UNIVERSIDADE FEDERAL DE SANTA CATARINA

HOPI

Um Editor de Requisitos e
Casos de Uso para a
Modelagem de Sistemas

Marcel Horner
Rafael de Andreis Pires

Florianópolis - SC
2004

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

HOPI

Um Editor de Requisitos e
Casos de Uso para a
Modelagem de Sistemas

Marcel Horner
Rafael de Andreis Pires

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

Florianópolis - SC
2004

Marcel Horner
Rafael de Andreis Pires

HOPI
Um Editor de Requisitos e
Casos de Uso para a
Modelagem de Sistemas

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

_____
Raul Sidnei Wazlawick
Orientador

_____
Antonio Carlos Mariani
Membro da Banca

_____
Ricardo Pereira e Silva
Membro da Banca

Florianópolis, 08 de dezembro de 2004

Dedicamos este trabalho aos nossos pais,
amigos e familiares que nos apoiaram sempre
durante o caminho que trilhamos até aqui.

Agradecimentos

Agradecemos a toda a população brasileira por suportar a Universidade em que estudamos e tudo o que está envolvido em seu funcionamento e aos professores realmente bons que tivemos em nossa vida acadêmica, àqueles que levam a sério e se dedicam ao que fazem e que foram os responsáveis pela formação profissional que temos hoje.

Sumário

Lista de Figuras

Resumo

Este trabalho consiste na implementação de uma ferramenta CASE para a análise de requisitos e definição de casos de uso de sistemas.

Estão descritos os testes realizados com as outras ferramentas deste tipo, juntamente com a modelagem para o novo sistema, o HOPI. Esta modelagem utiliza a metodologia proposta por [Larman2000]. A implementação foi baseada em J2EE para acesso via web da ferramenta.

Palavras–chave: requisito, caso de uso, Java, J2EE e design patterns.

Abstract

This document describes the process of implementation of a CASE tool for requirements analysis and use cases definitions of systems.

The tests that were made with other tools of this kind are described together with the modeling of the new system, the HOPI. This modeling uses [Larman2000] proposed methodology. The implementation was based on J2EE for web access of the tool.

Keywords: requirement, use case, Java, J2EE and design pattern.

Introdução

Com o constante avanço na tecnologia acerca dos componentes de hardware aumenta a capacidade de processamento dos computadores modernos. Isso induz também a possibilidade do desenvolvimento de softwares mais poderosos e complexos.

Para lidar com tal complexidade e possibilitar a realização da implementação e manutenção de sistemas de forma eficiente muito se tem investido em pesquisas na área de engenharia de software.

Parte importante do processo de desenvolvimento consiste na elaboração da documentação de todas as etapas, desde o levantamento de requisitos até a manutenção do sistema já implantado. Esta documentação visa proporcionar um maior entendimento sobre o sistema com relação a questões como quais funcionalidades o mesmo deve atender e a maneira como está internamente estruturado.

Com as descobertas e invenções nessa área logo surgiram ferramentas computacionais para auxiliar os analistas de sistemas na elaboração do projeto, são as chamadas ferramentas CASE – Computer-Aided Software Engineering (Engenharia de Software Auxiliada por Computador). E é com esse propósito que apresentamos o HOPI.

Objetivos

Implementar uma ferramenta CASE que suporte as etapas de levantamento de requisitos e definição de casos de uso da modelagem de sistemas.

Justificativa

Apesar da inúmera quantidade de ferramentas CASE existentes no mercado atualmente, não conseguimos encontrar nenhuma que trate de requisitos e casos de uso da forma como gostaríamos.

Com base nas ferramentas que tivemos a possibilidade de utilizar observamos que o seu principal enfoque é com relação aos diagramas UML e a geração automática de código a partir destes elementos. Não estamos questionando a utilidade destas funcionalidades, mas sim a falta de suporte à elaboração da parte textual da documentação que precede estas etapas.

A definição correta e clara acerca dos requisitos de um sistema no início do processo evita desentendimentos entre analistas, projetistas e desenvolvedores. É fundamental estabelecer qual é o escopo do projeto, definindo quais funcionalidades o sistema gerado deve suportar e quais as suas limitações. Esta prática possibilita que toda a equipe envolvida no processo almeje um objetivo comum, sem divergências e evitando que sejam gastos esforços na concepção de elementos que não condizem com os esperados para o sistema a ser implementado.

Na área comercial a análise de requisitos contribui também para a elaboração de contratos de desenvolvimento de sistemas. Não é difícil encontrar empresas que já tenham passado por alguma experiência desagradável com clientes em virtude de contratos mal redigidos. O que normalmente acontece é que o sistema não fica bem especificado, permitindo ao cliente solicitar alterações após a conclusão da modelagem e da implementação. Se nada foi definido no contrato o cliente está no seu direito ao solicitar alterações que atendam as suas expectativas.

Além disso, é interessante manter o relacionamento entre requisitos e casos de uso. Ao observarmos a ausência de casos de uso relacionados podemos identificar quais requisitos ainda não foram atendidos. Quando observamos os requisitos relacionados ao caso de uso podemos compreender mais claramente a sua funcionalidade.

É em virtude destes motivos que propusemos a implementação da ferramenta apresentada nesta monografia, denominada HOPI.

Metodologia

Para a criação do trabalho foi feita uma pesquisa para enumerar as ferramentas de análise existentes. Foram realizados testes em cada uma delas para identificar os pontos positivos e as deficiências de cada uma, bem como para nos certificarmos de que realmente não existe nenhuma ferramenta similar a que estamos propondo.

Depois desta etapa de pesquisa realizamos a análise e o projeto da ferramenta. O primeiro passo foi identificar os requisitos, ou seja, o que esperamos que o sistema faça. Com base nestas informações elaboramos os casos de uso que descrevem como seria o uso do sistema para atender as funcionalidades estabelecidas, como também o seu relacionamento com elementos externos a ele. A partir da observação dos casos de uso foram identificadas as entidades que compõe o sistema e então criado um modelo conceitual para a visualização destes relacionamentos.

Foram definidos os contratos das operações básicas do sistema como: inclusão, alteração e remoção de registros.

Depois foram gerados os diagramas de colaboração que descrevem o funcionamento interno de entidades da solução em computador bem como o que alimenta as operações e o que obtemos após o processo. O último passo do projeto da aplicação foi a geração do diagrama de classes que descreve a estrutura do sistema sob o ponto de vista da solução computacional.

Realizamos a implementação do editor com base nos artefatos de análise gerados bem como nos conhecimentos obtidos durante este processo.

Por fim apresentamos os resultados obtidos e as conclusões acerca do que foi realizado. Outro ponto apresentado nas considerações consiste na enumeração das sugestões para os trabalhos futuros que apontam o que pode ser melhorado no sistema gerado e também o que ainda pode ser desenvolvido daqui para frente.

Revisão Bibliográfica

Para uma melhor definição de quais características devem ser seguidas e quais devem ser evitadas foi realizado um estudo sobre as ferramentas CASE existentes no mercado atualmente.

Ideogramic UML v2.3.3 – Em se tratando de levantamento de requisitos e identificação de casos de uso esta ferramenta é bastante fraca. Não apresenta nenhum tipo de auxílio quanto à etapa de levantamento de requisitos. Possui apenas diagramas de classe, de seqüência, de estados e de casos de uso. Os casos de uso nesta ferramenta são apenas um nome, não existindo espaço para nenhum tipo de descrição ou detalhamento. Além das notas (que podem ser incluídas nos diagramas como um elemento) o único espaço onde é possível escrever um texto explicativo é um campo de formatação livre relativo ao projeto como um todo.

Magic Draw UML v7.2 – Esta ferramenta não abrange a fase de análise de requisitos, e conseqüentemente as referências cruzadas entre os mesmos com os casos de uso. O seu principal enfoque está nos diagramas: de Atividades, de Casos de Uso, de Classe, de Colaboração, de Conteúdo, de Estado, de Implementação, de Robustez, de Seqüência e de Web. A ferramenta não auxilia o usuário através do processo de análise e projeto. O usuário tem a liberdade de criar o diagrama que desejar em qualquer ordem, sem muita relação entre eles. Para navegar pelos diversos elementos gerados temos uma árvore similar a do Windows Explorer. Os componentes dos diagramas de seqüência, de colaboração, de atividades e de estados ficam como ramos do caso de uso. Já os demais diagramas criam seus componentes na raiz e não como ramos próprios. Isto torna a navegação um tanto confusa. Apesar disto os diagramas são de fácil criação e possuem uma aparência visual bastante agradável. Outro ponto positivo é a barra com a escala de zoom e o quadrado de navegação pelo diagrama que estamos visualizando.

Visual Paradigm for UML Modeler Edition v3.1 – Esta ferramenta possibilita a criação de diagramas de Casos de Uso, de Classe, de Seqüência, de

Colaboração, de Estado, de Atividade, de Componente e de Implantação. Apresenta ainda uma análise textual, onde é possível identificar classes, atores e casos de uso candidatos, o "workflow" do negócio e diagrama de Cartões CRC. A análise de requisitos não é abordada pela ferramenta.

Poseidon for UML Professional Edition v2.2 – Nesta ferramenta não podemos encontrar nenhum tipo de suporte à análise de requisitos e nem à identificação dos casos de uso. Permite a criação de diagramas de Classe, de Casos de Uso, de Estado, de Atividades, de Colaboração, de Seqüência e de Implantação / Objeto / Componente. Documentação em formato de texto livre pode ser inserida em qualquer elemento de qualquer diagrama. Porém ao tentarmos navegar pelos elementos com esta documentação visível percebemos que a ferramenta torna-se extremamente lenta, o que é bastante irritante.

ArgoUML v0.14 – A navegação é bastante ruim, do tipo árvore, onde elementos de um diagrama localizam-se no mesmo nível do diagrama propriamente dito. A ferramenta desconsidera a existência da etapa de análise de requisitos, apresentando apenas as opções de criação de diagramas de Classe, Casos de Uso, Estado, Atividade, Colaboração, Implantação e Seqüência. Cada elemento dos diagramas possui uma documentação relacionada, onde consta o autor, a versão, a data, um check para se foi descontinuado, um campo com o título "See" e uma parte de texto livre para a documentação propriamente dita.

Visual Case 2 v2.6.5 – Esta ferramenta possibilita a geração dos diagramas de Caso de Uso, Classes, Seqüência, Colaboração, Estados, Atividades, Implantação, Componentes, de Fluxo e de Fluxo de Dados. Possui ainda um módulo para criação do banco de dados de forma visual. A navegação pelo sistema é através de uma árvore. Não apresenta nenhum tipo de suporte a parte de levantamento de requisitos nem de nenhuma documentação escrita.

Software through Pictures v8.3 – A ferramenta possibilita a geração de diagramas de Caso de Uso, Classe, Seqüência, Colaboração, Estado, Atividade, Componente, Implantação e de Esteriótipo. A ferramenta apresenta uma parte com tabelas onde podemos incluir os requisitos. Estes, entretanto, não são

relacionados a nenhum caso de uso. O único relacionamento existente entre requisitos e casos de uso é que o nome do caso de uso pode ser escolhido a partir de um requisito, mas é só. A versão trial do sistema possui um mecanismo de verificação de licença horrível. Para cada ação que se desejamos fazer é necessário escolhermos o arquivo de licença que acessa um servidor para verificações.

UML Diagrammer v4.16 – Com esta ferramenta é possível criar todos os tipos de diagramas, porém não possui qualquer tipo de auxílio nesta tarefa, parecendo mais um simples editor gráfico com alguns modelos próprios para UML. Não possui qualquer suporte ao levantamento de requisitos.

UML Studio v7.1 – Novamente uma ferramenta sem suporte ao levantamento de requisitos. Possui alguns templates que auxiliam no desenvolvimento da análise, separação de cada fase de análise, e três funcionalidades interessantes: geração de código (Java, C++), geração de documentação e engenharia reversa.

Enterprise Architect v3.60 – Ferramenta completa, possui suporte ao levantamento de requisitos e conta com diagramas de casos de uso. Possibilita a confecção de qualquer tipo de diagrama UML, geração de código, documentação, engenharia reversa, esquemas de BD, faz limitação de nível de acesso por usuário, e navegação rápida entre os elementos do projeto. Não podemos considerar como sendo o editor de requisitos e casos de uso que buscamos pois não é possível detalhar os passos dos cenários principal, alternativo e seus relacionamentos.

BridgePoint Development Suíte – Não possui suporte à fase de requisitos. Trabalha com um conceito de UML executável que possibilita além da automatização da geração de código, executá-lo a partir das especificações UML, com possibilidade inclusive de inserção de breakpoints e depuração. Possibilita análise em ambiente colaborativo por ter suporte a múltiplos usuários em rede, além de possibilitar a reutilização de diagramas em forma de módulos entre projetos.

Proxy Source Design – Ferramenta gratuita, que possui suporte apenas a Diagramas de Classe e Casos de Uso. Extremamente fraca.

Estrutura do Documento

A estrutura deste documento bem como o conteúdo existente em cada capítulo ficou assim estabelecido:

Capitulo 1 – Definição do Sistema

Nesta etapa definimos quais serão as funcionalidades do sistema HOPI proposto, gerando os Requisitos. Além disso, descrevemos os processos e a interação entre os usuários e o sistema HOPI para que seja possível a realização das atividades desejadas. Desta forma são identificados os Casos de Uso

Capitulo 2 – Análise

A partir dos artefatos de análise gerados no Capítulo 1 seguimos com a especificação do sistema a ser criado. Expandimos os casos de uso identificando mais algumas características que serão úteis para a geração do modelo conceitual e dos contratos de operações do sistema, sendo estes gerados ainda neste capítulo. O modelo conceitual apresenta as entidades envolvidas no problema e seus relacionamentos.

Capitulo 3 – Projeto

O projeto é a última etapa antes da implementação do sistema HOPI. É neste momento que obtemos os diagramas de colaboração que descrevem o funcionamento interno do sistema HOPI bem como a interação das entidades que o compõem. Finalmente, após gerar todos estes artefatos podemos definir o diagrama de classes do sistema HOPI. Este representa, com detalhes, a estrutura interna do editor proposto, com as classes que o compõe, seus atributos e métodos.

Capitulo 4 – Implementação

Neste capítulo podemos encontrar algumas definições acerca das tecnologias envolvidas na implementação do sistema HOPI. Apresentamos também alguns relatos sobre o que foi realizado e algumas observações relevantes.

Capítulo 5 – Resultados

Colocamos aqui as observações acerca do sistema HOPI obtido em comparação ao que era esperado. Definimos se os objetivos puderam ser alcançados ou não e quais foram os pontos fortes e fracos da aplicação.

Capitulo 6 – Conclusão

Apresentamos as considerações acerca do processo de análise, projeto e implementação da ferramenta proposta e alguns apontamentos com relação a aspectos que podem ser melhorados no sistema gerado bem como possíveis implementações a serem realizadas em trabalhos futuros.

Capítulo 1 - Definição do Sistema

A primeira etapa do projeto consiste na definição do sistema HOPI. É necessário identificar bem claramente o que este deverá ou não fazer.

1.1 Requisitos

O sistema HOPI deve permitir o cadastramento de usuários. Este cadastramento visa controlar o acesso e manter um histórico das modificações realizadas por cada um deles. Deve ser possível conceder o acesso de administrador para os usuários desejados.

Usuários com o perfil de administrador devem poder cadastrar novos projetos, usuários e categorias, bem como modificar a lista de acesso de cada projeto. Deve ser possível incluir ou remover usuários da lista de leitores ou escritores em um projeto.

Um usuário com o perfil de editor pode modificar um projeto criando, visualizando, modificando ou removendo requisitos funcionais, não-funcionais, atores, casos de uso, passos do caso de uso, variantes e fluxos alternativos, que por sua vez, também são formados por passos. Os leitores devem poder visualizar todas as partes do projeto não podendo, porém, incluir, modificar nem remover nenhum dos itens citados.

Os requisitos funcionais do projeto correspondem a uma lista das funcionalidades que o sistema a ser gerado deve atender. Neste cada requisito funcional deve ter um nome, uma descrição e a indicação se é oculto ou evidente. Os requisitos evidentes são aqueles que necessitam de uma requisição do usuário para que ocorram ou, mesmo que sem a sua requisição ocorram com a ciência do mesmo; os ocultos, ao contrário, são aqueles imperceptíveis aos usuários.

Os requisitos não-funcionais definem características para o projeto e / ou para requisitos funcionais. Estas características normalmente são classificadas em alguma das seguintes categorias: usabilidade, confiabilidade, performance, configurabilidade, segurança, implementação, interface, empacotamento e legais.

Estas são as categorias inicialmente existentes no sistema HOPI, sendo possível modificar, cadastrar novas ou excluir as existentes. Além da categoria, os requisitos não-funcionais devem ter um nome, uma descrição, indicação se são desejáveis ou obrigatórios e se são permanentes ou transitórios.

Os casos de uso são descrições, em passos, do processo realizado para que cada um dos requisitos seja satisfeito. Existe uma relação direta, não necessariamente de um para um, entre requisitos funcionais e casos de uso. Cada caso de uso é formado por um nome, uma lista dos atores envolvidos, uma lista de atores interessados na sua realização, a lista de pré-condições necessárias para que o caso de uso ocorra, a lista das pós-condições que devem ser satisfeitas após a realização do caso de uso, descrições acerca das variações tecnológicas envolvidas com o caso de uso, uma lista com as dúvidas e questões em aberto e os passos que formam o caso de uso e suas respectivas variantes e fluxos alternativos. Tanto as variantes quanto os fluxos alternativos são compostos por passos, sendo chamados ações corretivas no fluxo alternativo que, conta ainda, com a finalização para indicar o que deve acontecer após o tratamento da exceção.

A lista dos Requisitos do sistema, segundo a notação UML, pode ser encontrada a partir da Tabela 1. dos Anexos, sob o item Requisitos do Sistema, na página 45.

1.2 Casos de Uso de Alto Nível

Um caso de uso de alto nível descreve, de uma maneira sucinta, o transcorrer do processo realizado a partir da interação entre o usuário e o sistema a ser desenvolvido a fim de satisfazer, uma ou mais, funcionalidades especificadas pelos requisitos funcionais.

É de conhecimento geral que sistemas de computador normalmente não possuem uma utilização linear, sendo possível realizar ou não, em qualquer ordem as atividades e funcionalidades existentes nestes programas. Portanto o caso de uso aqui proposto não é necessariamente o caminho mais comumente

utilizado. Ao invés disto, tenta apresentar todas as possíveis atividades realizadas pelos atores.

A seguir descrevemos o caso de uso para o nosso editor, que consiste na criação de uma modelagem de sistema, no que diz respeito às fases de levantamento de requisitos e definição de casos de uso.

Caso de Uso Modelar Sistema

O Administrador solicita a criação de um novo projeto. O Escritor preenche o nome, o sumário executivo e a data de início do projeto. O Escritor inclui os usuários correspondentes nas listas de leitores e escritores. Os atores que participarão dos casos de uso deste projeto são incluídos pelo Escritor, que informa seus nomes e descrições, para serem selecionados mais tarde. O Escritor inclui os requisitos funcionais do sistema, informando nome, se é oculto ou não, e a sua descrição. Para cada requisito funcional o Escritor inclui seus respectivos requisitos não-funcionais associados que são formados por um nome, uma categoria, a sua descrição e a sua classificação como obrigatório ou desejável e permanente ou transitório. O Escritor inclui os requisitos não-funcionais do projeto. O Escritor adiciona também os casos de uso para o projeto, informando o nome, a finalidade, a visão geral, selecionando os atores envolvidos, incluindo os interessados pelo caso de uso, as pré-condições para a realização, as pós-condições que devem ser satisfeitas após a realização bem sucedida, as variações tecnológicas envolvidas, duvidas acerca do processo que são cadastradas como questões em aberto e seleciona também quais requisitos funcionais este caso de uso atende. Além disso, o Escritor preenche também os passos que compõem o fluxo principal e as variantes e fluxos alternativos relacionados aos mesmos. Variantes, além de um nome, são também compostas por passos. Os fluxos alternativos são identificados pelas exceções que os iniciam, suas ações corretivas (passos que tratam a exceção) e a finalização, indicando o ponto de retorno da execução após o tratamento. O Escritor solicita que o novo projeto seja salvo no sistema encerrando este caso de uso.

O Caso de Uso de Alto nível segundo a notação proposta por [Larman2000] pode ser encontrado nos Anexos, como item do tópico Casos de Uso, na página 50.

Capítulo 2 - Análise

Antes da realização de qualquer interação entre o usuário e o sistema HOPI é necessário que o usuário se identifique.

Os atores que participam de cada caso de uso são usuários do sistema HOPI, com seu respectivo nível de acesso, ou o administrador. Antes de listar os casos de uso explicaremos com maiores detalhes quais são os atores envolvidos:

- Usuário – qualquer pessoa com permissão de acesso ao sistema HOPI, sem nenhum nível de acesso específico.
- Leitor – são usuários que estão cadastrados como leitores em um dado projeto. Este perfil esta, portanto, relacionado ao projeto. Não seria correto dizer que alguém tem acesso de leitor no sistema HOPI, mas sim que tem acesso de leitor em um determinado projeto do sistema HOPI. Estes usuários podem visualizar todos os elementos gerados para o projeto em questão.
- Escritor – assim como leitor, escritores são usuários cadastrados como escritores em um dado projeto, e da mesma forma o seu perfil não se aplica ao sistema HOPI em geral. Os escritores podem visualizar as mesmas informações que os leitores, podendo ainda incluir, modificar ou remover qualquer elemento de análise.
- Administrador – é um tipo de usuário do sistema HOPI. Seu perfil de acesso não é vinculado a nenhum projeto. O Administrador é o encarregado da criação, modificação ou remoção de usuários e categorias de requisitos não-funcionais. O Administrador pode visualizar, criar, modificar ou remover qualquer elemento de qualquer projeto, independente de estar cadastrado na lista de escritores ou não. A finalidade desta definição não é incentivar que o Administrador contribua na modelagem de todos os projetos do sistema HOPI, mas que possa modificar a listagem com os usuários que têm permissão nos projetos em momentos de emergência. Desta forma se algum problema for identificado e nenhum Escritor do projeto estiver disponível para realizá-las o Administrador pode se encarregar da correção ou nomear um novo Escritor para realizar a tarefa.

2.1 Entidades com Operações Cadastrais

Para qualquer sistema que estivermos planejando desenvolver teremos formulários para manutenção das informações armazenadas. As operações que podemos realizar sobre estas informações são: listar, incluir, modificar e remover. Este tipo de comportamento, portanto, não é novidade para analistas e desenvolvedores, por isso apenas listaremos as entidades que tenham tais características, ficando subentendido que para cada uma delas teremos as operações citadas anteriormente.

As características acerca dos pré-requisitos (condições que devem ser satisfeitas para a realização das operações) estão especificadas na seção **Contratos de Operações do Sistema**.

