FUNDAÇÃO DE ENSINO "EURÍPIDES SOARES DA ROCHA"
CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA - UNIVEM
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

BRUNO FELIPE CERQUEIRA SILVA
HELTON HIDEHARU HIGUTE

BRD TOOL: FERRAMENTA PARA GERAÇÃO DE APLICAÇÃO BASEADA EM REGRAS DE NEGÓCIO.

MARÍLIA
2005

BRUNO FELIPE CERQUEIRA SILVA
HELTON HIDEHARU HIGUTE

BRD TOOL: FERRAMENTA PARA GERAÇÃO DE APLICAÇÃO BASEADA EM REGRAS DE NEGÓCIO.

Monografia apresentada na Fundação de Ensino Eurípides Soares da Rocha, Mantenedora do Centro Universitário Eurípides de Marília-UNIVEM, como trabalho de conclusão de curso.
Orientador: Prof. Dr. Edmundo Sérgio Spoto.

MARÍLIA
2005

HIGUTE, Helton Hideharu; SILVA, Bruno Felipe Cerqueira.
BRD Tool: ferramenta para geração de aplicação baseada em regras de negócio / Helton Hideharu Higute / Bruno Felipe Cerqueira Silva; orientador: Prof. Dr. Edmundo Sérgio Spoto. Marília, SP [s.n.], 2005.

144 f.

Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Centro Universitário Eurípedes de Marília, Fundação de Ensino Eurípedes Soares da Rocha.
1. Engenharia de Software
CDD: 005.1.

Bruno Felipe Cerqueira Silva

BRD TOOL: FERRAMENTA PARA GERAÇÃO DE APLICAÇÃO BASEADA EM REGRAS DE NEGÓCIO

Banca examinadora da monografia apresentada ao Curso de Bacharelado em Ciência da Computação do UNIVEM/F.E.E.S.R., para obtenção do Título de Bacharel em Ciência da Computação.

Nota: _____(_____)

Orientador: Edmundo Sérgio Spoto _____

1º Examinador: Ildeberto Aparecido Rodello _____

2º Examinador: Maria Istela Cagnin _____

Marília, 23 de novembro de 2005.

Helton Hideharu Higute

BRD TOOL: FERRAMENTA PARA GERAÇÃO DE APLICAÇÃO BASEADA EM REGRAS DE NEGÓCIO

Banca examinadora da monografia apresentada ao Curso de Bacharelado em Ciência da Computação do UNIVEM/F.E.E.S.R., para obtenção do Título de Bacharel em Ciência da Computação.

Nota: _____(_____)

Orientador: Edmundo Sérgio Spoto _____

1º Examinador: Ildeberto Aparecido Rodello _____

2º Examinador: Maria Istela Cagnin _____

Marília, 23 de novembro de 2005.

AGRADECIMENTOS

Em primeio lugar, agradecemos a Deus, por ter nos ajudado a chegar até aqui, pois sem ele, nada disso seria possível; em segundo lugar, agradecemos nosso orientador Prof. Dr. Edmundo Sérgio Spoto.

Agradecemos nossos pais e amigos, por todo o incentivo que nos dão, por acreditarem e confiarem em nós.

Agradecemos, também a todos que colaboraram para que este trabalho fosse concluído, ao graduando e colega de turma Gregory Petterson Zanelato, pela ajuda para a realização desse projeto, e a todos os nossos amigos, colegas de turma, especialmente Carlos Eduardo Sanvido, Aparecido Nardo Junior e Cristiano Takano, que nos acompanharam em todos os momentos, durante esses quatro anos de faculdade.

HIGUTE, Helton Hideharu; SILVA, Bruno Felipe Cerqueira. **BRD Tool: ferramenta para geração de aplicação baseada em regras de negócio**. 2005. Trabalho de Conclusão de Curso – Centro Universitário Eurípedes de Marília, Fundação de Ensino Eurípedes Soares da Rocha, 2005.

RESUMO

O Desenvolvimento de Software é uma área que investe em qualidade e segurança na obtenção do produto a ser construído. Neste projeto é apresentada uma proposta de uma ferramenta que apóia no desenvolvimento de um software baseado em regras de negócio de um banco de dados existente. São apresentadas as etapas do projeto de construção da ferramenta bem como da formação do banco de dados que controla todas as informações geradas na ferramenta. A ferramenta desenvolvida tem a finalidade de facilitar o desenvolvimento de aplicações para seus usuários, otimizando tempo e qualidade, sendo que a mesma é gerada a partir de eventuais regras do negócio necessárias para a empresa.

HIGUTE, Helton Hideharu; SILVA, Bruno Felipe Cerqueira. **BRD Tool: ferramenta para geração de aplicação baseada em regras de negócio**. 2005. Trabalho de Conclusão de Curso – Centro Universitário Eurípedes de Marília, Fundação de Ensino Eurípedes Soares da Rocha, 2005.

ABSTRACT

The Software development is an area which focus the quality and safety about the matter on obtaining the product to be built. In this project is presented a tool proposal that supports the software development based on the business rules of an existent database. The project stages of building this tool are presented as well as the database creation which controls all the information generated in the tool. This developed tool has the purpose of facilitating the applications development for any eventually rules from any business needs, accordingly to the company.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

ASP – *Active Server Pages*

CSS – *Cascade Style Sheet*

BRD Tool – *Business Rules Description Tool*

DER – Diagrama Entidade Relacionamento

EJB – *Enterprise JavaBeans*

HTML – *HyperText Markup Language*

HTTP – *HyperText Transfer Protocol*

J2EE – *Java to Enterprise Edition*

JSP – *Java Server Pages*

JDBC – *Java DataBase Connectivity*

OBCD – *Open DataBase Connectivity*

PHP – *Hypertext PreProcessor*

SGBD – Sistema Gerenciador de Banco de Dados

SGBDR – Sistema Gerenciador de Banco de Dados Relacional

SQL – *Structured Query Language*

UML – *Unified Modeling Language*

XML – *Extensible Markup Language*

XHTML – *Extensible HyperText Markup Language*

INTRODUÇÃO

Atualmente há necessidade de se obter sistemas que agilizem e facilitem as tarefas de desenvolvimento de sistemas, e, se possível, encontrar uma forma em que não apenas programadores nas mais diversas linguagens possam desenvolver, mas sim pessoas comuns que necessitam de algum aplicativo que facilite sua vida, como gerentes de empresas, administradores, chefes de departamento etc.

Existem no mercado diversas ferramentas que facilitam o desenvolvimento de aplicativos nas mais diversas linguagens de programação, que disponibilizam facilmente formas de construções de formulários, banco de dados e alguns tipos de validações, porém, estas ainda exigem o conhecimento de algumas tecnologias por parte do usuário, como por exemplo, noções de relacionamento entre tabelas do banco de dados e conhecimentos técnicos para construir as validações necessárias.

A finalidade deste trabalho é construir uma ferramenta capaz de gerar aplicações *Web* dinamicamente, através da construção de processos de negócio, onde os usuários consigam informar os parâmetros e regras de negócio necessários inerentes ao problema a ser resolvido.

Para que seja útil, é necessário que a usabilidade da ferramenta seja voltada para o tipo de usuário para a qual é destinada. Sendo destinada a gerentes de negócio, chefes de departamento e outros colaboradores de uma organização, há a necessidade de a *interface* e os processos de construção de um processo de negócio utilizarem a mesma linguagem que esses administradores.

A agilidade no desenvolvimento de uma aplicação por parte do usuário, é atingida conseqüentemente pela facilidade na construção da aplicação desejada, não existindo a necessidade de conhecimentos técnicos por parte dos utilizadores, e também de mão de obra especializada. Os próprios gerentes de negócio, chefes de departamento e outros utilizadores poderão construir suas aplicações sem muita dificuldade e dependência de pessoal.

Motivação

Muitas dificuldades são encontradas quando existe a necessidade de se desenvolver um sistema para uma determinada empresa, dificuldade esta encontrada primeiramente quando um analista de sistemas tem o dever de entrevistar o administrador do negócio em questão. Na maioria dos casos, o administrador do negócio não consegue passar todo o conhecimento para o analista, que por sua vez, nem sempre entende totalmente o que o administrador deseja.

Por conta do processo de desenvolvimento de *software* utilizado atualmente, que requisita um *software* ou necessita da adição de alguma funcionalidade ao existente, existe uma dependência muito grande dos desenvolvedores envolvidos no projeto.

Este trabalho utilizou tais dificuldades para desenvolver uma solução: em vez do programador desenvolver o sistema, o administrador do negócio consegue desenvolver a aplicação desejada com base em seus conhecimentos das regras de negócio para qual trabalha.

Objetivo

O objetivo deste trabalho é obter uma ferramenta que consiga gerar aplicações *Web* com base em casos de uso de projeto desenvolvidos pelo conhecedor do negócio. Ferramenta esta que deve funcionar em ambiente *Web* possibilitando o acesso de qualquer parte.

Para a obtenção da mesma é necessário modelar um repositório para as informações necessárias e construir métodos para que possamos fazer a extração dos metadados do banco de dados do cliente.

Organização deste Trabalho

No Capítulo de introdução foi abordado o objetivo e a motivação para estudar e desenvolver a ferramenta, tendo em vista que na pesquisa foram levantados os conhecimentos necessários para o desenvolvimento da ferramenta.

No Capítulo 1 apresentam-se as definições e terminologias necessárias para o entendimento do trabalho e no Capítulo 2 as técnologias utilizadas.

No Capítulo 3 é abordado o funcionamento da ferramenta no nível de usuário e no nível de processamento, abrangendo-se os principais termos utilizados neste trabalho como DER (Diagrama Entidade Relacionamento) do banco de dados da ferramenta construída.

No último Capítulo encontram-se as conclusões baseadas em todo o processo de desenvolvimento da ferramenta e trabalhos futuros.

CAPÍTULO 1 - DEFINIÇÕES E TERMINOLOGIAS

Neste capítulo são apresentadas as terminologias e as definições que são a base da construção da Ferramenta BRD Tool (*Business Rules Description Tool*).

1.1. Casos de Uso

Caso de Uso é utilizado para a descrição do comportamento do *software* quando o mesmo estiver finalizado, ou seja, os casos de uso não descrevem como o *software* é construído, e sim como deve se comportar. Os casos de uso foram inicialmente propostos em uma metodologia de desenvolvimento de sistemas orientado a objetos, e posteriormente foram incorporados a UML (*Unified Modeling Language*), visto que na UML são representados por diagramas, como por exemplo, diagrama de classe, diagrama de caso de uso, diagrama de seqüência, entre outros (LARMAN, 2004).

1.2. Regras de Negócio

São definidas como regras de negócio os aspectos que controlam e definem os processos de negócio. Não são diferentes dos requisitos funcionais exigidos para um determinado negócio, diferindo apenas na importância estratégica que apresentam sobre o processo (DALLAVALLE, 2000; CAZARINI, 2000).

Para uma apresentação palpável, pode-se dizer que as regras de negócio definem como o cliente deve ser atendido pela empresa, como a empresa deve decidir uma compra, qual a regra de construção do preço final de um produto etc.

1.3. Banco de Dados

De acordo com Silberschatz, Korth e Sudarshan (1999), podemos definir banco de dados como um repositório de dados, ou seja, onde os dados relevantes de uma determinada aplicação são armazenados. Os mesmos podem ser manipulados por várias operações, de forma a poder adicionar novos arquivos ao banco de dados (*createdb*), inserir (*insert*), atualizar (*update*) e eliminar (*delete*) registros dos arquivos existentes e recuperar (*select*) dados das tabelas existentes.

1.3.1. Metadados

Os metadados geralmente são armazenados em tabelas ou arquivos do próprio Sistema Gerenciador de Banco de Dados (SGBD), sendo que estes, segundo Date (1991), nada mais são que “dados sobre dados”, ou seja, são dados relevantes do banco de dados que descrevem as tabelas, campos, tipos entre outros, sendo assim, também pode ser chamado de dicionário de dados.

1.3.2. Entidades

Entidade é um objeto (tabela) abstrato ou concreto, que é composta por um conjunto de atributos que descreve o mesmo. Ela pode ser classificada como forte ou fraca, ou seja, uma entidade forte não necessita de outra para a sua existência, sendo assim independente, já a entidade fraca é dependente de outra para que haja a sua existência. Essa classificação ocorre quando é dada a cardinalidade do relacionamento (ELMASRI e NAVATHE 2005).

1.3.3. Diagrama Entidade Relacionamento

Diagrama Entidade Relacionamento (DER) nada mais é que a representação gráfica da estrutura lógica do banco de dados, sendo assim, facilita o entendimento do banco de dados a ser analisado. (SILBERSCHATZ, KORTH, SUDARSHAN 1999).

No diagrama pode-se encontrar quatro tipos de relacionamento, “muitos para muitos”, “um para muitos”, “muitos para um” e “um para um”, que são representados por uma cardinalidade nos relacionamentos entre as tabelas.

Uma das formas pela qual o DER pode ser representado é através de formas geométricas, como retângulo, elipses, losangos e linhas , onde cada forma geométrica possui um significado, sendo que o retângulo representa a entidade (tabela), a elipse o atributo (campo), o losango o relacionamento e a linha representa a conexão dos atributos com sua respectiva entidade e a conexão entre as entidades.

1.4. Considerações Finais

As definições e terminolgias acima mencionadas são necessárias para facilitar o entendimento do leitor na construção da ferramenta, além disso o leitor também pode utilizar algumas definições para melhorar o entendimento da proposta da ferramenta.

CAPÍTULO 2 - TECNOLOGIAS UTILIZADAS

Neste Capítulo são abordadas as tecnologias utilizadas para a construção da ferramenta BRD Tool, ou seja, como a ferramenta foi desenvolvida para *Web*, é explicada as características de desenvolvimento *Web*, a plataforma J2EE e a API *Java DataBase Connectivity* (JDBC).

2.1. Desenvolvimento Web

A Internet é uma rede de computadores interligados em toda a extensão do globo terrestre que prove diversos tipos de serviços a seus usuários. Um destes serviços é a disponibilização de páginas *Web* sendo que é necessário um servidor que contenha alguns aplicativos instalados, tais como, aplicação para servir páginas, não obrigatoriamente uma linguagem de programação que seja executada no servidor como PHP, JSP, ASP, entre outros.

Os servidores *Web* trabalham recebendo e enviando requisições e respostas no formato do protocolo *Hypertext Transfer Protocol* (HTTP). HTTP é um protocolo do nível de aplicação, que possui objetividade e rapidez necessárias para suportar sistemas de informação distribuídos cooperativos de hipermídia (FIELDING et al., 1999).

O protocolo HTTP é do tipo *stateless*, ou seja, não armazena o estado (dados sobre a conexão). Com isto, assim que um usuário efetua uma requisição, é criada uma conexão com

o servidor, a requisição é respondida e a conexão é desfeita, este mesmo usuário, se efetuar outra conexão, o protocolo já o desconhece.

Para o desenvolvimento de *websites* ou aplicativos para Internet é necessária a aplicação de diversos padrões e linguagens, a maioria destes são definidos e gerenciados pela W3C *Consortium* também conhecida como World Wide Web Consortium, que é um consórcio internacional de organizações que trabalham para desenvolver, gerenciar e assegurar o crescimento ao longo do tempo da *Web*.

As linguagens e padrões mais utilizados para tal desenvolvimento são o HTML ou XHTML, CSS, *JavaScript* e, para adicionar alguns recursos mais avançados e adicionar dinamismo às páginas, são utilizadas linguagens de programação que rodem do lado servidor da aplicação *Web*, tais como PHP, JSP, ASP, *ColdFusion*, entre outras.

O HTML e o XHTML, uma extensão do HTML, são linguagens utilizadas para definir páginas *Web*, compostas por *tags* onde o desenvolvedor deve seguir as regras das mesmas para que depois de desenvolvidas os navegadores possam recuperá-las e mostrá-las para os usuários.

Os CSS (*Cascade Style Sheet*) são chamados de folhas de estilo, são formas simples de adicionar estilos para a página desenvolvida, como cores, fontes diferentes, posicionamentos etc. (BOS et al., 1998)

JavaScript é uma linguagem de programação desenvolvida pela Netscape que roda do lado cliente da página visualizada. Com ela é possível disponibilizar algum tipo de dinamismo nas páginas mostradas para o cliente, como validações de campos, mensagens de alerta, entre outros. O único problema é que a utilização de *JavaScript* depende muito das implementações de *browser's* existentes.

Existem diversas linguagens para a programação do lado servidor de uma página *Web*, uma delas é a linguagem *Java Server Pages* (JSP). JSP é uma linguagem que roda do

lado servidor de uma aplicação *Web.* Para utilização de JSP é necessário ter um servidor e um *container* JSP para que as páginas possam ser disponibilizadas.

2.2. J2EE

A plataforma J2EE fornece uma metodologia baseada em componentes para o projeto, desenvolvimento, montagem e implantação dos aplicativos empresariais. Ela oferece um modelo de aplicativo distribuído em multicamadas, ou seja, existem três localizações, sendo que se pode encontrar na máquina do cliente, na máquina do servidor J2EE e na máquina do servidor de banco de dados. J2EE também oferece componentes reutilizáveis, um modelo de segurança unificado, controle flexível de transações e suporte de serviços *Web*.

Na máquina do cliente são encontrados componentes como clientes de aplicativo e *applets*, sendo o primeiro fornece um modo para os usuários manipularem as tarefas que requerem uma *interface* de uuário mais rica. Já nos *applets*, eles executam pequenos aplicativos na máquina virtual instalada no *browser Web*.

Na máquina do servidor J2EE são encontrados componentes da tecnologia JSP (*Java Server Pages*) sendo que este é mais abordado na Seção 2.2.1., e o componente EJB (*Enterprise JavaBeans*) que é utilizado para gerenciar fluxos de dados entre componentes que rodam na máquina de cliente e componentes que rodam na máquina do servidor J2EE, e entre componentes que rodam na máquina do servidor J2EE e componentes que rodam na máquina do servidor de banco de dados.

2.2.1. JSP

JSP (*Java Server Pages*) é uma tecnologia voltada para *Web* desenvolvida pela *Sun*, sendo que o JSP é uma extensão de *servlet* e ambas trabalham com solicitações e requisições HTTP.

Segundo Kurniawan (2004), uma das diferenças de trabalhar com JSP ao invés de *servlet*, é que o programador JSP não tem vínculo com o *webdesigner* no que diz respeito à programação, ou seja, quando programa-se em *servlet*, o programador precisa enviar todo código HTML pelo seu programa, sendo assim, mesmo que uma página tenha pouco código, o programador precisa mandar uma longa instrução HTML, além disso, quando o w*ebdesigner* quiser fazer qualquer tipo de alteração, ele tem que pedir para que o programador *servlet* faça a alteração. Já a programação JSP além de suportar HTML e XML é mais dinâmica, ou seja, o programador JSP adiciona seu código dentro de *tags* especiais (<%.....%>) em qualquer lugar do código HTML, onde o mesmo é salvo com a extensão .jsp, sendo assim o programador JSP não tem vínculo com o *webdesigner*.

O processamento de páginas JSP inicia-se com a requisição da mesma pelo usuário. Após o recebimento desta, o *container* JSP verifica se existe um arquivo da página previamente compilado e se este corresponde a mesma versão disponível na fonte da aplicação. Depois o *container* JSP interpreta a página e envia a resposta para o usuário. O conteúdo desta resposta é apenas texto ou código HTML, o usuário da página não consegue visualizar o código fonte da aplicação.

Como já dito no Capítulo “Desenvolvimento *Web*”, as requisições e respostas de aplicações w*eb* utilizam o protocolo HTTP para o recebimento e o envio de requisições e respostas. E necessário lembrar também que o HTTP é do tipo *stateless*.

O JSP fornece meios e recursos para que seja mantido o estado entre requisições de páginas *Web*, estes são sessão e *cookies*.

As sessões são variáveis armazenadas no servidor, que podem conter valores que são mantidos durante toda a utilização de páginas *Web* de um *site*. Todas as vezes que existe uma requisição por parte do usuário, o *container* JSP, ao iniciar o processamento da requisição, recupera a sessão do usuário que efetuou a requisição para que possa disponibilizar o conteúdo dentro das páginas JSP a serem processadas.

Os *cookies* também armazenam conteúdos para manter o estado entre as requisições e respostas, porém eles são armazenados no computador do usuário e toda vez que é efetuada uma requisição, são enviados juntamente os *cookies* que pertencem àquele domínio da página solicitada.

2.2.1.1. Funcionamento do JSP

Para executar uma página JSP é necessário um *container* JSP, que neste caso é o Tomcat 5.5.8. Dentro deste *container* há um *servlet* especial chamado *page compiler* (compilador de página), que transforma um *container servlet* em um *container* JSP.

Quando uma página JSP é chamada inicialmente, a mesma é analisada e compilada, se não encontrar erro, a classe *servlet jsp* é carregada na memória, sendo que esta sempre deve estar atualizada. Para verificar se há necessidade de atualização, o compilador de página *servlet* compara o carimbo de horário no *servlet jsp* com a página JSP, se a página JSP for

mais atual, então precisa-se fazer a recompilação, caso contrário não há a necessidade da recompilação.

2.3. JDBC

Java DataBase Connectivity (JDBC) é uma *Application Programming Interface* (API) desenvolvida pela *Sun* com objetivo de fazer o acesso ao banco de dados de forma padronizada, ou seja, padronizar o acesso a qualquer tipo de Sistema Gerenciador de Banco de Dados.

Visando o ganho de tempo e estabilidade no desenvolvimento desta API, a Sun se baseou em outras APIs bem sucedidas, como por exemplo a *Open DataBase Connectivity* (ODBC), sendo que esta foi criada para padronizar o acesso a banco de dados no ambiente Windows.

Esta API tem a finalidade de ser flexível e simples para os programadores. De acordo com Reese (2001) “o conjunto das classes que implementam as interfaces JDBC para um determinado mecanismo de banco de dados é chamado *driver* JDBC”, sendo que estes são fornecidos e desenvolvidos por seus respectivos fabricantes. Como conseqüência, o programador se preocupa somente com sua aplicação, conforme a Figura 2.1.

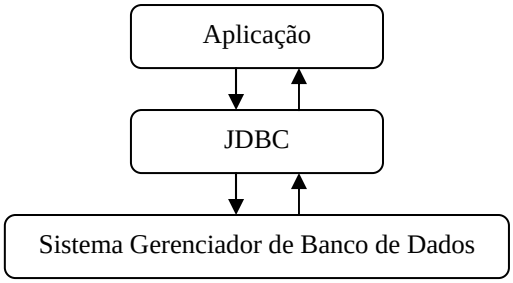

Figura 2.1: Arquitetura JDBC

2.4. Considerações Finais

Para que haja um melhor entendimento por parte do leitor com relação ao desenvolvimento da ferramenta, é necessário que o mesmo tenha absorvido as informações acima mencionadas, de forma a conseguir relacionar a tecnologia utilizada em cada etapa do desenvolvimento no Capítulo 3.

CAPÍTULO 3 - FERRAMENTA BRD TOOL

A Ferramenta BRD Tool foi construída com base em aplicação *Web* usando *Java Server Pages* (JSP), sistema gerenciador de banco de dados PostgreSQL, e utilizando-se o Tomcat 5.5.8 como *container* JSP.

A escolha do Sistema Gerenciador de Banco de Dados Relacional (SGBDR) PostgreSQL foi feita tendo em vista que é um SGBD livre (*open source*), pois a construção da ferramenta tem o intuito de ser comercializada, além disso o SGBDR PostgreSQL é multiplataforma e oferece funcionalidade necessária para a ferramenta.

Neste Capítulo é abordado a arquitetura da ferramenta, o modelo do banco de dados da ferramenta e é criado um estudo de caso para poder mostrar o resultado obtido.

3.1. Arquitetura

Na Figura 3.1. ilustra-se a arquitetura da ferramenta em questão, assim pode-se visualizar com maior clareza como é que é feito todo o processamento para gerar um caso de uso com sua respectiva regra de negócio.

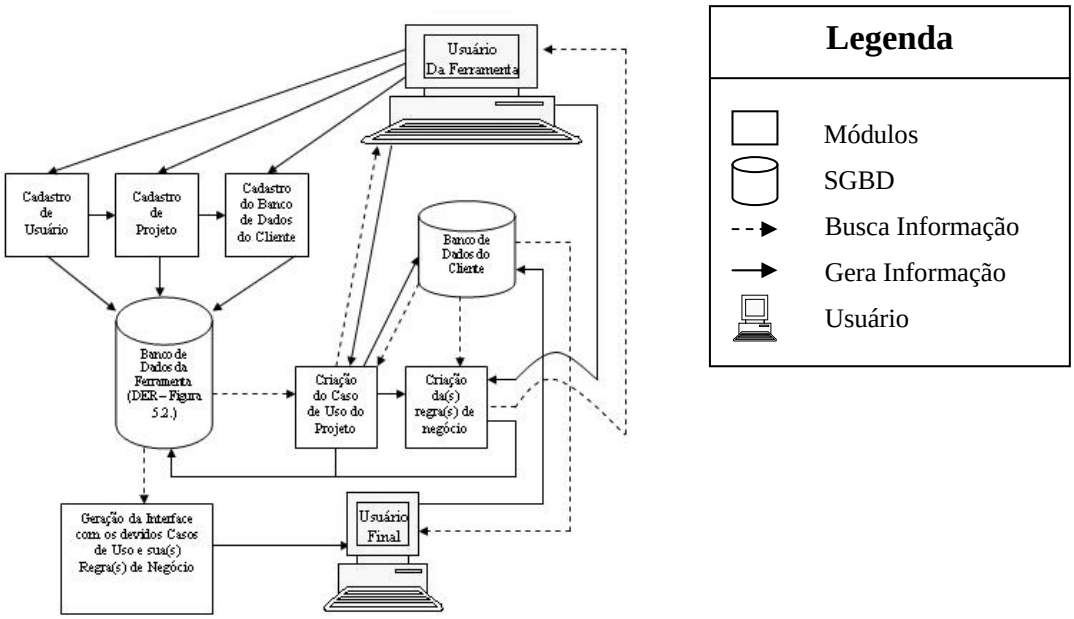

Figura 3.1: Arquitetura da ferramenta BRD Tool

Primeiramente terá que ser feitos todos os cadastros para poder dar início à criação do caso de uso e sua repectiva regra de negócio. No cadastro de usuário são armazenados os dados dos usuários que são responsáveis por determinados projetos, sendo que os mesmos são determinados no cadastro de projeto, ou seja, quando cadastrar um projeto, são selecionados os usuários responsáveis pelo projeto em questão, além de armazenar os dados do projeto.

No cadastro do banco de dados do cliente são armazenados os dados do banco de dados do cliente, que são utilizados para fazer a conexão com o banco de dados do cliente e a extração dos metadados.

Na criação dos eventuais casos de uso e sua regra de negócio, são utilizados os dados do banco de dados do cliente (metadados), de modo a trazer flexibilidade para que a ferramenta possa trabalhar em cima de um banco de dados já existente. A forma como é trabalhada na extração do mesmo é detalhada na Seção 3.3..

No cadastro de caso de uso e regra de negócio são exigidas todas as informações necessárias para que a ferramenta consiga realizar a geração, informações essas que são armazenadas no banco de dados da ferramenta.

Após todas as informações necessárias estarem armazenadas no banco de dados da ferramenta, torna-se possível a criação de um formulário ou de um relatório para o usuário final, e o mesmo é criado na linguagem JSP, além de continuar utilizando o banco de dados já existente (banco de dados do cliente).

Na Seção 3.2 há uma ilustração do diagrama de entidade e relacionamento do banco de dados da ferramenta, para melhor entender como as informações são armazenadas.

3.2. Modelo do Banco de Dados

Segundo Date (1991), banco de dados relacional nada mais é que relacionamento entre entidades (objetos), sendo que a mesma deve ser feita de forma coerente e segura através de chaves primárias e chaves estrangeiras conforme se pode ver no Diagrama Entidade Relacionamento (DER) abaixo (Figura 3.2).

A Figura 3.2. descreve o relacionamento entre as tabelas necessárias para a Ferramenta BRD Tool, através de linhas, sendo que em cada tabela são apresentados seus respectivos campos, que estes podem ser chave primária e/ou chave estrangeira.

A representação do campo que é chave primária é feita através de uma figura que simboliza uma chave, e o campo que é chave estrangeira é representado através da sigla “FK” entre parênteses.

