<head>
<title>Movimento_contas</title>
<script>
        function confirma(){
                if (confirm("Deseja apagar este registro?") == true) {
                        return true;
                } else {
                        return false;
                }
        }
</script>
</head>

<body bgcolor="#ffffff">
<html:form action="/movimento_contas">

<input type="hidden" name="oid">
<input type="hidden" name="p_pagina">

<jsp:include page="barra.jsp">
        <jsp:param name="p_nm_form" value="movimento_contasForm" />
</jsp:include>

<table width="75%" border="1" bgcolor="#c8c8c8" bordercolor="#ffffff" align="center">
        <tr>
                <td>
                        <table width="100%" border="0" bgcolor="ffffff" bordercolor="#000000">
                          <tr>
                            <td>Data
                            <td align="right">Nr
                            <td align="right">Emp
                            <td align="right">Conta
                            <td>Descrição
                            <td align="right">Valor
                        <logic:iterate id="listaBeans" name="movimento_contasForm"
property="movimento_contas_lista">
                          <tr>
                            <td width="10%" align="right"><bean:write name="listaBeans"
property="dt_movto"/>
                            <td width="5%" align="right"><bean:write name="listaBeans"
property="nr_movto"/>
                            <td width="5%" align="right"><bean:write name="listaBeans"
property="cd_emp"/>
                            <td width="10%" align="right"><bean:write name="listaBeans"
property="nr_conta"/>
                            <td width="30%"><bean:write name="listaBeans" property="ds_movto"/>
                            <td width="10%" align="right"><bean:write name="listaBeans"
property="vl_movto"/>
                            <%java.util.HashMap params;%>
                            <td width="30%">
                                        <bean:define id="new_oid" name="listaBeans"
property="oid"/>
<%
                                    params = new java.util.HashMap();
                                        params.put("evento","Apagar");
                                        params.put("oid",new_oid);
                                        params.put("p_pagina",request.getParameter("p_pagina"));
                                        request.setAttribute("parametros", params);
%>                                      <html:link page="/movimento_contas.do" name="parametros"
onclick="return confirma()">
                                                <bean:message key="nota_fluxo.apagar"/>
                                        </html:link>
                                             
                                        <bean:define id="new_oid" name="listaBeans"
property="oid"/>
<%
                                    params = new java.util.HashMap();
                                        params.put("evento","Entra Alterar");
                                        params.put("oid",new_oid);
                                        params.put("p_pagina",request.getParameter("p_pagina"));
                                        request.setAttribute("parametros", params);
%>                                      <html:link page="/movimento_contas.do" name="parametros" >
```

```
                                                <bean:message key="nota_fluxo.entra_alterar"/>
                                        </html:link>

                                </logic:iterate>
                                </table>
                        </td>
                </tr>
</table>

<html:hidden name="movimento_contasForm" property="movimento_contas_pesq.dt_movto" />
<html:hidden name="movimento_contasForm" property="movimento_contas_pesq.nr_movto" />
<html:hidden name="movimento_contasForm" property="movimento_contas_pesq.cd_emp" />
<html:hidden name="movimento_contasForm" property="movimento_contas_pesq.nr_conta" />
<html:hidden name="movimento_contasForm" property="movimento_contas_pesq.ds_movto" />
<html:hidden name="movimento_contasForm" property="movimento_contas_pesq.vl_movto" />


<jsp:include page="contadores.jsp">
        <jsp:param name="p_nm_form" value="movimento_contasForm" />
</jsp:include>

<html:messages id="msg" message="true">
        <bean:write name="msg"/><br>
</html:messages>

</html:form>


</body>
</html>
```

lista_nota_fiscal.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<html>
<head>
<title>Nota_Fiscal</title>
<script>
        function confirma(){
                if (confirm("Deseja apagar este registro?") == true) {
                        return true;
                } else {
                        return false;
                }
        }
</script>
</head>

<body bgcolor="#ffffff">
<html:form action="/nota_fiscal">

<input type="hidden" name="oid">
<input type="hidden" name="p_pagina">

<jsp:include page="barra.jsp">
        <jsp:param name="p_nm_form" value="nota_fiscalForm" />
</jsp:include>

<logic:iterate id="listaBeans" name="nota_fiscalForm" property="nota_fiscal_lista">
<table width="75%" border="1" bgcolor="#c8c8c8" bordercolor="#ffffff" align="center">
        <tr>
                <td>
                        <table width="100%" border="0" bgcolor="ffffff" bordercolor="#000000">
                          <tr>
                                <td width="15%" align="right">Empresa:
                                <td width="35%"><bean:write name="listaBeans" property="cd_emp"/>
                                                                        Nr: <bean:write name="listaBeans"
property="nr_nota"/>
                             <td width="15%" align="right">
                             <%java.util.HashMap params;%>
                             <td width="35%">
                                        <bean:define id="new_oid" name="listaBeans"
property="oid"/>
<%
                                params = new java.util.HashMap();
                                        params.put("evento","Apagar");
                                        params.put("oid",new_oid);
                                        params.put("p_pagina",request.getParameter("p_pagina"));
                                        request.setAttribute("parametros", params);
%>                                      <html:link page="/nota_fiscal.do" name="parametros"
onclick="return confirma()">
                                                <bean:message key="nota_fluxo.apagar"/>
                                        </html:link>
                                             
                                        <bean:define id="new_oid" name="listaBeans"
property="oid"/>
```

```
<%
                                        params = new java.util.HashMap();
                                             params.put("evento","Entra Alterar");
                                             params.put("oid",new_oid);
                                             params.put("p_pagina",request.getParameter("p_pagina"));
                                             request.setAttribute("parametros", params);
%>                                           <html:link page="/nota_fiscal.do" name="parametros" >
                                                    <bean:message key="nota_fluxo.entra_alterar"/>
                                             </html:link>
                                                  
                                             <bean:define id="new_oid" name="listaBeans"
property="oid"/>
<%
                                        params = new java.util.HashMap();
                                             params.put("evento","Confirmar");
                                             params.put("oid",new_oid);
                                             params.put("p_pagina",request.getParameter("p_pagina"));
                                             request.setAttribute("parametros", params);
%>                                           <html:link page="/nota_fiscal.do" name="parametros" >
                                                    <bean:message key="nota_fluxo.confirmar"/>
                                             </html:link>
                              <tr>
                                <td width="15%" align="right">Data:
                                <td width="35%"><bean:write name="listaBeans" property="dt_ref"/>
                                <td width="15%" align="right">Cliente:
                                <td width="35%"><bean:write name="listaBeans" property="cd_cli"/>
                              <tr>
                                <td width="15%" align="right">Dt Vencimento:
                                <td width="35%"><bean:write name="listaBeans" property="dt_vencto"/>
                                <td width="15%" align="right">Vl Bruto:
                                <td width="35%"><bean:write name="listaBeans" property="vl_bruto"/>
                              <tr>
                                <td width="15%" align="right">Vl Liquido:
                                <td width="35%"><bean:write name="listaBeans" property="vl_liquido"/>
                                <td width="15%" align="right">Conta:
                                <td width="35%"><bean:write name="listaBeans" property="cd_conta"/>

                            </table>
                    </td>
            </tr>
</table>

<jsp:include page="contadores.jsp">
            <jsp:param name="p_nm_form" value="nota_fiscalForm" />
</jsp:include>

<IFRAME align="center" height="70%" frame name="nota_item"
src="nota_item.do?nota_item_pesq.cd_emp=<bean:write name="listaBeans"
property="cd_emp"/>&nota_item.cd_emp=<bean:write name="listaBeans"
property="cd_emp"/>&nota_item_pesq.nr_nota=<bean:write name="listaBeans"
property="nr_nota"/>&nota_item.nr_nota=<bean:write name="listaBeans" property="nr_nota"/>"
width="100%" scrolling="auto" frameborder="no"> </IFRAME>
</logic:iterate>

<html:hidden name="nota_fiscalForm" property="nota_fiscal_pesq.cd_emp"/>
<html:hidden name="nota_fiscalForm" property="nota_fiscal_pesq.nr_nota"/>
<html:hidden name="nota_fiscalForm" property="nota_fiscal_pesq.dt_ref"/>
<html:hidden name="nota_fiscalForm" property="nota_fiscal_pesq.cd_cli"/>
<html:hidden name="nota_fiscalForm" property="nota_fiscal_pesq.vl_bruto"/>
<html:hidden name="nota_fiscalForm" property="nota_fiscal_pesq.dt_vencto"/>
<html:hidden name="nota_fiscalForm" property="nota_fiscal_pesq.vl_liquido"/>
<html:hidden name="nota_fiscalForm" property="nota_fiscal_pesq.cd_conta"/>

<html:messages id="msg" message="true">
            <bean:write name="msg"/><br>
</html:messages>

</html:form>

</body>
</html>
```