Para o nosso projeto as entidades com operações cadastrais são: Usuário, Projeto, Requisito Funcional, Requisito Não-Funcional, Categoria de Requisito Não-Funcional, Ator, Caso de Uso, Lista de Atores do Caso de Uso, Pré-Condições do Caso de Uso, Pós Condições do Caso de Uso, Requisitos Relacionados ao Caso de Uso, Interessados no Caso de Uso, Variação Tecnológica do Caso de Uso, Questão em Aberto do Caso de Uso, Fluxo Alternativo do Caso de Uso, Passo do Caso de Uso, Variante do Caso de Uso.

A tabela com as entidades com operações cadastrais segundo a notação proposta por [Larman2000] pode ser encontrada na Tabela 12. dos Anexos, sob o tópico Lista de Conceitos com Operações Cadastrais, na página 49.

2.2 Casos de Uso Expandidos

Nesta etapa é realizada a expansão dos casos de uso de alto nível, descritos na fase de planejamento. Precisamos extrair um maior nível de detalhes acerca das condições que precisam ser satisfeitas para o início do caso de uso bem como o que devemos obter ao terminar a sua execução com sucesso.

Na notação UML existe bastante diferença entre os casos de uso de alto nível e os casos de uso expandidos. Porém escrevendo-os de uma maneira textual a descrição encontrada nos casos de uso expandidos seria bastante

similar aquela já encontrada nos casos de uso de alto nível. A diferença seria a inclusão das pré-condições, que são as exigências para que o caso de uso possa ser iniciado, e das pós-condições, que definem o resultado que deve ser obtido ao término do caso de uso se o mesmo tiver sido bem sucedido. A descrição do caso de uso em si permanece exatamente a mesma encontrada no caso de uso de alto nível.

Caso de Uso Modelar Sistema

Para que este caso de uso ocorra é preciso existir pelo menos um usuário com o privilégio de administrador no sistema HOPI, e o mesmo precisa estar logado.

O Administrador solicita a criação de um novo projeto. O Escritor preenche o nome, o sumário executivo e a data de início do projeto. O Escritor inclui os usuários correspondentes nas listas de leitores e escritores. Os atores que participarão dos casos de uso deste projeto são incluídos pelo Escritor, que informa seus nomes e descrições, para serem selecionados mais tarde. O Escritor inclui os requisitos funcionais do sistema que está sendo criado, informando nome, se é oculto ou não, e a sua descrição. Para cada requisito funcional o Escritor inclui seus respectivos requisitos não-funcionais associados que são formados por um nome, uma categoria, a sua descrição e a sua classificação como obrigatório ou desejável e permanente ou transitório. O Escritor inclui requisitos não-funcionais do projeto. O Escritor adiciona também os casos de uso para o projeto, informando o nome, a finalidade, a visão geral, selecionando os atores envolvidos, incluindo os interessados pelo caso de uso, as pré-condições para a realização, as pós-condições que devem ser satisfeitas após a realização bem sucedida, as variações tecnológicas envolvidas, duvidas acerca do processo que serão cadastradas como questões em aberto e seleciona também quais requisitos funcionais este caso de uso atende. Além disso, o Escritor preenche também os passos que compõem o fluxo principal, suas variantes, que são formadas por um nome, e por um conjunto de passos, e os fluxos alternativos que possuem o nome da exceção que os inicia, um conjunto de passos que são as ações corretivas e a finalização, que indica o ponto de retorno após o tratamento

da exceção. O Escritor solicita que o novo projeto seja salvo no sistema HOPI encerrando este caso de uso.

Após a realização do caso de uso o novo projeto deve ter sido incluído no sistema HOPI. Os requisitos e casos de uso devem ter sido criados e associados ao projeto.

A expansão do caso de uso segundo a notação proposta por [Larman2000] pode ser encontrada nos Anexos, como item do tópico Casos de Uso, página 50.

2.3 Modelo Conceitual

O modelo conceitual especifica como ocorre o relacionamento entre as diferentes entidades do sistema HOPI. Antes de descrever tais relacionamentos será apresentada uma lista com o nome e a definição de cada uma destas entidades, bem como alguns atributos mais importantes:

Usuário: identifica as pessoas que utilizam o sistema HOPI. É composto pelo nome da pessoa, o nome do usuário, a senha, um marcador para indicar se é administrador ou não, um e-mail e o telefone.

Projeto: o projeto é a entidade que representa o processo de análise. Cada sistema a ser modelado é representado por uma entidade deste tipo.

Requisito Funcional: representam as características que um sistema a ser modelado deve atender para que satisfaça as necessidades levantadas. Requisitos funcionais são aqueles que definem “o que” deve ser feito pelo sistema a ser implementado.

Requisito Não-Funcional: define “como” será realizado “o que” deve ser feito pelo sistema. Podem definir questões gerais relativas ao projeto ou características relativas a requisitos funcionais. São classificados em diferentes categorias.

Categoria de Requisito Não-Funcional: caracteriza os requisitos não-funcionais com base na sua natureza. Os tipos comumente utilizados são: usabilidade, confiabilidade, performance, configurabilidade, segurança, implementação, interface, empacotamento, legais.

Ator: cada ator representa um tipo de pessoa que interage com o sistema a ser modelado na vida real. Cada ator desempenha um papel, assumindo diferentes responsabilidades.

Caso de Uso: são processos bem definidos, formados por passos, que definem um fluxo principal e devem ser realizados em uma ordem específica a fim de satisfazer as necessidades estipuladas pelos requisitos funcionais. Por esta razão casos de uso e requisitos funcionais estão intimamente ligados. O fluxo principal é o caminho naturalmente seguido em um caso de uso.

Lista de Atores do Caso de Uso: mantém a relação dos atores que interagem na realização de cada caso de uso.

Pré-Condições: é o conjunto de itens que representa as exigências que precisam ser satisfeitas para que seja possível a realização de um caso de uso.

Pós Condições: são as operações de sistema que precisam ser atendidas para considerarmos o caso de uso como bem sucedido. Entre estas podemos citar a criação, alteração ou exclusão de instâncias no sistema e a criação ou remoção de associações entre elas.

Requisitos Relacionados ao Caso de Uso: é a lista de quais requisitos o caso de uso em questão atende.

Interessados: é uma lista dos atores que, apesar de não participarem da realização do caso de uso, estão de alguma forma relacionados com o mesmo.

Variação Tecnológica: são notas e observações acerca das possíveis opções tecnológicas as serem utilizadas para a execução do caso de uso. Podem definir

ou descartar algum tipo de tecnologia, em virtude do seu custo, facilidade de uso, confiabilidade ou qualquer outro motivo.

Questão em Aberto do Caso de Uso: listagem de dúvidas levantadas durante o mapeamento dos passos para a realização do caso de uso. Normalmente são questões relativas a normas e procedimentos encontrados no próprio cliente, que muitas vezes não podem ser contatados imediatamente e por isso as dúvidas são anotadas para que possam ser tratadas e esclarecidas mais tarde. Por isso quando concluído o projeto não poderão permanecer nenhuma questão em aberto.

Fluxo Alternativo do Caso de Uso: consiste no fluxo que o caso de uso pode tomar em um determinado passo caso ocorra alguma exceção. Nele encontramos passos que indicam como esta exceção será tratada bem como qual será o ponto para onde a execução retornará após este tratamento.

Passo do Caso de Uso: cada caso de uso é formado por um ou mais passos. Um passo representa uma interação breve entre um ator e o sistema que está sendo modelado. É recomendado que cada passo represente uma troca de informação. Na grande maioria das vezes se um passo não representa uma troca de informação este poderia ser removido do caso de uso. Nestes casos o passo assume uma função mais ilustrativa, de contextualização.

Os passos do caso de uso descrevem os acontecimentos que naturalmente ocorrem durante o processo. O que podemos ter algumas vezes são passos variantes, onde em um determinado ponto do processo mais de um caminho pode ser seguido.

Um passo pode também gerar algum tipo de erro durante o processo, nesse caso temos os chamados passos de exceção que formam os fluxos alternativos, sempre relacionados a algum passo do fluxo principal.

Variante do Caso de Uso: quando um caso de uso pode seguir mais de um caminho diferente e não conseguimos definir um deles como sendo o fluxo principal incluímos todas as opções como variantes.

O elemento principal do nosso modelo é a entidade Projeto. Os Usuários podem relacionar-se com a entidade Projeto de três formas diferentes: se forem Administradores podem criar ou excluir Projetos, se forem Leitores podem apenas consultar as informações e se forem Escritores podem modificar o conteúdo do Projeto.

Os principais elementos que compõe um Projeto são Requisitos e Casos de Uso. Os Requisitos podem ser Funcionais ou Não-Funcionais. Cada Requisito Funcional define uma funcionalidade que o Projeto deve atender. Os Não-Funcionais especificam em mais detalhes as características dos Funcionais ou do projeto. Estas características estão divididas em diferentes Categorias.

Os Casos de Uso atendem as necessidades levantadas com a identificação dos Requisitos do sistema. O Caso de Uso é realizado por Atores (ou tem a participação dos mesmos). As Pré-Condições definem situações que precisam ser verdadeiras para a realização do Caso de Uso. As Pós-Condições são o resultado que deve ser obtido após a finalização bem sucedida do Caso de Uso.

O Fluxo Principal agrega os Passos que serão realizados para a conclusão do Caso de Uso. Passos podem levar a Variantes e também a Fluxos Alternativos. Tanto as Variantes quanto os Fluxos Alternativos são compostos por outros Passos.

2.4 Contratos de Operações do Sistema

Os contratos de operações de um sistema definem quais são as circunstâncias necessárias para a realização de determinadas ações, bem como qual é o resultado obtido após a realização das mesmas.

Para isso temos uma lista de pré-condições, que precisam ser verdadeiras para o início da operação, e outra lista com as pós-condições, que indicam o que foi modificado após a execução da operação.

Criar Usuário

- Pré-Condições: o usuário que criará o novo usuário precisa ser administrador e não pode existir nenhum outro usuário com o mesmo *username* informado.
- Pós-Condições: o novo usuário deve ter sido criado no sistema.

Editar Usuário

- Pré-Condições: o usuário que editará o novo usuário precisa ser administrador.
- Pós-Condições: o usuário deve ter recebido os novos valores para o nome, *password*, email, telefone e para a indicação se é administrador ou não.

Remover Usuário

- Pré-Condições: o usuário que deseja remover o outro usuário precisa ser administrador.
- Pós-Condições: o usuário deve ser removido de todas as listas de escritores e leitores dos projetos e depois do sistema.

Criar Projeto

- Pré-Condições: o usuário precisa ser administrador e não pode existir nenhum outro projeto com o mesmo nome informado.
- Pós-Condições: o novo projeto deve ter sido criado no sistema.

Editar Projeto

- Pré-Condições: o usuário precisa ser administrador e não pode existir nenhum outro projeto com o mesmo nome informado.
- Pós-Condições: o projeto deve ter recebido os novos valores para o nome, data inicial e para o sumário executivo.

Remover Projeto

- Pré-Condições: o usuário precisa ser administrador.
- Pós-Condições: os atores, requisitos, casos de uso e o projeto devem ter sido removidos.

Criar Requisito Funcional

- Pré-Condições: o usuário precisa constar na lista de escritores e não pode existir nenhum outro requisito funcional com o nome informado.
- Pós-Condições: o novo requisito funcional deve ter sido incluído.

Editar Requisito Funcional

- Pré-Condições: o usuário precisa constar na lista de escritores e não pode existir nenhum outro requisito funcional com o nome informado.
- Pós-Condição: o requisito funcional deve ter recebido os novos valores para o nome, descrição e para a indicação se é oculto ou não.

Remover Requisito Funcional

- Pré-Condições: o usuário precisa constar na lista de escritores.
- Pós-Condição: o requisito funcional deve ser removido de todas as listas que relacionam requisitos aos casos de uso e depois todos os seus requisitos não-funcionais associados, bem como o próprio requisito funcional, devem ser removidos.

Criar Requisito Não-Funcional

- Pré-Condições: o usuário precisa constar na lista de escritores e não pode existir nenhum requisito não funcional com o mesmo nome informado.
- Pós-Condições: o novo requisito não-funcional deve ter sido incluído e deve ter sido associado ao projeto ou ao requisito funcional.

Editar Requisito Não-Funcional

- Pré-Condições: o usuário precisa constar na lista de escritores e não pode existir nenhum requisito não funcional com o mesmo nome informado.
- Pós-Condição: o requisito não-funcional deve ter recebido os novos valores para o nome, descrição e para as indicações se é obrigatório ou não e se é permanente ou não.

Remover Requisito Não-Funcional

- Pré-Condições: o usuário precisa constar na lista de escritores.
- Pós-Condição: o requisito não-funcional deve ter sido removido.

Criar Categoria de Requisito Não-Funcional

- Pré-Condições: o usuário precisa ser administrador e não pode existir nenhuma outra categoria com o nome informado.
- Pós-Condições: a nova categoria deve ter sido criada no sistema.

Editar Categoria de Requisito Não-Funcional

- Pré-Condições: o usuário precisa ser administrador e não pode existir nenhuma outra categoria com o nome informado.
- Pós-Condições: a categoria deve ter recebido o novo valor para o nome e descrição.

Remover Categoria de Requisito Não-Funcional

- Pré-Condições: o usuário precisa ser administrador e não pode existir nenhum requisito não-funcional da categoria que desejamos destruir.
- Pós-Condições: a categoria deve ter sido removida do sistema.

Criar Ator

- Pré-Condições: o usuário precisa constar na lista de escritores do projeto e não pode existir nenhum outro ator com o nome informado.
- Pós-Condições: o novo ator deve ter sido incluído no sistema.

Editar Ator

- Pré-Condições: o usuário precisa constar na lista de escritores e não pode existir nenhum outro ator com o nome informado.
- Pós-Condições: o ator deve receber os novos valores para o nome e descrição.

Remover Ator

- Pré-Condições: o usuário precisa constar na lista de escritores do projeto.
- Pós-Condições: o ator deve ter sido removido da lista de atores de todos os casos de uso dos quais fazia parte e deve ter sido removido do sistema.

Criar Caso de Uso

- Pré-Condições: o usuário precisa constar na lista de escritores do projeto e não pode existir nenhum outro caso de uso com o nome informado.
- Pós-Condições: o novo caso de uso deve ter sido criado.

Editar Caso de Uso

- Pré-Condições: o usuário precisa constar na lista de escritores do projeto e não pode existir nenhum outro caso de uso com o nome informado.
- Pós-Condições: o caso de uso deve ter recebido os novos valores para o nome, finalidade e visão geral.

Remover Caso de Uso

- Pré-Condições: o usuário deve constar na lista de escritores do projeto.
- Pós-Condições: o caso de uso deve ter sido removido, juntamente com todos os seus elementos, pré e pós-condições, lista de atores e fluxo principal (com seus respectivos passos e fluxos alternativos e variantes).

Incluir Ator na Lista de Atores do Caso de Uso

- Pré-Condições: o usuário deve constar na lista de escritores do projeto.
- Pós-Condições: o ator deve ter sido incluído na lista de atores do caso de uso.

Remover Ator da Lista de Atores do Caso de Uso

- Pré-Condições: o usuário deve constar na lista de escritores do projeto.
- Pós-Condições: o ator deve ter sido removido da lista de atores do caso de uso.

Incluir Pré-Condição do Caso de Uso

- Pré-Condições: o usuário deve constar na lista de escritores do projeto.
- Pós-Condições: a pré-condição deve ter sido incluída no caso de uso.

Editar Pré-Condição do Caso de Uso

- Pré-Condições: o usuário deve constar na lista de escritores do projeto.
- Pós-Condições: a pré-condição deve ter recebido a nova descrição.

Remover Pré-Condição do Caso de Uso

- Pré-Condições: o usuário deve constar na lista de escritores do projeto.
- Pós-Condições: a pré-condição deve ter sido removida do caso de uso.

Incluir Pós-Condição do Caso de Uso

- Pré-Condições: o usuário deve constar na lista de escritores do projeto.
- Pós-Condições: a pós-condição deve ter sido incluída no caso de uso.

Editar Pós-condição do Caso de Uso

- Pré-Condições: o usuário deve constar na lista de escritores do projeto.
- Pós-Condições: a pós-condição deve ter recebido a nova descrição.

Remover Pós-condição do Caso de Uso

- Pré-Condições: o usuário deve constar na lista de escritores do projeto.
- Pós-Condições: a pós-condição deve ter sido removida do caso de uso.

Incluir Fluxo Alternativo no Passo do Caso de Uso

- Pré-Condições: o usuário deve constar na lista de escritores do projeto.
- Pós-Condições: o fluxo alternativo deve ter sido incluído no passo.

Remover Fluxo Alternativo

- Pré-Condições: o usuário deve constar na lista de escritores do projeto.
- Pós-Condições: o fluxo alternativo deve ter sido removido do passo.

Incluir Passo no Fluxo*

- Pré-Condições: o usuário deve constar na lista de escritores do projeto.
- Pós-Condições: o passo deve ter sido incluído no fluxo.

Editar Passo do Fluxo*

- Pré-Condições: o usuário deve constar na lista de escritores do projeto.
- Pós-Condições: o passo deve ter recebido o novo valor para a sua descrição.

Remover Passo do Fluxo*

- Pré-Condições: o usuário deve constar na lista de escritores do projeto.
- Pós-Condições: o passo deve ter sido removido do fluxo.

* Os contratos para a Inclusão, Edição e Remoção dos passos são exatamente iguais para o Fluxo Principal e Variantes, sendo apenas diferenciados pelo local onde são acionados e objetos que visam manipular.

Os mesmos contratos listados anteriormente podem ser encontrados em formato OCL - Object Constraint Language - nos Anexos, sob o item Contratos de Operações do Sistema, iniciando na página 54.

Capítulo 3 - Projeto

3.1 Diagramas de Colaboração

Os diagramas de colaboração são um dos artefatos mais importantes dentre os gerados no processo de análise e projetos de sistemas. Grande parte do tempo despendido nessas etapas deveria ser utilizado na criação destes diagramas. Isto porque, além da sua importância, são também um dos mais complexos.

Sua finalidade é demonstrar como as pós-condições definidas nos contratos serão satisfeitas. Para isso utilizaremos alguns artefatos de análise gerados anteriormente. Como já citado, os contratos indicam o que deve ser feito, o modelo conceitual ajuda a definir as classes de software que teremos que criar para garantir o cumprimento das tarefas e os casos de uso nos fornecem informações adicionais acerca do processo.

Por se tratar de uma ferramenta de edição, quer dizer, tem como função principal as tarefas de INCLUSÃO, EDIÇÃO, REMOÇÃO e LISTAGEM sobre os elementos conceituais, chegamos a conclusão de que um único diagrama genérico de cada tarefa seria o suficiente para descrever o processo, uma vez que seriam basicamente os mesmos se fossemos entrar em detalhes para cada conceito.

As operações sobre determinado elemento ocorrerão na sua respectiva tela de edição, por exemplo, não faz sentido editarmos um Ator em pleno cadastro de Usuário; portanto o elemento já estará a mão, bastando apenas atualizá-lo ao final da operação.

Os Diagramas de Colaboração podem ser encontrados a partir da Figura 2. dos Anexos, sob o tópico Diagramas de Colaboração, página 66.

3.1.1 Inclusão

Toda vez que quisermos incluir um novo elemento devemos estar na tela a qual este elemento pertence, por exemplo, não faz sentido a possibilidade de

inclusão de uma nova Pós Condição a um Caso de Uso estando em plena visualização dos Usuários do sistema. Quando quisermos incluir uma nova Pós Condição devemos estar visualizando todas as Pós Condições existentes para determinado Caso de Uso, restando apenas criar a nova estância, colocá-la na listagem corrente e então atualizar no devido Caso de Uso.

3.1.2 Edição

No caso da edição, as únicas diferenças com relação à inclusão é que ao invés de criarmos uma nova instância apenas vamos pegar a instância selecionada, mudar o atributo de interesse, substituí-la na listagem corrente e então atualizar a lista no elemento ao qual pertence.

3.1.3 Remoção

Na remoção o item selecionado é descartado através da sua posição na lista, e logo após lista é atualizada no item ao qual pertence.

3.1.4 Listagem

Não poderia ser mais simples: uma vez que precisamos estar num determinado Caso de uso para listar suas Pré Condições, basta apenas chamar a função getPreCondicoes do casoDeUso.

3.2 Diagrama de Classes

Com base no Modelo Conceitual e nos Diagramas de Colaboração podemos cria o Diagrama de Classes, que apresenta as classes de software que compõem o sistema que será a implementação dos artefatos gerados na modelagem. Com a criação deste diagrama teremos os atributos e seus respectivos tipos, métodos, relacionamentos, dependências e navegabilidade das classes de software.

O Diagrama gerado pode ser encontrado a partir da união da Figura 6. com a Figura 7. dos Anexos, sob os tópicos Diagrama de Classes – parte I, página 68, e Diagrama de Classes – parte II, página 69.

Capítulo 4 - Implementação

O objetivo com a análise e projeto de sistemas é gerar o conhecimento e a documentação necessários para possibilitar que a implementação ocorra de uma maneira correta e eficiente. Esta é, portanto, fundamental para a validação dos elementos gerados no processo de análise e projeto, elementos estes que foram descritos até aqui, neste documento.

É através do sistema obtido que podemos avaliar o resultado e decidir se o objetivo estabelecido foi, ou não, alcançado. A seguir serão descritas algumas das tecnologias, ferramentas e especificações utilizadas na implementação do nosso sistema:

4.1 Eclipse

Eclipse é uma ferramenta de programação que possui código aberto e é mantido pela Eclipse Foundation. O Eclipse conta com inúmeros plug-ins, desenvolvidos pelas mais variadas empresas, pela própria comunidade de código aberto a também por profissionais da área de desenvolvimento de software.

Ele surgiu a partir do código fonte de uma ferramenta da IBM chamada WebSphere Workbech que foi doada para a comunidade de software livre pela empresa.

Para o nosso projeto utilizamos a versão 3.0 do Eclipse, juntamente com o plug-in JBoss IDE, para a integração com o nosso servidor de aplicações e suporte ao uso de XDoclet para a geração automática de parte do código da aplicação.

4.2 Struts

O Struts é um framework que auxilia a parte de controle da aplicação. A sua configuração é feita a partir de um arquivo XML. As requisições do usuário

passam a ser direcionadas para o Struts que repassa aos componentes gerados pelo programador.

Para a utilização do Struts é necessária a criação de alguns elementos segundo a padronização estabelecida pelos seus idealizadores. Basicamente os elementos envolvidos na sua utilização são páginas JSP, formulários, ações, referências para as mensagens e os arquivos de configuração.