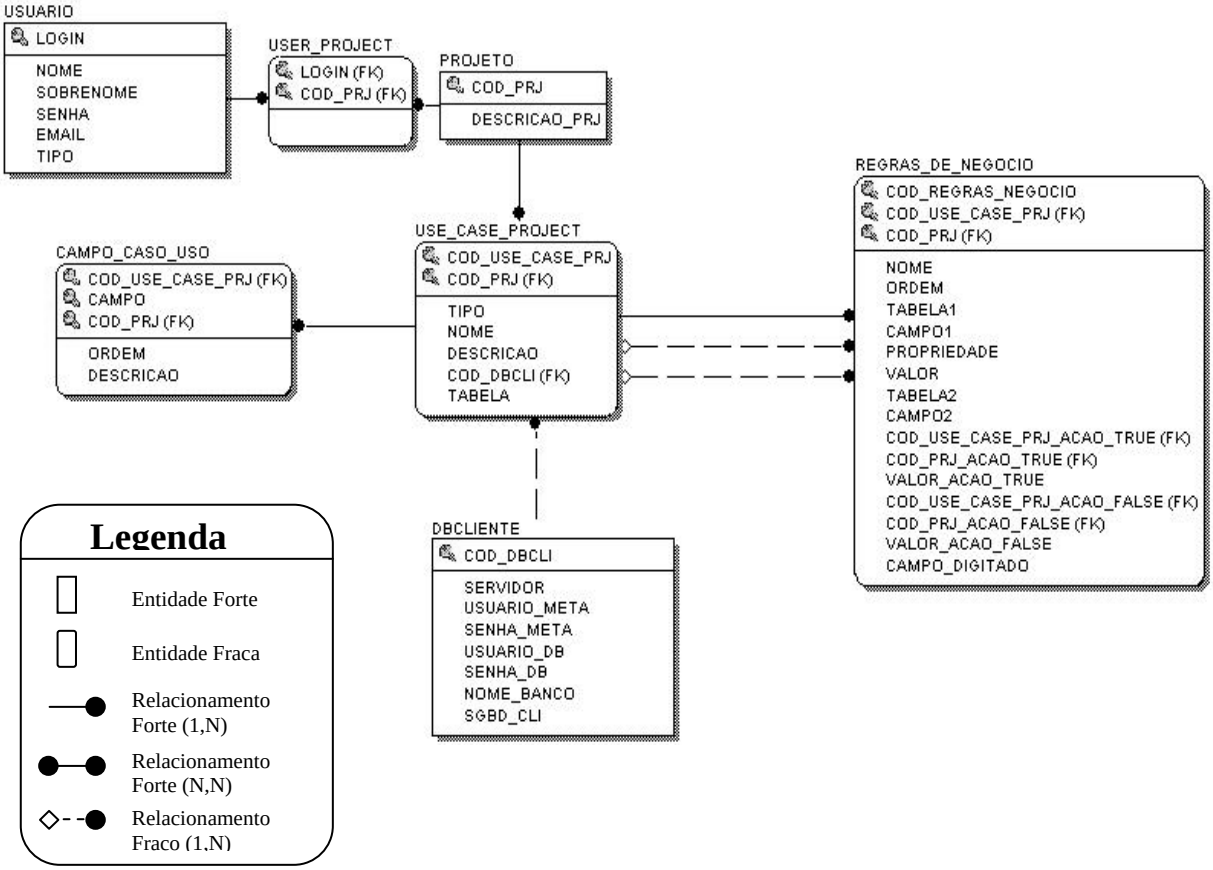

Figura 3.2: Diagrama Entidade Relacionamento (DER)

3.3. Estudo de Caso

Inicialmente é abordado a parte onde a ferramenta faz a verificação do usuário (*login*), ou seja, a ferramenta verifica se o usuário está cadastrado, se a condição for verdadeira, o mesmo verifica em qual projeto o usuário está atuando. Se o usuário não está cadastrado ou se o mesmo está cadastrado porém não estiver participando de nenhum projeto, é enviado uma mensagem informando qual restrição está ocorrendo, ou seja, usuário inválido ou usuário que não está cadastrado em nenhum projeto, conforme ilustram as Figuras 3.3 e 3.4, respectivamente.

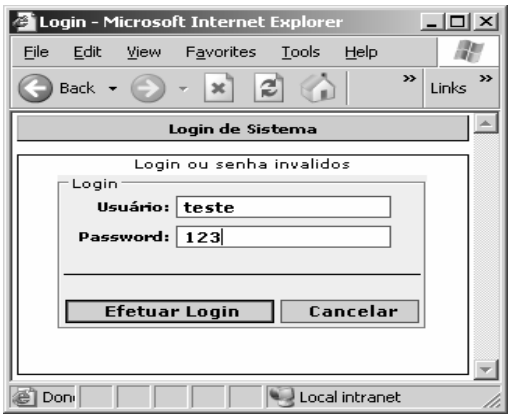

Figura 3.3: Tela de *Login* – Erro de usuário ou senha

Figura 3.4: Tela de *Login* – Nenhum projeto cadastrado para o usuário

Quando a verificação de usuário é efetuada com sucesso, abre-se uma página com o controle da ferramenta (conforme Figura 3.5), ou seja, o usuário aceito (*logado*) pode cadastrar novos usuários, cadastrar banco de dados do cliente, cadastrar novos projetos e pode criar casos de uso com suas respectivas regras de negócio para cada projeto.

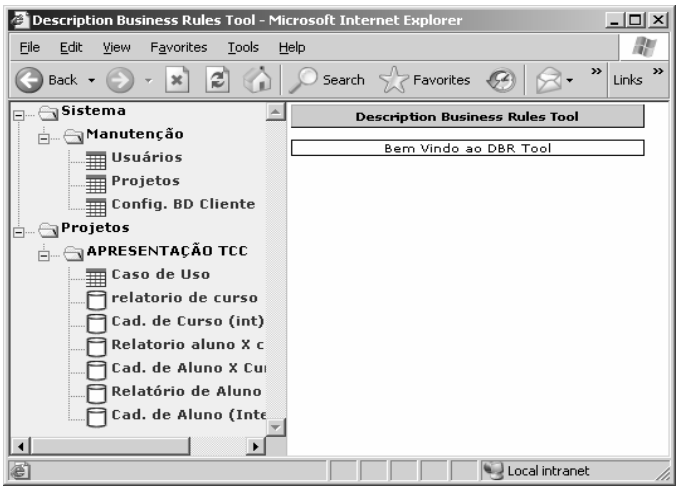

Figura 3.5: Tela de apresentação da ferramenta

Pode ser observado na Figura 3.5. um menu (na lateral) no qual é possível visualizar os projetos em que o usuário está cadastrado. Quando o usuário é *logado* no sistema, os

projetos em que o usuário está cadastrado são armazenados em uma variável de sessão para que os projetos possam ser exibidos na próxima página, conforme é mostrado na Figura 3.5.

O cadastro de usuário é simples, sendo que no mesmo pode-se fazer a inserção, exclusão e alteração. A busca não é necessária, pois todos os usuários estão listados em uma página anterior à página de cadastro de usuário, conforme a Figura 3.6.

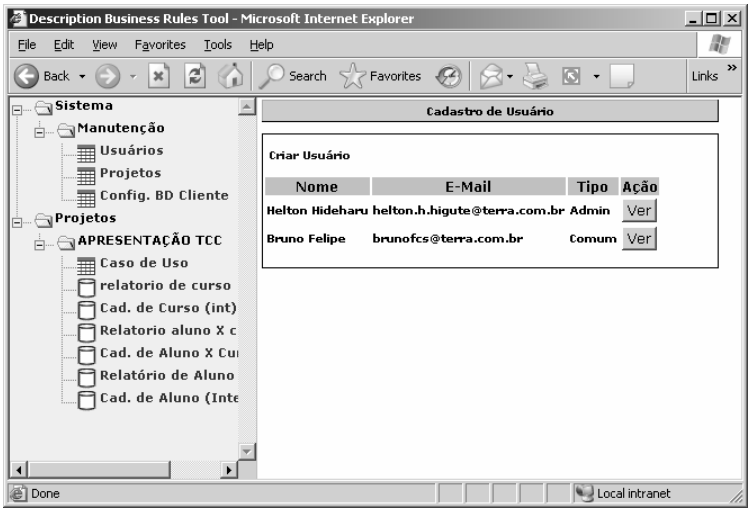

Figura 3.6: Tela de usuários cadastrados

Para criar novos usuários basta clicar no *link* “Criar Usuário”, onde redirecionando-se para a página de cadastro de usuário (Figura 3.7). O botão “Ver” (mostrada na Figura 3.6) redireciona para a página de cadastro de usuário, com a diferença que os campos estão preenchidos com os respectivos dados do usuário, podendo os mesmos serem alterados, excluídos ou simplesmente consultados com maiores detalhes.

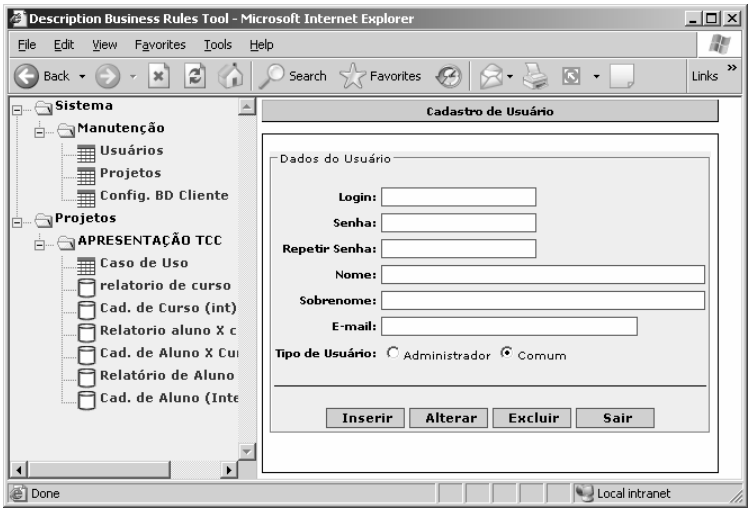

Figura 3.7: Tela de cadastro de usuário

O cadastro de projetos tem a finalidade de cadastrar projetos relacionando o(s) usuário(s) com o projeto, ou seja, em um projeto pode ter um ou mais usuários cadastrados para criação e/ou manutenção do mesmo. A metodologia empregada para o cadastro de projeto é a mesma que foi utilizada para o cadastro de usuário, conforme se vê nas Figuras 3.8 e 3.9.

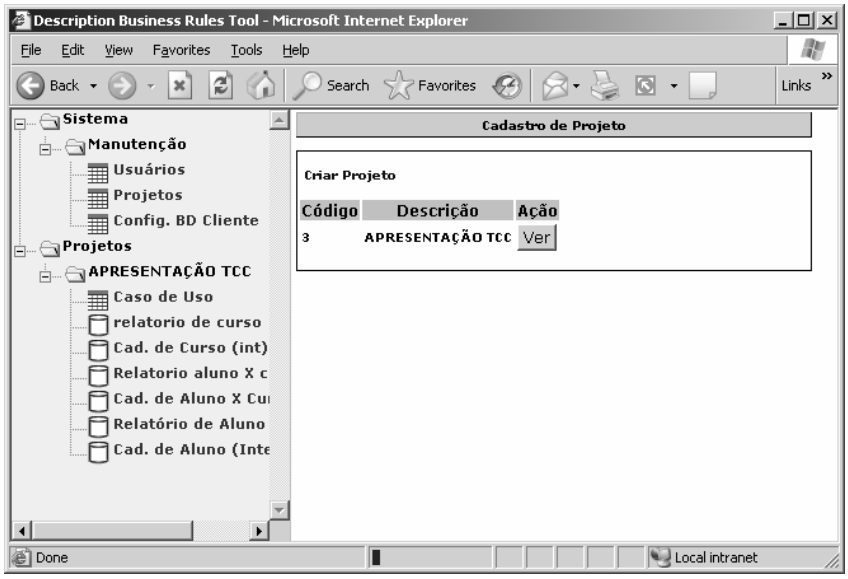

Figura 3.8: Tela de projetos cadastrados

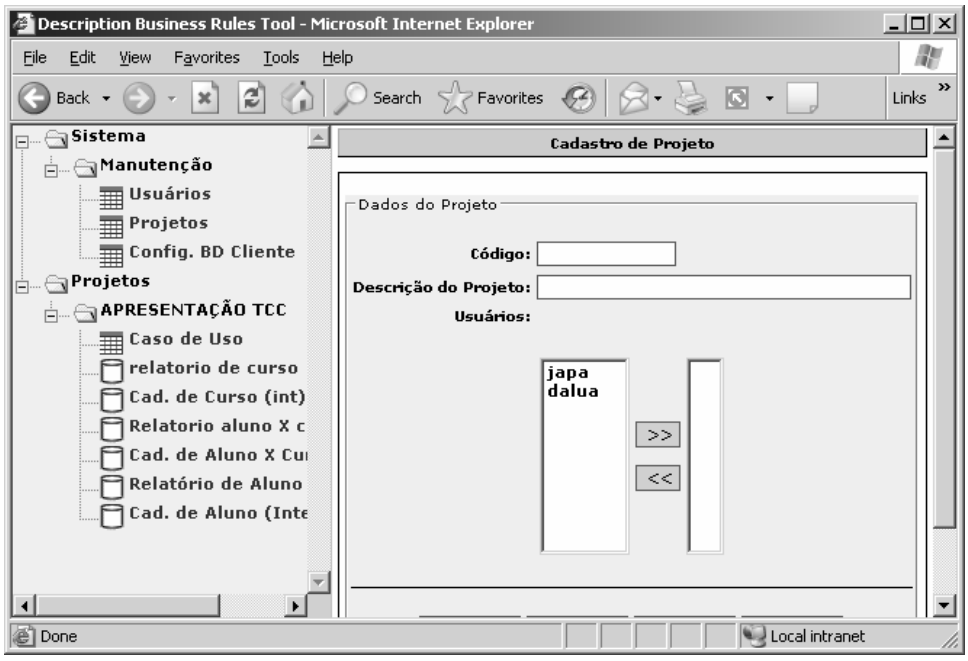

Figura 3.9: Tela de cadastro de projeto

No cadastro da configuração do banco de dados do cliente são cadastrados os dados necessários para poder realizar a conexão com o mesmo e fazer a extração dos metadados, sendo que este é utilizado para a criação de casos de uso e regras de negócio. A metodologia empregada para o cadastro da configuração do banco de dados é a mesma que foi utilizada para o cadastro de usuário e o cadastro de projetos (Figuras 3.10 e 3.11).

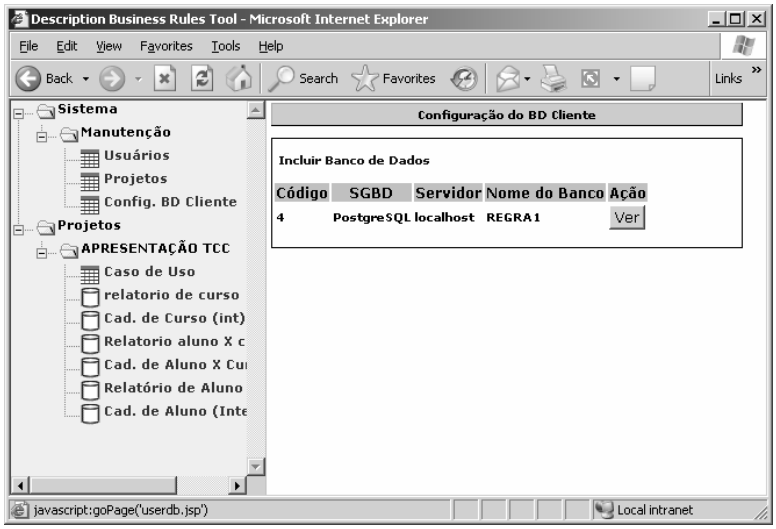

Figura 3.10: Tela de banco de dados do cliente cadastrados

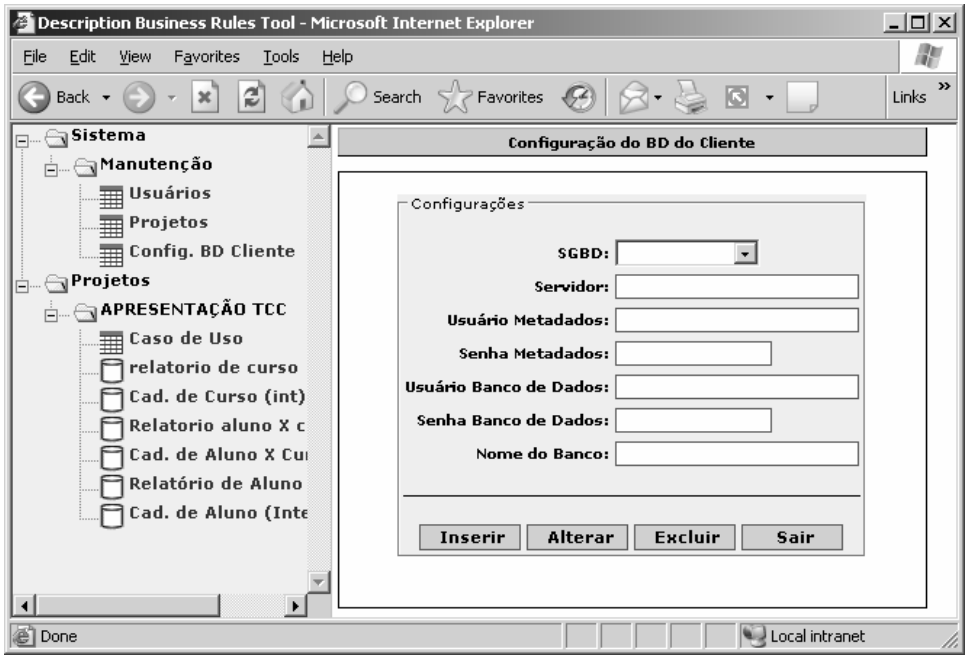

Figura 3.11: Tela de cadastro do banco de dados do cliente

Depois de ter cadastrado as informações necessárias, pode-se começar a criar os casos de uso com suas respectivas regras de negócio de um determinado projeto.

Primeiramente é informado à ferramenta o tipo do caso de uso, podendo ser um relatório ou um formulário, o nome do caso de uso e uma breve descrição do mesmo (Figura 3.12).

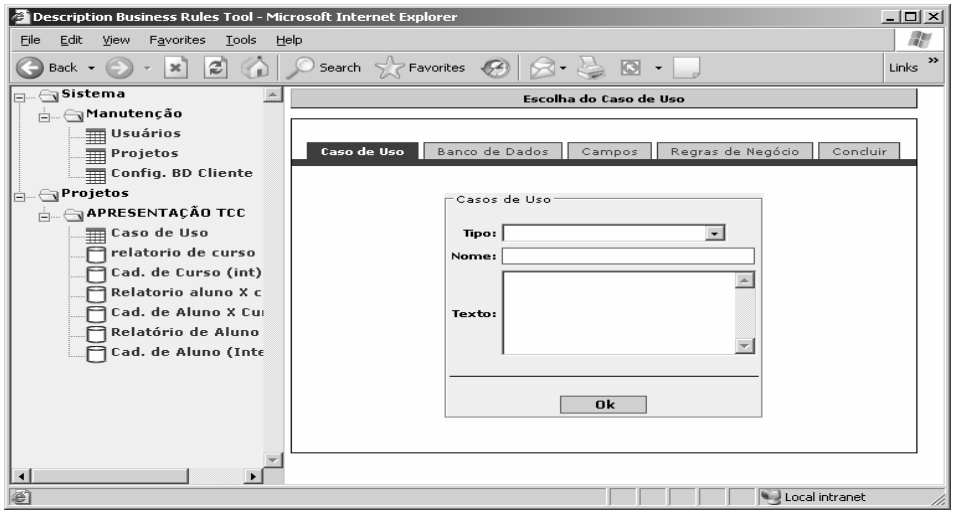

Figura 3.12: Montagem do Caso de Uso – Tipo, nome, descrição do caso de uso.

Seguindo a ordem das abas encontradas na página de cadastro de caso de uso, a escolha do banco de dados (*database*) é feita, bem como as tabelas envolvidas (Figura 3.13), visto que o banco de dados é exibido no *combo* através de uma classe chamada `ConexaoUserDB`, sendo que esta é chamada por um *bean*.

A classe `ConexaoUserDB` faz a conexão com o banco de dados da ferramenta, e logo em seguida executa uma instrução SQL através do método `executeQuery` da classe `Statement`, sendo que esta retorna uma ou mais linhas para a classe `ResultSet`. No caso foi executada uma instrução SQL em que retornasse todas as linhas da tabela “DBCLIENTE”, e através do método `getString` da classe `ResultSet`, foram listados apenas os nomes dos bancos de dados.

Quando são listadas em um *combo* todas as tabelas encontradas no banco de dados selecionado, é invocado a classe `ChangeFieldDB` através de um outro *bean*.

A classe `ChangeFieldDB` faz a conexão com o banco de dados do cliente, e logo em seguida a classe `DatabaseMetaData` recebe o método `getMetaData` da classe `Connection`, tendo em vista que a classe `DatabaseMetaData` é utilizada para extrair os nomes das tabelas através do método `getTables`, retornando o resultado para a classe `ResultSet`.

Figura 3.13: Montagem do Caso de Uso – Escolha do banco de dados do usuário e a tabela.

Em seguida, na aba “Campos” devem ser selecionados os campos necessários para a construção do relatório ou do formulário (Figura 3.14).

Os campos são exibidos por meio da invocação da classe `ChangeFields` por um *bean*, onde o método `startChangeFields` da classe `ChangeFieldDB` executa uma instrução SQL através do método `executeQuery` da classe `Statement`, e esta instrução é atribuída a um `ResultSet`. Logo após esta etapa, o método `getMetaData` da classe `resultSet` é atribuída à classe `ResulSetMetaData,` e a mesma é utilizada para buscar os nomes dos campos através do método `getColumnLabel.`

Pode-se observar que alguns campos já são selecionados automaticamente, isso ocorre quando existe alguma chave primária ou algum campo onde não possa ser nulo (Figura 3.14).

A verificação quanto à chave primária é feita pelo método `startGetPrimaryKey` da classe `ChangeFieldDB`, onde a mesma retorna um `ResultSet` com os campos que são chave primária. Após a conexão com o banco de dados do cliente, é atribuída à classe `DatabaseMetaData` o método `getMetaData` da classe `Connection`, sendo que a classe `DatabaseMetaData` é utilizada para atribuição do resultado para o `ResultSet` através do método `getPrimaryKeys`.

A verificação quanto ao campo pode ser nula ou não é feita na própria página JSP através do método `isNullLable` da classe `ResultSetMetaData`, onde é percorrido campo por campo e passado para o método `isNullLabel` um valor inteiro como referência, e este valor é percorrido por uma estrutura de repetição cujo valor máximo é obtido pelo método `getColumnCount` da classe `ResultSetMetaData`, em que este retorna à quantidade de colunas de uma tabela.

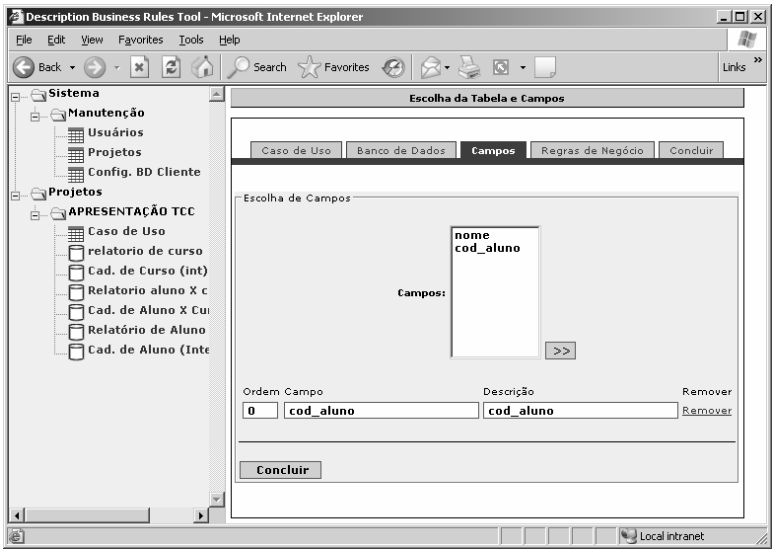

Figura 3.14: Montagem do Caso de Uso – Escolha dos campos da tabela selecionada.

Na aba “Regras de Negócio” são criadas eventuais regras de negócio sendo que inicialmente há somente o tipo comparação. Para inserir regras de negócio, basta clicar no *link* “Adicionar”, conforme ilustrada na Figura 3.15.

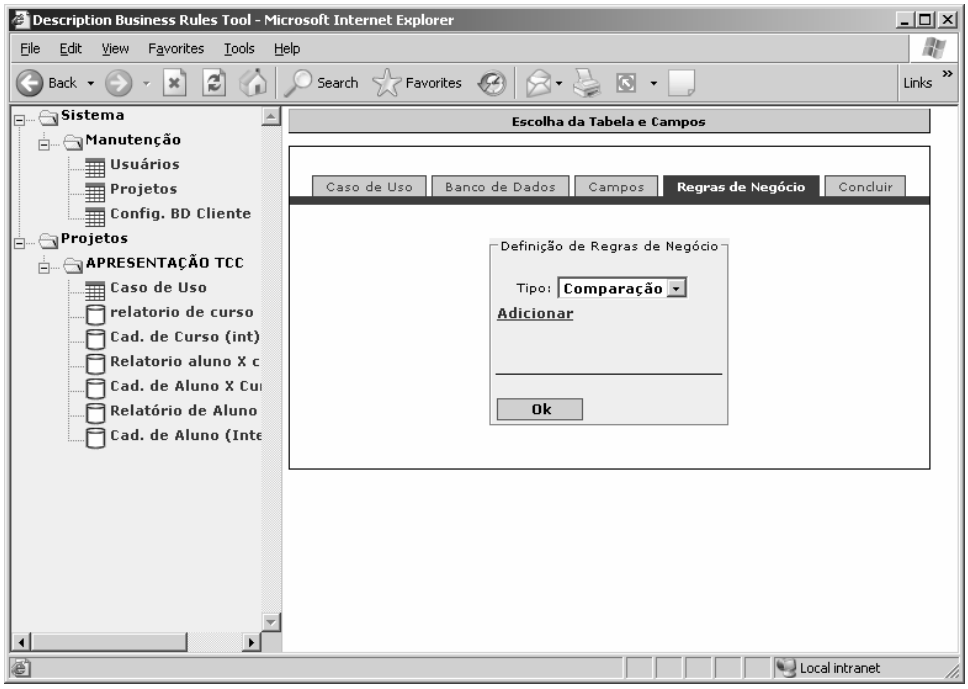

Figura 3.15: Montagem do Caso de Uso – Tela de exibição das regras de negócio criadas.

No cadastro de regras de negócio, deve ser informado o nome da regra de negócio a ser criada, a tabela a ser comparada com o respectivo campo selecionado, o tipo de comparação (>,<,=,!= ...), o valor a ser comparado, caso seja estático, ou o campo selecionado, caso seja dinâmico, a ação a ser tomada, podendo a mesma invocar um outro caso de uso ou apenas mostrar uma mensagem tanto para a ação verdadeira como para a ação falsa. Se a ação for uma mensagem, então deve-se informar qual mensagem deseja enviar para o usuário.

A comparação também pode ser entre campos de tabelas distintas, sendo assim, devendo informar qual a tabela e o campo da tabela a ser comparada Figura 3.16.

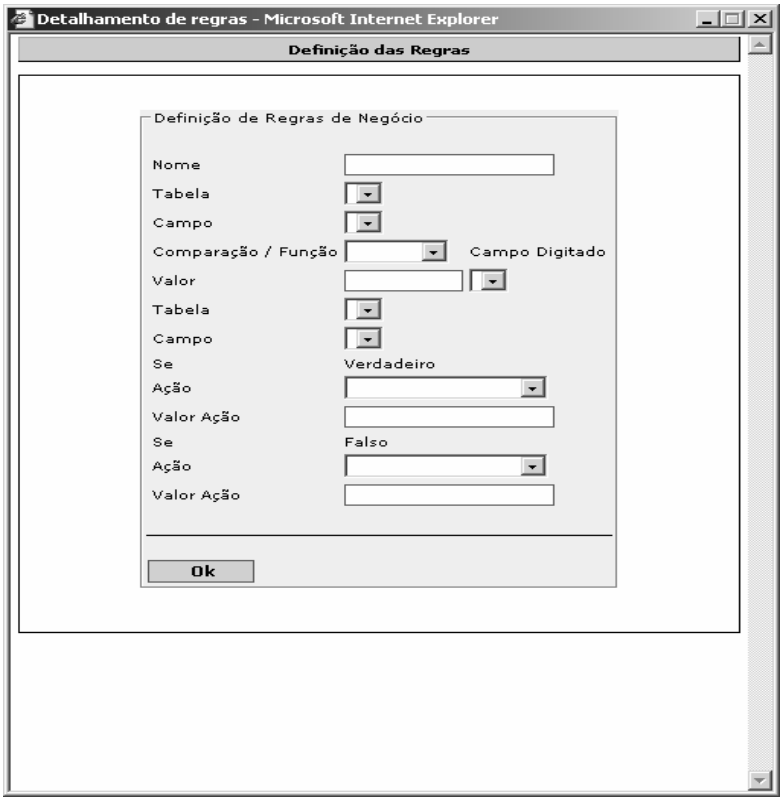

Figura 3.16: Montagem do Caso de Uso – Montagem das eventuais regras de negócio.

A forma como é efetuada a seleção da tabela e dos campos é a mesma utilizada na aba “Banco de Dados” e na aba “Campos”.

E finalmente na aba “Concluir”, é feito apenas um retorno (Figura 3.17), e o usuário deve verificar se estão corretos os procedimentos que o sistema deve tomar.