lista_nota_item.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<html>
<head>
<title>Nota_item</title>
<script>
            function confirma(){
                        if (confirm("Deseja apagar este registro?") == true) {
                                    return true;
                        } else {
                                    return false;
                        }
            }
```

```
</script>
</head>

<body bgcolor="#ffffff">
<html:form action="/nota_item">

<input type="hidden" name="oid">
<input type="hidden" name="p_pagina">

<jsp:include page="barra.jsp">
        <jsp:param name="p_nm_form" value="nota_itemForm" />
</jsp:include>


<table width="75%" border="1" bgcolor="#c8c8c8" bordercolor="#ffffff" align="center">
        <tr>
                <td>
                        <table width="100%" border="0" bgcolor="ffffff" bordercolor="#000000">
                          <tr>
                            <td align="right">Empresa
                            <td align="right">Nota
                            <td align="right">Cnae
                            <td align="right">Valor
                            <td align="right">
                                <td align="right">
                        <logic:iterate id="listaBeans" name="nota_itemForm"
property="nota_item_lista">
                          <tr>
                            <td width="15%" align="right"><bean:write name="listaBeans"
property="cd_emp"/>
                            <td width="15%" align="right"><bean:write name="listaBeans"
property="nr_nota"/>
                            <td width="15%" align="right"><bean:write name="listaBeans"
property="cd_cnae"/>
                            <td width="15%" align="right"><bean:write name="listaBeans"
property="vl_item"/>
                            <%java.util.HashMap params;%>
                            <td width="40%">
                                        <bean:define id="new_oid" name="listaBeans"
property="oid"/>
<%
                                    params = new java.util.HashMap();
                                        params.put("evento","Apagar");
                                        params.put("oid",new_oid);
                                        params.put("p_pagina",request.getParameter("p_pagina"));
                                        request.setAttribute("parametros", params);
%>                                      <html:link page="/nota_item.do" name="parametros"
onclick="return confirma()">
                                                <bean:message key="nota_fluxo.apagar"/>
                                        </html:link>
                                             
                                        <bean:define id="new_oid" name="listaBeans"
property="oid"/>
<%
                                    params = new java.util.HashMap();
                                        params.put("evento","Entra Alterar");
                                        params.put("oid",new_oid);
                                        params.put("p_pagina",request.getParameter("p_pagina"));
                                        request.setAttribute("parametros", params);
%>                                      <html:link page="/nota_item.do" name="parametros" >
                                                <bean:message key="nota_fluxo.entra_alterar"/>
                                        </html:link>
                        </logic:iterate>
                        </table>
                </td>
        </tr>
</table>

<html:hidden name="nota_itemForm" property="nota_item.cd_emp"/>
<html:hidden name="nota_itemForm" property="nota_item.nr_nota"/>

<html:hidden name="nota_itemForm" property="nota_item_pesq.cd_emp"/>
<html:hidden name="nota_itemForm" property="nota_item_pesq.nr_nota"/>
<html:hidden name="nota_itemForm" property="nota_item_pesq.cd_cnae"/>
<html:hidden name="nota_itemForm" property="nota_item_pesq.vl_item"/>

<jsp:include page="contadores.jsp">
        <jsp:param name="p_nm_form" value="nota_itemForm" />
</jsp:include>

<html:messages id="msg" message="true">
        <bean:write name="msg"/><br>
</html:messages>

</html:form>


</body>
```

```
</html>
```

lista_ramo_Atividade.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<html>
<head>
<title>Ramo_atividade</title>
<script>
        function confirma(){
                if (confirm("Deseja apagar este registro?") == true) {
                        return true;
                } else {
                        return false;
                }
        }
</script>
</head>

<body bgcolor="#ffffff">
<html:form action="/ramo_atividade">

<input type="hidden" name="oid">
<input type="hidden" name="p_pagina">

<jsp:include page="barra.jsp">
        <jsp:param name="p_nm_form" value="ramo_atividadeForm" />
</jsp:include>

<table width="75%" border="1" bgcolor="#c8c8c8" bordercolor="#ffffff" align="center">
        <tr>
                <td>
                        <table width="100%" border="0" bgcolor="ffffff" bordercolor="#000000">
                          <tr>
                            <td align="right">Código
                            <td class="TxtGeral">Descrição
                        <logic:iterate id="listaBeans" name="ramo_atividadeForm"
property="ramo_atividade_lista">
                          <tr>
                            <td width="10%" align="right"><bean:write name="listaBeans"
property="cd_atividade"/>
                            <td width="60%"><bean:write name="listaBeans" property="ds_atividade"/>
                            <%java.util.HashMap params;%>
                            <td width="35%">
                                        <bean:define id="new_oid" name="listaBeans"
property="oid"/>
<%
                                    params = new java.util.HashMap();
                                        params.put("evento","Apagar");
                                        params.put("oid",new_oid);
                                        params.put("p_pagina",request.getParameter("p_pagina"));
                                        request.setAttribute("parametros", params);
%>                                      <html:link page="/ramo_atividade.do" name="parametros"
onclick="return confirma()">
                                                <bean:message key="nota_fluxo.apagar"/>
                                        </html:link>
                                             
                                        <bean:define id="new_oid" name="listaBeans"
property="oid"/>
<%
                                    params = new java.util.HashMap();
                                        params.put("evento","Entra Alterar");
                                        params.put("oid",new_oid);
                                        params.put("p_pagina",request.getParameter("p_pagina"));
                                        request.setAttribute("parametros", params);
%>                                      <html:link page="/ramo_atividade.do" name="parametros" >
                                                <bean:message key="nota_fluxo.entra_alterar"/>
                                        </html:link>

                        </logic:iterate>
                        </table>
                </td>
        </tr>
</table>

<html:hidden name="ramo_atividadeForm" property="ramo_atividade_pesq.cd_atividade"/>
<html:hidden name="ramo_atividadeForm" property="ramo_atividade_pesq.ds_atividade"/>

<jsp:include page="contadores.jsp">
        <jsp:param name="p_nm_form" value="ramo_atividadeForm" />
</jsp:include>

<html:messages id="msg" message="true">
        <bean:write name="msg"/><br>
</html:messages>
```

```
</html:form>


</body>
</html>
```

lista_tipo_contas_pagar.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<html>
<head>
<title>Cnae</title>
<script>
        function confirma(){
                if (confirm("Deseja apagar este registro?") == true) {
                        return true;
                } else {
                        return false;
                }
        }
</script>
</head>