As páginas JSP são o limite entre o sistema e o mundo real, responsável pela comunicação entre o usuário e a aplicação. Depois que o usuário confirmar uma solicitação de entrada de dados no sistema as informações são passadas a um elemento do tipo Form, que espelha os campos existentes na página JSP. Se for necessário, e implementado pelo desenvolvedor, as informações dos campos são validades e se nenhum erro for encontrado a requisição da ação pretendida ao usuário é repassada para a ação. Caso alguma informação fornecida pelo usuário não esteja de acordo com o que era esperado o framework adiciona algumas mensagens de erro e devolve a execução para a página JSP que exibe as mensagens de erro adicionadas.

A ação por sua vez realiza alguma operação ou solicita algum serviço. No nosso caso a ação se comunica com o WebDelegate que por sua vez se comunica com o BusinessDelegate, ambos design patterns do tipo delegate utilizados para encapsular as chamadas e a comunicação entre as camadas, diminuindo o acoplamento entre elas.

As mensagens de erro passadas pelos Forms, bem como qualquer informação textual existente em páginas JSP, deve ser recuperada de um arquivo que mantém o relacionamento entre um nome e o conteúdo da mensagem. Desta forma o processo de internacionalização de software torna-se muito mais simples, sendo apenas necessária a tradução do arquivo que contém as mensagens e suas referências.

4.3 J2EE

A plataforma J2EE – Java 2 Platform, Enterprise Edition – oferece uma abordagem baseada em componentes para o design, desenvolvimento,

montagem e publicação de aplicações coorporativas. Oferece um modelo distribuído de aplicações multicamadas.

J2EE busca oferecer uma alternativa para que aplicações coorporativas possam ser modeladas e produzidas mais rapidamente, por um valor mais barato e com menos recursos.

Tendo em vista a necessidade de tornar o processo de desenvolvimento cada vez mais eficiente surgiram os design patterns. Estes são soluções consagradas para problemas que comumente são encontrados em diferentes aplicações. Não faria sentido que os projetistas tivessem que re-pensar a solução para os mesmos problemas todas as vezes que os encontrassem. A seguir listamos alguns design patterns utilizados na implementação do nosso projeto:

- Arquitetura MVC – Model-View-Controller: separa os sistemas nas camadas de apresentação, de controle e de negócio. A camada de apresentação (View) é responsável pelo conteúdo a ser exposto ao usuário, bem como em prover meios do usuário inserir as informações no sistema. A camada de controle (Controller) realiza a integração entre a de apresentação e a de negócio. A camada de negócio (Model) é que contém as regras aplicadas a cada sistema específico. Quanto ao re-aproveitamento de código podemos dizer que as camadas de apresentação e controle seriam as mais genéricas, e por isso mais re-aproveitáveis, enquanto que a de negócios seria a camada que mantém as especificidades de cada aplicação e por esta razão mais dificilmente re-aproveitável.

- Facade: este design pattern procura simplificar a utilização de um grande número de classes e módulos centralizando as chamadas para acesso aos mesmos. Além disso, um facade bem projetado pode resolver o problema gerado por uma coleção de classes mal projetadas.

- Business Delegate: tem a função de repassar as chamadas recebidas aos devidos objetos responsáveis em executar determinada tarefa.

- Service Locator: este design pattern realiza o armazenamento de referências a serviços já invocados devolvendo as instâncias em memória, evitando assim acessos repetidos ao servidor.

- Value Objects (VO): este design pattern, também conhecido como Data Transfer Object (DTO) ou Data Object (DO), é proposto para melhorar a comunicação entre cliente e servidor. A idéia por trás dos VOs baseia-se no fato de que repetidas requisições remotas de acesso ao banco e recuperação de informações são muito caras e dispendiosas. A proposta é que o servidor recupere todas as informações de uma única vez e crie um objeto para armazená-las, devolvendo o objeto ao usuário que depois recupera cada informação desejada, já em sua máquina local.

- Data Access Objects (DAO): utilizado para ocultar a existência do banco de dados através de objetos, cujos atributos são iguais aos campos de determinada tabela no banco. Internaliza funções de busca, armazenamento e remoção.

4.4 JBoss

JBoss é um servidor de aplicações implementado inteiramente em Java. Tem suporte ao padrão J2EE. É o servidor de aplicações de código aberto líder de mercado. Utiliza internamente o servidor web Tomcat.

Conclusão

Considerações Finais

A aplicação desenvolvida mantém as informações desejadas acerca dos requisitos e casos de uso dos sistemas a serem modelados. Cada item a ser informado sobre estes elementos é representado por diferentes campos, apresentados ao usuário pela interface. O sistema controla os campos que devem ser obrigatoriamente preenchidos, o que garante a consistência e padronização entre os projetos gerados.

O suporte a relacionamento entre requisitos funcionais e casos de uso também foi implementado, proporcionando um maior entendimento do sistema a ser gerado com o auxílio do HOPI.

A aplicação atendeu as nossas expectativas ao satisfazer os requisitos de armazenar as informações sobre as etapas de análise de requisitos e definição de casos de uso do processe de modelagem de sistemas.

Além disso, atende de maneira satisfatória o requisito de manter os relacionamentos entre estes elementos onde é possível, com apenas um clique de mouse, acessar o elemento relacionado.

Trabalhos Futuros

As tarefas que desejávamos realizar neste trabalho e que não conseguimos foram:

- Testes em ambientes com uma grande quantidade de usuários acessando o sistema simultaneamente.
- Testes com o banco de dados com um grande volume de informações.
- Criação das páginas de ajuda para todas as telas do sistema.

Além dos itens anteriores podemos citar outros que podem ser desenvolvidos:

- Acreditamos ser adequado criar um novo tipo de acesso chamado Gerente. Este ficaria responsável por manter os projetos no sistema. O Administrador, tipo de usuário que detém essa função atualmente seria um privilégio para permitir aos usuários modificarem questões relativas ao sistema e não mais aos projetos.
- Desenvolver relatórios que apresentem os elementos de análise gerados de maneira visualmente agradável e própria para a impressão.
- Implementar as demais fases de análise e projeto que consistem, entre outras, de: modelo conceitual, diagramas de seqüência, contratos de operações do sistema, diagramas de colaboração e diagrama de classes.
- Implementar a geração automática de código a partir de elementos gerados no sistema HOPI.

Referências Bibliográficas

[Larman2000] LARMAN, C. *Utilizando UML e Padrões: Uma introdução à análise e ao projeto orientados a objetos*. Porto Alegre: Bookman, 2000.

[GBM2003] GALLARDO, D.; BURNETTE, E. e MCGOVERN, R. *Eclipse in Action – A Guide for Java Developers*. Manning, 2003.

[HDFW2003] HUSTED, T., DUMOULIN, C., FRANCISCUS, G. e WINTERFELDT, D. *Struts in Action – Building web applications with the leading Java framework*. Manning, 2003.

[WR2004] WALIS, C. e RICHARDS, N. *XDoclet in Action*. Manning, 2004.

[D&D2002] DEITEL, H. e DEITEL, P. *Java Como Programar*. Bookman, 4ª edição, 2002.

[Kurniawan2002] KURNIAWAN, B. *Java for the Web with Servlets, JSP, and EJB: A Developer's Guide to J2EE Solutions.* New Riders Publishing, 2002.

[Ashmore2004] ASHMORE, D. *The J2EE Architect's Handbook – How to be a successful technical architect for J2EE applications*. DVT Press, 2004.

Glossário

Análise de Requisitos: este é um passo adotado na modelagem de sistemas que consiste em levantar as funcionalidades esperadas para o sistema a ser desenvolvido.

Breakpoints: termo utilizado em programação para denominar pontos de parada aplicados ao código do programa. Desta forma o desenvolvedor pode suspender a execução de um programa para continuá-la passo a passo verificando os valores das variáveis e procurando possíveis erros.

Business Delegate: design pattern (ver item design pattern) que consiste em centralizar as solicitações, repassando-as aos devidos objetos responsáveis em executar cada tarefa.

CASE: sigla para Computer-Aided Software Engineering.

Casos de Uso: processos definidos para satisfazerem as exigências dos requisitos do sistema que está sendo modelado. São interações entre usuários e o sistema.

Computer-Aided Software Engineering: programas de computador utilizados para auxiliar no processo de modelagem de sistemas, em português Engenharia de Software Auxiliada por Computador.

Contratos: definem quais são as circunstâncias necessárias para a realização de determinadas ações em um sistema de computador, bem como o resultado esperado após a realização das mesmas.

DAO: sigla para Data Access Obects.

Data Access Objects: design pattern (ver item design pattern) utilizado para ocultar a existência do banco de dados através de objetos, cujos atributos são

iguais aos campos de determinada tabela no banco. Internaliza funções de busca, armazenamento e remoção.

Data Objects: o mesmo que Value Objects.

Data Transfer Object: o mesmo que Value Objects.

Design Patterns: padrões de desenvolvimento de sistemas que se baseiam em soluções consagradas para problemas que comumente são encontrados na concepção dos mesmos.

DO: sigla para Data Objects.

DTO: sigla para Data Transfer Objects.

Eclipse: ferramenta de programação que possui código aberto e é mantido pela Eclipse Foundation (ver item Eclipse Foundation).

Eclipse Foundation: é uma corporação sem fins lucrativos formada para auxiliar a criação, evolução, promoção e suporte do Eclipse (ver item Eclipse) a para cultivar tanto uma comunidade de código aberto quanto um ecossistema de produtos, funcionalidades e serviços complementares.

Facade: design pattern (ver item design pattern) que procura simplificar a estrutura e utilização de um grande número de classes e módulos centralizando as chamadas para acesso aos mesmos.

Fluxos Alternativos: consiste no fluxo que o caso de uso pode tomar em um determinado passo caso ocorra alguma exceção. Nele encontramos passos que indicam como esta exceção será tratada bem como qual será o ponto para onde a execução retornará após este tratamento.

Framework: é uma coleção de componentes de software que auxiliam os programadores na concepção de novas aplicações.

Hardware: componentes mecânicos, magnéticos, eletrônicos e elétricos que formam um computador.

HTML: sigla para HyperText Markup Language.

HyperText Markup Language: linguagem utilizada para gerar documentos hipertexto para a apresentação na Internet.

J2EE: sigla para Java 2 Enterprise Edition.

Java: linguagem de programação orientada a redes de computadores desenvolvida pela Sun Microsystems para que os programas gerados a partir desta linguagem possam ser executados em qualquer máquina, independente de hardware e sistema operacional, sem riscos de vírus ou qualquer outro tipo de danos para os arquivos existentes na máquina.

Java 2 Enterprise Edition: abordagem baseada em componentes para o design, desenvolvimento, montagem e publicação de aplicações coorporativas. Oferece um modelo distribuído de aplicações multicamadas.

JBoss: servidor de aplicações implementado inteiramente em Java (ver item Java). Tem suporte ao padrão J2EE (ver item Java 2 Enterprise Edition). É o servidor de aplicações de código aberto líder de mercado. Utiliza internamente o servidor web Tomcat (ver item Tomcat).

JavaServerPages: página com o tradicional código HTML (ver item HTML) acrescido de código Java (ver item Java).

JSP: sigla para JavaServerPages.

Modelo Conceitual: artefato definido pela UML que especifica como ocorre o relacionamento entre diferentes entidades do sistema de computador.

MVC: sigla para Model-View-Controller.

Password: senha utilizada pelo usuário do sistema HOPI para se identificar.

Servlet: um programa em Java (ver item Java) que é executado com parte de um serviço normalmente um servidor web que responde as solicitações dos clientes.

Service Locator: design pattern (ver item design pattern) que realiza o armazenamento de referências a serviços já invocados devolvendo as instâncias em memória, evitando assim acessos repetidos ao servidor.

Software: programa de computador que prove instruções que possibilitam ao hardware desempenhar determinadas funções.

Struts: framework (ver item framework) para auxiliar no processo de desenvolvimento de aplicações web.

Template: modelos pré-estabelecidos para auxiliar na construção de novos artefatos.

Tomcat: servidor que é um container para aplicações web baseadas em Java que foi criado para rodar Servlets (ver item Servlets) e JavaServerPages(JSP) (ver item JavaServerPages) em aplicações web.

UML: sigla para Unified Modeling Language.

Unified Modeling Language: junção das três mais conceituadas linguagens de modelagem orientadas a objetos (Booch de Grady, OOSE de Jacobson e o OMT de Rumbaugh). Define padrões para a representação de artefatos que auxiliam no processo de definição e entendimento do funcionamento de um sistema de computador.

Username: nome utilizado para diferenciar e identificar os usuários do sistema HOPI.

Value Objects: design pattern (ver item design pattern neste glossário) proposto para melhorar a comunicação entre cliente e servidor. A idéia por trás dos VOs baseia-se no fato de que repetidas requisições remotas de acesso ao banco e recuperação de informações são muito caras e dispendiosas. A proposta é que o servidor recupere todas as informações de uma única vez e crie um objeto para armazená-las, devolvendo o objeto ao usuário que depois recupera cada informação desejada, já em sua máquina local.

Variações Tecnológicas: são notas e observações acerca das possíveis opções tecnológicas as serem utilizadas para a execução de um caso de uso. Podem definir ou descartar algum tipo de tecnologia, em virtude do seu custo, facilidade de uso, confiabilidade ou qualquer outro motivo.

Variantes: quando um caso de uso pode seguir mais de um caminho diferente e não conseguimos definir um deles como sendo o fluxo principal incluímos todas as opções como variantes.

VO: sigla para Value Objects.

Anexos

Requisitos do Sistema

Tabela 1. Requisito Manter Informações Cadastrais Sobre Usuários do Sistema.

F1 **Manter Informações Cadastrais sobre Usuários do Sistema**				Oculto ( )
Descrição: O usuário deve poder incluir, visualizar, alterar e remover Usuários do Sistema.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF1.1 Controle de Acesso	Incluir – só pode ser acessado por usuários com o perfil de Escrita ou de Administrador. Visualizar – qualquer usuário pode visualizar as suas próprias informações e o Administrador pode visualizar as informações de todos os usuários. Alterar – só pode ser acessado por usuários com o perfil de Administrador. Todos os usuários serão capazes de modificar a sua senha de acesso ao sistema. Remover - só pode ser acessado por usuários com o perfil de Administrador.	Segurança		
NF1.2 Informações Cadastrais	Atributos obrigatórios não modificáveis: nome do usuário. Atributos obrigatórios e modificáveis: senha, nome. Atributos não obrigatórios e modificáveis: um marcador para indicar se é administrador ou não, e-mail e telefone.	Especificação		

Tabela 2. Requisito Manter Informações Cadastrais Sobre Projeto.

F2 **Manter Informações Cadastrais sobre Projeto**				Oculto ( )
Descrição: O usuário deve poder criar, visualizar, alterar e remover um Projeto.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF2.1 Controle de Acesso	Criar – só pode ser acessado por usuários com o perfil de Administrador. Visualizar – pode ser acessado por quaisquer usuários cadastrados no projeto, independente de nível de acesso. Alterar – só pode ser acessado por usuários com o perfil de Escrita cadastrados no projeto ou de Administrador. Remover – só pode ser acessado por usuários com o perfil de Administrador.	Segurança		
NF2.2 Informações Cadastrais	Atributos obrigatórios não modificáveis: nome do projeto, data de criação (será atribuída automaticamente), nome do criador (também atribuído automaticamente), nome do cliente, prazo de entrega. Atributos não obrigatórios e modificáveis: sumário executivo.	Especificação		
NF2.3 Projeto deve ter Escritor	Cada projeto deve ter pelo menos um usuário cadastrado com perfil de Escritor.	Especificação		
NF2.4 Lista de Clientes	Deve ser possível selecionar o nome da Empresa solicitante a partir de uma lista.	Interface		

Tabela 3. Requisito Manutenção de usuários cadastrados em um Projeto.

F3 **Manutenção de Usuários Cadastrados em um Projeto**				Oculto ( )
Descrição: O Administrador deve ser capaz de incluir (permitindo acesso) e remover (bloqueando acesso) usuários de um respectivo Projeto.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF3.1 Controle de Acesso	A função só pode ser acessada por um usuário com o perfil de Administrador.	Segurança		

Tabela 4. Requisito Manter Informações Cadastrais sobre Requisito Funcional de Projeto.

F4 **Manter Informações Cadastrais sobre Requisito Funcional de Projeto**				Oculto ( )
Descrição: O usuário deve poder incluir, visualizar, alterar e remover Requisitos Funcionais em um Projeto.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF4.1 Controle de Acesso	Incluir – só pode ser acessado por usuários com o perfil de Escrita no projeto ou de Administrador. Visualizar – pode ser acessado por quaisquer usuários cadastrados no projeto, independente de nível de acesso. Alterar – só pode ser acessado por usuários com o perfil de Escrita no projeto ou de Administrador. Remover - só pode ser acessado por usuários com o perfil de Escrita no projeto ou de Administrador.	Segurança		
NF4.2 Informações Cadastrais	Atributos obrigatórios não modificáveis: referência do requisito de projeto (para futuras referências e gerado automaticamente pelo sistema). Atributos obrigatórios e modificáveis: nome do requisito de projeto, descrição do requisito e a indicação se é oculto ou evidente.	Especificação		
NF4.3 Lista de Requisitos Não-Funcionais	Deve apresentar uma lista dos requisitos não-funcionais já cadastrados.	Interface		

Tabela 5. Requisito Manter Informações Cadastrais sobre Categoria de Requisito Não-Funcional.

F5 **Manter Informações Cadastrais sobre Categoria de Requisito Não-Funcional**				Oculto ( )
Descrição: O usuário deve poder incluir, visualizar, alterar e remover Categorias dos Requisitos Não-Funcionais.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF5.1 Controle de Acesso	Incluir – só pode ser acessado por usuários com o perfil de Administrador. Visualizar - só pode ser acessado por usuários com o perfil de Administrador. Alterar – só pode ser acessado por usuários com o perfil de Administrador. Remover - só pode ser acessado por usuários com o perfil de Administrador.	Segurança		
NF5.2 Informações Cadastrais	Atributos obrigatórios não modificáveis: nome da categoria do requisito não-funcional. Atributos não obrigatórios e modificáveis: descrição.	Especificação		

Tabela 6. Requisito Manter Informações Cadastrais sobre Requisito Não-Funcional.

F6 **Manter Informações Cadastrais sobre Requisito Não-Funcional**				Oculto ( )
Descrição: O usuário deve poder incluir, visualizar, alterar e remover Requisitos Não-Funcionais em um Requisito ou Projeto.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF6.1 Controle de Acesso	Incluir – só pode ser acessado por usuários com o perfil de Escrita no projeto ou de Administrador Visualizar – pode ser acessado por quaisquer usuários cadastrados no projeto, independente de nível de acesso. Alterar – só pode ser acessado por usuários com o perfil de Escrita no projeto ou de Administrador. Remover - só pode ser acessado por usuários com o perfil de Escrita no projeto ou de Administrador.	Segurança		
NF6.2 Informações Cadastrais	Atributos obrigatórios não modificáveis: referência do requisito não-funcional (para futuras referências e gerado automaticamente pelo sistema). Atributos obrigatórios e modificáveis: nome do requisito não-funcional, descrição do requisito, categoria do requisito não-funcional, indicação se é desejável ou obrigatório e uma indicação se é transitório ou permanente.	Especificação		
NF6.3 Lista de Categoria	Deve ser possível selecionar a categoria do requisito não-funcional a partir de uma lista.	Interface		

Tabela 7. Requisito Manter Informações Cadastrais sobre Atores

F7 **Manter Informações Cadastrais sobre Atores**				Oculto ( )
Descrição: O usuário deve poder incluir, visualizar, alterar e remover Atores em um Projeto.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF7.1 Controle de Acesso	Incluir – só pode ser acessado por usuários com o perfil de Escrita ou de Administrador. Visualizar – pode ser acessado por quaisquer usuários cadastrados no projeto, independente de nível de acesso. Alterar – só pode ser acessado por usuários com o perfil de Escrita ou de Administrador. Remover - só pode ser acessado por usuários com o perfil de Escrita ou de Administrador.	Segurança		
NF7.2 Informações Cadastrais	Atributos obrigatórios modificáveis: nome. Atributos não obrigatórios e modificáveis: descrição.	Especificação		

Tabela 8. Requisito Manter Informações Cadastrais sobre Caso de Uso.

F8 **Manter Informações Cadastrais sobre Caso de Uso**				Oculto ( )
Descrição: O usuário deve poder incluir, visualizar, alterar e remover Casos de Uso em um Projeto.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF8.1 Controle de Acesso	Incluir – só pode ser acessado por usuários com o perfil de Escrita no projeto ou de Administrador. Visualizar – pode ser acessado por quaisquer usuários cadastrados no projeto, independente de nível de acesso. Alterar – só pode ser acessado por usuários com o perfil de Escrita no projeto ou de Administrador. Remover - só pode ser acessado por usuários com o perfil de Escrita no projeto ou de Administrador.	Segurança		
NF8.2 Informações Cadastrais	Atributos obrigatórios não modificáveis – referência (para futuras referências e gerado automaticamente pelo sistema). Atributos obrigatórios e modificáveis – um nome, os atores envolvidos e as referências cruzadas (requisitos correlacionados). Atributos não obrigatórios e modificáveis – as pré-condições para que o mesmo ocorra, as pós-condições que esperamos serem satisfeitas após a execução bem sucedida do caso de uso, interessados no resultado da operação, finalidade, visão geral, variações tecnológicas, questões em aberto e os passos que compõem o fluxo principal.	Especificação		
NF8.3 Lista de Atores	Deve ser possível selecionar os atores desejados de uma lista.	Interface		
NF8.4 Lista de Pré e Pós-Condições	Deve ser possível visualizar as pré e pós-condições já cadastradas para o caso de uso.	Interface		
NF8.5 Fluxo Principal e Elementos Relacionados	Deve ser possível visualizar o fluxo principal do caso de uso, bem como suas variantes e o fluxo de exceção com suas ações corretivas.	Interface		

Tabela 9. Requisito Manter Informações Cadastrais sobre Variantes.

F9 **Manter Informações Cadastrais sobre Variantes**				Oculto ( )
Descrição: O usuário deve poder incluir, visualizar, alterar e remover variantes de um passo do fluxo principal do caso de uso de um projeto. Variantes são utilizadas quando não é possível definir um caminho como sendo o mais comumente utilizado. Neste caso teremos uma variante, que é formada por um nome e uma seqüência de passos, para cada possível caminho a ser seguido.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF9.1 Controle de Acesso	Incluir – só pode ser acessado por usuários com o perfil de Escrita no projeto ou de Administrador. Visualizar – pode ser acessado por quaisquer usuários cadastrados no projeto, independente de nível de acesso. Alterar – só pode ser acessado por usuários com o perfil de Escrita no projeto ou de Administrador. Remover - só pode ser acessado por usuários com o perfil de Escrita no projeto ou de Administrador.	Segurança		
NF9.2 Informações Cadastrais	Atributos obrigatórios e não modificáveis – referência da variante dentro do passo do caso de uso: referencia o passo relativo a variante incluída (gerada automaticamente pelo sistema). Atributos obrigatórios e modificáveis – nome da variante. Atributos não obrigatórios e modificáveis – descrição da variante e seus passos.	Especificação		

Tabela 10. Requisito Manter Informações Cadastrais sobre Fluxo Alternativo.