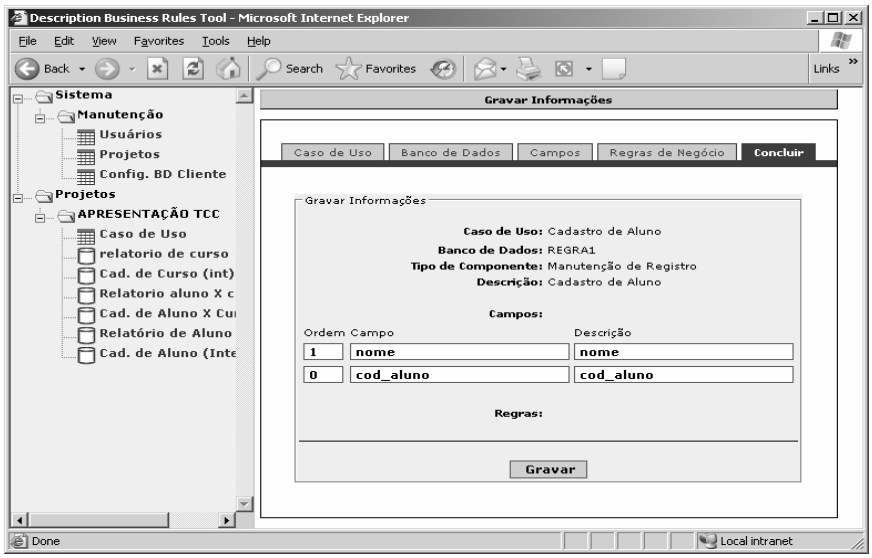

Figura 3.17: Montagem do Caso de Uso – *Feedback* do caso de uso a ser criado.

No momento em que o usuário clica em gravar, o sistema efetua a inserção no banco de dados de todos os ítens selecionados ao longo do cadastro de Caso de Uso do Projeto, ou seja, a cada escolha do usuário, os valores não são armazenados imediatamente no banco de dados, são sim armazenados em variáveis de sessão para que na conclusão tais valores sejam armazenados de uma só vez.

As variáveis de sessão existem exatamente para que um *container* consiga manter o estado de acessos de usuários individualmente, com isso, torna-se possível o armazenamento de dados entre a navegação de uma determinada aplicação *Web* para cada usuário, nas diversas páginas pertencentes à aplicação.

A variável de sessão utilizada para armazenamento dos valores escolhidos pelo usuário que está utilizando a aplicação, que após o *click* no botão de gravar é percorrida e seus valores inseridos no banco de dados, é um *ArrayList* com índices numéricos que contêm a seguinte estrutura:

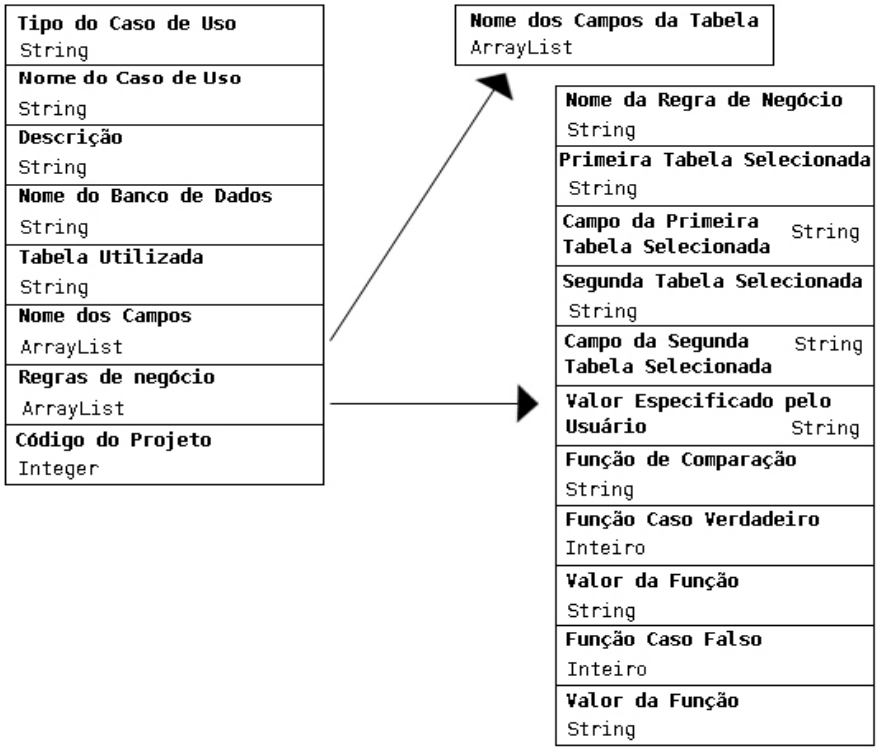

Figura 3.18: Estrutura do *ArrayList*.

Esta estrutura, então, é utilizada para o armazenamento do estado dos valores selecionados pelo usuário de desenvolvimento do cadastro de um caso de uso de projeto. Após a visualização do usuário da tela de concluir, são mostrados os dados selecionados ao longo do cadastro. Esta variável de sessão é decomposta e seus valores inseridos no banco de dados.

A escolha de uma variável de sessão para o armazenamento de valores foi utilizada para efetuar a inserção dos dados do cadastro de casos de uso de projeto de uma única vez, para mantermos a atomicidade da operação. Ficaria muito mais difícil de ser controlada a atomicidade da operação se os dados fossem incluídos operação a operação do cadastro de caso de uso de projeto.

3.4. Demonstração de Como a Aplicação é Gerada

No módulo de geração, após o usuário ter efetuado o *login*, a aplicação mostra os projetos e casos de uso de projetos do usuário *logado*, iniciando-se a visualização dos casos de uso do projeto cadastrados previamente no módulo explicado acima, que o mesmo deseja utilizar.

Todo o processo de geração da aplicação do usuário é efetuado dinamicamente, ou seja, não é gerado nenhum tipo de código prévio e armazenado em algum arquivo para uso posterior, e sim os dados para o caso de uso de projeto escolhido e a aplicação de geração de todos os componentes necessários para que possam ser mostrados para o usuário e o mesmo inicializar a utilização.

No momento que o usuário seleciona um caso de uso de projeto, a aplicação efetua a seleção dos dados e armazena em uma variável que contém a mesma estrutura da variável de sessão que armazena os dados para que o caso de uso de projeto quando cadastrado seja armazenado no banco de dados. Neste momento é iniciada a geração da aplicação para a visualização do usuário.

Os primeiros passos efetuados são para que usuário possa identificar na tela o caso de uso de projeto selecionado e a identificação do tipo de caso de uso para que o sistema possa iniciar a geração da aplicação para o usuário, então, respectivamente é mostrado o nome do caso de uso de projeto no topo da página visualizada e em seguida a aplicação identifica o tipo de caso de uso selecionado. A identificação do tipo indica para a aplicação se ela deve gerar um formulário para a manutenção de registro caso o tipo seja manutenção de registro, ou deve mostrar uma tabela com os dados de um relatório caso o tipo seja relatório.

Seguem abaixo os procedimentos tomados pela aplicação, numerados e discriminados em seguida, para o tipo de caso de uso igual a relatório.

1 – Geração da tabela em linguagem HTML.

Este trecho inicia a geração de *tags* HTML para a construção de uma tabela, incluindo os cabeçalhos dos campos desta tabela que são iguais aos nomes dos campos selecionados no cadastro de caso de uso do projeto.

2 – Geração da *query* em linguagem SQL

Este procedimento irá criar uma SQL capaz de retornar os dados do banco de dados que o usuário especificou no cadastro de uso do projeto.

Para isto são utilizados os dados dos campos selecionados no cadastro, as tabelas e, caso existam, as regras de negócio. Os campos selecionados são os mesmos mostrados no cabeçalho da tabela HTML, como exige a linguagem SQL são colocados após a instrução *SELECT*. Quando efetuado o cadastro do caso de uso de projeto, o usuário seleciona apenas

uma tabela, no caso de não existir então, as regras de negócio são adicionadas à tabela selecionada após a indicação da tabela no *FROM* da SQL e a geração da SQL é terminada.

Caso exista alguma regra de negócio, as tabelas cadastradas nestas regras também são adicionadas após a cláusula *FROM* da SQL. O problema é que quando existir mais de uma tabela na cláusula *FROM* é necessária a igualação das chaves para que os dados retornados pela *query* sejam válidos para o usuário. Este processo de igualação das chaves é efetuado pelo método `checkPKandFK` da classe `ForeignKey` utilizado como *bean* da página de geração da aplicação. Este método recebe como parâmetro as tabelas, através de extrações de metadados do banco de dados selecionado pelo usuário. Este método identifica as chaves que devem ser igualadas e retorna uma *string* contendo a igualação, por fim esta *string* é concatenada à SQL após a cláusula *WHERE*.

A cláusula *WHERE* também contém as restrições cadastradas na regra de negócio no módulo anterior. Neste cadastro são selecionados as tabelas, campos e valores para a respectiva regra de negócio. No caso de as regras de negócio conterem apenas campos a serem comparados, tais campos são adicionados à cláusula *WHERE* como também a função de comparação entre estes, que podem ser as explicadas anteriormente. Também pode ser que o usuário, ao invés de escolher apenas tabelas e campos a serem selecionados, escolha apenas uma tabela e campo e um valor, neste caso é adicionada à cláusula *WHERE* apenas a comparação do campo com o valor.

3 – Execução da *query* gerada pela aplicação

Após a geração da SQL, a mesma deve ser executada para que o resultado obtido para o relatório seja mostrado ao usuário. Para tal execução é utilizado o método `selectQuery` da classe `ExecuteQuery` que foi adicionada como um *bean* para esta página.

O método `selectQuery` recebe como parâmetro uma *string*, que corresponde à *query* SQL, que deve ser executado no banco de dados. Após a execução é recuperado um objeto do tipo `ResultSet` com o resultado da *query* executada.

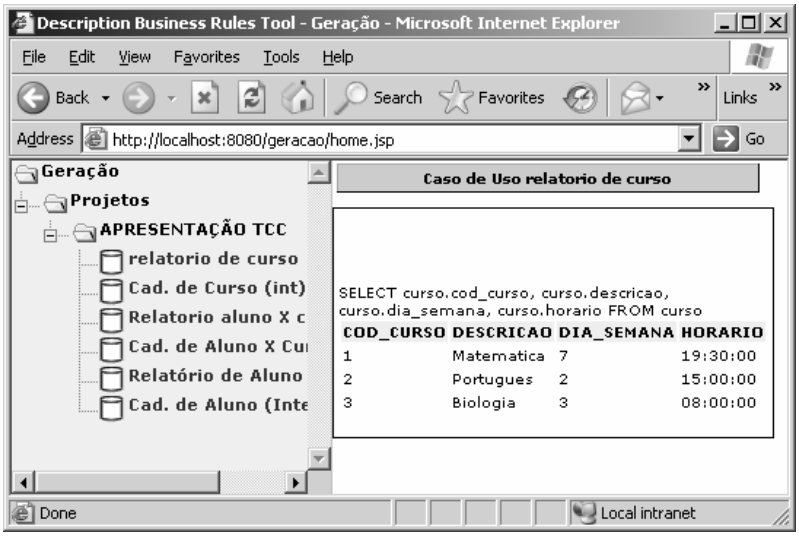

Figura 3.19: Relatório gerado a partir do caso de uso cadastrado

Após todos estes procedimentos mostra-se o resultado para o usuário, conforme a Figura 3.19.

Este objeto do tipo `ResultSet` contém o resultado da SQL executada, então se inicia a geração do trecho de código HTML correspondente às linhas e colunas da tabela já criada anteriormente.

Para casos de uso de projeto do tipo manutenção de registro, em vez de tabelas HTML com os registros das tabelas obedecendo às regras de negócio cadastradas, são gerados formulários HTML para que o usuário possa efetuar a manutenção dos registros.

Seguem abaixo os procedimentos tomados pela aplicação, numerados e discriminados em seguida, para o tipo de caso de uso igual à manutenção de registro.

1 – Geração de código *JavaScript* para validação de alguns dados do lado cliente

Códigos *JavaScript* são executados localmente nas máquinas clientes, então, a aplicação gera um código *JavaScript* para que possam ser validados os dados do formulário de manutenção de registro em casos que o campo não possam estar vazio, ou seja, os campos

que no banco de dados não possam ser nulos ou sejam chaves primárias são verificados pelo *JavaScript* gerado pela aplicação.

A geração do código *JavaScript* inicia percorrendo os campos selecionados que foram cadastrados no caso de uso do projeto, verificando se o mesmo pode ser nulo, caso não possa, é gerado um trecho de código *JavaScript* que verifica se a digitação deste campo no formulário que irá ser criado posteriormente foi efetuada. No caso deste trecho encontrar inconsistência na digitação, é gerada uma mensagem para o usuário e o envio do formulário é cancelado.

2 – Geração de campos tipo *hidden* com os valores necessários no processamento do formulário.

São três campos imprescindíveis para o processamento do formulário após os dados serem preenchidos. São eles, campo de identificação da tabela, campo de identificação do código do projeto no qual o usuário selecionou o caso de uso e campo de identificação do caso de uso do projeto selecionado.

Foram criados campos do tipo *hidden*, em vez de utilizar variáveis de sessão, principalmente para casos de alteração de código, por motivo de que, se colocarmos campos do tipo *hidden*, neste caso, ficará mais organizado, ou seja, todos os dados necessários para o processamento do formulário virá da mesma fonte, o formulário.

3 – Geração de Código HTML correspondente aos campos HTML

Neste momento a aplicação irá percorrer os campos selecionados no cadastro de caso de uso de projeto e efetuar a geração de código HTML correspondente às descrições dos campos e dos próprios campos para preenchimento.

O procedimento é iniciado com a criação de uma tabela HTML com duas colunas. Uma conterá a descrição do campo e a outra o próprio campo para digitação, cada par de

descrição e campo do caso de uso de projeto estará em uma linha diferente, um abaixo do outro.

A descrição do campo mostrada para visualização do usuário é o próprio nome do campo.

Na criação dos campos são gerados códigos HTML com as respectivas propriedades dos campos *input* da linguagem: tipo, nome, *id*, tamanho. O tipo do campo é fixo igual a *text*, este tipo de campo corresponde a um *edit*, é o tipo de campo em que o usuário digita o conteúdo. Os nomes dos campos na geração dos *edits* são os mesmos nomes dos campos na tabela do banco de dados, igualmente para os *id's*, estes campos possuem os mesmos nomes dos campos na tabela do banco de dados para que no processamento do formulário seja possível identificá-los e efetuar a recuperação de seus valores.

A propriedade de tamanho do campo da *tag input* do HTML é informada apenas se o campo for do tipo *string.* Para tal verificação é utilizado o método `checkField` da classe `CheckAttribute` utilizado como *bean* para esta página. Este método retorna um `ArrayList` contendo um valor *booleano* indicando se o campo é chave primário ou não, o tipo do campo, o tamanho do campo, um valor *booleano* indicando se o campo pode ter valor nulo e o nome do campo na tabela de onde ele pertence se for o caso de ser chave estrangeira.

Ainda neste processo de geração de código HTML para criação dos campos para posterior digitação do usuário, são criados *links* para busca nos casos em que o campo que está sendo gerado seja chave estrangeira de outra tabela. Isto torna possível a não obrigatoriedade de o usuário conhecer previamente um código de uma outra tabela. Este abre página de busca para o usuário, passando como parâmetro para a mesma, o nome do campo da tabela de origem da chave estrangeira, a tabela de origem da chave estrangeira e o banco de dados.

Na página de busca, o usuário pode navegar entre os registros existentes. Depois de encontrar o registro que contém a chave desejada e clicar sobre o *link* deste registro, o valor da chave é transportado para o formulário gerado dinamicamente.

Nos casos de uso de projeto que é do tipo de manutenção de relatório, ainda existe o processamento do formulário gerado, que é iniciado após o usuário ter completado o preenchimento do formulário e clicado no botão “Insere”. No momento em que o usuário clicar no botão para iniciar o processamento do formulário, é acionada a função *JavaScript*, gerada dinamicamente, que é responsável por efetuar a validação dos campos que são chave primária e os campos que não podem conter valores nulos. No caso em que tudo passe por esta função sem nenhuma inconsistência, os dados do formulário são enviados.

Para o processamento do formulário é utilizada a mesma página em que foi efetuada a geração da aplicação para o usuário. Este processamento também é construído por passos até o momento em que os dados informados para o usuário possam ser avaliados e inseridos no banco de dados.

Seguem abaixo a seqüência ordenada dos procedimentos tomados para o processamento do formulário enviado:

1 – Recuperação da Ação

É recuperado o valor do botão que o usuário clicou, este procedimento é utilizado para identificar o tipo de ação que o usuário solicitou. No caso, o sistema só possui uma opção “Insere”, mas este procedimento foi desenvolvido para posteriormente identificar as ações que existem no formulário para casos de uso de projeto do tipo manutenção de registro, que são “Insere”, “Altera” e “Exclui”.

2 – Recuperação das Regras de Negócio

Neste procedimento são recuperadas todas as regras de negócio informadas no cadastro de caso de uso de projeto, o sistema então inicia uma varredura para avaliar e efetuar as construções necessárias para as regras de negócio.

3 – Geração de SQL avaliadora de regra de negócio

Para cada regra de negócio cadastrada para o caso de uso de projeto é gerada uma SQL dinamicamente capaz de avaliar se os dados informados pelo usuário no formulário condizem com as regras cadastradas.

A geração começa selecionando todos os campos das tabelas envolvidas na regra de negócio com a utilização do * na instrução *SELECT*, após este procedimento são adicionadas à cláusula *FROM* os campos envolvidos na regra de negócio.

Em seguida é iniciada a construção da cláusula *WHERE* onde são adicionadas todas as restrições contidas na regra. A construção da cláusula *WHERE* depende muito da construção da regra de negócio efetuada no módulo anterior, se o usuário selecionou duas tabelas e dois campos, estes são comparados; se o usuário selecionou uma tabela e um campo e o campo valor do cadastro da regra de negócio foi preenchido, o campo da tabela é comparado com o valor preenchido; e para o último caso em que o usuário cadastrou a regra de negócio para comparação de um valor de um campo do formulário, tal condição é atribuída à cláusula *WHERE*.

Para todos os casos de escolha de avaliação cadastrados na regra de negócio são utilizadas as funções de comparação oferecidas por tal cadastro. Também nesta mesma cláusula ainda são efetuadas as mesmas atribuições de comparações de chaves primárias necessárias de todas as tabelas envolvidas, utilizando o mesmo método e classe do *bean* adicionado à página descrita na geração de SQL do relatório.

4 – Execução da SQL gerada e avaliação da regra de negócio

Após todo o processo de geração da SQL dinamicamente, é necessário efetuar a execução da mesma. Este procedimento é efetuado pelo método `selectQuery` da classe `ExecuteQuery` adicionado como *bean* para esta página.

Este método é chamado passando-se como parâmetro uma *string* contendo a SQL gerada, e retornando um valor verdadeiro ou falso. O valor de retorno deste método é avaliado e, para os casos em que o valor é verdadeiro, é efetuada mais uma verificação, a que avalia se no cadastro de regra de negócio no caso de uso do projeto foi selecionado um caso de uso ou uma mensagem. Para a escolha do caso de uso, o usuário é redirecionado para o caso de uso selecionado na regra de negócio, e em caso de mensagem, é gerado um trecho de código *Javascript* mostrando uma mensagem para o usuário, mensagem esta que também foi informada no cadastro da regra de negócio no caso de uso do projeto.

Para o caso de a regra de negócio ser avaliada como falsa, são efetuadas as mesmas verificações e ações para o caso em que for avaliado como verdadeira, com a diferença de que os dados utilizados pelo processamento do formulário são os que no cadastro da regra de negócio foram selecionadas nos campos que indicam a ação como falsa.

5 – Inserção dos dados informados no banco de dados

Este procedimento finaliza o processamento do formulário, verificando como a regra de negócio foi avaliada. Caso a regra tenha sido avaliada como verdadeira, então é iniciada a geração de código SQL para inserir os dados no banco de dados.

A geração da SQL começa informando os campos da tabela que irão receber valor e, em seguida, são passados seus valores após a cláusula *VALUES*. É necessário observar que quando os campos são do tipo *string*, verificados pelo método `checkAttribute` já explicado acima, é adicionado um conjunto de aspas no início e no final, caracteres estes que servem para delimitar *strings* em SQL.

Após a geração, a SQL gerada é executada pelo método `insertQuery` da classe `ExecuteQuery`, e com a execução desta *query* os dados são inseridos no banco de dados.

3.5. Resultados Obtidos

Neste tópico são mostrados formulários e relatórios gerados pela ferramenta BRD Tool com regras de negócio simples. As etapas para criação do aplicativo desejado foram abordadas anteriormente.

O exemplo utilizado para realizar a geração dos formulários e dos relatórios é baseado em um sistema acadêmico bem simples, ou seja, somente estão sendo abordados o cadastro de aluno, cadastro de curso e o cadastro de aluno com curso. O DER do exemplo utilizado é mostrado na Figura 3.20.

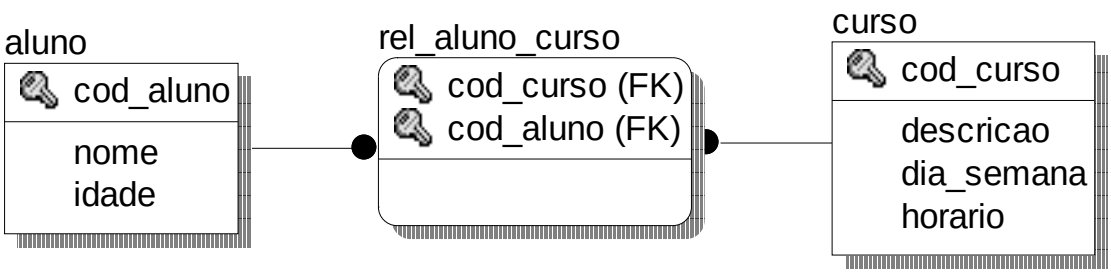

Figura 3.20: DER utilizado no estudo de caso.

Na Figura 3.21, ilustra-se um relatório gerado pela BRD Tool, em que a mesma filtra somente os cursos cadastrados na segunda feira.

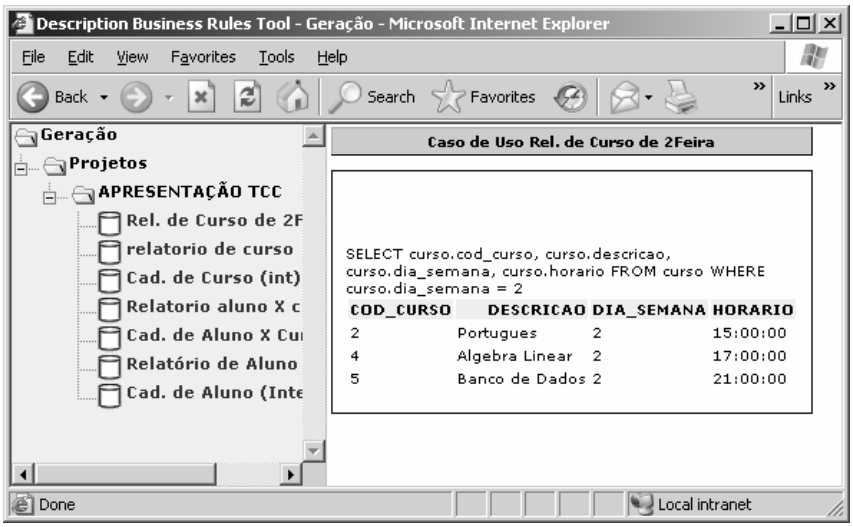

Figura 3.21: Relatório gerado pela ferramenta BRD Tool, onde são impressos somente os cursos cadastrados na segurnda-feira

Na Figura 3.22, está apresentado um cadastro de curso gerado pela ferramenta BRD Tool, que faz a verificação de chave primária, ou seja, não permite cadastrar curso com um código já existente. Quando ocorre algum tipo de erro é emitido um alerta para o usuário.

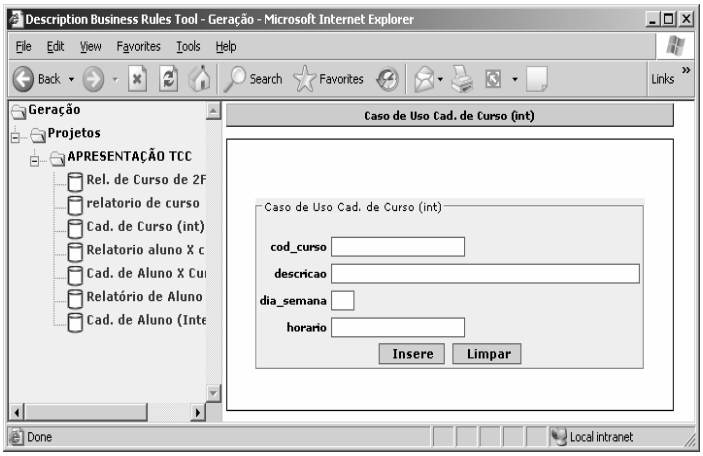

Figura 3.22: Formulário de curso gerado pela ferramenta BRD Tool.

Já na Figura 3.23, temos um cadastro de aluno relacionado com o curso, e nesse caso, quando há campos que são chave estrangeira, é criado um *link* ao lado do mesmo para que o usuário tenha a opção de selecionar o curso e/ou o aluno desejado através de uma consulta. Esta consulta imprime todos os dados dos alunos ou dos cursos de modo que as chaves

primárias acabam se tornando um *link*. Ao serem clicados, são enviados para o respectivo campo os valores necessários, mostrada na Figura 3.24.

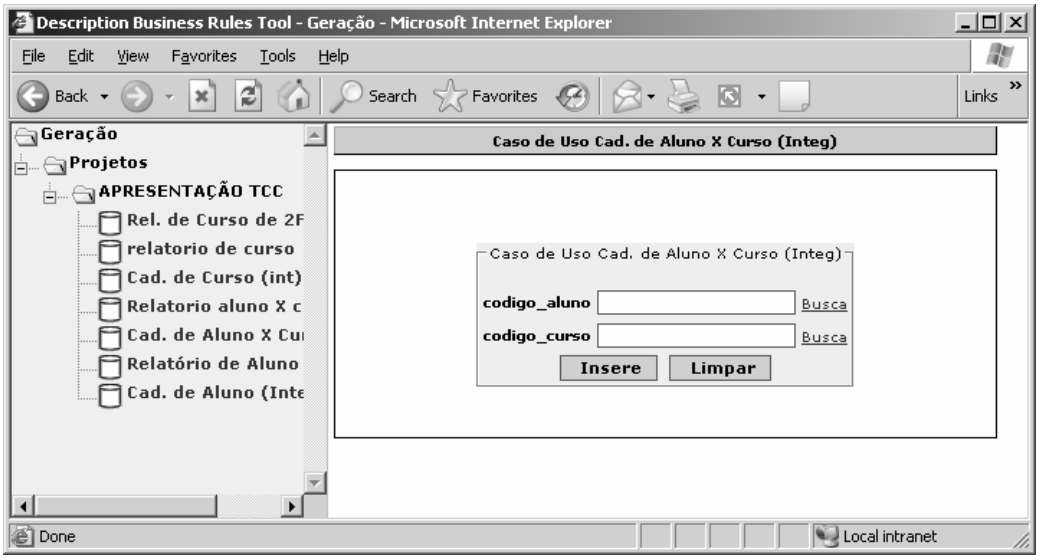

Figura 3.23: Formulário de aluno relacionado com o curso gerado pela ferramenta.

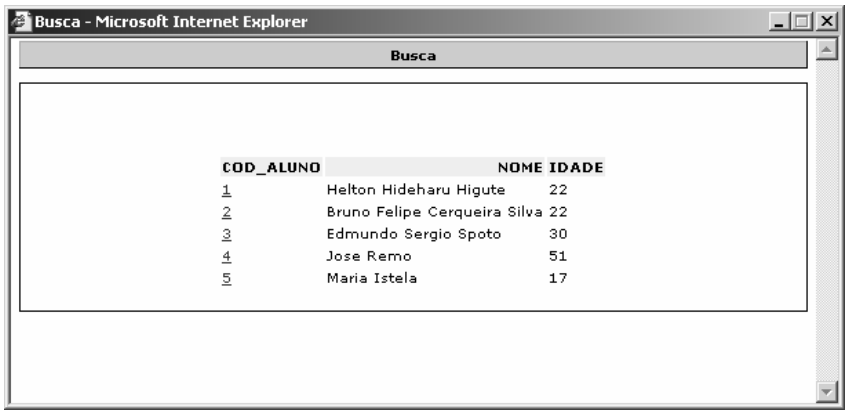

Figura 3.24: Relatório gerado a partir do *link* ao lado do campo que é chave estrangeira (Figura 3.23).

CONCLUSÃO

Cresceu muito o número de pesquisas baseadas em regras de negócio. Dentro destas pesquisas são encontradas formas muito distintas de se modelar e tratar as regras de negócio. Este trabalho constrói uma idéia de modelagem de regras de negócio simplificado, aplicando-se esforços para encontrar facilidades para os conhecedores de negócio e desenvolvedores.

Conclui-se que uma ferramenta deste porte é muito bem vinda, pois facilita o desenvolvimento de aplicações por pessoas que não possuem muito conhecimento em programação, e por ser completa, torna-se muito útil no processo de desenvolvimento de aplicações para uma empresa.