<body bgcolor="#ffffff">
<html:form action="/tipo_contas_pagar">

<input type="hidden" name="oid">
<input type="hidden" name="p_pagina">

<jsp:include page="barra.jsp">
        <jsp:param name="p_nm_form" value="tipo_contas_pagarForm" />
</jsp:include>

<table width="75%" border="1" bgcolor="#c8c8c8" bordercolor="#ffffff" align="center">
        <tr>
                <td>
                        <table width="100%" border="0" bgcolor="ffffff" bordercolor="#000000">
                          <tr>
                            <td align="right">Código
                            <td class="TxtGeral">Descrição
                        <logic:iterate id="listaBeans" name="tipo_contas_pagarForm"
property="tipo_contas_pagar_lista">
                          <tr>
                            <td width="10%" align="right"><bean:write name="listaBeans"
property="cd_tp_pagar"/>
                            <td width="60%"><bean:write name="listaBeans" property="ds_tp_pagar"/>
                            <td width="30%" align="right">
                            <%java.util.HashMap params;%>
                                        <bean:define id="new_oid" name="listaBeans"
property="oid"/>
<%
                                    params = new java.util.HashMap();
                                        params.put("evento","Apagar");
                                        params.put("oid",new_oid);
                                        params.put("p_pagina",request.getParameter("p_pagina"));
                                        request.setAttribute("parametros", params);
%>                                      <html:link page="/tipo_contas_pagar.do" name="parametros"
onclick="return confirma()">
                                                <bean:message key="nota_fluxo.apagar"/>
                                        </html:link>
                                             
                                        <bean:define id="new_oid" name="listaBeans"
property="oid"/>
<%
                                    params = new java.util.HashMap();
                                        params.put("evento","Entra Alterar");
                                        params.put("oid",new_oid);
                                        params.put("p_pagina",request.getParameter("p_pagina"));
                                        request.setAttribute("parametros", params);
%>                                      <html:link page="/tipo_contas_pagar.do" name="parametros" >
                                                <bean:message key="nota_fluxo.entra_alterar"/>
                                        </html:link>
                        </logic:iterate>
                        </table>
                </td>
        </tr>
</table>

<html:hidden name="tipo_contas_pagarForm" property="tipo_contas_pagar_pesq.cd_tp_pagar"/>
<html:hidden name="tipo_contas_pagarForm" property="tipo_contas_pagar_pesq.ds_tp_pagar"/>

<jsp:include page="contadores.jsp">
        <jsp:param name="p_nm_form" value="tipo_contas_pagarForm" />
</jsp:include>

<html:messages id="msg" message="true">
```

```
        <bean:write name="msg"/><br>
</html:messages>

</html:form>


</body>
</html>
```

lista_tipo_contas_receber.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<html>
<head>
<title>Cnae</title>
<script>
        function confirma(){
                if (confirm("Deseja apagar este registro?") == true) {
                        return true;
                } else {
                        return false;
                }
        }
</script>
</head>

<body bgcolor="#ffffff">
<html:form action="/tipo_contas_receber">

<input type="hidden" name="oid">
<input type="hidden" name="p_pagina">

<jsp:include page="barra.jsp">
        <jsp:param name="p_nm_form" value="tipo_contas_receberForm" />
</jsp:include>

<table width="75%" border="1" bgcolor="#c8c8c8" bordercolor="#ffffff" align="center">
        <tr>
                <td>
                        <table width="100%" border="0" bgcolor="ffffff" bordercolor="#000000">
                          <tr>
                            <td align="right">Código
                            <td class="TxtGeral">Descrição
                        <logic:iterate id="listaBeans" name="tipo_contas_receberForm"
property="tipo_contas_receber_lista">
                          <tr>
                            <td width="10%" align="right"><bean:write name="listaBeans"
property="cd_tp_receber"/>
                            <td width="60%"><bean:write name="listaBeans"
property="ds_tp_receber"/>
                            <td width="30%" align="right">
                            <%java.util.HashMap params;%>
                                        <bean:define id="new_oid" name="listaBeans"
property="oid"/>
<%
                                    params = new java.util.HashMap();
                                        params.put("evento","Apagar");
                                        params.put("oid",new_oid);
                                        params.put("p_pagina",request.getParameter("p_pagina"));
                                        request.setAttribute("parametros", params);
%>                                      <html:link page="/tipo_contas_receber.do" name="parametros"
onclick="return confirma()">
                                                <bean:message key="nota_fluxo.apagar"/>
                                        </html:link>
                                             
                                        <bean:define id="new_oid" name="listaBeans"
property="oid"/>
<%
                                    params = new java.util.HashMap();
                                        params.put("evento","Entra Alterar");
                                        params.put("oid",new_oid);
                                        params.put("p_pagina",request.getParameter("p_pagina"));
                                        request.setAttribute("parametros", params);
%>                                      <html:link page="/tipo_contas_receber.do" name="parametros"
>
                                                <bean:message key="nota_fluxo.entra_alterar"/>
                                        </html:link>
                        </logic:iterate>
                        </table>
                </td>
        </tr>
</table>

<html:hidden name="tipo_contas_receberForm" property="tipo_contas_receber_pesq.cd_tp_receber"/>
<html:hidden name="tipo_contas_receberForm" property="tipo_contas_receber_pesq.ds_tp_receber"/>
```

```
<jsp:include page="contadores.jsp">
        <jsp:param name="p_nm_form" value="tipo_contas_receberForm" />
</jsp:include>

<html:messages id="msg" message="true">
        <bean:write name="msg"/><br>
</html:messages>

</html:form>


</body>
</html>
```

barra.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<%String p_nm_form = request.getParameter("p_nm_form");%>

<script>
function submete(n_pagina){
        <%=p_nm_form%>.p_pagina.value = n_pagina;
    <%=p_nm_form%>.submit();
}
</script>

<table width="75%" align="center" border="0">
<tr>
        <td align="center">
                <logic:equal name="<%=p_nm_form%>" property="mostraAnteriores" value="N">
                 <img src="imagens/previous.gif" border="0"></a>
                 <img src="imagens/previous.gif" border="0"></a>
                </logic:equal>

                <logic:equal name="<%=p_nm_form%>" property="mostraAnteriores" value="S">
                        <a href="javascript:submete('<bean:write name="<%=p_nm_form%>"
property="pagina_prim"/>')"><img src="imagens/previous.gif" border="0"></a>
                        <a href="javascript:submete('<bean:write name="<%=p_nm_form%>"
property="pagina_ant"/>')"><img src="imagens/previous.gif" border="0"></a>
                </logic:equal>

                <logic:equal name="<%=p_nm_form%>" property="mostraProximos" value="N">
                        <img src="imagens/next.gif" border="0"></a>
                        <img src="imagens/next.gif" border="0"></a>
                </logic:equal>