F10 **Manter Informações Cadastrais sobre Fluxo Alternativo**				Oculto ( )
Descrição: O usuário deve poder incluir, visualizar, alterar e remover fluxos alternativos para passos do fluxo principal do caso de uso de um projeto. Fluxos alternativos conjuntos de ações corretivas e uma finalização que definem o tratamento para exceções que venham a ocorrer em determinados passos do caso de uso. Deverá ser cadastrado um fluxo alternativo para cada possível exceção.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF10.1 Controle de Acesso	Incluir – só pode ser acessado por usuários com o perfil de Escrita no projeto ou de Administrador. Visualizar – pode ser acessado por quaisquer usuários cadastrados no projeto, independente de nível de acesso. Alterar – só pode ser acessado por usuários com o perfil de Escrita no projeto ou de Administrador. Remover - só pode ser acessado por usuários com o perfil de Escrita no projeto ou de Administrador.	Segurança		
NF10.2 Informações Cadastrais	Atributos obrigatórios e não modificáveis – referência do fluxo alternativo dentro do passo do caso de uso: referencia o passo que pode originar o fluxo alternativo (gerada automaticamente pelo sistema). Atributos obrigatórios e modificáveis – nome do fluxo alternativo. Atributos não obrigatórios e modificáveis – ações corretivas do fluxo alternativo e a finalização.	Especificação		

Requisitos Não-Funcionais Gerais

Tabela 11. Requisitos Não Funcionais Gerais.

Nome	Restrição	Categoria	Desejável	Permanente
NFG1 Intuitividade	O sistema deve ser intuitivo e guiar o usuário pelo processo de análise. Deve ser claro para ele o que deverá fazer a seguir, e o que precisa estar pronto para avançar para a próxima etapa.	Usabilidade		
NFG2 Navegabilidade	O sistema deve proporcionar uma fácil navegação. O usuário deve poder rapidamente consultar os casos de uso relacionados a um requisito funcional e vice-versa. O usuário deve poder encontrar o que deseja em poucos cliques do mouse e sem ter que percorrer muitas telas.	Usabilidade		

Lista de Conceitos com Operações Cadastrais

Tabela 12. Lista de Conceitos com Operações Cadastrais.

CONCEITO	DESCRIÇÃO	REFERÊNCIAS CRUZADAS
Usuário	Identifica as pessoas que utilizam o sistema. É composto pelo nome da pessoa, o nome do usuário, a senha, um marcador para indicar se é administrador ou não, um e-mail e o telefone.	F1
Projeto	O projeto é a entidade que representa o processo de análise. Cada sistema a ser modelado é representado por uma entidade deste tipo.	F2
Requisito Funcional	Representam as características que um sistema deve atender para que satisfaça as necessidades do projeto de um sistema computacional. Requisitos funcionais são aqueles que definem “o que” deve ser feito pelo sistema a ser implementado.	F4
Requisito Não-Funcional	Tanto o projeto como requisitos funcionais podem possuir requisitos não-funcionais associados. Requisitos não funcionais definem “como” será realizado “o que” deve ser feito pelo sistema. São classificados em diferentes categorias.	F6
Categoria de Requisito Não-Funcional	Caracteriza os requisitos não-funcionais com base na sua natureza. Os tipos comumente utilizados são: usabilidade, confiabilidade, performance, configurabilidade, segurança, implementação, interface, empacotamento, legais.	F5
Atores	Cada ator representa um tipo de usuário que interage com o sistema na vida real. Cada um será responsável por desempenhar diferentes papéis, com diferentes níveis de responsabilidade e privilégio de acesso.	F7
Caso de Uso	São processos bem definidos, formados por passos, que definem um fluxo principal, e devem ser realizados em uma ordem específica a fim de satisfazer as necessidades estipuladas pelos requisitos funcionais. Por esta razão casos de uso e requisitos funcionais estão intimamente ligados. O fluxo principal é o caminho naturalmente seguido em um caso de uso.	F8
Lista de Atores do Caso de Uso	Mantém a relação dos atores que interagem na realização de cada caso de uso.	F8
Pré-condições do Caso de Uso	É o conjunto de itens que representa as exigências que precisam ser satisfeitas para que seja possível a realização de um caso de uso.	F8
Pós-condições do Caso de Uso	São as operações de sistema que precisam ser atendidas para considerarmos o caso de uso como bem sucedido. Entre estas podemos citar a criação, alteração ou exclusão de instâncias no sistema e a criação ou remoção de associações entre elas.	F8
Requisitos Relacionados ao Caso de Uso	É a lista de quais requisitos funcionais o caso de uso em questão atende.	F8
Interessados no Caso de Uso	É uma lista dos atores que, apesar de não participar da realização do caso de uso, estão de alguma forma relacionados com o caso de uso.	F8
Variações Tecnológicas do Caso de Uso	São notas e observações que acerca das possíveis opções tecnológicas as serem utilizadas para a execução do caso de uso. Podem definir ou descartar algum tipo de tecnologia, em virtude do seu custo, facilidade de uso, confiabilidade ou qualquer que seja o motivo.	F8
Questões em Aberto do Caso de Uso	Listagem de dúvidas levantadas durante o mapeamento dos passos para a realização do caso de uso. Normalmente são questões relativas a normas e procedimentos encontrados no próprio cliente, que muitas vezes não podem ser contatados imediatamente e por isso as dúvidas são anotadas para que possam ser tratadas e esclarecidas mais tarde. Por isso quando concluído o projeto não poderão permanecer nenhuma questão em aberto.	F8
Passo do Caso de Uso	Cada caso de uso é formado por um ou mais passos. Um passo representa uma interação breve entre um ator e o sistema. É recomendado que cada passo represente uma troca de informação. Na grande maioria das vezes se um passo não representa uma troca de informação este poderia ser removido do caso de uso. Nestes casos o passo assume uma função mais ilustrativa, de contextualização. Os passos do caso de uso descrevem os acontecimentos que naturalmente ocorrem durante o processo. O que podemos ter algumas vezes são passos variantes, onde em um determinado ponto do processo mais de um caminho pode ser seguido. Um passo pode também gerar algum tipo de erro durante o processo, nesse caso temos os chamados passos de exceção que formam os fluxos alternativos, sempre relacionados a algum passo do fluxo principal.	F8
Variante do Caso de Uso	Quanto um caso de uso pode seguir mais de um caminho diferente e não conseguimos definir um deles como sendo o fluxo principal incluímos todas as opções como variantes.	F9
Passo da Variante	Descrevem os acontecimentos envolvidos na realização da variante.	F9
Fluxo Alternativo do Caso de Uso	Consiste no fluxo que o caso de uso pode tomar em um determinado passo caso ocorra alguma exceção. Nele encontramos ações corretivas que indicam como esta exceção será tratada bem como qual será o ponto para onde a execução retornará após este tratamento.	F10
Ação Corretiva do Fluxo Alternativo	Representa cada passo realizado no fluxo alternativo para tratar a exceção gerada.	F10
Finalização do Fluxo Alternativo	Indica o ponto de retorno da execução do sistema após a correção da exceção tratada pelo fluxo alternativo.	F10

Casos de Uso

Casos de Uso de Alto Nível

Caso de Uso:	Modelar Sistema
Atores:	Administrador, Escritor
Tipo:	Primário
Descrição:	O Administrador solicita a criação de um novo projeto. O Escritor preenche o nome, o sumário executivo e a data de início do projeto. O Escritor inclui os usuários correspondentes nas listas de leitores e escritores. Os atores que participarão dos casos de uso deste projeto são incluídos pelo Escritor, que informa seus nomes e descrições, para serem selecionados mais tarde. O Escritor inclui os requisitos funcionais do sistema, informando nome, se é oculto ou não, e a sua descrição. Para cada requisito funcional o Escritor inclui seus respectivos requisitos não-funcionais associados que são formados por um nome, uma categoria, a sua descrição e a sua classificação como obrigatório ou desejável e permanente ou transitório. O Escritor inclui requisitos não-funcionais no projeto. O Escritor adiciona também os casos de uso para o projeto, informando o nome, a finalidade, a visão geral, selecionando os atores envolvidos, incluindo os interessados pelo caso de uso, as pré-condições para a realização, as pós-condições que devem ser satisfeitas após a realização bem sucedida, as variações tecnológicas envolvidas, duvidas acerca do processo que serão cadastradas como questões em aberto e seleciona também quais requisitos funcionais este caso de uso atende. Além disso, o Escritor preenche também os passos que compõem o fluxo principal, suas variantes, que são formadas por um nome, e por um conjunto de passos, e os fluxos alternativos que possuem o nome da exceção que os inicia, um conjunto de passos que são as ações corretivas e a finalização, que indica o ponto de retorno após o tratamento da exceção. O Escritor solicita que o novo projeto seja salvo no sistema encerrando este caso de uso.

Casos de Uso Expandidos

Modelar Sistema

Atores:

Administrador, Escritor

Interessados:

Leitor

Pré-condições:

- Deve existir pelo menos um usuário com o privilégio de administrador.
- O Administrador deve estar logado no sistema.

Pós-condições:

- O novo projeto deve ter sido criado.
- Os requisitos funcionais devem ter sido criados e associados ao projeto.
- Os requisitos não-funcionais devem ter sido criados e associados ao projeto e /ou aos requisitos funcionais.
- Os casos de uso devem ter sido criados e associados ao projeto.

Referências Cruzadas:

Variações Tecnológicas:

Questões em Aberto:

Fluxo Principal:

1. O Administrador solicita a inclusão do novo projeto.
2. O Escritor preenche as informações referentes ao projeto: nome, sumário executivo e data de início.
3. O Escritor inclui os usuários que devem ter acesso ao projeto nas listas de leitores e escritores.
4. O Escritor inclui os atores que serão utilizados nos casos de uso, informando seus nomes e descrições.
5. O Escritor inclui os requisitos funcionais do sistema, informando o nome, se é oculto ou não, e a sua descrição.

6. O Escritor inclui os requisitos não-funcionais associados a cada requisito funcional que devem ter um nome, uma categoria, uma descrição e suas classificações como obrigatório ou desejável e permanente ou transitório.
7. O Escritor inclui requisitos não-funcionais associados ao projeto.
8. O Escritor adiciona também os casos de uso para o projeto, informando o nome, a finalidade, a visão geral, selecionando os atores envolvidos, incluindo os interessados pelo caso de uso, as pré-condições para a realização, as pós-condições que devem ser satisfeitas após a realização bem sucedida, as variações tecnológicas envolvidas, duvidas acerca do processo que serão cadastradas como questões em aberto e seleciona também quais requisitos funcionais este caso de uso atende.
9. O Escritor inclui os passos que compõem o fluxo principal.
10. O Escritor inclui as variantes para os passos que as exigem, informando o nome e o conjunto de passos que as compõe.
11. O Escritor inclui os fluxos alternativos para os passos que os exigem, informando o nome da exceção que os inicia, um conjunto de passos que são as ações corretivas e a finalização, que indica o ponto de retorno após o tratamento da exceção.
12. O Escritor solicita que o novo projeto seja salvo no sistema encerrando este caso de uso.

Variantes:

Fluxos Alternativos:

Modelo Conceitual

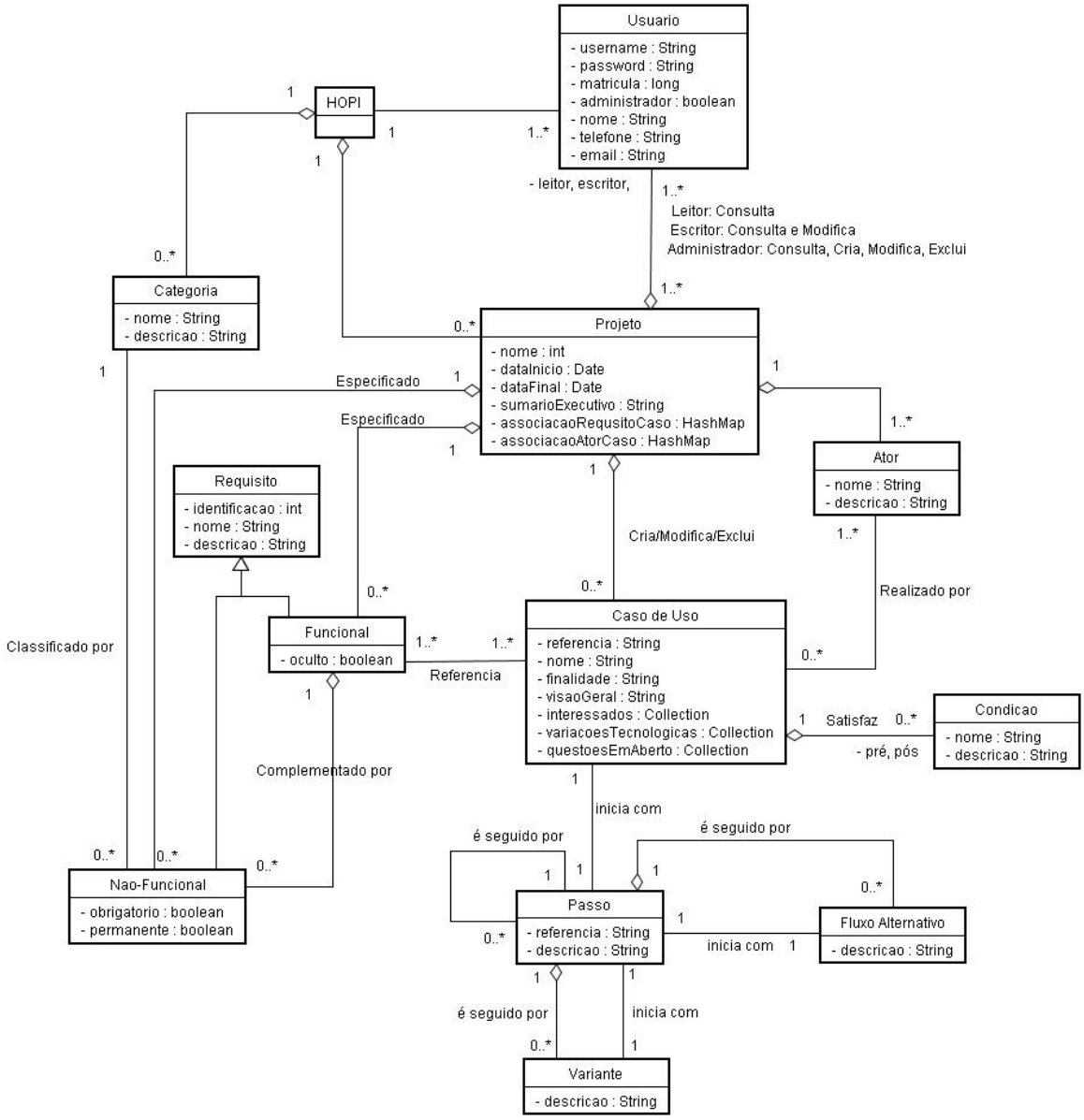

Figura 1. Modelo Conceitual

Contratos de Operações do Sistema

objetoCorrente : elemento aberto.

objetoSelecionado : elemento selecionado em uma lista.

Usuário

Criar

Sistema::criarUsuario(nome, username, password, email, telefone, administrador)

- pre:
  - usuarioCorrente.administrador = true
  - usuarios → not exists(u | u.username = username)
- post:
  - usuarios → including(Usuario.new())

Editar

Sistema::editarUsuario(nome, password, email, telefone,administrador)

- pre:
  - usuarioCorrente.administrador = true
- post:
  - usuarioCorrente.nome = nome
  - usuarioCorrente.password = password
  - usuarioCorrente.email = email
  - usuarioCorrente.telefone = telefone
  - usuarioCorrente.administrador = administrador

Destruir

Sistema::destruirUsuario(usuario)

- pre:
  - usuarioCorrente.administrador = true
- post:
  - projetos → forAll(projeto.leitores → removing(usuario))
  - projetos → forAll(projeto.escritores → removing(usuario))
  - usuario.destroy()

Projeto

Criar

Sistema::criarProjeto(nome, dataInicial, sumarioExecutivo)

- pre:
  - usuarioCorrente.administrador = true
  - projetos → not exists(p | p.nome = nome)
- post:
  - projetos → including(Projeto.new())

Editar

Sistema::editarProjeto(nome, dataInicial, sumarioExecutivo)

- pre:
  - usuarioCorrente.administrador = true
  - projetos → not exists(p | p.nome = nome)
- post:
  - projetoCorrente.nome = nome
  - projetoCorrente.dataInicial = dataInicial
  - projetoCorrente.sumarioExecutivo = sumarioExecutivo

Destruir

Sistema::destruirProjeto(projeto)

- pre:
  - usuarioCorrente.administrador = true
- post:
  - atores → forAll(ator.destroy())
  - requisitosFuncionais → forAll(requisitoFuncional.destroy())
  - requisitosNaoFuncionais → forAll(requisitoNaoFuncional.destroy())
  - casosDeUso → forAll(casoDeUso.destroy())
  - projeto.destroy()

Requisito Funcional

Criar

Projeto::criarRequisitoFuncional(nome, descricao, oculto)

- pre:
  - escritores → exists(e | e = sistema.usuarioCorrente)
  - requisitosFuncionais → not exists(r | r.nome = nome)
- post:
  - requisitosFuncionais → including(RequisitoFuncional.new())

Editar

Projeto::editarRequisitoFuncional(nome, descricao, oculto)

- pre:
  - escritores → exists(e | e = sistema.usuarioCorrente)
  - requisitosFuncionais → not exists(r | r.nome = nome)
- post:
  - requisitoFuncionalCorrente.nome = nome
  - requisitoFuncionalCorrente. descricao = descricao
  - requisitoFuncionalCorrente. oculto = oculto

Destruir

Projeto::destruirRequisitoFuncional(requisitoFuncional)

- pre:
  - escritores → exists(e | e = sistema.usuarioCorrente)
- post:
  - casosDeUso → forAll(casoDeUso.requisitosFuncionais → removing(requisitoFuncional))
  - requisitoFuncional.requisitosNaoFuncionais → forAll(requisitoNaoFuncional.destroy())
  - requisitoFuncional.destroy()

Requisito Não-Funcional

Criar

RequisitoFuncional | Projeto::criarRequisitoNaoFuncional (nome, descricao, obrigatorio, permanente)

- pre:
  - escritores → exists(e | e = sistema.usuarioCorrente)
  - requisitosNaoFuncionais → not exists(r | r.nome = nome)
- post:
  - requisitosNaoFuncionais → including(RequisitoNaoFuncional.new())

Editar

RequisitoFuncional | Projeto::editarRequisitoNaoFuncional(nome, descricao, obrigatorio,permanente)

- pre:
  - escritores → exists(e | e = sistema.usuarioCorrente)
  - requisitosNaoFuncionais → not exists(r | r.nome = nome)
- post:
  - requisitoNaoFuncionalCorrente.nome = nome
  - requisitoNaoFuncionalCorrente.descricao = descricao
  - requisitoNaoFuncionalCorrente.obrigatorio = obrigatorio
  - requisitoNaoFuncionalCorrente.permanente = permanente

Destruir

RequisitoFuncional |
Projeto::destruirRequisitoNaoFuncional(requisitoNaoFuncional)

- pre:
  - escritores → exists(e | e = sistema.usuarioCorrente)
- post:
  - requisitoNaoFuncional.destroy()

Categoria de Requisito Não-Funcional

Criar

Sistema::criarCategoriaDeRequisitoNaoFuncional(nome, descricao)

- pre:
  - self.usuarioCorrente.administrador = true
  - self.categorias → not exists(c | c.nome = nome)
- post:
  - self.categorias → including(Categoria.new())

Editar

Sistema::editarCategoriaDeRequisitoNaoFuncional (nome, descricao)

- pre:
  - self.usuarioCorrente.administrador = true
  - self.categorias → not exists(c | c.nome = nome)
- post:
  - self.categoriaCorrente.nome = nome
  - self.categoriaCorrente.descricao = descricao

Destruir

Sistema::destruirCategoriaDeRequisitoNaoFuncional ()

- pre:
  - self.usuarioCorrente.administrador = true
  - self.requisitosNaoFuncionais →
    not exists(r | r = self.categoriaSelecionada)
- post:
  - self.categoriaSelecionada.destroy()

Ator

Criar

Projeto::criarAtor(nome, descricao)

- pre:
  - sistema.projetoCorrente.escritores →
    exists(e | e = sistema.usuarioCorrente)
  - sistema.projetoCorrente.atores → not exists(a | a.nome = nome)
- post:
  - sistema.projetoCorrente.atores → including(Ator.new())

Editar

Projeto::editarAtor(nome, descricao)

- pre:
  - sistema.projetoCorrente.escritores →
    exists(e | e = sistema.usuarioCorrente)
  - sistema.projetoCorrente.atores → not exists(a | a.nome = nome)
- post:
  - sistema.atorCorrente.nome = nome
  - sistema.atorCorrente.descricao = descricao

Destruir

Projeto::destruirAtor()

- pre:
  - sistema.projetoCorrente.escritores →
    exists(e | e = sistema.usuarioCorrente)
- post:
  - sistema.projetoCorrente.casosDeUso →
    forAll(casoDeUso.atores → removing(sistema.atorSelecionado)
  - sistema.atorSelecionado.destroy()

Caso de Uso

Criar

Projeto::criarCasoDeUso(nome, finalidade, visaoGeral)

- pre:
  - sistema.projetoCorrente.escritores →
    exists(e | e = sistema.usuarioCorrente)
  - sistema.projetoCorrente.casosDeUso → not exists(c | c.nome = nome)
- post:
  - sistema.projetoSelecionaod.casosDeUso → including(CasoDeUso.new())

Editar

Projeto::editarCasoDeUso(nome, finalidade, visaoGeral)

- pre:
  - sistema.projetoCorrente.escritores →
    exists(e | e = sistema.usuarioCorrente)
  - sistema.projetoCorrente.casosDeUso → not exists(c | c.nome = nome)
- post:
  - sistema.casoDeUsoCorrente.nome = nome
  - sistema.casoDeUsoCorrente.finalidade = finalidade
  - sistema.casoDeUsoCorrente.visaoGeral = visaoGeral

Destruir

Projeto::destruirCasoDeUso()

- pre:
  - sistema.projetoCorrente.escritores →
    exists(e | e = sistema.usuarioCorrente)
- post:
  - sistema.casoDeUsoSelecionado.fluxoPrincipal.destroy()
  - sistema.casoDeUsoSelecionado.preCondicoes →
    forAll(preCondicao.destroy())
  - sistema.casoDeUsoSelecionado.posCondicoes →
    forAll(posCondicao.destroy())

- sistema.casoDeUsoSelecionado.atores → destroy()
- sistema.casoDeUsoSelecionado.destroy()