Problemas Encontrados

Contudo, após o desenvolvimento da ferramenta foram encontrados diversos problemas na forma com que são modeladas e processadas as regras de negócio. Em primeiro lugar, a ferramenta efetua as validações de regras de negócio no conjunto de dados do banco de dados do cliente, isso faz com que no caso de uma entrada de dados do usuário na aplicação gerada, os dados sejam comparados com os que existem no banco de dados. Percebeu-se a necessidade de fazer com que as regras de negócio cadastradas pelo usuário

sejam avaliadas no nível da aplicação, e não do banco de dados, devido à existência de regras de negócio que não necessitem de dados já pré-existentes.

Outro problema encontrado foi quanto à forma fornecida para o usuário modelar a regra de negócio necessitada. Disponibilizou-se um cadastro de regras de negócio, onde o usuário consegue fornecer apenas uma cláusula para efetuar o cadastro da regra de negócio. Para a modelagem percebeu-se a necessidade de disponibilizar uma forma de cadastro de regra de negócio, onde o usuário consiga informar diversos conjuntos de restrições para a regra de negócio, e também a possibilidade de se adicionar diversas regras para o caso de uso a ser cadastrado.

Trabalhos Futuros

Encontram-se alguns problemas na modelagem e no desenvolvimento da aplicação, contudo, aqui são propostos meios de se melhorar a ferramenta no futuro.

Na fase em que o usuário seleciona o banco de dados para iniciar a construção do caso de uso de projeto, observou-se a necessidade de se remover a seleção da tabela e permitir que esta seja selecionada na aba de seleção dos campos, para que possa ser selecionada mais de uma tabela para o caso de uso de projeto.

Também há necessidade de se contruir meios para que o usuário consiga modelar as regras de negócio, de forma a poder cadastrar diversas restrições para uma mesma regra. Com isso, torna-se possível efetuar diversos tipos de operações lógicas para a regra, que no momento só utiliza uma operação lógica.

Quanto às validações de regras de negócio, apenas no nível do banco de dados é necessário uma reformulação para que no momento da aplicação das regras, as mesmas sejam avaliadas em nível de aplicação, onde são mais necessárias.

Objetivos Atingidos

No trabalho apresentado é visto que a aplicação desenvolvida consegue recuperar infomações dos metadados, aplicar regras de negócio cadastrados, no nível do banco de dados do cliente e gerar o formulário ou relatório para aplicação que foi cadastrada .

No geral, a construção de programas baseados em casos de usos e regras de negócio pode ser um mecanismo muito útil para diferentes tipos de empresas que possuem diferentes regras de negócios. Com a ferramenta BRD Tool facilita-se muito o reuso e o aproveitamento de técnicas já construídas para outras empresas.

No entanto, para o desenvolvimento desta ferramenta é necessário ter um conhecimento muito grande de um abiente de negócios, e percebe-se que a forma de se descrever um negócio deveria ser bem mais complexa e abrangente.

BIBLIOGRAFIA

BOS, Bert; LIE, Hakon Wium; LILLEY, Chris; JACOBS, Ian**. Cascading Style Sheets Level 2, CSS Especification.** Disponível em <http://www.w3.org/TR/1998/REC-CSS2-19980512/>. Acesso em: 20 out 2005.

DALLAVALLE, S.I.; CAZARINI E.W. **Regras do Negócio: fator chave de sucesso no processo de desenvolvimento de sistemas de informação**. ENEGEP – Encontro Nacional de Engenharia de Produção, 2000.

DATE, C. J. **Introdução a sistemas de banco de dados.** Rio de Janeiro: Campus, 1991.

DEITEL, H. M.; DEITEL P. J. **Java, Como Programar**. Porto Alegre: Bookman, 2003.

ELMASRI, Ramez; NAVATHE, Shamkant B. Sistema de banco de dados – Fundamentos e aplicações. São Paulo: Addison Wesley, 2005.

FIELDING, R.; IRVINE, U. C.; GETTYS, J.; MOGUL, J.; FRYSTYK, H.; MASINTER, L.; LEACH, P.; BERNERS-LEE, T. **Hypertext Transfer Protocol – HTTP/1.1.** Disponível em: <http://www.w3.org/protocols/rfc2616/rfc2616.html>. Acesso em: 30 out. 2005.

KURNIAWAN, Budi. **Java para a web com servlets, JSP e EJB.** Rio de Janeiro: Editora Ciência Moderna, 2002.

KURNIAWAN, Budi. **Programando em JavaServer Faces.** Rio de Janeiro: Editora Ciência Moderna, 2004.

LARMAN, Craig. **Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao Processo Unificado** – 2ª. Edição – Porto Alegre: Bookman, 2004.

PÁDUA, Silvia Inês Dallavalle; PÁDUA, Fabiana Serralha Miranda; CAZARINI, Edson Walmir; INAMASU, Ricardo Yassushi. **Modelo de atores e recursos, seu potencial e**

importância no modelo organizacional. Disponível em <http://www.simpep.feb.unesp.br/anais10/gestaodainformacao/arq13.PDF>. Acesso em: 17 out 2005.

REESE, George. **JDBC e Java.** São Paulo: Berkeley Brasil, 2001.

SANTOS, Cláudio**. Qualidade das Regras de Negócios com uso da Lógica Fuzzy.** Disponível em <www.alats.org.br/eventos/ICongressoALATS/Claudio_Santos.pdf>. Acesso em: 25 mar 2005.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. **Sistema de banco de dados.** São Paulo: MAKRON Books, 1999.

SULLINS, Benjamin G.; WHIPPLE, Mark B. **EJB: Livro de Receitas.** Rio de Janeiro: Editora Ciencia Moderna, 2004.

APÊNDICE A – CÓDIGO FONTE DA FERRAMENTA BRD TOOL

A.1 - userdb.jsp

```
<%@ page contentType="text/html" language="java" import="java.sql.*"
import="javax.servlet.jsp.PageContext"%>
<jsp:useBean id="conUserDB" class="conexaoSuperBaan.ConexaoUserDB"/>
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/shtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pt" lang="pt">
<head>
<title>Configuração do Banco de Dados do Usuário</title>
<style type="text/css">
<!--
@import url("common/common.css");
-->
</style>
</head>
<body>
<div id="pagetitle">Configuração do BD Cliente</div>
<div id="main">
      <br />
      <div id="browse">
            <a href="configureuserdb.jsp?cod_db=">Incluir Banco de
Dados</a><br /><br />
            <table class="tablebrowse">
            <tr>
                  <th>Código</th>
                  <th>SGBD</th>
                  <th>Servidor</th>
                  <th>Nome do Banco</th>
                  <th>Ação</th>
            </tr>
            <%
          try {
                  int color = 0;
                  conUserDB.setConsultaManagerCliDB();
                  ResultSet temp =
                        conUserDB.getResultado();
                              temp.next();
                              do{
                         %>
      <td><%out.print(temp.getString("cod_dbcli"));%></td>
      <td><%out.print(temp.getString("sgbd_cli"));%></td>
      <td><%out.print(temp.getString("servidor"));%></td>
      <td><%out.print(temp.getString("nome_banco"));%></td>
     <td>
      <input type="button"
      onClick="document.location.href='configureuserdb.jsp?cod_db=<%out.pri
nt(temp.getString("cod_dbcli")); %>'" value="Ver" class="actbutton" />
<!-- <input type="button" value="Testar" class="actbutton" />  -->
                                                                  </td>
</tr>
```

```
      <%
      }while (temp.next());
      }catch (Exception e) {e.printStackTrace();}
      %>
                  </table>
            </div>
            <br />
      </div>
</body>
</html>
```

A.2 - user.jsp

```
<%@ page contentType="text/html" language="java" import="java.sql.*"
import="javax.servlet.jsp.PageContext"%>
<jsp:useBean id="con" class="conexaoSuperBaan.Conexao"/>
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/shtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pt" lang="pt">
<head>
<title>Cadastro de Usuário do Sistema</title>
<style type="text/css">
<!--
@import url("common/common.css");
-->
</style>
</head>
<body>
      <div id="pagetitle">Cadastro de Usuário</div>
      <div id="main">
            <br />
            <div id="browse">
                  <a href="manageruser.jsp?login=&loginField
disabled">Criar Usuário</a><br /><br />
                  <table class="tablebrowse">
                        <tr>
                              <th>Nome</th>
                              <th>E-Mail</th>
                              <th>Tipo</th>
                              <th>Ação</th>
                        </tr>
                              <%
                           try {
                          int color = 0;
                               con.setConsultaManagerUser();
                              ResultSet temp =
                                    con.getResultado();
                              temp.next();
                            do{
                         %>
            <td><%out.print(temp.getString("nome"));%></td>
            <td><%out.print(temp.getString("email"));%></td>
            <td><%out.print(temp.getString("tipo"));%></td>
            <td>
            <input type="button"
      onClick="document.location.href='manageruser.jsp?login=<%out.print(te
mp.getString("login"));%>'" value="Ver" class="actbutton" />
            </td>
            </tr>
                              <%
```

```
                        }while (temp.next());
                    }catch (Exception e) {e.printStackTrace();}
                %>
                    </table>
            </div>
            <br />
      </div>
</body>
</html>
```

A.3 - tabbar.jsp

```
<script language="javascript">
<!--
function sendTo(page, fromClick) {

      if(fromClick == 1) {
            changeuc.acao.click();
      }
      else
            location.href =  page;

}
-->
</script>
<%
String act = request.getRequestURI();
%>
            <div class="tabbar">
                  <ul>
                        <li><a href="#"
                        <%
                        if(act.indexOf("changeuc.jsp") != -1) {
                              out.print("class=\"tabact\"");
      out.print("onClick=\"sendTo('changeuc.jsp',1);\"");
                        } else {
      out.print("onClick=\"sendTo('changeuc.jsp',0);\"");
                  }
                        %>
                        >Caso de Uso</a></li>
                        <li><a href="changedb2.jsp"
                  <%
                  if(act.indexOf("changedb2.jsp") != -1) {
                        out.print("class=\"tabact\"");
      out.print("onClick=\"sendTo('changedb2.jsp',1);\"");
                  } else {
      out.print("onClick=\"sendTo('changedb2.jsp',0);\"");
      }
                  %>
                        >Banco de Dados</a></li>
                        <li><a href="changefields.jsp"
                        <%
            if(act.indexOf("changefields.jsp") != -1) {
                  out.print("class=\"tabact\"");
      out.print("onClick=\"sendTo('changefields.jsp',1);\"");
      } else {
      out.print("onClick=\"sendTo('changefields.jsp',0);\"");    }
                        %>
                        >Campos</a></li>
```

```
                        <li><a href="changerules.jsp"
                        <%
                              if(act.indexOf("changerules.jsp") != -1) {
            out.print("class=\"tabact\"");
      out.print("onClick=\"sendTo('changerules.jsp',1);\"");
      } else {
      out.print("onClick=\"sendTo('changerules.jsp',0);\"");
                        }
                        %>
                        >Regras de Negócio</a></li>
                        <li><a href="saveinfo.jsp"
                        <%
                        if(act.indexOf("saveinfo.jsp") != -1) {
                              out.print("class=\"tabact\"");
      out.print("onClick=\"sendTo('saveinfo.jsp',1);\"");
                        } else {
      out.print("onClick=\"sendTo('saveinfo.jsp',0);\"");
                  }
                              %>
                              >Concluir</a></li>
                        </ul>
            </div>
```

A.4 - saveinfo.jsp

```
<%@ page contentType="text/html" language="java" import="java.sql.*"%>
<jsp:useBean id="connDB"
class="conexaoSuperBaan.ConexaoBusinessRulesUseCase"/>
<%
      java.util.ArrayList UseCase = new java.util.ArrayList();
      UseCase = (java.util.ArrayList) session.getAttribute("USECASE");

      String acao = request.getParameter("acao");
      String prjCode = (String)UseCase.get(7);
      if(acao != null) {
            connDB.startInsertAllData(UseCase);

            for(int i = 0; i < 20; i++) {
                  if(i == 5 || i == 6) {
                        UseCase.set(i, new java.util.ArrayList());
                  } else {
                        UseCase.set(i, "");
                  }
            }
            session.setAttribute("USECASE",UseCase);
      response.sendRedirect("changeuc.jsp?prjcode="+prjCode);
      }

%>
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/shtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pt" lang="pt">
<head>
<title>Gravar Informações</title>
<style type="text/css">
<!--
@import url("common/common.css");
```

```
@import url("common/tabs.css");
-->
</style>
</head>
<body>
      <div id="pagetitle">Gravar Informações</div>
      <div id="main">
            <br /><br />
<%
      pageContext.include("tabbar.jsp");
%>
            <br /><br />
            <div id="form">
             <table class="tableform">
                <tr>
                  <td><form name="saveinfo" id="saveinfo">
                        <fieldset>
                        <legend>Gravar Informações</legend>
                        <table class="tableform">
                        <tr>
                       <td width="181"> </td>
                        </tr>
                     <tr>
                   <td class="descfield">Caso de Uso:</td>
                   <td width="201"><%= UseCase.get(1) %></td>
                     </tr>
                  <tr> </tr>
                 <tr>
              <td class="descfield">Banco de Dados:</td>
              <td><%= UseCase.get(3) %></td>
           </tr>
             <tr>
              <td class="descfield">Tipo de Componente:</td>
                 <td><%
                        String type = (String)UseCase.get(0);
                        if(type.equals("1"))
                              out.print("Manutenção de Registro");
                        else
                              out.print("Relatório");
                    %></td>
              </tr>
         <tr>
         <td class="descfield">Descrição:</td>
          <td><%= UseCase.get(2) %></td>
            </tr>
               <tr>
                 <td> </td>
                 <td> </td>
                 </tr>
              <tr>
                <td class="descfield">Campos:</td>
                <td> </td>
              </tr>
          <tr>
      <td valign="top" colspan="2"><table class="tableform" border="0">
          <tr>
            <td>Ordem</td>
            <td>Campo</td>
            <td>Descrição</td>
          </tr>
      <%
```

```
java.util.ArrayList selectedFields =
                  (java.util.ArrayList)UseCase.get(5);
      int count = selectedFields.size();
      while(count != 0) {
      count--;
      %>
        <tr>
          <td><input type="text" name="ord" id="ord" value="<%= count %>"
size="3" /></td>
          <td><input type="text" name="field<%= count %>" id="field<%=
count %>" value="<%= (String)selectedFields.get(count) %>" size="30"/></td>
          <td><input type="text" name="desc<%= count %>" id="desc<%= count
%>" value="<%= (String)selectedFields.get(count) %>" size="30" /></td>
        </tr>
      <%
      }
      %>
            </table></td>
           </tr>
               <tr>
               <td colspan="2"> </td>
               </tr>
               <tr>
                <td class="descfield">Regras:</td>
                <td> </td>
               </tr>
               <tr>
                <td colspan="2">
      <%
      String temp = ((java.util.ArrayList)UseCase.get(6)).toString();
      temp = temp.replaceAll("],","<br><br>");
      temp = temp.replace("[","");
      temp = temp.replace("]","");
      out.print(temp);
      %>
            </td>
             </tr>
               <tr>
                 <td colspan="2"> </td>
               </tr>
               <tr>
                <td colspan="2" class="tdline">
               </tr>
               <tr>
                <td colspan="2"> </td>
               </tr>
               <tr>
                <td colspan="2" class="fieldbuttons">
                      <input type="submit" name="acao" value="Gravar"
class="buttonform" />
                </td>
                  </tr>
          </table>
      </fieldset>
    </form>
      </td>
                      </tr>
                 </table>
            </div>
            <br />
      </div>
```

```
</body>
</html>
```

A.5 - ruledetail.jsp

```
<%@ page contentType="text/html" language="java" import="java.sql.*"%>
<jsp:useBean id="selDB" class="conexaoSuperBaan.ConexaoUserDB"/>
<jsp:useBean id="selDBField" class="conexaoSuperBaan.ChangeFieldDB"/>
<jsp:useBean id="selRuleBusiness"
class="conexaoSuperBaan.ConexaoBusinessRulesUseCase"/>
<%
    String acao = request.getParameter("acao");
    String mensagem="";
    java.util.ArrayList UseCase = new java.util.ArrayList();
      java.util.ArrayList tables = new java.util.ArrayList();
    java.util.ArrayList fieldsForTable1 = new
                              java.util.ArrayList();
    java.util.ArrayList fieldsForTable2 = new
                              java.util.ArrayList();
      UseCase = (java.util.ArrayList)
                        session.getAttribute("USECASE");
      ResultSet temp2;
      ResultSet temp;
      ResultSetMetaData fields;
    String table1= request.getParameter("table1");
    String table2 = request.getParameter("table2");
    String name = request.getParameter("name");
    String value = request.getParameter("value");
    String compfun = request.getParameter("compfunction");
      String actiontrue = request.getParameter("actiontrue");
    String valuetrue = request.getParameter("valuetrue");
    String actionfalse=request.getParameter("actionfalse");
    String valuefalse = request.getParameter("valuefalse");
    String fields2Selected =
            request.getParameter("fieldsfromtable2");
    String fields1Selected =
            request.getParameter("fieldsfromtable1");
   String digitedField =
                  request.getParameter("fielddigited");
                  if(name == null)
                        name="";
                  if(value == null)
                        value="";
                  if(compfun == null)
                        compfun="";
                  if(actiontrue == null)
                        actiontrue = "";
                  if(valuetrue == null)
                        valuetrue = "";
                  if(actionfalse == null)
                        actionfalse = "";
                  if(valuefalse == null)
                        valuefalse = "";
      UseCase = (java.util.ArrayList)
                        session.getAttribute("USECASE");
      selDBField.setBancoDados((String)UseCase.get(3));
      try{
            boolean test = selDBField.startChangeBanco();
            if (test) {
```

```
                  temp2 = selDBField.getResultado();
            do{
                  tables.add(temp2.getString(3));
                  }while (temp2.next());
            }
            }catch(Exception e){mensagem = e.getMessage();}
            if(table1 != null && table1.equals("") == false)
      {
              selDBField.setBanco_DB(table1);
              fields = selDBField.startChangeFields();
              int numberOfColumns = fields.getColumnCount();
                for(int i =1; i <= numberOfColumns; i++) {
                  fieldsForTable1.add(fields.getColumnName(i));
                }
            }
            if(table2 != null && table2.equals("") == false)
      {
              selDBField.setBanco_DB(table2);
              fields = selDBField.startChangeFields();
              int numberOfColumns = fields.getColumnCount();
              for(int i =1; i <= numberOfColumns; i++) {
                  fieldsForTable2.add(fields.getColumnName(i));
             }
            }

      if(acao == null) {
      %>
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/shtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pt" lang="pt">
<head>
<title>Detalhamento de regras</title>
<style type="text/css">
<!--
@import url("common/common.css");
@import url("common/tabs.css");
-->
</style>
</head>
<body>
      <div id="pagetitle">Defini&ccedil;&atilde;o das Regras </div>
<div id="main">
      <br /><br />
      <div id="form">
            <table class="tableform">
                  <tr>
                  <td>
                  <form name="ruledetail" id="ruledetail" method="post">
                  <fieldset>
                   <legend>Defini&ccedil;&atilde;o de Regras de
                   Neg&oacute;cio</legend>
                  <table class="tableform">
                  <tr>
                    <td> 
              </td>
            </tr>
            <tr>
      <td valign="top">
      <table class="tableform" border="0">
<tr>
```

```
<td>Nome</td>
<td colspan="2"><input type="text" name="name" value="<%= name %>"/></td>
</tr>
<tr>
<td>Tabela</td>
<td colspan="2"><select name="table1" id="table1"
onchange="document.ruledetail.submit()">
      <option value="" selected></option>
 <%
      int count = tables.size();
       String table1Selected =request.getParameter("table1");
      String selected = "";
      while(count != 0) {
        count--;
        if(table1Selected != null &&
table1Selected.equals((String)tables.get(count)))
            selected = "selected";
        else
             selected = "";                                          %>
      <option value="<%out.print((String)tables.get(count));%>" <%=
selected %>><%out.print((String)tables.get(count));%></option>
      <%
      }
      %>
</select></td>
</tr>
<tr>
<td>Campo</td>
<td colspan="2"><select name="fieldsfromtable1">
value="" selected></option>
<%
      count = fieldsForTable1.size();
      selected = "";
      while(count != 0) {
      count--;
      if(fields1Selected != null &&
fields1Selected.equals((String)fieldsForTable1.get(count)))
      selected = "selected";
      else
            selected = "";
      %>
      <option value="<%out.print((String)fieldsForTable1.get(count));%>"
<%= selected %>><%out.print((String)fieldsForTable1.get(count));%></option>
      <%
 }
%>
</select></td>
</tr>
<tr>
<td>Compara&ccedil;&atilde;o / Fun&ccedil;&atilde;o</td>
<td><select name="compfunction">
      <option value="" <% if(compfun.equals(""))     out.print("selected");
%>></option>
     <option value="=" <% if(compfun.equals("="))    out.print("selected");
%>>=</option>
     <option value="&gt;" <% if(compfun.equals(">")) out.print("selected");
%>>&gt;</option>
      <option value="&lt;" <% if(compfun.equals("<"))
      out.print("selected"); %>>&lt;</option>
     <option value="!=" <% if(compfun.equals("!="))  out.print("selected");
%>>!=</option>
```

```
     <option value="MAX" <% if(compfun.equals("MAX"))
      out.print("selected"); %>>MAX</option>
     <option value="COUNT" <% if(compfun.equals("COUNT"))
      out.print("selected"); %>>COUNT</option>
     <option value="AVG" <% if(compfun.equals("AVG"))
      out.print("selected"); %>>AVG</option>
</select></td>
<td>Campo Digitado</td>
</tr>
<tr>
<td>Valor</td>
<td><input name="value" type="text" size="10" value="<%= value %>"/></td>
<td><select name="fielddigited">
      <option value="" selected></option>
      <%
      count = fieldsForTable1.size();
      selected = "";
      while(count != 0) {
      count--;
      if(digitedField != null &&
((String)fieldsForTable1.get(count)).equals(digitedField))
            selected = "selected";
      else
            selected = "";
      %>
      <option value="<%out.print((String)fieldsForTable1.get(count));%>"
<%= selected %>><%out.print((String)fieldsForTable1.get(count));%></option>
   <%
      }
      %>
</select></td>
</tr>
<tr>
<td>Tabela</td>
<td colspan="2"><select name="table2" id="table2"
onchange="document.ruledetail.submit()">
      <option value="" selected></option>
      <%
      count = tables.size();
      String table2Selected = request.getParameter("table2");
      selected = "";
      while(count != 0) {
      count--;
      if(table2Selected != null &&
      table2Selected.equals((String)tables.get(count)))
            selected = "selected";
      else
            selected = "";
      %>
      <option value="<%out.print((String)tables.get(count));%>" <%=
selected %>><%out.print((String)tables.get(count));%></option>
      <%
      }
      %>
            </select>
              </td></tr>
                  <tr>
                    <td>Campo</td>
            <td colspan="2"><select name="fieldsfromtable2">
          <option value=""></option>
      <%
```

```
      count = fieldsForTable2.size();
      selected = "";
      while(count != 0) {
      count--;
            if(fields2Selected != null &&
fields2Selected.equals((String)fieldsForTable2.get(count)))
                  selected = "selected";
      else
                  selected = "";

      %>

      <option value="<%out.print((String)fieldsForTable2.get(count));%>"
<%= selected %>><%out.print((String)fieldsForTable2.get(count));%></option>
      <%
      }
      %>
</select></td></tr>
<tr>
<td>Se</td>
<td colspan="2">Verdadeiro</td>
</tr>
<tr>
<td>A&ccedil;&atilde;o</td>
<td colspan="2"><select name="actiontrue">
      <option value="" <% if(actiontrue.equals(""))  out.print("selected");
%>></option>
     <option value="Mensagem" <% if(actiontrue.equals("Mensagem"))
out.print("selected"); %>>Menssagem</option>
      <%
      try {
selRuleBusiness.selectCasoUsoCadastrado(Integer.parseInt((String)UseCase.ge
t(7)));
temp = selRuleBusiness.getResultado();
          do{
                                                %>
      <option value="<%out.print(temp.getString("cod_use_case_prj"));%>" <%
if(actiontrue.equals(temp.getString("cod_use_case_prj")))
out.print("selected"); %>><%out.print(temp.getString("nome"));%></option>
            <%
            }while (temp.next());
      }catch (Exception e) {e.printStackTrace();}
                                  %>
</select></td></tr>
<tr>
<td>Valor A&ccedil;&atilde;o </td>
<td colspan="2"><input type="text" name="valuetrue" value="<%= valuetrue
%>" /></td></tr>
<tr>
<td>Se</td>
<td colspan="2">Falso</td>
                                                                        </tr>
<tr>
<td>A&ccedil;&atilde;o</td>
<td colspan="2"><select name="actionfalse">
<option value="" <% if(actionfalse.equals("")) out.print("selected");
%>></option>
<option value="Mensagem" <% if(actionfalse.equals("Mensagem"))
out.print("selected"); %>>Menssagem</option>
<%
      try {
```

```
selRuleBusiness.selectCasoUsoCadastrado(Integer.parseInt((String)UseCase.ge
t(7)));
temp = selRuleBusiness.getResultado();
      do{
          %>
                  <option
value="<%out.print(temp.getString("cod_use_case_prj"));%>" <%

      if(actionfalse.equals(temp.getString("cod_use_case_prj")))
out.print("selected"); %>><%out.print(temp.getString("nome"));%></option>
      <%
      }while (temp.next());
     }catch (Exception e) {e.printStackTrace();}
     %>
</select></td></tr>
<tr>
<td>Valor A&ccedil;&atilde;o </td>
<td colspan="2"><input type="text" name="valuefalse" value="<%= valuefalse
%>"/></td>
</tr>
      </table>
            </td>
            </tr>
          <tr>
          <td> </td>
          </tr>
          <tr>
          <td colspan="2" class="tdline"></td>
          </tr>
          <tr>
          <td> </td>
            </tr>
            <tr>
        <td>
<span class="fieldbuttons">
<input type="submit" value="Ok" name="acao" class="buttonform" />
                                                        </span></td>
                                                          </tr>
                                                    </table>
                                              </fieldset>
                                        </form>
                                  </td>
                            </tr>
                      </table>
            </div>
            <br />
      </div>
</body>
</html>
<%
} else {

      java.util.ArrayList ruleDetail = new java.util.ArrayList();

      if(name.equals("") == false) {
            ruleDetail.add(name);
            ruleDetail.add(table1);
            ruleDetail.add(fields1Selected);
            ruleDetail.add(table2);
            ruleDetail.add(fields2Selected);
            ruleDetail.add(value);
```

```
            ruleDetail.add(compfun);
            ruleDetail.add(actiontrue);
            ruleDetail.add(valuetrue);
            ruleDetail.add(actionfalse);
            ruleDetail.add(valuefalse);
            ruleDetail.add(digitedField);
            ((java.util.ArrayList)UseCase.get(6)).add(ruleDetail);
      }

%>
<Script language="javascript">
<!--
      window.opener.location.href = "changerules.jsp";
      window.close();
-->
</Script>


<%
}
%>
```

A.6 - project.jsp

```
<%@ page contentType="text/html" language="java" import="java.sql.*"
import="javax.servlet.jsp.PageContext"%>
<jsp:useBean id="con" class="conexaoSuperBaan.ConexaoPrj"/>
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/shtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pt" lang="pt">
<head>
<title>Cadastro de Projeto</title>
<style type="text/css">
<!--
@import url("common/common.css");
-->
</style>
</head>
<body>
      <div id="pagetitle">Cadastro de Projeto</div>
      <div id="main">
            <br />
            <div id="browse">
                  <a href="managerprj.jsp?codigo=">Criar Projeto</a><br
/><br />
                  <table class="tablebrowse">
                        <tr>
                              <th>Código</th>
                              <th>Descrição</th>
                              <th>Ação</th>
                        </tr>
                              <%
                                 try {
                                   int color = 0;
                                   con.setConsultaManagerPrj();
                                   ResultSet temp = con.getResultado();
                                   temp.next();
                                    do{
                                       %>
                                     <tr>
```

```
<td><%out.print(temp.getString("cod_prj"));%></td>
<td><%out.print(temp.getString("descricao_prj"));%></td>
<td>
      <input type="button"
      onClick="document.location.href='managerprj.jsp?codigo=<%out.print
      (temp.getString("cod_prj"));%>'" value="Ver" class="actbutton" />
</td>
                                                            </tr>
                                                    <%
                                                    }while (temp.next());
                                            }catch (Exception e) {
                                                    e.printStackTrace();
                                            }
                                    %>
                  </table>
            </div>
            <br />
      </div>
</body>
</html>
```