                <logic:equal name="<%=p_nm_form%>" property="mostraProximos" value="S">
                        <a href="javascript:submete('<bean:write name="<%=p_nm_form%>"
property="pagina_prox"/>')"><img src="imagens/next.gif" border="0"></a>
                        <a href="javascript:submete('<bean:write name="<%=p_nm_form%>"
property="pagina_ult"/>')"><img src="imagens/next.gif" border="0"></a>
                </logic:equal>

        </td>
</tr>
<tr>
        <td align="center">
                <html:submit property="evento">
                        <bean:message key="nota_fluxo.entra_inserir"/>
                </html:submit>
                <html:submit property="evento">
                        <bean:message key="nota_fluxo.entra_consultar"/>
                </html:submit>
        </td>
</tr>

</table>
```

cabeçalho.jsp

```
<%@ page language="java" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>

<logic:notPresent name="usuarioForm">
        <% request.getSession().invalidate();
        // invalida sessão
        // para não permitir que usuários que cadastrem em outra sessão
        // permitam que outros entrem direto pelo mainmenu.jsp
        // permite q um usuário visualize coisas do outro se entrar depois na mesma sessão
        // criar atributo de sessão que coloque o usuário que está logado
        %>
        <logic:redirect href="form/index.jsp"/>
</logic:notPresent>
```

contadores.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<%String p_nm_form = request.getParameter("p_nm_form");%>
<table align="center">
        <tr><td class="TxtGeral"><bean:write name="<%=p_nm_form%>" property="contadorAtual"/> 
de  
                        <bean:write name="<%=p_nm_form%>" property="contador"/>  linhas
selecionadas
                </td>
</table>
```

index.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<html>
<head>
        <title>Login</title>
</head>
<body>
        <html:form action="/login">
        <table width="50%" border="1" bgcolor="#c8c8c8" bordercolor="#ffffff" align="center">
                <tr>
                        <td>
                                <table width="100%" border="0" bgcolor="ffffff"
bordercolor="#000000">
                                        <tr>
                                                <td class="TxtGeral" width="33%" align="right">
                                                        Usuário:

                                                <td class="TxtGeral" width="67%">
                                                        <html:text maxlength="30" style="text-
align: left" name="usuarioForm" property="usuario.nm_usuario"/>
                                        <tr>
                                                <td class="TxtGeral" width="33%" align="right">
                                                        Senha:
                                                <td class="TxtGeral" width="67%">
                                                        <html:password maxlength="30" style="text-
align: left" name="usuarioForm" property="usuario.ds_senha"/>
                                        <tr>
                                                <td class="TxtGeral" width="33%" align="right">
                                                <td class="TxtGeral" width="67%">
                                                <html:submit property="evento">
                                                        <bean:message
key="nota_fluxo.login.valida"/>
                                                </html:submit>
                                </table>
        </table>
        <html:messages id="msg" message="true">
                <bean:write name="msg"/><br>
        </html:messages>
        </html:form>
</body>
</html>
```

principal.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<html>
<head>
        <title>Principal</title>
</head>
<body>
        <table width="100%" border="0" bgcolor="ffffff" bordercolor="#000000">
                <tr>
                        <td width="20%" valign="top">
                                <table width="100%" border="0" bgcolor="ffffff"
bordercolor="#000000">
                                        <tr>
                                                <td>
                                                        <html:link page="/empresa.do"
target="principal">
                                                                <bean:message
key="nota_fluxo.empresa"/>
                                                        </html:link>
                                        <tr>
                                                <td>
                                                        <html:link page="/cnae.do"
target="principal">
                                                                <bean:message
key="nota_fluxo.cnae"/>
```

```
                                        </html:link>
                                    <tr>
                                        <td>
                                        <html:link page="/cliente.do"
target="principal">
                                            <bean:message
key="nota_fluxo.cliente"/>
                                        </html:link>
                                    <tr>
                                        <td>
                                        <html:link page="/contas_pagar.do"
target="principal">
                                            <bean:message
key="nota_fluxo.contas_pagar"/>
                                        </html:link>
                                    <tr>
                                        <td>
                                        <html:link page="/contas_receber.do"
target="principal">
                                            <bean:message
key="nota_fluxo.contas_receber"/>
                                        </html:link>
                                    <tr>
                                        <td>
                                        <html:link page="/contas.do"
target="principal">
                                            <bean:message
key="nota_fluxo.contas"/>
                                        </html:link>
                                    <tr>
                                        <td>
                                        <html:link page="/movimento_contas.do"
target="principal">
                                            <bean:message
key="nota_fluxo.movimento_contas"/>
                                        </html:link>
                                    <tr>
                                        <td>
                                        <html:link page="/nota_fiscal.do"
target="principal">
                                            <bean:message
key="nota_fluxo.nota_fiscal"/>
                                        </html:link>
                                    <tr>
                                        <td>
                                        <html:link page="/nota_item.do"
target="principal">
                                            <bean:message
key="nota_fluxo.nota_item"/>
                                        </html:link>
                                    <tr>
                                        <td>
                                        <html:link page="/ramo_atividade.do"
target="principal">
                                            <bean:message
key="nota_fluxo.ramo_atividade"/>
                                        </html:link>
                                    <tr>
                                        <td>
                                        <html:link page="/tipo_contas_pagar.do"
target="principal">
                                            <bean:message
key="nota_fluxo.tipo_contas_pagar"/>
                                        </html:link>
                                    <tr>
                                        <td>
                                        <html:link page="/tipo_contas_receber.do"
target="principal">
                                            <bean:message
key="nota_fluxo.tipo_contas_receber"/>
                                        </html:link>
                                </table>
                            <td width="80%">
                                <IFRAME align="center" height="400" frame name="principal" src=""
                                width="100%" scrolling="auto" frameborder="no"> </IFRAME>
            </table>
    </body>
    </html>
```

ApplicationResource.properties

```
# Resources for parameter 'br.com.nota_fluxo.ApplicationResources'
# Project P/nota_fluxo

errors.required=O campo {0} deve ser preenchido.
errors.minlength={0} não pode ter menos de {1} caracteres.
errors.maxlength={0} não pode ter mais de {1} caracteres.
errors.invalid={0} é inválido.
```

```
errors.byte={0} deve ser do tipo byte.
errors.short={0} deve ser do tipo short.
errors.integer={0} deve ser um numero inteiro.
errors.long={0} deve ser do tipo long.
errors.float={0} deve ser do tipo float.
errors.double={0} deve ser do tipo double.

errors.date={0} não é uma data.
errors.range={0} não está no limite de {1} até {2}.
errors.creditcard={0} é um número inválido de cartão de crédito.
errors.email={0} é um endereço de email inválido.

nota_fluxo.entra_inserir=Entra Novo
nota_fluxo.entra_consultar=Entra Consultar
nota_fluxo.entra_alterar=Entra Alterar
nota_fluxo.inserir=Novo
nota_fluxo.apagar=Apagar
nota_fluxo.alterar=Alterar
nota_fluxo.lista=Lista
#nota_fluxo.inicio=Inicio

nota_fluxo.inserir.erro=Não é permitido inserir um registro Duplicado
nota_fluxo.alterar.erro=O registro foi excluido por outro usuário

empresaForm.cd_emp.displayname=Código
error.sql= {0}

nota_fluxo.login.valida=Entrar
nota_fluxo.login.erro=Senha inválida
nota_fluxo.login.preencha=Os campos devem ser preenchidos

nota_fluxo.empresa=Empresa
nota_fluxo.cliente=Cliente
nota_fluxo.cnae=Cnae
nota_fluxo.contas_pagar=Contas pagar
nota_fluxo.contas_receber=Contas receber
nota_fluxo.contas=Contas
nota_fluxo.movimento_contas=Movimento contas
nota_fluxo.nota_fiscal=Nota fiscal
nota_fluxo.nota_item=Nota Item
nota_fluxo.ramo_atividade=Ramo Atividade
nota_fluxo.tipo_contas_pagar=Tipo Contas Pagar
nota_fluxo.tipo_contas_receber=Tipo Contas Receber
nota_fluxo.usuario=Usuário

nota_fluxo.confirmar=Confirmar
```

web.xml

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

<!DOCTYPE web-app
  PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"
  "http://java.sun.com/j2ee/dtds/web-app_2_2.dtd">

