

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CURSO DE SISTEMAS DE INFORMAÇÃO
DISCIPLINA: INE 5632 – PROJETOS II

Uma Ferramenta Introdutória à Formação de Pessoal
na Implantação de CRM Baseado em
Gerenciamento de Projetos Ágeis



HENRIQUE DE BRITTO FARIAS
ROBERTA LINGNAU DE OLIVEIRA
FLORIANÓPOLIS, 2005

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CURSO DE SISTEMAS DE INFORMAÇÃO
DISCIPLINA: INE 5632 – PROJETOS II

Uma Ferramenta Introdutória à Formação de Pessoal
na Implantação de CRM Baseado em
Gerenciamento de Projetos Ágeis

Orientadora: Profa. Dra. Maria Marta Leite

Banca Examinadora:
Alessandra Schweitzer
Elizabeth Sueli Specialski

HENRIQUE DE BRITTO FARIAS
ROBERTA LINGNAU DE OLIVEIRA
FLORIANÓPOLIS, 2005

Índice

Índice	3
Lista de Abreviaturas e Siglas	5
1 Apresentação	6
2 Introdução	6
2.1 <i>Conhecer ou Não-Conhecer o Cliente</i>	7
2.2 <i>Satisfação ou Insatisfação do Cliente</i>	7
2.3 <i>O Tempo do Cliente</i>	7
2.4 <i>Questões Financeiras</i>	8
2.5 <i>Processo de Formação e Educação</i>	9
2.6 <i>Outras Considerações</i>	10
3 Tema e Delimitação do Tema	11
3.1 <i>Tema</i>	11
3.2 <i>Delimitação do Tema</i>	11
4 Objetivos	12
4.1 <i>Objetivo Geral</i>	12
4.2 <i>Objetivos Específicos</i>	12
5 Motivação	12
6 Justificativa	13
7 Principais Conceitos Relacionados a CRM	15
7.1 <i>Definição</i>	15
7.2 <i>CRM como Filosofia e Não Metodologia</i>	15
7.3 <i>A Importância da Personalização</i>	16
7.4 <i>A Satisfação do Cliente em Primeiro Lugar</i>	16
7.5 <i>Adoção de Estratégias para CRM</i>	17
7.6 <i>CRM não é Tecnologia</i>	17
7.7 <i>Comunicando-se com o Cliente</i>	17
8 Motivações para a Implantação de CRM	18
8.1 <i>Encantar o Cliente Externo</i>	18
8.2 <i>Encantar o Cliente Interno</i>	18
8.3 <i>Melhorar o Desempenho Financeiro</i>	19
8.4 <i>Aumentar a Fidelidade e Satisfação do Cliente</i>	19
9 A Contribuição da Agilidade para a Implantação de CRM	21
9.1 <i>O que significa Agilidade?</i>	21
9.2 <i>O Manifesto Ágil</i>	21
9.2.1 <i>FDD</i>	23
9.3 <i>Agilidade em CRM</i>	24

10 Implantação de CRM.....	25
10.1 <i>Etapas na Implantação de CRM</i>	27
10.2 <i>Implantação com Tecnologia</i>	27
10.3 <i>Pessoas e sua Importância na Implantação do CRM</i>	277
10.4 <i>Fatores Críticos de Sucesso do Processo de Formação</i>	28
10.4.1 <i>Formação e Disponibilidade de Pessoal</i>	288
10.4.2 <i>Processo de Formação Constante</i>	288
10.4.3 <i>Evolução do Processo de Formação e suas Mudanças</i>	299
10.4.4 <i>As Conseqüências do Processo de Formação</i>	299
10.4.5 <i>Os Benefícios do Processo de Formação e o Desejo de Aprender</i>	299
10.4.6 <i>O Tempo</i>	30
10.4.7 <i>Igualdade para Todos</i>	30
10.4.8 <i>Educação e Processo de Formação Caminham Juntos</i>	30
10.5 <i>Preparando os Colaboradores para o Gerenciamento de Relacionamento com o Cliente</i>	30
10.5.1 <i>Ensinar Colaboradores a Gerenciar Informações</i>	311
10.5.2 <i>Obter Informações Sobre os Clientes</i>	31
10.5.3 <i>O Processo Decisório ao Longo da Organização</i>	322
10.5.3.1 <i>Tipos de Decisão</i>	33
10.5.3.2 <i>Modelos de Tomada de Decisão</i>	33
10.5.4 <i>Compartilhamento de Informações na Organização</i>	35
10.5.5 <i>Mudanças que Impactam nos Colaboradores</i>	36
11 O Aplicativo Desenvolvido	36
11.1 <i>Introdução à Ferramenta e Projeto de Desenvolvimento</i>	366
11.2 <i>A Usabilidade e a Ferramenta</i>	377
11.3 <i>Funcionalidades</i>	38
11.4 <i>Escolha do Banco de Dados</i>	38
11.4.1 <i>Histórico do Postgresql</i>	38
11.4.1.1 <i>O PostgreSQL hoje</i>	39
11.5 <i>O Framework Struts</i>	39
11.5.1 <i>Frameworks</i>	39
11.5.2 <i>Definição do Framework Struts</i>	40
11.6 <i>Escolha do Servidor de Aplicação</i>	41
11.6.1 <i>Tomcat</i>	41
11.6.2 <i>Acesso ao aplicativo</i>	41
12 Dificuldades Encontradas.....	41
13 Trabalhos Futuros.....	42
14 Conclusão.....	42
15 Referências.....	43
Apêndices	47

Lista de Abreviaturas e Siglas

- APM – Agile Project Management*
- BSD – Berkeley Software Distribution*
- CRM – Customer Relationship Management*
- FDD – Feature-Driven Developmen*
- HTTP – HyperText Transfer Protocol*
- JDBC – Java Database Connectivity*
- JNDI – Java Naming and Directory Interface*
- J2EE – Java 2 Platform, Enterprise Edition*
- JSP – JavaServer Pages*
- MVC – Model View Controller*
- SGBD – Sistema Gerenciador de Banco de Dados*
- SSL – Secure Socket Layer*
- TI – Tecnologia da Informação*
- XP – Extreme Programming*

1 Apresentação

Este documento visa apresentar o trabalho desenvolvido como atividade da disciplina INE5632 – Projetos II. O projeto sendo desenvolvido está relacionado à área de Gerenciamento de Projetos Ágeis aplicado ao Processo de Formação Gerenciamento do Relacionamento com os Clientes.

2 Introdução

O mercado atual mostra-se extremamente competitivo, apresentando organizações que buscam constantemente a evolução dos serviços prestados aos seus clientes. O crescimento que visa lucros, o aumento da participação no mercado e a excelência em seus produtos e serviços faz parte desta busca. Sendo assim, novos modelos de gerência de organizações têm surgido. Ferramentas que auxiliam as organizações a atingir esses objetivos e o Gerenciamento do Relacionamento com o Cliente, daqui por diante dito somente CRM, têm ganhado destaque. O motivo para tal destaque é a visão de que é essencial ter clientes satisfeitos para que o negócio tenha sucesso.

São muitos os motivos que podem fazer com que uma organização perca seus clientes facilmente, levando-a ao seu fim. Através de uma boa estratégia CRM, as organizações podem aumentar a rentabilidade da base de clientes e reter clientes de alto valor (TRONCHIN, 2002).

A retenção ou fidelidade é conseguida pela utilização do conhecimento individual de cada cliente, que possibilita a personalização de aspectos da relação com os mesmos. Um bom serviço ao cliente é o mínimo que se espera de qualquer organização e isso está se tornando cada vez mais comum, deixando de ser um diferencial competitivo (TRONCHIN, 2002).

As organizações vêm tomando consciência que é muito mais barato manter um cliente do que investir em publicidade e propaganda para a aquisição de novos, de acordo com estudos divulgados por Philip Kotler (Administração de Marketing - a edição do novo milênio, 2000, Prentice Hall, p. 71). A filosofia por trás do CRM é tratar clientes diferentes de forma diferente. Conhecer as diferenças em termos de valor e de necessidades é fundamental para que a organização possa saber onde e como deve investir para personalizar a atenção aos melhores clientes (TRONCHIN, 2002).

Tendo essas considerações compreendidas procura-se agora analisar os motivos que podem levar uma organização a perder os seus clientes, pois isto nos

servirá de alerta para não cometer os mesmo erros, impedindo uma organização de obter destaque no mercado.

Existem cinco fatores críticos que podem levar uma organização a perder seus clientes (NEKI Technologies, 2004). Estes fatores são descritos a seguir.

2.1 Conhecer ou Não-Conhecer o Cliente

Há diversos tipos de clientes. O cliente de varejo, que é aquele que adquire o produto ou o serviço final e normalmente é representado por um indivíduo ou uma família. Há também o cliente que compra um produto e o adiciona ao produto que ele fabrica para venda a terceiros. Também existe aquele que adquire um produto para simplesmente vendê-lo. E ainda existem os clientes internos, que seriam as pessoas ou unidades de negócio dentro de sua organização (SWIFT, 2001).

Sendo assim, é necessário atender as necessidades de informações e conhecimento desses clientes para conhecer o que eles preferem. Examinando a distância, todos têm algo em comum e ao analisar-lhes estes se dividem em grupos, obtendo-se ao final muitos segmentos (SWIFT, 2001). Dessa forma demanda esforço conhecer os clientes, e deve-se ter em mente que estes esperam que você os conheça.

2.2 Satisfação ou Insatisfação do Cliente

Com um mercado extremamente competitivo e acirrado, as organizações estão fazendo o máximo esforço para agradar os seus clientes. O perfil dos clientes tem mudado muito nos últimos tempos, apresentando um perfil de impaciência e intolerância a falhas, atrasos e imperfeições. Presencia-se uma série de organizações repensando os seus relacionamentos com a finalidade de criar oportunidades para suas estratégias empresariais voltadas aos clientes. As organizações têm demonstrado um forte desejo de ter um espaço cativo no mercado, embora muitas não realizem os primeiros passos para estreitar os laços com o cliente – a avaliação do seu grau de satisfação. Tal avaliação deve ser periódica e constantemente realizada, de forma que não aborreça ou constranja os clientes.

2.3 O Tempo do Cliente

Nos anos 80, o mercado deu início a um processo de centralização de atenções no cliente, embora em uma escala reduzida e sem grande significância. Grande

parte das organizações continuou tentando dizer aos seus clientes como se comportar ao invés de responder as suas perguntas. Neste momento, a pressão que deveria ser imposta por parte dos clientes para que se fizessem empreendimentos de acordo com suas necessidades não foi exercida. Atualmente, a pressão existente por parte dos clientes é forte e contínua. As atividades corporativas que afetam o cliente devem ser realizadas no prazo determinado, sem atrasos, evitando aborrecimentos para os clientes e constrangimentos para a equipe. Antigamente as ações e decisões eram realizadas e tomadas no tempo da organização. Hoje, as tarefas devem ser executadas no tempo do cliente, sendo este tempo extremamente variável e um fator decisivo no relacionamento da organização com os seus clientes.

2.4 Questões Financeiras

A alta direção, os gerentes e demais colaboradores de uma organização devem ter a ciência de que investir na satisfação de seus clientes é um fator decisivo na forma de relacionar-se com os mesmos. A definição de uma estratégia para realização de tais investimentos é necessária. Crises econômicas também podem afetar a implantação destas estratégias.

Cita-se como referência um caso de marketing. A organização americana *L.L. Bean* pertence ao setor de vestuário e equipamentos para esporte por catálogo e reembolso postal (ZAIOWICZ, 2004). Tem como destaque o trabalho diretamente relacionado à satisfação do cliente.

Objetivando a conscientização de seus funcionários sobre a importância do cliente como foco principal, a organização ostenta um cartaz em seus escritórios expressando a idéia que tem sobre cliente:

"Um cliente é a pessoa mais importante do mundo neste escritório... Quer ele se comunique pessoalmente ou por carta. Um cliente não depende de nós... Nós é que dependemos dele. Um cliente não interrompe nosso trabalho... É a finalidade dele. Não estamos fazendo um favor de servi-lo... Ele está nos fazendo um favor dando-nos a oportunidade de fazê-lo. Um cliente não é alguém com quem discutir ou debater. Ninguém jamais venceu uma discussão com um cliente. Um cliente é uma pessoa que nos traz seus desejos. É nossa obrigação lidar com eles de maneira lucrativa para ele e para nós".

Segundo estudos desenvolvidos por Frederick Reichheld (Harvard Business School Press, 1996) e divulgados por Philip Kotler (Administração de Marketing - a

edição do novo milênio, 2000, Prentice Hall, p. 71), todo e qualquer esforço oferecido pela organização para satisfazer e reter seus clientes deve ser sempre reforçado, dado que a aquisição de novos clientes pode custar até cinco vezes mais do que os investimentos feitos para satisfazer e reter clientes atuais.

Outro ponto importante levantado por tais estudos é que a atividade de criar clientes para uma organização deve ser contínua, pois, uma organização perde, em média, 10% de seus clientes a cada ano, pelos mais diversificados motivos, conforme definidos ao longo do presente trabalho.

Para apurar os custos em relação à aquisição de novos clientes, apresenta-se um modelo adaptado (ZAIOWICZ Consultoria Fiscal, 2004):

Calculando-se que uma organização deve oferecer um número médio (NMV) de quatro visitas para converter um potencial cliente em cliente, multiplica-se este valor (4) pelo custo médio de uma visita de vendas para prospecção de cliente (CPC), computando aqui todos os custos incorridos na operação (y). Em termos práticos, o modelo da equação pode ser compreendido da seguinte forma:

- Custo de Prospecção de Cliente: $CPC = y$ (R\$)
- Número Médio de Visitas: $NMV = 4$
- Custo de Aquisição Novo Cliente: $CNC = y$ (R\$) x (4)

Conforme apresentado pelos estudos citados acima, as estimativas realizadas através dos cálculos declarados comprovam que o investimento em novos clientes é cinco vezes maior do que o investimento em satisfazer os clientes atuais. Entretanto, é oportuno apurar os resultados de acordo com o caso de cada organização.

Conclusivamente, uma relação entre o que a organização despense para a criação de novos clientes e o quanto disponibiliza para manter os clientes existentes deve ser realizada. Cabe então à organização identificar as melhores oportunidades e trabalhá-las de forma construtiva para o seu futuro.

2.5 Processo de Formação e Educação

Os colaboradores de uma organização representam diretamente a organização na qual trabalham. Através deles as características de uma organização transparecem e o nível de importância do cliente fica evidente. Assim sendo, a organização deve investir firmemente na forma como se apresenta – seus colaboradores.

Uma preocupação que deve habitar a mente dos líderes das organizações é: como assegurar que o processo de formação e educação revertam em sucesso para a

organização? Alguns dos motivos para a existência desta preocupação são: tempo investido, ausência de pessoal nas atividades da organização e investimento financeiro, que por muitas vezes pode ser alto. Esse trabalho procura aprofundar-se mais nessa fase tão importante do processo, sugerindo maneiras práticas de realizar o treinamento com o aplicativo desenvolvido. O detalhamento disso será visto mais adiante.

2.6 Outras Considerações

Um motivo particular para a perda de clientes é a falta de uma definição sólida por parte da organização de seu público alvo e da forma de comunicação com o cliente. As organizações precisam saber e conhecer a vantagem de uma estratégia de relacionamento. Para isto deve-se avaliar a necessidade de se implantar uma estratégia CRM. Através do CRM a organização acabará formando uma base sólida para manter seus clientes existentes, podendo ainda entrar em mercados ainda não explorados. É com essa filosofia que este trabalho se dispõe a ajudar a resolver os problemas anteriormente apresentados, ou pelo menos encaminhar a solução.

Apesar da implantação do Gerenciamento do Relacionamento com o Cliente ser vital para um bom andamento da organização, a falta de tempo, pouco material que ajude as organizações nesse processo e a escassez de recursos propiciam um ambiente desfavorável para a prática do mesmo, principalmente as organizações de médio e pequeno porte. Dessa forma, o CRM é deixado de lado, quando deveria ser um ponto estratégico.

De acordo com Leite (Leite 2004, p. 21), já que o CRM deve ser uma tarefa tanto operacional quanto estratégica, para que ocorra um bom andamento de uma organização, sua implantação pode ser tratada como um Projeto, ou um conjunto deles. Assim sendo, podem ser utilizados as premissas e os conceitos abordados pelo Gerenciamento de Projetos para colocá-lo em prática, e ainda mais precisamente, pode ser utilizado um Gerenciamento de Projetos Ágeis para satisfazer de forma mais ágil os anseios das organizações relativos ao aspecto CRM. Através de uma Gestão de Projetos de forma ágil, as organizações deverão obter um tempo de resposta muito menor para os seus investimentos, considerando-se o Gerenciamento de Projetos Tradicional. Desta forma, a centralização no cliente passa a ser o foco, fazendo com que as organizações tornem operacional este foco, o que deveria acontecer desde a sua fundação.

3 Tema e Delimitação do Tema

Através do contexto relatado anteriormente, formulamos o problema que trata este trabalho: De que forma tornar a implantação do Gerenciamento de Relacionamento com o Cliente mais rápida e operacional para as organizações, sem deixar de considerar os conceitos de CRM?

3.1 Tema

O tema do Trabalho de Conclusão de Curso é a análise do Gerenciamento do Relacionamento com Clientes sob a ótica dos conceitos de Gerenciamento de Projetos Ágeis.

3.2 Delimitação do Tema

O trabalho proposto tem como foco o estudo e análise dos conceitos fornecidos pelo Gerenciamento de Projetos Ágeis com o fim de facilitar o processo de formação de pessoal em uma organização. Também visa estabelecer como estes conceitos podem ser utilizados para a implantação de um Gerenciamento de Relacionamento com Cliente ou CRM. Procura-se, entre outros objetivos, desenvolver uma ferramenta que instrua e motive os colaboradores, a alta direção e demais interessados a pensar de acordo com os preceitos de CRM.

Reconhece-se que as administrações das organizações não podem perder tempo, pois tempo significa dinheiro. Sendo assim, há a necessidade da implantação de um projeto de Gerenciamento de Relacionamento com o Cliente em um curto período de tempo. Esse é o desafio de nosso estudo. Através da utilização de um Gerenciamento de Projetos Ágeis, o tempo de resposta às perguntas da administração deve ser minimizado, aumentando o seu interesse em aprimorar a forma de relacionar-se com os seus clientes.

Este estudo parte do princípio que a importância do Gerenciamento do Relacionamento com os Clientes ainda não é totalmente entendida pelo pessoal envolvido dentro das organizações, sendo tal entendimento adquirido através da aplicação que será desenvolvida.

4 Objetivos

A seguir serão apresentados os objetivos do presente trabalho.

4.1 Objetivo Geral

Apresentar estudos de como Gerenciamento de Projetos Ágeis é interessante para aplicação de CRM em uma organização e, com base na importância da fase de processo de formação, desenvolver uma ferramenta que visa introduzir os conceitos básicos de CRM utilizando Gerenciamento de Projetos Ágeis.

4.2 Objetivos Específicos

1. Definir o que é CRM;
2. Definir o que é Gerenciamento de Projetos Ágeis;
3. Demonstrar como Gerenciamento de Projetos Ágeis pode ser aplicado ao processo de formação em CRM;
4. Desenvolver aplicação básica que facilite o processo de formação em CRM utilizando Gerenciamento de Projetos Ágeis.

5 Motivação

Primeiramente, a grande motivação para este trabalho é a de enfatizar a importância do cliente junto às organizações. Há poucas referências que ajudam as pequenas e médias organizações na implantação de uma estratégia alinhada ao cliente (Leite, 2004, p. 229). Também muitas das organizações que estão em busca dessa filosofia têm fracassado nessa tentativa, muitas vezes por erro no foco do investimento, que tem sido basicamente em tecnologia, desconsiderando muitos outros mecanismos importantes para que a organização dê prioridade aos seus clientes. As que obtiveram sucesso conseguiram por procurar implantar uma estratégia de negócios completa (Leite, 2004, p. 229). Ajudar nesse processo é força motivadora para esse trabalho.

Em segundo plano permanece a necessidade das organizações em ter o seu tempo gasto da melhor forma possível. Vislumbra-se a metodologia de Gerenciamento de Projetos Ágeis como o meio de dar um retorno rápido de uma implantação à administração. Procura-se assim mostrar com maior clareza o andamento do projeto como um todo, para que a administração da organização saiba e tenha cada vez mais a

certeza de que o seu precioso tempo está sendo bem usufruído, e que trará resultados satisfatórios. Tal clareza pode ser obtida através da definição de etapas enxutas e bem-definidas. As etapas devem ser realizadas num curto período de tempo e têm como característica a adaptabilidade a mudanças que por ventura possam surgir ao longo do projeto. O foco são os indivíduos e suas interações, gerando um trabalho em conjunto para atingir os objetivos traçados. Além disso, o processo é interativo e incremental, gerando *feedback* constante para a administração da organização.

Por último, mas não menos importante, é vital para uma organização preocupar-se com a forma que se apresenta para o mercado. Um dos pontos a serem analisados quando se trata de apresentação são os colaboradores e o leque de oportunidades que surgem quando estes procuram demonstrar com dedicação a preocupação com a satisfação do cliente. Assim sendo, o processo de formação constante e atualização do currículo dos colaboradores são cuidados importantes a serem tomados por qualquer organização que tem como objetivo a centralização do cliente. A organização e seus colaboradores devem estar atentos às constantes evoluções na forma de relacionamento com clientes. As experiências adquiridas devem ser aproveitadas de forma que a organização possa melhorar seus próprios relacionamentos.

6 Justificativa

Vivencia-se hoje a Era da Informação, tendo como suporte diversas tecnologias modernas. Dentre tais tecnologias, encontra-se a informática, representada tanto pelo hardware quanto pelo software, propiciando grandes conquistas na comunicação. Graças a tais conquistas, os computadores tornaram-se indispensáveis nas instituições, e estão se tornando um recurso cada vez mais presente nos lares. Como conseqüência, tanto o software quanto o hardware dos sistemas de informática necessitam cada vez mais de profissionais devidamente habilitados (UFSC, Curso de Sistemas de Informação, 2001).

As organizações, forçadas pelo dinamismo do crescente mercado atual, passam a exigir mais responsabilidade e produção de seu quadro de colaboradores. Os colaboradores, por sua vez, necessitam buscar incessantemente acesso à informação, que se dá de forma fácil e rápida através da utilização da informática.

O curso de Sistemas de Informação tem como objetivo, formar profissionais que, apoiados nos conceitos e técnicas de informática, teoria de sistemas e administração, contribuam na solução de problemas de tratamento de informação nas

organizações, por meio da concepção, construção e manutenção de modelos informatizados de automação corporativa (UFSC, Curso de Sistemas de Informação, 2001).

A informação, além de ter suporte das diversas tecnologias modernas, possui ainda o apoio do pessoal envolvido em sua concepção e desenvolvimento. Assim sendo, o profissional de Sistemas de Informação também possui responsabilidade sobre tais indivíduos e suas interações.

A utilização do Gerenciamento de Projetos Ágeis é o elo entre o curso de Sistemas de Informação e o CRM, pois de acordo com a sua premissa básica, indivíduos e interações são mais importantes que processos e ferramentas. CRM se trata disso: da interação do cliente com a organização, tornando evidente a capacidade de um profissional de Sistemas de Informação o trabalho e estudo voltado para esta área.

Outra justificativa importante é que gerenciar uma organização sem estar preocupado com os clientes tem sido, em muitas vezes, o grande motivo de instabilidade e até mesmo naufrágio econômico das organizações. As organizações objetivam aumentar cada vez mais sua carteira de clientes, muitas vezes sem considerar a importância de um bom relacionamento com os seus clientes efetivos. Conhecer melhor os seus clientes, os seus concorrentes, o alcance de seu produto e os comportamentos de consumo são indispensáveis para que a organização seja competitiva e enfrente as futuras mudanças do mercado (SWIFT, 2001). São inúmeras as perguntas a serem respondidas antes de se abrir um negócio e ao longo da existência da organização.

Pesquisar reduz as incertezas, auxilia a alta direção e os chefes das organizações na tomada de decisões corretas e possibilita a análise da organização, do ambiente ou do negócio em questão (Opinião Consultoria, 2004). Porém muitas organizações não conseguem dar os passos iniciais para implantar o CRM. Consideramos que as Técnicas de Gerência de Projetos Ágeis podem ser muito úteis, sendo seus benefícios detalhados de forma apropriada no decorrer do estudo, e é dentro desta perspectiva que focaremos nosso trabalho.

Essa estratégia de negócios é identificada como a tecnologia alinhada com mudanças organizacionais e mudanças nos processos de negócio. Os investimentos devem ter objetivos claros, ou seja, metas que devem ser alcançadas, mudanças que devem ser efetuadas, entre outros. Tais objetivos são traçados de acordo com um entendimento correto da definição de CRM. Também com este trabalho procura-se esclarecer melhor estes assuntos.

7 Principais Conceitos Relacionados a CRM

7.1 Definição

Se formos procurar uma definição para CRM encontraremos várias, porém pode ser encarado como uma estratégia empresarial que tem como objetivo colocar em prática um conjunto de ações baseadas no marketing de relacionamento com os clientes. (LEITE, 2004). Outros autores indicam que essa estratégia visa selecionar e gerenciar clientes para otimizar o valor do negócio em longo prazo (LEE, 2000). Em complemento a isso, Ronald Swift diz que a definição mais útil está contida no próprio termo CRM. A palavra importante seria *relacionamento* (SWIFT, 2001). Organizações atualmente afirmam que mantêm relacionamentos com seus clientes, mas isto não existe de fato, pois todo o contato que têm é transacional. Pedidos de compra, preços acordados, contatos visando previsão de consumo, etc. Para que ocorra um relacionamento, as comunicações devem ser de mão dupla integradas, registradas e gerenciadas. Isto é importante, pois somente com dados históricos e detalhamento das transações e comunicações é que um relacionamento pode ser efetivamente mantido (SWIFT, 2001).

7.2 CRM como Filosofia e Não Metodologia

É bom enfatizar que CRM requer uma filosofia e cultura de negócios baseada no cliente que suporte um marketing eficaz, vendas e processos de serviço (LEE, 2000). Se esta abordagem for bem executada tornará seus clientes leais, pois os mesmos estarão com suas exigências satisfeitas ou até mesmo excedidas. Deve abranger a organização por completo, pois suas atividades de alguma forma sempre impactam no cliente. CRM é inseparável das demais atividades que as organizações realizam e todos assumem alguma responsabilidade na gerência dos relacionamentos com clientes. Como resultado da filosofia de CRM a organização preocupa-se em assegurar que o cliente tenha o que deseja da organização, estendendo os serviços em nível de personalização. Isso envolve a capacidade da organização para (SWIFT, 2001):

- Descobrir clientes;
- Conhecer os mesmos;
- Manter comunicação com eles;

- Assegurar que eles recebam o que desejam da organização – não somente quanto ao aspecto do produto, mas em cada detalhe de como a organização lida com eles;
- Verificar se eles recebem o que lhes foi prometido – desde que seja lucrativo;
- Assegurar que o cliente seja mantido, com objetivo de lucratividade em longo prazo.

7.3 A Importância da Personalização

A personalização identifica as necessidades e as exigências de clientes individuais em massa. O atendimento precisa parecer tão pessoal a ponto do cliente querer continuar com a organização. Os benefícios disso são a duração longa do convívio, os clientes felizes e a redução dos custos de serviços ao cliente, marketing e vendas (GREENBERG, 2001).

Uma organização deve ser capaz de identificar seus clientes de forma individual e única. Deve possibilitar a diferenciação dos clientes por suas necessidades e valor para a organização, criando mecanismos para facilitar a interação com os mesmos, estimulando a manutenção de uma relação de aprendizado; e personalizando o relacionamento da organização com o cliente. O objetivo é promover eficiência e lucratividade (PEPPERS & ROGERS GROUP, 2001).

7.4 A Satisfação do Cliente em Primeiro Lugar

Algo que precisa ficar esclarecido é que CRM não diz respeito a preços, ou envio de grandes quantidades de correspondência ou muitas ligações para clientes em potencial, mas sim trabalhar de forma a ter capacidade para gerar produtos, serviços, respostas e satisfação do cliente.

Um cliente satisfeito representa a possibilidade do aumento de sua interação com a organização e crescimento de sua fidelidade. Também é a melhor forma de fazer marketing, pois mostrará para todos a sua satisfação, atraindo novos clientes, parcerias e possibilidades. Além de ser uma das melhores maneiras de realizar a propaganda da organização, é também muito mais barato manter um cliente satisfeito do que investir na aquisição de novos clientes. Segundo um estudo publicado na revista *The Industry Standard*, apresentado por Paul Greenberg em seu livro *CRM na velocidade da Luz*, custa cinco vezes mais conquistar um novo cliente do que mantê-lo.

7.5 Adoção de Estratégias para CRM

A raiz da centralização no cliente é permitir que o mesmo auxilie na condução das tarefas e dos processos existentes. É uma parceira constante que deixa de existir a partir do momento que a organização perde o interesse pelo seu cliente, ou então o cliente encontra uma organização que o satisfaça mais adequadamente.

Uma das maneiras da organização demonstrar interesse pelo cliente é por adotar estratégias voltadas para os mesmos. Este é o primeiro passo, é como o CRM começa. Assim sendo, se o CRM começa através da adoção de estratégias voltadas para os clientes, inicia no topo da organização, junto ao seu gerenciamento executivo. Faz-se necessária a existência de um patrocinador, algum colaborador pertencente ao gerenciamento executivo que possa ajudar a identificar os recursos e reduzir a resistência dos empregados, além de poder assegurar que o sistema sobreviva. (BROWN, 2001).

A maioria das organizações inicia sua organização focando em sua estrutura interna – não nos clientes – tornando o processo de centralização no cliente uma tarefa árdua. Pode levar à redefinição de papéis funcionais, demanda reengenharia de processos e é suportada, não conduzida, por uma tecnologia CRM (LEE, 2000).

7.6 CRM não é Tecnologia

O emprego da tecnologia logicamente pode ser proveitoso, mas o que se salienta é que o sucesso do CRM depende mais da estratégia do que o investimento realizado em tecnologia (FEI, 2001). Sua aplicação não pode estar vinculada somente à tecnologia. Sendo considerada como estratégia, ações relativas ao papel das pessoas e a avaliação dos processos organizacionais relativos ao cliente são conceitos básicos e, em alguns casos, independentes de tecnologia (LEITE, 2004).

7.7 Comunicando-se com o Cliente

Os autores também enfatizam a importância da comunicação com o cliente e seu auxílio na condução das tarefas. Uma publicação da *Symnetics Business Transformation* sugere que essa comunicação caracteriza-se por um ciclo de três fases: Conhecimento, Estratégia e Ação (SYMNETICS, 2000), onde:

- Conhecimento é a fase que a organização através da reunião das informações obtidas dos contatos com os clientes passa a entender suas necessidades e conhecer suas peculiaridades, possibilitando a segmentação em níveis bastante detalhados.
- Estratégia é a fase onde a partir dos resultados obtidos na fase anterior, determinam-se estratégias refinadas e assertivas junto aos diferentes segmentos de mercado.
- Ação, como terceira fase, parte da implementação das estratégias definidas, monitoração dos resultados e ampliação do conhecimento sobre os clientes através da análise dos resultados das ações.

Esse ciclo se repete continuamente, buscando o aperfeiçoamento constante e atingir as metas da organização (SYMNETICS, 2000).

8 Motivações para a Implantação de CRM

São muitas as motivações para a implantação de um Gerenciamento de Relacionamento com o Cliente. Quando as organizações implantam o CRM objetivam alcançar quatro dos principais desafios organizacionais (BRETZKE, 2002):

8.1 Encantar o Cliente Externo

Gerenciar o relacionamento com os clientes em tempo real propicia um aumento na produtividade e precisão da organização. As promessas de atendimento realizadas quando são satisfeitas, é agregado muito mais valor ao produto ou serviço oferecido, podendo garantir uma experiência de marca “*caring and daring*”¹, criando uma comunidade que efetivamente acredita no produto ou serviço de uma organização(BRETZKE, 2002). Dessa maneira, a comunidade que acredita neste produto ou serviço torna-se propaganda viva da organização.

8.2 Encantar o Cliente Interno

A mudança nos processos gerenciais facilita a aceitação da filosofia CRM ao longo de toda a organização. A prática de atividades de marketing e CRM é

¹ Expressão usada por Stan Rapp, um dos principais autores sobre Marketing Direto, para designar estratégias de Marketing que cuidam do cliente ao mesmo tempo em que são ousadas, ou melhor, inovam e não medem esforços para encantar o cliente.

viabilizada através da tecnologia utilizada como base para obtenção de informações dos clientes em tempo real. Obtém-se então, valor agregado a cada interação com o cliente, produtos customizados são desenvolvidos e presta-se atendimento personalizado. A estratégia de CRM proporciona a oportunidade de examinar sua organização por intermédio dos olhos dos clientes, proporcionando valor agregado no relacionamento (BRETZKE, 2002). Dessa maneira, a organização está cuidando não somente do cliente em si, mas de sua estrutura como um todo, utilizando-se das informações obtidas dos seus clientes para melhora contínua nos serviços prestados e produtos vendidos.

Diante de tal cenário, ocorre uma redução dos problemas interdepartamentais, pois a organização passa a unir-se em um único objetivo comum: a satisfação do cliente. Assim sendo, todos os departamentos sentem-se apoiados uns nos outros nessa missão, aumentando as motivações e gerando altos níveis de satisfação entre os colaboradores.

8.3 Melhorar o Desempenho Financeiro

O relacionamento propicia que a organização conheça o cliente. Isto muitas vezes se traduz em melhora no desempenho financeiro. Este fato se explica pois para uma melhora do desempenho financeiro, todas as oportunidades de vendas devem ser aproveitadas. Isto ocorrendo, a mão de obra torna-se mais produtiva, pois os colaboradores estão motivados, os investimentos alcançam o retorno esperado no tempo definido, a previsão de vendas torna-se mais realista e confiável, os estoques trabalham dentro de um limite apropriado, as compras são efetuadas eficientemente e a produção é otimizada, reduzindo desperdício, perdas e custos. Além disso, podendo desenvolver uma comunicação dirigida para conquistar novos clientes e fidelizar os atuais com ofertas que realmente atendem os clientes e diferenciam a organização, verifica-se que o próprio custo da comunicação e do trabalho da força de vendas será reduzido permitindo maior retorno do investimento.

8.4 Aumentar a Fidelidade e Satisfação do Cliente

Clientes satisfeitos são a alma de qualquer negócio bem-sucedido e clientes fiéis são essenciais para a sustentação da liderança no negócio (JONES, 2002). Uma questão a ser respondida é: como uma organização evolui de clientes simplesmente satisfeitos para clientes fiéis? Inúmeras organizações acreditam que apenas fornecer uma melhoria constante em seus serviços e produtos seja suficiente para manter seus

clientes satisfeitos e fiéis. Outras organizações vão um pouco mais além, estabelecendo programas de fidelidade que propiciam descontos e benefícios para clientes frequentes. Mas, em longo prazo, nenhuma dessas políticas aumentam significativamente a fidelidade do cliente. O motivo para isto é que tais políticas não estão centradas no cliente individualmente, mas sim no cliente como parte integrante de uma grande massa sem diferenciação.

Através de um Gerenciamento de Relacionamento com o Cliente, o valor do cliente para a organização é expandido, com base na sua definição individual de valor. Existem algumas lições importantes sobre a fidelização de clientes (JONES, 2002):

- Um bom produto ou serviço será sempre a base para o desenvolvimento da fidelidade do cliente. Sem ele, nenhum programa para fortalecer o relacionamento de uma organização com seus clientes será bem-sucedido.
- As organizações devem desenvolver produtos e serviços personalizados de acordo com o que os clientes individuais querem e não de acordo com o que as organizações estão atualmente preparadas para oferecer ou de acordo com aquilo que *acham* que os clientes querem.
- Desenvolver a fidelidade do cliente deve ser um compromisso de toda a organização. Todas as atividades da organização devem servir ao objetivo de criar valor para o cliente.
- Reunir informações sobre os clientes não é tudo. Por mais sofisticada que seja, a tecnologia da informação é simplesmente um instrumento para melhorar a qualidade do produto e do serviço para atender às necessidades individuais dos clientes.
- O impacto econômico do foco de uma organização na fidelidade do cliente torna-se evidente apenas após um longo período de tempo. Portanto, a avaliação em longo prazo do comportamento do cliente é essencial para se compreender o retorno sobre o investimento em iniciativas de fidelização. Sem avaliação, uma organização pode perder prematuramente o seu foco na fidelidade do cliente.
- Nunca perca de vista as atividades dos seus concorrentes.

9 A Contribuição da Agilidade para a Implantação de CRM

9.1 O que significa Agilidade

A definição de agilidade segundo Jim Highsmith (HIGHSMITH, 2002, prefácio, p. xxiii) é “a habilidade tanto de criar quanto de adequar-se a mudanças de uma forma que incremente a eficiência em um ambiente de trabalho turbulento”. Organizações destacam-se quando conseguem ser melhor que seus concorrentes ao responder a mudanças que se apresentam necessárias. Sendo assim, são capazes de mudar de direção rapidamente e são flexíveis, significando a capacidade de ver que as coisas como têm sido feitas podem não funcionar tão bem dali para frente. Organizações ágeis também entendem que o equilíbrio entre ordem e caos determina o sucesso (HIGHSMITH, 2002, prefácio, p. xxiv).

9.2 O Manifesto Ágil

Em novembro de 2001, um grupo de dezessete pessoas, referências mundiais em desenvolvimento de software se reuniram e surgiu então o Manifesto Ágil, que define os seguintes princípios (AGILE MANIFESTO, 2001):

- Indivíduos e interações são mais importantes que processos e ferramentas.
- Software funcionando é mais importante do que documentação completa e detalhada.
- Colaboração com o cliente é mais importante do que negociação de contratos.
- Adaptação a mudanças é mais importante do que seguir o plano inicial.