Lista de Atores do Caso de Uso

Incluir Ator no Caso de Uso

CasoDeUso::incluirAtorNaLista(ator)

- pre:
  - sistema.projetoCorrente.escritores →
    exists(e | e = sistema.usuarioCorrente)
- post:
  - sistema.casoDeUsoCorrente.atores → including(ator)

Remover Ator do Caso de Uso

CasoDeUso::removerAtorDaLista(ator)

- pre:
  - sistema.projetoCorrente.escritores →
    exists(e | e = sistema.usuarioCorrente)
- post:
  - sistema.casoDeUsoSelecionado.atores → removing(ator)

Pré-condição do Caso de Uso

Incluir Pré-condição do Caso de Uso

CasoDeUso::incluirPreCondicao(descricao)

- pre:
  - sistema.projetoCorrente.escritores →
    exists(e | e = sistema.usuarioCorrente)
- post:
  - sistema.casoDeUsoCorrente.preCondicoes →
    including(PreCondicao.new())

Editar Pré-Condição do Caso de Uso

CasoDeUso::editarPreCondicao(descricao)

- pre:
  - sistema.projetoCorrente.escritores →
    exists(e | e = sistema.usuarioCorrente)
- post:
  - sistema.preCondicaoCorrente.descricao = descricao

Remover Pré-Condição do Caso de Uso

CasoDeUso::removerPreCondicao(preCondicao)

- pre:
  - sistema.projetoCorrente.escritores →
    exists(e | e = sistema.usuarioCorrente)
- post:
  - sistema.casoDeUsoCorrente.preCondicoes → removing(preCondicao)

Pós-condição do Caso de Uso

Incluir Pós-condição do Caso de Uso

CasoDeUso::incluirPosCondicao(descricao)

- pre:
  - sistema.projetoCorrente.escritores →
    exists(e | e = sistema.usuarioCorrente)
- post:
  - sistema.casoDeUsoCorrente.posCondicoes →
    including(PosCondicao.new())

Editar Pós-condição do Caso de Uso

CasoDeUso::editarPosCondicao(descricao)

- pre:
  - sistema.projetoCorrente.escritores →
    exists(e | e = sistema.usuarioCorrente)
- post:
  - sistema.posCondicaoCorrente.descricao = descricao

Remover Pós-condição do Caso de Uso

CasoDeUso::removerPosCondicao(posCondicao)

- pre:
  - sistema.projetoCorrente.escritores →
    exists(e | e = sistema.usuarioCorrente)
- post:
  - sistema.casoDeUsoCorrente.posCondicoes → removing(posCondicao)

Fluxo Alternativo do Caso de Uso

Incluir Fluxo Alternativo

Passo::incluirFluxoAlternativo()

- pre:
  - sistema.projetoCorrente.escritores →
    exists(e | e = sistema.usuarioCorrente)
- post:
  - sistema.fluxosAlternativos → including(FluxoAlternativo.new())

Remover Fluxo Alternativo

Passo::removerFluxoAlternativo(fluxoAlternativo)

- pre:
  - sistema.projetoCorrente.escritores →
    exists(e | e = sistema.usuarioCorrente)
- post:
  - sistema.fluxosAlternativos → removing(fluxoAlternativo)

Passo do Fluxo

Incluir Passo

Fluxo::incluirPasso(indice, descricao)

- pre:
  - sistema.projetoCorrente.escritores →
    exists(e | e = sistema.usuarioCorrente)
- post:
  - sistema.passos → including(Passo.new())

Editar Passo

Fluxo::editarPasso(descricao)

- pre:
  - sistema.projetoCorrente.escritores →
    exists(e | e = sistema.usuarioCorrente)
- post:
  - sistema.passoCorrente.descricao = descricao

Remover Passo

Fluxo::removerPasso(passo)

- pre:
  - sistema.projetoCorrente.escritores →
    exists(e | e = sistema.usuarioCorrente)
- post:
  - sistema.passos → removing(passo)
  - passo.destroy()

Diagramas de Colaboração

Inclusão

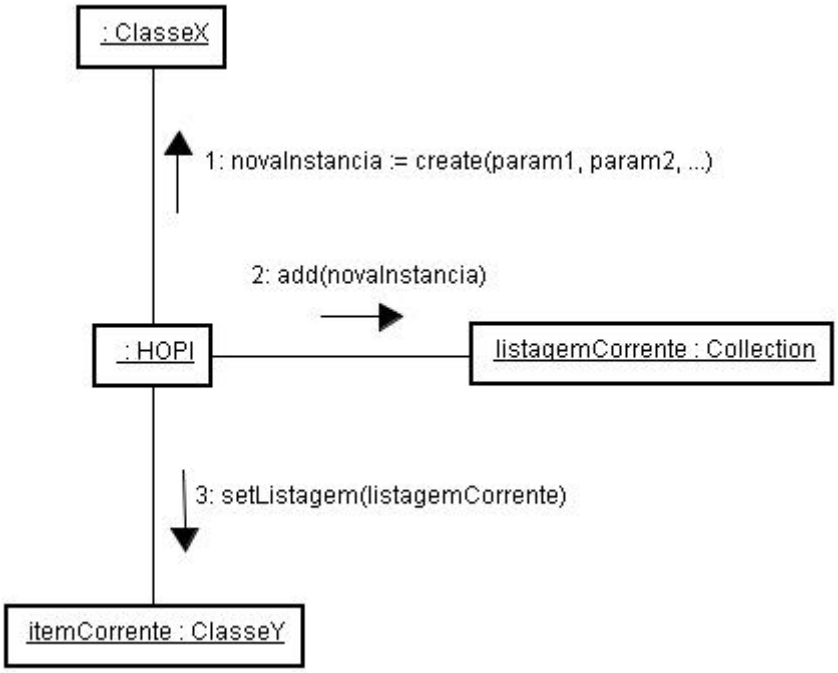

Figura 2. Diagrama de Colaboração Genérico de Inclusão

Edição

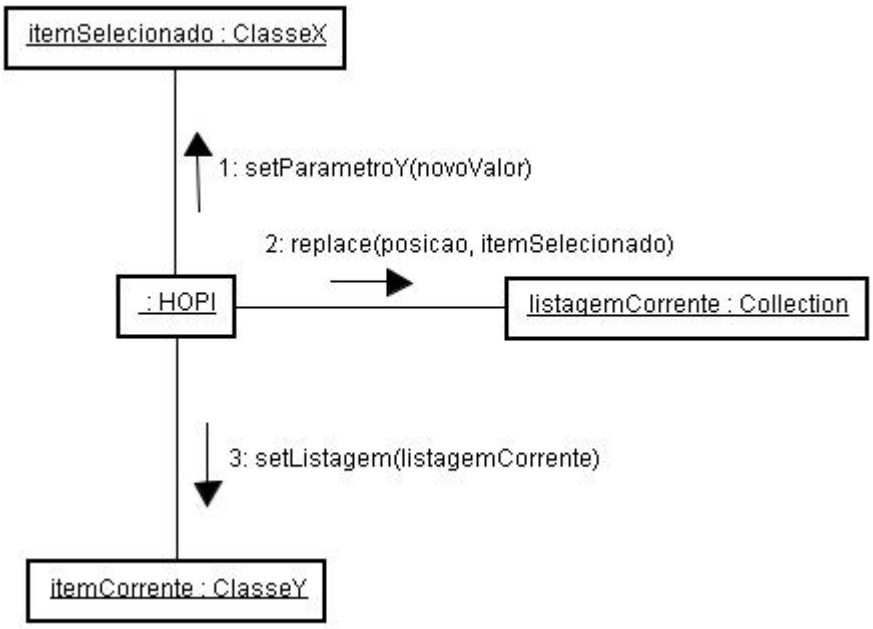

Figura 3. Diagrama de Colaboração Genérico de Edição

Remoção

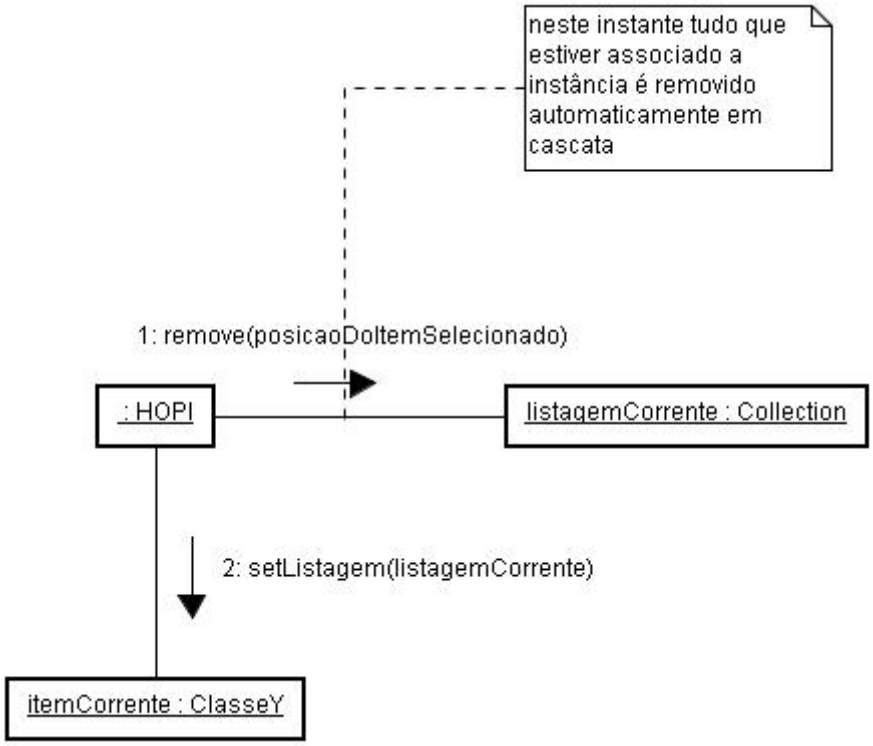

Figura 4. Diagrama de Colaboração Genérico de Remoção

Listagem

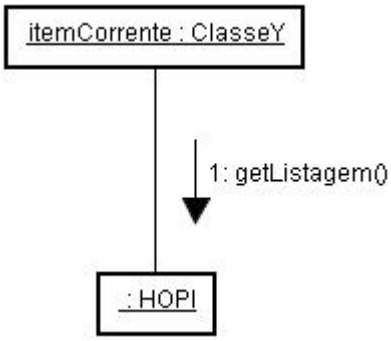

Figura 5. Diagrama de Colaboração Genérico de Listagem

Diagrama de Classes – Parte I

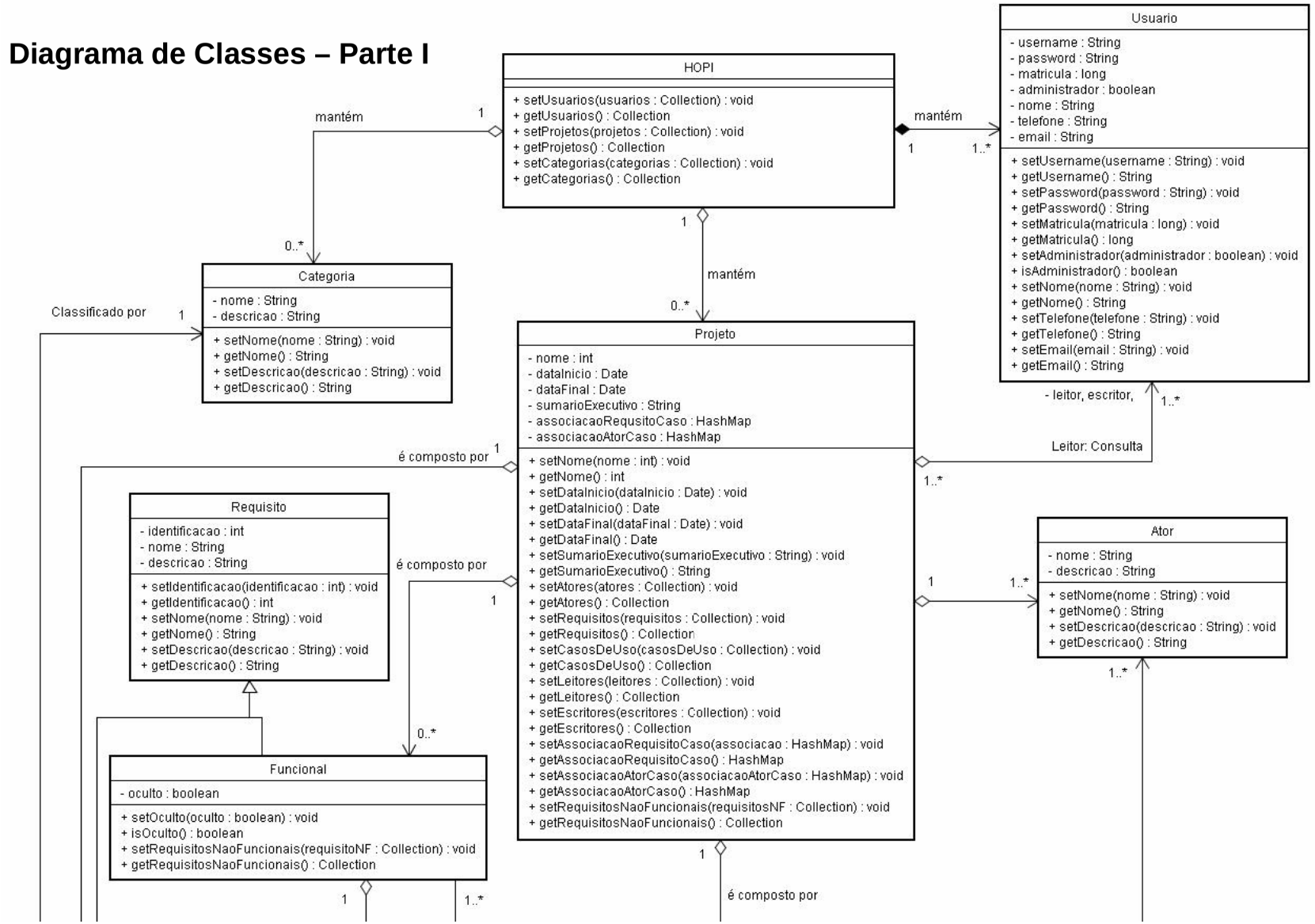

Figura 6 – Diagrama de Classes, parte I.

Diagrama de Classes – Parte II

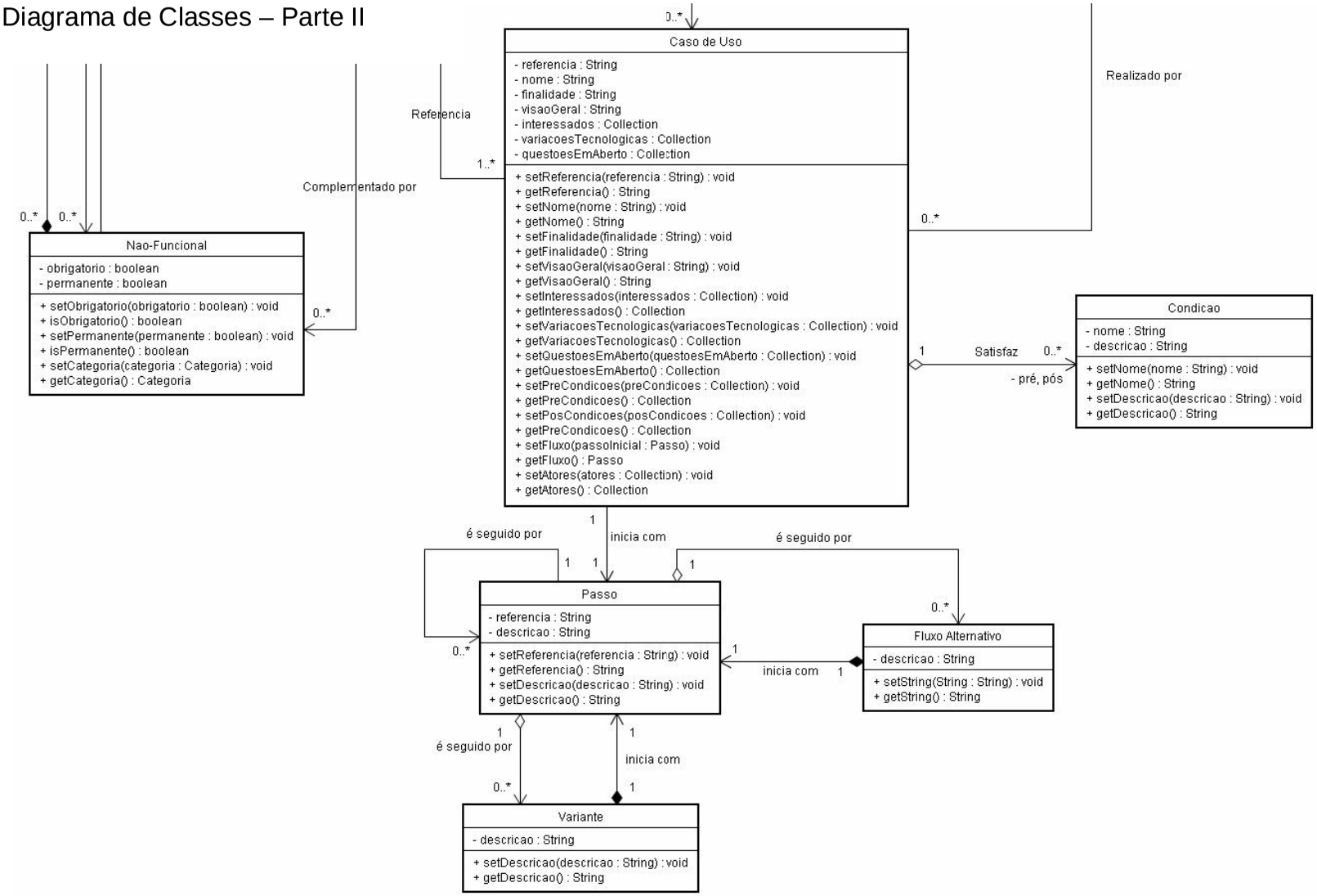

Figura 7 – Diagrama de Classes, parte II.

Diagramas de Interface

Diagrama de Estados de Navegação

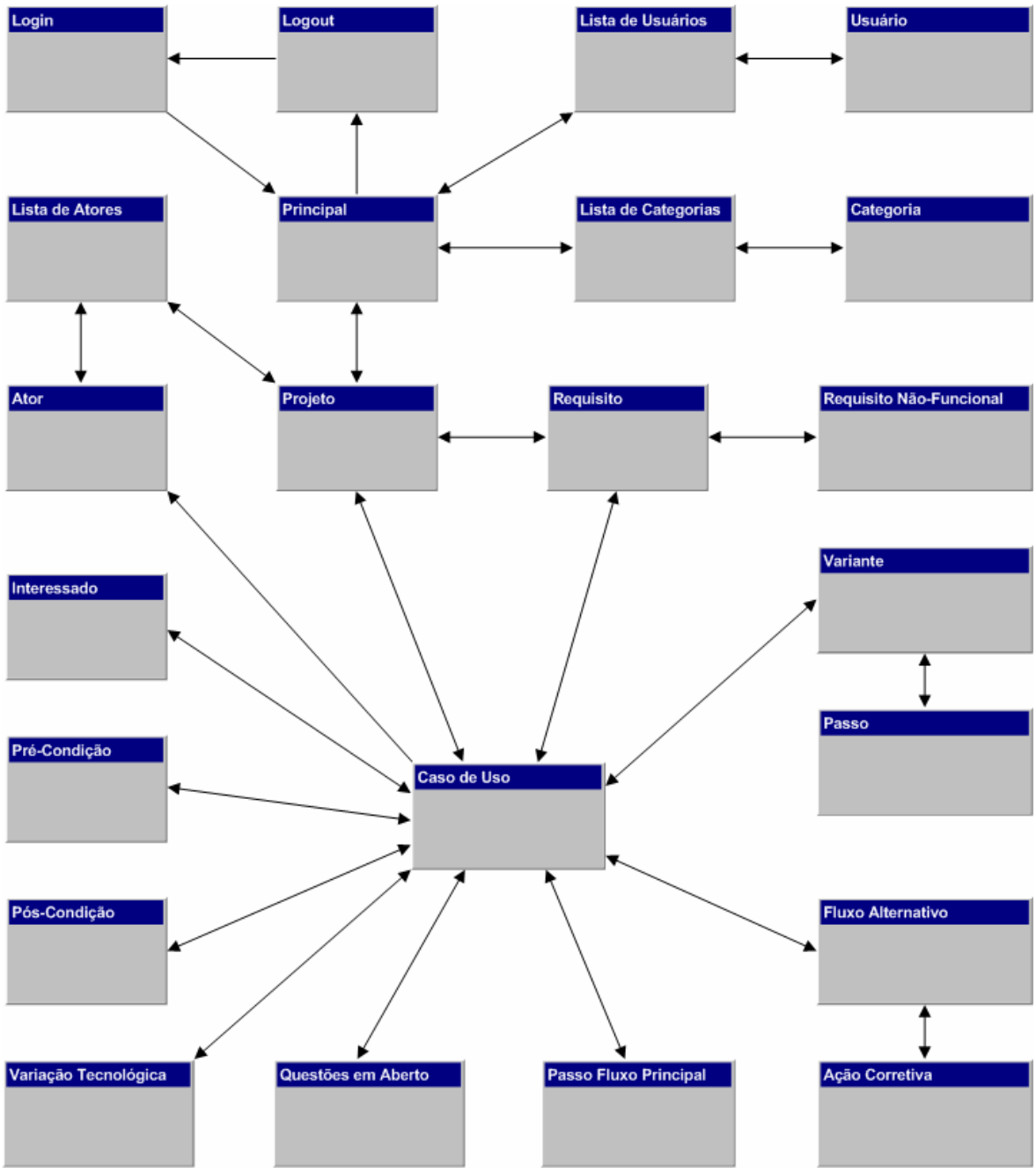

Figura 8. Diagrama de navegação entre as telas do sistema.

Protótipos das Telas do Sistema

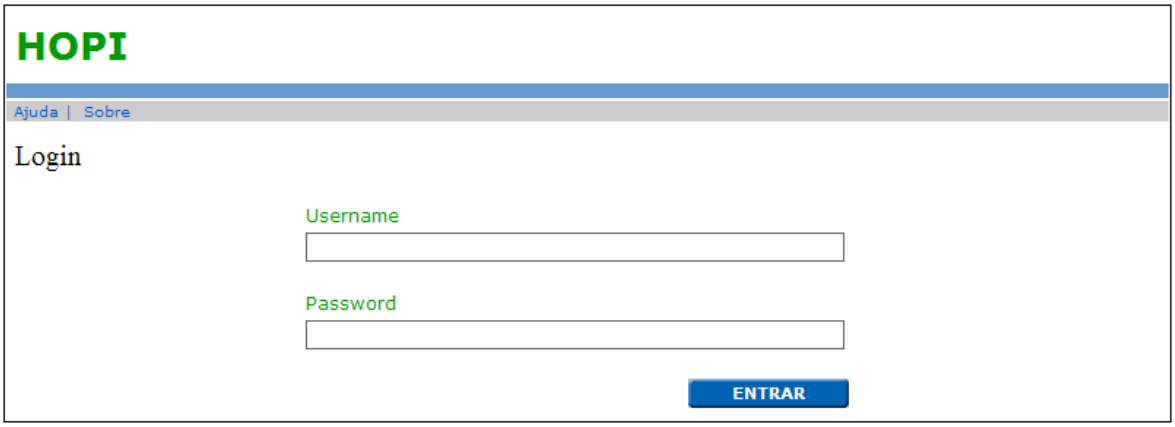

Figura 9. Tela de login do sistema.