A.7 - menu.jsp

```
<%@ page contentType="text/html" language="java" import="java.sql.*"%>
<jsp:useBean id="conUser" class="conexaoSuperBaan.Conexao"/>

<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/shtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pt" lang="pt">
<head>
<title>Menu</title>
<style type="text/css">
<!--
@import url("common/menu.css");
-->
</style>
<script type="text/javascript" src="./menu/TreeMenu.js"></script>
<script type="text/javascript" src="./jscripts/functions.js"></script>
</head>
<body class="treeMenu">
<script type="text/javascript">
      objTreeMenu_1 = new TreeMenu("menu/images", "objTreeMenu_1", "_self",
"treeMenuDefault", true, false);

      newNode = objTreeMenu_1.addItem(new TreeNode('Sistema', 'folder.gif',
'', true, true, '', 'detail', 'folder-expanded.gif', ''));
      newNode_1 = newNode.addItem(new TreeNode('Manutenção', 'folder.gif',
'', true, true, '', 'detail', 'folder-expanded.gif', ''));
    newNode_1_1 = newNode_1.addItem(new TreeNode('Usuários',
'../../images/tables.png', 'javascript:goPage(' + "'user.jsp'" + ')',
false, true, '', '_self', '', ''));
      newNode_1_2 = newNode_1.addItem(new TreeNode('Projetos',
'../../images/tables.png', 'javascript:goPage(' + "'project.jsp'" + ')',
false, true, '', '_self', '', ''));

      newNode_1_5 = newNode_1.addItem(new TreeNode('Config. BD Cliente',
'../../images/tables.png', 'javascript:goPage(' + "'userdb.jsp'" + ')',
false, true, '', '_self', '', ''));
```

```
<!--  newNode_1_6 = newNode_1.addItem(new TreeNode('Config. BD Sistema',
'../../images/tables.png', 'javascript:goPage(' + "'configsystemdb.html'" +
')', false, true, '', '_self', '', '')); -->

      newNode2 = objTreeMenu_1.addItem(new TreeNode('Projetos',
'folder.gif', '', true, true, '', 'detail', 'folder-expanded.gif', ''));

<%
      int i=0;
      java.util.ArrayList Menu = new java.util.ArrayList();
      Menu = (java.util.ArrayList) session.getAttribute("MENU");
      for(i = 0; i < Menu.size(); i++) {
            java.util.ArrayList mnuProj = new java.util.ArrayList();
            mnuProj =(java.util.ArrayList) Menu.get(i);
%>
            newNode2_<%=i%> = newNode2.addItem(new
TreeNode('<%out.print((String)mnuProj.get(0));%>', 'folder.gif', '', true,
true, '', 'detail', 'folder-expanded.gif', ''));
            newNode2_<%=i%>_1 = newNode2_<%=i%>.addItem(new TreeNode('Caso
de Uso', '../../images/tables.png', 'javascript:goPage(' +
"'changeuc.jsp?prjcode=<%out.print((String)mnuProj.get(1));%>'" + ')',
false, true, '', '_self', 'folder-expanded.gif', ''));
<%
            ResultSet temp;

      if(conUser.selectUseCase(Integer.parseInt((String)mnuProj.get(1)))) {
            int count = 2;
            temp = conUser.getResultado();
              do{
%>
                  newNode2_<%=i%>_<%=count%> = newNode2_<%=i%>.addItem(new
TreeNode('<%= (String)temp.getString(2) %>', '../../images/usecase.png',
'javascript:goPage(' + "'<%= (String)temp.getString(1) %>'" + ')', false,
true, '', '_self', 'folder-expanded.gif', ''));
<%
              }while (temp.next());
            }
      }
%>
      objTreeMenu_1.drawMenu();
      objTreeMenu_1.writeOutput();
      objTreeMenu_1.resetBranches();
</script>

</body>
</html>
```

A.8 - manageruser.jsp

```
<%@ page contentType="text/html" language="java" import="java.sql.*"%>
<jsp:useBean id="con" class="conexaoSuperBaan.Conexao"/>
<%
String login = request.getParameter("login");
String senha = request.getParameter("senha");
String repSenha = request.getParameter("repSenha");
String nome = request.getParameter("nome");
String sobrenome = request.getParameter("sobrenome");
String email = request.getParameter("email");
String tipo = request.getParameter("tipo");
```

```
String checkAdmin = "";
String checkComum = "";
String disable = "";
String mensagem = "";
String acao=request.getParameter("acao");
if(acao != null) {
       if(acao.equals("Inserir")) {
                  try {
      con.setLogin(request.getParameter("loginField"));
      con.setSenha(request.getParameter("senhaField"));
      con.setNome(request.getParameter("nomeField"));
      con.setSobrenome(request.getParameter("sobrenomeField"));
      con.setEmail(request.getParameter("emailField"));
      con.setTipo(request.getParameter("radiobutton"));
                              con.insertDataManagerUser();
                  }catch (Exception e) {
                              e.printStackTrace();
                              mensagem = e.getMessage();
                  }
}
      if(acao.equals("Alterar")) {
                  try {
      con.setLogin(request.getParameter("loginField"));
      con.setSenha(request.getParameter("senhaField"));
      con.setNome(request.getParameter("nomeField"));
      con.setSobrenome(request.getParameter("sobrenomeField"));
      con.setEmail(request.getParameter("emailField"));
      con.setTipo(request.getParameter("radiobutton"));
                              con.updateDataManagerUser();
                              response.sendRedirect("user.jsp");
                  }catch (Exception e) {
                        e.printStackTrace();
                        mensagem = e.getMessage();
                  }

      }
      if(acao.equals("Excluir")) {
                  try {
      con.setLogin(request.getParameter("loginField"));
                              con.deleteDataManagerUser();
                              response.sendRedirect("user.jsp");

                  }catch (Exception e) {
                              e.printStackTrace();
                              mensagem = e.getMessage();
                  }
      }

}
if(request.getParameter("login").equals("")== false) {
            try{
              con.setLogin(request.getParameter("login"));
              boolean test = con.selectDataManagerUser();

                        if (test)
                        {

                              ResultSet temp = con.getResultado();
                              senha = temp.getString("senha");
                              nome = temp.getString("nome");
                              sobrenome = temp.getString("sobrenome");
```

```
                              email = temp.getString("email");
                              disable = "disabled";
                              if(temp.getString("tipo").equals("Admin")){
                                    checkAdmin = "checked";}
                              else{
                                    checkComum= "checked";}
                        }
            }catch(Exception e){
                  e.printStackTrace();
                  mensagem = e.getMessage();
            }

}else {
                              login="";
                              senha = "";
                              nome = "";
                              sobrenome = "";
                              email = "";
                              disable = "enabled";
                              checkComum= "checked";
}

%>

<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/shtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pt" lang="pt">
<head>
<title>Cadastro de Usuário do Sistema</title>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<style type="text/css">
<!--
@import url("common/common.css");
-->
</style>
</head>


      <script language="JavaScript">
      function tecla(){
      if((event.keyCode >= 33)&&(event.keyCode <= 64)){
      event.returnValue = false;}
      }

      function valida() {
            var msgErro = "";
            var erro = false;


            if(document.getElementById("loginField").value == "") {
                  msgErro = msgErro + "Login obrigatório.\n";
                  erro = true;
            }
            if(document.getElementById("nomeField").value == "") {
                  msgErro = msgErro + "Nome obrigatório.\n";
                  erro = true;
            }
            if(document.getElementById("senhaField").value == "") {
                  msgErro = msgErro + "Senha obrigatório.\n";
```

```
                erro = true;
            }
            if(document.getElementById("emailField").value == "") {
                msgErro = msgErro + "E-mail obrigatório.\n";
                erro = true;
            }
            if(document.manageruser.senhaField.value !=
document.manageruser.repeatPassField.value) {
                msgErro = msgErro + "Senha não Confere.\n";

                erro = true;
            }


            if(erro == true) {
                alert(msgErro);
                return false;
            } else
                return true;

      }


      </script>

<body>
      <div id="pagetitle">Cadastro de Usuário</div>
      <div id="main">
            <br />
            <%= mensagem %>
            <div id="form">
                <table class="tableform">
                      <tr>
                            <td>
                                  <form name="manageruser" method="post"
onSubmit="return valida();">
                                        <fieldset>
                                              <legend>Dados do
Usuário</legend>
                                          <table class="tableform">
                                    <tr>
                                      <td> </td>
                                    </tr>
                                     <tr>
                                        <td class="descfield">Login:</td>
                                        <td><input name="loginField"
type="text" id="loginField" value="<%=login%>" /></td>
                                    </tr>
                                    <tr>
                                      <td class="descfield">Senha:</td>
                                      <td><input name="senhaField"
type="password" size="20" id="senhaField" value="<%=senha%>"/></td>
                                    </tr>
                                    <tr>
                                      <td class="descfield">Repetir
Senha:</td>
                                      <td><input name="repeatPassField"
type="password" id="repeatPassField" value=""<%=repSenha%>/></td>
                                    </tr>
                                    <tr>
                                        <td class="descfield">Nome:</td>
```

```
                                        <td><input name="nomeField"
type="text" size="45" id="nomeField" onkeypress="tecla()"
value="<%=nome%>"/></td>
                                    </tr>
                                    <tr>
                                        <td class="descfield">Sobrenome:</td>
                                        <td><input name="sobrenomeField"
type="text" size="45" id="sobrenomeField" onkeypress="tecla()"
value="<%=sobrenome%>"/></td>
                                    </tr>
                                                <tr>
                                        <td class="descfield">E-mail:</td>
                                        <td><input name="emailField"
type="text" size="35" id="emailField" value="<%=email%>"/></td>
                                    </tr>
                                    <tr>
                                      <td class="descfield">Tipo de
Usuário:</td>
                                      <td>
                                        <input name="radiobutton"
type="radio" value="Admin" class="radio" <%= checkAdmin %>/>Administrador
                                        <input name="radiobutton"
type="radio" value="Comum" class="radio" <%= checkComum %>/>Comum
                                                        </td>
                                    </tr>
                                    <tr>
                                      <td> </td>
                                    </tr>
                                    <tr>
                                      <td colspan="2" class="tdline"></td>
                                    </tr>
                                    <tr>
                                      <td> </td>
                                    </tr>
                                    <tr>
                                      <td colspan="2" class="fieldbuttons">
                                                          <input
type="submit"  name = "acao" value="Inserir" class="buttonform" <%= disable
%>/>
                                          <input type="submit"  name = "acao"
value="Alterar" class="buttonform" />
                                          <input type="submit"  name = "acao"
value="Excluir" class="buttonform" />
                                          <input type="button"
onClick="document.location.href='user.jsp'" name = "btnSairCadUsuario"
value="Sair" class="buttonform" />
                                      </td>
                                    </tr>
                                  </table>
                                           </fieldset>
                                      </form>
                                </td>
                            </tr>
                        </table>
                </div>
                <br />
        </div>
</body>
</html>
```

A.9 - managerprj.jsp

```
<%@ page contentType="text/html" language="java" import="java.sql.*"%>
<jsp:useBean id="conPrj" class="conexaoSuperBaan.ConexaoPrj"/>
<jsp:useBean id="conUser" class="conexaoSuperBaan.Conexao"/>
<jsp:useBean id="conUserPrj" class="conexaoSuperBaan.ConexaoUserPrj"/>

<%
String codigo = request.getParameter("codigo");
String descricao = request.getParameter("descricao");
String login = request.getParameter("login");

String disable="";
String mensagem = "";
String acao=request.getParameter("acao");
String[] destino;
if(acao != null) {
       if(acao.equals("Inserir")) {

                  try {
      conPrj.setCodigoPrj(Integer.parseInt(request.getParameter("codigoFiel
d")));

      conPrj.setDescricao(request.getParameter("descricaoField"));
                              conPrj.insertDataManagerPrj();

                              destino =
request.getParameterValues("destino");
                              for( int i = 0; i < destino.length; i++ ) {

      conUserPrj.setCodigoPrj(Integer.parseInt(request.getParameter("codigo
Field")));
                                    conUserPrj.setLogin(destino[i]);
                                    conUserPrj.insertDataManagerUserPrj();
                              }
                  }catch (Exception e) {
                              e.printStackTrace();
                              mensagem=e.getMessage();

                  }
      }
      if(acao.equals("Alterar")) {
                  try {
      conPrj.setCodigoPrj(Integer.parseInt(request.getParameter("codigoFiel
d")));

      conPrj.setDescricao(request.getParameter("descricaoField"));
                              conPrj.updateDataManagerPrj();

      conUserPrj.setCodigoPrj(Integer.parseInt(request.getParameter("codigo
Field")));
                              conUserPrj.deleteDataManagerUserPrj();
                              destino =
request.getParameterValues("destino");
                              for( int i = 0; i < destino.length; i++ ) {
      conUserPrj.setCodigoPrj(Integer.parseInt(request.getParameter("codigo
Field")));
                                    conUserPrj.setLogin(destino[i]);
      conUserPrj.insertDataManagerUserPrj();
                              }
```

```
                              response.sendRedirect("project.jsp");
                  }catch (Exception e) {
                        e.printStackTrace();
                        mensagem=e.getMessage();
                  }

      }

      if(acao.equals("Excluir")) {
                  try {
      conPrj.setCodigoPrj(Integer.parseInt(request.getParameter("codigoFiel
d")));
                              conPrj.deleteDataManagerPrj();
                              response.sendRedirect("project.jsp");
                  }catch (Exception e) {
                              e.printStackTrace();
                              mensagem=e.getMessage();
                  }
      }

}

if(request.getParameter("codigo").equals("")== false) {

      try{
conPrj.setCodigoPrj(Integer.parseInt(request.getParameter("codigo")));
               boolean test = conPrj.selectDataManagerPrj();
                              if (test)
                              {
                                    ResultSet temp = conPrj.getResultado();
                                    descricao = temp.getString("descricao_prj");
                                    disable = "disabled";
                              }
            }catch(Exception e){
                  e.printStackTrace();
                  mensagem=e.getMessage();
            }
}else {
                              codigo = "";
                              descricao = "";
                              disable = "enabled";
}




%>

<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/shtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pt" lang="pt">
<head>
<title>Cadastro de Projeto</title>
<style type="text/css">
<!--
@import url("common/common.css");
-->
</style>
</head>
      <script language="JavaScript">
```

```
	function adiciona() {
			objUser = document.getElementById('origem');
			objUserSend = document.getElementById('destino');

			var tem = 0;
			for(var i = 0; i < objUser.length; i++) {
				if(objUser[i].selected) {
					tem = 0;
					for(var j = 0; j < objUserSend.length; j++) {
						if(objUserSend[j].value ==
objUser[i].value)
							tem = 1;
					}
					if(tem == 0) {
						objUserSend[objUserSend.length] = new
Option(objUser[i].value, objUser[i].value);
					}
				}
			}
	}

	function remove() {
		objUserSend = document.getElementById('destino');
		for(var i = 0; i < objUserSend.length; i++ ) {
			if(objUserSend[i].selected) {
				objUserSend[i] = null;
			}
		}
	}


	function valida() {
		var msgErro = "";
		var erro = false;


		if(document.getElementById("codigoField").value == "") {
			msgErro = msgErro + "Codigo obrigatório.\n";
			erro = true;
		}
		if(document.getElementById("descricaoField").value == "") {
			msgErro = msgErro + "Descrição obrigatória.\n";
			erro = true;
		}

		objSendUser = document.getElementById('destino');
		for( var i = 0; i < objSendUser.length; i++ ) {
				objSendUser[i].selected = true;
		}

		if(erro == true) {
			alert(msgErro);
			return false;
		} else
			return true;


	}
```

```
      </script>
<body>
      <div id="pagetitle">Cadastro de Projeto</div>
      <div id="main">
            <br />
            <%= mensagem %>
            <div id="form">
                  <table class="tableform">
                        <tr>
                              <td>
                                    <form name="project" method="post"
onSubmit="return valida();" >
                                          <fieldset>
                                                <legend>Dados do
Projeto</legend>
<table class="tableform">
      <tr><td colspan="2"> </td></tr>
      <tr>
      <td class="descfield">Código:</td>
      <td><input name="codigoField" type="text" id="codigoField"
value="<%=codigo%>" size="13"/></td>
      </tr>
      <tr>
      <td class="descfield">Descrição do Projeto:</td>
      <td><input name="descricaoField" type="text" id="descricaoField"
value="<%=descricao %>" size="40"/></td>
      </tr>
      <tr>
      <td class="descfield">Usuários:</td>
      <td> </td>
      </tr>
      <tr><td colspan="2"> </td></tr>
      <tr>
      <td colspan="2">
      <table class="tableform" align="center">
      <tr>
            <td>
                  <select name="origem" id="origem" size="10"
multiple="multiple">
                  <%
                  try{
                        conUser.setConsultaManagerUser();
                        ResultSet temp = conUser.getResultado();
                        temp.next();
                        do{
                        %>
                              <option value="<%out.print(temp.
      getString("login"));%>"><%out.print(temp.getString("login"));%></opti
on>                                                                     <%
                  }while (temp.next());
                  }catch (Exception e) {e.printStackTrace(); }
%>
                  </select>
            </td>
            <td>
                  <input type="button" value="&gt;&gt;"
onClick="adiciona();" /><br /><br />
                  <input type="button" value="&lt;&lt;" onClick="remove();"
/>
            </td>
            <td>
```

```
            <select name="destino" id="destino" size="10" multiple >
                  <%
                  if(request.getParameter("codigo").equals("")== false)
                  {
                    try{
                      conUserPrj.setCodigoPrj(Integer.parseInt(codigo));
                        conUserPrj.setSelectManagerUserPrj();
                        ResultSet temp = conUserPrj.getResultado();
                        temp.next();
                        do{
                              if(temp.getString("login").equals("") != true
) {
                        %>
<option
value="<%out.print(temp.getString("login"));%>"><%out.print(temp.getString(
"login"));%></option>

<%
            }         }while (temp.next());
                        }catch (Exception e) {e.printStackTrace();}
                  }
      %>
                  </select>
            </td>
      </tr>
      </table>
</td>
</tr>
      <tr><td colspan="2"> </td></tr>
      <tr><td colspan="2" class="tdline"></td></tr>
      <tr><td colspan="2"> </td></tr>
      <tr>
      <td colspan="2" class="fieldbuttons">
      <input type="submit" name = "acao" value="Inserir" class="buttonform"
<%=disable %>/>
      <input type="submit" name = "acao" value="Alterar"
class="buttonform"/>
      <input type="submit" name = "acao" value="Excluir"
class="buttonform"/>
      <input type="button" onClick="document.location.href='project.jsp'"
value="Sair" class="buttonform"/>
                                                                  </td>
                                                            </tr>
                                                      </table>
                                                </fieldset>
                                          </form>
                                    </td>
                              </tr>
                  </table>
            </div>
            <br />
      </div>
</body>
</html>
```

A.10 - main.jsp

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

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/shtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pt" lang="pt">
<head>
<title>Description Business Rules Tool</title>
<style type="text/css">
<!--
@import url("common/common.css");
-->
</style>
</head>
<body>
      <div id="pagetitle">Description Business Rules Tool</div>
      <div id="main">
            Bem Vindo ao DBR Tool
      </div>
</body>
</html>
```

A.11 - index.jsp

```
<%@ page contentType="text/html" language="java" import="java.sql.*"%>
<jsp:useBean id="conUser" class="conexaoSuperBaan.Conexao"/>

<%
String acao=request.getParameter("acao");
String mensagem = "";

if(acao != null) {

      if(acao.equals("Efetuar Login")) {

            conUser.setLogin(request.getParameter("login"));
            conUser.setSenha(request.getParameter("password"));
            if( conUser.startLogin() == true) {
                  request.getSession(true);
              if (conUser.selectProjects() == true ) {
                        java.util.ArrayList UseCase = new
java.util.ArrayList();
                        java.util.ArrayList Menu = new
java.util.ArrayList();
                        for(int i = 0; i < 20; i++) {
                              if(i == 5 || i == 6) {
                                    UseCase.add(i, new
java.util.ArrayList());
                              } else {
                                    UseCase.add(i, "");

                              }
                        }
                        session.setAttribute("USECASE",UseCase);
                        int i = 0;
                        ResultSet temp = conUser.getResultado();
                  do{
                              java.util.ArrayList mnuProj = new
java.util.ArrayList();
                              mnuProj.add(temp.getString("descricao_prj"));
                              mnuProj.add(temp.getString("cod_prj"));
                              Menu.add(i, mnuProj);
                  }while (temp.next());
```

```
                        session.setAttribute("MENU", Menu);
                        response.sendRedirect("home.jsp");
                  } else {
                        mensagem = "Nao existe nenhum projeto para este
usuario";
                  }
            } else {
                  mensagem = "Login ou senha invalidos";
            }


      }

}
%>


<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/shtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pt" lang="pt">
<head>
<title>Login</title>
<style type="text/css">
<!--
@import url("common/common.css");
-->
</style>
</head>
<body>
      <div id="pagetitle">Login de Sistema</div>
      <div id="main">
            <%= mensagem %>
            <br />
            <div id="form">
                  <table class="tableform">
                        <tr>
                              <td>
                                    <form name="login" method="post" >
                                          <fieldset>
                                                <legend>Login</legend>
                                                <table class="tableform">
                                                      <tr>
                                    <td class="descfield">Usuário:</td>
                                    <td><input type="text" name="login"
                                    id="login"/></td>
                                    </tr>
                                    <tr>
                                    <td class="descfield">Password:</td>
                                    <td><input type="text" name="password"
                                    id="password"/></td>
                                    </tr>

      <tr><td> </td></tr>
                        <tr><td colspan="2" class="tdline"></td></tr>

      <tr><td> </td></tr>
                        <tr>
                        <td colspan="2" class="fieldbuttons">

      <input type="submit" name='acao' value="Efetuar Login" />
```

```
      <input type="reset" value="Cancelar" />
                                                                    </td>
                                                                </tr>
                                                            </table>


                                            </fieldset>
                                        </form>
                                    </td>
                                </tr>
                            </table>
                        </div>
                        <br />
                    </div>
</body>
</html>
```

A.12 - home.jsp

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>Description Business Rules Tool</title>
</head>

<frameset rows="*" cols="220,*" framespacing="0" frameborder="NO"
border="0">
  <frame src="menu.jsp" name="leftFrame" scrolling="YES">
  <frame src="main.jsp" name="mainFrame">
</frameset>
<noframes><body>
</body></noframes>
</html>
```

A.13 - configureuserdb.jsp

```
<%@ page contentType="text/html" language="java" import="java.sql.*"%>
<jsp:useBean id="con" class="conexaoSuperBaan.ConexaoUserDB"/>
<%
String cod_db = request.getParameter("cod_db");
String servidor = request.getParameter("servidor");
String usermeta = request.getParameter("usermeta");
String senhameta = request.getParameter("senhameta");
String userdb = request.getParameter("userdb");
String senhadb = request.getParameter("senhadb");
String nomebanco = request.getParameter("nomebanco");
String sgbd = request.getParameter("sgbd");
String selOracle="";
String selPostgreSQL="";
String disable = "";
String mensagem = "";
```

```
String acao=request.getParameter("acao");



if(acao != null) {


       if(acao.equals("Inserir")) {

                  try {

      con.setServidor(request.getParameter("servidorField"));

      con.setSGBD(request.getParameter("sgbdField"));

      con.setUsuarioMeta(request.getParameter("userMetaField"));

      con.setSenhaMeta(request.getParameter("senhaMetaField"));

      con.setUsuarioDB(request.getParameter("userDBField"));
      con.setSenhaDB(request.getParameter("senhaDBField"));
      con.setNomeBanco(request.getParameter("nomeBancoField"));
      con.setSGBD(request.getParameter("sgbdField"));
                              con.insertDataManagerCliDB();
                  }catch (Exception e) {
                              e.printStackTrace();
                              mensagem = e.getMessage();
                  }
}
      if(acao.equals("Alterar")) {
                  try {
      con.setServidor(request.getParameter("servidorField"));
      con.setSGBD(request.getParameter("sgbdField"));
      con.setUsuarioMeta(request.getParameter("userMetaField"));
      con.setSenhaMeta(request.getParameter("senhaMetaField"));
      con.setUsuarioDB(request.getParameter("userDBField"));
      con.setSenhaDB(request.getParameter("senhaDBField"));
      con.setNomeBanco(request.getParameter("nomeBancoField"));
      con.setSGBD(request.getParameter("sgbdField"));
      con.setCodigoUserDB(Integer.parseInt(cod_db));
                              con.updateDataManagerCliDB();
                              response.sendRedirect("userdb.jsp");
                  }catch (Exception e) {
                        e.printStackTrace();
                        mensagem = e.getMessage();
                  }

      }

      if(acao.equals("Excluir")) {
                  try {

                              con.setCodigoUserDB(Integer.parseInt(cod_db));
                              con.deleteDataManagerCliDB();
                              response.sendRedirect("userdb.jsp");

                  }catch (Exception e) {
                              e.printStackTrace();
                              mensagem = e.getMessage();
                  }
      }
```

```
}

if(request.getParameter("cod_db").equals("")== false) {
            try{
                  con.setCodigoUserDB(Integer.parseInt(cod_db));
              boolean test = con.selectDataManagerCliDB();

                        if (test)
                        {

                              ResultSet temp = con.getResultado();
                              servidor = temp.getString("servidor");
                              usermeta = temp.getString("usuario_meta");
                              senhameta = temp.getString("senha_meta");
                              userdb = temp.getString("usuario_db");
                              senhadb = temp.getString("senha_db");
                              nomebanco = temp.getString("nome_banco");
                              disable = "disabled";

                              if
(temp.getString("sgbd_cli").equals("Oracle")){

                                    selOracle="selected";

                              }
                              if
(temp.getString("sgbd_cli").equals("PostgreSQL")){
                                    selPostgreSQL="selected";
                              }
                        }
            }catch(Exception e){
                  e.printStackTrace();
                  mensagem = e.getMessage();
            }

}else {
                              servidor = "";
                              usermeta = "";
                              senhameta = "";
                              userdb = "";
                              senhadb = "";
                              nomebanco = "";
                              sgbd = "";
                              disable = "enabled";

}

%>

<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/shtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pt" lang="pt">
<head>
<title>Configuração do Banco de Dados do Usuário</title>
<style type="text/css">
<!--
@import url("common/common.css");
-->
</style>
```

```
</head>
<body>
      <div id="pagetitle">Configuração do BD do Cliente</div>
      <div id="main">
            <br />
            <%= mensagem %>
            <div id="form">
                  <table class="tableform">
                        <tr>
                              <td>
                                    <form name="configureuserdb"
method="post">
                                          <fieldset>
      <legend>Configurações</legend>
                  <table class="tableform">
                        <tr><td colspan="2"> </td></tr>
                        <tr>
                        <td class="descfield">SGBD:</td>
                        <td>
                        <select name="sgbdField" id="sgbdField">
                        <option value=""  ></option>
                        <option value="Oracle" <%=selOracle
%>>Oracle</option>
                        <option value="PostgreSQL" <%=selPostgreSQL
%>>PostgreSQL</option>
                        </select>
                  </td>
            </tr>
      <tr>
<td class="descfield">Servidor:</td>
<td><input type="text" size="25" name="servidorField" id="servidorField"
value="<%=servidor%>" /></td>
</tr>
                  <tr>
                  <td class="descfield">Usuário Metadados:</td>
                  <td><input type="text" size="25" name="userMetaField"
id="userMetaField" value="<%=usermeta%>"/></td>
                  </tr>
                  <tr>
                  <td class="descfield">Senha Metadados:</td>
                  <td><input type="password" size="15"
name="senhaMetaField" id="senhaMetaField" value="<%=senhameta%>"/></td>
                  </tr>
                  <tr>
                  <td class="descfield">Usuário Banco de Dados:</td>
                  <td><input type="text" size="25" name="userDBField"
id="userDBField" value="<%=userdb%>"/></td>
                  </tr>
                  <tr>
                  <td class="descfield">Senha Banco de Dados:</td>
                  <td><input type="password" size="15" name="senhaDBField"
id="senhaDBField" value="<%=senhadb%>" /></td>
                  </tr>
                  <tr>
                  <td class="descfield">Nome do Banco:</td>
                  <td><input type="text" size="25" name="nomeBancoField"
id="nomeBancoField" value="<%=nomebanco%>"/></td>
                  </tr>
                  <tr><td colspan="2"> </td></tr>
                        <tr><td colspan="2" class="tdline"></td></tr>
                        <tr><td colspan="2"> </td></tr>
```

```
                  <tr>
                  <td colspan="2" class="fieldbuttons">
                  <input type="submit" name = "acao" value="Inserir"
class="buttonform" <%= disable %>/>
                  <input type="submit" name = "acao" value="Alterar"
class="buttonform"/>
                  <input type="submit" name = "acao" value="Excluir"
class="buttonform"/>
                  <input type="button"
onClick="document.location.href='userdb.jsp'" value="Sair"
class="buttonform"/>
                                                                  </td>
                                                            </tr>
                                                      </table>
                                                </fieldset>
                                          </form>
                                    </td>
                              </tr>
                  </table>
            </div>
            <br />
      </div>
</body>
</html>
```