Apesar do Manifesto Ágil ser diretamente direcionado ao desenvolvimento de software, Gary Chin (2004) trata sobre a avaliação de quando a gerência ágil de projetos deve ser considerada. De acordo com Leite (2004, p. 149), como os projetos de implantação de estratégias para CRM envolvem primordialmente o desenvolvimento ou aprimoramento de novos processos, sob o possível suporte tecnológico, podemos considerar que sua aplicabilidade está bem posicionada, fundamentando a utilização do Manifesto Ágil no presente trabalho.

Podemos encarar o manifesto ágil como uma inquietação motivada por (HILMER, 2004):

- Reação às metodologias pesadas.
- Menos orientação ao documento mais orientação ao código.
- Métodos claros e adaptáveis.
- Processos ágeis: minimamente carregados com atividades que consomem e não produzem nenhum ganho visível.

É importante entendermos que metodologias ágeis (Kon e Goldman, 2003)

- É uma atitude, e não um processo prescritivo.
- É um suplemento aos métodos existentes, ele não é uma metodologia completa.
- É uma forma efetiva de se trabalhar em conjunto para atingir as necessidades das partes interessadas no projeto.
- É algo que funciona na prática, não é teoria acadêmica.
- Não é um ataque à documentação, pelo contrário aconselha a criação de documentos que tem valor.
- Utilizam o modelo iterativo e incremental.
- O cliente é faz parte da equipe de desenvolvimento.

Os pontos focais do desenvolvimento ágil são as pessoas, relacionamentos e incertezas (HIGHSMITH, 2002). O ambiente ágil de desenvolvimento de software precisa ter princípios e valores colaborativos, uma perspectiva caótica e uma metodologia flexível que seja eficaz, não esbarrando na burocracia.

Alguns acreditam que ser ágil significa menor quantidade de processos, documentos resumidos e menos formalidades, mas a perspectiva correta é bem mais ampla. Mesmo que menor quantidade de processos e menos cerimônias possam diminuir os custos de desenvolvimento, estes não são suficientes para produzir agilidade. Por focar nas pessoas e suas interações, dando poderes individuais para realizar decisões rápidas e adaptar seus próprios processos, são a chave para um ambiente ágil (HIGHSMITH, 2002).

Sendo assim, esse ambiente organizacional privilegia as interações dinâmicas dos indivíduos e equipes ao invés de procurar seguir padronizações estáticas.

9.2.1 FDD

Várias técnicas foram elaboradas como modelo para que os conceitos supracitados fossem seguidos. Dentre eles destacamos o FDD (*Feature Driven Development*), ou simplesmente Desenvolvimento Dirigido a Funcionalidades. Esta é uma metodologia ágil criada por Jeff de Luca e Peter Code, reconhecida em 1997. Seu desenvolvimento consiste em dois estágios principais (NERI, 2005):

- Descobrir uma lista de *features* a serem implementadas
- Implementar baseado nas *features*

Entende-se *feature* como sendo uma funcionalidade, propriedade, comportamento ou característica que faz parte do projeto em questão. O estágio do descobrimento das *features* é o processo mais crítico, pois a qualidade com a qual se identifica essa lista define quão preciso o projeto será controlado, quão extensível e manutenibilidade que o código terá. (NERI, 2005). Requer a participação integral do cliente junto à equipe de desenvolvimento.

Com a identificação das *features* deriva-se um diagrama de classes de análise com suas responsabilidades bem definidas. Essas classes auxiliam os desenvolvedores e usuários, pois procura ter um linguajar comum. É importante que as funcionalidades sejam pequenas o suficiente para que o seu desenvolvimento seja feito em pequenas iterações (NERI, 2005).

A implementação começa através do agrupamento de um conjunto de *features* relacionadas a um mesmo assunto. Pode-se chamar esse agrupamento de pacote de trabalho. Esse pacote desenvolvido deve apresentar uma parte do software onde o usuário poderá utilizá-lo. Cada iteração inclui (NERI, 2005):

- *Kickoff meeting* (reunião introdutória) para um pacote de trabalho, onde detalhes das *features* devem ser esclarecidos;
- Projeto, onde classes, métodos e documentação necessárias são criados;
- Revisão de Projeto, onde um projetista experiente faz a avaliação e o aceita ou rejeita;
- Desenvolvimento, onde são feitos a implementação e testes de unidades de trabalho definidas nas *features*;
- Revisão de Código, onde se pode utilizar a técnica de programação em pares. Técnica onde dois programadores realizam o serviço em

conjunto, em um mesmo computador, atentos ao mesmo trecho de código;

- Fechamento, onde as *features* são liberadas em um “*build*” (versão funcional do aplicativo).

Abaixo é apresentada a figura 1 que ilustra o processo de forma gráfica.

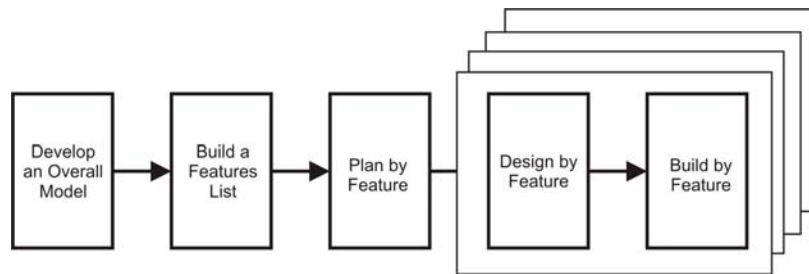


Figura 1 – FDD

9.3 Agilidade em CRM

Implantar CRM é muitas vezes trabalhoso e, como é uma estratégia que possui vários conceitos, demanda esforço. Mesmo assim é interessante que esse processo se dê de forma ágil, pois dessa forma a competitividade e as vantagens que CRM possibilita são trazidas à tona mais rapidamente.

Isto está de acordo com Jim Highsmith, que em seu livro “Gerenciamento de Projetos Ágeis” afirma que não é somente o desenvolvimento de novos produtos que está exigindo redução no tempo de cronogramas, mas também as práticas de negócios como implementação do CRM (HIGHSMITH, 2004 p.3).

O mesmo autor, porém, acautela que a agilidade nada tem a ver com correria. Cita-se a frase “seja rápido, mas não corra”. Em outras palavras, faça as coisas certas, aprendendo como fazê-las de forma rápida, ou ágil. Ele caracteriza agilidade como a habilidade de balancear flexibilidade e estabilidade (HIGHSMITH, 2004 p.16), e é o que faz a diferença frente ao gerenciamento de projetos tradicional, que muitas vezes definem o contexto de forma demasiada, planejando os projetos em um grau de detalhamento muito alto, deixando pouco espaço para a agilidade fluir.

Sendo assim, pode-se afirmar que duas ações devem ocorrer para ter sucesso com CRM: planejar uma solução abrangente e impulsionar esta solução, tornando-a contínua (PEPPERS & ROGERS GROUP, 2001). Mesmo sem desconsiderar essas

premissas é necessário pensar em como tornar esse processo ágil, para obter ganhos da forma mais rápida possível. A agilidade traz motivação e aporte de fundos para a continuação do CRM (PEPPERS & ROGERS GROUP, 2001). Mas conseguir maximizar os retornos em curto prazo, para a implantação ganhar força é desafiador. Uma proposta para isso é que a organização passe por um processo de implantação de CRM em ciclos de desenvolvimento (PEPPERS & ROGERS GROUP, 2001). Entre os ciclos é essencial avaliar os resultados, determinar as novas estratégias e impulsionar o CRM.

Segundo Leite (LEITE, 2004) a implantação de CRM deve ser composta por um conjunto de ações planejadas, executadas e avaliadas e deve ter a participação de todas as áreas envolvidas com o cliente: marketing, vendas, atendimento e tecnologia. Este cenário leva ao despertar para uma visão holística das organizações para que elas possam manter relacionamentos suportados por processos operacionais ágeis e com tecnologia de suporte adequada (LEITE, 2004).

10 Implantação de CRM

A implantação de CRM está associada a um conjunto de práticas que envolvem vários elementos dentro de uma organização. Dentre estes elementos destacam-se as estratégias, as pessoas, os processos e a tecnologia, sendo estes considerados seus principais pilares de sustentação (LEITE, 2004). Como esses elementos se relacionam é o objetivo deste tópico.

O primeiro passo após a decisão de implantação do CRM em uma organização é a elaboração de um planejamento. Além disto, os processos internos precisam estar estruturados para a nova realidade (SYMNETICS, 2000) e, como o CRM é uma mudança de atitude, as pessoas devem estar envolvidas no processo.

A iniciativa de CRM também deve estar alinhada à estratégia de negócio da organização. A experiência revela que muitas das falhas para realizar o potencial de benefícios em projetos de CRM advêm principalmente da desvinculação do projeto das questões estratégicas do negócio. Uma das maneiras de averiguar isto é criar um mapa de inter-relacionamento dos objetivos estratégicos (SYMNETICS, 2000). Com esse mapa elaborado a organização é capaz de identificar quais processos reestruturar para que o CRM funcione como facilitador de suas estratégias. Deve-se dar ênfase principal nos processos relacionais como, por exemplo, Marketing, Vendas, e Atendimento, pois lidam diretamente com o cliente externo.

10.1 Etapas de Implantação de CRM

Sobre as etapas de implantação que CRM alguns autores as definem sob o aspecto da gerência de projetos. A figura 1, abaixo, apresenta a representação gráfica do modelo do processo de implantação de CRM utilizado pela Peppers & Rogers.

Ela é muito clara sobre o que as empresas devem realizar para colocar em prática o marketing individualizado, porém algumas etapas de análise do negócio e de estratégias para CRM não são contempladas (LEITE, 2004). Ela não considera, por exemplo, aspectos diretamente relacionados à implantação de CRM tais como avaliação de custos e de riscos.

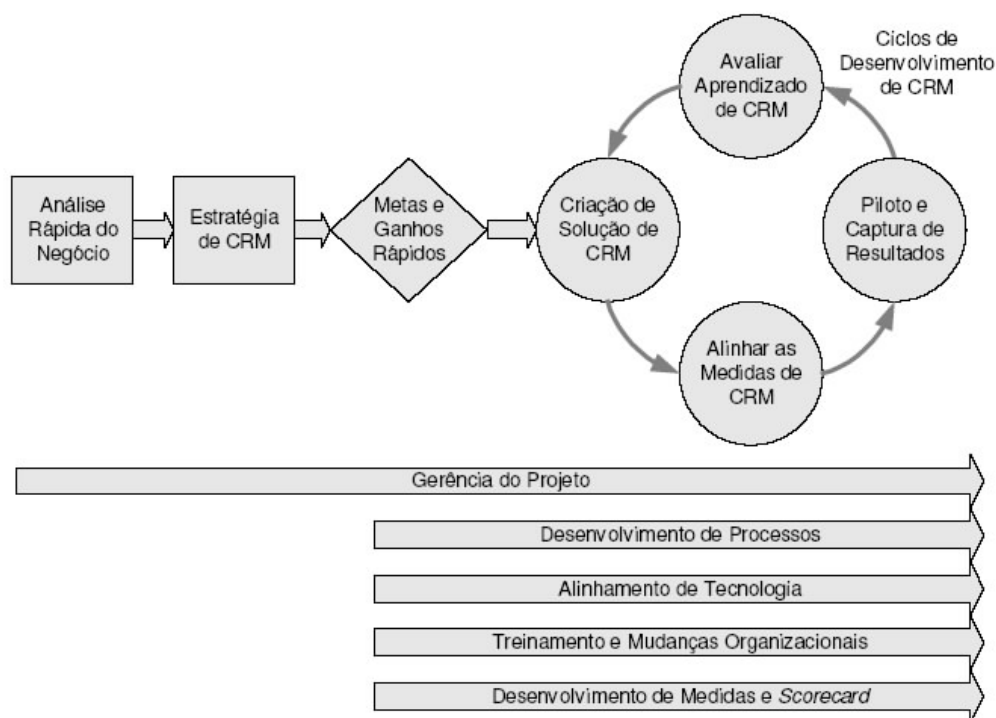


Figura 1 – O processo de Implantação de CRM proposto por Peppers and Rogers.

Um ponto que vale ressaltar tendo em vista o objetivo deste trabalho é a importância que se deve dar à fase de formação de pessoal, descrita na figura acima como Treinamento e Mudanças organizacionais.

10.2 Implantação com Tecnologia

Nesse momento de implantação pode ser interessante a avaliação de ferramenta de tecnologia que auxilie no processo. Vale a pena também estabelecer planos de ação detalhados e definir recompensas (SYMNETICS, 2000).

Seja com o auxílio de uma ferramenta de tecnologia ou não, fica claro que as habilidades tecnológicas estão mudando mais rápido do que o conjunto de habilidades da maioria dos empregados e principalmente com a cultura da maioria das companhias. Considere-se a rapidez que a tecnologia e os modelos de gerenciamento estão avançando, e se terá uma dica da velocidade que as organizações precisarão mudar nos próximos meses e anos (PEPPERS & ROGERS GROUP, 2003). Nesse contexto chegamos às pessoas.

10.2 Pessoas e sua Importância na Implantação do CRM

Uma organização deve estar apta a acompanhar as mudanças tecnológicas trazidas pelo mercado. A fim de acompanhar tais mudanças, deve estar preparada para formar seus colaboradores e formá-los constantemente. É neste ponto priorizam-se as pessoas que podem realmente fazer diferença no processo de implantação de estratégias para CRM. As pessoas podem ser analisadas em dois aspectos: o primeiro deles está relacionado à excelência no atendimento e a segunda à compreensão das responsabilidades individuais no processo (LEITE, 2004).

A excelência no atendimento advém de não apenas atender os clientes com eficiência, utilizando a tecnologia, mas também atender com simpatia, cortesia e eficácia. E para isso os colaboradores devem estar preparados, cientes dessa responsabilidade. Uma premissa básica para CRM é que é necessário que existam relacionamentos com os clientes para que eles possam ser gerenciados e que relacionamentos são mantidos por pessoas, ainda que estas representem organizações. Estratégias competitivas definidas pela alta administração de qualquer organização são colocadas em prática por pessoas, e relacionamentos, em última instância, são mantidos por pessoas. A tecnologia facilita em muito este trabalho, mas pode não ser funcional sem um pessoal consciente e motivado (LEITE, 2004). Assim, a preparação de pessoal, através da formação e aprimoramento, é requisito indispensável para a aplicação bem sucedida de novas estratégias. Questiona-se então agora como deve ser feita a preparação desses agentes da organização.

10.3 Fatores Críticos de Sucesso do Processo de Formação

O processo de formação dos colaboradores de uma organização deve ser realizado de forma a assegurar o seu sucesso. Deve justificar o investimento realizado por parte da organização. Este processo exige uma série de investimentos, dentre os quais podemos destacar investimentos financeiros e de disponibilidade de tempo, de tal forma que a organização necessita da seguridade de que o processo alcançará um bom resultado, se possível acima do esperado. Serão tratados brevemente os fatores críticos de sucesso (NEKI Technologies, 2004) de um processo de formação para que este esteja mais próximo de ser alcançado.

10.3.1 Formação e Disponibilidade de Pessoal

Uma das primeiras questões a serem consideradas é a provável impossibilidade de colocar todo o pessoal em processo de formação. Considere-se uma organização com um número significativo de colaboradores que resolva realizar um processo de formação intensivo que ocupe uma semana inteira. Parte dos colaboradores deve ser mantida fora do processo, assegurando o funcionamento do legado existente. Ao mesmo tempo em que tal situação se caracteriza, outra em paralelo ocorre: aqueles que não estão participando têm a necessidade e o interesse de alguma forma envolver-se no processo do qual seus colegas estão participando. Assim sendo, é necessário manter os colaboradores a par do que está sendo apresentado aos outros no processo de formação. A organização deve indicar como serão trazidos para o processo e qual o papel que cada um possui neste novo cenário. Incentivo e disponibilização do material utilizado devem fazer parte da política adotada pela organização, demonstrando preocupação com o bem estar e constante atualização de sua equipe. Caso a organização não desempenhe tal papel junto aos seus colaboradores, corre o risco de perdê-los, estando fadada ao remanejamento de pessoal para garantir suporte às suas tarefas e processos.

10.3.2 Processo de Formação Constante

Outro fator crítico para o sucesso de um processo de formação é torná-lo um hábito e ser realizado periodicamente. O processo deve ser realizado não somente pelo

peçoal de nível gerencial, mas por todo o peçoal da organização. As equipes devem trabalhar conjuntamente como times, significando que cada uma delas deve ter um conhecimento básico comum, objetivando sempre alcançar conhecimentos mais aprimorados.

10.3.3 Evolução do Processo de Formação e suas Mudanças

Uma organização que deseja centralizar-se no cliente, preocupada com o processo de formação de seus colaboradores, deve estar ciente das constantes mudanças e das conseqüências trazidas por elas. As formas de educar são muitas. A organização deve selecionar aquela que melhor se adapta à sua realidade, procurando sempre estar informada sobre melhorias no processo de formação e educação.

10.3.4 As Conseqüências do Processo de Formação

São muitos os motivos que levam um colaborador a deixar o seu ambiente de trabalho e lançar-se ao mercado do em busca de novos horizontes. Salário incompatível com a função, ambiente inadequado, colegas de trabalho dos quais é impossível gostar, enfim, a lista poderia estender-se. Qualquer organização que resolve re-educar sua equipe deve estar preparada para pagar aquilo que os colaboradores merecerão após o término do processo de formação.

10.3.5 Os Benefícios do Processo de Formação e o Desejo de Aprender

O processo de formação deve estar focado no que é bom para o colaborador, tanto em nível organizacional quanto em nível individual. Todos deverão estar cientes dos benefícios trazidos pelo processo e a organização deve incentivar sua equipe a aprender.

As mudanças existentes no mercado são assustadoras e constantes, fazendo com que o sentimento de um processo de formação contínuo necessite ser cultivado dentro da organização e junto aos seus colaboradores. Todos devem ser incentivados a sempre ler livros, realizar cursos, dinâmicas, o que for possível para atualizar suas técnicas e experiências.

10.3.6 O Tempo

Os processos de formação devem ser fornecidos no momento em que são necessários, não meses antes ou depois. As pessoas esquecem aquilo que aprenderam um mês após terem o feito, a não ser que apliquem imediatamente após o processo o que lhes foi transmitido. Em contrapartida, quando um dado processo de formação demora ser realizado a produtividade e a qualidade do produto ou serviço são comprometidas.

10.3.7 Igualdade para Todos

Colaboradores experientes tendem a tentar se convencer após um certo tempo que o processo de formação não mais é necessário, desmotivando colegas de equipe e até mesmo a si próprios. O desejo de aprender mais uma vez deve ser cultivado, eliminando a frase “Eu já fiz isto antes” do vocabulário de todos os membros da equipe.

10.3.8 Educação e Processo de Formação Caminham Juntos

O processo de formação oferece aos colaboradores a técnica para realizar seu trabalho e a educação fornece conhecimento para entender seu trabalho. A contribuição mais importante que um programa educacional pode fazer é explicar a inter-relação dos conceitos e técnicas (NEKI Technologies, 2004).

10.4 Preparando os Colaboradores para o Gerenciamento de Relacionamento com o Cliente

O processo de formação não é um custo de negócio, mas sim um investimento que deve ter um retorno positivo (MONGER, 2005). Para tanto, aliada aos fatores críticos de sucesso de um processo de formação, deve existir uma preparação dos colaboradores para o direcionamento ao cliente.

A seguir são apresentadas cinco etapas específicas para preparar a organização para competir efetivamente na era da competição direcionada ao cliente (PEPPERS & ROGERS GROUP, 2004).

10.4.1 Ensinar Colaboradores a Gerenciar Informações

O processo de formação e o aprendizado são distintos. Aprendizado é o que é feito durante a vida. O processo de formação é algo que é realizado a fim de tornar-se proficiente numa determinada habilidade ou no trabalho.

Gerenciar, organizar, compartilhar e acessar informações são habilidades críticas, assim como as habilidades de ler e escrever. Poucos são os colaboradores que possuem tais habilidades estratégicas para um bom relacionamento entre a organização e o cliente.

O processo de formação provido aos colaboradores reflete diretamente o desejo da organização. Se a meta é assegurar que novos colaboradores saibam lidar com ligações em um *call center*, por exemplo, demonstrar o manual de treinamento não será suficiente. Os colaboradores necessitarão de experiências trabalhando com pessoas no papel de cliente, treinamento durante ligações para clientes e análises de ligações gravadas. Dessa forma, a informação compartilhada tende a ser gerenciada de forma mais adequada.

Quando Walter Wriston era o CEO do Citibank, previu que dados sobre dinheiro se tornariam tão valioso quanto dinheiro (PEPPERS & ROGERS GROUP, 2004). Dados são elementos essenciais em qualquer estratégia de negócio. Não há outro modo para uma companhia compartilhar informações sobre seus clientes de forma mais eficiente e ágil do que através de um Banco de Dados, sendo este tão precioso quanto a qualidade das informações que ele contém.

Assim sendo, as informações contidas no Banco de Dados e o Banco de Dados em si, aliam-se à capacidade do colaborador de gerenciar, organizar, compartilhar e acessar tais informações, tornando o colaborador um diferencial extremamente competitivo para a organização.

10.4.2 Obter informações sobre os clientes

Kotler define o marketing como sendo uma atividade humana dirigida para a satisfação das necessidades e desejos dos consumidores, através de processos de troca (KOTLER, 1972). Levitt prega que marketing nada mais é do que um processo de atrair e manter clientes (LEVITT, 1974). Independente da definição de marketing, o mesmo evoluiu. Seu foco primeiramente era na produção; posteriormente, no produto; em seguida, na distribuição; após isso, era focado na divulgação (publicidade e propaganda,

marcas, promoções); e, finalmente, o foco passou a ser o cliente. O mercado, apesar de ter a consciência de que o foco é no cliente, possui muitas organizações que falam para o cliente, e não com o cliente. Cria-se então uma ilusão de estar satisfazendo o foco principal, quando na verdade os olhares estão centrados na divulgação do produto ou serviço, tendo que contar excessivamente com mensagens publicitárias.

Há ainda bastante espaço para mensagens publicitárias, mas não mais espaço que detectar a necessidade do cliente, mostrando que tal necessidade existe, e, por fim, criar o desejo de satisfazer a necessidade existente. É necessária a compreensão das necessidades únicas de cada cliente. Vantagem competitiva pode ser em grande parte adquirida através do conhecimento que as organizações possuem sobre seus clientes, conhecimentos estes que as organizações concorrentes não têm. Como tal conhecimento só é obtido através do relacionamento da organização com o cliente, é necessário incentivar os clientes para que tenham vontade de relacionar-se com a organização (PEPPERS & ROGERS GROUP, 2004).

Uma boa estratégia é pregar junto aos colaboradores um princípio simples, porém importante: lembrar-se das informações para seus clientes, não sobre eles. Pergunta-se se um cliente gostaria que uma certa informação fosse lembrada, podendo esta ser utilizada para beneficiar o relacionamento entre as partes. Se a resposta à pergunta é uma negativa, a informação não deve ser utilizada (PEPPERS & ROGERS GROUP, 2004).

10.4.3 O processo decisório ao longo da organização

CRM pode ocorrer independentemente do uso de tecnologia, porém dada a sua utilização, que logicamente tem potencial de beneficiar muito na estratégia da organização, podem ocorrer necessidades específicas de capacitação.

Quando é falado sobre Tecnologia da Informação e Marketing, estão sendo tratados assuntos que afetam diretamente os clientes e seus relacionamentos com a organização, ou seja, tomadas de decisão que visam atingir o cliente e obter retorno para a organização, seja este retorno financeiro ou não.

As tomadas de decisão devem ser uma preocupação constante das organizações, pois grande parte delas influencia diretamente o cliente. Qualquer escolha que se faça em uma organização implica diretamente em uma tomada de decisão. É um processo que visa solucionar determinado problema. Pode-se definir problema como sendo uma situação onde o estado atual é diferente do estado desejado (OLIVEIRA,

2004). A resposta para o problema pode ser obtida através da procura e seleção de uma solução ou ação que criará valor, podendo o problema ser constituído de diversas naturezas (RASKIN, 2003), como:

- Busca de melhores recursos;
- Decidir como fornecer um serviço
- Saber como lidar com o mercado e suas constantes mudanças.

10.4.3.1 Tipos de Decisão

Existem dois tipos básicos de decisão: as programadas e as não-programadas.

As decisões programadas caracterizam-se por serem repetitivas e rotineiras. Fornecem estabilidade, aumento de eficiência e redução de custos, pois são tomadas de acordo com políticas, procedimentos ou regras previamente definidos.

As decisões não-programadas são novas e desestruturadas. Suas soluções são encontradas à medida que os problemas aparecem, requerendo atividades de pesquisa para encontrar a solução. “Se um problema não surge com frequência suficiente para ser resolvido por uma política, ele deve ser resolvido por uma decisão não-programada” (OLIVEIRA, 2004).

A criação de uma estratégia organizacional requer decisões não programadas para encontrar a melhor forma de criar valor, através de habilidades e recursos não necessariamente existentes. Assim sendo, intuição e criatividade devem fazer parte dos colaboradores de uma organização a fim de encontrar a melhor solução.

10.4.3.2 Modelos de Tomada de Decisão

A princípio, a decisão era vista como um processo racional, onde as decisões faziam com que as organizações se ajustassem perfeitamente ao seu ambiente (RASKIN, 2003). Atualmente, reconhece-se que a decisão é um processo incerto onde a busca por soluções pode ou não ser favorável.

O Modelo Racional define que o processo de tomada de decisão possui três estágios:

- Identificação e definição do problema;
- Geração de alternativas de soluções;
- Seleção e implementação da solução.

As situações seriam “ideais”, sem incertezas e ambigüidades, e os custos de gestão e de informação seriam ignorados (RASKIN, 2003). Assim sendo, tal modelo mostra-se irrealista e simplista, pois assume que todas as informações necessárias para a tomada de decisão são conhecidas.

Os modelos modernos fornecem um panorama mais realista do processo.

O Modelo Carnegie visualiza a organização como um conjunto de diferentes interesses. A tomada de decisão ocorre através de compromissos e negociações entre colaboradores de diferentes funções e áreas. O modelo reconhece que o processo não é único e com regras objetivas, mas sim um processo aonde regras vão sendo estabelecidas de acordo com objetivos e interesses. O ambiente é incerto e as informações são incompletas. É racional no ponto onde os colaboradores agem intencionalmente para encontrar a melhor solução para alcançar seus objetivos.

O Modelo Incrementalista possui os colaboradores realizando uma seleção de cursos de ação de forma incremental, diferentes daqueles já utilizados, e vão corrigindo ou evitando erros através de sucessivas mudanças incrementais, que podem levar a um curso de ação completamente novo (RASKIN, 2003). Durante esse processo, os objetivos organizacionais e a forma de alcançá-los podem sofrer modificações, mas de forma gradativa, permitindo que ações corretivas possam ser tomadas.

O Modelo Desestruturado desenvolvido por Mintzberg, descreve como é o processo de tomada de decisão quando o nível de incerteza é alto. Reconhece sua natureza incremental, pois acontece em pequenos passos que em conjunto levam a uma decisão. Ao encontrarem-se obstáculos, as alternativas são repensadas, tornando o processo não linear. O processo desenvolve-se de forma desestruturada e não previsível. Tenta explicar as decisões não programadas, enquanto o Modelo Incrementalista tenta explicar a melhoria das decisões programadas ao longo do tempo.

O Modelo da Lata de Lixo leva ao extremo a visão desestruturada do processo de tomada de decisão. De acordo com ele, os responsáveis por tomar decisões deveriam propor soluções a problemas que não existem, mas que iriam surgir e seriam resolvidos com as soluções disponíveis.

Como a realidade enfrentada pelas organizações está distante do processo definido no Modelo Racional, aproximando-se mais para lidarem com as incertezas do ambiente, estão constantemente tomando decisões de maneira não programadas, desestruturadas e como uma lata-de-lixo (RASKIN, 2003).

10.4.4 Compartilhamento de Informações na Organização

São inúmeras as organizações atualmente que afirmam ter pleno conhecimento sobre seus clientes e suas necessidades. Realizando uma análise mais cuidadosa, teremos uma realidade bem diferente.

Citemos o caso de uma organização farmacêutica, que possui vários representantes junto à classe médica. Tais representantes realizam visitas periódicas para demonstração de novos medicamentos, apresentação de informações sobre mudanças realizadas em medicamentos existentes e fornecimento de amostras gratuitas. Caso tal representante saia da organização ou fique incapacitado de realizar as visitas, pode ser encontrada a situação em que o representante substituto não possui informação alguma acerca dos médicos a serem visitados. Ele não sabe o horário em que a visitação é mais adequada, desconhece quais medicamentos o médico prefere trabalhar, e, por fim, o mais importante – não faz idéia de como o médico gosta de ser tratado e quais informações são interessantes para ele serem lembradas. Assim sendo, as informações importantes sobre a tarefa a ser desempenhada centram-se no representante, sendo que este leva consigo as ferramentas necessárias para tornar o trabalho satisfatório.

Outro caso que pode ocorrer é um cliente prover informações que podem gerar rentabilidade para a organização, mas tais informações não são repassadas para a pessoa certa. Existem muitos casos onde vendedores não compartilham casos de sucesso, e até de fracasso, com outros vendedores, por terem receio de perder seu lugar de destaque, ou por medo de ter sua renda pessoal prejudicada, e, não esquecendo da situação “eu posso ser demitido caso ele tenha um desempenho melhor do que o meu”. Softwares de gerenciamento de contato são instalados, mas no local reservado para as particularidades que tornam o cliente singular, encontram-se espaços vazios, desprovidos das informações tão necessárias para um bom relacionamento entre a organização e o cliente.

Independente da situação, a organização precisa medir e recompensar atitudes e comportamentos que deseja presenciar dentro do ambiente de trabalho. No caso de uma central de atendimento ao cliente, ao invés de medir a eficiência somente através da média de duração do atendimento, tratar aspectos quanto à qualidade do atendimento em si, ou seja, medir a quantidade de problemas resolvidos na primeira chamada, ou ainda o número de novos *prospects*² que os colaboradores geram. Outra

² Pessoa não-consumidora de um determinado produto ou serviço que tem potencial de vir a se tornar um consumidor, se devidamente motivado. Também chamado Cliente Potencial.

forma de incentivo é a definição de equipes. Tais equipes recebem uma meta periódica, e caso alcancem tal meta, recebem uma premiação, estimulando o espírito de coletividade e troca de informações e dicas úteis.

10.4.5 Mudanças que Impactam nos Colaboradores

As organizações devem estar preparadas para ensinar aos seus colaboradores que as necessidades dos clientes podem sofrer mudanças, tornando-se impossível prever resultados específicos. Os colaboradores devem estar prontos para as constantes mudanças que podem ocorrer, encarando-as como um fato da vida profissional. Os colaboradores que possuem tal preparo são mais flexíveis e mais sensíveis às necessidades dos clientes e suas mudanças, podendo contribuir com sua flexibilidade e sensibilidade para o crescimento da organização no geral.

11 O Aplicativo Desenvolvido

11.1 Introdução à Ferramenta e Projeto de Desenvolvimento

Algo que pode preocupar os responsáveis pela implantação de CRM dentro de uma organização na hora do processo de formação é o desafio de atingir globalmente os colaboradores e motivá-los a aprender sobre CRM. Pensando nisso este trabalho procurou desenvolver um aplicativo pré-formação que seja de fácil acesso e instrutivo. Esta ferramenta não tem como objetivo realizar o processo de formação por si só, mas sim colaborar com os esforços dos responsáveis pela implantação das estratégias CRM, ajudando pessoas que tiveram pouco ou nenhum contato com CRM a desvendar um pouco de seus mistérios e conceituar alguns pontos relativos ao tema. Deverá servir também como motivador, aguçando a vontade dos colaboradores em aprender. Serão focados os conceitos básicos para conscientizar um determinado grupo de colaboradores sobre a importância do cliente e de tê-lo como elemento fundamental do negócio da organização.

Quanto à fase de planejamento e desenvolvimento da ferramenta chegou-se à conclusão que deveria ser utilizada alguma técnica ágil para o projeto. Isto foi interessante, pois consolida nossas premissas de que esta é a forma mais adequada para obter retorno rápido a um esforço realizado e abriu-se a possibilidade de demonstrar de forma prática as vantagens de se adotar tal filosofia.

Identificamos que a ferramenta podia ser particionada em funcionalidades que, se planejadas adequadamente, determinariam passos simples que deveriam ser seguidos para que a cada interação seja obtida uma versão executável do aplicativo. Dessa forma o desenvolvimento foi baseado em FDD (*Feature Driven Development*), pois oferecia liberdade de mesclar metodologias, desenvolver iterativamente, com entregas frequentes e tangíveis e também suporta desenvolvimento ágil com rápidas adaptações. Como se acredita que a filosofia é mais importante que a metodologia foram feitas adaptações no modelo para que melhor se adequasse a nossas necessidades.

Dessa forma foi elaborada uma lista de funcionalidades (*Features List*), em conjunto com um cronograma, ambos anexos, que abrangem as funcionalidades independentemente. Vale ressaltar que apesar do cronograma contemplar as funcionalidades sendo feitas seqüencialmente elas não estão relacionadas, possibilitando um desenvolvimento em paralelo, se necessário, ou invertendo a ordem das funcionalidades, dando liberdade ao projeto.

11.2 A Usabilidade e a Ferramenta

Ter uma ferramenta que auxilie o processo de formação dos colaboradores e não pensar em como esse possa ser agradável para o usuário é ir contra a lógica de pensar nos clientes. Nesse caso específico o cliente interno, que seria o colaborador. Por isso a necessidade de se utilizar técnicas de usabilidade.

Destacamos aqui a metodologia de design centrado no usuário. Essa metodologia já está sendo desenvolvida há mais de 20 anos, e começou no âmbito de softwares e ultimamente também atinge os websites. Essa metodologia oferece uma gama de ferramentas práticas para guiar o projeto em direção à facilidade de uso e maximização de entendimento para os seus usuários. Não é objeto desse estudo detalhar essa metodologia, porém seus vários princípios que vão desde classificar e agrupar as informações para melhor entendimento e administração até a formação do modelo mental do usuário são práticos e devem ser estudados pelas empresas que procurem utilizar ferramentas eletrônicas na formação de seu pessoal.

11.3 Funcionalidades

Como o processo de formação é um dos passos mais importantes da implantação do CRM em uma organização, esse processo necessita ser agradável. A ferramenta tem como premissa ser funcional, mas também divertida. Dessa forma o colaborador ficaria atraído a fazer os exercícios. Pode ser feita como lazer, já que a ferramenta possui uma área com bateria de perguntas, exercícios de relacionamento entre colunas e preenchimento de lacunas. As respostas podem ser verificadas, para que haja o efeito instrutivo, e os resultados armazenados em um banco de dados. Ele estará localizado em um servidor *web*, com um banco de dados para armazenamento das informações. Os colaboradores poderão acessá-lo de qualquer computador com um browser adequado. Neste mesmo banco ocorre o armazenamento das questões que são dispostas nas atividades, essas são flexíveis, facilmente adaptadas às necessidades de cada organização, dependendo de uma simples atualização destas questões.

11.4 Escolha do Banco de Dados

A escolha do PostgreSQL foi fundamentada no fato dele ser um programa distribuído sob a licença BSD, o que torna o seu código fonte disponível e o seu uso livre para aplicações comerciais ou não (PostgreSQL Brasil,2004). A seguir é trazido um histórico acerca do PostgreSQL, mostrando sua evolução até o presente momento.

11.4.1 Histórico do Postgresql

O PostgreSQL derivou do projeto Postgres da Universidade de Berkley. Foi originalmente patrocinado pelo DARPA (Agência de Projetos de Pesquisa Avançada para Defesa), ARO (Departamento de Pesquisa Militar), NSF (Fundação Científica Nacional) e ESL Inc. A implementação do projeto iniciou em 1986 e em 1987 tornou-se operacional. A primeira versão lançada para o público externo foi em 1989.

O Postgres95 teve mudanças radicais em relação ao projeto original. Seu código foi totalmente revisado, o tamanho dos códigos fontes foi reduzido em 25%, e a linguagem SQL foi implementada como interface padrão. O desempenho foi consideravelmente melhorado e novos recursos foram adicionados. Em 1996 o projeto foi rebatizado "PostgreSQL", para enfatizar a relação do Postgres original com a linguagem SQL.

11.4.1.1 O PostgreSQL Hoje

A equipe do projeto cresceu e se espalhou pelo mundo. O Grupo Global de Desenvolvimento do PostgreSQL tem membros nos Estados Unidos, Canadá, Japão, Rússia, vários países da Europa e alguns outros. Esse grupo é formado essencialmente por organizações especializadas em PostgreSQL, organizações usuárias do sistema, além dos pesquisadores acadêmicos e programadores independentes. Além da programação, essa comunidade é responsável pela documentação, tradução, criação de ferramentas de modelagem e gerenciamento, e elaboração de extensões e acessórios.

Pela riqueza de recursos e conformidade com os padrões, ele é um SGBD muito adequado para o estudo universitário do modelo relacional, além de ser uma ótima opção para organizações implementarem soluções de alta confiabilidade sem altos custos de licenciamento.

Alguns recursos presentes na versão mais recente:

- Sub-consultas;
- Controle de concorrência multi-versão (MVCC);
- Integridade referencial;
- Funções armazenadas (*Stored Procedures*), que podem ser escritas em várias linguagens de programação;
- Gatilhos (*Triggers*);
- Tipos definidos pelo usuário;
- Esquemas (*Schemas*);
- Conexões SSL.