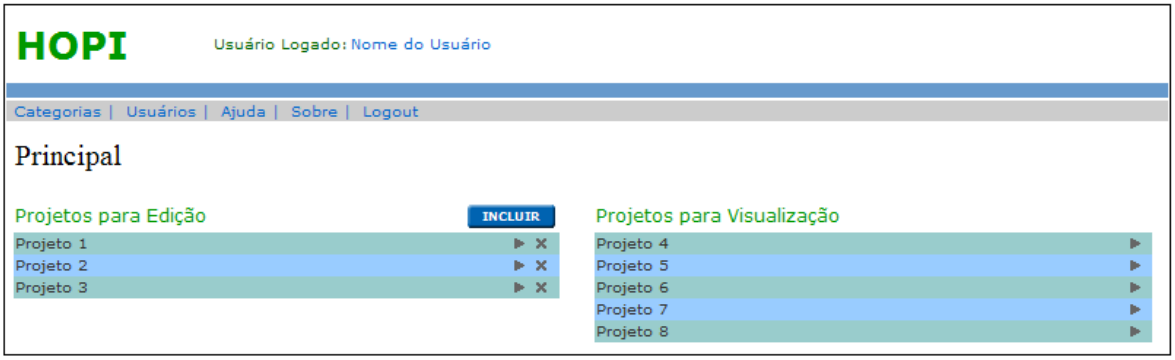

Figura 10. Tela principal do sistema.

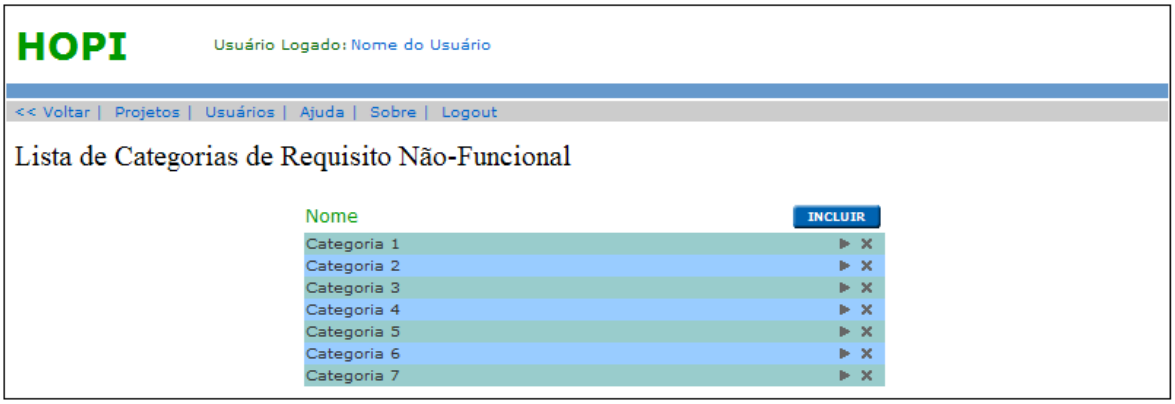

Figura 11. Tela com a listagem das categorias dos requisitos não-funcionais.

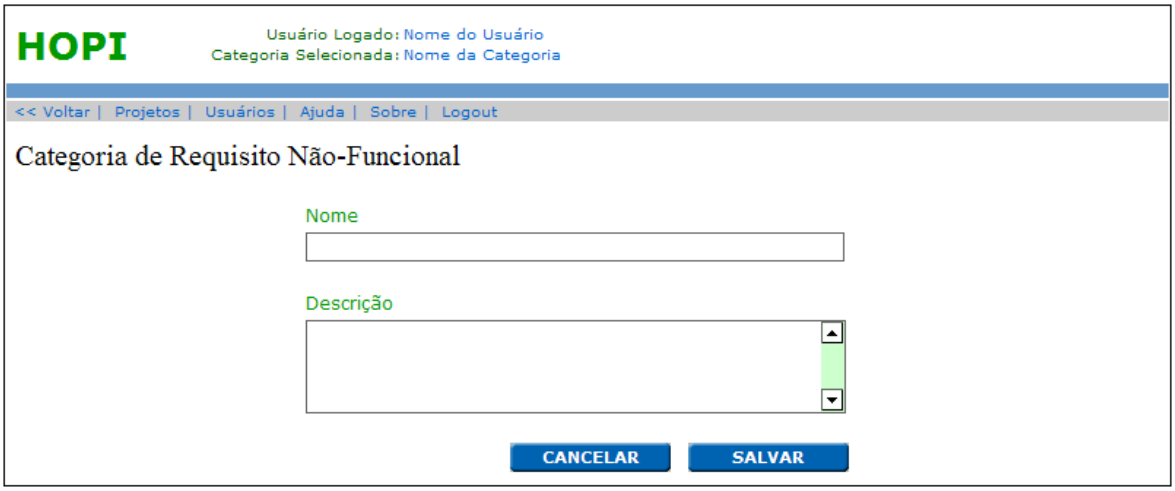

Figura 12. Tela com o formulário da categoria de requisito não-funcional.

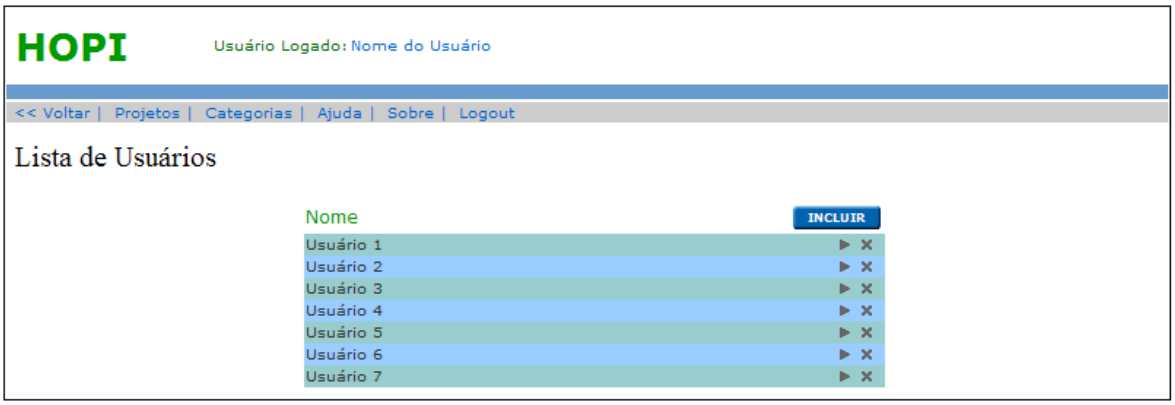

Figura 13. Tela com a listagem dos usuários do sistema.

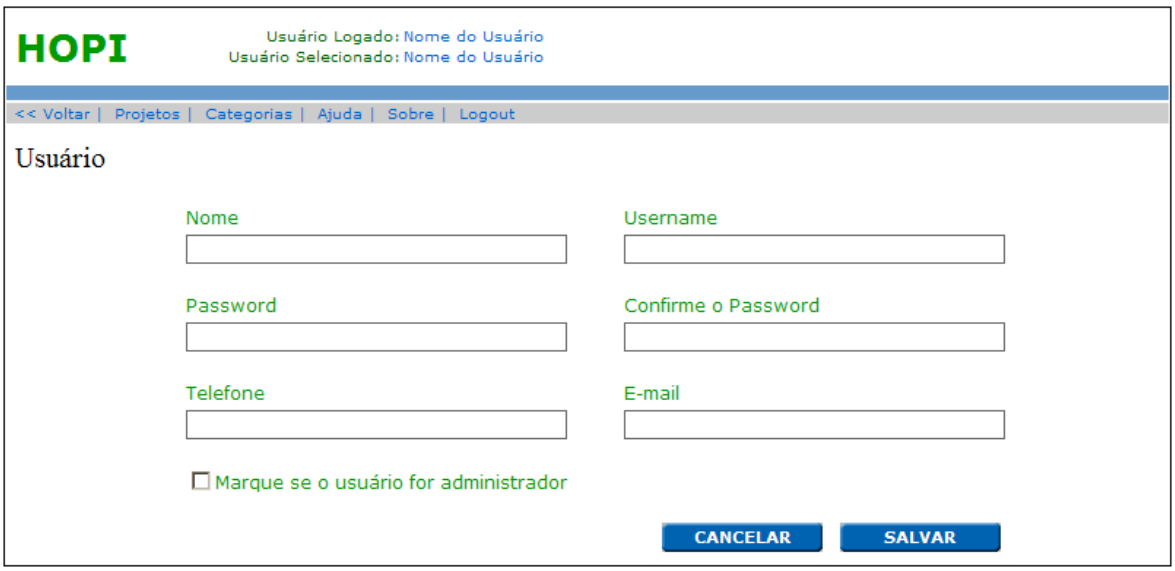

Figura 14. Tela com o formulário do usuário do sistema.

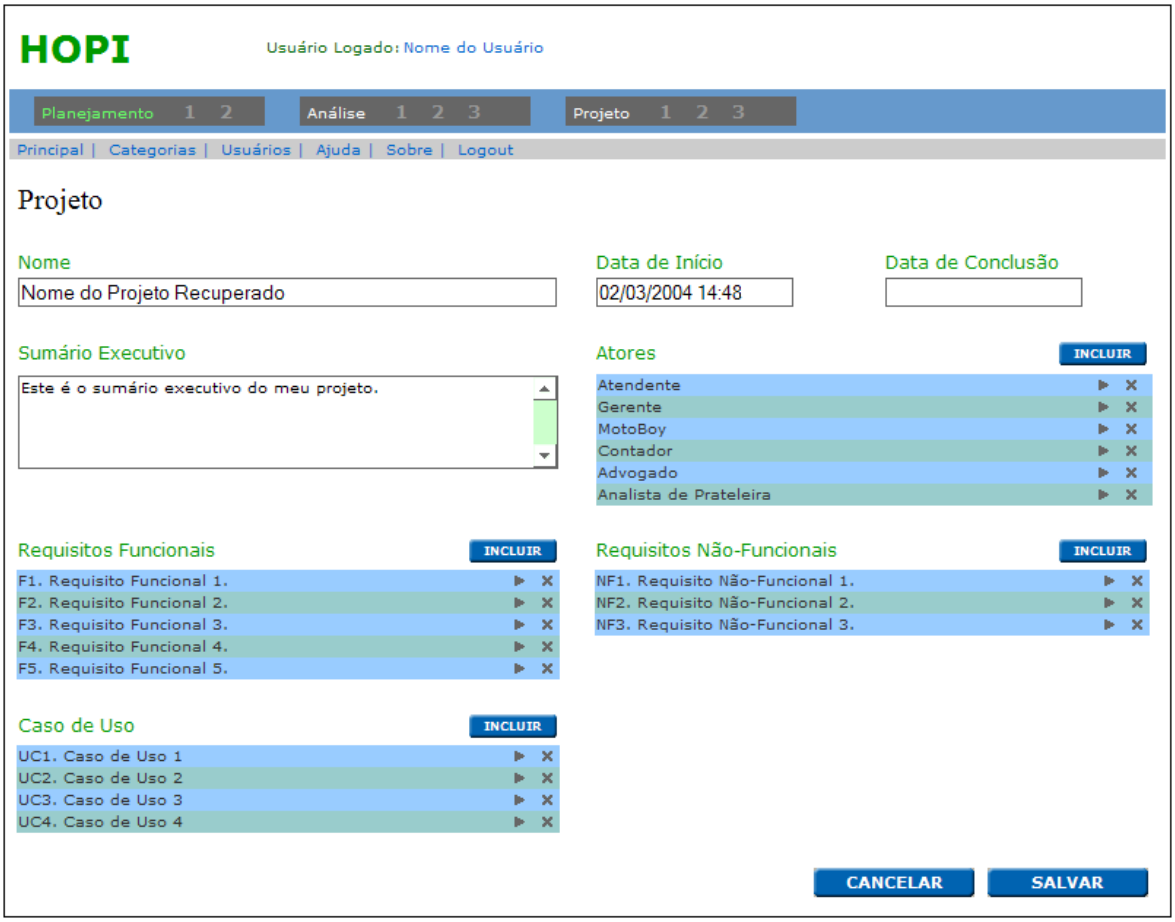

Figura 15. Tela com o formulário do projeto.

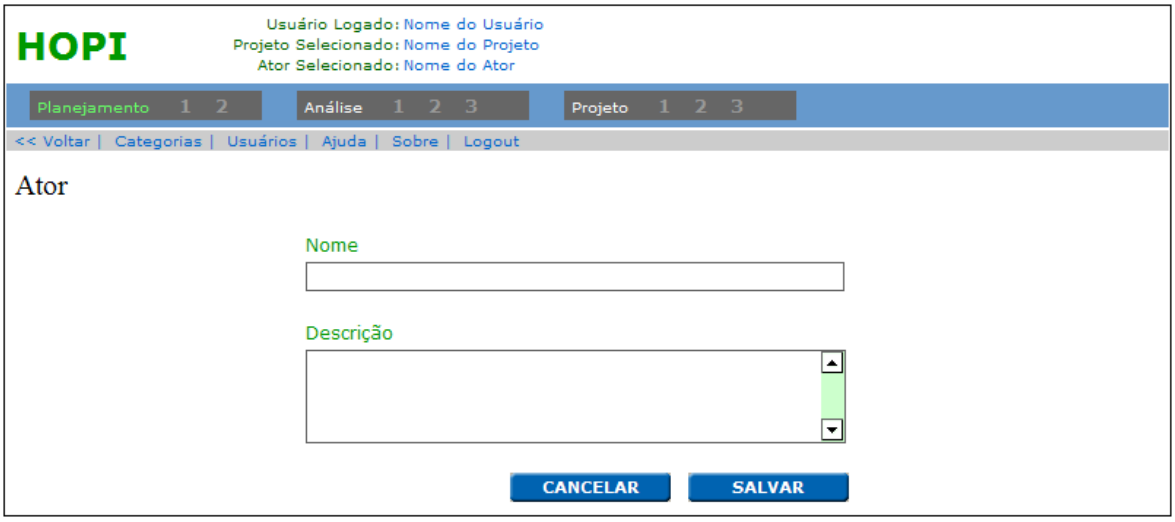

Figura 16. Tela com o formulário do ator.

Figura 17. Tela com o formulário do requisito funcional.

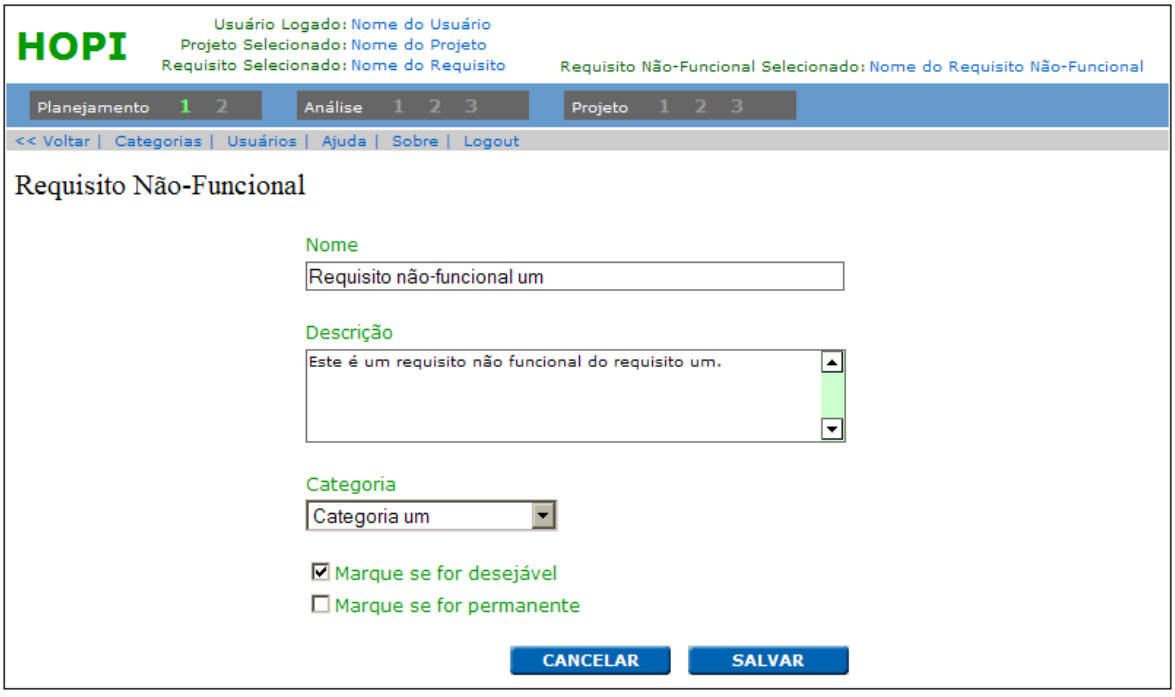

Figura 18. Tela com o formulário do requisito não-funcional.

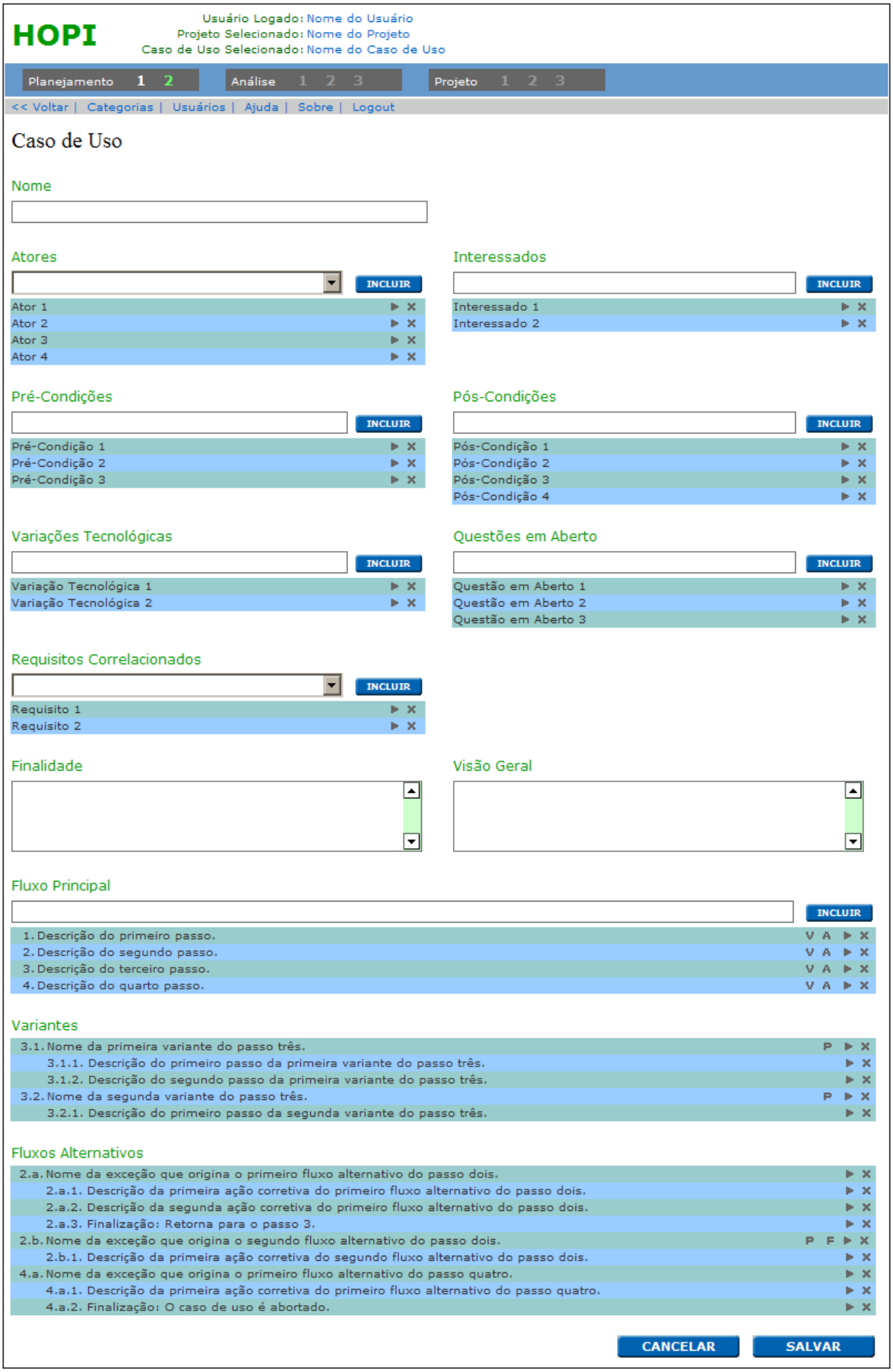

Figura 19. Tela com o formulário do caso de uso.

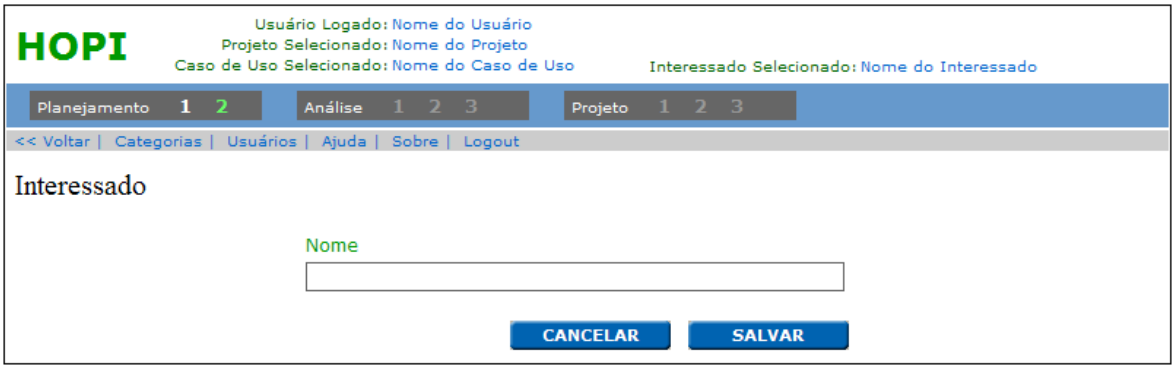

Figura 20. Tela com o formulário do interessado no caso de uso.