<web-app>
        <display-name>Notas Fiscais e Fluxo de Caixa</display-name>
    <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>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>1</load-on-startup>
    </servlet>

    <servlet-mapping>
        <servlet-name>action</servlet-name>
        <url-pattern>*.do</url-pattern>
    </servlet-mapping>

        <!-- Struts Tag Library Descriptors -->
        <taglib>
                <taglib-uri>/WEB-INF/struts-bean.tld</taglib-uri>
                <taglib-location>/WEB-INF/struts-bean.tld</taglib-location>
        </taglib>

        <taglib>
```

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

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

 <welcome-file-list>
    <welcome-file>form/index.jsp</welcome-file>
  </welcome-file-list>

    <resource-ref>
                <description>Postgres Datasource Teste</description>
                <res-ref-name>jdbc/dspostgres</res-ref-name>
                <res-type>javax.sql.DataSource</res-type>
                <res-auth>Container</res-auth>
        </resource-ref>

</web-app>
```

struts-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts-config PUBLIC "-//Apache Software Foundation//DTD Struts Configuration 1.1//EN"
"http://jakarta.apache.org/struts/dtds/struts-config_1_1.dtd">
<struts-config>

    <!-- ========== Data Source Configuration =============================== -->
    <data-sources />

    <!-- ========== Form Bean Definitions =================================== -->
    <form-beans>
        <form-bean name="empresaForm"
         type="br.com.nota_fluxo.form.EmpresaForm"/>
        <form-bean name="usuarioForm"
         type="br.com.nota_fluxo.form.UsuarioForm" />
        <form-bean name="clienteForm"
         type="br.com.nota_fluxo.form.ClienteForm" />
        <form-bean name="cnaeForm"
         type="br.com.nota_fluxo.form.CnaeForm" />
        <form-bean name="contas_pagarForm"
         type="br.com.nota_fluxo.form.Contas_pagarForm" />
        <form-bean name="contas_receberForm"
         type="br.com.nota_fluxo.form.Contas_receberForm" />
        <form-bean name="contasForm"
         type="br.com.nota_fluxo.form.ContasForm" />
        <form-bean name="movimento_contasForm"
         type="br.com.nota_fluxo.form.Movimento_contasForm" />
        <form-bean name="nota_fiscalForm"
         type="br.com.nota_fluxo.form.Nota_fiscalForm" />
        <form-bean name="nota_itemForm"
         type="br.com.nota_fluxo.form.Nota_itemForm" />
        <form-bean name="ramo_atividadeForm"
         type="br.com.nota_fluxo.form.Ramo_atividadeForm" />
        <form-bean name="tipo_contas_pagarForm"
         type="br.com.nota_fluxo.form.Tipo_contas_pagarForm" />
        <form-bean name="tipo_contas_receberForm"
         type="br.com.nota_fluxo.form.Tipo_contas_receberForm" />
    </form-beans>

    <!-- ========== Global Exception Definitions ============================ -->
    <global-exceptions>

    </global-exceptions>

    <!-- ========== Global Forward Definitions ============================== -->
    <global-forwards>

    </global-forwards>

    <!-- ========== Action Mapping Definitions ============================== -->
    <action-mappings>
        <action
            attribute="empresaForm"
            name="empresaForm"
            parameter="evento"
            path="/empresa"
            scope="request"
            type="br.com.nota_fluxo.action.EmpresaAction"
            validate="false">
            <forward name="lista" path="/form/lista_empresa.jsp" />
            <forward name="editar" path="/form/editar_empresa.jsp" />
            <forward name="success" path="/success.jsp" />
        </action>
        <action
            attribute="usuarioForm"
```

```
    name="usuarioForm"
    parameter="evento"
    path="/login"
    scope="session"
    type="br.com.nota_fluxo.action.LoginAction"
    validate="false">
    <forward name="lista" path="/form/principal.jsp" />
    <forward name="editar" path="/form/index.jsp" />
</action>
<action
    attribute="clienteForm"
    name="clienteForm"
    parameter="evento"
    path="/cliente"
    scope="request"
    type="br.com.nota_fluxo.action.ClienteAction"
    validate="false">
    <forward name="lista" path="/form/lista_cliente.jsp" />
    <forward name="editar" path="/form/editar_cliente.jsp" />
</action>
<action
    attribute="cnaeForm"
    name="cnaeForm"
    parameter="evento"
    path="/cnae"
    scope="request"
    type="br.com.nota_fluxo.action.CnaeAction"
    validate="false">
    <forward name="lista" path="/form/lista_cnae.jsp" />
    <forward name="editar" path="/form/editar_cnae.jsp" />
</action>
<action
    attribute="contas_pagarForm"
    name="contas_pagarForm"
    parameter="evento"
    path="/contas_pagar"
    scope="request"
    type="br.com.nota_fluxo.action.Contas_pagarAction"
    validate="false">
    <forward name="lista" path="/form/lista_contas_pagar.jsp" />
    <forward name="editar" path="/form/editar_contas_pagar.jsp" />
</action>
<action
    attribute="contas_receberForm"
    name="contas_receberForm"
    parameter="evento"
    path="/contas_receber"
    scope="request"
    type="br.com.nota_fluxo.action.Contas_receberAction"
    validate="false">
    <forward name="lista" path="/form/lista_contas_receber.jsp" />
    <forward name="editar" path="/form/editar_contas_receber.jsp" />
</action>
<action
    attribute="contasForm"
    name="contasForm"
    parameter="evento"
    path="/contas"
    scope="request"
    type="br.com.nota_fluxo.action.ContasAction"
    validate="false">
    <forward name="lista" path="/form/lista_contas.jsp" />
    <forward name="editar" path="/form/editar_contas.jsp" />
</action>
<action
    attribute="movimento_contasForm"
    name="movimento_contasForm"
    parameter="evento"
    path="/movimento_contas"
    scope="request"
    type="br.com.nota_fluxo.action.Movimento_contasAction"
    validate="false">
    <forward name="lista" path="/form/lista_movimento_contas.jsp" />
    <forward name="editar" path="/form/editar_movimento_contas.jsp" />
</action>
<action
    attribute="nota_fiscalForm"
    name="nota_fiscalForm"
    parameter="evento"
    path="/nota_fiscal"
    scope="request"
    type="br.com.nota_fluxo.action.Nota_fiscalAction"
    validate="false">
    <forward name="lista" path="/form/lista_nota_fiscal.jsp" />
    <forward name="editar" path="/form/editar_nota_fiscal.jsp" />
</action>
<action
    attribute="nota_itemForm"
```