11.5 O Framework Struts

A escolha do *Framework Struts* foi fundamentada no fato do mesmo possuir uma arquitetura aberta e voltada para a *web*. O reuso das classes que compõe um *framework* possibilita a construção de aplicações de forma rápida e fácil.

11.5.1 Frameworks

Frameworks são classes que formam um conjunto e que podem ser facilmente reutilizadas para a construção de aplicações. Para que isto ocorra, simplesmente adicionam-se as classes que atendem às funcionalidades requeridas por

uma determinada aplicação ao *framework* para obter-se uma aplicação pronta para ser utilizada.

As classes que compõe um *framework* estão baseadas nos princípios de reutilização de software, que objetiva:

- Aumento de produtividade;
- Retenção de conhecimento;
- Redução de prazos;
- Aumento de interoperabilidade;
- Otimização de recursos;
- Redução de custos;
- Melhoria da qualidade;
- Aumento de competitividade.

11.5.2 Definição do Framework Struts

O Projeto *Jakarta Struts* é um projeto de código aberto patrocinado pela *Apache Software Foundation*. 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 transformando-se 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* 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 o projeto *Tomcat*, que também será utilizado nesse trabalho. O projeto *Tomcat* será brevemente descrito posteriormente.

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*.

11.6 Escolha do Servidor de Aplicação

A escolha do servidor de aplicação *Tomcat* do projeto *Apache Jakarta* foi fundamentada no fato de ser um software livre e possuir o código aberto. É um servidor robusto e eficiente para ser utilizado no presente trabalho.

11.6.1 Tomcat

O *Tomcat* é um servidor de aplicações *Java* para *web*. É software livre e de código aberto desenvolvido dentro do conceituado projeto *Apache Jakarta* e oficialmente endossado pela *Sun* como a Implementação de Referência (RI) para as tecnologias *Java Servlet* e *JSP* (D'AVILA, 2003).

O *Tomcat* é um Container Web, parte da plataforma *J2EE* que abrange as tecnologias *Servlet* e *JSP*, incluindo tecnologias de apoio relacionadas como *Realms* e *segurança*, *JNDI Resources* e *JDBC DataSources*. Tem a capacidade de atuar também como servidor *web/HTTP*, ou pode funcionar integrado a um servidor *web* dedicado (D'AVILA, 2003).

11.6.2 Acesso ao Aplicativo

O aplicativo está disponível na web com acesso em qualquer browser no endereço: http://200.247.141.68:8080/aprendizado_crm

12 Dificuldades Encontradas

As premissas definidas pelo Gerenciamento de Relacionamento com Cliente exercem uma série de mudanças impactantes na estrutura de qualquer organização que resolve abraçar uma nova forma de apresentar-se ao mercado. Assim sendo, enquanto estudávamos tais premissas e a viabilidade de realizá-las de forma ágil, surgiu a preocupação quanto à maneira de aplicar tal estudo e transformá-lo em nosso trabalho de conclusão de curso.

Nasceu então a idéia de realizar um estudo de caso, fazendo a implantação desde o início em uma pequena academia. Como qualquer trabalho que vai amadurecendo durante o tempo, verificamos que tínhamos que estudar assuntos que não vislumbravam o foco principal do nosso trabalho como, por exemplo, o mercado atual de academias.

Por fim, nossa intenção foi direcionada a apresentar sugestões para o processo de formação que fossem eficientes e ágeis sem fugir ao conceito básico de CRM. Foi assim que se chegou à conclusão que implementar um aplicativo que auxiliasse nesse processo daria o embasamento necessário ao nosso trabalho, tornando-o prático. Deixou-se de lado, portanto, a possibilidade de realizar o estudo de caso.

13 Trabalhos Futuros

Consideramos que os estudos teóricos poderiam ser aprofundados até uma compreensão completa de como ocorre toda a fase de formação, manutenção e monitoramento dos resultados do processo. Fica também sugerida a complementação da parte prática, onde toda a fase inicial do processo de formação é contemplada.

14 Conclusão

Trabalhar com CRM é cada vez mais imperativo para as empresas que querem sobreviver no mercado e desejam conseguir lucros cada vez maiores.

Envolver seus colaboradores é essencial para que esse processo e filosofia funcionem, pois é a partir deles que a empresa se apresenta ao cliente, foco principal de CRM.

Portanto, dar atenção ao processo de formação não é desperdiçar dinheiro, mas sim um investimento que dá retorno pelos efeitos que acabam proporcionando.

Fazer tudo isso aplicando técnicas de agilidade é uma boa prática, pois o feedback dos investimentos são apresentados e visualizados de forma mais rápida, motivando a continuação da implantação.

15 Referências

AGILE MANIFESTO. **History: The Agile Manifesto**. Disponível em <<http://agilemanifesto.org/history.html>>. Acessado em: 24 mai 2005.

BROWN, Stanley. **CRM – Customer Relationship Management – Uma Ferramenta Estratégica para o Mundo do E-business**. São Paulo: Makron Books do Brasil, 2001.

CHIN, Gary. **Agile Project Management**. New York: Amacon, 2004.

CONSULTORIA, Opinião. **Porque pesquisar?** Disponível em: <<http://www.wbrinternet.com.br/opiniaio/informacoes.htm>>. Acessado em: 20 jun 2004.

CURRY, Jay; CURRY, Adam. **Conceitos de CRM para Pequenas e Médias Organizações**. White Paper, 2000.

D'AVILA, Márcio. **Tutorial Tomcat - Instalação e Configuração Básica**. Disponível em: <<http://www.mhavila.com.br/topicos/java/tomcat.html>>. Visualizado em: 24 mai 2005.

Dicionário Publicitário On-line. Disponível em: <<http://geocities.yahoo.com.br/dicpublicitario/p.htm>>. Acessado em: 05 de mai de 2005.

FERNANDES, A. **Administração Inteligente – Novos Caminhos para as Organizações do Século XXI**. São Paulo: Editora Futura, 2001.

FILHO, Wilson de Pádua Paula. **Engenharia de Software – Fundamentos, Métodos e Padrões**. Rio de Janeiro: LTC – Livros Técnicos e Científicos Editora, 2001.

GREENBERG, Paul. **CRM na velocidade da Luz**. Editora Campus, 2001.

HIGHSMITH, Jim. **Agile Project Management, Creating Innovative Products**. Addison-Wesley Editora, 2004.

HIGHSMITH, Jim. **Agile Software Development Ecosystems**. Addison-Wesley Editora, 2002.

HUSLY, Harry; YU, Jonh. **Issues in Struts Adoption**. Disponível em: <http://www.scioworks.net/devnews/articles/struts_adoption_issues/>. Acessado em: 10 de abr de 2005.

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

JAKARTA Struts Framework Disponível em: <<http://struts.apache.org>>. Acessado em 19 de abr de 2005.

JONES, Thomas. **Fidelização de clientes – medindo, gerenciando e criando valor**. Disponível em: <<http://www.guiarh.com.br/p64.htm>>. Acessado em: 24 de abr de 2005.

KON, Fábio; Goldman, Alfredo. **Métodos Ágeis e Programação Extrema**. Disponível em: <<http://www.ime.usp.br/~gold/xp/SBES03-extended06.pdf>>. Acessado em: 28 nov 2004.

KOTLER, Philip. **Marketing Management**. USA: Pentice-Hall, 1972.

LEE, Dick. **Redefining CRM**. Disponível em: <<http://dcrm.infotoday.com/articles/default.asp?ArticleID=855>>. Acessado em: 20 de jun de 2004.

LEITE, Maria Marta. **Pressupostos para Implantação de Estratégias de Relacionamento com os Clientes em Pequenas e Médias Organizações: uma Abordagem baseada em Gerenciamento de Projetos**. Tese (Doutorado em Engenharia de Produção). Curso de Pós-graduação em Engenharia de Produção. Universidade Federal de Santa Catarina, Florianópolis, 2004.

LEVITT, Theodore. **The Managerial Merry-go-round**. USA: HarvardBusiness Review, 1974.

HILMER. **Metodologias Ágeis**. Disponível em:

<<http://www.redes.unb.br/material/ESOO/Metodologias%20%C1geis.pdf>>. Visualizado em: 10 dez 2004.

MONGER, Jodie. **Customer-Driven Training: Using the Voice of the Customer to Maximize your ROI**. Disponível em:

<<http://www.crm2day.com/library/EEEEupuykVzhsJPUEJ.php>>. Acessado em: 03 de mai de 2005.

NEKI Technologies. Disponível em: <<http://www.NEKI-technologies.com.br>>. Acessado em: 22 nov 2004.

OLIVEIRA, Cezar Augusto de. **O Processo de Tomada de Decisão**. Disponível em:

<<http://www.professorcezar.adm.br/Textos/O%20processo%20de%20tomada%20de%200decisao.pdf>>. Acessado em: 27 mai 2005.

PEPPERS & ROGERS GROUP. **Esqueça os Obstáculos da Tecnologia: o CRM a toda a velocidade**. White Paper, 2001.

PEPPERS & ROGERS GROUP. **Preparando seus Funcionários para a Competição Direcionada ao Cliente**. Disponível em:

<http://www.1to1.com.br/artigo_1to1.php3?identf=71>. Acessado em: 28 nov 2004.

POSTGRESQL Brasil. **Histórico do PostgreSQL**. Disponível em:

<http://www.postgresql.org.br/conteudo.php?inc=conteudo/texto&tx=6&id_aba=4>. Acessado em: 19 abr 2005.

RASKIN, Sara Fichman. **Tomada de decisão e aprendizagem organizacional**.

Disponível em: <<http://www.pr.gov.br/batebyte/edicoes/2003/bb135/tomada.shtml>>. Acessado em: 27 mai 2005.

SILVEIRA, Paulo. **Satisfação e fidelidade dos clientes**. Disponível em:

<<http://www.academianovak.com.br/comunidade/modules.php?name=News&file=print&sid=79>>. Acessado em: 24 de abr de 2005.

SIMON, H.A. **Administrative Behavior: a study of decision making processes in administrative organizations**. New York : McMillan, 2ed., 1957.

SWIFT, Ronald. **CRM – O Revolucionário Marketing de Relacionamento com o Cliente**. Editora Campus, 2001.

SYMNETICS Business Transformation. Excellence report de Abril de 2000. Ano IV – nº 7.

TOMCAT Project. Disponível em: <<http://jakarta.apache.org/tomcat/index.html>>.

Acessado em: 19 de abr de 2005.

TRONCHIN, Valsoi. **CRM não é bom serviço ao cliente**. Disponível em: <http://www.1to1.com.br/artigo_1to1.php3?identf=97>. Acessado em: 28 nov 2004.

TRONCHIN, Valsoi. **CRM, rentabilidade e fidelidade**. Disponível em: <http://www.1to1.com.br/artigo_1to1.php3?identf=98>. Acessado em: 28 nov 2004.

UFSC, Curso de Sistemas de Informação. **O curso de Sistemas de Informação**.

Disponível em: <<http://www.inf.ufsc.br/sin/index.html>>. Acessado em: 04 abr 2005.

ZAIDOWICZ Consultoria Fiscal. **Custo e Manutenção de Cliente**. Disponível em: <http://www.zaidowicz.com.br/ver_boletim_executivo.php?codigo=111>. Acessado em:

24 de maio de 2005.

Apêndice A

Funcionalidades Aplicativo Crescer (Features List)

- Boas-Vindas
- Cadastro (login e senha)
- Banco de Dados (conterá informações do colaborador e resultados das atividades elaboradas no aplicativo)
 - o Modelagem do Banco
 - o Conexão com o aplicativo
- Tela de escolha para as atividades educativas (quiz)
 - o Quiz:
 - Bateria de perguntas
 - Armazenamento das respostas
 - Resultados
 - Gabarito
- Layout
- Testes

Apêndice B

Perguntas Existentes No Banco de Dados Do Aplicativo Crescer

- 1 – Qual dessas frases se relaciona melhor com CRM?
 - a) O cliente é nossa razão de ser.
 - b) Temos de produzir ao máximo.
 - c) O que importa é dinheiro no bolso.
 - d) O que eu quero é sossego.

- 2 – Dentro de nossa empresa, quem deve estar interessado no cliente?
 - a) Todos
 - b) O presidente
 - c) O pessoal do marketing.
 - d) Ninguém. Nosso produto é o melhor.

- 3 – A empresa deve se importar com o cliente quando:
 - a) Sempre

- b) Quando quer vender seu produto.
- c) Quando começam a acontecer reclamações
- d) Nunca

4 – Você caracterizaria CRM como:

- a) Um modelo de gestão onde o relacionamento com o cliente é o foco principal da empresa.
- b) Uma estratégia para investir em tecnologia
- c) Um modo de a empresa ficar conhecida no mercado.
- d) É o Conselho Regional de Medicina.

5 – Quais dos adjetivos abaixo se relacionam melhor com uma empresa que aplica CRM?

- a) Fidelidade do Cliente, Satisfação e Lucro.
- b) Alto custo de produção e Burocracia.
- c) Tecnologia de ponta e Processos automatizados.
- d) Boa, Bonita e Barata.

6 – O CRM procura em primeira instância dar maior atenção à:

- a) Satisfação dos clientes existentes.
- b) Encontrar novos clientes.
- c) Conseguir os clientes dos concorrentes.
- d) Vender nossos produtos.

Apêndice C - Artigo

Gerenciamento do Relacionamento com Clientes utilizando Gerenciamento de Projetos Ágeis

Henrique de Britto Farias e Roberta Oliveira - 2005

hfarias@inf.ufsc.br / beta@inf.ufsc.br

Resumo

Este artigo procura apresentar estudos de como Gerenciamento de Projetos Ágeis é interessante para aplicação de CRM em uma empresa. Esse estudo tem

ênfase na fase de treinamento, onde se chega a desenvolver parcialmente uma ferramenta que visa auxiliar a ensinar os conceitos básicos de CRM.

Palavras-Chave

CRM; Gerenciamento de Projetos Ágeis; Treinamento; Agilidade; Estratégia;

1. Introdução

O mercado atual mostra-se extremamente competitivo, apresentando empresas que buscam constantemente a evolução dos serviços prestados. Crescimento que visa lucros, aumento da participação no mercado e excelência em seus produtos e serviços faz parte desta busca. Sendo assim, novos modelos de gerência têm surgido. Ferramentas que auxiliam as empresas a atingir esses objetivos e o Gerenciamento do Relacionamento com o Cliente, daqui por diante dito somente CRM, têm ganhado destaque. O motivo para tal destaque é a visão de que é essencial ter clientes satisfeitos para que o negócio vá bem. Ainda sobre CRM, este pode ser visto como uma tarefa tanto operacional quanto estratégica. E para que ocorra um bom andamento de uma empresa, sua implantação pode ser tratada como um Projeto (Leite 2004, p. 21), ou um conjunto deles. Assim sendo, podem ser utilizados as premissas e os conceitos abordados pelo Gerenciamento de Projetos para colocá-lo em prática, e ainda mais precisamente, pode ser utilizado um Gerenciamento de Projetos Ágeis para satisfazer de forma mais ágil os anseios das empresas relativos ao aspecto CRM. Através de uma Gestão de Projetos de forma ágil, as empresas deverão obter um tempo de resposta muito menor para os seus investimentos, considerando-se o Gerenciamento de Projetos Tradicional. Desta forma, a centralização no cliente passa a ser o foco, fazendo com que as empresas tornem operacional este foco, o que deveria acontecer desde a sua fundação.

Através do contexto relatado acima, formulamos o nosso problema: De que forma tornar a implantação do Gerenciamento de Relacionamento com o Cliente mais rápido e operacional para as empresas, sem deixar de considerar as premissas básicas de CRM?

2. CRM

Se formos procurar uma definição para CRM encontraremos várias, porém pode ser encarado como uma estratégia empresarial que tem como objetivo colocar em prática um conjunto de ações que tenha esta filosofia: O cliente como a razão de ser do negócio. Procura estabelecer um relacionamento achegado com os clientes. (Leite 2004)

Deve abranger a empresa por completo, pois suas atividades de alguma forma sempre impactam no cliente, desde o atendente do call center e o vendedor, até o presidente da empresa, que aliás precisa estar engajado nesse processo também pois CRM começa no topo da organização. Começa com gerenciamento executivo. (Brown 2001)

Alguns confundem CRM como uma mera ferramenta de tecnologia. (FEI 2001) A tecnologia com certeza pode auxiliar no desenvolvimento desta estratégia, mas é fundamental não desconsiderar muitos outros mecanismos importantes para que a organização dê prioridade aos seus clientes. A aplicação de CRM não pode estar vinculada somente à tecnologia (Leite 2004), porque ações relativas ao papel das pessoas e a avaliação dos processos organizacionais tendo como foco o cliente são independentes de tecnologia.

Feito esse embasamento surge a pergunta: Porque usar CRM? Qual o objetivo de aplicar essa estratégia de relacionamento? Uma empresa sem clientes não sobrevive, então ela deve estar preocupada em manter a sua carteira de clientes, não perdê-los. E um cliente satisfeito representa a possibilidade do aumento de sua

interação com a empresa. Ele compra mais, relaciona-se mais, conseqüentemente tornando-se fiel. Aproveitando-se disso é uma ótima forma de fazer marketing, pois o cliente satisfeito mostra para todos a sua satisfação (Symnetics 2001), o que vai acabar atraindo novos clientes para a empresa, novas parcerias e possibilidades.

Quanto ao desempenho financeiro é muito mais barato manter um cliente satisfeito do que investir na aquisição de novos clientes. Só isso já traz economia, mas não para por aí. Conhecendo o cliente as oportunidades de vendas são melhores aproveitadas. Os investimentos têm o retorno esperado, a previsão de vendas é mais realista o que torna produção otimizada, enfim, entre outras coisas no mínimo reduz desperdícios. Outra vantagem econômica é que as estratégias de marketing podem ser direcionadas (Greenberg, 2001).

CRM preocupa-se também em assegurar que o cliente tenha o que deseja da organização, estendendo os serviços em nível de personalização. É o conhecimento individual, de cada cliente, que possibilita isso. Alguns autores chamam isso de marketing 1 a 1, que é a busca da personalização da mesma forma que era feito há tempos atrás, onde o dono de uma empresa conhecia pessoalmente os clientes e suas necessidades. E hoje se tenta aproximar disso, muitas vezes com o auxílio de tecnologia (Peppers and Rogers Group, 2001).

CRM procura resolver também a comunicação e interação com o cliente. Ouvindo o cliente, comunicando-se com ele é que possível satisfazê-lo, deixando que ele palpite, auxilie nas tarefas, dessa forma pode-se re-estabelecer estratégias e tomar ações refinadas. Se isso for feito a chance da sua empresa estar indo pelo caminho errado, fica minimizado (Symnetics, 2000).

Todos esses aspectos em conjunto visam criar um relacionamento onde o cliente está satisfeito tornando-se leal à empresa.

3. Agilidade

A definição de agilidade segundo um autor chamado Jim Highsmith é “a habilidade tanto de criar quanto se adequar à mudanças de uma forma que incremente a eficiência em um ambiente de trabalho turbulento”. (Highsmith, 2002) Capacidade de mudar de direção rapidamente sendo flexível. Organizações ágeis também entendem que o equilíbrio entre ordem e caos determina o sucesso.

Através de uma Gestão de Projetos de forma ágil, as empresas deverão obter um tempo de resposta muito menor para os seus investimentos, considerando-se o Gerenciamento de Projetos Tradicional. É a tentativa de dar um retorno rápido de um determinado desenvolvimento para a administração.

Outra vantagem seria de mostrar com maior clareza o andamento do projeto como um todo. Assim a administração da empresa vai saber e ter cada vez mais a certeza de que o tempo, que é precioso, está sendo bem usufruído, e que trará resultados satisfatórios.

Para que essas coisas aconteçam, se tornem realidade, em Projetos Ágeis o ambiente é colaborativo. As pessoas se ajudam. Novamente volta-se à definição feita de agilidade. É necessária uma ordem, mas com equilíbrio pra não esbarrar na burocracia.

Algo que necessita ser esclarecido, é que Projetos Ágeis não tem a ver com menor quantidade de processos, documentos resumidos e menos formalidades. Pode até ter um pouco disso, mas isso não é suficiente para trazer agilidade. O que se procura é focar nas pessoas e suas interações, que isto esteja bem resolvido, e não procurar seguir padronizações estáticas. (Highsmith, 2002) Um detalhe é que nesse tipo de gerenciamento pessoas individuais têm de ter poderes para realizar decisões rápidas deixando o ambiente ágil.

3. CRM com Agilidade

De acordo com Maria Marta Leite, a implantação de CRM pode ser tratada como um projeto (Leite, 2004). E se utilizada a técnica de Projetos Ágeis há benefícios, porque dado um investimento feito na implantação de CRM, se os resultados têm retorno acelerado, isso traz motivação e aporte de fundos para continuar o processo. (Peppers and Rogers Group, 2001). Uma proposta para esse resultado aconteça é que a empresa passe por um processo de implantação de CRM em ciclos de desenvolvimento. Sempre avaliando os resultados, determinando as novas estratégias e impulsionando o CRM, para que se torne contínuo.

4. As pessoas

Um grande pilar da estrutura desse estudo aqui apresentado são as pessoas. Isto acontece pois relacionamentos não se fazem sem as pessoas, mesmo que se esteja representando uma organização. São as pessoas que podem fazer diferença no processo de implantação de estratégias para CRM e para agilidade. Por isso deve-se ter preocupação diferenciada. Há necessidade de manter as equipes conscientes e motivadas (Leite, 2004). Assim, a preparação de pessoal, através da formação e aprimoramento, por meio de treinamento, é requisito indispensável.

5. Os desafios de formar os colaboradores

A fase de formação é uma fase cheia de desafios, e entre eles nós alistamos alguns como principais (Neki Technologies, 2004):

- Necessidade de atingir globalmente os colaboradores.
- Falta de Tempo para o processo de formação.
- Constante manutenção da formação dos colaboradores.

- Motivação – Criar o desejo de aprender nos colaboradores.

O processo de formação dos colaboradores de uma organização deve ser realizado de forma a assegurar o seu sucesso. Este processo não é um custo de negócio, mas sim um investimento que deve ter um retorno positivo (MONGER, 2005). Deve justificar o investimento realizado por parte da organização. E já que estamos falando desse processo no âmbito da implantação de CRM deve existir uma preparação dos colaboradores para o direcionamento ao cliente. Preocupa-se então principalmente com (Peppers and Rogers Group, 2004):

- Ensinar Colaboradores a Gerenciar Informações
- Motivar o compartilhamento de informações na Organização
- Ensinar que mudanças nos clientes impactam nos colaboradores

Portanto, depois de analisados os desafios nada mais claro que se procure vencer esses desafios, então foi criada uma aplicação que se propõe a ensinar os conceitos básicos de CRM, de forma didática e simples.

6. Aplicativo Crescer

Procurou-se desenvolver um aplicativo pré-formação que seja de fácil acesso e instrutivo. Esta ferramenta não tem como objetivo realizar o processo de formação por si só, mas sim colaborar com os esforços dos responsáveis pela implantação das estratégias CRM, ajudando pessoas que tiveram pouco ou nenhum contato com CRM a desvendar um pouco de seus mistérios e conceituar alguns pontos relativos ao tema. Deverá servir também como motivador, aguçando a vontade dos colaboradores em aprender. A ferramenta tem como premissa ser funcional, mas também divertida. Dessa forma o colaborador ficaria atraído a fazer os exercícios. Pode ser feita como lazer, já

que a ferramenta possui uma área com bateria de perguntas. Ela pode ser acessada no seguinte endereço: http://200.247.141.68:8080/aprendizado_crm.

Possui um banco de dados com as informações dos colaboradores e perguntas, que podem ser dinamicamente alteradas, segundo a necessidade do instrutor.

7. Referências

NEKI Technologies. Disponível em: <<http://www.NEKI-technologies.com.br>>. Acessado em: 22 nov 2004.

PEPPERS & ROGERS GROUP. **Esqueça os Obstáculos da Tecnologia: o CRM a toda a velocidade**. White Paper, 2001.

PEPPERS & ROGERS GROUP. **Preparando seus Funcionários para a Competição Direcionada ao Cliente**. Disponível em: <http://www.1to1.com.br/artigo_1to1.php3?idenf=71>. Acessado em: 28 nov 2004.

BROWN, Stanley. **CRM – Customer Relationship Management – Uma Ferramenta Estratégica para o Mundo do E-business**. São Paulo: Makron Books do Brasil, 2001.

HIGHSMITH, Jim. **Agile Project Management, Creating Innovative Products**. Addison-Wesley Editora, 2004.

HIGHSMITH, Jim. **Agile Software Development Ecosystems**. Addison-Wesley Editora, 2002.

LEITE, Maria Marta. **Pressupostos para Implantação de Estratégias de Relacionamento com os Clientes em Pequenas e Médias Organizações: uma Abordagem baseada em Gerenciamento de Projetos**. Tese (Doutorado em Engenharia de Produção). Curso de Pós-graduação em Engenharia de Produção. Universidade Federal de Santa Catarina, Florianópolis, 2004.

MONGER, Jodie. **Customer-Driven Training: Using the Voice of the Customer to Maximize your ROI**. Disponível em: <<http://www.crm2day.com/library/EEEEupuykVzhsJPueJ.php>>. Acessado em: 03 de mai de 2005.

SYMNETICS Business Transformation. Excellence report de Abril de 2000. Ano IV – nº 7.

GREENBERG, Paul. **CRM na velocidade da Luz**. Editora Campus, 2001.

Apêndice D

Projeto Lógico do Banco de Dados

1 Apresentação

O presente apêndice tem por objetivo a apresentação do Projeto Lógico do Banco de Dados utilizado em conjunto com a ferramenta desenvolvida.

2 Tabelas

Algumas observações:

- Os atributos que estão em vermelho são chaves-estrangeiras.
- Os atributos que estão sublinhados são chaves-primárias

2.1.1 Tabela Escolaridade

A tabela de Escolaridade possui seis níveis possíveis de escolaridade:

- Primeiro Grau Incompleto
- Primeiro Grau Completo
- Segundo Grau Incompleto
- Segundo Grau Completo
- Terceiro Grau Incompleto
- Terceiro Grau Completo

A tabela é estruturada conforme consta a seguir:

- tab_escolaridade: (cd_escolaridade, ds_escolaridade)

2.1.2 Criação da Tabela

```
CREATE TABLE public.tab_escolaridade
(
  cd_escolaridade numeric(2) NOT NULL,
  ds_escolaridade varchar(50),
  CONSTRAINT tab_escolaridade_pkey PRIMARY KEY (cd_escolaridade)
) WITH OIDS;
```

2.2.1 Tabela Empresa

A tabela de Empresa possui a(s) empresa(s) cadastradas no sistema. É estruturada conforme consta a seguir:

- tab_empresa: (cd_emp, nm_emp)

2.2.2 Criação da Tabela

```
CREATE TABLE public.tab_empresa
```

```
(
  cd_emp numeric(2) NOT NULL,
  nm_emp varchar(50),
  CONSTRAINT pk_empresa PRIMARY KEY (cd_emp)
) WITH OIDS;
```

2.3.1 Tabela Setor

A tabela de Setor possui os setores existentes dentro de uma determinada empresa. A tabela é estruturada conforme consta a seguir:

- tab_setor: (cd_emp, cd_setor, ds_setor)

O atributo cd_emp referencia a Tabela Empresa.

2.3.2 Criação da Tabela

```
CREATE TABLE public.tab_setor
(
  cd_emp numeric(2) NOT NULL,
  cd_setor numeric(2) NOT NULL,
  ds_setor varchar(50),
  CONSTRAINT tab_setor_pkey PRIMARY KEY (cd_emp, cd_setor),
  CONSTRAINT fk_setorfk_setor FOREIGN KEY (cd_emp) REFERENCES
public.tab_empresa (cd_emp) ON UPDATE RESTRICT ON DELETE RESTRICT
) WITH OIDS;
```

2.4.1 Tabela Colaborador

A tabela Colaborador possui os colaboradores existentes dentro de um determinado setor de uma determinada empresa. Está estruturada da seguinte maneira:

- tab_colaborador:

(cd_emp,cd_setor,cd_colaborador,nm_colaborador,id_sexo,dt_admissao,cd_escolaridade)

O atributo `cd_emp` referencia a Tabela Empresa; o atributo `cd_setor` referencia a Tabela Setor; o atributo `cd_escolaridade` referencia a Tabela Escolaridade; `dt_admissao` é a data de admissão do Colaborador, no formato `dd/mm/aaaa` e o atributo `idsexo` informa o sexo do colaborador da seguinte maneira:

- 'F' no caso de feminino
- 'M' no caso de masculino.

2.4.2 Criação da Tabela

```
CREATE TABLE public.tab_colaborador
(
  cd_emp numeric(2) NOT NULL,
  cd_setor numeric(2) NOT NULL,
  cd_colaborador numeric(2) NOT NULL,
  nm_colaborador varchar(80),
  idsexo varchar(1) NOT NULL,
  dt_admissao date NOT NULL,
  cd_escolaridade numeric(2) NOT NULL,
  CONSTRAINT tab_colaborador_pkey PRIMARY KEY (cd_colaborador),
  CONSTRAINT fk_escolaridadefk_escolaridade FOREIGN KEY (cd_escolaridade)
REFERENCES public.tab_escolaridade (cd_escolaridade) ON UPDATE RESTRICT ON DELETE
RESTRICT,
  CONSTRAINT fk_setorfk_setor FOREIGN KEY (cd_emp, cd_setor) REFERENCES
public.tab_setor (cd_emp, cd_setor) ON UPDATE RESTRICT ON DELETE RESTRICT
) WITH OIDS;
```

2.5.1 Tabela Usuário

A tabela Usuário possui os usuários e senhas dos colaboradores para o acesso ao sistema. Encontra-se estruturada da seguinte maneira:

- `tab_usuario`: (`nm_usuario`,`ds_senha`)

2.5.2 Criação da Tabela

```
CREATE TABLE public.tab_usuario
(
  nm_usuario varchar(30) NOT NULL,
  ds_senha varchar(8) NOT NULL,
  CONSTRAINT tab_usuario_pkey PRIMARY KEY (nm_usuario)
) WITH OIDS;
```

2.6.1 Tabela Estatística de Acesso

A tabela Estatística é responsável por gravar a entrada de um determinado colaborador no sistema. Está definida conforme a seguir:

- tab_estatistica_acesso: (dt_acesso,nm_usuario)

O atributo nm_usuario e referencia a Tabela Usuário. O atributo dt_acesso está no formato timestamp 1999-01-08 04:05:06.

2.6.2 Criação da Tabela

```
CREATE TABLE public.tab_estatistica_acesso
(
  dt_acesso timestamp NOT NULL,
  nm_usuario varchar(30) NOT NULL,
  CONSTRAINT tab_estatistica_acesso_pkey PRIMARY KEY (dt_acesso, nm_usuario),
  CONSTRAINT fk_usuario FOREIGN KEY (nm_usuario) REFERENCES
public.tab_usuario (nm_usuario) ON UPDATE RESTRICT ON DELETE RESTRICT
) WITH OIDS;
```

2.7.1 Tabela Pergunta

A tabela Pergunta contém todas as perguntas a serem realizadas para os colaboradores, a fim de verificar o conhecimento dos mesmos a respeito dos conceitos relacionados ao Gerenciamento de Relacionamento com o Cliente. A tabela está organizada conforme consta a seguir:

- tab_pergunta: (cd_pergunta,ds_pergunta)

2.7.2 Criação da Tabela

```
CREATE TABLE public.tab_pergunta
(
  cd_pergunta numeric(2) NOT NULL,
  ds_pergunta varchar(150) NOT NULL,
  CONSTRAINT pk_pergunta PRIMARY KEY (cd_pergunta)
) WITH OIDS;
```

2.8.1 Tabela Resposta

A tabela Resposta possui todas as respostas que são disponibilizadas para selecionar acerca de uma determinada pergunta. Está definida conforme a seguir:

- tab_resposta: (cd_pergunta,cd_resposta,ds_resposta,id_status)

O atributo cd_pergunta referencia a Tabela Pergunta. O atributo id_status é definido da seguinte maneira:

- 'C' para a resposta que é correta
- 'I' para a resposta que é incorreta.

2.8.2 Criação da Tabela

```
CREATE TABLE public.tab_resposta
(
  cd_pergunta numeric(2) NOT NULL,
  cd_resposta numeric(3) NOT NULL,
  ds_resposta varchar(150) NOT NULL,
```

```

        id_status varchar(1) NOT NULL,
        CONSTRAINT pk_resposta PRIMARY KEY (cd_pergunta, cd_resposta),
        CONSTRAINT fk_perguntafk_pergunta FOREIGN KEY (cd_pergunta) REFERENCES
public.tab_pergunta (cd_pergunta) ON UPDATE RESTRICT ON DELETE RESTRICT
    ) WITH OIDS;

```

2.9.1 Tabela Resultado

A tabela Resultado possui um histórico do desempenho dos colaboradores, ou seja, quais perguntas foram respondidas e qual a resposta informada, se foi a correta ou não. Está estruturada da seguinte maneira:

- tab_resultado: (nm_usuario,cd_pergunta,cd_resposta, id_status)

Os atributos cd_colaborador e nm_usuario referenciam a Tabela Usuário. Os atributos cd_pergunta, cd_resposta e id_status referenciam a Tabela Resposta.