A.14 - changeuc.jsp

```
<%@ page contentType="text/html" language="java" import="java.sql.*"%>
<%
      String acao = request.getParameter("acao");
      java.util.ArrayList UseCase = new java.util.ArrayList();
      UseCase = (java.util.ArrayList) session.getAttribute("USECASE");

      String prjCode = request.getParameter("prjcode");

      if(prjCode == null) {
      } else {
            if(((String)UseCase.get(7)) == null ||
((String)UseCase.get(7)).equals(prjCode) == false) {
                        for(int i = 0; i < 20; i++) {
                              if(i == 5 || i == 6) {
                                    UseCase.set(i, new
java.util.ArrayList());
                              } else {
                                    UseCase.set(i, "");

                              }
                        }
                        UseCase.set(7,prjCode);
                        session.setAttribute("USECASE",UseCase);
            } else {
                  UseCase.set(7, request.getParameter("prjcode"));
            }
      }
      if(acao != null) {
            UseCase.set(0, request.getParameter("typeusecase"));
            UseCase.set(1, request.getParameter("nameuc"));
            UseCase.set(2, request.getParameter("descuc"));
            response.sendRedirect("changedb2.jsp");
```

```
	}
%>
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/shtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pt" lang="pt">
<head>
<title>Escolha do Caso de Uso</title>
<style type="text/css">
<!--
@import url("common/common.css");
@import url("common/tabs.css");
-->
</style>
<script language="javascript">
<!--
	window.parent.leftFrame.location.reload();
-->
</script>
</head>
<body>
	<div id="pagetitle">Escolha do Caso de Uso</div>
	<div id="main">
		<br /><br />
<%
	pageContext.include("tabbar.jsp");
%>
		<br /><br />
		<div id="form">
			<table class="tableform">
				<tr>
					<td>
						<form name="changeuc" id="changeuc">
							<fieldset>
								<legend>Casos de
Uso</legend>
									<table class="tableform">
<tr>
  <td> </td>
</tr>
  <tr>
  <td class="descfield">Tipo:</td>
  <td>
                                                                <select
name="typeusecase" id="typeusecase">

<option value=""></option>
	<%
	if(UseCase.size() > 0) {
	if(((String)UseCase.get(0)).equals("0")) {
	%>
<option value="0" selected>Relat&oacute;rio</option>
	<%
	} else {
	%>
<option value="0">Relat&oacute;rio</option>
	<%
	}
	if(((String)UseCase.get(0)).equals("1")) {
	%>
```

```
<option value="1" selected>Manuten&ccedil;&atilde;o de Registro</option>
      <%
      } else {
      %>

<option value="1">Manuten&ccedil;&atilde;o de Registro</option>
      <%
      }
      } else {
      %>

<option value="1">Manuten&ccedil;&atilde;o de Registro</option>

<option value="0">Relat&oacute;rio</option>
      <%
      }
      %>

                                                                  </select>
                                                            </td>
                                                      </tr>
                  <tr>
                  <td class="descfield">Nome:</td>
                  <td>
                  <%
                  String nameuc = (String)UseCase.get(1);
                  if(nameuc == null)
                        nameuc = "";
                  %>
                  <input type="text" size="31" name="nameuc" id="nameuc"
value="<%= nameuc %>"/>
                  </td>
            </tr>
            <tr>
            <td class="descfield">Texto:</td>
            <td>
                  <%
                  String descuc = (String)UseCase.get(2);
                  if(descuc == null)
                  descuc = "";
                  %>
            <textarea cols="23" rows="5" name="descuc" id="descuc"><%=
descuc %></textarea>
                                                                  </td>
                                                                  </tr>
                                    <tr>
                                      <td colspan="2"> </td>
                                    </tr>
                                    <tr>
                                      <td colspan="2" class="tdline"></td>
                                    </tr>
                                    <tr>
                                      <td colspan="2"> </td>
                                    </tr>
                                    <tr>
                                      <td colspan="2" class="fieldbuttons">
                                          <input type="submit" name='acao'
value="Ok" class="buttonform" />
                                      </td>
                                    </tr>
```

```
                                    </table>
                                              </fieldset>
                                        </form>
                                  </td>
                            </tr>
                      </table>
                </div>
                <br />
      </div>
</body>
</html>
```

A.15 - changerules.jsp

```
<%@ page contentType="text/html" language="java" import="java.sql.*"%>
<%
      java.util.ArrayList UseCase = (java.util.ArrayList)
session.getAttribute("USECASE");
      String acao = request.getParameter("acao");

      if(acao != null)
            response.sendRedirect("saveinfo.jsp");


%>
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/shtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pt" lang="pt">
<head>
<title>Escolha da Tabela e Campos</title>
<style type="text/css">
<!--
@import url("common/common.css");
@import url("common/tabs.css");
-->
</style>
<script language="javascript">
function openWindow()
{
      var windowProps =
'statusbar=no,menubar=no,copyhistory=no,location=no,resizable=no,toolbar=no
,scrollbars=yes,width=500,height=600';
      newwin = window.open('ruledetail.jsp', '', windowProps);
}
</script>
</head>
<body>
      <div id="pagetitle">Escolha da Tabela e Campos</div>
      <div id="main">
            <br /><br />
<%
      pageContext.include("tabbar.jsp");
%>
      <br /><br />
      <div id="form">
            <table class="tableform">
                  <tr>
                  <td>
```

```
                <form name="changeuc" id="changeuc">
                <fieldset>
                <legend>Defini&ccedil;&atilde;o de Regras de
Neg&oacute;cio</legend>
                <table class="tableform">
                <tr>
                  <td colspan="2"> 
                  </td>
                </tr>
                <tr>
             <td width="45" align="right">Tipo:</td>
                 <td width="133"><select name="select">
            <option
value="Compara&ccedil;&atilde;o">Compara&ccedil;&atilde;o</option>
                </select></td>
                </tr>
                <tr>
                 <td colspan="2"><a href="#" class="alink"
onclick="openWindow();">Adicionar</a></td>
                  </tr>
                <tr>
                <td colspan="2" valign="top">
                <br />
                <table class="tableform" border="0">
                 <tr>
                <td></td>
                 </tr>
                 <tr>
                <td>
                                                                        <%

      String temp = ((java.util.ArrayList)UseCase.get(6)).toString();
      temp = temp.replaceAll("],","<br><br>");
      temp = temp.replace("[","");
      temp = temp.replace("]","");
      out.print(temp);
      %></td>
                </tr>
            </table>
      </td>
      </tr>
          <tr>
          <td colspan="2"> </td>
          </tr>
          <tr>
                <td colspan="3" class="tdline"></td>
                </tr>
                <tr>
                <td colspan="2"> </td>
                </tr>
                 <tr>
              <td colspan="2">
                  <span class="fieldbuttons">
                  <input type="submit" name="acao" value="Ok"
class="buttonform" />
                                                              </span></td>
                                                          </tr>
                                                      </table>
                                                  </fieldset>
                                              </form>
                                          </td>
```

```
                        </tr>
                  </table>
            </div>
            <br />
      </div>
</body>
</html>
```

A.16 - changefields.jsp

```
<%@ page contentType="text/html" language="java" import="java.sql.*"%>
<jsp:useBean id="selDBField" class="conexaoSuperBaan.ChangeFieldDB"/>
<%
                  java.util.ArrayList UseCase = new java.util.ArrayList();

                  UseCase = (java.util.ArrayList)
session.getAttribute("USECASE");

                  String baseSelecionada = (String)UseCase.get(3);
                  String tabelaSelecionada = (String)UseCase.get(4);

if(baseSelecionada != null && baseSelecionada.equals("") == false &&
tabelaSelecionada != null && tabelaSelecionada.equals("") == false) {

                  String acao = request.getParameter("acao");
                  String btnAdiciona = request.getParameter("adiciona");
                  String btnRemove = request.getParameter("remove");

                  String mensagem="";
                  ResultSetMetaData temp2;
                  java.util.ArrayList fields = new java.util.ArrayList();
                  java.util.ArrayList predefinedFields = new
java.util.ArrayList();
                  java.util.ArrayList selectedFields = new
java.util.ArrayList();

                  selectedFields = (java.util.ArrayList)UseCase.get(5);

                  selDBField.setBancoDados((String)UseCase.get(3));
                  selDBField.setBanco_DB((String)UseCase.get(4));
                  temp2 = selDBField.startChangeFields();

                  if(btnRemove != null) {
                        for(int i = 0; i < selectedFields.size(); i++) {
                              if(
((String)selectedFields.get(i)).equals(btnRemove) == true)
                                    selectedFields.remove(i);
                        }
                  }

                  int numberOfColumns = temp2.getColumnCount();
                  boolean exists = false;
                  for(int i =1; i <= numberOfColumns; i++) {
                  fields.add(temp2.getColumnName(i));
                        if(temp2.isNullable(i) ==
ResultSetMetaData.columnNoNulls) {
                              predefinedFields.add(temp2.getColumnName(i));
```

```
                              for(int j = 0; j < selectedFields.size(); j++
) {

      if(((String)selectedFields.get(j)).equals(temp2.getColumnName(i)) ==
true) {
                                          exists = true;
                                    }
                              }
                              if(exists == false)

      selectedFields.add(temp2.getColumnName(i));
                              exists = false;
                        }
                  }

                  if(btnAdiciona != null) {
                        String selecteds[] =
request.getParameterValues("fields");
                        exists = false;
                        for(int i = 0; i < selecteds.length; i++) {
                              for(int j = 0; j < selectedFields.size(); j++
) {

      if(((String)selectedFields.get(j)).equals(selecteds[i]) == true) {
                                          exists = true;
                                    }
                              }

                              if(exists == false)
                                    selectedFields.add(selecteds[i]);

                              exists = false;
                        }
                  }

                  if(acao != null) {
                        response.sendRedirect("changerules.jsp");
                  }

%>
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/shtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pt" lang="pt">
<head>
<title>Escolha da Tabela e Campos</title>
<style type="text/css">
<!--
@import url("common/common.css");
@import url("common/tabs.css");
-->
</style>
</head>
<body>
      <div id="pagetitle">Escolha da Tabela e Campos</div>
      <div id="main">
            <br /><br />
<%
      pageContext.include("tabbar.jsp");
%>
            <br /><br />
```

```
      <div id="form">
            <table class="tableform">
                  <tr>
                  <td>
                        <form name="changeuc" id="changeuc">
                        <fieldset>
                              <legend>Escolha de Campos</legend>
                              <table class="tableform">
                              <tr>
                              <td>
                              <table class="tableform" align="center">
                                <tr>
                                    <td colspan="2"> </td>
                                </tr>
                                <tr>
                              <td class="descfield">Campos:</td>
                              <td nowrap>
                              <select name="fields" id="fields" size="10"
multiple="multiple" >
                        <%
                        int count = fields.size();
                        String selected = "";
                        while(count != 0) {
                        count--;
/*
            if(((String)UseCase.get(4)).equals((String)tables.get(count)))
            selected = "selected";
      else
            selected = "";*/
      %>
      <option value="<%out.print((String)fields.get(count));%>" <%=
selected %>><%out.print((String)fields.get(count));%></option>

      <%
            }
      %>
</select>
<input type="submit" name="adiciona" id="adiciona" value=">>" />
      </td>
</tr>
      </table>
            </td>
            </tr>
            <tr>
            <td valign="top">
            <table class="tableform" border="0">
             <tr>
            <td colspan="2"> </td>
            </tr>
            <tr>
      <td>Ordem</td>
      <td>Campo</td>
      <td>Descrição</td>
      <td>Remover</td>
</tr>
      <%
      count = selectedFields.size();
      while(count != 0) {
      count--;
      %>
        <tr>
```

```
          <td><input type="text" name="ord" id="ord" value="<%= count %>"
size="3" /></td>
          <td><input type="text" name="field<%= count %>" id="field<%=
count %>" value="<%= (String)selectedFields.get(count) %>" size="30"/></td>
          <td><input type="text" name="desc<%= count %>" id="desc<%= count
%>" value="<%= (String)selectedFields.get(count) %>" size="30" /></td>
      <td><a href="changefields.jsp?remove=<%=
(String)selectedFields.get(count) %>">Remover</a></td>
        </tr>
      <%
      }
      %>
      </table>
                  </td>
                  </tr>
                  <tr>
                  <td> </td>
                  </tr>
                  <tr>
                  <td colspan="2" class="tdline"></td>
                  </tr>
                  <tr>
                  <td> </td>
                  </tr>
                  <tr>
                  <td><input type="submit" name="acao" value="Concluir"
/></td>
                  </tr>
                                              </table>
                                          </fieldset>
                                      </form>
                                  </td>
                              </tr>
                  </table>
            </div>
            <br />
      </div>
</body>
</html>
<%
} else {
      out.println("Base de Dados ou Tabela não selecionada");
}
%>
```

A.17 - changedb2.jsp

```
<%@ page contentType="text/html" language="java" import="java.sql.*"%>
<jsp:useBean id="selDB" class="conexaoSuperBaan.ConexaoUserDB"/>
<jsp:useBean id="selDBField" class="conexaoSuperBaan.ChangeFieldDB"/>
<%

                  String acao = request.getParameter("acao");
                  String banco = request.getParameter("BaseDados");
                  String mensagem="";
                  String texto = "";
                  ResultSet temp2;
                  java.util.ArrayList tables = new java.util.ArrayList();
                  java.util.ArrayList UseCase = new java.util.ArrayList();
```

```
                  UseCase = (java.util.ArrayList)
session.getAttribute("USECASE");

                  if(banco != null && banco.equals((String)UseCase.get(3))
== false)
                        UseCase.set(3, banco);
                  else
                        banco = (String)UseCase.get(3);

                  if(acao != null) {
                        UseCase.set(3, banco);
                        UseCase.set(4,
request.getParameter("selectBancoDados"));
                        response.sendRedirect("changefields.jsp");

                  } else {
                        //out.print((String)request.getParameter("acao"));
                  }
                  if (banco!=null ){
                        UseCase.set(3, banco);
                  try{
                         selDBField.setBancoDados(banco);
                         boolean test = selDBField.startChangeBanco();
                        if (test)
                        {
                        temp2 = selDBField.getResultado();
                          do{
                              tables.add(temp2.getString(3));
                           }while (temp2.next());
                      }
                  }catch(Exception e){mensagem = e.getMessage();}
             }
%>
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/shtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pt" lang="pt">
<head>
<title>Escolha do Banco de Dados</title>
<style type="text/css">
<!--
@import url("common/common.css");
@import url("common/tabs.css");
-->
</style>
<script language="javascript">
function selDB(valor){
var txt=new String();
var testeJS=valor;
if (valor != ''){
            spSelDB.innerHTML = txt;
}else{
txt='';
txt+='<select name="" >';
txt+='<option value=""></option>';
txt+='</select>';
spSelDB.innerHTML = txt;
}

}
</script>
```

```
</head>
<body>
      <div id="pagetitle">Escolha do Banco de Dados</div>
      <div id="main">
            <br /><br />
<%
      pageContext.include("tabbar.jsp");
%>
            <br/><br />
            <%//=mensagem %>

            <div id="form">
                  <table class="tableform">
                        <tr>
                              <td>
                        <form name="changeuc" id="changeuc">
                        <fieldset>
                        <legend>Escolha do Banco de Dados</legend>
                          <table class="tableform">
                           <tr>
                             <td> </td>
                           </tr>
                        <tr>
                        <td class="descfield">Banco de Dados:</td>
                        <td>
                        <select name="BaseDados" id="BaseDados"
onchange="document.changeuc.submit()">
                        <option value=""></option>
                        <%
                           try {
                               selDB.setConsultaManagerCliDB();
                                ResultSet temp =selDB.getResultado();
                                temp.next();
                                String selected = "";
                                do{
      if(((String)UseCase.get(3)).equals(temp.getString("nome_banco")))
      {
            selected = "selected";
      } else
            selected = "";
                              %>
      <option value="<%out.print(temp.getString("nome_banco"));%>" <%=
selected %>><%out.print(temp.getString("nome_banco"));%></option>
      <%
                                 }while (temp.next());
                               }catch (Exception e) {e.printStackTrace(); }
                                  %>
                                 </select>
                        </td>
                  </tr>
                  <tr>
                  <td class="descfield">Tabela: </td>
                  <td>
      <select name="selectBancoDados">
      <%
      int count = tables.size();
      String selected;
      while(count != 0) {
      count--;
      out.println((String)UseCase.get(4));
```

```
      out.println(" -- ");
      out.println((String)tables.get(count));
      if(((String)UseCase.get(4)).equals((String)tables.get(count)))
            selected = "selected";
      else
            selected = "";
      %>
      <option value="<%out.print((String)tables.get(count));%>" <%=
selected %>><%out.print((String)tables.get(count));%></option>
      <%
            }
      %>
      </select>
                                    </td>
                               </tr>
                                    <tr>
                                      <td> </td>
                                    </tr>
                                    <tr>
                                      <td colspan="2" class="tdline"></td>
                                    </tr>
                                    <tr>
                                      <td> </td>
                                    </tr>
                                    <tr>
                                     <td colspan="2" class="fieldbuttons">
                                     <input type="submit" name='acao'
value="Ok" class="buttonform" />
                                    </td>
                                    </tr>
                                  </table>
                                           </fieldset>
                                     </form>
                               </td>
                          </tr>
                     </table>
            </div>
            <br />
      </div>
</body>
</html>
```

APÊNDICE B – BEANS DA FERRAMENTA BRD TOOL

B.1 - ChangeFieldDB.java

```
package conexaoSuperBaan;

import java.sql.Connection;
import java.sql.DatabaseMetaData;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.sql.Statement;


public class ChangeFieldDB{


	public Connection con;
	public Statement stm;
	public ResultSet res = null;
	private String banco_dados = null;
	private String fields_DB = null;
	private String table_DB = null;
	private     DatabaseMetaData dmd = null;

	public void setBancoDados(String banco_dados){
		this.banco_dados = banco_dados;
	}
	public void setFields(String fields_DB){
		this.fields_DB = fields_DB;
	}
	public void setBanco_DB(String table_DB){
		this.table_DB = table_DB;
	}
	public String getBancoDados(){
		return banco_dados;
	}
	public String getFields(){
		return fields_DB;
	}
	public String getBanco_DB(){
		return table_DB;
	}
	public ChangeFieldDB(){

	}

	public void startConnection(){

		String driver = "org.postgresql.Driver";
```

```
		String url = "jdbc:postgresql://localhost:5432/"+banco_dados;
		try{
		Class.forName(driver).newInstance();
		con = DriverManager.getConnection(url,"postgres","root");
		//stm = con.createStatement();
		}
		catch(Exception e){
				e.printStackTrace();
		}
	}

	public boolean startChangeBanco(){

		startConnection();

		boolean test = false;

		try {

			DatabaseMetaData dmd = con.getMetaData();
			String[] types = {"TABLE"};
		       res = dmd.getTables(null, null, "%", types);
			if (res.next()) {
					res.getString(3);
					test=true;

				}else{test=false;}

		}catch (SQLException e){e.getMessage();}

		return test;
	}


	public ResultSetMetaData startChangeFields(){
		startConnection();
		ResultSetMetaData rsmd = null;

		try {
			int i;

			stm = con.createStatement();
			ResultSet rs = stm.executeQuery("Select * from
"+table_DB);
			rsmd = rs.getMetaData();
			int numCols = rsmd.getColumnCount();

			for(i=1; i<=numCols;i++){
				rsmd.getColumnLabel(i);
			}

		}catch (SQLException e){e.getMessage();}
		return rsmd;

	}

	public ResultSet startGetPrimaryKey()
	{

		ResultSet resultSetPK = null;
		try {
```

```
                dmd = con.getMetaData();
                resultSetPK = dmd.getPrimaryKeys(null, null, table_DB);
            } catch(Exception e) {
                        e.getMessage();
            }
            return resultSetPK;
      }

      public ResultSet getResultado() {
            return res;
      }


}
```

B.2 - CheckAtribute.java

```
package conexaoSuperBaan;

import java.sql.Connection;
import java.sql.DatabaseMetaData;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.Statement;
import java.util.ArrayList;


public class CheckAtribute{


      public Connection con;
      public Statement stm;
      public ResultSet res = null;
      private String banco_dados = null;
      private     DatabaseMetaData dmd = null;

      public void setBancoDados(String banco_dados){
            this.banco_dados = banco_dados;
      }

      public String getBancoDados(){
            return banco_dados;
      }

      public CheckAtribute(){

      }

      public void startConnection(){

            String driver = "org.postgresql.Driver";
            String url = "jdbc:postgresql://localhost:5432/"+banco_dados;
            try{
            Class.forName(driver).newInstance();
            con = DriverManager.getConnection(url,"postgres","root");
            stm = con.createStatement();
            }
            catch(Exception e){
```

```
                        e.printStackTrace();
                }
        }


        public ArrayList checkFields(String table, String field){// verifica
seo campo pode ser nulo
                ArrayList resultado = new ArrayList();
                for(int i = 0; i < 5; i++)
                        resultado.add("");
                ArrayList tempPK = new ArrayList();
                ArrayList auxFK = new ArrayList();
                startConnection();
                ResultSet resultSetPK = null;
                ResultSet resultSetFK = null;
                try {
                        dmd = con.getMetaData();
                        resultSetPK = dmd.getPrimaryKeys(null, null, table);
                                while(resultSetPK.next()){

        tempPK.add(resultSetPK.getString("COLUMN_NAME"));
                                }
                        for (int i=0;i<tempPK.size();i++){
                                if (tempPK.get(i).equals(field)){
                                        resultado.set(0,"true");
                                        break;
                                }else{resultado.set(0,"false");}
                        }
                        ResultSet rs = stm.executeQuery("Select * from "+table);
                        ResultSetMetaData rsmd = rs.getMetaData();
                        int numCols = rsmd.getColumnCount();
                                for(int i=1; i<=numCols;i++){
                                        if (rsmd.getColumnLabel(i).equals(field)){

        resultado.set(1,rsmd.getColumnTypeName(i));

        resultado.set(2,""+rsmd.getColumnDisplaySize(i));
                                                if (rsmd.isNullable(i)==0){
                                                        resultado.set(3,"false");
                                                        }else {resultado.set(3,"true");}
                                                break;
                                        }else{
                                                resultado.set(1,"");
                                                resultado.set(2,"");
                                                resultado.set(3,"");
                                                }
                                }
                        resultSetFK = dmd.getImportedKeys(null,null,table);
                                while(resultSetFK.next()){
                                        ArrayList tempAux = new ArrayList();

        tempAux.add(0,resultSetFK.getString("FKCOLUMN_NAME"));

        tempAux.add(1,resultSetFK.getString("PKTABLE_NAME"));

        tempAux.add(2,resultSetFK.getString("PKCOLUMN_NAME"));
                                        auxFK.add(tempAux);
                                 }
                        System.out.print(auxFK + field);
                        boolean passou = false;
                        for (int j=0;j<auxFK.size();j++){
```

```
                        if
(field.equals((String)((ArrayList)auxFK.get(j)).get(0))){
                              resultado.set(4,auxFK.get(j));
                              passou = true;
                        }
                  }
                  if(passou == false) {
                        resultado.set(4,new ArrayList());
                  }
            } catch(Exception e) {
                        System.out.print(e.getMessage());
            }
            return resultado;
      }


      public ResultSet getResultado() {
            return res;
      }


}
```

B.3 - Conexao.java

```
package conexaoSuperBaan;

import java.sql.*;

public class Conexao {

      public Connection con;
      public Statement stm;
      public ResultSet res = null;
      private String login = null;
      private String nome = null;
      private String sobrenome = null;
      private String senha = null;
      private String email = null;
      private String tipo = null;

      public Conexao() {

            try {

                  Class.forName("org.postgresql.Driver");
                  con =
                     DriverManager.getConnection(
"jdbc:postgresql://localhost:5432/Super Baan", "postgres", "root");
                  stm = con.createStatement();

            } catch (Exception e) {
                  System.out.println("não foi possível conectar ao banco" +
e.getMessage());
            }
      }

      public void setLogin(String login){
            this.login = login;
```

```
            if (this.login != login)
                  if(this.login.trim().length() == 0)
                        login = null;
                  else login = this.login;
      }

      public void setNome(String nome){
            this.nome = nome;
      }

      public void setSobrenome(String sobrenome){
            this.sobrenome = sobrenome;
      }

      public void setSenha(String senha){
            this.senha = senha;
      }

      public void setEmail(String email){
            this.email = email;
      }

      public void setTipo(String tipo){
            this.tipo = tipo;
      }


      public String getLogin(){
            return login;
      }

      public String getNome(){
            return nome;
      }

      public String getSobrenome(){
            return sobrenome;
      }

      public String getSenha(){
            return senha;
      }

      public String getEmail(){
            return email;
      }

      public String getTipo(){
            return tipo;
      }

      public void insertDataManagerUser(){

            try {
                  String query = "insert into
usuario(login,nome,sobrenome,senha,email,tipo)
values('"+login+"','"+nome+"','"+sobrenome+"','"+senha+"','"+email+"','"+ti
po+"')";
                  stm.executeUpdate(query);
```

```
            }catch (SQLException e){System.out.println("Erro na inserção:"
+ e.getMessage());}

      }

      public boolean updateDataManagerUser(){

            boolean test = false;

            try {
                  String query = "update usuario set login = '" + login+
"', nome = '" + nome+ "',sobrenome = '" + sobrenome+ "',senha = '" + senha+
"', email = '" + email+ "', tipo = '" + tipo+ "' where login = '" + login +
"'";

                  int line = stm.executeUpdate(query);

                  if (line > 0)
                        test = true;
                  else
                        test = false;

            }catch (SQLException e){System.out.println("Erro na alteração:"
+ e.getMessage());}

            return test;
      }

      public boolean deleteDataManagerUser(){

       boolean test = false;

            try {
                  String query = "delete from usuario where login='" +
login+"'";
                  int line = stm.executeUpdate(query);

                  if(line > 0)
                        test = true;
                  else
                        test = false;

            }catch (SQLException e){System.out.println("Erro na exclusão: "
+ e.getMessage());}

            return test;
      }

      public boolean selectDataManagerUser(){

            boolean test = false;

            try {
                  String query = "select * from usuario where
login='"+login+"'";
                  res = stm.executeQuery(query);

                  if (res.next()){test = true;}
                  else{test = false;}
```

```
            }catch (SQLException e){System.out.println("Erro na seleção: "
+ e.getMessage());}

            return test;
      }

      public boolean startLogin(){

            boolean test = false;

            try {
                  String query = "select * from usuario where
login='"+login+"' AND senha ='"+senha+"'";
                  res = stm.executeQuery(query);

                  if (res.next()){test = true;}
                  else{test = false;}

            }catch (SQLException e){System.out.println("Erro no login: " +
e.getMessage());}

            return test;
      }

      public boolean selectProjects(){

            boolean test = false;

            try {
                  String query = "select cod_prj, descricao_prj from
projeto where cod_prj in ( SELECT cod_prj FROM user_project WHERE login =
'"+login+ "')";
                  res = stm.executeQuery(query);

                  if (res.next()){test = true;}
                  else{test = false;}

            }catch (SQLException e){System.out.println("Erro na seleção: "
+ e.getMessage());}

            return test;
      }

      public boolean selectUseCase(int cod_prj){

            boolean test = false;

            try {
                  String query = "select cod_use_case_prj, nome from
use_case_project where cod_prj ="+cod_prj;
                  res = stm.executeQuery(query);

                  if (res.next()){test = true;}
                  else{test = false;}

            }catch (SQLException e){System.out.println("Erro na seleção: "
+ e.getMessage());}

            return test;
      }
```

```
	public void setConsultaManagerUser() {

		try {
			res = stm.executeQuery("select * from usuario");
		}
		catch (SQLException e){
			e.printStackTrace();
		}

	}

	public ResultSet getResultado() {
		return res;
	}
}
```