```
            name="nota_itemForm"
            parameter="evento"
            path="/nota_item"
            scope="request"
            type="br.com.nota_fluxo.action.Nota_itemAction"
            validate="false">
            <forward name="lista" path="/form/lista_nota_item.jsp" />
            <forward name="editar" path="/form/editar_nota_item.jsp" />
        </action>
        <action
            attribute="ramo_atividadeForm"
            name="ramo_atividadeForm"
            parameter="evento"
            path="/ramo_atividade"
            scope="request"
            type="br.com.nota_fluxo.action.Ramo_atividadeAction"
            validate="false">
            <forward name="lista" path="/form/lista_ramo_atividade.jsp" />
            <forward name="editar" path="/form/editar_ramo_atividade.jsp" />
        </action>
        <action
            attribute="tipo_contas_pagarForm"
            name="tipo_contas_pagarForm"
            parameter="evento"
            path="/tipo_contas_pagar"
            scope="request"
            type="br.com.nota_fluxo.action.Tipo_contas_pagarAction"
            validate="false">
            <forward name="lista" path="/form/lista_tipo_contas_pagar.jsp" />
            <forward name="editar" path="/form/editar_tipo_contas_pagar.jsp" />
        </action>
        <action
            attribute="tipo_contas_receberForm"
            name="tipo_contas_receberForm"
            parameter="evento"
            path="/tipo_contas_receber"
            scope="request"
            type="br.com.nota_fluxo.action.Tipo_contas_receberAction"
            validate="false">
            <forward name="lista" path="/form/lista_tipo_contas_receber.jsp" />
            <forward name="editar" path="/form/editar_tipo_contas_receber.jsp" />
        </action>
    </action-mappings>

    <!-- ========== Controller Configuration ================================= -->
    <controller processorClass="org.apache.struts.tiles.TilesRequestProcessor" />

    <!-- ========== Message Resources Definitions ============================ -->

    <!-- ========== Plug Ins Configuration ================================== -->
    <message-resources parameter="br.com.nota_fluxo.ApplicationResources" />
    <plug-in className="org.apache.struts.tiles.TilesPlugin">

        <!-- Path to XML definition file -->

        <!-- Set Module-awareness to true -->
        <set-property property="definitions-config" value="/WEB-INF/tiles-defs.xml" />
        <set-property property="moduleAware" value="true" />
    </plug-in>
    <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>

    <global>

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

    </global>

    <formset>
                <form name="/empresa">
```

```
                <field    property="empresa.cd_emp" depends="required">
                     <arg0 key="empresaForm.cd_emp.displayname"/>
                </field>
        </form>
    </formset>