2.9.2 Criação da Tabela

```

CREATE TABLE public.tab_resultado
(
    nm_usuario varchar(30) NOT NULL,
    cd_pergunta numeric(2) NOT NULL,
    cd_resposta numeric(3) NOT NULL,
    id_status varchar(1) NOT NULL,
    CONSTRAINT tab_resultado_pkey PRIMARY KEY (nm_usuario, cd_pergunta),
    CONSTRAINT fk_respostafk_resposta FOREIGN KEY (cd_pergunta, cd_resposta)
REFERENCES public.tab_resposta (cd_pergunta, cd_resposta) ON UPDATE RESTRICT ON
DELETE RESTRICT,
    CONSTRAINT fk_usuariofk_usuario FOREIGN KEY (nm_usuario) REFERENCES
public.tab_usuario (nm_usuario) ON UPDATE RESTRICT ON DELETE RESTRICT
) WITH OIDS;

```

Apêndice E – Código Fonte do Aplicativo Crescer

Classe Colaborador.java

```
package br.com.aprendizado_crm.bean;
public class Colaborador {

    private String cd_emp;
    private String cd_setor;
    private String cd_colaborador;
    private String nm_colaborador;
    private String idsexo;
    private String dt_admissao;
    private String cd_escolaridade;
    private String oid;

    public String getCd_colaborador() {
        return cd_colaborador;
    }
    public String getCd_emp() {
        return cd_emp;
    }
    public String getCd_escolaridade() {
        return cd_escolaridade;
    }
    public String getCd_setor() {
        return cd_setor;
    }
    public String getDt_admissao() {
        return dt_admissao;
    }
    public String getIdsexo() {
        return idsexo;
    }
    public String getNm_colaborador() {
        return nm_colaborador;
    }
    public String getOid() {
        return oid;
    }
    public void setCd_colaborador(String string) {
        cd_colaborador = string;
    }
    public void setCd_emp(String string) {
        cd_emp = string;
    }
    public void setCd_escolaridade(String string) {
        cd_escolaridade = string;
    }
    public void setCd_setor(String string) {
        cd_setor = string;
    }
}
```

```

    }
    public void setDt_admissao(String string) {
        dt_admissao = string;
    }
    public void setId_sexo(String string) {
        id_sexo = string;
    }
    public void setNm_colaborador(String string) {
        nm_colaborador = string;
    }
    public void setOid(String string) {
        oid = string;
    }
}

```

Classe Empresa.java

```

package br.com.aprendizado_crm.bean;
public class Empresa {

    private String cd_emp;
    private String nm_emp;
    private String oid;

    public String getCd_emp() {
        return cd_emp;
    }
    public void setCd_emp(String cd_emp) {
        this.cd_emp = cd_emp;
    }
    public String getNm_emp() {
        return nm_emp;
    }
    public void setNm_emp(String nm_emp) {
        this.nm_emp = nm_emp;
    }
    public String getOid() {
        return oid;
    }
    public void setOid(String oid) {
        this.oid = oid;
    }
}

```

Classe Escolaridade.java

```

package br.com.aprendizado_crm.bean;
public class Escolaridade {

    private String cd_escolaridade;
    private String ds_escolaridade;

```

```

private String oid;

public String getCd_escolaridade() {
    return cd_escolaridade;
}
public void setCd_escolaridade(String cd_escolaridade) {
    this.cd_escolaridade = cd_escolaridade;
}
public String getDs_escolaridade() {
    return ds_escolaridade;
}
public void setDs_escolaridade(String ds_escolaridade) {
    this.ds_escolaridade = ds_escolaridade;
}
public String getOid() {
    return oid;
}
public void setOid(String oid) {
    this.oid = oid;
}
}

```

Classe Pergunta.java

```

package br.com.aprendizado_crm.bean;
public class Pergunta {

    private String oid;
    private String cd_pergunta;
    private String ds_pergunta;

    public String getCd_pergunta() {
        return cd_pergunta;
    }
    public String getDs_pergunta() {
        return ds_pergunta;
    }
    public String getOid() {
        return oid;
    }
    public void setCd_pergunta(String string) {
        cd_pergunta = string;
    }
    public void setDs_pergunta(String string) {
        ds_pergunta = string;
    }
    public void setOid(String string) {
        oid = string;
    }
}

```

Classe Resposta.java

```
package br.com.aprendizado_crm.bean;
public class Resposta {

    private String cd_pergunta;
    private String cd_resposta;
    private String ds_resposta;
    private String id_status;
    private String oid;
    private String ds_pergunta;

    public String getCd_pergunta() {
        return cd_pergunta;
    }
    public String getCd_resposta() {
        return cd_resposta;
    }
    public String getDs_pergunta() {
        return ds_pergunta;
    }
    public String getDs_resposta() {
        return ds_resposta;
    }
    public String getId_status() {
        return id_status;
    }
    public String getOid() {
        return oid;
    }
    public void setCd_pergunta(String string) {
        cd_pergunta = string;
    }
    public void setCd_resposta(String string) {
        cd_resposta = string;
    }
    public void setDs_pergunta(String string) {
        ds_pergunta = string;
    }
    public void setDs_resposta(String string) {
        ds_resposta = string;
    }
    public void setId_status(String string) {
        id_status = string;
    }
    public void setOid(String string) {
        oid = string;
    }
}
```

Classe Resultado.java

```
package br.com.aprendizado_crm.bean;
public class Resultado {
    private String nm_usuario;
    private String cd_pergunta;
    private String cd_resposta;
    private String id_status;
    private String oid;

    private String ds_pergunta;
    private String ds_resposta;

    public String getCd_pergunta() {
        return cd_pergunta;
    }
    public void setCd_pergunta(String cd_pergunta) {
        this.cd_pergunta = cd_pergunta;
    }
    public String getCd_resposta() {
        return cd_resposta;
    }
    public void setCd_resposta(String cd_resposta) {
        this.cd_resposta = cd_resposta;
    }
    public String getDs_pergunta() {
        return ds_pergunta;
    }
    public void setDs_pergunta(String ds_pergunta) {
        this.ds_pergunta = ds_pergunta;
    }
    public String getDs_resposta() {
        return ds_resposta;
    }
    public void setDs_resposta(String ds_resposta) {
        this.ds_resposta = ds_resposta;
    }
    public String getId_status() {
        return id_status;
    }
    public void setId_status(String id_status) {
        this.id_status = id_status;
    }
    public String getNm_usuario() {
        return nm_usuario;
    }
    public void setNm_usuario(String nm_usuario) {
        this.nm_usuario = nm_usuario;
    }
    public String getOid() {
        return oid;
    }
}
```

```
    }  
    public void setOid(String oid) {  
        this.oid = oid;  
    }  
}
```

Classe Setor.java

```
package br.com.aprendizado_crm.bean;  
public class Setor {  
  
    private String cd_emp;  
    private String cd_setor;  
    private String ds_setor;  
    private String oid;  
  
    public String getCd_emp() {  
        return cd_emp;  
    }  
    public String getCd_setor() {  
        return cd_setor;  
    }  
    public String getDs_setor() {  
        return ds_setor;  
    }  
    public String getOid() {  
        return oid;  
    }  
    public void setCd_emp(String string) {  
        cd_emp = string;  
    }  
    public void setCd_setor(String string) {  
        cd_setor = string;  
    }  
    public void setDs_setor(String string) {  
        ds_setor = string;  
    }  
    public void setOid(String string) {  
        oid = string;  
    }  
}
```

Classe Usuário.java

```
package br.com.aprendizado_crm.bean;  
public class Usuario {  
  
    private String nm_usuario;  
    private String ds_senha;
```



```

private String oid;

public String getDs_senha() {
    return ds_senha;
}
public void setDs_senha(String ds_senha) {
    this.ds_senha = ds_senha;
}
public String getNm_usuario() {
    return nm_usuario;
}
public void setNm_usuario(String nm_usuario) {
    this.nm_usuario = nm_usuario;
}
public String getOid() {
    return oid;
}
public void setOid(String oid) {
    this.oid = oid;
}
}

```

Classe GravaEntrada.java

```

package br.com.aprendizado_crm.bean;
public class Estatistica_acesso {

    private String dt_acesso;
    private String nm_usuario;
    private String oid;

    public String getDt_acesso() {
        return dt_acesso;
    }
    public String getNm_usuario() {
        return nm_usuario;
    }
    public String getOid() {
        return oid;
    }
    public void setDt_acesso(String string) {
        dt_acesso = string;
    }
    public void setNm_usuario(String string) {
        nm_usuario = string;
    }
    public void setOid(String string) {
        oid = string;
    }
}

```

Classe Estatística_acesso.java

```
package br.com.aprendizado_crm.bean;
public class Estatistica_acesso {

    private String dt_acesso;
    private String nm_usuario;
    private String oid;

    public String getDt_acesso() {
        return dt_acesso;
    }
    public String getNm_usuario() {
        return nm_usuario;
    }
    public String getOid() {
        return oid;
    }
    public void setDt_acesso(String string) {
        dt_acesso = string;
    }
    public void setNm_usuario(String string) {
        nm_usuario = string;
    }
    public void setOid(String string) {
        oid = string;
    }
}
}
```

Classe Útil.java

```
package br.com.aprendizado_crm.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;
    }
}
}
```

Classe ColaboradorForm.java

```
package br.com.aprendizado_crm.form;
import java.io.Serializable;
import java.util.ArrayList;
import br.com.aprendizado_crm.bean.Colaborador;
```

```

public class ColaboradorForm extends PadraoForm implements Serializable {

    private static final long serialVersionUID = 1L;
    ArrayList colaborador_lista = new ArrayList();
    Colaborador colaborador = new Colaborador();
    Colaborador colaborador_pesq = new Colaborador();

    public Colaborador getColaborador() {
        return colaborador;
    }
    public ArrayList getColaborador_lista() {
        return colaborador_lista;
    }
    public Colaborador getColaborador_pesq() {
        return colaborador_pesq;
    }
    public void setColaborador(Colaborador colaborador) {
        this.colaborador = colaborador;
    }
    public void setColaborador_lista(ArrayList list) {
        colaborador_lista = list;
    }
    public void setColaborador_pesq(Colaborador colaborador) {
        colaborador_pesq = colaborador;
    }
}

```

Classe EmpresaForm.java

```

package br.com.aprendizado_crm.form;
import java.io.Serializable;
import java.util.ArrayList;
import br.com.aprendizado_crm.bean.Empresa;

public class EmpresaForm extends PadraoForm implements Serializable {

    private static final long serialVersionUID = 1L;
    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;
    }
}

```

Classe EscolaridadeForm.java

```

package br.com.aprendizado_crm.form;
import java.io.Serializable;
import java.util.ArrayList;
import br.com.aprendizado_crm.bean.Escolaridade;

public class EscolaridadeForm extends PadraoForm implements Serializable {

    private static final long serialVersionUID = 1L;
    ArrayList escolaridade_lista = new ArrayList();
    Escolaridade escolaridade = new Escolaridade();
    Escolaridade escolaridade_pesq = new Escolaridade();

    public Escolaridade getEscolaridade() {
        return escolaridade;
    }
    public ArrayList getEscolaridade_lista() {
        return escolaridade_lista;
    }
    public Escolaridade getEscolaridade_pesq() {
        return escolaridade_pesq;
    }
    public void setEscolaridade(Escolaridade escolaridade) {
        this.escolaridade = escolaridade;
    }
    public void setEscolaridade_lista(ArrayList list) {
        escolaridade_lista = list;
    }
    public void setEscolaridade_pesq(Escolaridade escolaridade) {
        escolaridade_pesq = escolaridade;
    }
}

```

Classe Estatística_acessoForm.java

```

package br.com.aprendizado_crm.form;
import java.io.Serializable;
import java.util.ArrayList;
import br.com.aprendizado_crm.bean.Estatistica_acesso;

```

```

public class Estatistica_acessoForm extends PadraoForm implements Serializable {

    private static final long serialVersionUID = 1L;
    ArrayList estatistica_acesso_lista = new ArrayList();
    Estatistica_acesso estatistica_acesso = new Estatistica_acesso();
    Estatistica_acesso estatistica_acesso_pesq = new Estatistica_acesso();

    public Estatistica_acesso getEstatistica_acesso() {
        return estatistica_acesso;
    }
    public ArrayList getEstatistica_acesso_lista() {
        return estatistica_acesso_lista;
    }
    public Estatistica_acesso getEstatistica_acesso_pesq() {
        return estatistica_acesso_pesq;
    }
    public void setEstatistica_acesso(Estatistica_acesso estatistica_acesso) {
        this.estatistica_acesso = estatistica_acesso;
    }
    public void setEstatistica_acesso_lista(ArrayList list) {
        estatistica_acesso_lista = list;
    }
    public void setEstatistica_acesso_pesq(Estatistica_acesso estatistica_acesso) {
        estatistica_acesso_pesq = estatistica_acesso;
    }
}

```

Classe PadraoForm.java

```

package br.com.aprendizado_crm.form;
import java.io.Serializable;
import org.apache.struts.validator.ValidatorActionForm;
import br.com.aprendizado_crm.util.Util;

public class PadraoForm extends ValidatorActionForm implements Serializable {

    private static final long serialVersionUID = 1L;
    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;
    private int nr_pergunta_atual;
    private int nr_resposta_atual;

    public int getNr_resposta_atual() {

```

```

        return nr_resposta_atual;
    }

    public void setNr_resposta_atual(int nr_resposta_atual) {
        this.nr_resposta_atual = nr_resposta_atual;
    }

    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;
    }
    public int getNr_pergunta_atual() {
        return nr_pergunta_atual;
    }
    public void setNr_pergunta_atual(int i) {
        nr_pergunta_atual = i;
    }
}

```

Classe PerguntaForm.java

```

package br.com.aprendizado_crm.form;
import java.io.Serializable;
import java.util.ArrayList;
import br.com.aprendizado_crm.bean.Pergunta;

public class PerguntaForm extends PadraoForm implements Serializable {

    private static final long serialVersionUID = 1L;
    ArrayList pergunta_lista = new ArrayList();
    Pergunta pergunta = new Pergunta();
    Pergunta pergunta_pesq = new Pergunta();

    public Pergunta getPergunta() {
        return pergunta;
    }
    public ArrayList getPergunta_lista() {
        return pergunta_lista;
    }
    public Pergunta getPergunta_pesq() {
        return pergunta_pesq;
    }
    public void setPergunta(Pergunta pergunta) {
        this.pergunta = pergunta;
    }
    public void setPergunta_lista(ArrayList list) {
        pergunta_lista = list;
    }
}

```



```

    }
    public void setPergunta_pesq(Pergunta pergunta) {
        pergunta_pesq = pergunta;
    }
}

```

Classe RespostaForm.java

```

package br.com.aprendizado_crm.form;
import java.io.Serializable;
import java.util.ArrayList;
import br.com.aprendizado_crm.bean.Resposta;

public class RespostaForm extends PadraoForm implements Serializable {

    private static final long serialVersionUID = 1L;
    ArrayList resposta_lista = new ArrayList();
    Resposta resposta = new Resposta();
    Resposta resposta_pesq = new Resposta();

    public Resposta getResposta() {
        return resposta;
    }
    public ArrayList getResposta_lista() {
        return resposta_lista;
    }
    public Resposta getResposta_pesq() {
        return resposta_pesq;
    }
    public void setResposta(Resposta resposta) {
        this.resposta = resposta;
    }
    public void setResposta_lista(ArrayList list) {
        resposta_lista = list;
    }
    public void setResposta_pesq(Resposta resposta) {
        resposta_pesq = resposta;
    }
}

```

Classe ResultadoForm.java

```

package br.com.aprendizado_crm.form;
import java.io.Serializable;
import java.util.ArrayList;
import br.com.aprendizado_crm.bean.Resultado;

public class ResultadoForm extends PadraoForm implements Serializable {

    private static final long serialVersionUID = 1L;
    ArrayList resultado_lista = new ArrayList();

```

```

Resultado resultado = new Resultado();
Resultado resultado_pesq = new Resultado();

public Resultado getResultado() {
    return resultado;
}
public ArrayList getResultado_lista() {
    return resultado_lista;
}
public Resultado getResultado_pesq() {
    return resultado_pesq;
}
public void setResultado(Resultado resultado) {
    this.resultado = resultado;
}
public void setResultado_lista(ArrayList list) {
    resultado_lista = list;
}
public void setResultado_pesq(Resultado resultado) {
    resultado_pesq = resultado;
}
}

```

Classe SetorForm.java

```

package br.com.aprendizado_crm.form;
import java.io.Serializable;
import java.util.ArrayList;
import br.com.aprendizado_crm.bean.Setor;

public class SetorForm extends PadraoForm implements Serializable {

    private static final long serialVersionUID = 1L;
    ArrayList setor_lista = new ArrayList();
    Setor setor = new Setor();
    Setor setor_pesq = new Setor();

    public Setor getSetor() {
        return setor;
    }
    public ArrayList getSetor_lista() {
        return setor_lista;
    }
    public Setor getSetor_pesq() {
        return setor_pesq;
    }
    public void setSetor(Setor setor) {
        this.setor = setor;
    }
    public void setSetor_lista(ArrayList list) {
        setor_lista = list;
    }
}

```

```

    }
    public void setSetor_pesq(Setor setor) {
        setor_pesq = setor;
    }
}

```

Classe UsuarioForm.java

```

package br.com.aprendizado_crm.form;
import java.io.Serializable;
import java.util.ArrayList;
import br.com.aprendizado_crm.bean.Usuario;

public class UsuarioForm extends PadraoForm implements Serializable {

    private static final long serialVersionUID = 1L;
    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;
    }
}

```

Classe ColaboradorSQL.java

```

package br.com.aprendizado_crm.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.aprendizado_crm.bean.Empresa;
import br.com.aprendizado_crm.bean.Escolaridade;

```

```

import br.com.aprendizado_crm.bean.Colaborador;
import br.com.aprendizado_crm.bean.Setor;
import br.com.aprendizado_crm.util.Util;

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

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

    public ArrayList carregaDisplay(ArrayList lista, Connection con)
    throws SQLException {
        ArrayList retorno = new ArrayList();
        ArrayList aux = new ArrayList();
        Colaborador colaborador = new Colaborador();
        retorno = lista;
        for (int i = 0; i < lista.size(); i++) {
            colaborador = (Colaborador) lista.get(i);
            //
            if (Util.trataNulo(colaborador.getCd_escolaridade()) != null) {
                Escolaridade escolaridade = new Escolaridade();
                EscolaridadeSQL escolaridadeSQL = new EscolaridadeSQL();
                escolaridade.setCd_escolaridade(colaborador.getCd_escolaridade());
                aux = escolaridadeSQL.select(con, 0, escolaridade, 1, "", "");
                if (aux.size() != 0) {
                    escolaridade = (Escolaridade) aux.get(0);
                    colaborador.setDs_escolaridade(escolaridade.getDs_escolaridade());
                }
            }
            //
            if (Util.trataNulo(colaborador.getCd_setor()) != null) {
                Setor setor = new Setor();
                SetorSQL setorSQL = new SetorSQL();
                setor.setCd_setor(colaborador.getCd_setor());
                aux = setorSQL.select(con, 0, setor, 1, "", "");
                if (aux.size() != 0) {
                    setor = (Setor) aux.get(0);
                    colaborador.setDs_setor(setor.getDs_setor());
                }
            }
            //
            if (Util.trataNulo(colaborador.getCd_emp()) != null) {
                Empresa empresa = new Empresa();
                EmpresaSQL empresaSQL = new EmpresaSQL();
                empresa.setCd_emp(colaborador.getCd_emp());
                aux = empresaSQL.select(con, 0, empresa, 1, "", "");
                if (aux.size() != 0) {
                    empresa = (Empresa) aux.get(0);
                    colaborador.setNm_emp(empresa.getNm_emp());
                }
            }
        }
    }
}

```

```

        }
    }
}
return retorno;
}

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 {
            Colaborador colaborador = new Colaborador();
            colaborador.setCd_emp(rs.getString("cd_emp"));
            colaborador.setCd_setor(rs.getString("cd_setor"));

            colaborador.setCd_colaborador(rs.getString("cd_colaborador"));
            colaborador.setNm_colaborador(rs.getString("nm_colaborador"));
            colaborador.setCd_escolaridade(rs.getString("cd_escolaridade"));
            colaborador.setDt_admissao(rs.getString("dt_admissao"));
            colaborador.setIdsexo(rs.getString("idsexo"));
            colaborador.setOid(rs.getString("oid"));
            retorno.add(colaborador);
        }
        nr_reg++;
    }
    return retorno;
}

public ArrayList select(
    Connection con,
    int p_pagina,
    Colaborador obj,
    int p_nr_reg,
    String p_oid,
    String p_mostra_contadores)
throws SQLException {
    ArrayList retorno = new ArrayList();
    String sql =
    "select "
    + " cd_emp"
    + ", cd_setor"
    + ", cd_colaborador"
    + ", cd_escolaridade"
    + ", nm_colaborador"
    + ", to_char(dt_admissao,'dd/mm/yyyy') as dt_admissao"

```

```

+ ", id_sexo"
+ ", oid "
+ "from tab_colaborador "
+ "where "
+ " id_sexo = coalesce(?,id_sexo) "
dt_admissao) " + " and dt_admissao = coalesce(to_date(?, 'dd/mm/yyyy'),
+ " and nm_colaborador = coalesce(?,nm_colaborador) "
+ " and cd_escolaridade = coalesce(?, cd_escolaridade) ";
if (Util.trataNulo(obj.getCd_colaborador()) == null) {
    sql = sql + "and cd_colaborador = coalesce(?, cd_colaborador)";
} else {
    sql = sql + "and cd_colaborador = ? ";
}

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

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 3 " + "offset ? " + "limit ? ";
PreparedStatement pstmt = con.prepareStatement(sql);
// Coloca os parâmetros
pstmt setObject(1, Util.trataNulo(obj.getId_sexo()));
pstmt setObject(2, Util.trataNulo(obj.getDt_admissao()));
pstmt setObject(3, Util.trataNulo(obj.getNm_colaborador()));
pstmt setObject(4, Util.trataNulo(obj.getCd_escolaridade()));
pstmt setObject(5, Util.trataNulo(obj.getCd_colaborador()));
pstmt setObject(6, Util.trataNulo(obj.getCd_setor()));
pstmt setObject(7, Util.trataNulo(obj.getCd_emp()));
pstmt setObject(8, Util.trataNulo(obj.getOid()));
ResultSet rs=coloca_contadores(pstmt,9,p_pagina,p_nr_reg
,p_oid,p_mostra_contadores);
return this.carregaObjetos(rs, p_oid);
}

public void insert(Colaborador obj, Connection con) throws SQLException {

```

```

String sql =
    " insert into tab_colaborador "
    + "( cd_emp, cd_setor,cd_colaborador,nm_colaborador,
cd_escolaridade, dt_admissao, id_sexo ) "
    + " values "
    + "(?,?,?,?,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_setor()));
pstmt.setObject(3, Util.trataNulo(obj.getCd_colaborador()));
pstmt.setObject(4, Util.trataNulo(obj.getNm_colaborador()));
pstmt.setObject(5, Util.trataNulo(obj.getCd_escolaridade()));
pstmt.setObject(6, Util.trataNulo(obj.getDt_admissao()));
pstmt.setObject(7, Util.trataNulo(obj.getId_sexo()));
//
pstmt.executeUpdate();
pstmt.close();
}

```

```

public void update(Colaborador obj, Connection con) throws SQLException {
String sql =
    " update tab_colaborador "
    + " set "
    + " nm_colaborador = ?"
    + ", cd_setor = ?"
    + ", cd_escolaridade = ?"
    + ", dt_admissao = to_date(?, 'dd/mm/yyyy'"
    + ", id_sexo = ?"
    + " where oid = ? ";
PreparedStatement pstmt = con.prepareStatement(sql);
pstmt.setObject(1, Util.trataNulo(obj.getNm_colaborador()));
pstmt.setObject(2, Util.trataNulo(obj.getCd_setor()));
pstmt.setObject(3, Util.trataNulo(obj.getCd_escolaridade()));
pstmt.setObject(4, Util.trataNulo(obj.getDt_admissao()));
pstmt.setObject(5, Util.trataNulo(obj.getId_sexo()));
pstmt.setObject(6, Util.trataNulo(obj.getOid()));
pstmt.executeUpdate();
pstmt.close();
}

```

```

public void delete(String p_oid, Connection con) throws SQLException {
String sql = "delete from tab_colaborador 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;
    }
}

```

Classe EmpresaSQL.java

```

package br.com.aprendizado_crm.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.aprendizado_crm.bean.Empresa;
import br.com.aprendizado_crm.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.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"
+ ", oid "
+ "from tab_empresa "
+ "where "
+ " 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.getNm_emp()));
pstmt.setObject(2, Util.trataNulo(obj.getCd_emp()));
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(Empresa obj, Connection con) throws SQLException {
String sql =
" insert into tab_empresa "
+ "( cd_emp,nm_emp ) "
+ " values "
+ "(?,?) ";
PreparedStatement pstmt = con.prepareStatement(sql);
//
pstmt.setObject(1, Util.trataNulo(obj.getCd_emp()));
pstmt.setObject(2, Util.trataNulo(obj.getNm_emp()));
//
pstmt.executeUpdate();
pstmt.close();
}

```

```

public void update(Empresa obj, Connection con) throws SQLException {
    String sql =
        " update tab_empresa "
        + " set "
        + " nm_emp = ?"
        + " where oid = ? ";
    PreparedStatement pstmt = con.prepareStatement(sql);
    pstmt.setObject(1, Util.trataNulo(obj.getNm_emp()));
    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 tab_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;
    }
}

```

Classe EscolaridadeSQL.java

```

package br.com.aprendizado_crm.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.aprendizado_crm.bean.Escolaridade;
import br.com.aprendizado_crm.util.Util;

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

    Category cat = Category.getInstance(EscolaridadeSQL.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 {
        Escolaridade escolaridade = new Escolaridade();
        escolaridade.setCd_escolaridade(rs.getString("cd_escolaridade"));
        escolaridade.setDs_escolaridade(rs.getString("ds_escolaridade"));
        escolaridade.setOid(rs.getString("oid"));
        retorno.add(escolaridade);
    }
    nr_reg++;
}
return retorno;
}

public ArrayList select(
    Connection con,
    int p_pagina,
    Escolaridade obj,
    int p_nr_reg,
    String p_oid,
    String p_mostra_contadores)
    throws SQLException {
    ArrayList retorno = new ArrayList();
    String sql =
        "select "
        + " cd_escolaridade"
        + ", ds_escolaridade"
        + ", oid "
        + "from tab_escolaridade "
        + "where "
        + " ds_escolaridade                = coalesce(?,ds_escolaridade)
";
    if (Util.trataNulo(obj.getCd_escolaridade()) == null) {
        sql = sql + "and cd_escolaridade                =                coalesce(?,
cd_escolaridade)";
    } else {
        sql = sql + "and cd_escolaridade                = ? ";
    }

    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_escolaridade()));
        pstmt.setObject(2, Util.trataNulo(obj.getCd_escolaridade()));
        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(Escolaridade obj, Connection con) throws SQLException {
        String sql =
            " insert into tab_escolaridade "
            + "( cd_escolaridade,ds_escolaridade ) "
            + " values "
            + "(?,?) ";
        PreparedStatement pstmt = con.prepareStatement(sql);
        //
        pstmt.setObject(1, Util.trataNulo(obj.getCd_escolaridade()));
        pstmt.setObject(2, Util.trataNulo(obj.getDs_escolaridade()));
        //
        pstmt.executeUpdate();
        pstmt.close();
    }

    public void update(Escolaridade obj, Connection con) throws SQLException {
        String sql =
            " update tab_escolaridade "
            + " set "
            + " ds_escolaridade = ?"
            + " where oid = ? ";
        PreparedStatement pstmt = con.prepareStatement(sql);
        pstmt.setObject(1, Util.trataNulo(obj.getDs_escolaridade()));
        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 tab_escolaridade 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;
    }
}

```

Classe Estatística_acessoSQL.java

```
package br.com.aprendizado_crm.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.aprendizado_crm.bean.Estatistica_acesso;
import br.com.aprendizado_crm.util.Util;

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

    Category cat = Category.getInstance(Estatistica_acessoSQL.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 {
                Estatistica_acesso estatistica_acesso = new Estatistica_acesso();
                estatistica_acesso.setNm_usuario(rs.getString("nm_usuario"));
                estatistica_acesso.setDt_acesso(rs.getString("dt_acesso"));
                estatistica_acesso.setOid(rs.getString("oid"));
                retorno.add(estatistica_acesso);
            }
            nr_reg++;
        }
        return retorno;
    }

    public ArrayList select(
        Connection con,
        int p_pagina,
        Estatistica_acesso obj,
        int p_nr_reg,
        String p_oid,
        String p_mostra_contadores)
        throws SQLException {
        ArrayList retorno = new ArrayList();
```



```

String sql =
"select "
+ " nm_usuario"
+ ", to_char(dt_acesso,'dd/mm/yyyy') as dt_acesso"
+ ", oid "
+ "from tab_estatistica_acesso "
+ "where "
+ " dt_acesso          = coalesce(to_date(?, 'dd/mm/yyyy'), dt_acesso) "
+ " and nm_usuario    = coalesce(?,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.getDt_acesso()));
pstmt.setObject(2, Util.trataNulo(obj.getNm_usuario()));
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(Estatistica_acesso obj, Connection con) throws SQLException {
String sql =
    " insert into tab_estatistica_acesso "
    + "( nm_usuario,dt_acesso ) "
    + " values "
    + "(?,?) ";
PreparedStatement pstmt = con.prepareStatement(sql);
//
pstmt.setObject(1, Util.trataNulo(obj.getNm_usuario()));
pstmt.setObject(2, Util.trataNulo(obj.getDt_acesso()));
//
pstmt.executeUpdate();
pstmt.close();
}

public void delete(String p_oid, Connection con) throws SQLException {
String sql = "delete from tab_estatistica_acesso 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;
    }
}

```

Classe PerguntaSQL.java

```

package br.com.aprendizado_crm.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.aprendizado_crm.bean.Pergunta;
import br.com.aprendizado_crm.util.Util;

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

    Category cat = Category.getInstance(PerguntaSQL.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 {
                Pergunta pergunta = new Pergunta();
                pergunta.setCd_pergunta(rs.getString("cd_pergunta"));
                pergunta.setDs_pergunta(rs.getString("ds_pergunta"));
                pergunta.setOid(rs.getString("oid"));
                retorno.add(pergunta);
            }
            nr_reg++;
        }
        return retorno;
    }

    public ArrayList select(
        Connection con,
        int p_pagina,
        Pergunta obj,

```

```

int p_nr_reg,
String p_oid,
String p_mostra_contadores)
throws SQLException {
ArrayList retorno = new ArrayList();
String sql =
"select "
+ " cd_pergunta"
+ ", ds_pergunta"
+ ", oid "
+ "from tab_pergunta "
+ "where "
+ " ds_pergunta                = coalesce(?,ds_pergunta) ";
if (Util.trataNulo(obj.getCd_pergunta()) == null) {
    sql = sql + "and cd_pergunta                = coalesce(?, cd_pergunta)";
} else {
    sql = sql + "and cd_pergunta                = ? ";
}

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_pergunta()));
pstmt.setObject(2, Util.trataNulo(obj.getCd_pergunta()));
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(Pergunta obj, Connection con) throws SQLException {
String sql =
    " insert into tab_pergunta "
    + "( cd_pergunta,ds_pergunta ) "
    + " values "
    + "(?,?) ";
PreparedStatement pstmt = con.prepareStatement(sql);
//
pstmt.setObject(1, Util.trataNulo(obj.getCd_pergunta()));
pstmt.setObject(2, Util.trataNulo(obj.getDs_pergunta()));
//
pstmt.executeUpdate();
pstmt.close();
}

```

```

public void update(Pergunta obj, Connection con) throws SQLException {
    String sql =
        " update tab_pergunta "
        + " set "
        + " ds_pergunta = ?"
        + " where oid = ? ";
    PreparedStatement pstmt = con.prepareStatement(sql);
    pstmt.setObject(1, Util.trataNulo(obj.getDs_pergunta()));
    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 tab_pergunta 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;
    }
}

```

Classe RespostaSQL.java

```
package br.com.aprendizado_crm.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.aprendizado_crm.bean.Pergunta;
import br.com.aprendizado_crm.bean.Resposta;
import br.com.aprendizado_crm.util.Util;

```

```

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

```

```

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

```

```

    public ArrayList carregaDisplay(ArrayList lista, Connection con)

```

```

throws SQLException {
    ArrayList retorno = new ArrayList();
    ArrayList aux = new ArrayList();
    Resposta resposta = new Resposta();
    retorno = lista;
    for (int i = 0; i < lista.size(); i++) {
        resposta = (Resposta) lista.get(i);

        if (Util.trataNulo(resposta.getCd_pergunta()) != null) {
            Pergunta pergunta = new Pergunta();
            PerguntaSQL perguntaSQL = new PerguntaSQL();
            pergunta.setCd_pergunta(resposta.getCd_pergunta());
            aux = perguntaSQL.select(con, 0, pergunta, 1, "", "");
            if (aux.size() != 0) {
                pergunta = (Pergunta) aux.get(0);
                resposta.setDs_pergunta(pergunta.getDs_pergunta());
            }
        }
    }
    return retorno;
}

```

```

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 {
            Resposta resposta = new Resposta();
            resposta.setCd_resposta(rs.getString("cd_resposta"));
            resposta.setDs_resposta(rs.getString("ds_resposta"));
            resposta.setCd_pergunta(rs.getString("cd_pergunta"));
            resposta.setId_status(rs.getString("id_status"));
            resposta.setOid(rs.getString("oid"));
            retorno.add(resposta);
        }
        nr_reg++;
    }
    return retorno;
}

```

```

public ArrayList select(
    Connection con,
    int p_pagina,
    Resposta obj,
    int p_nr_reg,

```

```

String p_oid,
String p_mostra_contadores)
throws SQLException {
ArrayList retorno = new ArrayList();
String sql =
"select "
+ " cd_resposta"
+ ", cd_pergunta"
+ ", ds_resposta"
+ ", id_status"
+ ", oid "
+ "from tab_resposta "
+ "where "
+ " id_status                = coalesce(?,id_status) "
+ " and ds_resposta         = coalesce(?,ds_resposta) ";

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

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

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

sql = sql + "order by 2,1 " + "offset ? " + "limit ? ";
PreparedStatement pstmt = con.prepareStatement(sql);
// Coloca os parâmetros
pstmt.setObject(1, Util.trataNulo(obj.getId_status()));
pstmt.setObject(2, Util.trataNulo(obj.getDs_resposta()));
pstmt.setObject(3, Util.trataNulo(obj.getCd_pergunta()));
pstmt.setObject(4, Util.trataNulo(obj.getCd_resposta()));
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(Resposta obj, Connection con) throws SQLException {
String sql =
    " insert into tab_resposta "

```



```

        + "( cd_pergunta,cd_resposta,ds_resposta, id_status ) "
        + " values "
        + "(?,?) ";
PreparedStatement pstmt = con.prepareStatement(sql);
//
pstmt.setObject(1, Util.trataNulo(obj.getCd_pergunta()));
pstmt.setObject(2, Util.trataNulo(obj.getCd_resposta()));
pstmt.setObject(3, Util.trataNulo(obj.getDs_resposta()));
pstmt.setObject(4, Util.trataNulo(obj.getId_status()));
//
pstmt.executeUpdate();
pstmt.close();
}

public void update(Resposta obj, Connection con) throws SQLException {
    String sql =
        " update tab_resposta "
        + " set "
        + " ds_resposta = ?"
        + ", id_status = ?"
        + " where oid = ? ";
    PreparedStatement pstmt = con.prepareStatement(sql);
    pstmt.setObject(1, Util.trataNulo(obj.getDs_resposta()));
    pstmt.setObject(2, Util.trataNulo(obj.getId_status()));
    pstmt.setObject(3, Util.trataNulo(obj.getOid()));
    pstmt.executeUpdate();
    pstmt.close();
}

public void delete(String p_oid, Connection con) throws SQLException {
    String sql = "delete from tab_resposta 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;
}
}
}

```

Classe ResultadoSQL.java

```

package br.com.aprendizado_crm.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.aprendizado_crm.bean.Pergunta;
import br.com.aprendizado_crm.bean.Resposta;
import br.com.aprendizado_crm.bean.Resultado;
import br.com.aprendizado_crm.util.Util;

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

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

    public ArrayList carregaDisplay(ArrayList lista, Connection con)
    throws SQLException {
        ArrayList retorno = new ArrayList();
        ArrayList aux = new ArrayList();
        Resultado resultado = new Resultado();
        retorno = lista;
        for (int i = 0; i < lista.size(); i++) {
            resultado = (Resultado) lista.get(i);

            if (Util.trataNulo(resultado.getCd_pergunta()) != null) {
                Pergunta pergunta = new Pergunta();
                PerguntaSQL perguntaSQL = new PerguntaSQL();
                pergunta.setCd_pergunta(resultado.getCd_pergunta());
                //aqui a gente pos "" e nao sabe se vai funcionar
                aux = perguntaSQL.select(con, 0, pergunta, 1, "", "");
                if (aux.size() != 0) {
                    pergunta = (Pergunta) aux.get(0);
                    resultado.setDs_pergunta(pergunta.getDs_pergunta());
                }
            }
            //
            if (Util.trataNulo(resultado.getCd_resposta()) != null) {
                Resposta resposta = new Resposta();
                RespostaSQL respostaSQL = new RespostaSQL();
                resposta.setCd_resposta(resultado.getCd_resposta());
                resposta.setCd_pergunta(resultado.getCd_pergunta());
                //aqui a gente pos "" e nao sabe se vai funcionar
                aux = respostaSQL.select(con, 0, resposta, 1, "", "");
                if (aux.size() != 0) {
                    resposta = (Resposta) aux.get(0);
                    resultado.setDs_resposta(resposta.getDs_resposta());
                }
            }
        }
        return retorno;
    }
}

```

```

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 {
            Resultado resultado = new Resultado();

            resultado.setCd_resposta(rs.getString("cd_resposta"));
            resultado.setNm_usuario(rs.getString("nm_usuario"));
            resultado.setCd_pergunta(rs.getString("cd_pergunta"));
            resultado.setId_status(rs.getString("id_status"));
            resultado.setOid(rs.getString("oid"));
            retorno.add(resultado);
        }
        nr_reg++;
    }
    return retorno;
}

```

```

public ArrayList select(
    Connection con,
    int p_pagina,
    Resultado obj,
    int p_nr_reg,
    String p_oid,
    String p_mostra_contadores)
    throws SQLException {
    ArrayList retorno = new ArrayList();
    String sql =
    "select "
    + " cd_resposta"
    + ", cd_pergunta"
    + ", nm_usuario"
    + ", id_status"
    + ", oid "
    + "from tab_resultado "
    + "where "
    + " id_status                = coalesce(?,id_status) "
    + " and nm_usuario           = coalesce(?,nm_usuario) ";

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

```

```

    }

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

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

    sql = sql + "order by 3,2,1 " + "offset ? " + "limit ? ";
    PreparedStatement pstmt = con.prepareStatement(sql);
    // Coloca os parâmetros
    pstmt setObject(1, Util.trataNulo(obj.getId_status()));
    pstmt setObject(2, Util.trataNulo(obj.getNm_usuario()));
    pstmt setObject(3, Util.trataNulo(obj.getCd_pergunta()));
    pstmt setObject(4, Util.trataNulo(obj.getCd_resposta()));
    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(Resultado obj, Connection con) throws SQLException {
    String sql =
        " insert into tab_resultado "
        + "( cd_pergunta,cd_resposta,nm_usuario,id_status ) "
        + " values "
        + "(?,?,?,?) ";
    PreparedStatement pstmt = con.prepareStatement(sql);
    //
    pstmt setObject(1, Util.trataNulo(obj.getCd_pergunta()));
    pstmt setObject(2, Util.trataNulo(obj.getCd_resposta()));
    pstmt setObject(3, Util.trataNulo(obj.getNm_usuario()));
    pstmt setObject(4, Util.trataNulo(obj.getId_status()));
    //
    pstmt.executeUpdate();
    pstmt.close();
}

/*public void update(Resultado obj, Connection con) throws SQLException {
    String sql =
        " update tab_resultado "
        + " set "
        + " nm_usuario = ?"
        + ", id_status = ?"