B.4 - ConexaoBusinessRulesUseCase.java

```
package conexaoSuperBaan;

import java.sql.Connection;
import java.sql.DatabaseMetaData;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.*;
import javax.swing.text.DefaultEditorKit.InsertBreakAction;


public class ConexaoBusinessRulesUseCase{


	public Connection con;
	public Statement stm;
	public ResultSet res = null;



	public ConexaoBusinessRulesUseCase(){
		startConnection();
	}

	public void startConnection(){

		String driver = "org.postgresql.Driver";
		String url = "jdbc:postgresql://localhost:5432/Super Baan";
		try{
		Class.forName(driver).newInstance();
		con = DriverManager.getConnection(url,"postgres","root");
		stm = con.createStatement();

		}
		catch(Exception e){
				e.printStackTrace();
		}
```

```
	}

	public boolean startInsertAllData(ArrayList data){
		ResultSet resCodDB = null;
		ResultSet resMaxUseCase = null;
		int cod_db, cod_UseCase;
		startConnection();

		boolean test = false;
		String InsertBusinessRules="";
		try {
			String querySelCodDB = "select cod_dbcli from dbcliente
where nome_banco ='"+data.get(3)+"');";
			resCodDB = stm.executeQuery(querySelCodDB);
			resCodDB.next();
			cod_db = resCodDB.getInt("cod_dbcli");
			System.out.println("passou 1 - \n"+querySelCodDB+"\n");

// ---------------------------------------------------------------------
			String queryInsertUseCasePrj = "insert into
use_case_project values (" +

	"default,"+data.get(7)+","+data.get(0)+",'"+data.get(1)+

	"','"+data.get(2)+"',"+cod_db+",'"+data.get(4)+"'); ";
			System.out.println("passou 2 -
\n"+queryInsertUseCasePrj+"\n");
			stm.executeUpdate(queryInsertUseCasePrj);

//----------------------------------------------------------------------
			String querySelMAXUseCase = "select max(cod_use_case_prj)
from use_case_project";
			resMaxUseCase = stm.executeQuery(querySelMAXUseCase);
			resMaxUseCase.next();
			cod_UseCase = resMaxUseCase.getInt("MAX");
			System.out.println("passou 3 -
\n"+querySelMAXUseCase+"\n");
//----------------------------------------------------------------------
			ArrayList temp = (ArrayList) data.get(5);
			for (int i=0; i<temp.size();i++){
				String queryInsertFieldsUseCase = "insert into
campo_caso_uso (cod_use_case_prj,"+

	"campo,cod_prj) values ("+cod_UseCase+",'"+

	temp.get(i)+"',"+data.get(7)+");";
					stm.executeUpdate(queryInsertFieldsUseCase);
					System.out.println("passou 4  -
\n"+queryInsertFieldsUseCase+"\n");
			}
			temp = (ArrayList) data.get(6);

			for (int i=0; i<temp.size();i++){
			ArrayList temp2 = (ArrayList)temp.get(i);
			String cod_use_case_prj_acao_true =null,
cod_prj_acao_true=null;
			String cod_use_case_prj_acao_false=null,
cod_prj_acao_false=null;
			//System.out.println(temp2.get(8));
```

```
                  InsertBusinessRules = "insert into regras_de_negocio
(cod_regras_negocio,  cod_use_case_prj,"+
                                                      "cod_prj, nome,
tabela1, campo1, propriedade, valor, tabela2, campo2,"+

      "cod_use_case_prj_acao_true, cod_prj_acao_true, valor_acao_true,"+

      "cod_use_case_prj_acao_false, cod_prj_acao_false, valor_acao_false,
campo_digitado)"+

      "values(default,"+

      cod_UseCase+","+data.get(7)+",'"+temp2.get(0)+"','"+

      temp2.get(1)+"','"+temp2.get(2)+"','"+temp2.get(6)+"','"+

      temp2.get(5)+"','"+temp2.get(3)+"','"+temp2.get(4)+ "',";


      if((((String)temp2.get(7)).equals("Mensagem")==false)&&(((String)temp
2.get(7)).equals("")==false)){
                        InsertBusinessRules+=((String)temp2.get(7))+",";
                        InsertBusinessRules+=((String)data.get(7))+",";


                  }else{
                        InsertBusinessRules+="null,";
                        InsertBusinessRules+="null,";
                  }
                  InsertBusinessRules+="'"+temp2.get(8)+"',";

      if((((String)temp2.get(9)).equals("Mensagem")==false)&&(((String)temp
2.get(9)).equals("")==false)){
                        InsertBusinessRules+=((String)temp2.get(9))+",";
                        InsertBusinessRules+=((String)data.get(7))+",";

                  }else{
                        InsertBusinessRules+="null,";
                        InsertBusinessRules+="null,";
                  }

      InsertBusinessRules+="'"+temp2.get(10)+"','"+temp2.get(11)+"');";

                  //+",'"+temp2.get(8)+
                        //
      "',"+Integer.parseInt(cod_use_case_prj_acao_false)+","+Integer.parseI
nt(cod_prj_acao_false)+",'"+temp2.get(10)+"')";
                  stm.executeUpdate(InsertBusinessRules);
                  System.out.println("passou 5 -
\n"+InsertBusinessRules+"\n");
                  }
            }catch (SQLException e){System.out.println("Erro na inserção:"
+ e.getMessage()+InsertBusinessRules);}

            return test;
      }

      public boolean selectCasoUsoCadastrado(int prj){

            boolean test = false;
```

```
            try {
                  String query = "select cod_use_case_prj, nome from
use_case_project where cod_prj ="+prj;
                  res = stm.executeQuery(query);

                  if (res.next()){test = true;}
                  else{test = false;}

            }catch (SQLException e){System.out.println("Erro na seleção: "
+ e.getMessage());}

            return test;
      }

      public ResultSet getResultado() {
            return res;
      }  }
```

B.5 - ConexaoPrj.java

```
package conexaoSuperBaan;

import java.sql.*;

public class ConexaoPrj {

      public Connection con;
      public Statement stm;
      public ResultSet res = null;
      private int cod_prj = 0;
      private String descricao = null;

      public ConexaoPrj() {

            try {

                  Class.forName("org.postgresql.Driver");
                  con =
                     DriverManager.getConnection(
"jdbc:postgresql://localhost:5432/Super Baan", "postgres", "root");
                  stm = con.createStatement();

            } catch (Exception e) {
                  System.out.println("não foi possível conectar ao banco" +
e.getMessage());
            }
      }

      public void setCodigoPrj(int cod_prj){
            this.cod_prj = cod_prj;

      }

      public void setDescricao(String descricao){
            this.descricao = descricao;
      }


      public int getCodigoPrj(){
```

```
            return cod_prj;
      }

      public String getDescricao(){
            return descricao;
      }

      public void insertDataManagerPrj(){

            try {
                  String query = "insert into
projeto(cod_prj,descricao_prj) values("+cod_prj+",'"+descricao+"')";
                  stm.executeUpdate(query);

            }catch (SQLException e){System.out.println("Erro na inserção:"
+ e.getMessage());}

      }

      public boolean updateDataManagerPrj(){

            boolean test = false;

            try {
                  String query = "update projeto set descricao_prj = '" +
descricao+ "' where cod_prj = " + cod_prj ;

                  int line = stm.executeUpdate(query);

                  if (line > 0)
                        test = true;
                  else
                        test = false;

            }catch (SQLException e){System.out.println("Erro na alteração:"
+ e.getMessage());}

            return test;
      }

      public boolean deleteDataManagerPrj(){

       boolean test = false;

            try {
                  String query = "delete from projeto where cod_prj= " +
cod_prj;
                  int line = stm.executeUpdate(query);

                  if(line > 0)
                        test = true;
                  else
                        test = false;

            }catch (SQLException e){System.out.println("Erro na exclusão: "
+ e.getMessage());}

            return test;
      }

      public boolean selectDataManagerPrj(){
```

```
            boolean test = false;

            try {
                  String query = "select * from projeto where
cod_prj="+cod_prj;
                  res = stm.executeQuery(query);

                  if (res.next()){test = true;}
                  else{test = false;}

            }catch (SQLException e){System.out.println("Erro na seleção: "
+ e.getMessage());}

            return test;
      }

      public void setConsultaManagerPrj() {

            try {
                  res = stm.executeQuery("select * from projeto");
            }
            catch (SQLException e){
                  e.printStackTrace();
            }

      }
      public ResultSet getResultado() {
            return res;
      }
}
```

B.6 - ConexaoRegraNegocio.java

```
package conexaoSuperBaan;

import java.sql.*;

public class ConexaoRegraNegocio {

      public Connection con;
      public Statement stm;
      public ResultSet res = null;
      private int cod_prj = 0;
      private int cod_regras_negocio=0;
      private int cod_use_case_prj=0;
      private String nome= null;
      private int ordem=0;
      private String tabela1= null;
      private String campo1= null;
      private String propriedade= null;
      private String valor= null;
      private String tabela2= null;
      private String campo2= null;
      private String avaliar_se= null;
      private int cod_use_case_prj_acao=0;
      private int cod_prj_acao=0;
      private String valor_acao= null;
```

```
	public ConexaoRegraNegocio() {

		try {

			Class.forName("org.postgresql.Driver");
			con =
			   DriverManager.getConnection(
"jdbc:postgresql://localhost:5432/Super Baan", "postgres", "root");
			stm = con.createStatement();

		} catch (Exception e) {
			System.out.println("Não foi possível conectar ao banco" +
e.getMessage());
		}
	}

	public void setCodigoRegraNegocio(int cod_regras_negocio){
		this.cod_regras_negocio = cod_regras_negocio;

	}

	public void setCodigoUseCase(int cod_use_case_prj){
		this.cod_use_case_prj = cod_use_case_prj;
	}

	public void setCodigoPrj(int cod_prj){
		this.cod_prj = cod_prj;
	}

	public void setNomeRegraNegocio(String nome){
		this.nome = nome;
	}

	public void setOrdemRegraNegocio(int ordem){
		this.ordem = ordem;
	}

	public void setTabela1RegraNegocio(String tabela1){
		this.tabela1 = tabela1;
	}

	public void setCampo1RegraNegocio(String campo1){
		this.campo1 = campo1;
	}

	public void setPropRegraNegocio(String propriedade){
		this.propriedade = propriedade;
	}

	public void setValorRegraNegocio(String valor){
		this.valor = valor;
	}

	public void setTabela2RegraNegocio(String tabela2){
		this.tabela2 = tabela2;
	}

	public void setCampo2RegraNegocio(String campo2){
		this.campo2 = campo2;
	}
```

```
	public void setAvaliarSeRegraNegocio(String avaliar_se){
		this.avaliar_se = avaliar_se;
	}

	public void setCodigoUseCasePrjAcao(int cod_use_case_prj_acao){
		this.cod_use_case_prj_acao = cod_use_case_prj_acao;
	}

	public void setCodigoPrjAcao(int cod_prj_acao){
		this.cod_prj_acao = cod_prj_acao;
	}

	public void setValorAcao(String valor_acao){
		this.valor_acao = valor_acao;
	}
// ----------------------------------------------------------------------

	public int getCodigoRegraNegocio(){
		return cod_regras_negocio;
	}

	public int getCodigoUseCase(){
		return cod_use_case_prj;
	}

	public int getCodigoPrj(){
		return cod_prj;
	}

	public String getNomeRegraNegocio(){
		return nome;
	}

	public int getOrdemRegraNegocio(){
		return ordem;
	}

	public String getTabela1RegraNegocio(){
		return tabela1;
	}

	public String getCampo1RegraNegocio(){
		return campo1;
	}

	public String getPropRegraNegocio(){
		return propriedade;
	}

	public String getValorRegraNegocio(){
		return valor;
	}

	public String getTabela2RegraNegocio(){
		return tabela2;
	}

	public String getCampo2RegraNegocio(){
		return campo2;
	}
```

```
	public String getAvaliarSeRegraNegocio(){
		return avaliar_se;
	}

	public int getCodigoUseCasePrjAcao(){
		return cod_use_case_prj_acao;
	}

	public int getCodigoPrjAcao(){
		return cod_prj_acao;
	}

	public String getValorAcao(){
		return valor_acao;
	}

//-----------------------------------------------------------------------


	public void insertDataBusinessRules(){

		try {

			String query = "insert into regras_de_negocio() values+"
+
				"(default,"+cod_use_case_prj+","+cod_prj+",'"+
				nome+"','"+ordem +",'"+tabela1+"','"+campo1+

	"','"+propriedade+"','"+valor+"','"+tabela2+"','"+campo2+"','"+avalia
r_se+

	"','"+cod_use_case_prj_acao+","+cod_prj_acao+",'"+valor_acao+"')";
			stm.executeUpdate(query);

		}catch (SQLException e){System.out.println("Erro na inserção:"
+ e.getMessage());}

	}

	public boolean updateDataBusinessRules(){

		boolean test = false;

		try {

			String query = "update regras_de_negocio set
cod_use_case_prj = "+
			cod_use_case_prj+", cod_prj= "+cod_prj+",nome ='"+
			nome+"', ordem ="+ordem
+",tabela1='"+tabela1+"',campo1='"+campo1+

	"',propriedades='"+propriedade+"',valor='"+valor+"',tabela2='"+tabela
2+

	"',campo2='"+campo2+"',avaliar_se='"+avaliar_se+"',cod_use_case_prj_a
cao="+cod_use_case_prj_acao+

	",cod_prj_acao="+cod_prj_acao+",valor_acao='"+valor_acao+"')";
			int line = stm.executeUpdate(query);

			if (line > 0)
```

```
                        test = true;
                  else
                        test = false;

            }catch (SQLException e){System.out.println("Erro na alteração:"
+ e.getMessage());}

            return test;
      }

      public boolean deleteDataBusinessRules(){

       boolean test = false;

            try {

                  String query = "delete from regras_de_negocio where
cod_regras_negocio= " +
cod_regras_negocio+",cod_use_case_prj="+cod_use_case_prj+"',cod_prj="+cod_p
rj;
                  int line = stm.executeUpdate(query);

                  if(line > 0)
                        test = true;
                  else
                        test = false;

            }catch (SQLException e){System.out.println("Erro na exclusão: "
+ e.getMessage());}

            return test;
      }

      public boolean selectDataManagerPrj(){

            boolean test = false;

            try {
                  String query = "select * from regras_de_negocio where
cod_regras_negocio= " +
cod_regras_negocio+",cod_use_case_prj="+cod_use_case_prj+"',cod_prj="+cod_p
rj;
                  res = stm.executeQuery(query);

                  if (res.next()){test = true;}
                  else{test = false;}

            }catch (SQLException e){System.out.println("Erro na seleção: "
+ e.getMessage());}

            return test;
      }

      public void setConsultaManagerPrj() {

            try {
                  res = stm.executeQuery("select * from
regras_de_negocio");
            }
            catch (SQLException e){
                  e.printStackTrace();
```

```
            }

      }



      public ResultSet getResultado() {
            return res;
      }
}
```

B.7 - ConexaoUserDB.java

```
package conexaoSuperBaan;

import java.sql.*;

public class ConexaoUserDB {

      public Connection con;
      public Statement stm;
      public ResultSet res = null;
      private int cod_dbcli = 0;
      private String servidor = null;
      private String usuario_meta = null;
      private String senha_meta = null;
      private String usuario_db = null;
      private String senha_db = null;
      private String nome_banco = null;
      private String sgbd_cli = null;

      public ConexaoUserDB() {

            try {

                  Class.forName("org.postgresql.Driver");
                  con =
                     DriverManager.getConnection(
"jdbc:postgresql://localhost:5432/Super Baan", "postgres", "root");
                  stm = con.createStatement();

            } catch (Exception e) {
                  System.out.println("não foi possível conectar ao banco" +
e.getMessage());
            }
      }

//-----------------------------------------------------------------------

      public void setCodigoUserDB(int cod_dbcli){
            this.cod_dbcli = cod_dbcli;
      }

      public void setServidor(String servidor){
            this.servidor = servidor;
      }
```

```
	public void setUsuarioMeta(String usuario_meta){
		this.usuario_meta = usuario_meta;
	}

	public void setSenhaMeta(String senha_meta){
		this.senha_meta = senha_meta;
	}

	public void setUsuarioDB(String usuario_db){
		this.usuario_db = usuario_db;
	}

	public void setSenhaDB(String senha_db){
		this.senha_db = senha_db;
	}

	public void setNomeBanco(String nome_banco){
		this.nome_banco = nome_banco;
	}

	public void setSGBD(String sgbd_cli){
		this.sgbd_cli = sgbd_cli;
	}

//---------------------------------------------------------------------

	public int getCodigoUserDB(){
		return cod_dbcli;
	}

	public String getServidor(){
		return servidor;
	}

	public String setUsuarioMeta(){
		return usuario_meta;
	}

	public String setSenhaMeta(){
		return senha_meta;
	}

	public String setUsuarioDB(){
		return usuario_db;
	}

	public String setSenhaDB(){
		return senha_db;
	}

	public String setNomeBanco(){
		return nome_banco;
	}

	public String setSGBD(){
		return sgbd_cli;
	}


//---------------------------------------------------------------------
```

```
	public void insertDataManagerCliDB(){

		try {
			String query = "insert into
dbcliente(cod_dbcli,servidor,usuario_meta,senha_meta,usuario_db,senha_db,no
me_banco,sgbd_cli) values(default,'"+servidor+"','"+usuario_meta
+"','"+senha_meta+"','"+usuario_db+"','"+senha_db+"','"+nome_banco+"','"+sg
bd_cli+"')";
			stm.executeUpdate(query);

		}catch (SQLException e){System.out.println("Erro na inserção:"
+ e.getMessage());}

	}

	public boolean updateDataManagerCliDB(){

		boolean test = false;

		try {
			String query = "update dbcliente set servidor = '" +
servidor+ "', usuario_meta = '"+usuario_meta +"', senha_meta =
'"+senha_meta+"', usuario_db= '"+usuario_db+"',senha_db = '"+senha_db+"',
nome_banco = '"+nome_banco+"', sgbd_cli = '"+sgbd_cli+"' where
cod_dbcli="+cod_dbcli;

			int line = stm.executeUpdate(query);

			if (line > 0)
				test = true;
			else
				test = false;

		}catch (SQLException e){System.out.println("Erro na alteração:"
+ e.getMessage());}

		return test;
	}

	public boolean deleteDataManagerCliDB(){

	 boolean test = false;

		try {
			String query = "delete from dbcliente where cod_dbcli= "
+ cod_dbcli;
			int line = stm.executeUpdate(query);

			if(line > 0)
				test = true;
			else
				test = false;

		}catch (SQLException e){System.out.println("Erro na exclusão: "
+ e.getMessage());}

		return test;
	}

	public boolean selectDataManagerCliDB(){
```

```
            boolean test = false;

            try {
                  String query = "select * from dbcliente where
cod_dbcli="+cod_dbcli;
                  res = stm.executeQuery(query);

                  if (res.next()){test = true;}
                  else{test = false;}

            }catch (SQLException e){System.out.println("Erro na seleção: "
+ e.getMessage());}

            return test;
      }

      public void setConsultaManagerCliDB() {

            try {
                  res = stm.executeQuery("select * from dbcliente");
            }
            catch (SQLException e){
                  e.printStackTrace();
            }
      }




      public ResultSet getResultado() {
            return res;
      }
}
```

B.8 - ConexaoUserPrj.java

```
package conexaoSuperBaan;

import java.sql.*;

public class ConexaoUserPrj {

      public Connection con;
      public Statement stm;
      public ResultSet res = null;
      private int cod_prj = 0;
      private String login = null;

      public ConexaoUserPrj() {

            try {

                  Class.forName("org.postgresql.Driver");
                  con =
                     DriverManager.getConnection(
"jdbc:postgresql://localhost:5432/Super Baan", "postgres", "root");
                  stm = con.createStatement();
```

```
            } catch (Exception e) {
                  System.out.println("não foi possível conectar ao banco" +
e.getMessage());
            }
      }

      public void setCodigoPrj(int cod_prj){
            this.cod_prj = cod_prj;
      }

      public void setLogin(String login){
            this.login = login;
      }


      public int getCodigoPrj(){
            return cod_prj;
      }

      public String getLogin(){
            return login;
      }

      public void insertDataManagerUserPrj(){

            try {
                  String query = "insert into user_project(login,cod_prj)
values('"+login+"',"+cod_prj+")";
                  stm.executeUpdate(query);

            }catch (SQLException e){System.out.println("Erro na inserção:"
+ e.getMessage());}

      }


      public boolean deleteDataManagerUserPrj(){

       boolean test = false;

            try {
                  String query = "delete from user_project where cod_prj= "
+ cod_prj;
                  int line = stm.executeUpdate(query);

                  if(line > 0)
                        test = true;
                  else
                        test = false;

            }catch (SQLException e){System.out.println("Erro na exclusão: "
+ e.getMessage());}

            return test;
      }

      public void setSelectManagerUserPrj() {

            try {
                  res = stm.executeQuery("select * from user_project where
cod_prj="+cod_prj);
```

```
		}
		catch (SQLException e){
			e.printStackTrace();
		}

	}

	public void setConsultaManagerUserPrj() {

		try {
			res = stm.executeQuery("select * from user_project");
		}
		catch (SQLException e){
			e.printStackTrace();
		}

	}


	public ResultSet getResultado() {
		return res;
	}
}
```

B.9 - executeQuery.java

```
package conexaoSuperBaan;

import java.sql.Connection;
import java.sql.DatabaseMetaData;
import java.sql.DriverManager;
import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;



public class executeQuery{


	public Connection con;
	public Statement stm;
	public ResultSet res = null;
	private String banco_dados = null;


	public void setBancoDados(String banco_dados){
		this.banco_dados = banco_dados;
	}

	public String getBancoDados(){
		return banco_dados;
	}
```

```
	public executeQuery(){

	}

	public void startConnection(){

		String driver = "org.postgresql.Driver";
		String url = "jdbc:postgresql://localhost:5432/"+banco_dados;
		try{
		Class.forName(driver).newInstance();
		con = DriverManager.getConnection(url,"postgres","root");
		stm = con.createStatement();
		}
		catch(Exception e){
				e.printStackTrace();
		}
	}

	public void insertQuery(String query){
		startConnection();
		try {
			stm.executeUpdate(query);
		}catch (SQLException e){System.out.println("Erro na seleção: "
+ e.getMessage());}

	}

	public boolean updateQuery(String query){
		boolean test = false;
		try {
			int line = stm.executeUpdate(query);
			if (line > 0)
				test = true;
			else
				test = false;
		}catch (SQLException e){System.out.println("Erro na alteração:"
+ e.getMessage());}
		return test;
	}

	public boolean deleteQuery(String query){
		 boolean test = false;
			try {
				int line = stm.executeUpdate(query);
				if(line > 0)
					test = true;
				else
					test = false;
			}catch (SQLException e){System.out.println("Erro na
exclusão: " + e.getMessage());}
			return test;
		}

	public boolean selectQuery(String query){
		startConnection();
		boolean test = false;

		try {
			System.out.println("passei 1" + query);
			res = stm.executeQuery(query);
```

```
                System.out.println("passei 2" + query);
                if (res.next()){test = true;System.out.println("passei 3"
+ query);}
                else{test = false;System.out.println("passei 4" +
query);}

            }catch (SQLException e){System.out.println("Erro na seleção: "
+ e.getMessage());}

            return test;
      }


      public ResultSet getResultado() {
            return res;
      }


}
```

B.10 - ForeignKey.java

```
package conexaoSuperBaan;

import java.sql.Connection;
import java.sql.DatabaseMetaData;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.ArrayList;


public class ForeignKey{

      public Connection con;
      public Statement stm;
      public ResultSet res = null;
      private String banco_dados = null;

      public void setBancoDados(String banco_dados){
            this.banco_dados = banco_dados;
      }

      public String getBancoDados(){
            return banco_dados;
      }


      public void startConection(){
            String driver = "org.postgresql.Driver";
            String url = "jdbc:postgresql://localhost:5432/"+banco_dados;
            try{
            Class.forName(driver).newInstance();
            con = DriverManager.getConnection(url,"postgres","root");
            stm = con.createStatement();

            }
            catch(Exception e){
                        e.printStackTrace();
```

```
            }
      }

      public ArrayList checkForeingKey(ArrayList table){
        ArrayList resultAll = new ArrayList();
        ArrayList auxPK = new ArrayList();
        ArrayList auxFK = new ArrayList();
            try{
                  startConection();
                  DatabaseMetaData dmd = con.getMetaData();
                  ResultSet resultSetPK =
dmd.getPrimaryKeys(null,null,(String)table.get(0));
                        while(resultSetPK.next()){

      auxPK.add(resultSetPK.getString("COLUMN_NAME"));
                        }
                  resultAll.add(0,auxPK);

                  ResultSet resultSetFKC =
dmd.getImportedKeys(null,null,((String)table.get(0)));
                        while(resultSetFKC.next()){
                              ArrayList tempAux = new ArrayList();

      tempAux.add(0,resultSetFKC.getString("FKCOLUMN_NAME"));

      tempAux.add(1,resultSetFKC.getString("PKTABLE_NAME"));

      tempAux.add(2,resultSetFKC.getString("PKCOLUMN_NAME"));
                              auxFK.add(tempAux);
                        }
                  resultAll.add(1,auxFK);

            }catch(Exception e){e.printStackTrace();}
      return resultAll;
      }

      public String checkPKandFK(ArrayList table){ // pega os campos que
sao PK e FK
            String sql="";
              ArrayList auxPK = new ArrayList();
              ArrayList auxFK = new ArrayList();
              auxPK = (ArrayList)checkForeingKey(table).get(0);
              auxFK = (ArrayList)checkForeingKey(table).get(1);
                  for (int i=0; i<auxPK.size();i++){
                        for (int j=0;j < auxFK.size();j++){
                              if
(((String)auxPK.get(i)).equals((String)((ArrayList)auxFK.get(j)).get(0)) &&
((String)((ArrayList)auxFK.get(j)).get(1)).equals((String)table.get(1))){
                                    if (sql.equals("")){

      sql+=table.get(0)+"."+auxPK.get(i)+"="+table.get(1)+"."+((ArrayList)a
uxFK.get(j)).get(2);
                                    }else{sql+=" and
"+table.get(0)+"."+auxPK.get(i)+"="+table.get(1)+"."+((ArrayList)auxFK.get(
j)).get(2);}
                              }
                        }
                  }
                  System.out.print("AuxPK");
                  System.out.println(auxPK);
                  System.out.print("AuxFK");
```

```
                System.out.println(auxFK);
                System.out.println("\n\n\n");
                System.out.print("Tabela");
                System.out.println(table);
                System.out.print("SQL");
                System.out.println(sql);
          return sql;
          }

}
```

B.11 - SelectAllData.java

```
package conexaoSuperBaan;

import java.sql.Connection;

import java.sql.DriverManager;
import java.sql.ResultSet;

import java.sql.SQLException;
import java.sql.Statement;
import java.util.*;


public class SelectAllData{


      public Connection con;
      public Statement stm;
      public ResultSet res = null;



      public SelectAllData(){
            startConnection();
      }

      public void startConnection(){

            String driver = "org.postgresql.Driver";
            String url = "jdbc:postgresql://localhost:5432/Super Baan";
            try{
            Class.forName(driver).newInstance();
            con = DriverManager.getConnection(url,"postgres","root");
            stm = con.createStatement();

            }
            catch(Exception e){
                        e.printStackTrace();
            }
      }

      public ArrayList SelectAllDataBase(int CodUseCase, int CodPrj){
            ArrayList UseCase = new ArrayList();
            ResultSet selOne = null, selTwo=null, selThree=null;
            startConnection();
            for(int i = 0; i < 20; i++) {
                  if(i == 5 || i == 6) {
```

```
                        UseCase.add(i, new java.util.ArrayList());
                  } else {
                        UseCase.add(i, "");
                  }
            }
            try {
                  String SelUseCasePrj = "select * from
use_case_project,dbcliente where cod_use_case_prj="+CodUseCase+" and
cod_prj="+CodPrj+" and dbcliente.cod_dbcli= use_case_project.cod_dbcli";
                  selOne =  stm.executeQuery(SelUseCasePrj);
                  selOne.next();
                  UseCase.set(0,selOne.getString("tipo"));
                  UseCase.set(1,selOne.getString("nome"));
                  UseCase.set(2,selOne.getString("descricao"));
                  UseCase.set(3,selOne.getString("nome_banco"));
                  UseCase.set(11,selOne.getString("cod_dbcli"));
                  UseCase.set(4,selOne.getString("tabela"));
                  //-------------------------------------------------
                  String SelFields = "select * from campo_caso_uso where
cod_use_case_prj="+CodUseCase+" and cod_prj="+CodPrj;
                  selTwo = stm.executeQuery(SelFields);
                  while (selTwo.next()){

      ((ArrayList)UseCase.get(5)).add(selTwo.getString("campo"));

                  }
                  //-------------------------------------------------
                  String SelBusinessRules = "select * from
regras_de_negocio where cod_use_case_prj="+CodUseCase+" and
cod_prj="+CodPrj;
                  selThree = stm.executeQuery(SelBusinessRules);
                  ArrayList rules = new ArrayList();
                  while (selThree.next()){
                        ArrayList rulesTemp = new ArrayList();
                        rulesTemp.add(selThree.getString("nome"));
                        rulesTemp.add(selThree.getString("tabela1"));
                        rulesTemp.add(selThree.getString("campo1"));
                        rulesTemp.add(selThree.getString("tabela2"));
                        rulesTemp.add(selThree.getString("campo2"));
                        rulesTemp.add(selThree.getString("valor"));
                        rulesTemp.add(selThree.getString("propriedade"));

      rulesTemp.add(selThree.getString("cod_use_case_prj_acao_true"));

      rulesTemp.add(selThree.getString("valor_acao_true"));

      rulesTemp.add(selThree.getString("cod_use_case_prj_acao_false"));

      rulesTemp.add(selThree.getString("valor_acao_false"));

      rulesTemp.add(selThree.getString("campo_digitado"));
                        rules.add(rulesTemp);
                  }
                  UseCase.set(6,rules);
            }catch (SQLException e){System.out.println("Erro na seleção:" +
e.getMessage());}

            return UseCase;
      }

      public ResultSet getResultado() {
```

```
            return res;
        }

    }
```

APÊNDICE C – CÓDIGO FONTE PARA GERAÇÃO DA APLICAÇÃO

C.1 - home.jsp

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>Description Business Rules Tool - Geração</title>
</head>

<frameset rows="*" cols="220,*" framespacing="0" frameborder="NO"
border="0">
  <frame src="menu.jsp" name="leftFrame" scrolling="YES">
  <frame src="main.jsp" name="mainFrame">
</frameset>
<noframes><body>
</body></noframes>
</html>
```

C.2 - index.jsp

```
<%@ page contentType="text/html" language="java" import="java.sql.*"%>
<jsp:useBean id="conUser" class="conexaoSuperBaan.Conexao"/>

<%
String acao=request.getParameter("acao");
String mensagem = "";

if(acao != null) {

      if(acao.equals("Efetuar Login")) {
        conUser.setLogin(request.getParameter("login"));
        conUser.setSenha(request.getParameter("password"));

        if( conUser.startLogin() == true) {
          request.getSession(true);
            if (conUser.selectProjects() == true ) {
             java.util.ArrayList UseCase = new java.util.ArrayList();
             java.util.ArrayList Menu = new java.util.ArrayList();

            for(int i = 0; i < 20; i++) {
              if(i == 5 || i == 6) {
                  UseCase.add(i, new java.util.ArrayList());
            } else {
                  UseCase.add(i, "");
            }
```

```
        }
            session.setAttribute("USECASE",UseCase);
            int i = 0;
            ResultSet temp = conUser.getResultado();

            do{
            java.util.ArrayList mnuProj = new java.util.ArrayList();
            mnuProj.add(temp.getString("descricao_prj"));
            mnuProj.add(temp.getString("cod_prj"));
            Menu.add(i, mnuProj);
            }while (temp.next());
            session.setAttribute("MENU", Menu);
            response.sendRedirect("home.jsp");
      } else {
            mensagem = "Nao existe nenhum projeto para este usuario";
      }
      } else {
            mensagem = "Login ou senha invalidos";
      }