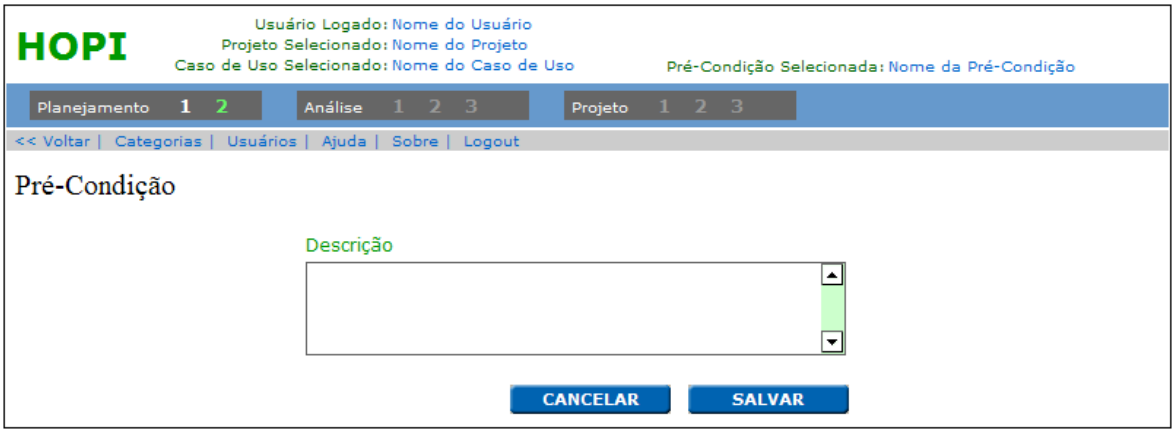

Figura 21. Tela com o formulário da pré-condição do caso de uso.

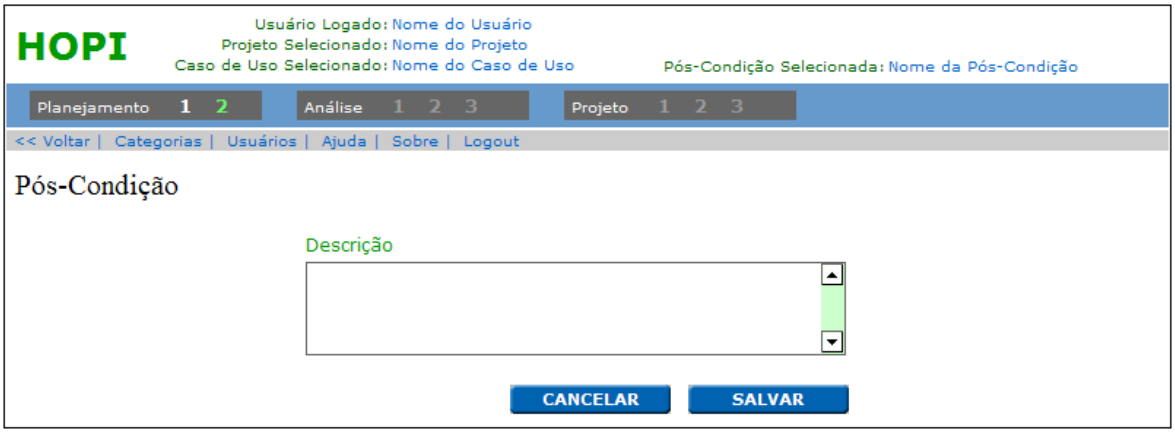

Figura 22. Tela com o formulário da pós-condição do caso de uso.

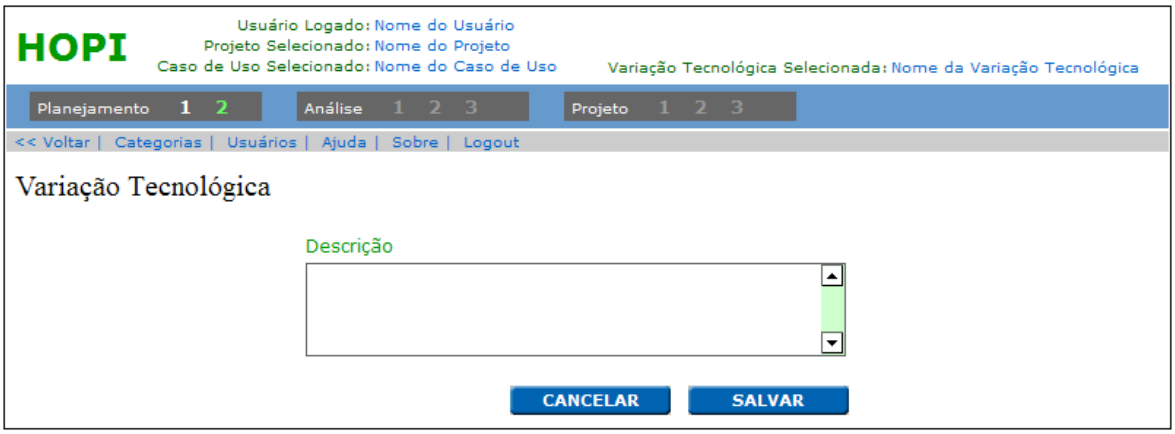

Figura 23. Tela com o formulário da variação tecnológica do caso de uso.

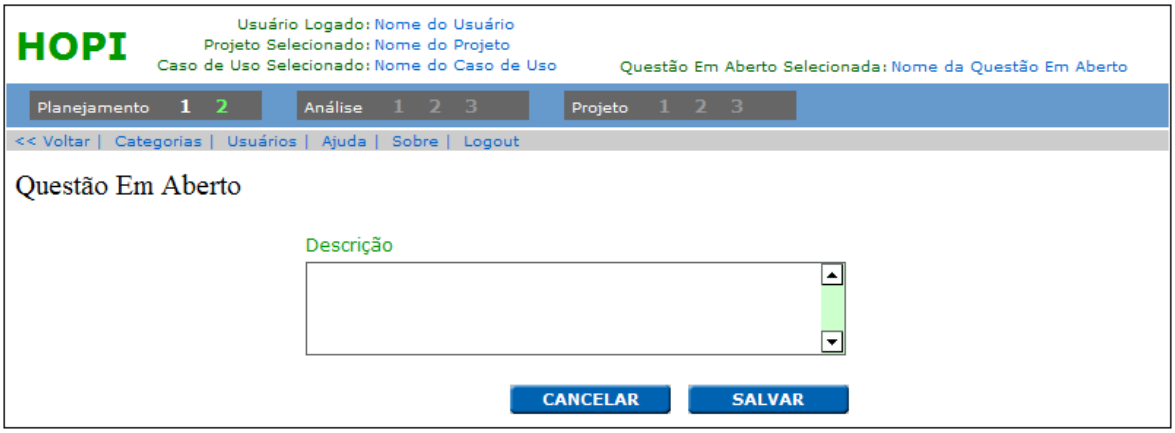

Figura 24. Tela com o formulário da questão em aberto do caso de uso.

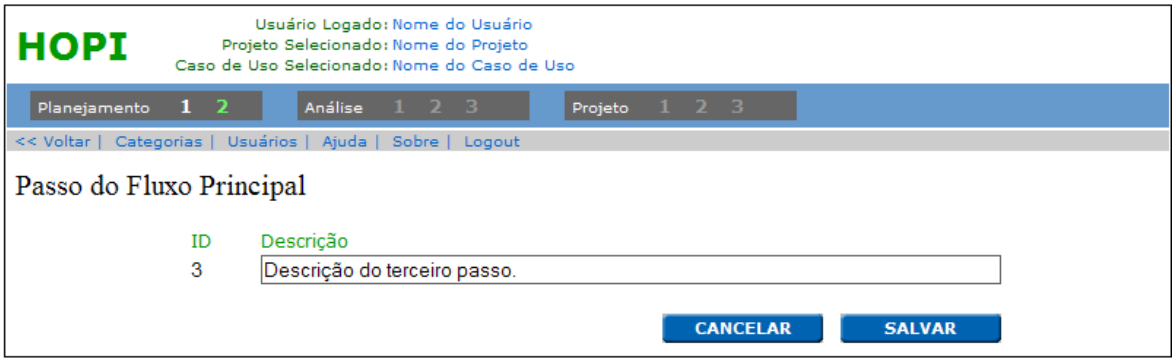

Figura 25. Tela com o formulário do passo do fluxo principal.

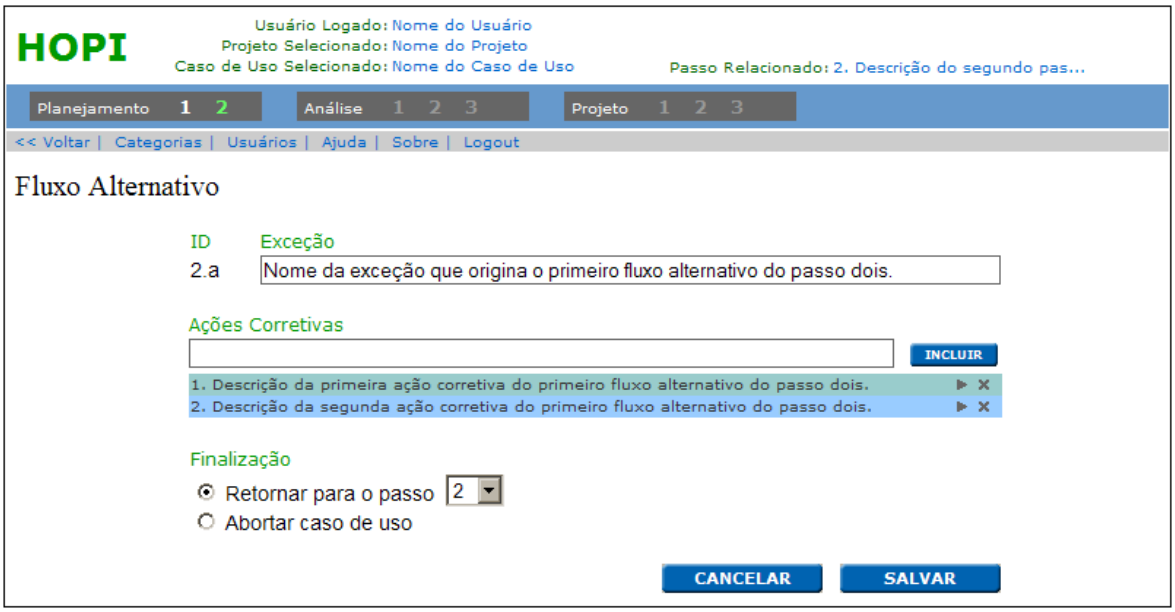

Figura 26. Tela com o formulário do fluxo alternativo.

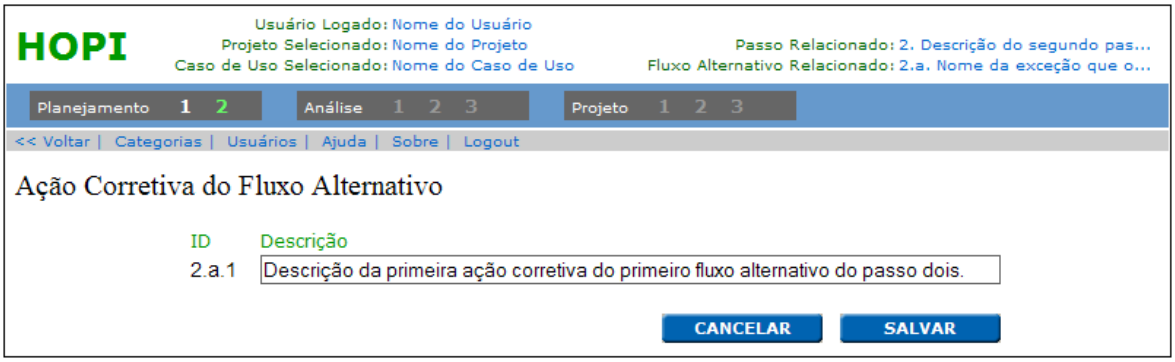

Figura 27. Tela com o formulário da ação corretiva do fluxo alternativo.

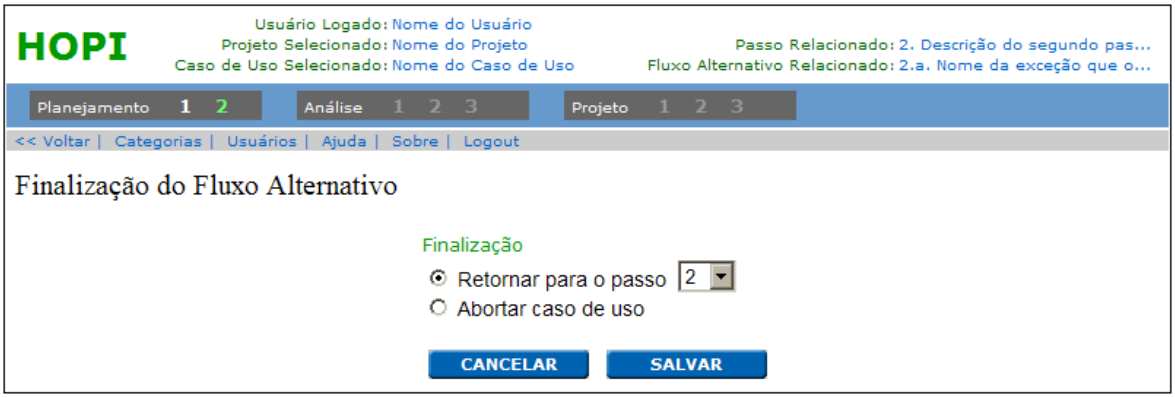

Figura 28. Tela com o formulário da finalização fluxo alternativo.

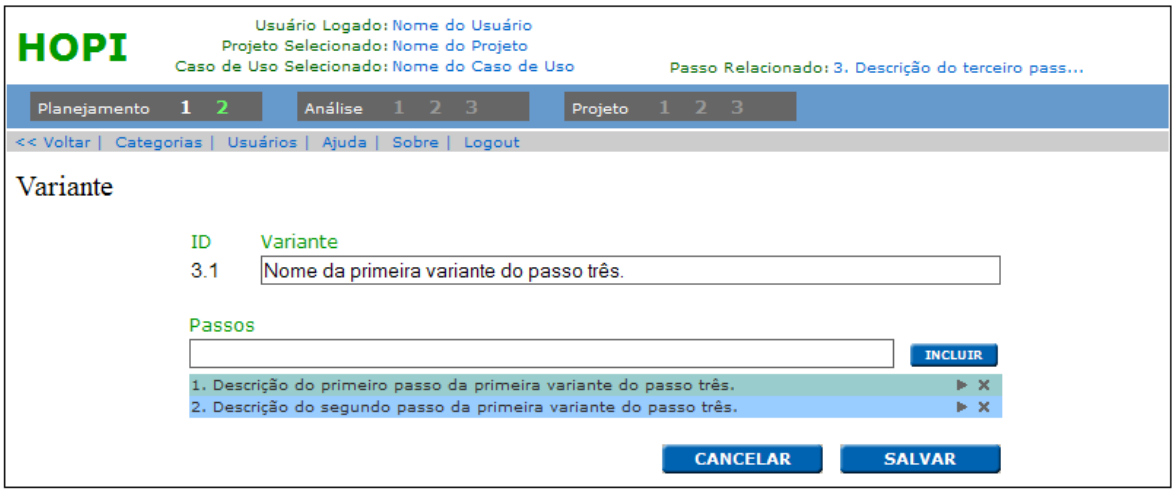

Figura 29. Tela com o formulário da variante do passo do fluxo principal.

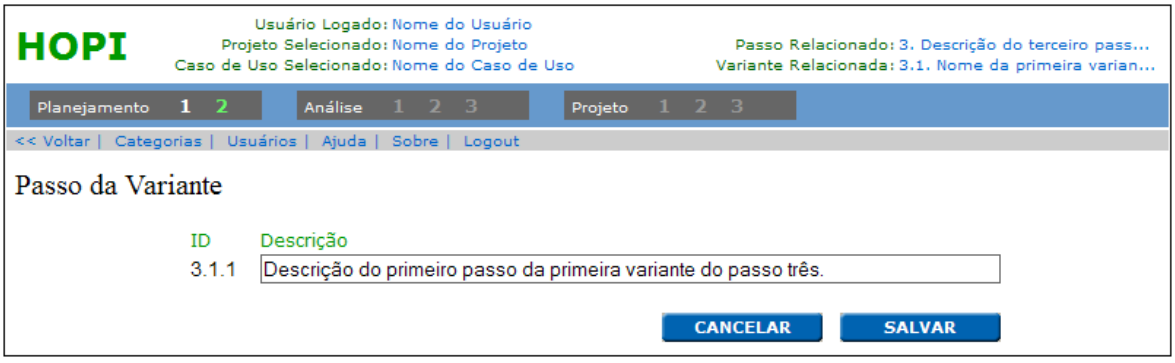

Figura 30. Tela com o formulário do passo da variante.

Diagramas de Estados das Interfaces

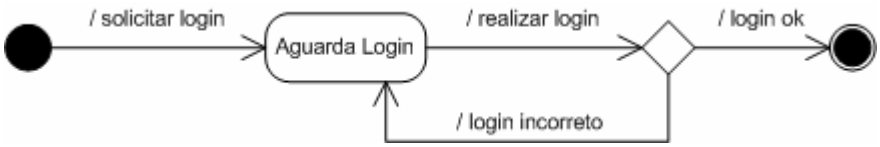

Figura 31. Diagrama de estados da tela de login.

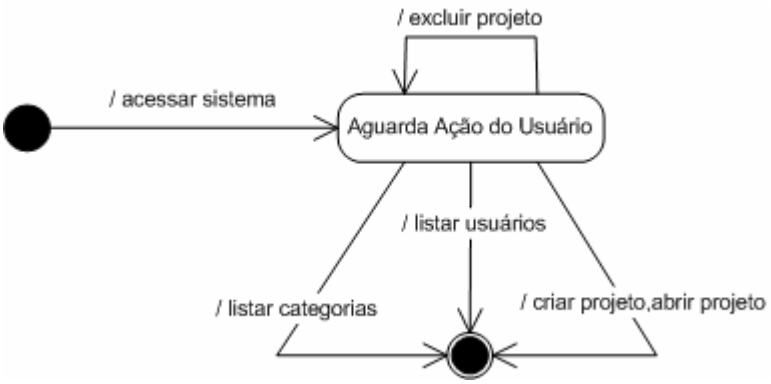

Figura 32. Diagrama de estados da tela principal do sistema.

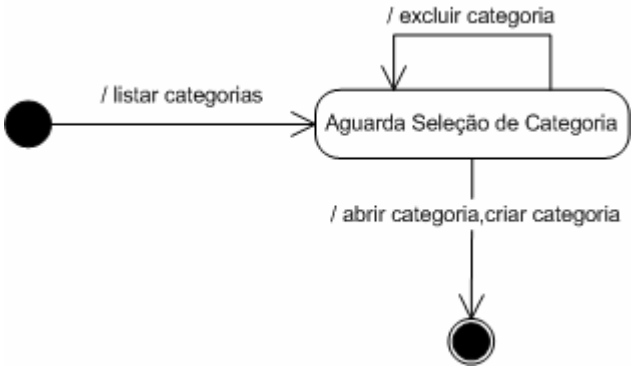

Figura 33. Diagrama de estados da listagem de categorias.

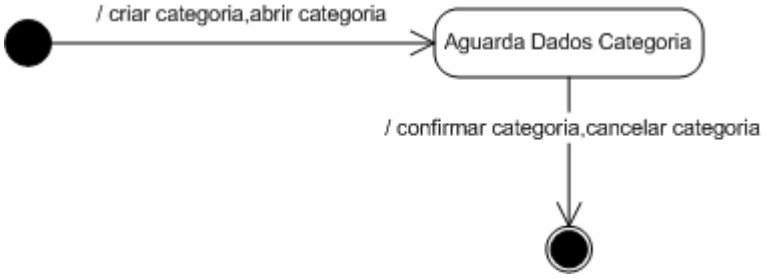

Figura 34. Diagrama de estados da tela de categoria.

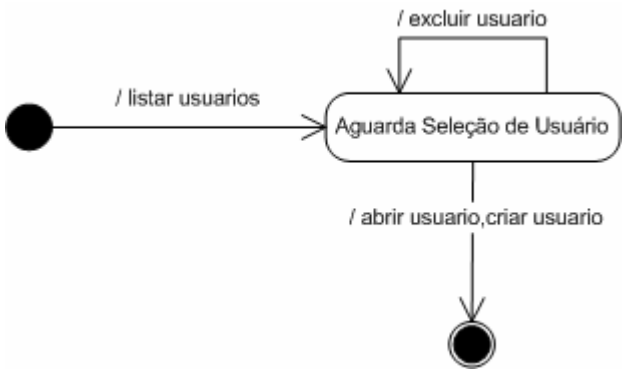

Figura 35. Diagrama de estados da tela de listagem dos usuários.

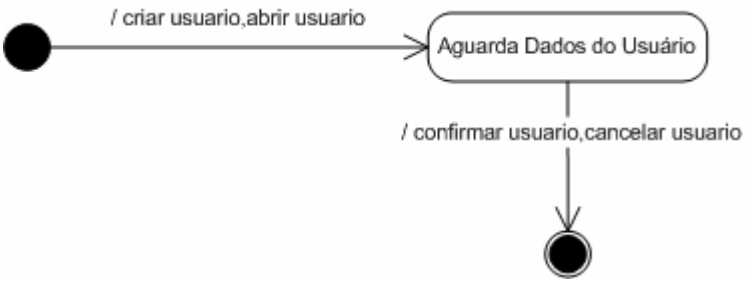

Figura 36. Diagrama de estados da tela de usuário.

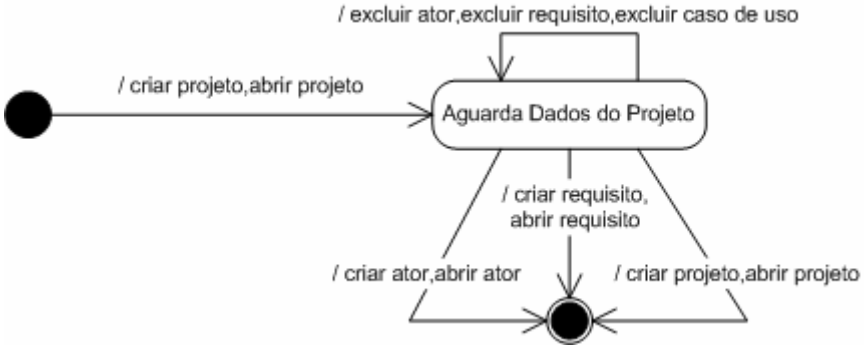

Figura 37. Diagrama de estados da tela de projeto.