```

```

        + " where oid = ? ";
        PreparedStatement pstmt = con.prepareStatement(sql);
        pstmt setObject(1, Util.trataNulo(obj.getNm_usuario()));
        pstmt setObject(2, Util.trataNulo(obj.getId_status()));
        pstmt setObject(3, Util.trataNulo(obj.getOid()));
        pstmt.executeUpdate();
        pstmt.close();
    }*/

    public void delete(String p_oid, Connection con) throws SQLException {
        String sql = "delete from tab_resultado 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;
}
}

```

Classe SetorSQL.java

```

package br.com.aprendizado_crm.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.aprendizado_crm.bean.Empresa;
import br.com.aprendizado_crm.bean.Setor;
import br.com.aprendizado_crm.util.Util;

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

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

    public ArrayList carregaDisplay(ArrayList lista, Connection con)
    throws SQLException {
        ArrayList retorno = new ArrayList();
        ArrayList aux = new ArrayList();
        Setor setor = new Setor();

```

```

returno = lista;
for (int i = 0; i < lista.size(); i++) {
    setor = (Setor) lista.get(i);

    if (Util.trataNulo(setor.getCd_emp()) != null) {
        Empresa empresa = new Empresa();
        EmpresaSQL empresaSQL = new EmpresaSQL();
        empresa.setCd_emp(setor.getCd_emp());
        aux = empresaSQL.select(con, 0, empresa, 1, "", "");
        if (aux.size() != 0) {
            empresa = (Empresa) aux.get(0);
            setor.setNm_emp(empresa.getNm_emp());
        }
    }
}
return returno;
}

private ArrayList carregaObjetos(ResultSet rs, String p_oid)
throws SQLException {
    ArrayList returno = 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 {
            Setor setor = new Setor();
            setor.setCd_emp(rs.getString("cd_emp"));
            setor.setCd_setor(rs.getString("cd_setor"));
            setor.setDs_setor(rs.getString("ds_setor"));
            setor.setOid(rs.getString("oid"));
            returno.add(setor);
        }
        nr_reg++;
    }
    return returno;
}

public ArrayList select(
    Connection con,
    int p_pagina,
    Setor obj,
    int p_nr_reg,
    String p_oid,
    String p_mostra_contadores)
throws SQLException {
    ArrayList returno = new ArrayList();
    String sql =
    "select "

```



```

+ " cd_emp"
+ ", cd_setor"
+ ", ds_setor"
+ ", oid "
+ "from tab_setor "
+ "where "
+ " ds_setor = coalesce(?,ds_setor) ";
if (Util.trataNulo(obj.getCd_setor()) == null) {
    sql = sql + "and cd_setor = coalesce(?, cd_setor)";
} else {
    sql = sql + "and cd_setor = ? ";
}

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 1,2 " + "offset ? " + "limit ? ";
PreparedStatement pstmt = con.prepareStatement(sql);
// Coloca os parâmetros
pstmt setObject(1, Util.trataNulo(obj.getDs_setor()));
pstmt setObject(2, Util.trataNulo(obj.getCd_setor()));
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(Setor obj, Connection con) throws SQLException {
String sql =
    " insert into tab_setor "
    + "( cd_emp,cd_setor,ds_setor ) "
    + " values "
    + "(?,?,?) ";
PreparedStatement pstmt = con.prepareStatement(sql);
//
pstmt setObject(1, Util.trataNulo(obj.getCd_emp()));
pstmt setObject(2, Util.trataNulo(obj.getCd_setor()));
pstmt setObject(3, Util.trataNulo(obj.getDs_setor()));
//
pstmt.executeUpdate();
pstmt.close();
}

```

```

}

public void update(Setor obj, Connection con) throws SQLException {
    String sql =
        " update tab_setor "
        + " set "
        + " ds_setor = ?"
        + " where oid = ? ";
    PreparedStatement pstmt = con.prepareStatement(sql);
    pstmt setObject(1, Util.trataNulo(obj.getDs_setor()));
    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 tab_setor 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;
    }
}

```

Classe UsuarioSQL.java

```

package br.com.aprendizado_crm.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.aprendizado_crm.bean.Usuario;
import br.com.aprendizado_crm.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.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 "
            + ", oid "
            + "from tab_usuario "
            + "where 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.getDs_senha()));
}

```

```

        pstmt.setObject(2, Util.trataNulo(obj.getNm_usuario()));
        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(Usuario obj, Connection con) throws SQLException {
        String sql =
            " insert into tab_usuario "
            + " ( nm_usuario,ds_senha ) "
            + " values "
            + "(?,?) ";
        PreparedStatement pstmt = con.prepareStatement(sql);
        //
        pstmt.setObject(1, Util.trataNulo(obj.getNm_usuario()));
        pstmt.setObject(2, Util.trataNulo(obj.getDs_senha()));
        //
        pstmt.executeUpdate();
        pstmt.close();
    }

    public void update(Usuario obj, Connection con) throws SQLException {
        String sql =
            " update tab_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 tab_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;
}
}

```

Css_crescer.css

```
.linha_1 {
  font-family: Verdana, Arial, Helvetica, sans-serif;
  font-size: 8pt;
  color: #000000;
  background-color: #F3F3F3;
}

.linha_2 {
  font-family: Verdana, Arial, Helvetica, sans-serif;
  font-size: 8pt;
  color: #000000;
  background-color: #ffffff;
}
body {
  scrollbar-track-color: #ffffff;
}
.Login {
  font-family: Verdana, Arial, Helvetica, sans-serif;
  font-size: 8pt;
  color: #ffffff
}

.TxtGeral {
  font-family: Verdana, Arial, Helvetica, sans-serif;
  font-size: 8pt;
  color: #000000;
}

.TxtTitulo {
  font-family: Verdana, Arial, Helvetica, sans-serif;
  font-size: 8pt;
  font-weight: bold;
  color: #63659C;
}

.TxtTitulo2 {
  font-family: Verdana, Arial, Helvetica, sans-serif;
  font-size: 8pt;
  font-weight: bold;
  color: #ffffff;
}

.TxtTopo {
  font-family: Verdana, Arial, Helvetica, sans-serif;
  font-size: 13pt;
  font-weight: bold;
  color: #ffffff;
}
```

```
.TxtTituloSub {
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 8pt;
    font-weight: bold;
    color: #000000;
    text-decoration: underline;
}

a:link {
    text-decoration: none;
    font-family: Verdana, Arial, Helvetica, sans-serif;
    color: #000000;
    font-size: 8pt;
}

a:visited {
    text-decoration: none;
    font-family: Verdana, Arial, Helvetica, sans-serif;
    color: #000000;
    font-size: 8pt;
}

a:hover{
    color: #000000;
    text-decoration: underline;
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 8pt;
}

.button
{
    BORDER-RIGHT: #000000 1px solid;
    BORDER-TOP: #000000 1px solid;
    BORDER-LEFT: #000000 1px solid;
    BORDER-BOTTOM: #000000 1px solid;
    text-decoration: none;
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 10px;
    font-weight: bold;
    color: #ffffff;
    BACKGROUND-COLOR: #63659C;
}

.campoCnpj {
    color: #000000;
    text-decoration: none;
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 10px;
    background-color: #FFFFFF;
    border: 1px #000000 solid;
    padding-top: 1px;
    padding-right: 1px;
    padding-bottom: 1px;
    padding-left: 1px;
}
```



```
margin-top: 1px;
margin-right: 1px;
margin-bottom: 1px;
margin-left: 1px;
width: 120px;
}
.campo {
color: #000000;
text-decoration: none;
font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 10px;
background-color: #FFFFFF;
border: 1px #000000 solid;
padding-top: 1px;
padding-right: 1px;
padding-bottom: 1px;
padding-left: 1px;
margin-top: 1px;
margin-right: 1px;
margin-bottom: 1px;
margin-left: 1px;
}

.endereco {
color: #000000;
width: 320px;
text-decoration: none;
font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 10px;
background-color: #FFFFFF;
scrollbar-track-color: #ffffff;
border: 1px #000000 solid;
padding-top: 1px;
padding-right: 1px;
padding-bottom: 1px;
padding-left: 1px;
margin-top: 1px;
margin-right: 1px;
margin-bottom: 1px;
margin-left: 1px;
}

.campoCodigo {
color: #000000;
width: 30px;
text-decoration: none;
font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 10px;
background-color: #FFFFFF;
border: 1px #000000 solid;
padding-top: 1px;
```

```
padding-right: 1px;
padding-bottom: 1px;
padding-left: 1px;
margin-top: 1px;
margin-right: 1px;
margin-bottom: 1px;
margin-left: 1px;
}
```

```
    .campoCodigo2 {
color: #000000;
width: 40px;
text-decoration: none;
font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 10px;
background-color: #FFFFFF;
border: 1px #000000 solid;
padding-top: 1px;
padding-right: 1px;
padding-bottom: 1px;
padding-left: 1px;
margin-top: 1px;
margin-right: 1px;
margin-bottom: 1px;
margin-left: 1px;
}
```

```
    .campoPeq {
color: #000000;
width: 20px;
text-decoration: none;
font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 10px;
background-color: #FFFFFF;
border: 1px #000000 solid;
padding-top: 1px;
padding-right: 1px;
padding-bottom: 1px;
padding-left: 1px;
margin-top: 1px;
margin-right: 1px;
margin-bottom: 1px;
margin-left: 1px;
}
```

```
    .campoMat {
color: #000000;
width: 50px;
text-decoration: none;
font-family: Verdana, Arial, Helvetica, sans-serif;
```

```
font-size: 10px;
background-color: #FFFFFF;
border: 1px #000000 solid;
padding-top: 1px;
padding-right: 1px;
padding-bottom: 1px;
padding-left: 1px;
margin-top: 1px;
margin-right: 1px;
margin-bottom: 1px;
margin-left: 1px;
}
```

```
.campoDesc {
color: #000000;
width: 210px;
text-decoration: none;
font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 10px;
background-color: #FFFFFF;
border: 1px #000000 solid;
padding-top: 1px;
padding-right: 1px;
padding-bottom: 1px;
padding-left: 1px;
margin-top: 1px;
margin-right: 1px;
margin-bottom: 1px;
margin-left: 1px;
}
```

```
.campoCep {
color: #000000;
width: 70px;
text-decoration: none;
font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 10px;
background-color: #FFFFFF;
border: 1px #000000 solid;
padding-top: 1px;
padding-right: 1px;
padding-bottom: 1px;
padding-left: 1px;
margin-top: 1px;
margin-right: 1px;
margin-bottom: 1px;
margin-left: 1px;
}
```

```
.campoData {
    color: #000000;
    width: 78px;
    text-decoration: none;
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 10px;
    background-color: #FFFFFF;
    border: 1px #000000 solid;
    padding-top: 1px;
    padding-right: 1px;
    padding-bottom: 1px;
    padding-left: 1px;
    margin-top: 1px;
    margin-right: 1px;
    margin-bottom: 1px;
    margin-left: 1px;
}
.campoTelefone {
    color: #000000;
    width: 98px;
    text-decoration: none;
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 10px;
    background-color: #FFFFFF;
    border: 1px #000000 solid;
    padding-top: 1px;
    padding-right: 1px;
    padding-bottom: 1px;
    padding-left: 1px;
    margin-top: 1px;
    margin-right: 1px;
    margin-bottom: 1px;
    margin-left: 1px;
}
```

Classe ColaboradorAction.java

```
package br.com.aprendizado_crm.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.aprendizado_crm.bean.Colaborador;
import br.com.aprendizado_crm.form.ColaboradorForm;
import br.com.aprendizado_crm.sql.ColaboradorSQL;
import br.com.aprendizado_crm.util.Util;

public class ColaboradorAction extends LookupDispatchAction {

    public Category cat = Category.getInstance(ColaboradorAction.class.getName());

    protected Map getKeyMethodMap(){
        Map map = new HashMap();
        map.put("aprendizado_crm.entra_inserir","entra_inserir");
        map.put("aprendizado_crm.entra_consultar","entra_consultar");
        map.put("aprendizado_crm.entra_alterar","entra_alterar");
        map.put("aprendizado_crm.inserir","inserir");
        map.put("aprendizado_crm.apagar","apagar");
        map.put("aprendizado_crm.alterar","alterar");
        map.put("aprendizado_crm.lista","lista");
        map.put("aprendizado_crm.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("colaborador "+usuario_form.getUsuario().getCd_emp());

        ColaboradorForm beanForm = (ColaboradorForm) form;
        ColaboradorSQL beanSQL = new ColaboradorSQL();
        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.setColaborador_lista(
            beanSQL.carregaDisplay(
                beanSQL.select(
                    con,
                    p_pagina,
                    beanForm.getColaborador_pesq(),
                    beanForm.getNr_reg(),
                    "", ""), con));
        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) {
    ColaboradorForm beanForm = (ColaboradorForm) 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");
    ColaboradorForm beanForm = (ColaboradorForm) form;
    ColaboradorSQL beanSQL = new ColaboradorSQL();
    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");
        }

        Colaborador v_colaborador = new Colaborador();
        v_colaborador.setCd_emp(beanForm.getColaborador().getCd_emp());

        v_colaborador.setCd_colaborador(beanForm.getColaborador().getCd_colaborador());
        beanForm.setColaborador_lista(
            beanSQL.select(con, 0, v_colaborador, 1, "", "N"));
        if (beanForm.getColaborador_lista().size() > 0) {
            throw new ExcecaoGeral(
                new ActionMessage("aprendizado_crm.inserir.erro"));
        } else {
            beanSQL.insert(beanForm.getColaborador(), con);
            con.commit();
            beanForm.setColaborador_lista(

                beanSQL.carregaDisplay(beanSQL.select(con,0,beanForm.getColaborador(),1,"", "N"),con));
            beanForm.setColaborador_pesq(new Colaborador());
            beanForm.setColaborador_lista(
                beanSQL.carregaDisplay(beanSQL.select(
                    con,
                    0,
                    beanForm.getColaborador_pesq(),
                    2000000000,
                    ((Colaborador)
                        beanForm.getColaborador_lista().get(0))
                        .getOid(), "N"),con));

            //
            beanForm.setNr_reg(1);
            p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
            beanForm.setColaborador_lista(
                beanSQL.carregaDisplay(beanSQL.select(
                    con,
                    p_pagina,
                    beanForm.getColaborador_pesq(),
                    beanForm.getNr_reg(),
                    "", ""),con));

```

```

        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 {
    ColaboradorForm beanForm = (ColaboradorForm) form;
    beanForm.setColaborador(new Colaborador());
    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");
    ColaboradorForm beanForm = (ColaboradorForm) form;
    ColaboradorSQL beanSQL = new ColaboradorSQL();
    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");
        }

        Colaborador v_colaborador = new Colaborador();
        v_colaborador.setOid(oid);
        v_colaborador =
            (Colaborador) beanSQL.select(con, 0, v_colaborador, 1,
"", "N").get(0);

        beanForm.setColaborador_lista(
            beanSQL.select(
                con,
                0,
                beanForm.getColaborador_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");
    ColaboradorForm beanForm = (ColaboradorForm) form;
    try {
        ColaboradorSQL beanSQL = new ColaboradorSQL();

        String oid = "-1";
        if (Util.trataNulo(request.getParameter("oid")) != null) {
            oid = request.getParameter("oid");
        }
    }

```

```

        Colaborador v_colaborador = new Colaborador();
        v_colaborador.setOid(oid);

        beanForm.setColaborador_lista(beanSQL.select(con,0,v_colaborador,1,"","N"));
        beanForm.setColaborador((Colaborador)
beanForm.getColaborador_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");
    ColaboradorForm beanForm = (ColaboradorForm) form;
    ColaboradorSQL beanSQL = new ColaboradorSQL();
    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.getColaborador(), con);
            con.commit();

            beanForm.setColaborador_lista(
                beanSQL.select(con, 0, beanForm.getColaborador(), 1, "", "N"));

```

```

        if (beanForm.getColaborador_lista().size() == 0) {
            throw new ExcecaoGeral(
                new ActionMessage("aprendizado_crm.alterar.erro"));
        }

        beanForm.setColaborador_lista(
            beanSQL.select(
                con,
                0,
                beanForm.getColaborador_pesq(),
                2000000000,
                beanForm.getColaborador().getOid(), "N"));

        //
        beanForm.setNr_reg(1);
        p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
        beanForm.setColaborador_lista(
            beanSQL.carregaDisplay(beanSQL.select(
                con,
                p_pagina,
                beanForm.getColaborador_pesq(),
                beanForm.getNr_reg(),
                "", ""), con));
        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"));
}
}

```

Classe EmpresaAction.java

```

package br.com.aprendizado_crm.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.aprendizado_crm.bean.Empresa;
import br.com.aprendizado_crm.form.EmpresaForm;
import br.com.aprendizado_crm.sql.EmpresaSQL;
import br.com.aprendizado_crm.util.Util;

public class EmpresaAction extends LookupDispatchAction {

    public Category cat = Category.getInstance(EmpresaAction.class.getName());

    protected Map getKeyMethodMap(){
        Map map = new HashMap();
        map.put("aprendizado_crm.entra_inserir","entra_inserir");
        map.put("aprendizado_crm.entra_consultar","entra_consultar");
        map.put("aprendizado_crm.entra_alterar","entra_alterar");
        map.put("aprendizado_crm.inserir","inserir");
        map.put("aprendizado_crm.apagar","apagar");
        map.put("aprendizado_crm.alterar","alterar");
        map.put("aprendizado_crm.lista","lista");
        map.put("aprendizado_crm.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(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.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("aprendizado_crm.inserir.erro"));
        } else {
            beanSQL.insert(beanForm.getEmpresa(), con);
            con.commit();
            beanForm.setEmpresa_lista(
                beanSQL.select(
                    con,
                    0,
                    beanForm.getEmpresa_pesq(),
                    2000000000,
                    ((Empresa)
                        beanForm.getEmpresa_lista().get(0))
                        .getOid(), "N"));
            //
            beanForm.setNr_reg(10);
            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("aprendizado_crm.alterar.erro"));
}

beanForm.setEmpresa_lista(
    beanSQL.select(
        con,
        0,
        beanForm.getEmpresa_pesq(),
        2000000000,
        beanForm.getEmpresa().getOid(), "N"));

//
beanForm.setNr_reg(10);
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"));
}
}

```

Classe EscolaridadeAction.java

```

package br.com.aprendizado_crm.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.aprendizado_crm.bean.Escolaridade;
import br.com.aprendizado_crm.form.EscolaridadeForm;
import br.com.aprendizado_crm.sql.EscolaridadeSQL;
import br.com.aprendizado_crm.util.Util;

public class EscolaridadeAction extends LookupDispatchAction {

    public Category cat = Category.getInstance(EscolaridadeAction.class.getName());

    protected Map getKeyMethodMap(){
        Map map = new HashMap();
        map.put("aprendizado_crm.entra_inserir","entra_inserir");
        map.put("aprendizado_crm.entra_consultar","entra_consultar");
        map.put("aprendizado_crm.entra_alterar","entra_alterar");
        map.put("aprendizado_crm.inserir","inserir");
        map.put("aprendizado_crm.apagar","apagar");
        map.put("aprendizado_crm.alterar","alterar");
        map.put("aprendizado_crm.lista","lista");
        map.put("aprendizado_crm.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("escolaridade "+usuario_form.getUsuario().getCd_escolaridade());

EscolaridadeForm beanForm = (EscolaridadeForm) form;
EscolaridadeSQL beanSQL = new EscolaridadeSQL();
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.setEscolaridade_lista(
        beanSQL.select(
            con,
            p_pagina,
            beanForm.getEscolaridade_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) {
    EscolaridadeForm beanForm = (EscolaridadeForm) 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");
EscolaridadeForm beanForm = (EscolaridadeForm) form;
EscolaridadeSQL beanSQL = new EscolaridadeSQL();
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");
        }

        Escolaridade v_escolaridade = new Escolaridade();

v_escolaridade.setCd_escolaridade(beanForm.getEscolaridade().getCd_escolaridade());
        beanForm.setEscolaridade_lista(
            beanSQL.select(con, 0, v_escolaridade, 1, "", "N"));
        if (beanForm.getEscolaridade_lista().size() > 0) {
            throw new ExcecaoGeral(
                new ActionMessage("aprendizado_crm.inserir.erro"));
        } else {
            beanSQL.insert(beanForm.getEscolaridade(), con);
            con.commit();
            beanForm.setEscolaridade_lista(

beanSQL.select(con,0,beanForm.getEscolaridade(),1,"","N"));
            beanForm.setEscolaridade_pesq(new Escolaridade());
            beanForm.setEscolaridade_lista(
                beanSQL.select(
                    con,
                    0,
                    beanForm.getEscolaridade_pesq(),
                    2000000000,
                    ((Escolaridade)
beanForm.getEscolaridade_lista().get(0))
                        .getOid(), "N"));
            //
            beanForm.setNr_reg(10);
            p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());

```

```

        beanForm.setEscolaridade_lista(
            beanSQL.select(
                con,
                p_pagina,
                beanForm.getEscolaridade_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 {
    EscolaridadeForm beanForm = (EscolaridadeForm) form;
    beanForm.setEscolaridade(new Escolaridade());
    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");
    EscolaridadeForm beanForm = (EscolaridadeForm) form;
    EscolaridadeSQL beanSQL = new EscolaridadeSQL();
    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");
    }

    Escolaridade v_escolaridade = new Escolaridade();
    v_escolaridade.setOid(oid);
    v_escolaridade =
        (Escolaridade) beanSQL.select(con, 0, v_escolaridade, 1,
"", "N").get(0);

    beanForm.setEscolaridade_lista(
        beanSQL.select(
            con,
            0,
            beanForm.getEscolaridade_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");
    EscolaridadeForm beanForm = (EscolaridadeForm) form;

```

```

try {
    EscolaridadeSQL beanSQL = new EscolaridadeSQL();

    String oid = "-1";
    if (Util.trataNulo(request.getParameter("oid")) != null) {
        oid = request.getParameter("oid");
    }

    Escolaridade v_escolaridade = new Escolaridade();
    v_escolaridade.setOid(oid);

    beanForm.setEscolaridade_lista(beanSQL.select(con,0,v_escolaridade,1,"","N"));
    beanForm.setEscolaridade((Escolaridade)
beanForm.getEscolaridade_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");
    EscolaridadeForm beanForm = (EscolaridadeForm) form;
    EscolaridadeSQL beanSQL = new EscolaridadeSQL();
    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.getEscolaridade(), con);
    con.commit();

    beanForm.setEscolaridade_lista(
        beanSQL.select(con, 0, beanForm.getEscolaridade(), 1, "",
"N"));

    if (beanForm.getEscolaridade_lista().size() == 0) {
        throw new ExcecaoGeral(
            new ActionMessage("aprendizado_crm.alterar.erro"));
    }

    beanForm.setEscolaridade_lista(
        beanSQL.select(
            con,
            0,
            beanForm.getEscolaridade_pesq(),
            2000000000,
            beanForm.getEscolaridade().getOid(), "N"));

    //
    beanForm.setNr_reg(10);
    p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
    beanForm.setEscolaridade_lista(
        beanSQL.select(
            con,
            p_pagina,
            beanForm.getEscolaridade_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"));

```

```
}  
}
```

Classe Estatística_ acessoAction.java

```
package br.com.aprendizado_crm.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.aprendizado_crm.bean.Estatistica_acesso;  
import br.com.aprendizado_crm.form.Estatistica_acessoForm;  
import br.com.aprendizado_crm.sql.Estatistica_acessoSQL;  
import br.com.aprendizado_crm.util.Util;  
  
public class Estatistica_acessoAction extends LookupDispatchAction {  
  
    public Category cat = Category.getInstance(Estatistica_acessoAction.class.getName());  
  
    protected Map getKeyMethodMap(){  
        Map map = new HashMap();  
        map.put("aprendizado_crm.entra_inserir","entra_inserir");  
        map.put("aprendizado_crm.entra_consultar","entra_consultar");  
        map.put("aprendizado_crm.entra_alterar","entra_alterar");  
        map.put("aprendizado_crm.inserir","inserir");  
        map.put("aprendizado_crm.apagar","apagar");  
        map.put("aprendizado_crm.alterar","alterar");  
        map.put("aprendizado_crm.lista","lista");  
        map.put("aprendizado_crm.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");

        Estatistica_acessoForm beanForm = (Estatistica_acessoForm) form;
        Estatistica_acessoSQL beanSQL = new Estatistica_acessoSQL();
        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.setEstatistica_acesso_lista(
                beanSQL.select(
                    con,
                    p_pagina,
                    beanForm.getEstatistica_acesso_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) {
        Estatistica_acessoForm beanForm = (Estatistica_acessoForm) form;
        beanForm.setMostraConsulta("S");
        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");
        Estatistica_acessoForm beanForm = (Estatistica_acessoForm) form;
        Estatistica_acessoSQL beanSQL = new Estatistica_acessoSQL();
        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");
            }

            Estatistica_acesso v_estatistica_acesso = new Estatistica_acesso();
            v_estatistica_acesso.setOid(oid);
            v_estatistica_acesso =
                (Estatistica_acesso) beanSQL.select(con, 0, v_estatistica_acesso, 1,
            "", "N").get(0);

            beanForm.setEstatistica_acesso_lista(
                beanSQL.select(
                    con,
                    0,
                    beanForm.getEstatistica_acesso_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");
    Estatistica_acessoForm beanForm = (Estatistica_acessoForm) form;
    try {
        Estatistica_acessoSQL beanSQL = new Estatistica_acessoSQL();

        String oid = "-1";
        if (Util.trataNulo(request.getParameter("oid")) != null) {
            oid = request.getParameter("oid");
        }

        Estatistica_acesso v_estatistica_acesso = new Estatistica_acesso();
        v_estatistica_acesso.setOid(oid);

        beanForm.setEstatistica_acesso_lista(beanSQL.select(con,0,v_estatistica_acesso,1,"", "N"));
        beanForm.setEstatistica_acesso((Estatistica_acesso)
        beanForm.getEstatistica_acesso_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"));
}
}

```

Classe ExcecaoGeral.java

```

package br.com.aprendizado_crm.action;

import org.apache.struts.action.ActionMessage;;

public class ExcecaoGeral extends Exception {
    /**
     *
     */
    private static final long serialVersionUID = 1L;
    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;
    }
}

```

Classe LoginAction.java

```

package br.com.aprendizado_crm.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.aprendizado_crm.bean.Grava_entrada;
import br.com.aprendizado_crm.bean.Usuario;
import br.com.aprendizado_crm.form.UsuarioForm;
import br.com.aprendizado_crm.sql.UsuarioSQL;
import br.com.aprendizado_crm.util.Util;

public class LoginAction extends LookupDispatchAction {

```

```

public Category cat = Category.getInstance(LoginAction.class.getName());

Grava_entrada grava_entrada = new Grava_entrada();

protected Map getKeyMethodMap() {
    Map map = new HashMap();
    map.put("aprendizado_crm.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/dsaprendizado_crm");
        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("aprendizado_crm.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,
                v_usuario,1,"","N").get(0));
            }
        }
    }
}

```

```

        request.getSession().setAttribute("usuario",
beanForm.getUsuario().getNm_usuario());

        grava_entrada.grava(request.getSession().getAttribute("usuario")+"";
        if
(beanForm.getUsuario().getNm_usuario().equals("ADMIN")) {
            return (mapping.findForward("administrador"));
        } else {
            return (mapping.findForward("lista"));
        }
        } else {
            throw new ExcecaoGeral(
                new ActionMessage("aprendizado_crm.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"));
}
}
}
}

```

Classe Pergunta_aAction.java

```

package br.com.aprendizado_crm.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.aprendizado_crm.bean.Pergunta;
import br.com.aprendizado_crm.form.PerguntaForm;

```



```

import br.com.aprendizado_crm.sql.PerguntaSQL;
import br.com.aprendizado_crm.util.Util;

public class Pergunta_aAction extends LookupDispatchAction {

    public Category cat = Category.getInstance(Pergunta_aAction.class.getName());

    protected Map getKeyMethodMap(){
        Map map = new HashMap();
        map.put("aprendizado_crm.entra_inserir","entra_inserir");
        map.put("aprendizado_crm.entra_consultar","entra_consultar");
        map.put("aprendizado_crm.entra_alterar","entra_alterar");
        map.put("aprendizado_crm.inserir","inserir");
        map.put("aprendizado_crm.apagar","apagar");
        map.put("aprendizado_crm.alterar","alterar");
        map.put("aprendizado_crm.lista","lista");
        map.put("aprendizado_crm.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("pergunta "+usuario_form.getUsuario().getCd_emp());

        PerguntaForm beanForm = (PerguntaForm) form;
        PerguntaSQL beanSQL = new PerguntaSQL();
        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.setPergunta_lista(
            beanSQL.select(
                con,
                p_pagina,
                beanForm.getPergunta_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) {
    PerguntaForm beanForm = (PerguntaForm) 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");
    PerguntaForm beanForm = (PerguntaForm) form;
    PerguntaSQL beanSQL = new PerguntaSQL();
    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");
}

Pergunta v_pergunta = new Pergunta();

v_pergunta.setCd_pergunta(beanForm.getPergunta().getCd_pergunta());
beanForm.setPergunta_lista(
    beanSQL.select(con, 0, v_pergunta, 1, "", "N"));
if (beanForm.getPergunta_lista().size() > 0) {
    throw new ExcecaoGeral(
        new ActionMessage("aprendizado_crm.inserir.erro"));
} else {
    beanSQL.insert(beanForm.getPergunta(), con);
    con.commit();
    beanForm.setPergunta_lista(
        beanSQL.select(con, 0, beanForm.getPergunta(), 1, "", "N");
        beanForm.setPergunta_pesq(new Pergunta());
        beanForm.setPergunta_lista(
            beanSQL.select(
                con,
                0,
                beanForm.getPergunta_pesq(),
                2000000000,
                ((Pergunta)
                    beanForm.getPergunta_lista().get(0))
                    .getOid(), "N"));
        //
        beanForm.setNr_reg(10);
        p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
        beanForm.setPergunta_lista(
            beanSQL.select(
                con,
                p_pagina,
                beanForm.getPergunta_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 {
    PerguntaForm beanForm = (PerguntaForm) form;
    beanForm.setPergunta(new Pergunta());
    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");
    PerguntaForm beanForm = (PerguntaForm) form;
    PerguntaSQL beanSQL = new PerguntaSQL();
    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");
    }

    Pergunta v_pergunta = new Pergunta();
    v_pergunta.setOid(oid);
    v_pergunta =
        (Pergunta) beanSQL.select(con, 0, v_pergunta, 1, "", "N").get(0);

    beanForm.setPergunta_lista(
        beanSQL.select(
            con,
            0,
            beanForm.getPergunta_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");
    PerguntaForm beanForm = (PerguntaForm) form;
    try {
        PerguntaSQL beanSQL = new PerguntaSQL();

        String oid = "-1";
        if (Util.trataNulo(request.getParameter("oid")) != null) {
            oid = request.getParameter("oid");
        }
    }
}

```

```

        Pergunta v_pergunta = new Pergunta();
        v_pergunta.setOid(oid);
        beanForm.setPergunta_lista(beanSQL.select(con,0,v_pergunta,1,"","N"));
        beanForm.setPergunta((Pergunta) beanForm.getPergunta_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");
    PerguntaForm beanForm = (PerguntaForm) form;
    PerguntaSQL beanSQL = new PerguntaSQL();
    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.getPergunta(), con);
            con.commit();

            beanForm.setPergunta_lista(
                beanSQL.select(con, 0, beanForm.getPergunta(), 1, "", "N"));
            if (beanForm.getPergunta_lista().size() == 0) {
                throw new ExcecaoGeral(

```

```

        new ActionMessage("aprendizado_crm.alterar.erro"));
    }

    beanForm.setPergunta_lista(
        beanSQL.select(
            con,
            0,
            beanForm.getPergunta_pesq(),
            2000000000,
            beanForm.getPergunta().getOid(), "N"));

    //
    beanForm.setNr_reg(10);
    p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
    beanForm.setPergunta_lista(
        beanSQL.select(
            con,
            p_pagina,
            beanForm.getPergunta_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"));
}
}

```

Classe PerguntaAction.java

```
package br.com.aprendizado_crm.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.aprendizado_crm.bean.Pergunta;
import br.com.aprendizado_crm.form.PerguntaForm;
import br.com.aprendizado_crm.sql.PerguntaSQL;
import br.com.aprendizado_crm.util.Util;

public class PerguntaAction extends LookupDispatchAction {

    public Category cat = Category.getInstance(PerguntaAction.class.getName());

    protected Map getKeyMethodMap(){
        Map map = new HashMap();
        map.put("aprendizado_crm.entra_inserir","entra_inserir");
        map.put("aprendizado_crm.entra_consultar","entra_consultar");
        map.put("aprendizado_crm.entra_alterar","entra_alterar");
        map.put("aprendizado_crm.inserir","inserir");
        map.put("aprendizado_crm.apagar","apagar");
        map.put("aprendizado_crm.alterar","alterar");
        map.put("aprendizado_crm.lista","lista");
        map.put("aprendizado_crm.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");

PerguntaForm beanForm = (PerguntaForm) form;
PerguntaSQL beanSQL = new PerguntaSQL();
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.setPergunta_lista(
        beanSQL.select(
            con,
            p_pagina,
            beanForm.getPergunta_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) {
    PerguntaForm beanForm = (PerguntaForm) 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");
    PerguntaForm beanForm = (PerguntaForm) form;
    PerguntaSQL beanSQL = new PerguntaSQL();
    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");
            }

            Pergunta v_pergunta = new Pergunta();

v_pergunta.setCd_pergunta(beanForm.getPergunta().getCd_pergunta());
            beanForm.setPergunta_lista(
                beanSQL.select(con, 0, v_pergunta, 1, "", "N"));
            if (beanForm.getPergunta_lista().size() > 0) {
                throw new ExcecaoGeral(
                    new ActionMessage("aprendizado_crm.inserir.erro"));
            } else {
                beanSQL.insert(beanForm.getPergunta(), con);
                con.commit();
                beanForm.setPergunta_lista(

                    beanSQL.select(con,0,beanForm.getPergunta(),1,"","N"));
                beanForm.setPergunta_pesq(new Pergunta());
                beanForm.setPergunta_lista(
                    beanSQL.select(
                        con,
                        0,
                        beanForm.getPergunta_pesq(),
                        2000000000,
                        ((Pergunta)
beanForm.getPergunta_lista().get(0))
                            .getOid(), "N"));
                //
                beanForm.setNr_reg(1);

```

```

        p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
        beanForm.setPergunta_lista(
            beanSQL.select(
                con,
                p_pagina,
                beanForm.getPergunta_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 {
    PerguntaForm beanForm = (PerguntaForm) form;
    beanForm.setPergunta(new Pergunta());
    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");
    PerguntaForm beanForm = (PerguntaForm) form;
    PerguntaSQL beanSQL = new PerguntaSQL();

```

```

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");
    }

    Pergunta v_pergunta = new Pergunta();
    v_pergunta.setOid(oid);
    v_pergunta =
        (Pergunta) beanSQL.select(con, 0, v_pergunta, 1, "", "N").get(0);

    beanForm.setPergunta_lista(
        beanSQL.select(
            con,
            0,
            beanForm.getPergunta_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");
PerguntaForm beanForm = (PerguntaForm) form;
try {
    PerguntaSQL beanSQL = new PerguntaSQL();

    String oid = "-1";
    if (Util.trataNulo(request.getParameter("oid")) != null) {
        oid = request.getParameter("oid");
    }

    Pergunta v_pergunta = new Pergunta();
    v_pergunta.setOid(oid);
    beanForm.setPergunta_lista(beanSQL.select(con,0,v_pergunta,1,"","N"));
    beanForm.setPergunta((Pergunta) beanForm.getPergunta_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");
    PerguntaForm beanForm = (PerguntaForm) form;
    PerguntaSQL beanSQL = new PerguntaSQL();
    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.getPergunta(), con);
        con.commit();

        beanForm.setPergunta_lista(
            beanSQL.select(con, 0, beanForm.getPergunta(), 1, "", "N"));
        if (beanForm.getPergunta_lista().size() == 0) {
            throw new ExcecaoGeral(
                new ActionMessage("aprendizado_crm.alterar.erro"));
        }

        beanForm.setPergunta_lista(
            beanSQL.select(
                con,
                0,
                beanForm.getPergunta_pesq(),
                2000000000,
                beanForm.getPergunta().getOid(), "N"));

        //
        beanForm.setNr_reg(1);
        p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
        beanForm.setPergunta_lista(
            beanSQL.select(
                con,
                p_pagina,
                beanForm.getPergunta_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"));
    }
}
```

Classe Resposta_aAction.java

```
package br.com.aprendizado_crm.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.aprendizado_crm.bean.Resposta;
import br.com.aprendizado_crm.form.RespostaForm;
import br.com.aprendizado_crm.sql.RespostaSQL;
import br.com.aprendizado_crm.util.Util;
```

```
public class Resposta_aAction extends LookupDispatchAction {
```

```
    public Category cat = Category.getInstance(Resposta_aAction.class.getName());
```

```
    protected Map getKeyMethodMap(){
        Map map = new HashMap();
        map.put("aprendizado_crm.entra_inserir","entra_inserir");
        map.put("aprendizado_crm.entra_consultar","entra_consultar");
        map.put("aprendizado_crm.entra_alterar","entra_alterar");
        map.put("aprendizado_crm.inserir","inserir");
        map.put("aprendizado_crm.apagar","apagar");
        map.put("aprendizado_crm.alterar","alterar");
        map.put("aprendizado_crm.lista","lista");
        map.put("aprendizado_crm.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("resposta "+usuario_form.getUsuario().getCd_emp());

        RespostaForm beanForm = (RespostaForm) form;
        RespostaSQL beanSQL = new RespostaSQL();
        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.setResposta_lista(
                beanSQL.select(
                    con,
                    p_pagina,
                    beanForm.getResposta_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) {
        RespostaForm beanForm = (RespostaForm) 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");
        RespostaForm beanForm = (RespostaForm) form;
        RespostaSQL beanSQL = new RespostaSQL();
        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");
                }

                Resposta v_resposta = new Resposta();
                v_resposta.setCd_resposta(beanForm.getResposta().getCd_resposta());

                v_resposta.setCd_pergunta(beanForm.getResposta().getCd_pergunta());
                beanForm.setResposta_lista(
                    beanSQL.select(con, 0, v_resposta, 1, "", "N"));
                if (beanForm.getResposta_lista().size() > 0) {
                    throw new ExcecaoGeral(
                        new ActionMessage("aprendizado_crm.inserir.erro"));
                } else {
                    beanSQL.insert(beanForm.getResposta(), con);
                }
            }
        }
    }

```

```

        con.commit();
        beanForm.setResposta_lista(
beanSQL.select(con,0,beanForm.getResposta(),1,"","N"));
        beanForm.setResposta_pesq(new Resposta());
        beanForm.setResposta_lista(
            beanSQL.select(
                con,
                0,
                beanForm.getResposta_pesq(),
                2000000000,
                ((Resposta)
beanForm.getResposta_lista().get(0))
                    .getOid(), "N"));
        //
        beanForm.setNr_reg(10);
        p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
        beanForm.setResposta_lista(
            beanSQL.select(
                con,
                p_pagina,
                beanForm.getResposta_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 {
    RespostaForm beanForm = (RespostaForm) form;