</form-validation>
```

Avaliação do Framework Struts para Implementação de Aplicações Web usando Padrão Modelo-Visão-Controlador

Alice Alves Corrêa
Universidade Federal de Santa Catarina – Depto. de Informática e Estatística
Florianópolis – SC. alice@yatech.net

Resumo

Este artigo analisa os aspectos específicos do Framework Struts. O padrão Modelo-Visão-Controlador é especificado. São relacionadas as capacidades e fraquezas do Framework, bem como aspectos relacionados a análise de custo de implantação para empresas.

1. Introdução

Grande parte do desenvolvimento de software está migrando para o ambiente da Internet. Nas empresas, a primeira "aparição" na Internet acontecia por meio de páginas estáticas desenvolvidas pela equipe do departamento de marketing ao invés da equipe de desenvolvimento de sistemas. Com o tempo, a necessidade de conteúdo dinâmico, e de formulários para comunicação com clientes e fornecedores e a necessidade de integração com sistemas legados fez com que a equipe de desenvolvimento precisasse participar do processo.

Ao mesmo tempo, o desenvolvimento está sendo realizado com novas tecnologias e ferramentas de construção. Novas arquiteturas surgem e o ritmo de inovação é enorme.

O desenvolvimento para a Internet exige conhecimentos específicos e requer adaptação dos desenvolvedores. Entretanto o problema é que algumas empresas são incapazes de aprender com os erros do passado. Entre eles podemos citar: sistemas de complexidade crescente, sem arquitetura, sem documentação, construídos sem planejamento, analise e design, e que posteriormente terão um altíssimo custo de manutenção. (BELLOQUIM, 2004)

2. O Framework Struts

O Projeto Jakarta Struts é um projeto de código aberto patrocinado pela Apache Software Foudation. O projeto Struts foi modelado com a intenção de ser um framework para a criação de Aplicações Web que facilmente separa a camada de

apresentação das camadas de transações e de persistência. . Sua implementação foi realizada na linguagem Java tendo como base o padrão de projeto (design pattern) Modelo-Visão-Controlador (MVC – Model-View-Controller). Desde sua concepção, Struts fornece um suporte bastante completo aos desenvolvedores de aplicações Web e está rapidamente se transformando em um framework de desenvolvimento dominante na comunidade de desenvolvedores Web (HUSTED, DUMOULIN, FRANCISCUS et. al, 2003).

O Framework possui esse nome em referência às estruturas que sustentam nossas construções civis. Essa é uma excelente analogia, pois ilustra o papel do Struts no desenvolvimento de aplicações Web. Quando estruturas físicas são construídas, os engenheiros utilizam estruturas para dar sustentação a cada andar de uma construção. De forma similar, os engenheiros de software utilizam o struts para dar sustentação a cada camada de uma aplicação (HUSTED, DUMOULIN, FRANCISCUS et. al, 2003).

O Struts impõe a utilização de um conjunto de regras bem definidas para delimitar camadas de interface, lógica de negócios e persistência com o objetivo de integrá-las, deixando que o desenvolvedor tenha controle do fluxo da lógica e conseqüentemente permitindo o gerenciamento do problema.

Embora o Framework tenha sido construído com as tecnologias J2EE, ele não é parte da plataforma J2EE padrão. Portanto podemos considerá-lo complementar a todos os padrões J2EE (não existem características que se sobrepõe ou que são incompatíveis). (HUSLY e YU, 2002).

O Framework Struts é uma implementação do lado do servidor do MVC usando uma combinação de JSPs, JSP tags e Servlets Java. A organização de componentes de desenvolvimento para a Web pode ser classificada em antes e depois da utilização do framework. Antes do Struts podemos verificar a mistura entre HTML e Java, separada da camada de serviços. Com a utilização do framework podemos verificar uma maior modularização na distribuição dos elementos. (DAVIS, 2001)

3. O MVC

O padrão MVC foi originado do SmallTalk, e consiste em três elementos: o Modelo, a Visão e o Controlador, conforme descrito abaixo (CAVANESS, 2002):

- Modelo: Representa os objetos de dados. O modelo é o que está sendo manipulado e apresentado ao usuário.
- Visão: É a apresentação do modelo na tela. É o objeto que apresenta o estado dos objetos de dados.

- Controlador: Define a maneira que a interface com o usuário reage às entradas de um usuário. É o objeto que manipula o Modelo.

Os benefícios trazidos pela utilização do MVC são os seguintes (CAVANESS, 2002):

- Confiabilidade: As camadas de apresentação e de transação têm uma clara delimitação, o que permite mudar a aparência da aplicação sem recompilar o código do modelo ou do controlador. Isto permite que alterações na aparência não afetem a estabilidade do código responsável pela execução de transações.
- Reutilização elevada e adaptabilidade: O MVC permite o uso de múltiplos tipos de visão, todas acessando o mesmo código no servidor.
- Baixo custo de desenvolvimento e ciclo de vida minimizado: O MVC torna possível possuir programadores de lógica da aplicação e designers responsáveis por manter as interfaces com o usuário, aumentando assim a produtividade e a qualidade do produto final promovida pela especialização das atividades.
- Desenvolvimento rápido: o tempo de desenvolvimento pode ser reduzido substancialmente porque os programadores focar apenas em transações, e os designers podem manter seu foco apenas na apresentação.

4. Vantagens e limitações do Framework Struts

Serão analisados os resultados obtidos com o desenvolvimento de uma aplicação Web, descrita nos capítulos anteriores, utilizando o Framework Struts, enfatizando as vantagens e limitações encontradas no framework, as mudanças ocorridas no processo de desenvolvimento, e as alterações no custo, no desempenho e na escalabilidade do produto final.

4.1 Pontos Fracos do Struts

Entre as principais limitações observadas no Framework Struts podemos citar (HUSTED, DUMOULIN, FRANCISCUS et. al, 2003):

- Inexistência de um modelo de eventos: O struts está intrinsecamente ligado ao modelo solicitação-resposta usado pelo protocolo HTTP, o que pode ser um fator limitante para desenvolvedores acostumados com eventos.
- Dificuldades para depuração: Não existe nenhum suporte para a depuração automática. Os desenvolvedores tem que fazer uso da criação de “pontos de interrupção” manuais escrevendo mensagens no registro do contêiner ou usando outra tecnologia como o log4j por exemplo.

- Modelo de Dados: O modelo de dados é deixado completamente em aberto para o desenvolvedor, não deixando nenhum tipo de recomendação nem possuindo um modelo de dados padrão.
- ActionServlet: Apenas um ActionServlet é usado em uma aplicação, o que pode ocasionar conflitos na configuração.
- Compreensão dos Componentes: É necessário o conhecimento dos componentes do Struts para que se consiga trabalhar com ele, incluindo o conhecimento das classes e de como elas interagem.
- Fornecedor sem Prioridade no Suporte: A Apache Software Foundation é mantida por voluntários, de forma que nem sempre é possível obter respostas e correções de forma imediata.
- Tamanho da Lista de E-mails: A lista da comunidade do Struts aumentou significativamente, e com isso pode ser difícil encontrar a informação desejada.
- Demora na atualização das versões do framework: As versões do Struts podem demorar a sair em comparação a outros produtos, sendo que muitos desenvolvedores têm que usar construções em fase de teste para poderem usar as últimas alterações.
- Limites da internacionalização: Os recursos para internacionalização funcionam adequadamente para mensagens de erro e rótulos, mas não são adequados para gerenciar blocos maiores de texto. Blocos maiores de texto necessitam de elementos de formatação (inicio de parágrafo, elementos em negrito ou itálico, fontes de tamanhos diferentes), o que é possível fazer, incluindo elementos de formatação HTML diretamente nas mensagens, entretanto não é uma solução adequada, pois mistura a formatação com o texto.
- Exceções do JSP: As exceções do JSP não são localizadas, de forma que aparecem sempre em inglês.

4.2 Pontos fortes do Struts

Como pontos fortes do Struts para desenvolvimento de aplicações Web podemos destacar as seguintes características:

- Centrado no Modelo Solicitação-Resposta: O Struts é desenvolvido no modelo solicitação-resposta do protocolo HTTP, sendo bem familiar para desenvolvedores web.
- Registro Padrão: O Struts não necessita de outro pacote para ser instalado, configurado ou entendido no contêiner.

- Registro da depuração opcional: O Struts grava opcionalmente mensagens de status que podem ser úteis para depuração.
- Modelo Neutro: O Struts não é atrelado a nenhuma camada particular de persistência.
- Configuração centralizada: A configuração do Struts encapsula detalhes para uma aplicação ou módulo.
- Arquivo de Recursos específicos: O Struts permite o uso de um arquivo de recursos específico para cada local, tornando possível a internacionalização de aplicações.
- Simplicidade: O Struts possui relativamente poucas classes básicas para que os desenvolvedores aprendam.
- Código Aberto: O Código Fonte é aberto, fazendo com que todos possam saber o conteúdo do framework.
- Comunidade de Desenvolvedores: Muitas versões com correções fornecidas por desenvolvedores estão disponíveis.