      }

}
%>
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/shtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pt" lang="pt">
<head>
<title>Login</title>
<style type="text/css">
<!--
@import url("common/common.css");
-->
</style>
</head>
<body>
      <div id="pagetitle">Login de Sistema</div>
      <div id="main">
            <%= mensagem %>
            <br />
            <div id="form">
                  <table class="tableform">
                  <tr>
                    <td>
                        <form name="login" method="post" >
                          <fieldset>
                            <legend>Login</legend>
                              <table class="tableform">
                              <tr>
                              <td class="descfield">Usuário:</td>
                              <td><input type="text" name="login"
id="login"/></td>
                              </tr>
                              <tr>
                              <td class="descfield">Password:</td>
      <td><input type="text" name="password" id="password"/></td>
      </tr>
      <tr><td> </td></tr>
      <tr><td colspan="2" class="tdline"></td></tr>
      <tr><td> </td></tr>
```

```
      <tr>
      <td colspan="2" class="fieldbuttons">
      <input type="submit" name='acao' value="Efetuar Login" />
      <input type="reset" value="Cancelar" />
                                                                  </td>
                                                            </tr>
                                                      </table>
                                                </fieldset>
                                          </form>
                                    </td>
                              </tr>
                        </table>
            </div>
            <br />
      </div>
</body>
</html>
```

C.3 - main.jsp

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/shtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pt" lang="pt">
<head>
<title>Description Business Rules Tool</title>
<style type="text/css">
<!--
@import url("common/common.css");
-->
</style>
</head>
<body>
      <div id="pagetitle">Description Business Rules Tool - Geração</div>
      <div id="main">
            Bem Vindo ao DBR Tool - Geração
      </div>
</body>
</html>
```

C.4 - menu.jsp

```
<%@ page contentType="text/html" language="java" import="java.sql.*"%>
<jsp:useBean id="conUser" class="conexaoSuperBaan.Conexao"/>

<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/shtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pt" lang="pt">
<head>
<title>Menu</title>
<style type="text/css">
<!--
@import url("common/menu.css");
-->
</style>
```

```
<script type="text/javascript" src="./menu/TreeMenu.js"></script>
<script type="text/javascript" src="./jscripts/functions.js"></script>
</head>
<body class="treeMenu">
<script type="text/javascript">
      objTreeMenu_1 = new TreeMenu("menu/images", "objTreeMenu_1", "_self",
"treeMenuDefault", true, false);

      newNode = objTreeMenu_1.addItem(new TreeNode('Geração', 'folder.gif',
'', true, true, '', 'detail', 'folder-expanded.gif', ''));

      newNode_1 = newNode.addItem(new TreeNode('Projetos', 'folder.gif',
'', true, true, '', 'detail', 'folder-expanded.gif', ''));
<%
      int i=0;
      java.util.ArrayList Menu = new java.util.ArrayList();
      Menu = (java.util.ArrayList) session.getAttribute("MENU");
      for(i = 0; i < Menu.size(); i++) {
            java.util.ArrayList mnuProj = new java.util.ArrayList();
            mnuProj =(java.util.ArrayList) Menu.get(i);
%>
            newNode_1_<%=i%> = newNode_1.addItem(new
TreeNode('<%out.print((String)mnuProj.get(0));%>', 'folder.gif', '', true,
true, '', 'detail', 'folder-expanded.gif', ''));
<%
            ResultSet temp;

      if(conUser.selectUseCase(Integer.parseInt((String)mnuProj.get(1)))) {

            int count = 2;
            temp = conUser.getResultado();
              do{
%>
                  newNode_2_<%=i%>_<%=count%> =
newNode_1_<%=i%>.addItem(new TreeNode('<%= (String)temp.getString(2) %>',
'../../images/usecase.png', 'javascript:goPage(' +
"'mountapp.jsp?prjcode=<%out.print((String)mnuProj.get(1));%>&usccode=<%=
(String)temp.getString(1) %>'" + ')', false, true, '', '_self', 'folder-
expanded.gif', ''));
<%
              }while (temp.next());
            }
      }
%>
      objTreeMenu_1.drawMenu();
      objTreeMenu_1.writeOutput();
      objTreeMenu_1.resetBranches();
</script>
</body>
</html>
```

C.5 - mountapp.jsp

```
<%@ page contentType="text/html" language="java" import="java.sql.*"%>
<jsp:useBean id="mountAppDB" class="conexaoSuperBaan.SelectAllData"/>
<jsp:useBean id="utilsBD" class="conexaoSuperBaan.executeQuery"/>
<jsp:useBean id="atrtibuteDB" class="conexaoSuperBaan.CheckAtribute"/>
<jsp:useBean id="foreignKeyDB" class="conexaoSuperBaan.ForeignKey"/>
<%
```

```
      java.util.ArrayList UseCase = new java.util.ArrayList();
      UseCase = (java.util.ArrayList) session.getAttribute("USECASE");

      String prjCode = request.getParameter("prjcode");
      String uscCode = request.getParameter("usccode");

if (prjCode != null && uscCode != null && prjCode.equals("") == false &&
uscCode.equals("") == false) {

      java.util.ArrayList GetedUseCase =
mountAppDB.SelectAllDataBase(Integer.parseInt(uscCode),
Integer.parseInt(prjCode));



      utilsBD.setBancoDados((String)GetedUseCase.get(3));
      atrtibuteDB.setBancoDados((String)GetedUseCase.get(3));
      foreignKeyDB.setBancoDados((String)GetedUseCase.get(3));

      // Trecho de processamento do formulário
      //_______------____________--------__________---------__________----
------_____________-----------__________----------
      String action = request.getParameter("action");
      if(action != null && action.equals("Insere") == true) {

            java.util.ArrayList rules =
(java.util.ArrayList)GetedUseCase.get(6);
            boolean resultadoRegra = true;
            String query = "SELECT * ";
            String From = " FROM ";
            String Where = " WHERE ";
            String MensagemVerdadeiro = " ";
            String MensagemFalso      = " ";
            String redirecionaVerdadeiro = " ";
            String redirecionaFalso      = " ";
            if(rules.size() > 0) {
             for(int i = 0; i < rules.size(); i++) {
      java.util.ArrayList ruleLine = (java.util.ArrayList)rules.get(i);
              query = "SELECT * ";
                if(((String)ruleLine.get(1)).equals("") == false) {
                  if(From.length() > 6) {
                    if(From.indexOf((String)ruleLine.get(1)) <= -1)
                        From += ", " + ruleLine.get(1);
                    } else {
                        From += ruleLine.get(1);
                  }
                 }
                if(((String)ruleLine.get(3)).equals("") == false) {
                  if(From.length() > 6) {
                    if(From.indexOf((String)ruleLine.get(3)) <= -1)
                        From += ", " + ruleLine.get(3);
                    } else
                        From += ruleLine.get(3);
                  }
            if(((String)ruleLine.get(3)).equals("") == false) {
              if( ((String)ruleLine.get(11)).equals("") == false) {
                if(Where.length() > 7)
                  Where += " AND " + ruleLine.get(1) + "." +
ruleLine.get(2) + " " + ruleLine.get(6) + " ";
                else
```

```
                  Where += " " + ruleLine.get(1) + "." + ruleLine.get(2) +
" " + ruleLine.get(6) + " ";
                  java.util.ArrayList proprertyAttribute =
                  atrtibuteDB.checkFields((String)ruleLine.get(1),
                  (String)ruleLine.get(2));
      if(proprertyAttribute.get(1).equals("int8"))
        Where +=  request.getParameter((String)ruleLine.get(11)) ;
      else
        Where += " '" + request.getParameter((String)ruleLine.get(11)) + "'
";
    } else {
       if(Where.length() > 7)
         Where += " AND " + ruleLine.get(1) + "." + ruleLine.get(2) + " " +
ruleLine.get(6) + " ";
      else
         Where += " " + ruleLine.get(1) + "." + ruleLine.get(2) + " " +
ruleLine.get(6) + " ";
         Where += " " + ruleLine.get(3) + "." + ruleLine.get(4);
      }
     } else {
            java.util.ArrayList proprertyAttribute =
            atrtibuteDB.checkFields((String)ruleLine.get(1),
            (String)ruleLine.get(2));
      if(((String)proprertyAttribute.get(1)).equals("int8") == true) {
        if(Where.length() > 7)
         Where += " AND " + ruleLine.get(1) + "." + ruleLine.get(2) + " " +
ruleLine.get(6) + " " + ruleLine.get(5);
        else
         Where += " " + ruleLine.get(1) + "." + ruleLine.get(2) + " " +
ruleLine.get(6) + " " + ruleLine.get(5);
      } else {
      if(Where.length() > 7)
        Where += " AND " + ruleLine.get(1) + "." + ruleLine.get(2) + " " +
ruleLine.get(6) + " '" + ruleLine.get(5) + "'";
      else
        Where += " " + ruleLine.get(1) + "." + ruleLine.get(2) + " " +
ruleLine.get(6) + " '" + ruleLine.get(5) + "'";
       }
      }
      java.util.ArrayList tmpFkPk = new java.util.ArrayList();
      tmpFkPk.add((String)ruleLine.get(1));
      tmpFkPk.add((String)ruleLine.get(3));
      String tmpString = "";
      tmpString = foreignKeyDB.checkPKandFK(tmpFkPk);
      if(tmpString.length() > 0)
      Where += " AND " + tmpString;
      if(ruleLine.get(7) == null) {
        if(MensagemVerdadeiro.length() > 0) {
            MensagemVerdadeiro += ", " + (String)ruleLine.get(8);
      } else
            MensagemVerdadeiro += (String)ruleLine.get(8);
      } else {
            redirecionaVerdadeiro = "mountapp.jsp?prjcode=" + prjCode +
"&usccode=" + ruleLine.get(7);
      }
      if(ruleLine.get(10) == null) {
        if(MensagemFalso.length() > 0) {
            MensagemFalso += ", " + (String)ruleLine.get(10);
        } else
            MensagemFalso += (String)ruleLine.get(10);
        } else
```

```
      redirecionaFalso = "mountapp.jsp?prjcode=" + prjCode + "&usccode=" +
ruleLine.get(9);
            }
      }

            query += From + Where;
            out.print(query);
            if(utilsBD.selectQuery(query)) {
                  if(MensagemVerdadeiro.length() > 0) {
                        out.print("<script language='javascript'>");
                        out.print("alert('" + MensagemVerdadeiro + "')");
                        out.print("</script>");
                  } else {
                        out.print("<script language='javascript'>");
                        out.print("alert('Verdadeiro')");
                        out.print("</script>");
                        response.sendRedirect(redirecionaVerdadeiro);
                  }
                  resultadoRegra = false;
            } else {
                  if(MensagemFalso.length() > 0) {
                        out.print("<script language='javascript'>");
                        out.print("alert('" + MensagemFalso + "')");
                        out.print("</script>");
                  } else {
                        out.print("<script language='javascript'>") ;
                        out.print("alert('Falso')");
                        out.print("</script>");
                        response.sendRedirect(redirecionaFalso);
                  }
            }

      if(resultadoRegra == true) {
            String queryInsert = "INSERT INTO " + GetedUseCase.get(4) + "
(";
            java.util.ArrayList fields =
(java.util.ArrayList)GetedUseCase.get(5);
            for(int i =0; i < fields.size(); i++) {
                  if(i == 0) {
                        queryInsert += fields.get(i);
                  } else
                        queryInsert += ", " + fields.get(i);
            }
            queryInsert += ") VALUES (";
            for(int i =0; i < fields.size(); i++) {
                  java.util.ArrayList proprertyAttribute =
atrtibuteDB.checkFields((String)GetedUseCase.get(4),
(String)fields.get(i));
                  String value = "";
                  String valueField =
request.getParameter((String)fields.get(i));
                  if(valueField == null || valueField.length() <= 0)
                        value = "null";
                  else {

      if(((String)proprertyAttribute.get(1)).equals("int8") == true) {
                              value = valueField;
                        } else {
                              value = "'" + valueField + "'";
                        }
                  }
```

```
                  if(i == 0)
                        queryInsert += value;
                  else
                        queryInsert += ", " + value;
            }
            queryInsert += ")";
            utilsBD.insertQuery(queryInsert);
      }
      }

      //_______------____________--------__________---------__________----
------_____________-----------__________----------
%>
<html>
<head>
<title>Caso de Uso <%= GetedUseCase.get(1) %></title>
<style type="text/css">
<!--
@import url("common/common.css");
@import url("common/tabs.css");
-->
</style>
<script language="JavaScript">
<!--
function validaForm() {
      var err = false;
      formulario = document.getElementById("processUseCase");
<%
      java.util.ArrayList fieldsjs =
(java.util.ArrayList)GetedUseCase.get(5);
      for(int i = 0; i < fieldsjs.size(); i++) {
            java.util.ArrayList proprertyAttribute =
atrtibuteDB.checkFields((String)GetedUseCase.get(4),
(String)fieldsjs.get(i));
            if(((String)proprertyAttribute.get(3)).equals("false")) {
                  out.print("if(formulario."+fieldsjs.get(i)+".value ==
\"\") {\n");
                  out.print("alert(\""+fieldsjs.get(i)+" vazio\");");
                  out.print("return false;");
                  out.print("}");
            }
      }
%>

      return !err;
}
function openFind(url)
{
      var windowProps =
'statusbar=no,menubar=no,copyhistory=no,location=no,resizable=no,toolbar=no
,scrollbars=yes,width=600,height=600';
      newwin = window.open(url, '', windowProps);
}
-->
</script>
</head>
<body>
      <div id="pagetitle">Caso de Uso <%= GetedUseCase.get(1) %></div>
      <div id="main">
            <br /><br />
<%
```

```
      pageContext.include("tabbar.jsp");
%>
            <br /><br />
            <div id="form">
                  <table class="tableform">
                        <tr>
                              <td>
<%

      if(((String)GetedUseCase.get(0)).equals("0")) {
            java.util.ArrayList fields =
            (java.util.ArrayList)GetedUseCase.get(5);
            String stringFields = "";
%>
            <table border="0" class="tableform">
            <tr class="relatdescfields">
                  <%
           for(int i =0; i < fields.size(); i++) {
             out.print("<td class=\"descfield\">"+fields.get(i)+"</td>");
            if(i == 0)
             stringFields += GetedUseCase.get(4) + "." + fields.get(i);
            else
             stringFields += ", " + GetedUseCase.get(4) + "." +
fields.get(i);
            }
            %>
            </tr>
            <%
            java.util.ArrayList rules =
            (java.util.ArrayList)GetedUseCase.get(6);
            String query = "";
            if(rules.size() <= 0) {
             query = "SELECT " + stringFields + " FROM " +
((String)GetedUseCase.get(4));
            } else {
             query = "SELECT " + stringFields;
             String From = " FROM " + ((String)GetedUseCase.get(4));
            String Where = " WHERE ";
           for(int i = 0; i < rules.size(); i++) {
            java.util.ArrayList ruleLine =
            (java.util.ArrayList)rules.get(i);
      if(((String)ruleLine.get(1)).equals(((String)GetedUseCase.get(4))) ==
false && ((String)ruleLine.get(1)).equals("")== false) {
            if (From.indexOf((String)ruleLine.get(1)) == -1)
              From += ", " + ruleLine.get(1);
            java.util.ArrayList tmpFkPk = new java.util.ArrayList();
            tmpFkPk.add((String)GetedUseCase.get(4));
            tmpFkPk.add((String)ruleLine.get(1));
            String tmpString = "";
            if(Where.length() <= 7) {
              tmpString = foreignKeyDB.checkPKandFK(tmpFkPk);
              if(tmpString.length() > 0)
                  Where += tmpString;
            } else {
            tmpString = foreignKeyDB.checkPKandFK(tmpFkPk);
            if(tmpString.length() > 0)
             Where += " AND " + tmpString;
            }
          }

      if(((String)ruleLine.get(3)).equals(((String)GetedUseCase.get(4))) ==
```

```
false && ((String)ruleLine.get(3)).equals("")== false &&
((String)ruleLine.get(3)).equals(((String)ruleLine.get(1))) == false) {
      if (From.indexOf((String)ruleLine.get(3)) == -1)
        From += ", " + ruleLine.get(3);
        java.util.ArrayList tmpFkPk = new java.util.ArrayList();
        tmpFkPk.add((String)GetedUseCase.get(4));
        tmpFkPk.add((String)ruleLine.get(3));
        String tmpString = "";
        if(Where.length() <= 7) {
        tmpString = foreignKeyDB.checkPKandFK(tmpFkPk);
        if(tmpString.length() > 0)
        Where += tmpString;
      } else {
      tmpString = foreignKeyDB.checkPKandFK(tmpFkPk);
      if(tmpString.length() > 0)
        Where += " AND " + tmpString;
      }
     }

      if(((String)ruleLine.get(5)).equals("") == false) {
        if(Where.length() <= 7) {
            Where += ((String)ruleLine.get(1)) + "." +
            ((String)ruleLine.get(2)) + " " + ((String)ruleLine.get(6)) + "
            " + ((String)ruleLine.get(5));
      } else {
            Where += " AND " + ((String)ruleLine.get(1)) + "." +
            ((String)ruleLine.get(2)) + " " + ((String)ruleLine.get(6)) + "
            " + ((String)ruleLine.get(5));
      }
   }
 }
 query += From + Where;
}
      out.print(query);
      if(utilsBD.selectQuery(query)) {
        ResultSet temp = utilsBD.getResultado();
        do {
            %>
            <tr>
            <%
            for(int i =0; i < fields.size(); i++) {
         out.print("<td>"+temp.getString((String)fields.get(i))+"</td>");
        }
            %>
      </tr>
      <%
      }while(temp.next());
      }
      %>
      </table>
<%
      } else {
%>
      <form name="processUseCase" id="processUseCase" onSubmit="return
      validaForm()" method="post">
      <input type="hidden" name="table" id="table" value="<%=
      GetedUseCase.get(4) %>">
      <input type="hidden" name="prjCode" id="prjCode" value="<%= prjCode
      %>">
      <input type="hidden" name="uscCode" id="uscCode" value="<%= uscCode
      %>">
```

```
      <fieldset>
        <legend>Caso de Uso <%= GetedUseCase.get(1) %></legend>
          <table border="0" class="tableform">
            <tr>
              <Td colspan="2"> </Td>
            </tr>
            <%
            java.util.ArrayList fields =
            (java.util.ArrayList)GetedUseCase.get(5);
            for(int i = 0; i < fields.size(); i++) {
                  java.util.ArrayList proprertyAttribute =
                  atrtibuteDB.checkFields((String)GetedUseCase.get(4),
                  (String)fields.get(i));
            %>
            <tr>
            <td class="descfield"><%= fields.get(i) %></td>
            <td><input type="text" name="<%= fields.get(i) %>" id="<%=
            fields.get(i) %>" <%
            if(((String)proprertyAttribute.get(1)).equals("varchar"))
            out.print("size=\""+proprertyAttribute.get(2)+"\""); %>/>
            <%
            java.util.ArrayList keytemp =
            (java.util.ArrayList)proprertyAttribute.get(4);
      if(((java.util.ArrayList)proprertyAttribute.get(4)).size() > 0 )

      out.print("<A href='#' onClick=\"openFind('search.jsp?key=" +
keytemp.get(2) + "&banco=" + GetedUseCase.get(3) + "&table=" +
keytemp.get(1) + "&field=" + keytemp.get(0) + "')\">Busca</a>");
                                                            %>
                              </td>
                        </tr>
                        <%
                        }
                  %>
            <tr>
            <td colspan="2" align="center">
            <input type="submit" name="action" id="action" value="Insere"
            /> 
            <input type="reset" value="Limpar" />
            </td>
            </tr>

                                    </table>
                                    </fieldset>
                              </form>
<%
                                    }
%>
                              </td>
                        </tr>
                  </table>
            </div>
            <br />
      </div>
</body>
</html>
<%
} else {
      out.print("Códigos não selecionados");
}
%>
```

C.6 - search.jsp

```
<%@ page contentType="text/html" language="java" import="java.sql.*"%>
<jsp:useBean id="selDBField" class="conexaoSuperBaan.ChangeFieldDB"/>
<jsp:useBean id="utilsBD" class="conexaoSuperBaan.executeQuery"/>
<%
      java.util.ArrayList UseCase = new java.util.ArrayList();
      UseCase = (java.util.ArrayList) session.getAttribute("USECASE");

      utilsBD.setBancoDados((String)request.getParameter("banco"));
      selDBField.setBancoDados((String)request.getParameter("banco"));
      selDBField.setBanco_DB((String)request.getParameter("table"));

      int index =0;

%>
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/shtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pt" lang="pt">
<head>
<title>Busca</title>
<style type="text/css">
<!--
@import url("common/common.css");
@import url("common/tabs.css");
-->
</style>
<script language="javascript">
<!--
function selClick(val, field) {
      window.opener.document.getElementById(field).value = val;
      window.close();
}
-->
</script>
</head>
<body>
      <div id="pagetitle">Busca</div>
      <div id="main">
            <br /><br />
<%
      pageContext.include("tabbar.jsp");
%>
            <br /><br />
            <div id="form">
                  <table class="tableform">
                    <tr>
                     <td>
<%                      String stringFields = "";
%>
                         <table border="0" class="tableform">
                         <tr class="relatdescfields">
            <%
            ResultSetMetaData temp2 = selDBField.startChangeFields();
            java.util.ArrayList fields = new java.util.ArrayList();
            int numberOfColumns = temp2.getColumnCount();
```

```
            for(int i =1; i <= numberOfColumns; i++) {
              fields.add(temp2.getColumnName(i));
    out.print("<td class=\"descfield\">"+temp2.getColumnName(i)+"</td>");
            if(
            ((String)temp2.getColumnName(i)).equals((String)request.getPara
            meter("key")))
              index = i;
              if(i == 1)
                  stringFields += request.getParameter("table") + "." +
                  temp2.getColumnName(i);
              else
                  stringFields += ", " + request.getParameter("table") +
                  "." + temp2.getColumnName(i);
              }
                                    %>
                          </tr>
                                    <%
            String query = "SELECT " + stringFields + " FROM " +
            request.getParameter("table");
                  if(utilsBD.selectQuery(query)) {
                    ResultSet temp = utilsBD.getResultado();
            do {
                  %>
                  <tr>
                  <%
            for(int i =0; i < fields.size(); i++) {
             if(i == (index-1))
                  out.print("<td><a href='#' onClick=\"selClick('" +
                  temp.getString((String)fields.get(i)) + "','" +
                  request.getParameter("field") +
                  "')\">"+temp.getString((String)fields.get(i))+"<a></td>")
                  ;
             else
        out.print("<td>"+temp.getString((String)fields.get(i))+"</td>");
            }
                  %>
                  </tr>
                  <%

            }while(temp.next());
            }
                                        %>
                                    </table>
                                </td>
                          </tr>
                  </table>
            </div>
            <br />
      </div>
</body>
</html>
```

APÊNDICE D – SCRIPT DO BANCO DE DADOS DA FERRAMENTA BRD TOOL

```
CREATE TABLE campo_caso_uso
(
  cod_use_case_prj int4 NOT NULL,
  campo varchar(20) NOT NULL,
  cod_prj int8 NOT NULL,
  ordem int4,
  descricao_campo text,
  CONSTRAINT campo_caso_uso_pkey PRIMARY KEY (cod_use_case_prj, cod_prj,
campo),
  CONSTRAINT campo_caso_uso_cod_use_case_prj_fkey FOREIGN KEY
(cod_use_case_prj, cod_prj) REFERENCES use_case_project (cod_use_case_prj,
cod_prj) ON UPDATE RESTRICT ON DELETE RESTRICT
)
WITHOUT OIDS;
ALTER TABLE campo_caso_uso OWNER TO postgres;


CREATE TABLE dbcliente
(
  cod_dbcli serial NOT NULL,
  servidor varchar(15),
  usuario_meta varchar(50),
  senha_meta varchar(50),
  usuario_db varchar(50),
  senha_db varchar(50),
  nome_banco varchar(70),
  sgbd_cli varchar(50),
  CONSTRAINT dbcliente_pkey PRIMARY KEY (cod_dbcli)
)
WITHOUT OIDS;
ALTER TABLE dbcliente OWNER TO postgres;

CREATE TABLE projeto
(
  cod_prj int8 NOT NULL,
  descricao_prj varchar(100),
  CONSTRAINT projeto_pkey PRIMARY KEY (cod_prj)
)
WITHOUT OIDS;
ALTER TABLE projeto OWNER TO postgres;

CREATE TABLE regras_de_negocio
(
  cod_regras_negocio serial NOT NULL,
  cod_use_case_prj int4 NOT NULL,
  cod_prj int8 NOT NULL,
  nome varchar(30),
  ordem int8,
```

```
  tabela1 varchar(30),
  campo1 varchar(30),
  propriedade varchar(5),
  valor varchar(100),
  tabela2 varchar(30),
  campo2 varchar(30),
  cod_use_case_prj_acao_true int4,
  cod_prj_acao_true int8,
  valor_acao_true text,
  cod_use_case_prj_acao_false int4,
  cod_prj_acao_false int8,
  valor_acao_false text,
  campo_digitado varchar(30),
  CONSTRAINT regras_de_negocio_pkey PRIMARY KEY (cod_regras_negocio,
cod_use_case_prj, cod_prj),
  CONSTRAINT fk1 FOREIGN KEY (cod_use_case_prj, cod_prj) REFERENCES
use_case_project (cod_use_case_prj, cod_prj) ON UPDATE RESTRICT ON DELETE
RESTRICT,
  CONSTRAINT fk2 FOREIGN KEY (cod_use_case_prj_acao_true,
cod_prj_acao_true) REFERENCES use_case_project (cod_use_case_prj, cod_prj)
ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT fk3 FOREIGN KEY (cod_use_case_prj_acao_false,
cod_prj_acao_false) REFERENCES use_case_project (cod_use_case_prj, cod_prj)
ON UPDATE RESTRICT ON DELETE RESTRICT
)
WITHOUT OIDS;
ALTER TABLE regras_de_negocio OWNER TO postgres;

CREATE TABLE use_case_project
(
  cod_use_case_prj serial NOT NULL,
  cod_prj int8 NOT NULL,
  tipo int8,
  nome varchar(30),
  descricao text,
  cod_dbcli int4 NOT NULL,
  tabela varchar(30),
  CONSTRAINT use_case_project_pkey PRIMARY KEY (cod_use_case_prj, cod_prj),
  CONSTRAINT use_case_project_bdcliente FOREIGN KEY (cod_dbcli) REFERENCES
dbcliente (cod_dbcli) ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT use_case_project_cod_prj_fkey FOREIGN KEY (cod_prj) REFERENCES
projeto (cod_prj) ON UPDATE RESTRICT ON DELETE RESTRICT
)
WITHOUT OIDS;
ALTER TABLE use_case_project OWNER TO postgres;


CREATE TABLE user_project
(
  login varchar(50) NOT NULL,
  cod_prj int8 NOT NULL,
  CONSTRAINT user_project_pkey PRIMARY KEY (login, cod_prj),
  CONSTRAINT user_project_cod_prj_fkey FOREIGN KEY (cod_prj) REFERENCES
projeto (cod_prj) ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT user_project_login_fkey FOREIGN KEY (login) REFERENCES usuario
(login) ON UPDATE RESTRICT ON DELETE RESTRICT
)
WITHOUT OIDS;
ALTER TABLE user_project OWNER TO postgres;
```

```
CREATE TABLE usuario
(
  login varchar(50) NOT NULL,
  nome varchar(50),
  sobrenome varchar(50),
  senha varchar(50),
  email varchar(80),
  tipo varchar(13),
  CONSTRAINT usuario_pkey PRIMARY KEY (login)
)
WITHOUT OIDS;
ALTER TABLE usuario OWNER TO postgres;
```

APÊNDICE E – SCRIPT DO BANCO DE DADOS DO ESTUDO DE CASO

```
CREATE TABLE aluno
(
  cod_aluno int8 NOT NULL,
  nome varchar(50),
  idade int4,
  CONSTRAINT aluno_pkey PRIMARY KEY (cod_aluno)
)
WITHOUT OIDS;
ALTER TABLE aluno OWNER TO postgres;


CREATE TABLE curso
(
  cod_curso int8 NOT NULL,
  descricao varchar(50) NOT NULL,
  dia_semana varchar(1) NOT NULL,
  horario time NOT NULL,
  CONSTRAINT curso_pkey PRIMARY KEY (cod_curso)
)
WITHOUT OIDS;
ALTER TABLE curso OWNER TO postgres;

CREATE TABLE rel_aluno_curso
(
  codigo_aluno int8 NOT NULL,
  codigo_curso int8 NOT NULL,
  CONSTRAINT rel_aluno_curso_pkey PRIMARY KEY (codigo_aluno, codigo_curso),
  CONSTRAINT rel_aluno_curso_codigo_aluno_fkey FOREIGN KEY (codigo_aluno)
REFERENCES aluno (cod_aluno) ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT rel_aluno_curso_codigo_curso_fkey FOREIGN KEY (codigo_curso)
REFERENCES curso (cod_curso) ON UPDATE RESTRICT ON DELETE RESTRICT
)
WITHOUT OIDS;
ALTER TABLE rel_aluno_curso OWNER TO postgres;
```