```

```

        beanForm.setResposta(new Resposta());
        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");
        RespostaForm beanForm = (RespostaForm) form;
        RespostaSQL beanSQL = new RespostaSQL();
        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");
            }

            Resposta v_resposta = new Resposta();
            v_resposta.setOid(oid);
            v_resposta =
                (Resposta) beanSQL.select(con, 0, v_resposta, 1, "", "N").get(0);

            beanForm.setResposta_lista(
                beanSQL.select(
                    con,
                    0,
                    beanForm.getResposta_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");
    RespostaForm beanForm = (RespostaForm) form;
    try {
        RespostaSQL beanSQL = new RespostaSQL();

        String oid = "-1";
        if (Util.trataNulo(request.getParameter("oid")) != null) {
            oid = request.getParameter("oid");
        }

        Resposta v_resposta = new Resposta();
        v_resposta.setOid(oid);
        beanForm.setResposta_lista(beanSQL.select(con,0,v_resposta,1,"","N"));
        beanForm.setResposta((Resposta) beanForm.getResposta_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");
RespostaForm beanForm = (RespostaForm) form;
RespostaSQL beanSQL = new RespostaSQL();
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.getResposta(), con);
        con.commit();

        beanForm.setResposta_lista(
            beanSQL.select(con, 0, beanForm.getResposta(), 1, "", "N"));
        if (beanForm.getResposta_lista().size() == 0) {
            throw new ExcecaoGeral(
                new ActionMessage("aprendizado_crm.alterar.erro"));
        }

        beanForm.setResposta_lista(
            beanSQL.select(
                con,
                0,
                beanForm.getResposta_pesq(),
                2000000000,
                beanForm.getResposta().getOid(), "N"));

        //
        beanForm.setNr_reg(10);
        p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
        beanForm.setResposta_lista(
            beanSQL.select(
                con,
                p_pagina,
                beanForm.getResposta_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"));
}
}

```

Classe RespostaAction.java

```

package br.com.aprendizado_crm.action;

import java.sql.Connection;
import java.util.ArrayList;
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.ActionMessages;
import org.apache.struts.actions.LookupDispatchAction;

import br.com.aprendizado_crm.bean.Pergunta;
import br.com.aprendizado_crm.bean.Resposta;
import br.com.aprendizado_crm.bean.Resultado;
import br.com.aprendizado_crm.form.RespostaForm;
import br.com.aprendizado_crm.sql.PerguntaSQL;
import br.com.aprendizado_crm.sql.RespostaSQL;
import br.com.aprendizado_crm.sql.ResultadoSQL;
import br.com.aprendizado_crm.util.Util;

public class RespostaAction extends LookupDispatchAction {

```

```

public Category cat = Category.getInstance(RespostaAction.class.getName());

protected Map getKeyMethodMap(){
    Map map = new HashMap();
    map.put("aprendizado_crm.entra_inserir","entra_inserir");
    map.put("aprendizado_crm.entra_consultar","entra_consultar");
    map.put("aprendizado_crm.entra_alterar","entra_alterar");
    map.put("aprendizado_crm.inserir","inserir");
    map.put("aprendizado_crm.apagar","apagar");
    map.put("aprendizado_crm.alterar","alterar");
    map.put("aprendizado_crm.lista","lista");
    map.put("aprendizado_crm.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");

    RespostaForm beanForm = (RespostaForm) form;
    RespostaSQL beanSQL = new RespostaSQL();
    //try {
        beanForm.setNr_reg(5); //Numero de registros da lista
        beanForm.setMostraConsulta("N");

        int nr_pergunta_atual = 0;
        if (Util.trataNulo(request.getParameter("nr_pergunta_atual")) == null) {
            nr_pergunta_atual = 1;
        } else {
            nr_pergunta_atual =
                (int)
Integer.parseInt(request.getParameter("nr_pergunta_atual"));
        }
        int nr_resposta_atual = 0;
        if (Util.trataNulo(request.getParameter("nr_resposta_atual")) == null) {

```

```

        nr_resposta_atual = 1;
    } else {
        nr_resposta_atual =
            (int)
Integer.parseInt(request.getParameter("nr_resposta_atual"));
    }

    //
    PerguntaSQL perguntaSQL = new PerguntaSQL();
    Pergunta pergunta = new Pergunta();
    pergunta.setCd_pergunta(nr_pergunta_atual+ "");

    Resposta resposta = new Resposta();
    resposta.setCd_pergunta(nr_pergunta_atual+ "");

    ArrayList x = beanSQL.carregaDisplay(beanSQL.select(con, 0, resposta, 1, "",
"", con));

    if (x.size() > 0) {
        resposta = (Resposta) beanSQL.carregaDisplay(beanSQL.select(con,
0, resposta, 1, "", ""), con).get(0);
    } else {
        Resultado resultado = new Resultado();
        ResultadoSQL resultadoSQL = new ResultadoSQL();

resultado.setNm_usuario(request.getSession().getAttribute("usuario")+ "");
        resultado.setCd_pergunta((nr_pergunta_atual-1)+ "");
        //resultado.setCd_resposta(beanForm.getResposta().getCd_resposta());

        ArrayList aux = resultadoSQL.select(con, 0, resultado, 1, "", "");
        //apaguei todos os registros de um usuário x caso exista.
        if (aux.size() > 0) {
            resultado = (Resultado) aux.get(0);
            resultadoSQL.delete(resultado.getId(), con);
        }

        RespostaSQL respSQL = new RespostaSQL();
        Resposta resp = new Resposta();

        resp.setCd_resposta((nr_resposta_atual)+ "");
        resp.setCd_pergunta((nr_pergunta_atual - 1)+ "");

        ArrayList status = respSQL.select(con, 0, resp, 1, "", "");

        if (status.size() > 0) {
            Resposta r = (Resposta) status.get(0);
            resultado.setId_status(r.getId_status());

resultado.setNm_usuario(request.getSession().getAttribute("usuario")+ "");
            resultado.setCd_pergunta((nr_pergunta_atual - 1)+ "");

```



```

        resultado.setCd_resposta((nr_resposta_atual)+"");

        resultadoSQL.insert(resultado, con);
        con.commit();
    }
    return (mapping.findForward("resultado"));
}
//
beanForm.getResposta().setDs_pergunta(resposta.getDs_pergunta());
beanForm.getResposta().setCd_pergunta(nr_pergunta_atual+"");
//
resposta = new Resposta();
resposta.setCd_pergunta(nr_pergunta_atual+"");
//
beanForm.setResposta_lista(
    beanSQL.select(
        con,
        0,
        resposta,
        5,
        "", ""));
beanForm.setContadoresPaginas(
    beanSQL.getContaSelect(),
    beanSQL.getContaSelectAtual());

if (nr_pergunta_atual > 1) {
    Resultado resultado = new Resultado();
    ResultadoSQL resultadoSQL = new ResultadoSQL();

resultado.setNm_usuario(request.getSession().getAttribute("usuario"+"");
resultado.setCd_pergunta((nr_pergunta_atual-1)+"");
//resultado.setCd_resposta(beanForm.getResposta().getCd_resposta());

    ArrayList aux = resultadoSQL.select(con, 0, resultado, 1, "", "");
    //apaguei todos os registros de um usuário x caso exista.
    if (aux.size() > 0) {
        resultado = (Resultado) aux.get(0);
        resultadoSQL.delete(resultado.getId(), con);
    }

    RespostaSQL respSQL = new RespostaSQL();
    Resposta resp = new Resposta();

    resp.setCd_resposta((nr_resposta_atual)+"");
    resp.setCd_pergunta((nr_pergunta_atual - 1)+"");

    ArrayList status = respSQL.select(con, 0, resp, 1, "", "");

    if (status.size() > 0) {
        Resposta r = (Resposta) status.get(0);
        resultado.setId_status(r.getId_status());
    }
}

```

```

resultado.setNm_usuario(request.getSession().getAttribute("usuario")+""");
                resultado.setCd_pergunta((nr_pergunta_atual - 1)+""");
                resultado.setCd_resposta((nr_resposta_atual)+""");

                resultadoSQL.insert(resultado, con);
                con.commit();
            }

        }
        //
        nr_pergunta_atual++;
        beanForm.setNr_pergunta_atual(nr_pergunta_atual);
        //

        /*} catch (Exception e) {
            msgs.add(
                ActionMessages.GLOBAL_MESSAGE,
                new ActionMessage("error.sql", e));
        }*/
        if (!(msgs.isEmpty())) {
            saveMessages(request, msgs);
        }
        return (mapping.findForward("lista"));
    }
}

```

Classe Resultado_aAction.java

```

package br.com.aprendizado_crm.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.aprendizado_crm.bean.Resultado;
import br.com.aprendizado_crm.form.ResultadoForm;
import br.com.aprendizado_crm.sql.ResultadoSQL;

```

```

import br.com.aprendizado_crm.util.Util;

public class Resultado_aAction extends LookupDispatchAction {

    public Category cat = Category.getInstance(Resultado_aAction.class.getName());

    protected Map getKeyMethodMap(){
        Map map = new HashMap();
        map.put("aprendizado_crm.entra_inserir","entra_inserir");
        map.put("aprendizado_crm.entra_consultar","entra_consultar");
        map.put("aprendizado_crm.entra_alterar","entra_alterar");
        map.put("aprendizado_crm.inserir","inserir");
        map.put("aprendizado_crm.apagar","apagar");
        map.put("aprendizado_crm.alterar","alterar");
        map.put("aprendizado_crm.lista","lista");
        map.put("aprendizado_crm.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");

        ResultadoForm beanForm = (ResultadoForm) form;
        ResultadoSQL beanSQL = new ResultadoSQL();

        try {
            beanForm.setNr_reg(6); //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"));
            }

            Resultado v_resultado = new Resultado();
            v_resultado.setId_status("C");
            beanForm.setResultado_lista(
                beanSQL.carregaDisplay(
                    beanSQL.select(
                        con,
                        p_pagina,
                        v_resultado,
                        6, "", ""),con));
            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"));
    }
}

```

Classe Resultado_bAction.java

```

package br.com.aprendizado_crm.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.aprendizado_crm.bean.Resultado;
import br.com.aprendizado_crm.form.ResultadoForm;
import br.com.aprendizado_crm.sql.ResultadoSQL;
import br.com.aprendizado_crm.util.Util;

```

```

public class Resultado_bAction extends LookupDispatchAction {

    public Category cat = Category.getInstance(Resultado_bAction.class.getName());

    protected Map getKeyMethodMap(){
        Map map = new HashMap();
        map.put("aprendizado_crm.entra_inserir","entra_inserir");
        map.put("aprendizado_crm.entra_consultar","entra_consultar");
        map.put("aprendizado_crm.entra_alterar","entra_alterar");
        map.put("aprendizado_crm.inserir","inserir");
        map.put("aprendizado_crm.apagar","apagar");
        map.put("aprendizado_crm.alterar","alterar");
        map.put("aprendizado_crm.lista","lista");
        map.put("aprendizado_crm.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("resultado "+usuario_form.getUsuario().getCd_emp());

        ResultadoForm beanForm = (ResultadoForm) form;
        ResultadoSQL beanSQL = new ResultadoSQL();
        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.setResultado_lista(
        beanSQL.select(
            con,
            p_pagina,
            beanForm.getResultado_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) {
    ResultadoForm beanForm = (ResultadoForm) 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");
    ResultadoForm beanForm = (ResultadoForm) form;
    ResultadoSQL beanSQL = new ResultadoSQL();
    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");
        }

        Resultado v_resultado = new Resultado();

        v_resultado.setNm_usuario(beanForm.getResultado().getNm_usuario());
        beanForm.setResultado_lista(
            beanSQL.select(con, 0, v_resultado, 1, "", "N"));
        if (beanForm.getResultado_lista().size() > 0) {
            throw new ExcecaoGeral(
                new ActionMessage("aprendizado_crm.inserir.erro"));
        } else {
            beanSQL.insert(beanForm.getResultado(), con);
            con.commit();
            beanForm.setResultado_lista(

                beanSQL.select(con,0,beanForm.getResultado(),1,"","N"));
            beanForm.setResultado_pesq(new Resultado());
            beanForm.setResultado_lista(
                beanSQL.select(
                    con,
                    0,
                    beanForm.getResultado_pesq(),
                    2000000000,
                    ((Resultado)
                        beanForm.getResultado_lista().get(0))
                        .getOid(), "N"));
            //
            beanForm.setNr_reg(10);
            p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
            beanForm.setResultado_lista(
                beanSQL.select(
                    con,
                    p_pagina,
                    beanForm.getResultado_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 {
    ResultadoForm beanForm = (ResultadoForm) form;
    beanForm.setResultado(new Resultado());
    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");
    ResultadoForm beanForm = (ResultadoForm) form;
    ResultadoSQL beanSQL = new ResultadoSQL();
    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");
        }
    }
}

```



```

Resultado v_resultado = new Resultado();
v_resultado.setOid(oid);
v_resultado =
    (Resultado) beanSQL.select(con, 0, v_resultado, 1, "", "N").get(0);

beanForm.setResultado_lista(
    beanSQL.select(
        con,
        0,
        beanForm.getResultado_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");
    ResultadoForm beanForm = (ResultadoForm) form;
    try {
        ResultadoSQL beanSQL = new ResultadoSQL();

        String oid = "-1";
        if (Util.trataNulo(request.getParameter("oid")) != null) {
            oid = request.getParameter("oid");
        }

        Resultado v_resultado = new Resultado();
        v_resultado.setOid(oid);
    }
}

```

```

        beanForm.setResultado_lista(beanSQL.select(con,0,v_resultado,1,"","N"));
        beanForm.setResultado((Resultado) beanForm.getResultado_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");
    ResultadoForm beanForm = (ResultadoForm) form;
    ResultadoSQL beanSQL = new ResultadoSQL();
    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.getResultado(), con);
            con.commit();

            beanForm.setResultado_lista(
                beanSQL.select(con, 0, beanForm.getResultado(), 1, "", "N"));
            if (beanForm.getResultado_lista().size() == 0) {
                throw new ExcecaoGeral(
                    new ActionMessage("aprendizado_crm.alterar.erro"));
            }
        }
    }
}

```

```

        beanForm.setResultado_lista(
            beanSQL.select(
                con,
                0,
                beanForm.getResultado_pesq(),
                2000000000,
                beanForm.getResultado().getOid(), "N"));

        //
        beanForm.setNr_reg(10);
        p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
        beanForm.setResultado_lista(
            beanSQL.select(
                con,
                p_pagina,
                beanForm.getResultado_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"));
}
}

```

Classe ResultadoAction.java

```

package br.com.aprendizado_crm.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.aprendizado_crm.form.ResultadoForm;
import br.com.aprendizado_crm.sql.ResultadoSQL;
import br.com.aprendizado_crm.util.Util;

public class ResultadoAction extends LookupDispatchAction {

    public Category cat = Category.getInstance(ResultadoAction.class.getName());

    protected Map getKeyMethodMap(){
        Map map = new HashMap();
        map.put("aprendizado_crm.entra_inserir","entra_inserir");
        map.put("aprendizado_crm.entra_consultar","entra_consultar");
        map.put("aprendizado_crm.entra_alterar","entra_alterar");
        map.put("aprendizado_crm.inserir","inserir");
        map.put("aprendizado_crm.apagar","apagar");
        map.put("aprendizado_crm.alterar","alterar");
        map.put("aprendizado_crm.lista","lista");
        map.put("aprendizado_crm.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());

ResultadoForm beanForm = (ResultadoForm) form;
ResultadoSQL beanSQL = new ResultadoSQL();

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.getResultado_pesq().setNm_usuario(request.getSession().getAttribute("usuario")+""");

    beanForm.setResultado_lista(
        beanSQL.carregaDisplay(
            beanSQL.select(
                con,
                p_pagina,
                beanForm.getResultado_pesq(),
                beanForm.getNr_reg(), "", "", con));
        /*beanSQL.select(
            con,
            p_pagina,
            beanForm.getResultado_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"));
}
}

```

Classe SetorAction.java

```
package br.com.aprendizado_crm.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.aprendizado_crm.bean.Setor;
import br.com.aprendizado_crm.form.SetorForm;
import br.com.aprendizado_crm.sql.SetorSQL;
import br.com.aprendizado_crm.util.Util;

public class SetorAction extends LookupDispatchAction {

    public Category cat = Category.getInstance(SetorAction.class.getName());

    protected Map getKeyMethodMap(){
        Map map = new HashMap();
        map.put("aprendizado_crm.entra_inserir","entra_inserir");
        map.put("aprendizado_crm.entra_consultar","entra_consultar");
        map.put("aprendizado_crm.entra_alterar","entra_alterar");
        map.put("aprendizado_crm.inserir","inserir");
        map.put("aprendizado_crm.apagar","apagar");
        map.put("aprendizado_crm.alterar","alterar");
        map.put("aprendizado_crm.lista","lista");
        map.put("aprendizado_crm.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("setor "+usuario_form.getUsuario().getCd_emp());

    SetorForm beanForm = (SetorForm) form;
    SetorSQL beanSQL = new SetorSQL();
    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.setSetor_lista(
            beanSQL.carregaDisplay(beanSQL.select(
                con,
                p_pagina,
                beanForm.getSetor_pesq(),
                beanForm.getNr_reg(),
                "", ""),con));
        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) {
        SetorForm beanForm = (SetorForm) 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");
        SetorForm beanForm = (SetorForm) form;
        SetorSQL beanSQL = new SetorSQL();
        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");
                }

                Setor v_setor = new Setor();
                v_setor.setCd_emp(beanForm.getSetor().getCd_emp());
                v_setor.setCd_setor(beanForm.getSetor().getCd_setor());
                beanForm.setSetor_lista(
                    beanSQL.select(con, 0, v_setor, 1, "", "N"));
                if (beanForm.getSetor_lista().size() > 0) {
                    throw new ExcecaoGeral(
                        new ActionMessage("aprendizado_crm.inserir.erro"));
                } else {
                    beanSQL.insert(beanForm.getSetor(), con);
                    con.commit();
                    beanForm.setSetor_lista(
                        beanSQL.select(con, 0, beanForm.getSetor(), 1, "", "N"));
                    beanForm.setSetor_pesq(new Setor());
                    beanForm.setSetor_lista(

```



```

        beanSQL.select(
            con,
            0,
            beanForm.getSetor_pesq(),
            2000000000,
            ((Setor) beanForm.getSetor_lista().get(0))
                .getOid(), "N");
        //
        beanForm.setNr_reg(10);
        p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
        beanForm.setSetor_lista(
            beanSQL.select(
                con,
                p_pagina,
                beanForm.getSetor_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 {
    SetorForm beanForm = (SetorForm) form;
    beanForm.setSetor(new Setor());
    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");
SetorForm beanForm = (SetorForm) form;
SetorSQL beanSQL = new SetorSQL();
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");
    }

    Setor v_setor = new Setor();
    v_setor.setOid(oid);
    v_setor =
        (Setor) beanSQL.select(con, 0, v_setor, 1, "", "N").get(0);

    beanForm.setSetor_lista(
        beanSQL.select(
            con,
            0,
            beanForm.getSetor_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");
    SetorForm beanForm = (SetorForm) form;
    try {
        SetorSQL beanSQL = new SetorSQL();

        String oid = "-1";
        if (Util.trataNulo(request.getParameter("oid")) != null) {
            oid = request.getParameter("oid");
        }

        Setor v_setor = new Setor();
        v_setor.setOid(oid);
        beanForm.setSetor_lista(beanSQL.select(con,0,v_setor,1,"","N"));
        beanForm.setSetor((Setor) beanForm.getSetor_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");
    SetorForm beanForm = (SetorForm) form;
    SetorSQL beanSQL = new SetorSQL();
    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.getSetor(), con);
        con.commit();

        beanForm.setSetor_lista(
            beanSQL.select(con, 0, beanForm.getSetor(), 1, "", "N"));
        if (beanForm.getSetor_lista().size() == 0) {
            throw new ExcecaoGeral(
                new ActionMessage("aprendizado_crm.alterar.erro"));
        }

        beanForm.setSetor_lista(
            beanSQL.select(
                con,
                0,
                beanForm.getSetor_pesq(),
                2000000000,
                beanForm.getSetor().getOid(), "N"));

        //
        beanForm.setNr_reg(10);
        p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
        beanForm.setSetor_lista(
            beanSQL.carregaDisplay(beanSQL.select(
                con,
                p_pagina,
                beanForm.getSetor_pesq(),
                beanForm.getNr_reg(),
                "", ""), con));
        beanForm.setContadoresPaginas(
            beanSQL.getContaSelect(),
            beanSQL.getContaSelectAtual());

    } catch (ExcecaoGeral e) {
        msggs.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"));
    }
}

```

Classe UsuarioAction.java

```

package br.com.aprendizado_crm.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.aprendizado_crm.bean.Usuario;
import br.com.aprendizado_crm.form.UsuarioForm;
import br.com.aprendizado_crm.sql.UsuarioSQL;
import br.com.aprendizado_crm.util.Util;

public class UsuarioAction extends LookupDispatchAction {

    public Category cat = Category.getInstance(UsuarioAction.class.getName());

    protected Map getKeyMethodMap(){
        Map map = new HashMap();
        map.put("aprendizado_crm.entra_inserir","entra_inserir");
        map.put("aprendizado_crm.entra_consultar","entra_consultar");
        map.put("aprendizado_crm.entra_alterar","entra_alterar");
        map.put("aprendizado_crm.inserir","inserir");
        map.put("aprendizado_crm.apagar","apagar");
        map.put("aprendizado_crm.alterar","alterar");
        map.put("aprendizado_crm.lista","lista");
    }
}

```

```

        map.put("aprendizado_crm.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("usuario"+usuario_form.getUsuario().getCd_emp());

        UsuarioForm beanForm = (UsuarioForm) form;
        UsuarioSQL beanSQL = new UsuarioSQL();
        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.setUsuario_lista(
                beanSQL.select(
                    con,
                    p_pagina,
                    beanForm.getUsuario_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) {
    UsuarioForm beanForm = (UsuarioForm) 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");
    UsuarioForm beanForm = (UsuarioForm) form;
    UsuarioSQL beanSQL = new UsuarioSQL();
    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");
            }

            Usuario v_usuario= new Usuario();
            v_usuario.setNm_usuario(beanForm.getUsuario().getNm_usuario());

```

```

        beanForm.setUsuario_lista(
            beanSQL.select(con, 0, v_usuario, 1, "", "N"));
        if (beanForm.getUsuario_lista().size() > 0) {
            throw new ExcecaoGeral(
                new ActionMessage("aprendizado_crm.inserir.erro"));
        } else {
            beanSQL.insert(beanForm.getUsuario(), con);
            con.commit();
            beanForm.setUsuario_lista(
                beanSQL.select(con, 0, beanForm.getUsuario(), 1, "", "N"));
            beanForm.setUsuario_pesq(new Usuario());
            beanForm.setUsuario_lista(
                beanSQL.select(
                    con,
                    0,
                    beanForm.getUsuario_pesq(),
                    2000000000,
                    ((Usuario)
                        beanForm.getUsuario_lista().get(0))
                        .getOid(), "N"));
            //
            beanForm.setNr_reg(10);
            p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
            beanForm.setUsuario_lista(
                beanSQL.select(
                    con,
                    p_pagina,
                    beanForm.getUsuario_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 {
    UsuarioForm beanForm = (UsuarioForm) form;
    beanForm.setUsuario(new Usuario());
    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");
    UsuarioForm beanForm = (UsuarioForm) form;
    UsuarioSQL beanSQL = new UsuarioSQL();
    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");
        }

        Usuario v_usuario= new Usuario();
        v_usuario.setOid(oid);
        v_usuario=
            (Usuario) beanSQL.select(con, 0, v_usuario, 1, "", "N").get(0);

        beanForm.setUsuario_lista(
            beanSQL.select(
                con,
                0,
                beanForm.getUsuario_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");
    UsuarioForm beanForm = (UsuarioForm) form;
    try {
        UsuarioSQL beanSQL = new UsuarioSQL();

        String oid = "-1";
        if (Util.trataNulo(request.getParameter("oid")) != null) {
            oid = request.getParameter("oid");
        }

        Usuario v_usuario= new Usuario();
        v_usuario.setOid(oid);
        beanForm.setUsuario_lista(beanSQL.select(con,0,v_usuario,1,"","N"));
        beanForm.setUsuario((Usuario) beanForm.getUsuario_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");
UsuarioForm beanForm = (UsuarioForm) form;
UsuarioSQL beanSQL = new UsuarioSQL();
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.getUsuario(), con);
        con.commit();

        beanForm.setUsuario_lista(
            beanSQL.select(con, 0, beanForm.getUsuario(), 1, "", "N"));
        if (beanForm.getUsuario_lista().size() == 0) {
            throw new ExcecaoGeral(
                new ActionMessage("aprendizado_crm.alterar.erro"));
        }

        beanForm.setUsuario_lista(
            beanSQL.select(
                con,
                0,
                beanForm.getUsuario_pesq(),
                2000000000,
                beanForm.getUsuario().getOid(), "N"));

        //
        beanForm.setNr_reg(10);
        p_pagina = beanForm.retornaPagina(beanSQL.getNr_atual());
        beanForm.setUsuario_lista(
            beanSQL.select(

```

```

        con,
        p_pagina,
        beanForm.getUsuario_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"));
}
}

```

aprendizado_crm.xml

```

<Context path="/aprendizado_crm" docBase="aprendizado_crm" debug="9" reloadable="true">
<Loader checkInterval="3"/>
<Logger className="org.apache.catalina.logger.SystemOutLogger" timestamp="true"/>

<Resource name="jdbc/dsaprendizado_crm" auth="Container" type="javax.sql.DataSource"/>

<ResourceParams name="jdbc/dsaprendizado_crm">
<parameter>
<name>factory</name>
<value>org.apache.commons.dbcp.BasicDataSourceFactory</value>
</parameter>
<parameter>
<name>driverClassName</name>
<value>org.postgresql.Driver</value>
</parameter>
<parameter>
<name>url</name>
<value>jdbc:postgresql://200.247.141.68:5432/aprendizado_crm</value>
</parameter>
<parameter>
<name>username</name>

```

```

    <value>postgres</value>
  </parameter>
  <parameter>
    <name>password</name>
    <value>postgres</value>
  </parameter>
  <parameter>
    <name>maxActive</name>
    <value>40</value>
  </parameter>
  <parameter>
    <name>maxIdle</name>
    <value>10</value>
  </parameter>
  <parameter>
    <name>maxWait</name>
    <value>-1</value>
  </parameter>
</ResourceParams>

</Context>

```

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.aprendizado_crm.form.EmpresaForm"/>
    <form-bean name="respostaForm"
      type="br.com.aprendizado_crm.form.RespostaForm"/>
    <form-bean name="usuarioForm"
      type="br.com.aprendizado_crm.form.UsuarioForm" />
    <form-bean name="resultadoForm"
      type="br.com.aprendizado_crm.form.ResultadoForm" />
    <form-bean name="perguntaForm"
      type="br.com.aprendizado_crm.form.PerguntaForm" />
    <form-bean name="colaboradorForm"
      type="br.com.aprendizado_crm.form.ColaboradorForm" />
    <form-bean name="escolaridadeForm"
      type="br.com.aprendizado_crm.form.EscolaridadeForm" />
    <form-bean name="setorForm"
      type="br.com.aprendizado_crm.form.SetorForm" />
    <form-bean name="estatistica_acessoForm"

```

```

        type="br.com.aprendizado_crm.form.Estatistica_acessoForm" />
    </form-beans>
<!-- ===== Action Mapping Definitions ===== -->
<action-mappings>
    <action
        attribute="resultadoForm"
        name="resultadoForm"
        parameter="evento"
        path="/resultado"
        scope="request"
        type="br.com.aprendizado_crm.action.ResultadoAction"
        validate="false">
        <forward name="lista" path="/form/lista_resultado.jsp" />
    </action>
    <action
        attribute="empresaForm"
        name="empresaForm"
        parameter="evento"
        path="/empresa"
        scope="request"
        type="br.com.aprendizado_crm.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="respostaForm"
        name="respostaForm"
        parameter="evento"
        path="/resposta"
        scope="request"
        type="br.com.aprendizado_crm.action.RespostaAction"
        validate="false">
        <forward name="lista" path="/form/lista_resposta.jsp" />
        <forward name="editar" path="/form/editar_resposta.jsp" />
        <forward name="success" path="/success.jsp" />
        <forward name="resultado" path="/resultado.do" />
    </action>
    <action
        attribute="usuarioForm"
        name="usuarioForm"
        parameter="evento"
        path="/login"
        scope="session"
        type="br.com.aprendizado_crm.action.LoginAction"
        validate="false">
        <forward name="lista" path="/form/principal.jsp" />
        <forward name="administrador" path="/form/administrador.jsp" />
        <forward name="editar" path="/form/index.jsp" />
    </action>

```

```
<action
  attribute="usuarioForm"
  name="usuarioForm"
  parameter="evento"
  path="/usuario"
  scope="session"
  type="br.com.aprendizado_crm.action.UsuarioAction"
  validate="false">
  <forward name="lista" path="/form/lista_usuario.jsp" />
  <forward name="editar" path="/form/editar_usuario.jsp" />
</action>
<action
  attribute="escolaridadeForm"
  name="escolaridadeForm"
  parameter="evento"
  path="/escolaridade"
  scope="session"
  type="br.com.aprendizado_crm.action.EscolaridadeAction"
  validate="false">
  <forward name="lista" path="/form/lista_escolaridade.jsp" />
  <forward name="editar" path="/form/editar_escolaridade.jsp" />
</action>
<action
  attribute="setorForm"
  name="setorForm"
  parameter="evento"
  path="/setor"
  scope="session"
  type="br.com.aprendizado_crm.action.SetorAction"
  validate="false">
  <forward name="lista" path="/form/lista_setor.jsp" />
  <forward name="editar" path="/form/editar_setor.jsp" />
</action>
<action
  attribute="colaboradorForm"
  name="colaboradorForm"
  parameter="evento"
  path="/colaborador"
  scope="session"
  type="br.com.aprendizado_crm.action.ColaboradorAction"
  validate="false">
  <forward name="lista" path="/form/lista_colaborador.jsp" />
  <forward name="editar" path="/form/editar_colaborador.jsp" />
</action>
<action
  attribute="perguntaForm"
  name="perguntaForm"
  parameter="evento"
  path="/pergunta_a"
  scope="session"
  type="br.com.aprendizado_crm.action.Pergunta_aAction"
```

```

        validate="false">
        <forward name="lista" path="/form/lista_pergunta.jsp" />
        <forward name="editar" path="/form/editar_pergunta.jsp" />
    </action>
    <action
        attribute="respostaForm"
        name="respostaForm"
        parameter="evento"
        path="/resposta_a"
        scope="session"
        type="br.com.aprendizado_crm.action.Resposta_aAction"
        validate="false">
        <forward name="lista" path="/form/lista_resposta_a.jsp" />
        <forward name="editar" path="/form/editar_resposta_a.jsp" />
    </action>
    <action
        attribute="resultadoForm"
        name="resultadoForm"
        parameter="evento"
        path="/resultado_a"
        scope="session"
        type="br.com.aprendizado_crm.action.Resultado_aAction"
        validate="false">
        <forward name="lista" path="/form/lista_resultado_a.jsp" />
    </action>
    <action
        attribute="resultadoForm"
        name="resultadoForm"
        parameter="evento"
        path="/resultado_b"
        scope="session"
        type="br.com.aprendizado_crm.action.Resultado_bAction"
        validate="false">
        <forward name="lista" path="/form/lista_resultado_b.jsp" />
        <forward name="editar" path="/form/editar_resultado_b.jsp" />
    </action>
    <action
        attribute="estatistica_acessoForm"
        name="estatistica_acessoForm"
        parameter="evento"
        path="/estatistica_acesso"
        scope="session"
        type="br.com.aprendizado_crm.action.Estatistica_acessoAction"
        validate="false">
        <forward name="lista" path="/form/lista_estatistica_acesso.jsp" />
        <forward name="editar" path="/form/editar_estatistica_acesso.jsp" />
    </action>
</action-mappings>
<!-- ===== Controller Configuration ===== -->
<controller processorClass="org.apache.struts.tiles.TilesRequestProcessor" />
<!-- ===== Plug Ins Configuration ===== -->

```



```

<message-resources parameter="br.com.aprendizado_crm.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>

```

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>Aprendizado CRM</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>
```

```
  <servlet>
```

```
    <servlet-name>CewolfServlet</servlet-name>
```

```
    <servlet-class>de.laures.cewolf.CewolfRenderer</servlet-class>
```

```
  </servlet>
```

```
</servlet-mapping>
```

```

        <servlet-name>CewolfServlet</servlet-name>
        <url-pattern>/cewolf/*</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>
    <mime-mapping>
        <extension>css</extension>
        <mime-type>text/css</mime-type>
    </mime-mapping>
    <welcome-file-list>
        <welcome-file>form/index.jsp</welcome-file>
    </welcome-file-list>
    <taglib>
        <taglib-uri>/WEB-INF/cewolf-1.1.tld</taglib-uri>
        <taglib-location>/WEB-INF/cewolf-1.1.tld</taglib-location>
    </taglib>
    <taglib>
        <taglib-uri>/WEB-INF/cewolf.tld</taglib-uri>
        <taglib-location>/WEB-INF/cewolf.tld</taglib-location>
    </taglib>
    <resource-ref>
        <description>Postgres Datasource Teste</description>
        <res-ref-name>jdbc/dsaprendizado_crm</res-ref-name>
        <res-type>javax.sql.DataSource</res-type>
        <res-auth>Container</res-auth>
    </resource-ref>

</web-app>

```

menu.css

```
/* level 0 inner */
```

```

a:link {
    text-decoration: none;
    font-family: Verdana, Arial, Helvetica, sans-serif;
    color: #63659C;
    font-size: 8pt;
}
a:visited {
    text-decoration: none;

```

```

        font-family: Verdana, Arial, Helvetica, sans-serif;
        color: #63659C;
        font-size: 8pt;
    }
    a:hover{
        color: #63659C;
        text-decoration: none;
        font-family: Verdana, Arial, Helvetica, sans-serif;
        font-size: 8pt;
    }
.m0l0iout {
    font-family: Verdana,Arial, Helvetica, sans-serif;
    font-size: 9px;
    text-decoration: none;
    padding: 4px;
    color: #63659C;
}
.m0l0iover {
    font: 9px Verdana,Arial, Helvetica, sans-serif;
    text-decoration: underline;
    padding: 4px;
    color: #63659C;
}
/* level 0 outer */
.m0l0oout {
    text-decoration : none;
    border : 0px solid #FFFFFF;
    background: #BFD58A;
}
.m0l0oover {
    text-decoration : none;
    border : 0px solid #FFFFFF;
    background: #BFD58A;
}
/* level 1 inner */
.m0l1iout {
    font: 9px Verdana,Arial, Helvetica, sans-serif;
    text-decoration: none;
    padding: 4px;
    color: #63659C;
}
.m0l1iover {
    font: bold 9px Verdana,Arial, Helvetica, sans-serif;
    text-decoration : none;
    padding: 4px;
    color: #63659C;
}
/* level 1 outer */
.m0l1oout {
    text-decoration : none;
    border : 1px solid #FFFFFF;

```

```

    background: #BFD58A;
    /* filter: alpha(opacity=85); */
}
.m0l1oover {
    text-decoration : none;
    border : 1px solid #FFFFFF;
    background: #BFD58A;
}
/* level 2 inner */
.m0l2iover {
    font: bold 9px Verdana,Arial, Helvetica, sans-serif;
    text-decoration : none;
    padding: 4px;
    color: #63659C;
}
/* level 2 outer */
.m0l2oout {
    text-decoration : none;
    border : 1px solid #FFFFFF;
    background: #BFD58A;
}
.m0l2oover {
    text-decoration : none;
    border : 1px solid #FFFFFF;
    background: #BFD58A;
}
}

```

menu.js

```

// Title: tigma menu
// Description: See the demo at url
// URL: http://www.softcomplex.com/products/tigma\_menu/
// Version: 2.0 (commented source)
// Date: 04-05-2003 (mm-dd-yyyy)
// Tech. Support: http://www.softcomplex.com/forum/forumdisplay.php?fid=40
// Notes: This script is free. Visit official site for further details.
// -----
// global collection containing all menus on current page
var A_MENUS = [];
// -----
// menu class
function menu (a_items, a_tpl) {

    // browser check
    if (!document.body || !document.body.style)
        return;

    // store items structure
    this.a_config = a_items;
    // store template structure
    this.a_tpl = a_tpl;
    // get menu id

```

```

this.n_id = A_MENU.length;
// declare collections
this.a_index = [];
this.a_children = [];
// assign methods and event handlers
this.expand = menu_expand;
this.collapse = menu_collapse;
this.onclick = menu_onclick;
this.onmouseout = menu_onmouseout;
this.onmouseover = menu_onmouseover;
this.onmousedown = menu_onmousedown;
// default level scope description structure
this.a_tpl_def = {
    'block_top' : 16,
    'block_left' : 16,
    'top' : 20,
    'left' : 4,
    'width' : 120,
    'height' : 22,
    'hide_delay' : 0,
    'expd_delay' : 0,
    'css' : {
        'inner' : "",
        'outer' : ""
    }
};
// assign methods and properties required to imulate parent item
this.getprop = function (s_key) {
    return this.a_tpl_def[s_key];
};
this.o_root = this;
this.n_depth = -1;
this.n_x = 0;
this.n_y = 0;
// init items recursively
for (n_order = 0; n_order < a_items.length; n_order++)
    new menu_item(this, n_order);
// register self in global collection
A_MENU[this.n_id] = this;
// make root level visible
for (var n_order = 0; n_order < this.a_children.length; n_order++)
    this.a_children[n_order].e_oelement.style.visibility = 'visible';
}
// -----
function menu_collapse (n_id) {
    // cancel item open delay
    clearTimeout(this.o_showtimer);
    // by default collapse to root level
    var n_tolevel = (n_id ? this.a_index[n_id].n_depth : 0);
    // hide all items over the level specified
    for (n_id = 0; n_id < this.a_index.length; n_id++) {