- Comunidade de Revendedores: O Struts é incorporado em vários outros produtos.
- Versões estáveis: As versões formais do Struts ficam em um período de testes sem prazo definido, para assegurar que o produto tenha uma alta qualidade.
- Extensões Tag: O Struts inclui um conjunto de extensões de tags de uso geral, além das que usam recursos do framework, para evitar a utilização de scriptlets.
- Bem documentado: A documentação do Struts geralmente é bem detalhada, de forma que raramente será necessário verificar o código fonte da aplicação.
- Baseado em Padrões: O Framework implementa vários padrões em sua arquitetura, de forma a serem bem familiares para a maior parte dos desenvolvedores.
- Extensível: Todas as definições padrão podem ser configuradas. O desenvolvedor pode personalizar classes como Action e ActionForm bem como carregar as subclasses em seus lugares.

4.3 Custo de Implantação do Struts

O custo total de implantação do struts não envolve apenas dinheiro, mas também tempo e aprendizado. Iremos abordar quatro fatores: custo de licença, custo de treinamento, custo de suporte e de manutenção. (HUSLY e YU, 2002).

Custo de licença: É um software de código aberto, sob a licença da Apache Software Foundation, portanto é gratuito. Para revendedores independentes de software, o framework pode ser redistribuído sem o pagamento de royalties sob uma licença relativamente liberal.

Custo de treinamento: É um dos maiores interesses quando se está adotando um novo framework. É importante lembrar que o Struts é um framework construído sobre a plataforma J2EE. Portanto é necessário que o desenvolvedor tenha experiência em JSP, Servlets e sobre o conceito MVC para assimilar as principais características do Struts usando uma a duas semanas no mínimo. Para os desenvolvedores que não possuírem esses conhecimentos prévios, logicamente levará mais tempo. Pode levar ainda mais tempo para que o aprendizado se torne produtivo, uma vez que aprender as características não é o mesmo que aprender o paradigma.

É interessante considerar o treinamento com instrutores para desenvolvedores que estarão expostos a uma plataforma totalmente nova. O custo em termos do dinheiro envolvido pode ser muito maior nesse caso. Entretanto antes de avaliar um custo de treinamento é importante verificar o nível do conhecimento dos desenvolvedores. O que é importante ressaltar com isso é que o custo com o treinamento com o Struts é relativamente menor do que com o treinamento em JSP e Servlets.

Custo de manutenção e suporte: O custo de suporte é difícil de mensurar, uma vez que como faz parte de uma comunidade de software aberto, um suporte excelente porém informal está disponível em listas de correspondência do Struts. Existe suporte pago de alguns vendedores como o Multitask (www.multitask.au). Existe outra opção de ter suporte diretamente de pessoas com grande conhecimento em Struts, uma vez que o seu código é disponível gratuitamente e pode ser modificado. Para projetos de desenvolvimento, uma regra que é comumente usada para avaliar do custo do suporte do Struts é igualar a metade do custo do suporte que um programador J2EE. (HUSLY e YU, 2002).

4.4 Análise de Escalabilidade e Desempenho

Escalabilidade geralmente significa escalabilidade em tempo de execução. Entretanto outro tipo de escalabilidade é igualmente importante: a escalabilidade do tamanho do projeto.

Escalabilidade em tempo de execução: O Struts compartilha muitas características do padrão Servlet. Os Servlets são muito leves, de forma que poucos recursos são necessários para inicializar e manter uma instância. Através da arquitetura

Servlet, uma instância gerencia requisições simultâneas(GABHART, 2003). Assim as Actions são sem estado e são compartilhadas por várias threads (linhas de execução). Dessa forma para cada requisição é criada uma nova linha de execução e o método service() é executado em cada thread conforme mostrado na figura 1. Isso significa que o número de instâncias Action é constante e não aumentam com o volume de tráfego. (HUSLY e YU, 2002).

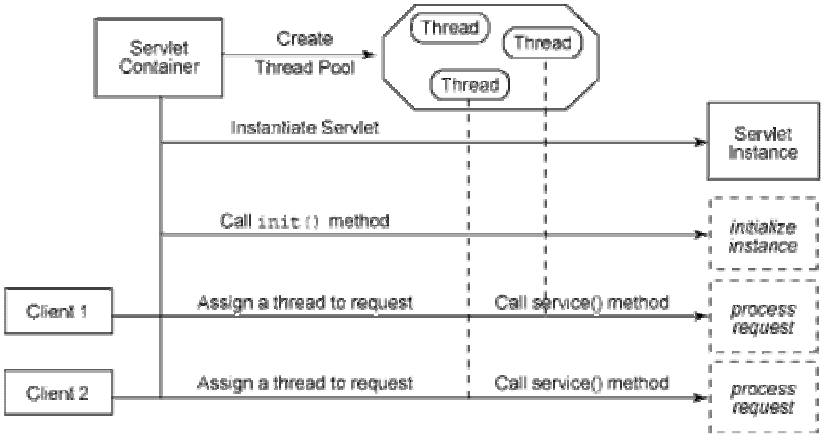

Figura 1 – O Modelo Servlet de linhas de execução (threads)(GABHART, 2003)

Cada escopo escolhido também possui implicações na escalabilidade e performance. Selecionar o escopo para um componente depende muito da forma como será utilizado.

Componentes com escopo de página (apenas para JSPs) ou requisição são os que menos causam problemas, pois os dados possuem um tempo de vida limitado.

O Escopo de aplicação é gerenciável se usado apenas para leitura de valores entre os Servlets de uma aplicação.

O estado no escopo de sessão é o que possui o maior impacto na escalabilidade e performance de uma aplicação Web. Para cada usuário conectado é acumulado o componente na sessão, de forma diferente ao escopo de aplicação que compartilha componentes com todos os usuários e Servlets. Também é importante lembrar que o escopo de sessão existe através das requisições, ao contrário do escopo de requisição, que descarta componentes quando a resposta é fornecida. (JONHSON, STEARNS, SINGH et. al, 2002).

Escalabilidade do tamanho do projeto: Esse aspecto está relacionado com a flexibilidade do framework em organizar o código. Struts introduz o suporte a sub-aplicações, o que permite o particionamento de uma aplicação em módulos independentes.

Desempenho: o Struts confia na introspecção Java Bean. Um dos aspectos relativo a desempenho que geralmente se comenta é a sobrecarga que o Struts ocasiona. Primeiramente deve estar claro que o que consome tempo na introspecção Java é a procura pelos métodos, e não a invocação deles. O Struts guarda essas informações, e elimina essa procura depois da inicialização. (HUSLY e YU, 2002).

5. Conclusão

O desenvolvimento de software para a Internet, desde o seu início, seguiu os mesmos caminhos que o desenvolvimento do software aplicativo percorreu há trinta anos atrás, cometendo os mesmos erros, gerando aplicações complexas e difíceis de manter. Infelizmente, vários dos preceitos pregados pela Engenharia de Software não foram incorporados ao desenvolvimento de aplicações Web.

É possível observar que desenvolver software para a Internet é muito mais do que se preocupar apenas com a apresentação da interface com o usuário. A interface com bancos de dados e o processamento da lógica de negócios precisam ser bem delimitados para permitir uma maior manutenibilidade e reuso de código das aplicações.

Uma das maiores vantagens de se usar um framework de desenvolvimento é a habilidade de estender e customizar as características de acordo com a necessidade da aplicação.

O framework Struts vem se firmando como um padrão no seu segmento, influenciado principalmente pela comunidade mundial de colaboradores que conferem estabilidade e robustez inestimáveis. A capacidade de ser utilizado em conjunto com várias tecnologias também pode ser citada como um fator importante para a sua popularidade.

Para conseguir utilizar toda a infra-estrutura oferecida pelo framework, é necessário que o desenvolvedor não conheça apenas o funcionamento e a forma de configuração do Struts, mas que conheça também Servlets e JSP.

Entretanto devemos a cada projeto avaliar a real necessidade da implantação de frameworks de desenvolvimento como o Struts, uma vez que estes requerem conhecimento de padrões e consequentemente adiciona complexidade à aplicação. Em projetos de pouca complexidade que sejam desenvolvidos e mantidos por apenas um indivíduo, não se justificaria o investimento efetuado para o aprendizado do framework. Mesmo nestes casos, seria interessante aplicar padrões como o MVC mesmo que sem o auxílio do framework. Já para projetos de sistemas mais complexos, não resta dúvida que o framework Struts contribui consideravelmente para a melhoria da qualidade do

software, auxiliando na divisão de tarefas entre desenvolvedores e designers, facilitando a evolução do sistema e aumentando a manutenibilidade.

6. Referências

CAVANESS, Chuck. Programing Jakarta Struts. O´Reilly & Associates Inc. 2002. Sebastopol, CA. USA.

HUSTED Ted, DUMOULIN Cedric, FRANCISCUS George et al. Struts in Action. Manning Publications Co.2003. USA.

Jakarta Struts Framework Disponível em: <http://struts.apache.org>. Acesso em 2 de dezembro de 2004.

Tomcat Project. Disponível em: <http://jakarta.apache.org/tomcat/index.html>. Acesso em 2 de dezembro de 2004.

BELLOQUIM, Átila. 2004. Desenvolvimento Tradicional x Desenvolvimento para a Internet. . Disponível em: <http://www.b4unews.com.br/conteudo.php?ct=noticias&idnoticia=426>. Acesso em 2 de novembro de 2004.

HUSLY, Harry, YU, Jonh. Issues in Struts Adoption. 2002. Disponível em: <http://www.scioworks.net/devnews/articles/struts_adoption_issues/>. Acesso em 10 de outubro de 2004.

JONHSON Mark, STEARNS Beth, SINGH Inderjeet et al. Desingin Enterprise Applications with the J2EE Platform. Addison-Wesley.2002 California USA.

DAVIS, Malcom. Struts, an open source MVC implementation.2001. Disponível em: <http://www-106.ibm.com/developerworks/java/library/j-struts/>. Acesso em 10 de outubro de 2004.

GABHART, Kyle. J2EE technologies for the stateless network. 2003. Disponível em: <http://www-106.ibm.com/developerworks/java/library/j-pj2ee1/>. Acesso em 2 de dezembro de 2004.