```

```

        var o_curritem = this.a_index[n_id];
        if (o_curritem.n_depth > n_tolevel && o_curritem.b_visible) {
            o_curritem.e_oelement.style.visibility = 'hidden';
            o_curritem.b_visible = false;
        }
    }
    // reset current item if mouse has gone out of items
    if (!n_id)
        this.o_current = null;
}
// -----
function menu_expand (n_id) {
    // expand only when mouse is over some menu item
    if (this.o_hidetimer)
        return;

    // lookup current item
    var o_item = this.a_index[n_id];
    // close previously opened items
    if (this.o_current && this.o_current.n_depth >= o_item.n_depth)
        this.collapse(o_item.n_id);
    this.o_current = o_item;
    // exit if there are no children to open
    if (!o_item.a_children)
        return;
    // show direct child items
    for (var n_order = 0; n_order < o_item.a_children.length; n_order++) {
        var o_curritem = o_item.a_children[n_order];
        o_curritem.e_oelement.style.visibility = 'visible';
        o_curritem.b_visible = true;
    }
}
// -----
function menu_onclick (n_id) {
    // don't go anywhere if item has no link defined
    return Boolean(this.a_index[n_id].a_config[1]);
}
function esconde_select(){
    var ff, elementos, fram;
    fram = frames
    for(k=0;k<frames.length;k++){
        ff=frames[k].document.forms
        //alert('tam ao esconder'+ff.length);
        for(j=0;j<ff.length;j++){
            elementos = frames[k].document.forms[j].elements;
            for(i=0;i<elementos.length;i++){
                if(elementos[i].type=="select-one" || elementos[i].type=="select-
multiple"){
                    frames[k].document.forms[j].elements[i].style.visibility='hidden' ;
                }
            }
        }
    }
}

```

```

        }
    }
}
function mostra_select(){
    var ff, elementos, fram;
    fram = frames
    for(k=0;k<frames.length;k++){
        ff=frames[k].document.forms
        //alert('tam ao esconder'+ff.length);
        for(j=0;j<ff.length;j++){
            elementos = frames[k].document.forms[j].elements;
            for(i=0;i<elementos.length;i++){
                if(elementos[i].type=="select-one" || elementos[i].type=="select-
multiple"){
                    frames[k].document.forms[j].elements[i].style.visibility='visible' ;
                }
            }
        }
    }
}
function menu_onmouseout (n_id) {
    mostra_select();
    // lookup new item's object
    var o_item = this.a_index[n_id];

    // apply rollout
    o_item.e_oelement.className = o_item.getstyle(0, 0);
    o_item.e_ielement.className = o_item.getstyle(1, 0);

    // update status line
    o_item.upstatus(7);

    // run mouseover timer
    this.o_hidetimer = setTimeout('A_MENUUS['+ this.n_id +'].collapse();',
        o_item.getprop('hide_delay'));
}
function menu_onmouseover (n_id) {
    esconde_select();
    // cancel mouseoute menu close and item open delay
    clearTimeout(this.o_hidetimer);
    this.o_hidetimer = null;
    clearTimeout(this.o_showtimer);
    // lookup new item's object
    var o_item = this.a_index[n_id];

    // update status line
    o_item.upstatus();
    // apply rollover
    o_item.e_oelement.className = o_item.getstyle(0, 1);

```

```

o_item.e_ielelement.className = o_item.getstyle(1, 1);

// if onclick open is set then no more actions required
if (o_item.getprop('expd_delay') < 0)
    return;

// run expand timer
this.o_showtimer = setTimeout('A_MENUS['+ this.n_id +'].expand(' + n_id + ');',
    o_item.getprop('expd_delay'));
}

// -----
// called when mouse button is pressed on menu item
// -----
function menu_onmousedown (n_id) {

    // lookup new item's object
    var o_item = this.a_index[n_id];

    // apply mouse down style
    o_item.e_oelement.className = o_item.getstyle(0, 2);
    o_item.e_ielelement.className = o_item.getstyle(1, 2);

    this.expand(n_id);
// this.items[id].switch_style('onmousedown');
}
// menu item Class
function menu_item (o_parent, n_order) {

    // store parameters passed to the constructor
    this.n_depth = o_parent.n_depth + 1;
    this.a_config = o_parent.a_config[n_order + (this.n_depth ? 3 : 0)];

    // return if required parameters are missing
    if (!this.a_config) return;

    // store info from parent item
    this.o_root = o_parent.o_root;
    this.o_parent = o_parent;
    this.n_order = n_order;

    // register in global and parent's collections
    this.n_id = this.o_root.a_index.length;
    this.o_root.a_index[this.n_id] = this;
    o_parent.a_children[n_order] = this;

    // calculate item's coordinates
    var o_root = this.o_root,
        a_tpl = this.o_root.a_tpl;

```



```

// assign methods
this.getprop = mitem_getprop;
this.getstyle = mitem_getstyle;
this.upstatus = mitem_upstatus;

this.n_x = n_order
    ? o_parent.a_children[n_order - 1].n_x + this.getprop('left')
    : o_parent.n_x + this.getprop('block_left');

this.n_y = n_order
    ? o_parent.a_children[n_order - 1].n_y + this.getprop('top')
    : o_parent.n_y + this.getprop('block_top');

// generate item's HMTL
document.write (
    '<a id="e' + o_root.n_id + '_' +
        + this.n_id + 'o" class="' + this.getstyle(0, 0) + " href=" + this.a_config[1] + ""
        + (this.a_config[2] && this.a_config[2]['tw'] ? ' target="'
        + this.a_config[2]['tw'] + "" : ") + ' style="position: absolute; top: '
        + this.n_y + 'px; left: ' + this.n_x + 'px; width: '
        + this.getprop('width') + 'px; height: '
        + this.getprop('height') + 'px; visibility: hidden;'
        + ' z-index: ' + this.n_depth + ';' +
        + 'onclick="return A_MENU[" + o_root.n_id + '].onclick('
        + this.n_id + ');" onmouseout="A_MENU[" + o_root.n_id + '].onmouseout('
        + this.n_id + ');" onmouseover="A_MENU[" + o_root.n_id +
].onmouseover('
        + this.n_id + ');" onmousedown="A_MENU[" + o_root.n_id +
].onmousedown('
        + this.n_id + ');"><div id="e' + o_root.n_id + '_' +
        + this.n_id + 'i" class="' + this.getstyle(1, 0) + "">
        + this.a_config[0] + "</div></a>\n"
    );
this.e_ielement = document.getElementById('e' + o_root.n_id + '_' + this.n_id + 'i');
this.e_oelement = document.getElementById('e' + o_root.n_id + '_' + this.n_id + 'o');

this.b_visible = !this.n_depth;

// no more initialization if leaf
if (this.a_config.length < 4)
    return;

// node specific methods and properties
this.a_children = [];

// init downline recursively
for (var n_order = 0; n_order < this.a_config.length - 3; n_order++)
    new menu_item(this, n_order);
}

// -----

```

```

// reads property from template file, inherits from parent level if not found
// -----
function mitem_getprop (s_key) {

    // check if value is defined for current level
    var s_value = null,
        a_level = this.o_root.a_tpl[this.n_depth];

    // return value if explicitly defined
    if (a_level)
        s_value = a_level[s_key];

    // request recursively from parent levels if not defined
    return (s_value == null ? this.o_parent.getprop(s_key) : s_value);
}
// -----
// reads property from template file, inherits from parent level if not found
// -----
function mitem_getstyle (n_pos, n_state) {

    var a_css = this.getprop('css');
    var a_oclass = a_css[n_pos ? 'inner' : 'outer'];

    // same class for all states
    if (typeof(a_oclass) == 'string')
        return a_oclass;

    // inherit class from previous state if not explicitly defined
    for (var n_currst = n_state; n_currst >= 0; n_currst--)
        if (a_oclass[n_currst])
            return a_oclass[n_currst];
}

// -----
// updates status bar message of the browser
// -----
function mitem_upstatus (b_clear) {
    window.setTimeout("window.status=unescape('" + (b_clear
        ? "
        : (this.a_config[2] && this.a_config[2]['sb']
        ? escape(this.a_config[2]['sb'])
        : escape(this.a_config[0]) + (this.a_config[1]
        ? ' (' + escape(this.a_config[1]) + ' '
        : "))) + '"", 10);
}

// -----
// that's all folks

menu_items.js

```

```

var MENU_ITEMS = [
  ['Cadastros', null, null,
    ['Corporativos', null, null,
      ['Empresas', 'empresa.do', {'tw' : 'principal'}],
      ['Setores', 'setor.do', {'tw' : 'principal'}],
      ['Colaboradores', 'colaborador.do', {'tw' : 'principal'}],
    ],
    ['Processo de Formação', null, null,
      ['Perguntas', 'pergunta_a.do', {'tw' : 'principal'}],
      ['Respostas', 'resposta_a.do', {'tw' : 'principal'}],
    ],
    ['Escolaridade', 'escolaridade.do', {'tw' : 'principal'}],
    ['Usuários', 'usuario.do', {'tw' : 'principal'}],
  ],
  ['Resultados', 'resultado_b.do', {'tw' : 'principal'}],
  ['Estatística de Acesso', 'estatistica_acesso.do', {'tw' : 'principal'}],
  ['Saída', 'javascript:fecha()'];
];

```

menu_tpl.js

```

var MENU_POS = [
{
  // item sizes
  'height': 16,
  'width': 140,
  // menu block offset from the origin:
  // for root level origin is upper left corner of the page
  //AQUI MEXE NA POSIÇÃO DO MENU!!
  'block_top': 119,
  'block_left': 20,
  // offsets between items of the same level
  'top': 0,
  'left': 130,
  // time in milliseconds before menu is hidden after cursor has gone out
  // of any items
  'hide_delay': 200,
  'expd_delay': 200,
  'css' : {
    'outer': ['m010oout', 'm010oover'],
    'inner': ['m010iout', 'm010iover']
  }
},
{
  'height': 14,
  'width': 200,
  'block_top': 23,
  'block_left': 30,
  'top': 21,
  'left': 0,

```

```

    'css': {
        'outer': ['m011oout', 'm011oover'],
        'inner': ['m011iout', 'm011iover']
    }
},
{
    'block_top': 15,
    'block_left': 50,
    'css': {
        'outer': ['m012oout', 'm012oover'],
        'inner': ['m011iout', 'm012iover']
    }
},
]

```

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>
<script>
        function placeFocus() {
            if (document.forms.length > 0) {
                var field = document.forms[0];
                for (i = 0; i < field.length; i++) {
                    if ((field.elements[i].type == "text") || (field.elements[i].type ==
"textarea") || (field.elements[i].type.toString().charAt(0) == "s")) {
                        document.forms[0].elements[i].focus();
                        break;
                    }
                }
            }
        }
    </script>
<link href="styles/css_crescer.css" rel="stylesheet" type="text/css">
<title>Crescer - Motivação para o Conhecimento</title>
<meta http-equiv="Content-Type" content="text/html;">
<!-- Fireworks MX Dreamweaver MX target. Created Tue Apr 19 00:14:52 GMT-0300 (Hora
oficial do Brasil) 2005-->
</head>
<body bgcolor="#ffffff" topmargin="0" leftmargin="0" rightmargin="0" bottommargin="0"
scroll="no" onload="placeFocus()">
    <html:form action="/login">
        <table width="100%" height="100%" border="0" cellpadding="0" cellspacing="0">
        <tr>
            <td valign="top" background="imagens/bg.gif" height="118" style="border-
bottom: 20px solid #BFD58A">
                <table border="0" cellpadding="0" cellspacing="0" width="760"
height="118">

```



```

var mydate=new Date()
var year=mydate.getYear()

if (year<2000)
    year="19"+year
var
var
var
if (daym<10)

    daym="0"+daym
var
dayarray=new      Array("Domingo","Segunda-Feira","Terça-Feira","Quarta-Feira","Quinta-Feira","Sexta-
Feira","Sábado")
var
montharray=new
Array("Janeiro","Fevereiro","Março","Abril","Maio","Junho","Julho","Agosto","Setembro","Outubro","Nove
mbro","Desembro")

document.write("<font class=TxtTitulo>"+dayarray[day]+", "+daym+" de "+montharray[month]+" de
"+year+".</font>")
</script>
<br><br>Bem-vindo ao Crescer. Por
favor, identifique-se.<br>
</blockquote>

</td>
</tr>
<tr>
<td class="TxtTitulo" align="right">Usuário:</td>
<td width="50"><html:text styleClass="campo"
maxlength="30" style="text-align: left" name="usuarioForm" property="usuario.nm_usuario"/></td>
<tr>
<td class="TxtTitulo" align="right">Senha:</td>
<td width="50"><html:password styleClass="campo"
maxlength="30" style="text-align: left" name="usuarioForm" property="usuario.ds_senha"/></td>
<td></td>
</tr>
<tr>
<td class="TxtGeral" align="center" colspan="2">
<html:submit styleClass="button"
property="evento">
<bean:message
key="aprendizado_crm.login.valida"/>
</html:submit>

```

```

        </td>
      </tr>
    </table>
  </td>
</tr>
<tr>
  <td class="TxtGeral">
    <blockquote><html:messages id="msg" message="true">
      <bean:write name="msg"/><br>
    </html:messages></blockquote>
  </td>
</tr>
<!--<tr>
  <td bgcolor="#BFD58A" height="20"></td>
</tr-->
<tr>
  <td bgcolor="#BFD58A" height="70" class="TxtTitulo"><center>"Não
importa a maneira que o cliente use para contatar.<br> A informação tem que fluir dentro da empresa, para que
exista um entendimento pleno da necessidade do cliente."
<br>Claúdio Szajman, Presidente do Grupo VR.</center> </td>
</tr>
</table>
</html:form>
</body>
</html>

```

administrador.jsp

```

<html>
<head>
<title>Crescer - Motivação para o Conhecimento - Sessão Administrativa</title>
<link href="styles/css_register.css" rel="stylesheet" type="text/css">
<link rel="stylesheet" href="menu/menu.css">
<script>
function fecha(){
  window.defaultStatus="Sair do Sistema";
  top.close();
}
function manual(){
  window.open("manual.jsp","Janela_manual","scrollbars = 1 , resizable = 1,width =
760,top = 0,left = 0,height = 550,menubar=0,status=1");
}
</script>
</head>
<body topmargin="0" leftmargin="0" rightmargin="0" bottommargin="0" scroll=no
bgcolor="#ffffff">
<script language="JavaScript" src="menu/menu.js"></script>
<script language="JavaScript" src="menu/menu_items.js"></script>
<script language="JavaScript" src="menu/menu_tpl.js"></script>
<script language="JavaScript">
  new menu (MENU_ITEMS, MENU_POS);

```

```

</script>
<!--<table width="100%" height="120" border="0" cellpadding="0" cellspacing="0"
width="760" >
  <tr>
    <td background="imagens/bg_admin.jpg">
      <table border="0" cellpadding="0" cellspacing="0" width="760" height="120"
>
        <!-- fwtable fwsrc="beta.png" fwbase="administrador.png"
fwstyle="Dreamweaver" fwdocid = "742308039" fwnested="0"
        <tr>
          <td></td>
          <td></td>
          <td></td>
          <td></td>
        </tr>
        <tr>
          <td colspan="2"></td>
          <td rowspan="3"></td>
          <td></td>
        </tr>
        <tr>
          <td colspan="2" background="imagens/administrador_r2_c1.jpg"></td>
          <td rowspan="2"></td>
          <td></td>
        </tr>
        <tr>
          <td colspan="2"></td>
          <td></td>
        </tr>
      </table>
    </td>
  </tr>
</table-->
<table width="100%" height="118" border="0" cellpadding="0" cellspacing="0">
  <tr>
    <td valign="top" background="imagens/bg.gif" height="118" style="border-
bottom: 20px solid #BFD58A">

```



```

height="118">
<table border="0" cellpadding="0" cellspacing="0" width="760"
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</table>
</td>
</tr>
</table>

```

```

<IFRAME align="center" height="74%" frame name="principal" src=""
width="100%" scrolling="auto" frameborder="no"> </IFRAME>
</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">
</a>
</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"/>')"></a>
            <a href="javascript:submete('<bean:write name="<%=p_nm_form%>"
property="pagina_ant"/>')"></a>
        </logic:equal>

        <logic:equal name="<%=p_nm_form%>" property="mostraProximos" value="N">
            </a>
            </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"/>')"></a>
            <a href="javascript:submete('<bean:write name="<%=p_nm_form%>"
property="pagina_ult"/>')"></a>
        </logic:equal>

    </td>
</tr>
<tr>
    <td align="center">
        <html:submit property="evento">
            <bean:message key="aprendizado_crm.entra_inserir"/>
        </html:submit>
        <html:submit property="evento">
            <bean:message key="aprendizado_crm.entra_consultar"/>
        </html:submit>
    </td>
</tr>
</table>

```

barra_estadistica.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">
      </a>
      </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"/>')"></a>
      <a href="javascript:submete('<bean:write name="<%=p_nm_form%>"
property="pagina_ant"/>')"></a>
    </logic:equal>

    <logic:equal name="<%=p_nm_form%>" property="mostraProximos" value="N">
      </a>
      </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"/>')"></a>
      <a href="javascript:submete('<bean:write name="<%=p_nm_form%>"
property="pagina_ult"/>')"></a>
    </logic:equal>

  </td>
</tr>
</table>

```

barra_resultado.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">
      </a>
      </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"/>')"></a>
                <a href="javascript:submete('<bean:write name="<%=p_nm_form%>"
property="pagina_ant"/>')"></a>
                </logic:equal>

                <logic:equal name="<%=p_nm_form%>" property="mostraProximos" value="N">
                </a>
                </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"/>')"></a>
                <a href="javascript:submete('<bean:write name="<%=p_nm_form%>"
property="pagina_ult"/>')"></a>
                </logic:equal>

        </td>
</tr>
<tr>
<td align="center">
        <html:submit property="evento">
                <bean:message key="aprendizado_crm.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"/>&nbsp; de &nbsp;
      <bean:write name="<%=p_nm_form%>" property="contador"/>&nbsp; linhas
selecionadas
    </td>
  </tr>
</table>

```

editar_colaborador.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>Colaborador</title>
<link href="styles/css_crescer.css" rel="stylesheet" type="text/css">
<script>
  function placeFocus() {
    if (document.forms.length > 0) {
      var field = document.forms[0];
      for (i = 0; i < field.length; i++) {
        if (!(field.elements[i].isDisabled)){
          if ((field.elements[i].type == "text") ||
(field.elements[i].type == "textarea") || (field.elements[i].type.toString().charAt(0) == "s")) {
            document.forms[0].elements[i].focus();
            break;
          }
        }
      }
    }
  }

  function obrigatorios() {
    if ((document.getElementsByName('colaborador.id_sexo')[0].value == "")
||((document.getElementsByName('colaborador.dt_admissao')[0].value == ""))
||((document.getElementsByName('colaborador.cd_escolaridade')[0].value == ""))
||((document.getElementsByName('colaborador.cd_setor')[0].value == ""))
||((document.getElementsByName('colaborador.cd_colaborador')[0].value == ""))
||((document.getElementsByName('colaborador.cd_emp')[0].value == ""))
||((document.getElementsByName('colaborador.nm_colaborador')[0].value == ""))){
      alert("Preencha Todos os Campos Obrigatórios");
      placeFocus();
      return false;
    }
  }
</script>
</head>

```

```

<body bgcolor="#ffffff" onload="placeFocus">
<html:form action="/colaborador">
<input type="hidden" name="action">
<center><font class="TxtTitulo">Colaborador - Cadastros<br><br></font></center>
<table width="75%" border="1" bgcolor="#ffffff" bordercolor="#63659C" align="center">
  <tr>
    <td>
      <table width="100%" border="0" bgcolor="#ffffff">
        <tr>
          <logic:notEqual    name="colaboradorForm"    property="mostraConsulta"
value="S">
            <html:hidden name="colaboradorForm" property="colaborador.oid"/>
            <td colspan="5" align="center">
              <logic:present                                name="colaboradorForm"
property="colaborador.oid">
                <html:submit    onclick="return    obrigatorios()"
property="evento">
                  <bean:message
key="aprendizado_crm.alterar"/>
                </html:submit>
              </logic:present>
            <logic:notPresent                                name="colaboradorForm"
property="colaborador.oid">
                <html:submit    onclick="return    obrigatorios()"
property="evento">
                  <bean:message
key="aprendizado_crm.inserir"/>
                </html:submit>
              </logic:notPresent>
            </td>
          </tr>
        <tr>
          <td align="right" class="TxtTitulo">Empresa:
          <td>
            <logic:present                                name="colaboradorForm"
property="colaborador.oid">
              <html:text          title="Campo          Desabilitado"
styleClass="campoCodigo" maxlength="2" style="text-align: right" name="colaboradorForm" readonly="true"
property="colaborador.cd_emp" />
            </logic:present>
            <logic:notPresent                                name="colaboradorForm"
property="colaborador.oid">
              <html:text          title="Digite o C3digo da Empresa"
styleClass="campoCodigo" maxlength="2" size="2" style="text-align: right" name="colaboradorForm"
property="colaborador.cd_emp" />
            </logic:notPresent>
          </td>
        </tr>
        <tr>
          <td align="right" class="TxtTitulo">Setor:

```

```

                <td><html:text title="Digite o Código do Setor"
styleClass="campoCodigo" maxLength="2" size="2" style="text-align: right" name="colaboradorForm"
property="colaborador.cd_setor" /></td>
            </tr>
            <tr>
                <td align="right" class="TxtTitulo">Colaborador:
                <td>
                    <logic:present name="colaboradorForm"
property="colaborador.oid">
                        <html:text title="Campo Desabilitado"
styleClass="campoCodigo" maxLength="2" style="text-align: right" name="colaboradorForm" readonly="true"
property="colaborador.cd_colaborador" />
                    </logic:present>
                    <logic:notPresent name="colaboradorForm"
property="colaborador.oid">
                        <html:text title="Digite o Código da Colaborador"
styleClass="campoCodigo" maxLength="2" size="2" style="text-align: right" name="colaboradorForm"
property="colaborador.cd_colaborador" />
                    </logic:notPresent>
                    <html:text styleClass="endereco" title="Digite o Nome do
Colaborador" maxLength="50" name="colaboradorForm" property="colaborador.nm_colaborador" />
            </tr>
            <tr>
                <td align="right" class="TxtTitulo">Data de Admissão:
                <td><html:text title="Digite a Data de Admissão no formato
dd/mm/aaaa" styleClass="campo" maxLength="10" style="text-align: center" name="colaboradorForm"
property="colaborador.dt_admissao" /></td>
            </tr>
            <tr>
                <td align="right" class="TxtTitulo">Escolaridade:
                <td><html:text title="Digite o Código da Escolaridade"
styleClass="campoCodigo" maxLength="2" size="2" style="text-align: right" name="colaboradorForm"
property="colaborador.cd_escolaridade" /></td>
            </tr>
            <tr>
                <td align="right" class="TxtTitulo">Sexo:</td>
                <td>
                    <select name="colaborador.id_sexo" class="TxtGeral">
                        <option value="F">Feminino</option>
                        <option value="M">Masculino</option>
                    </select>
                </td>
            </tr>
            <tr>
                <td align="right" class="TxtTitulo"></td>
                <td>
                    <html:hidden name="colaboradorForm"
property="colaborador_pesq.cd_emp"/>
                    <html:hidden name="colaboradorForm"
property="colaborador_pesq.cd_setor"/>
            </tr>

```

```

        <html:hidden                                name="colaboradorForm"
property="colaborador_pesq.cd_colaborador"/>
        <html:hidden                                name="colaboradorForm"
property="colaborador_pesq.nm_colaborador"/>
        <html:hidden                                name="colaboradorForm"
property="colaborador_pesq.id_sexo"/>
        <html:hidden                                name="colaboradorForm"
property="colaborador_pesq.dt_admissao"/>
        <html:hidden                                name="colaboradorForm"
property="colaborador_pesq.cd_escolaridade"/>
        </logic:notEqual>

        <logic:equal                                name="colaboradorForm"    property="mostraConsulta"
value="S">
            <tr>
                <td colspan="5" align="center">
                    <html:submit property="evento">
                        <bean:message key="aprendizado_crm.lista"/>
                    </html:submit>
                </td>
            </tr>
            <tr>
                <td align="right" class="TxtTitulo">Empresa:
                <td><html:text    title="Digite o Código do Setor"
styleClass="campoCodigo"    maxlength="2"    style="text-align: right"    name="colaboradorForm"
property="colaborador_pesq.cd_emp" /></td>
            </tr>
            <tr>
                <td align="right" class="TxtTitulo">Setor:
                <td><html:text    title="Digite o Código do Setor"
styleClass="campoCodigo"    maxlength="2"    style="text-align: right"    name="colaboradorForm"
property="colaborador_pesq.cd_setor" /></td>
            </tr>
            <tr>
                <td align="right" class="TxtTitulo">Colaborador:
                <td>
                    <html:text    title="Digite o Código da Colaborador"
styleClass="campoCodigo"    maxlength="2"    style="text-align: right"    name="colaboradorForm"
property="colaborador_pesq.cd_colaborador" />
                    <html:text    title="Digite o Nome do Colaborador"
styleClass="endereco"        maxlength="50"        name="colaboradorForm"
property="colaborador_pesq.nm_colaborador" />
            </tr>
            <td align="right" class="TxtTitulo">Data de Admissão:
            <td><html:text    title="Digite a Data de Admissão no formato
dd/mm/aaaa" styleClass="campo"    maxlength="10"    style="text-align: center"    name="colaboradorForm"
property="colaborador_pesq.dt_admissao" /></td>
            </tr>
            <tr>

```



```

        <td align="right" class="TxtTitulo">Escolaridade:
        <td><html:text title="Digite o Código da Escolaridade"
styleClass="campoCodigo" maxlength="2" style="text-align: right" name="colaboradorForm"
property="colaborador_pesq.cd_escolaridade" /></td>
    </tr>
    <tr>
        <td align="right" class="TxtTitulo">Sexo:</td>
        <td>
            <select name="colaborador_pesq.id_sexo" class="TxtGeral">
                <option value="">Todos</option>

                <option value="F">Feminino</option>

                <option value="M">Masculino</option>
            </select>
        </td>
    </tr>
</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>
<link href="styles/css_crescer.css" rel="stylesheet" type="text/css">
<script>
    function placeFocus() {
        if (document.forms.length > 0) {
            var field = document.forms[0];
            for (i = 0; i < field.length; i++) {
                if (!(field.elements[i].isDisabled)){
                    if ((field.elements[i].type == "text") ||
(field.elements[i].type == "textarea") || (field.elements[i].type.toString().charAt(0) == "s")) {
                        document.forms[0].elements[i].focus();
                        break;
                    }
                }
            }
        }
    }

```



```

                                <logic:present                                name="empresaForm"
property="empresa.oid">
                                <html:text                                title="Campo
                                Desabilitado"
styleClass="campoCodigo" maxLength="2" style="text-align: right" name="empresaForm" readonly="true"
property="empresa.cd_emp" />
                                </logic:present>
                                <logic:notPresent                                name="empresaForm"
property="empresa.oid">
                                <html:text                                title="Digite
                                o Código da Empresa"
styleClass="campoCodigo" maxLength="2" size="2" style="text-align: right" name="empresaForm"
property="empresa.cd_emp" />
                                </logic:notPresent>
                                <html:text styleClass="endereco" title="Digite o Nome da
Empresa" maxLength="50" name="empresaForm" property="empresa.nm_emp" />
                                <html:hidden                                name="empresaForm"
property="empresa_pesq.cd_emp"/>
                                <html:hidden                                name="empresaForm"
property="empresa_pesq.nm_emp"/>
                                </logic:notEqual>
                                <logic:equal name="empresaForm" property="mostraConsulta" value="S">
                                <tr>
                                <td colspan="5" align="center">
                                <html:submit property="evento">
                                <bean:message key="aprendizado_crm.lista"/>
                                </html:submit>
                                </td>
                                </tr>
                                <tr>
                                <td align="right" class="TxtTitulo">Empresa:
                                <td>
                                <html:text                                title="Digite
                                o Código da Empresa"
styleClass="campoCodigo" maxLength="2" style="text-align: right" name="empresaForm"
property="empresa_pesq.cd_emp" />
                                <html:text                                title="Digite
                                o Nome da Empresa"
styleClass="endereco" maxLength="50" name="empresaForm" property="empresa_pesq.nm_emp" />
                                </logic:equal>
                                </table>
                                </td>
                                </tr>
                                </table>
                                <html:messages id="msg" message="true">
                                <bean:write name="msg"/><br>
                                </html:messages>
                                </html:form>
                                </body>
                                </html>

```

editar_escolaridade.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>Escolaridade</title>
<link href="styles/css_crescer.css" rel="stylesheet" type="text/css">
<script>
    function placeFocus() {
        if (document.forms.length > 0) {
            var field = document.forms[0];
            for (i = 0; i < field.length; i++) {
                if (!(field.elements[i].isDisabled)){
                    if ((field.elements[i].type == "text") ||
(field.elements[i].type == "textarea") || (field.elements[i].type.toString().charAt(0) == "s")) {
                        document.forms[0].elements[i].focus();
                        break;
                    }
                }
            }
        }
    }

    function obrigatorios() {
        if ((document.getElementsByName('escolaridade.cd_escolaridade')[0].value == "") ||
(document.getElementsByName('escolaridade.ds_escolaridade')[0].value == "")){
            alert("Preencha Todos os Campos Obrigatórios");
            placeFocus();
            return false;
        }
    }
</script>
</head>
<body bgcolor="#ffffff" onload="placeFocus">
<html:form action="/escolaridade">
<input type="hidden" name="action">
<center><font class="TxtTitulo">Escolaridade - Cadastros<br><br></font></center>
<table width="75%" border="1" bgcolor="#ffffff" bordercolor="#63659C" align="center">
<tr>
<td>
<table width="100%" border="0" bgcolor="#ffffff">
<tr>
<td colspan="2">
<input type="text" name="escolaridadeForm" value="S"
property="mostraConsulta"
/>
<input type="hidden" name="escolaridadeForm"
property="escolaridade.oid"/>
<td colspan="5" align="center">
<input type="text" name="escolaridadeForm"
property="escolaridade.oid"/>
<input type="submit" value="Pesquisar"
onclick="return obrigatorios()"
property="evento"/>

```

```

key="aprendizado_crm.alterar"/>
        <bean:message
        </html:submit>
        </logic:present>
        <logic:notPresent
        name="escolaridadeForm"
property="escolaridade.oid">
        <html:submit
        onclick="return
        obrigatorios()"
property="evento">
        <bean:message
key="aprendizado_crm.inserir"/>
        </html:submit>
        </logic:notPresent>
    </td>
</tr>
<tr>
    <td align="right" class="TxtTitulo">Escolaridade:
    <td>
        <logic:present
        name="escolaridadeForm"
property="escolaridade.oid">
            <html:text
            title="Campo
            Desabilitado"
styleClass="campoCodigo"
maxlength="2"
style="text-align:
            right"
            name="escolaridadeForm"
readonly="true"
property="escolaridade.cd_escolaridade" />
            </logic:present>
            <logic:notPresent
            name="escolaridadeForm"
property="escolaridade.oid">
                <html:text
                title="Digite o
                Código da
                Escolaridade"
styleClass="campoCodigo"
maxlength="2"
size="2"
style="text-align:
                right"
                name="escolaridadeForm"
property="escolaridade.cd_escolaridade" />
                </logic:notPresent>
                <html:text
                styleClass="endereco"
                title="Digite a
                Descrição da
                Escolaridade"
                maxlength="50"
                name="escolaridadeForm"
                property="escolaridade.ds_escolaridade" />
                <html:hidden
                name="escolaridadeForm"
property="escolaridade_pesq.cd_escolaridade"/>
                <html:hidden
                name="escolaridadeForm"
property="escolaridade_pesq.ds_escolaridade"/>
                </logic:notEqual>
                <logic:equal
                name="escolaridadeForm"
                property="mostraConsulta"
value="S">
            <tr>
                <td colspan="5" align="center">
                    <html:submit
                    property="evento">
                        <bean:message
                        key="aprendizado_crm.lista"/>
                    </html:submit>
                </td>
            </tr>
        <tr>
            <td align="right" class="TxtTitulo">Escolaridade:
            <td>

```

```

                                <html:text title="Digite o Código da Escolaridade"
styleClass="campoCodigo"    maxLength="2"    style="text-align: right"    name="escolaridadeForm"
property="escolaridade_pesq.cd_escolaridade" />
                                <html:text title="Digite a Descrição da Escolaridade"
styleClass="endereco"      maxLength="50"      name="escolaridadeForm"
property="escolaridade_pesq.ds_escolaridade" />
                                </logic:equal>
                                </table>
                                </td>
                                </tr>
                                </table>
                                <html:messages id="msg" message="true">
                                <bean:write name="msg"/><br>
                                </html:messages>
                                </html:form>
                                </body>
                                </html>

```

editar_pergunta.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>Pergunta</title>
<link href="styles/css_crescer.css" rel="stylesheet" type="text/css">
<script>
        function placeFocus() {
            if (document.forms.length > 0) {
                var field = document.forms[0];
                for (i = 0; i < field.length; i++) {
                    if (!(field.elements[i].isDisabled)){
                        if ((field.elements[i].type == "text") ||
(field.elements[i].type == "textarea") || (field.elements[i].type.toString().charAt(0) == "s")) {
                            document.forms[0].elements[i].focus();
                            break;
                        }
                    }
                }
            }
        }

        function obrigatorios() {
            if ((document.getElementsByName('pergunta.cd_pergunta')[0].value == "") ||
(document.getElementsByName('pergunta.ds_pergunta')[0].value == "")){
                alert("Preencha Todos os Campos Obrigatórios");
                placeFocus();
                return false;
            }
        }

```

```

</script>
</head>
<body bgcolor="#ffffff" onload="placeFocus">
<html:form action="/pergunta_a">
<input type="hidden" name="action">
<center><font class="TxtTitulo">Pergunta - Cadastros<br><br></font></center>
<table width="75%" border="1" bgcolor="#ffffff" bordercolor="#63659C" align="center">
  <tr>
    <td>
      <table width="100%" border="0" bgcolor="#ffffff">
        <tr>
          <logic:notEqual      name="perguntaForm"      property="mostraConsulta"
value="S">
            <html:hidden name="perguntaForm" property="pergunta.oid"/>
            <td colspan="5" align="center">
              <logic:present                                name="perguntaForm"
property="pergunta.oid">
                <html:submit      onclick="return      obrigatorios()"
property="evento">
                  <bean:message
key="aprendizado_crm.alterar"/>
                </html:submit>
              </logic:present>
            <logic:notPresent                                name="perguntaForm"
property="pergunta.oid">
                <html:submit      onclick="return      obrigatorios()"
property="evento">
                  <bean:message
key="aprendizado_crm.inserir"/>
                </html:submit>
              </logic:notPresent>
            </td>
          </tr>
        <tr>
          <td align="right" class="TxtTitulo">Pergunta:
          <td>
            <logic:present                                name="perguntaForm"
property="pergunta.oid">
              <html:text      title="Campo      Desabilitado"
styleClass="campoCodigo" maxlength="2" style="text-align: right" name="perguntaForm" readonly="true"
property="pergunta.cd_pergunta" />
            </logic:present>
            <logic:notPresent                                name="perguntaForm"
property="pergunta.oid">
              <html:text      title="Digite o C3digo da Pergunta"
styleClass="campoCodigo" maxlength="2" size="2" style="text-align: right" name="perguntaForm"
property="pergunta.cd_pergunta" />
            </logic:notPresent>
            <html:text styleClass="endereco" title="Digite o Nome da
Pergunta" maxlength="150" name="perguntaForm" property="pergunta.ds_pergunta" />

```

```

        <html:hidden name="perguntaForm"
property="pergunta_pesq.cd_pergunta"/>
        <html:hidden name="perguntaForm"
property="pergunta_pesq.ds_pergunta"/>
    </logic:notEqual>

    <logic:equal name="perguntaForm" property="mostraConsulta" value="S">
        <tr>
            <td colspan="5" align="center">
                <html:submit property="evento">
                    <bean:message key="aprendizado_crm.lista"/>
                </html:submit>
            </td>
        </tr>
        <tr>
            <td align="right" class="TxtTitulo">Pergunta:
            <td>
                <html:text title="Digite o Código da Pergunta"
styleClass="campoCodigo" maxlength="2" style="text-align: right" name="perguntaForm"
property="pergunta_pesq.cd_pergunta" />
                <html:text title="Digite o Nome da Pergunta"
styleClass="endereco" maxlength="150" name="perguntaForm" property="pergunta_pesq.ds_pergunta" />
            </logic:equal>
        </table>
    </td>
</tr>
</table>
<html:messages id="msg" message="true">
    <bean:write name="msg"/><br>
</html:messages>
</html:form>
</body>
</html>

```

editar_resposta_a.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>Resposta</title>
<link href="styles/css_crescer.css" rel="stylesheet" type="text/css">
<script>
    function placeFocus() {
        if (document.forms.length > 0) {
            var field = document.forms[0];
            for (i = 0; i < field.length; i++) {
                if (!(field.elements[i].isDisabled)){
                    if ((field.elements[i].type == "text") ||
(field.elements[i].type == "textarea") || (field.elements[i].type.toString().charAt(0) == "s")) {

```



```

document.forms[0].elements[i].focus();
break;
}
}
}
}
}

function obrigatorios() {
    if ((document.getElementsByName('resposta.id_status')[0].value == "" ||
(document.getElementsByName('resposta.cd_pergunta')[0].value == "" ||
|(document.getElementsByName('resposta.cd_resposta')[0].value == "" ||
(document.getElementsByName('resposta.ds_resposta')[0].value == ""))) {
        alert("Preencha Todos os Campos Obrigatórios");
        placeFocus();
        return false;
    }
}
</script>
</head>
<body bgcolor="#ffffff" onload="placeFocus">
<html:form action="/resposta_a">
<input type="hidden" name="action">
<center><font class="TxtTitulo">Resposta - Cadastros<br><br></font></center>
<table width="75%" border="1" bgcolor="#ffffff" bordercolor="#63659C" align="center">
<tr>
<td>
<table width="100%" border="0" bgcolor="#ffffff">
<tr>
<td colspan="5" style="text-align: center;">
<table border="1" style="width: 100%; border-collapse: collapse;">
<tr>
 value="S"> |  | <html:submit type="button" value="Consultar" property="respostaForm" onclick="return obrigatorios()" name="respostaForm" /> |  |  |  |  |  |  |
```

```

        <tr>
            <td align="right" class="TxtTitulo">Pergunta:
            <td>
                <logic:present
                    name="respostaForm"
                    property="resposta.oid">
                    <html:text
                        title="Campo Desabilitado"
                        styleClass="campoCodigo"
                        maxLength="2"
                        style="text-align: right"
                        name="respostaForm"
                        readOnly="true"
                        property="resposta.cd_pergunta" />
                    </logic:present>
                    <logic:notPresent
                        name="respostaForm"
                        property="resposta.oid">
                    <html:text
                        title="Digite o Código da Pergunta"
                        styleClass="campoCodigo"
                        maxLength="2"
                        size="2"
                        style="text-align: right"
                        name="respostaForm"
                        property="resposta.cd_pergunta" />
                    </logic:notPresent>
                </td>
            </tr>
            <tr>
            <td align="right" class="TxtTitulo">Resposta:
            <td>
                <logic:present
                    name="respostaForm"
                    property="resposta.oid">
                    <html:text
                        title="Campo Desabilitado"
                        styleClass="campoCodigo"
                        maxLength="2"
                        style="text-align: right"
                        name="respostaForm"
                        readOnly="true"
                        property="resposta.cd_resposta" />
                    </logic:present>
                    <logic:notPresent
                        name="respostaForm"
                        property="resposta.oid">
                    <html:text
                        title="Digite o Código da Resposta"
                        styleClass="campoCodigo"
                        maxLength="2"
                        size="2"
                        style="text-align: right"
                        name="respostaForm"
                        property="resposta.cd_resposta" />
                    </logic:notPresent>
                    <html:text
                        styleClass="endereco"
                        title="Digite a Descrição da Resposta"
                        maxLength="150"
                        name="respostaForm"
                        property="resposta.ds_resposta" />
                </td>
            </tr>
            <tr>
            <td align="right" class="TxtTitulo">Status:</td>
            <td>
                <select name="resposta.id_status" class="TxtGeral">
                    <option value="C">Correta</option>
                    <option value="E">Incorreta</option>
                </select>
            </td>
        </tr>
        <html:hidden
            name="respostaForm"
            property="resposta_pesq.cd_pergunta"/>
        <html:hidden
            name="respostaForm"
            property="resposta_pesq.cd_resposta"/>
        <html:hidden
            name="respostaForm"
            property="resposta_pesq.ds_resposta"/>

```

```

        <html:hidden name="respostaForm"
property="resposta_pesq.id_status"/>
    </logic:notEqual>

    <logic:equal name="respostaForm" property="mostraConsulta" value="S">
        <tr>
            <td colspan="5" align="center">
                <html:submit property="evento">
                    <bean:message key="aprendizado_crm.lista"/>
                </html:submit>
            </td>
        </tr>
        <tr>
            <td align="right" class="TxtTitulo">Pergunta:
            <td>
                <html:text title="Digite o Código da Pergunta"
styleClass="campoCodigo" maxlength="2" style="text-align: right" name="respostaForm"
property="resposta_pesq.cd_pergunta" />
            <tr>
            <td align="right" class="TxtTitulo">Resposta:
            <td>
                <html:text title="Digite o Código da Resposta"
styleClass="campoCodigo" maxlength="2" style="text-align: right" name="respostaForm"
property="resposta_pesq.cd_resposta" />
                <html:text title="Digite a Descrição da Resposta"
styleClass="endereco" maxlength="150" name="respostaForm" property="resposta_pesq.ds_resposta" />
            <tr>
            <td align="right" class="TxtTitulo">Status:</td>
            <td>
                <select name="resposta_pesq.id_status" class="TxtGeral">
                    <option value="">Todas</option>

                    <option value="C">Correta</option>

                    <option value="E">Incorreta</option>
                </select>
            </td>
        </tr>
    </logic:equal>
</table>
</td>
</tr>
</table>
<html:messages id="msg" message="true">
    <bean:write name="msg"/><br>
</html:messages>
</html:form>
</body>
</html>

```

editar_resultado_b.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>Resultado</title>
<link href="styles/css_crescer.css" rel="stylesheet" type="text/css">
<script>
    function placeFocus() {
        if (document.forms.length > 0) {
            var field = document.forms[0];
            for (i = 0; i < field.length; i++) {
                if (!(field.elements[i].isDisabled)){
                    if ((field.elements[i].type == "text") ||
(field.elements[i].type == "textarea") || (field.elements[i].type.toString().charAt(0) == "s")) {
                        document.forms[0].elements[i].focus();
                        break;
                    }
                }
            }
        }
    }
</script>
</head>
<body bgcolor="#ffffff" onload="placeFocus">
<html:form action="/resultado_b">
<input type="hidden" name="action">
<center><font class="TxtTitulo">Resultado<br><br></font></center>
<table width="75%" border="1" bgcolor="#ffffff" bordercolor="#63659C" align="center">
<tr>
<td>
<table width="100%" border="0" bgcolor="#ffffff">
<logic:equal name="resultadoForm" property="mostraConsulta" value="S">
<tr>
<td colspan="5" align="center">
<html:submit property="evento">
<bean:message key="aprendizado_crm.lista"/>
</html:submit>
</td>
</tr>
</tr>
<tr>
<td align="right" class="TxtTitulo">Usuário:</td>
<td><html:text title="Digite o Nome do Usuário" styleClass="campo"
maxlength="30" name="resultadoForm" property="resultado_pesq.nm_usuario" /></td>
</tr>
<tr>
<td align="right" class="TxtTitulo">Pergunta:</td>
```

```

                <td><html:text title="Digite o Código da Pergunta"
styleClass="campoCodigo" maxlength="2" style="text-align: right" name="resultadoForm"
property="resultado_pesq.cd_pergunta" /></td>
            </tr>
            <tr>
                <td align="right" class="TxtTitulo">Resposta:</td>
                <td><html:text title="Digite o Código da Resposta"
styleClass="campoCodigo" maxlength="2" style="text-align: right" name="resultadoForm"
property="resultado_pesq.cd_resposta" /></td>
            </tr>
            <tr>
                <td align="right" class="TxtTitulo">Status:</td>
                <td>
                    <select name="resultado_pesq.id_status" class="TxtGeral">
                        <option value="">Todas</option>
                        <option value="C">Correta</option>
                        <option value="E">Incorreta</option>
                    </select>
                </td>
            </tr>
        </logic:equal>
    </table>
</td>
</tr>
</table>
<html:messages id="msg" message="true">
    <bean:write name="msg"/><br>
</html:messages>
</html:form>
</body>
</html>

```

editar_setor.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>Setor</title>
<link href="styles/css_crescer.css" rel="stylesheet" type="text/css">
<script>
    function placeFocus() {
        if (document.forms.length > 0) {
            var field = document.forms[0];
            for (i = 0; i < field.length; i++) {
                if (!(field.elements[i].isDisabled)){
                    if ((field.elements[i].type == "text") ||
(field.elements[i].type == "textarea") || (field.elements[i].type.toString().charAt(0) == "s")) {

```

```

document.forms[0].elements[i].focus();
break;
}
}
}
}
}

function obrigatorios() {
    if ((document.getElementsByName('setor.cd_emp')[0].value == "" ||
(document.getElementsByName('setor.cd_setor')[0].value == "" ||
(document.getElementsByName('setor.ds_setor')[0].value == ""))){
        alert("Preencha Todos os Campos Obrigatórios");
        placeFocus();
        return false;
    }
}
</script>
</head>
<body bgcolor="#ffffff" onload="placeFocus">
<html:form action="/setor">
<input type="hidden" name="action">
<center><font class="TxtTitulo">Setor - Cadastros<br><br></font></center>
<table width="75%" border="1" bgcolor="#ffffff" bordercolor="#63659C" align="center">
<tr>
<td>
<table width="100%" border="0" bgcolor="#ffffff">
<tr>
<logic:notEqual name="setorForm" property="mostraConsulta" value="S">
<html:hidden name="setorForm" property="setor.oid"/>
<td colspan="5" align="center">
<logic:present name="setorForm" property="setor.oid">
<html:submit onclick="return obrigatorios()"
property="evento">
<bean:message
key="aprendizado_crm.alterar"/>
</html:submit>
</logic:present>
<logic:notPresent name="setorForm" property="setor.oid">
<html:submit onclick="return obrigatorios()"
property="evento">
<bean:message
key="aprendizado_crm.inserir"/>
</html:submit>
</logic:notPresent>
</td>
</tr>
<tr>
<td align="right" class="TxtTitulo">Empresa:
<td>
<logic:present name="setorForm" property="setor.oid">

```

```

                                <html:text          title="Campo          Desabilitado"
styleClass="campoCodigo"  maxLength="2"  style="text-align: right"  name="setorForm"  readonly="true"
property="setor.cd_emp" />
                                </logic:present>
                                <logic:notPresent name="setorForm" property="setor.oid">
styleClass="campoCodigo"  maxLength="2"  size="2"  style="text-align: right"  name="setorForm"
property="setor.cd_emp" />
                                <html:text          title="Digite o Código da Empresa"
                                </logic:notPresent>
                                </tr>
                                <tr>
                                <td align="right" class="TxtTitulo">Setor:
                                <td>
                                <logic:present name="setorForm" property="setor.oid">
styleClass="campoCodigo"  maxLength="2"  style="text-align: right"  name="setorForm"  readonly="true"
property="setor.cd_setor" />
                                <html:text          title="Campo          Desabilitado"
                                </logic:present>
                                <logic:notPresent name="setorForm" property="setor.oid">
styleClass="campoCodigo"  maxLength="2"  size="2"  style="text-align: right"  name="setorForm"
property="setor.cd_setor" />
                                <html:text          title="Digite o Código do Setor"
                                </logic:notPresent>
                                <html:text styleClass="endereco" title="Digite a Descrição do
Setor" maxLength="50" name="setorForm" property="setor.ds_setor" />
                                <html:hidden name="setorForm" property="setor_pesq.cd_emp"/>
                                <html:hidden name="setorForm" property="setor_pesq.cd_setor"/>
                                <html:hidden name="setorForm" property="setor_pesq.ds_setor"/>
                                </logic:notEqual>
                                <logic:equal name="setorForm" property="mostraConsulta" value="S">
                                <tr>
                                <td colspan="5" align="center">
                                <html:submit property="evento">
                                <bean:message key="aprendizado_crm.lista"/>
                                </html:submit>
                                </td>
                                </tr>
                                <tr>
                                <td align="right" class="TxtTitulo">Setor:
                                <td>
                                <html:text          title="Digite o Código do Setor"
styleClass="campoCodigo"  maxLength="2"  style="text-align: right"  name="setorForm"
property="setor_pesq.cd_setor" />
                                <html:text          title="Digite a Descrição do Setor"
styleClass="endereco"  maxLength="50"  name="setorForm"  property="setor_pesq.ds_setor" />
                                </logic:equal>
                                </table>
                                </td>
                                </tr>
                                </table>

```

```

<html:messages id="msg" message="true">
  <bean:write name="msg"/><br>
</html:messages>
</html:form>
</body>
</html>

```

editar_usuario.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>Usuário</title>
<link href="styles/css_crescer.css" rel="stylesheet" type="text/css">
<script>
    function placeFocus() {
        if (document.forms.length > 0) {
            var field = document.forms[0];
            for (i = 0; i < field.length; i++) {
                if (!(field.elements[i].isDisabled)){
                    if ((field.elements[i].type == "text") ||
(field.elements[i].type == "textarea") || (field.elements[i].type.toString().charAt(0) == "s")) {
                        document.forms[0].elements[i].focus();
                        break;
                    }
                }
            }
        }
    }

    function obrigatorios() {
        if ((document.getElementsByName('usuario.nm_usuario')[0].value == "") ||
(document.getElementsByName('usuario.ds_senha')[0].value == "")){
            alert("Preencha Todos os Campos Obrigatórios");
            placeFocus();
            return false;
        }
    }
</script>
</head>
<body bgcolor="#ffffff" onload="placeFocus">
<html:form action="/usuario">
<input type="hidden" name="action">
<center><font class="TxtTitulo">Usuário - Cadastros<br><br></font></center>
<table width="75%" border="1" bgcolor="#ffffff" bordercolor="#63659C" align="center">
<tr>
<td>
<table width="100%" border="0" bgcolor="#ffffff">
<tr>

```



```

        <logic:notEqual name="usuarioForm" property="mostraConsulta" value="S">
            <html:hidden name="usuarioForm" property="usuario.oid"/>
            <td colspan="5" align="center">
                <logic:present name="usuarioForm" property="usuario.oid">
                    <html:submit onclick="return obrigatorios()"
property="evento">
                        <bean:message
key="aprendizado_crm.alterar"/>
                            </html:submit>
                        </logic:present>
                    <logic:notPresent name="usuarioForm"
property="usuario.oid">
                        <html:submit onclick="return obrigatorios()"
property="evento">
                            <bean:message
key="aprendizado_crm.inserir"/>
                                </html:submit>
                            </logic:notPresent>
                        </td>
                    </tr>
                <tr>
                    <td align="right" class="TxtTitulo">Usuário:
                    <td>
                        <logic:present name="usuarioForm" property="usuario.oid">
                            <html:text title="Campo Desabilitado"
styleClass="campo" maxLength="30" name="usuarioForm" readonly="true" property="usuario.nm_usuario" />
                        </logic:present>
                        <logic:notPresent name="usuarioForm"
property="usuario.oid">
                            <html:text title="Digite o Nome do Usuário"
styleClass="campo" maxLength="30" name="usuarioForm" property="usuario.nm_usuario" />
                        </logic:notPresent>
                        <html:text styleClass="campo" title="Digite a Senha Usuário"
maxLength="8" name="usuarioForm" property="usuario.ds_senha" />
                    <html:hidden name="usuarioForm"
property="usuario_pesq.nm_usuario"/>
                    <html:hidden name="usuarioForm"
property="usuario_pesq.ds_senha"/>
                </logic:notEqual>

                <logic:equal name="usuarioForm" property="mostraConsulta" value="S">
                    <tr>
                        <td colspan="5" align="center">
                            <html:submit property="evento">
                                <bean:message key="aprendizado_crm.lista"/>
                            </html:submit>
                        </td>
                    </tr>
                <tr>
                    <td align="right" class="TxtTitulo">Usuário:
                    <td>

```

```

                <html:text title="Digite o Nome do Usuário"
styleClass="campo" maxLength="30" name="usuarioForm" property="usuario_pesq.nm_usuario" />
                <html:text title="Digite a Senha Usuário" styleClass="campo"
maxLength="8" name="usuarioForm" property="usuario_pesq.ds_senha" />
                </logic:equal>
            </table>
        </td>
    </tr>
</table>
<html:messages id="msg" message="true">
    <bean:write name="msg"/><br>
</html:messages>
</html:form>
</body>
</html>

```

lista_colaborador.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>Colaborador</title>
<script>
    function confirma(){
        if (confirm("Deseja apagar este registro?") == true) {
            return true;
        } else {
            return false;
        }
    }
</script>
<link href="styles/css_crescer.css" rel="stylesheet" type="text/css">
</head>
<body bgcolor="#ffffff">
<html:form action="/colaborador">
<input type="hidden" name="oid">
<input type="hidden" name="p_pagina">
<center><font class="TxtTitulo">Colaborador - Cadastros</font></center>
<jsp:include page="barra.jsp">
    <jsp:param name="p_nm_form" value="colaboradorForm" />
</jsp:include>
<table width="75%" border="1" bgcolor="#ffffff" bordercolor="#63659C" align="center">
<tr>
<td>
                <table width="100%" cellpadding="2" cellspacing="2" border="0"
bgcolor="ffffff" bordercolor="#63659C">
                    <logic:iterate id="listaBeans" name="colaboradorForm"
property="colaborador_lista">
                        <tr>

```



```

        <td class="TxtGeral"><bean:write name="listaBeans"
property="dt_admissao"/></td>
    </tr>
    <tr>
        <td align="right" class="TxtTitulo">Escolaridade:</td>
        <td class="TxtGeral"><bean:write name="listaBeans"
property="cd_escolaridade"/> <bean:write name="listaBeans" property="ds_escolaridade"/></td>
    </tr>
    <tr>
        <td align="right" class="TxtTitulo">Sexo:</td>
        <td class="TxtGeral">
            <logic:equal name="listaBeans" property="id_sexo"
value="F">Feminino</logic:equal>
            <logic:equal name="listaBeans" property="id_sexo"
value="M">Masculino</logic:equal>
        </td>
    </tr>
</logic:iterate>
</table>
</td>
</tr>
</table>
<html:hidden name="colaboradorForm" property="colaborador_pesq.cd_emp"/>
<html:hidden name="colaboradorForm" property="colaborador_pesq.cd_setor"/>
<html:hidden name="colaboradorForm" property="colaborador_pesq.cd_colaborador"/>
<html:hidden name="colaboradorForm" property="colaborador_pesq.nm_colaborador"/>
<html:hidden name="colaboradorForm" property="colaborador_pesq.id_sexo"/>
<html:hidden name="colaboradorForm" property="colaborador_pesq.dt_admissao"/>
<html:hidden name="colaboradorForm" property="colaborador_pesq.cd_escolaridade"/>
<jsp:include page="contadores.jsp">
    <jsp:param name="p_nm_form" value="colaboradorForm" />
</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) {

```



```

        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);
    %>
border="0">
        <html:link page="/empresa.do" name="parametros" >
            

        </logic:iterate>
    </table>

</td>
</tr>
</table>
<html:hidden name="empresaForm" property="empresa_pesq.cd_emp"/>
<html:hidden name="empresaForm" property="empresa_pesq.nm_emp"/>
<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_escolaridade.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>Escolaridade</title>
<script>
    function confirma(){
        if (confirm("Deseja apagar este registro?") == true) {
            return true;
        } else {
            return false;
        }
    }
</script>
    <link href="styles/css_crescer.css" rel="stylesheet" type="text/css">
</head>
<body bgcolor="#ffffff">
<html:form action="/escolaridade">
<input type="hidden" name="oid">
<input type="hidden" name="p_pagina">
<center><font class="TxtTitulo">Escolaridade - Cadastros</font></center>

```



```

        </td>
    </tr>
</table>
<html:hidden name="escolaridadeForm" property="escolaridade_pesq.cd_escolaridade"/>
<html:hidden name="escolaridadeForm" property="escolaridade_pesq.ds_escolaridade"/>
<jsp:include page="contadores.jsp">
    <jsp:param name="p_nm_form" value="escolaridadeForm" />
</jsp:include>
<html:messages id="msg" message="true">
    <bean:write name="msg"/><br>
</html:messages>
</html:form>
</body>
</html>

```

lista_estadistica_acesso.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>Estatística de Acesso</title>
<script>
    function confirma(){
        if (confirm("Deseja apagar este registro?") == true) {
            return true;
        } else {
            return false;
        }
    }
</script>
<link href="styles/css_crescer.css" rel="stylesheet" type="text/css">
</head>
<body bgcolor="#ffffff">
<html:form action="/estatistica_acesso">
<input type="hidden" name="oid">
<input type="hidden" name="p_pagina">
<center><font class="TxtTitulo">Estatística de Acesso</font></center>
<jsp:include page="barra_estadistica.jsp">
    <jsp:param name="p_nm_form" value="estatistica_acessoForm" />
</jsp:include>
<table width="75%" border="1" bgcolor="#ffffff" bordercolor="#63659C" align="center">
    <tr>
        <td>
            <table width="100%" border="0" bgcolor="ffffff" bordercolor="#63659C">
                <tr>
                    <td align="left" class="TxtTitulo">Usuário</td>
                    <td align="center" class="TxtTitulo">Data de Acesso</td>
                </tr>
            </table>

```



```

        <logic:iterate          id="listaBeans"          name="estatistica_acessoForm"
property="estatistica_acesso_lista">
        <tr>
        <td align="left" class="TxtGeral"><bean:write name="listaBeans"
property="nm_usuario"/>
        <td align="center" class="TxtGeral"><bean:write name="listaBeans"
property="dt_acesso"/>
        <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="/estatistica_acesso.do"
name="parametros" onclick="return confirma()">
                
                </html:link>
        </logic:iterate>
        </table>
    </td>
</tr>
</table>
<html:hidden name="estatistica_acessoForm" property="estatistica_acesso_pesq.nm_usuario"/>
<html:hidden name="estatistica_acessoForm" property="estatistica_acesso_pesq.dt_acesso"/>
<jsp:include page="contadores.jsp">
    <jsp:param name="p_nm_form" value="estatistica_acessoForm" />
</jsp:include>
<html:messages id="msg" message="true">
    <bean:write name="msg"/><br>
</html:messages>
</html:form>
</body>
</html>

```

lista_pergunta.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>Pergunta</title>
<script>
    function confirma(){
        if (confirm("Deseja apagar este registro?") == true) {

```

```

        return true;
    } else {
        return false;
    }
}
</script>
<link href="styles/css_crescer.css" rel="stylesheet" type="text/css">
</head>
<body bgcolor="#ffffff">
<html:form action="/pergunta_a">
<input type="hidden" name="oid">
<input type="hidden" name="p_pagina">
<center><font class="TxtTitulo">Pergunta - Cadastros</font></center>
<jsp:include page="barra.jsp">
    <jsp:param name="p_nm_form" value="perguntaForm" />
</jsp:include>
<table width="85%" border="1" bgcolor="#ffffff" bordercolor="#63659C" align="center">
    <tr>
        <td>
            <table width="100%" border="0" bgcolor="ffffff" bordercolor="#63659C">
                <tr>
                    <td align="right" class="TxtTitulo">Código</td>
                    <td align="left" class="TxtTitulo">Descrição</td>
                </tr>
                <logic:iterate          id="listaBeans"          name="perguntaForm"
property="pergunta_lista">
                <tr>
                    <td align="right" class="TxtGeral"><bean:write name="listaBeans"
property="cd_pergunta"/>
                    <td align="left" class="TxtGeral"><bean:write name="listaBeans"
property="ds_pergunta"/>
                    <td width="15%" align="right">
                        <%java.util.HashMap params;%>
                    <td>
                        <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="/pergunta_a.do" name="parametros"
                                >
                                    
                                </html:link>
                                </td>
                </tr>
                </logic:iterate>
            </table>
        </td>
    </tr>
</table>

```

```

    </tr>
</table>
<html:hidden name="perguntaForm" property="pergunta_pesq.cd_pergunta"/>
<html:hidden name="perguntaForm" property="pergunta_pesq.ds_pergunta"/>
<jsp:include page="contadores.jsp">
    <jsp:param name="p_nm_form" value="perguntaForm" />
</jsp:include>
<html:messages id="msg" message="true">
    <bean:write name="msg"/><br>
</html:messages>
</html:form>
</body>
</html>

```

lista_resposta.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>
<link href="styles/css_crescer.css" rel="stylesheet" type="text/css">
<title>Perguntas</title>
<script>
    function submete_resposta(n_pagina, n_resposta_atual){
        respostaForm.nr_pergunta_atual.value = n_pagina;
        respostaForm.nr_resposta_atual.value = n_resposta_atual;
        respostaForm.submit();
    }
</script>
</head>
<body bgcolor="#ffffff">
<html:form action="/resposta">
<input type="hidden" name="oid">
<input type="hidden" name="p_pagina">
<input type="hidden" name="nr_resposta_atual">
<input type="hidden" name="nr_pergunta_atual">
<table width="80%" border="0" bgcolor="#ffffff" bordercolor="#ffffff" align="center">
    <tr>
        <td>
            <table width="100%" border="0" bgcolor="fffff" bordercolor="#000000">
                <tr>
                    <td class="TxtTitulo" colspan="3" align="left"><bean:write
name="respostaForm" property="resposta.cd_pergunta"/>. <bean:write
property="resposta.ds_pergunta"/><br><br></td>
                    <td align="right" colspan="3"><bean:write name="respostaForm"
property="resposta_lista">
                        <logic:iterate id="listaBeans" name="respostaForm"

```

```

                <td align="right"><input type="radio" name="retorno"
OnClick="javascript:submete_resposta('<bean:write name="respostaForm" property="nr_pergunta_atual"/>',
'<bean:write name="listaBeans" property="cd_resposta"/>')"></td>
                <td class="TxtGeral" align="left"><bean:write
name="listaBeans" property="ds_resposta"/></td>
            </tr>
        </logic:iterate>
    </table>
</td>
</tr>
</table>
<html:hidden name="respostaForm" property="resposta.cd_pergunta"/>
<html:hidden name="respostaForm" property="resposta_pesq.cd_pergunta"/>
<html:hidden name="respostaForm" property="resposta_pesq.ds_pergunta"/>
<html:hidden name="respostaForm" property="resposta_pesq.cd_resposta"/>
<html:hidden name="respostaForm" property="resposta_pesq.ds_resposta"/>
<html:messages id="msg" message="true">
    <bean:write name="msg"/><br>
</html:messages>
</html:form>
</body>
</html>

```

lista_resposta_a.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>Resposta</title>
<script>
    function confirma(){
        if (confirm("Deseja apagar este registro?") == true) {
            return true;
        } else {
            return false;
        }
    }
</script>
    <link href="styles/css_crescer.css" rel="stylesheet" type="text/css">
</head>
<body bgcolor="#ffffff">
<html:form action="/resposta_a">
<input type="hidden" name="oid">
<input type="hidden" name="p_pagina">
<center><font class="TxtTitulo">Resposta - Cadastros</font></center>
<jsp:include page="barra.jsp">
    <jsp:param name="p_nm_form" value="respostaForm" />
</jsp:include>
<table width="85%" border="1" bgcolor="#ffffff" bordercolor="#63659C" align="center">

```

```

<tr>
  <td>
    <table width="100%" border="0" bgcolor="ffffff" bordercolor="#63659C">
      <tr>
        <td align="right" class="TxtTitulo">Pergunta</td>
        <td align="right" class="TxtTitulo">Código</td>
        <td align="left" class="TxtTitulo">Descrição</td>
        <td align="left" class="TxtTitulo">Certa/Errada</td>
      </tr>
      <logic:iterate          id="listaBeans"          name="respostaForm"
property="resposta_lista">
        <tr>
          <td align="right" class="TxtGeral"><bean:write name="listaBeans"
property="cd_pergunta"/>
          <td align="right" class="TxtGeral"><bean:write name="listaBeans"
property="cd_resposta"/>
          <td align="left" class="TxtGeral"><bean:write name="listaBeans"
property="ds_resposta"/>
          <td class="TxtGeral" align="left" class="TxtGeral">
            <logic:equal      name="listaBeans"      property="id_status"
value="C">Correta</logic:equal>
            <logic:equal      name="listaBeans"      property="id_status"
value="E">Incorreta</logic:equal>
          </td>
          <td width="15%" align="right">
            <%java.util.HashMap params;%>
            <td>
              <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="/resposta_a.do" name="parametros"
border="0">
                
              </html:link>
            </td>
          </td>
        </tr>
      </logic:iterate>
    </table>
  </td>
</tr>
</table>
<html:hidden name="respostaForm" property="resposta_pesq.cd_resposta"/>
<html:hidden name="respostaForm" property="resposta_pesq.cd_pergunta"/>
<html:hidden name="respostaForm" property="resposta_pesq.ds_resposta"/>

```

```

<html:hidden name="respostaForm" property="resposta_pesq.id_status"/>
<jsp:include page="contadores.jsp">
  <jsp:param name="p_nm_form" value="respostaForm" />
</jsp:include>
<html:messages id="msg" message="true">
  <bean:write name="msg"/><br>
</html:messages>
</html:form>
</body>
</html>

```

lista_resultado.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>
<link href="styles/css_crescer.css" rel="stylesheet" type="text/css">
<title>Resultado</title>
</head>

<body bgcolor="#ffffff">
<html:form action="/resultado">
<input type="hidden" name="oid">
<input type="hidden" name="p_pagina">
<table width="100%" border="0" bgcolor="#ffffff" bordercolor="#ffffff" align="center">
  <tr>
    <td>
      <table width="100%" border="0" bgcolor="ffffff" bordercolor="#000000">
        <tr>
          <td class="TxtGeral">Olá <font class="TxtTitulo"><%=
request.getSession().getAttribute("usuario")%></font>, seu Desempenho foi o seguinte:<br><br></td>
        </tr>
        <tr>
          <td class="TxtTitulo" align="left">Pergunta</td>
          <td class="TxtTitulo" align="left">Resposta</td>
          <td class="TxtTitulo" align="left">Certa/Errada</td>
        </tr>
        <tr>
          <td colspan="3"><logic:iterate id="listaBeans" name="resultadoForm"
property="resultado_lista">
          <tr>
            <td class="TxtGeral" align="left"><bean:write name="listaBeans"
property="ds_pergunta"/></td>
            <td class="TxtGeral" align="left"><bean:write name="listaBeans"
property="ds_resposta"/></td>
            <td class="TxtGeral" align="left" class="TxtGeral">
              <logic:equal name="listaBeans" property="id_status"
value="C">Correta</logic:equal>
              <logic:equal name="listaBeans" property="id_status"
value="E">Incorreta</logic:equal>

```

```

        </td>
      </tr>
    </logic:iterate>
  <tr>
    <td class="TxtGeral"><BR><BR>Para recomençar, clique <a
href="resposta.do">AQUI</a></td>
  </tr>
  <tr>
    <td class="TxtGeral"><BR><BR>Para verificar as respostas
corretas, clique <a href="resultado_a.do">AQUI</a></td>
  </tr>
</table>
</td>
</tr>
</table>
<html:messages id="msg" message="true">
  <bean:write name="msg"/><br>
</html:messages>
</html:form>
</body>
</html>

```

lista_resultado_a.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>
<link href="styles/css_crescer.css" rel="stylesheet" type="text/css">
<title>Resultado</title>
</head>
<body bgcolor="#ffffff">
<html:form action="/resultado_a">
<input type="hidden" name="oid">
<input type="hidden" name="p_pagina">
<table width="100%" border="0" bgcolor="#ffffff" bordercolor="#ffffff" align="center">
  <tr>
    <td>
      <table width="100%" border="0" bgcolor="ffffff" bordercolor="#000000">
        <tr>
          <td class="TxtGeral">As respostas corretas são as
seguintes:</font><br><br></td>
        </tr>
        <tr>
          <td class="TxtTitulo" align="left">Pergunta</td>
          <td class="TxtTitulo" align="left">Resposta</td>
        </tr>
      </table>
    </td>
  </tr>
</table>
<logic:iterate id="listaBeans" name="resultadoForm"
property="resultado_lista">
  <tr>

```

```

        <td class="TxtGeral" align="left"><bean:write name="listaBeans"
property="ds_pergunta"/></td>
        <td class="TxtGeral" align="left"><bean:write name="listaBeans"
property="ds_resposta"/></td>
    </tr>
</logic:iterate>
    <tr>
        <td class="TxtGeral"><BR><BR>Para voltar à tela inicial,
clique <a href="aprender.jsp">AQUI</a></td>
    </tr>
</table>
</td>
</tr>
</table>
<html:messages id="msg" message="true">
    <bean:write name="msg"/><br>
</html:messages>
</html:form>
</body>
</html>

```

lista_resultado_b.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>Resultado</title>
<script>
    function confirma(){
        if (confirm("Deseja apagar este registro?") == true) {
            return true;
        } else {
            return false;
        }
    }
</script>
    <link href="styles/css_crescer.css" rel="stylesheet" type="text/css">
</head>
<body bgcolor="#ffffff">
<html:form action="/resultado_b">
<input type="hidden" name="oid">
<input type="hidden" name="p_pagina">
<center><font class="TxtTitulo">Resultado</font></center>
<jsp:include page="barra_resultado.jsp">
    <jsp:param name="p_nm_form" value="resultadoForm" />
</jsp:include>
<table width="300" border="1" bgcolor="#ffffff" bordercolor="#63659C" align="center">
    <tr>

```



```

        <td>
            <table width="100%" border="0" bgcolor="ffffff" bordercolor="#63659C">
                <tr>
                    <td align="left" class="TxtTitulo">Usuário</td>
                    <td align="right" class="TxtTitulo">Pergunta</td>
                    <td align="left" class="TxtTitulo">Status</td>
                </tr>
                <logic:iterate          id="listaBeans"          name="resultadoForm"
property="resultado_lista">
                <tr>
                    <td align="left" class="TxtGeral"><bean:write name="listaBeans"
property="nm_usuario"/></td>
                    <td align="right" class="TxtGeral"><bean:write name="listaBeans"
property="cd_pergunta"/></td>
                    <td class="TxtGeral" align="left" class="TxtGeral">
                        <logic:equal name="listaBeans" property="id_status"
value="C">Correta</logic:equal>
                        <logic:equal name="listaBeans" property="id_status"
value="E">Incorreta</logic:equal>
                    </td>
                </tr>
                <%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="/escolaridade.do"
name="parametros" onclick="return confirma()">
                    
                    </html:link>
                </td>
            </tr>
        </table>
    </td>
</tr>
</table>
<html:hidden name="resultadoForm" property="resultado_pesq.nm_usuario"/>
<html:hidden name="resultadoForm" property="resultado_pesq.cd_pergunta"/>
<html:hidden name="resultadoForm" property="resultado_pesq.cd_resposta"/>
<html:hidden name="resultadoForm" property="resultado_pesq.id_status"/>
<jsp:include page="contadores.jsp">
    <jsp:param name="p_nm_form" value="resultadoForm" />
</jsp:include>
<html:messages id="msg" message="true">
    <bean:write name="msg"/><br>
</html:messages>

```

```
</html:form>
</body>
</html>
```

lista_setor.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>Setor</title>
<script>
    function confirma(){
        if (confirm("Deseja apagar este registro?") == true) {
            return true;
        } else {
            return false;
        }
    }
</script>
    <link href="styles/css_crescer.css" rel="stylesheet" type="text/css">
</head>
<body bgcolor="#ffffff">
<html:form action="/setor">
<input type="hidden" name="oid">
<input type="hidden" name="p_pagina">
<center><font class="TxtTitulo">Setor - Cadastros</font></center>
<jsp:include page="barra.jsp">
    <jsp:param name="p_nm_form" value="setorForm" />
</jsp:include>
<table width="75%" border="1" bgcolor="#ffffff" bordercolor="#63659C" align="center">
    <tr>
        <td>
            <table width="100%" border="0" bgcolor="ffffff" bordercolor="#63659C">
                <tr>
                    <td align="right" class="TxtTitulo">Empresa</td>
                    <td align="left" class="TxtTitulo">Descrição</td>
                </tr>
                <tr>
                    <td align="right" class="TxtTitulo">Setor</td>
                    <td align="left" class="TxtTitulo">Descrição</td>
                </tr>
                <tr>
                    <td align="right" class="TxtGeral"><bean:write name="listaBeans"
property="cd_emp"/>
                    <td align="left" class="TxtGeral"><bean:write name="listaBeans"
property="nm_emp"/>
                    <td align="right" class="TxtGeral"><bean:write name="listaBeans"
property="cd_setor"/>
```



```

<% @ 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>
    <link href="styles/css_crescer.css" rel="stylesheet" type="text/css">
</head>
<body bgcolor="#ffffff">
<html:form action="/usuario">
<input type="hidden" name="oid">
<input type="hidden" name="p_pagina">
<center><font class="TxtTitulo">Usuário - Cadastros</font></center>
<jsp:include page="barra.jsp">
    <jsp:param name="p_nm_form" value="usuarioForm" />
</jsp:include>
<table width="75%" border="1" bgcolor="#ffffff" bordercolor="#63659C" align="center">
<tr>
<td>
        <table width="100%" border="0" bgcolor="ffffff" bordercolor="#63659C">
<tr>
<td align="left" class="TxtTitulo">Usuário</td>
<td align="left" class="TxtTitulo">Senha</td>
</tr>
<logic:iterate id="listaBeans" name="usuarioForm" property="usuario_lista">
<tr>
<td align="left" class="TxtGeral"><bean:write name="listaBeans"
property="nm_usuario"/>
<td align="left" class="TxtGeral"><bean:write name="listaBeans"
property="ds_senha"/>
<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);
                %>

```



```
width="600">
    <table height="118" border="0" cellpadding="0" cellspacing="0"
    <tr>
        <td></td>
        <td></td>
        <td></td>
    </tr>
    <tr>
        <td></td>
        <td></td>
        <td></td>
    </tr>
</table>
</td>
</tr>
<tr valign="top">
    <td align="center" width="100%">
        <IFRAME width="600" height="400" align="center" frame
name="principal" src="aprender.jsp" scrolling="auto" frameborder="no"> </IFRAME>
    </td>
</tr>
</table>
</body>
</html>
```