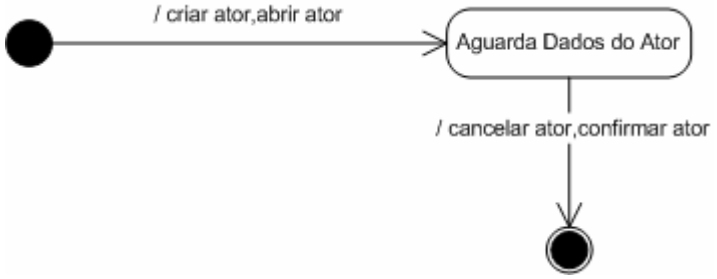

Figura 38. Diagrama de estados da tela de ator.

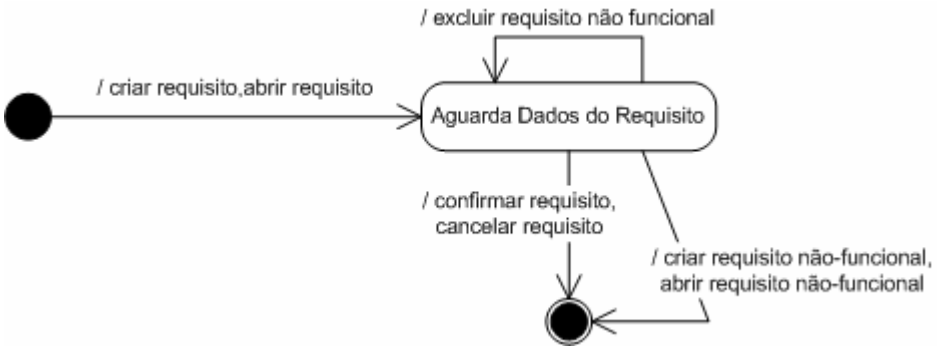

Figura 39. Diagrama de estados da tela de requisito funcional.

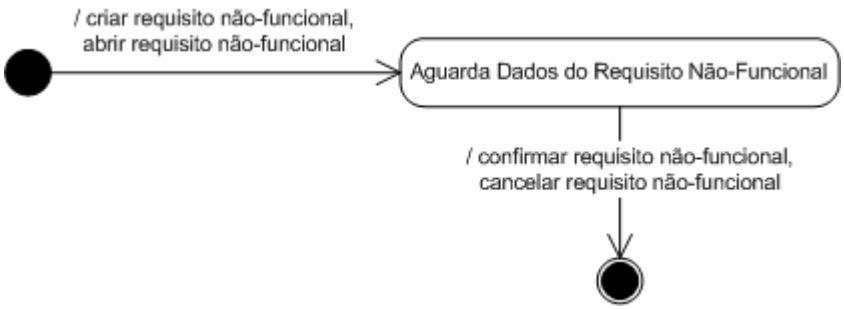

Figura 40. Diagrama de estados da tela de requisito não-funcional.

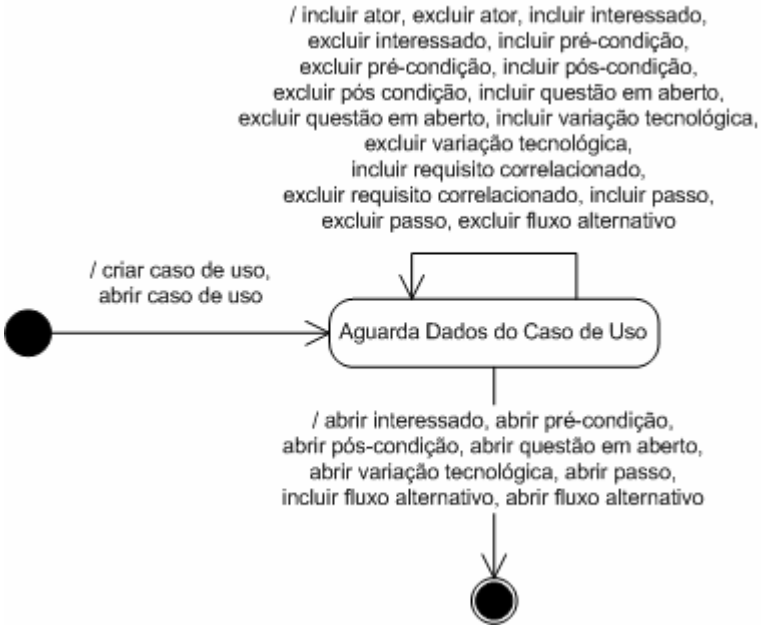

Figura 41. Diagrama de estados da tela de caso de uso.

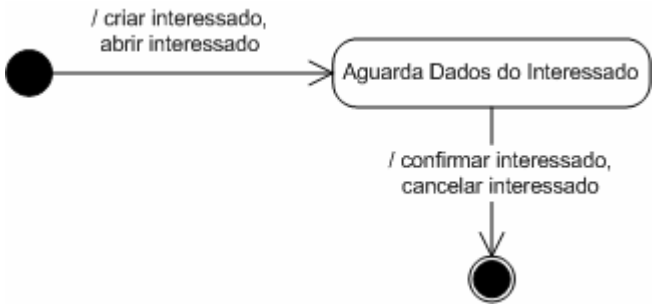

Figura 42. Diagrama de estados da tela de interessado.

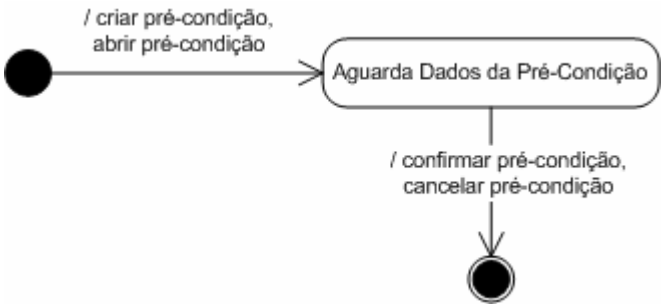

Figura 43. Diagrama de estados da tela de pré-condição.

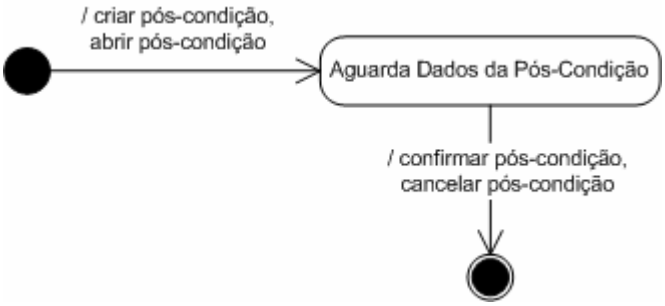

Figura 44. Diagrama de estados da tela de pós-condição.

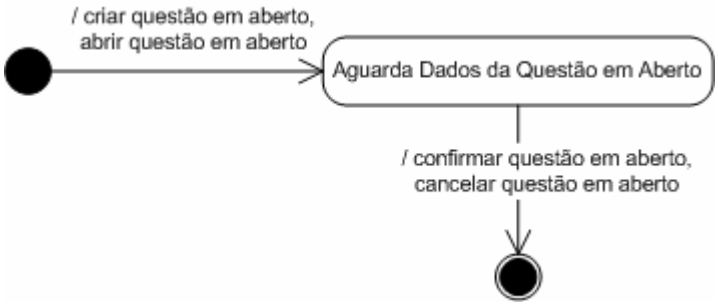

Figura 45. Diagrama de estados da tela de questão em aberto.

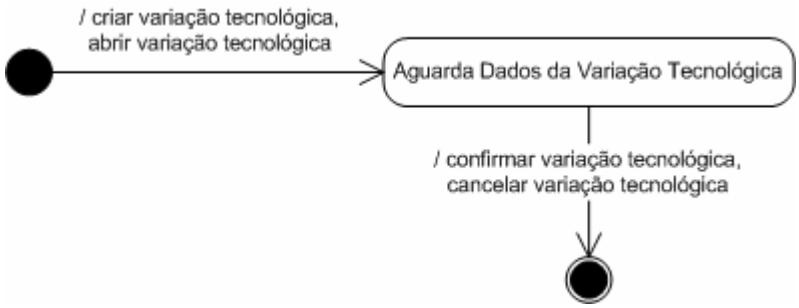

Figura 46. Diagrama de estados da tela de variação tecnológica.

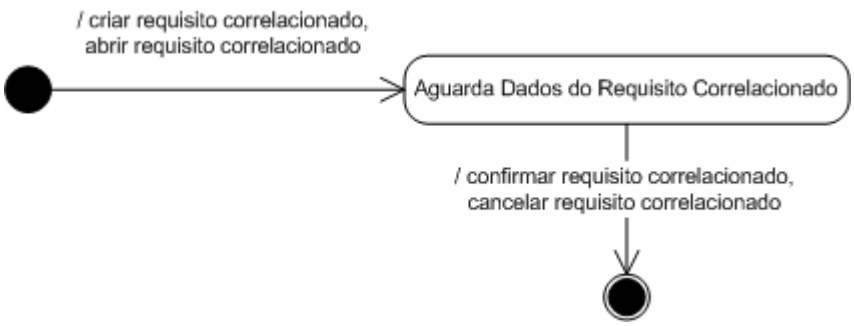

Figura 47. Diagrama de estados da tela de requisito correlacionado.

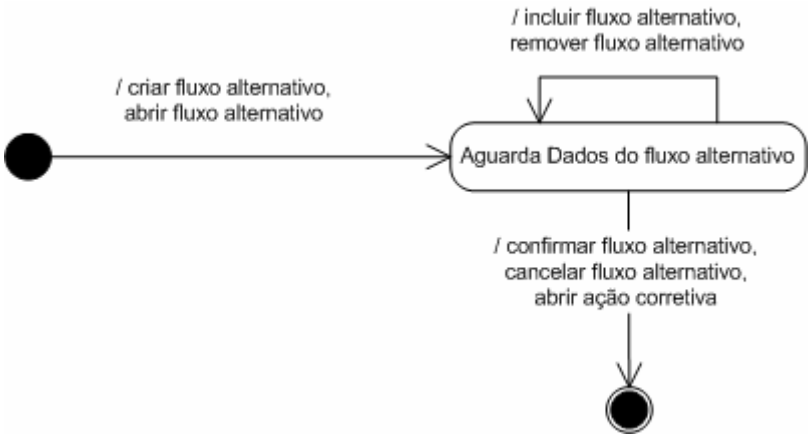

Figura 48. Diagrama de estados da tela de fluxo alternativo.

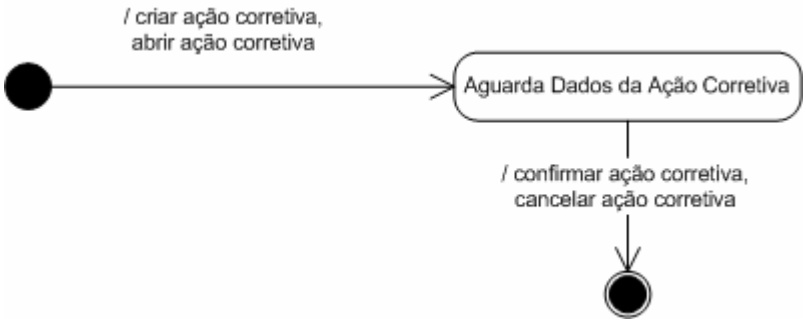

Figura 49. Diagrama de estados da tela de ação corretiva do fluxo alternativo.

Diagramas da Modelagem de Dados – Parte I

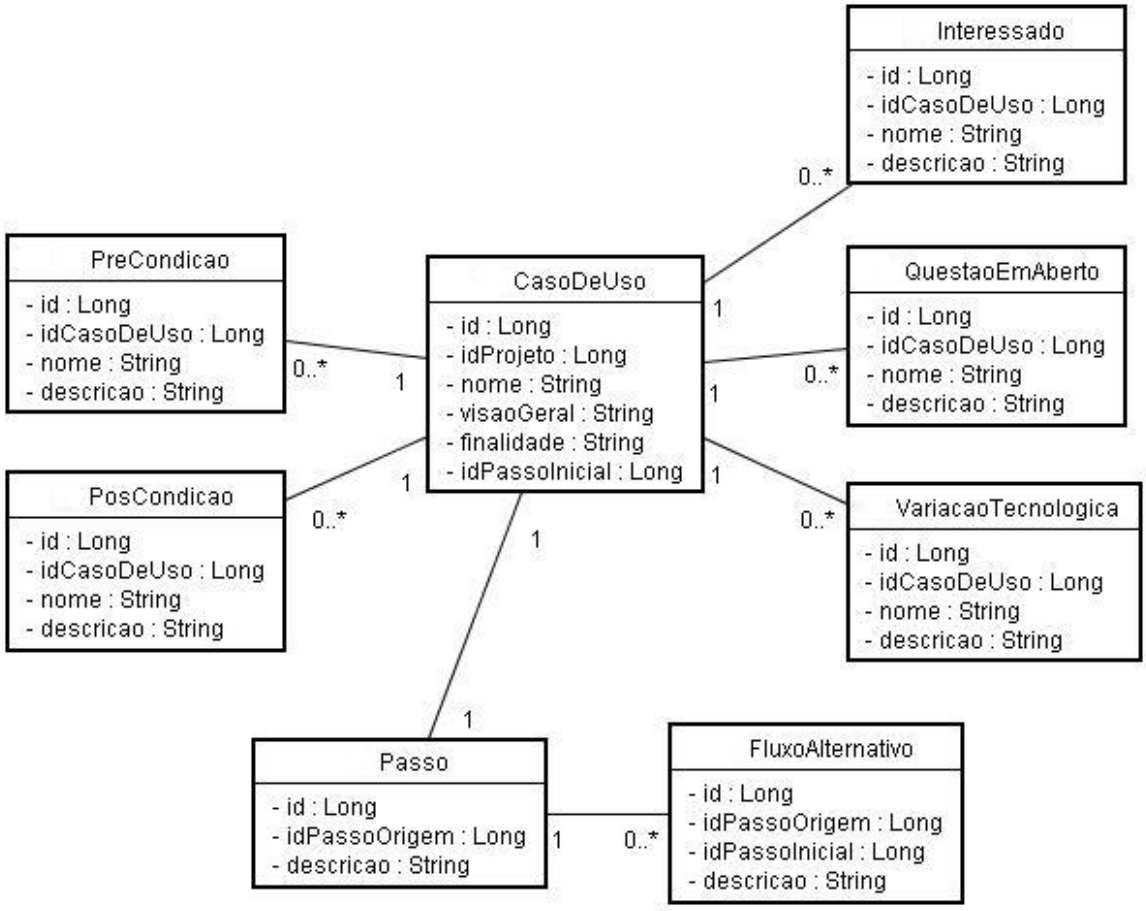

Figura 50. Diagrama da Modelagem de Dados, parte I.

Diagramas da Modelagem de Dados – Parte II

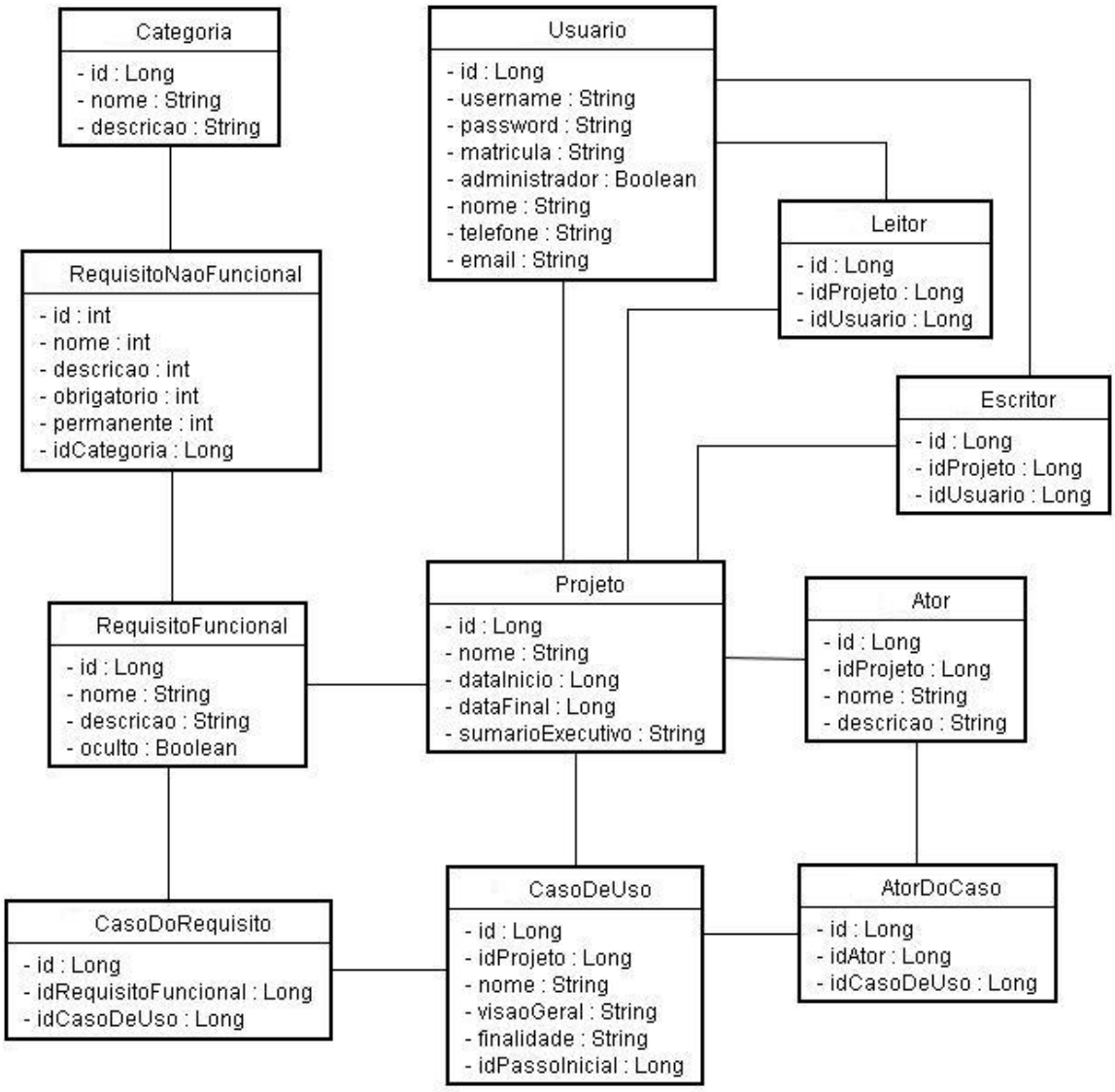

Figura 51. Diagrama da Modelagem de Dados, parte II.

Código Fonte

PACKAGE ORG.HOPI.DTO

DTOSessao.java

```
/**
 * Objeto com os dados do usuário a ser incluído na sessão.
 * @since 31/10/2004
 * @author Marcel Horner e Rafael Pires
 */
package org.hopi.DTO;

public class DTOSessao {
    Long id;
    String username;
    String nomeCompleto;

    public DTOSessao() {
    }
    /**
     * @return Returns the id.
     */
    public Long getId() {
        return id;
    }
    /**
     * @param id The id to set.
     */
    public void setId(Long id) {
        this.id = id;
    }
    /**
     * @return Returns the nomeCompleto.
     */
    public String getNomeCompleto() {
        return nomeCompleto;
    }
    /**
     * @param nomeCompleto The nomeCompleto to set.
     */
    public void setNomeCompleto(String nomeCompleto) {
        this.nomeCompleto = nomeCompleto;
    }
    /**
     * @return Returns the username.
     */
    public String getUsername() {
        return username;
    }
    /**
     * @param username The username to set.
     */
    public void setUsername(String username) {
        this.username = username;
    }
}
```

PACKAGE ORG.HOPI.WEB.DELEGATE

WebDelegate.java

```
package org.hopi.web.delegate;

import java.util.ArrayList;
import java.util.Iterator;

import javax.ejb.EJBException;

import org.hopi.ejb.entity.VOAtor;
import org.hopi.ejb.entity.VOCasoDeUso;
import org.hopi.ejb.entity.VOCategoria;
import org.hopi.ejb.entity.VOEscritor;
import org.hopi.ejb.entity.VOInteressado;
import org.hopi.ejb.entity.VOPosCondicao;
import org.hopi.ejb.entity.VOPreCondicao;
import org.hopi.ejb.entity.VOProjeto;
import org.hopi.ejb.entity.VOQuestaoEmAberto;
import org.hopi.ejb.entity.VORequisitoFuncional;
import org.hopi.ejb.entity.VORequisitoNaoFuncional;
import org.hopi.ejb.entity.VOUsuario;
import org.hopi.ejb.entity.VOVariacaoTecnologica;
import
org.hopi.ejb.entity.facade.CategoriaDeRequisitoNaoFuncionalFacade;
import
org.hopi.ejb.entity.facade.CategoriaDeRequisitoNaoFuncionalFacadeHome;
import org.hopi.ejb.entity.facade.ProjetoFacade;
import org.hopi.ejb.entity.facade.ProjetoFacadeHome;
import org.hopi.ejb.entity.facade.UsuarioFacade;
import org.hopi.ejb.entity.facade.UsuarioFacadeHome;
import org.hopi.ejb.session.UsuarioBusinessFacade;
import org.hopi.ejb.session.UsuarioBusinessFacadeHome;
import org.hopi.ejb.util.JNDI;
import org.hopi.ejb.util.ServiceLocator;
import org.hopi.exception.LoginInvalidoException;

/**
 *
 * Responsável por centralizar as requisições da camada web para a de
negócios.
 *
 * @since 30/10/2004
 * @author Marcel Horner e Rafael Pires
 */
public class WebDelegate
{

    public WebDelegate()
    {
        super();
    }

    public VOUsuario validarLogin(String username, String password)
throws LoginInvalidoException, Exception
    {
        try {
            UsuarioBusinessFacadeHome usuarioHome =
(UsuarioBusinessFacadeHome)ServiceLocator.getLocalHome(JNDI.BUSINESS_FAC
ADE_USUARIO);
```

```
            UsuarioBusinessFacade usuario = usuarioHome.create();

            VOUsuario vo = new VOUsuario();
            vo.setUsername(username);
            vo.setPassword(password);

            return usuario.validarLogin(vo);
        } catch (EJBException ejbe) {
            ejbe.printStackTrace();
            throw new Exception(ejbe);
        } catch (LoginInvalidoException lie) {
            lie.printStackTrace();
            throw new LoginInvalidoException();
        } catch (Exception e) {
            e.printStackTrace();
            throw new Exception(e);
        }
    }

    public ArrayList getPosCondicoes(Long idCasoDeUso)
    {
        ArrayList al = new ArrayList();
        try {
            ProjetoFacadeHome projetoHome =
(ProjetoFacadeHome)ServiceLocator.getLocalHome(JNDI.PROJETO_FACADE);
            ProjetoFacade projeto = projetoHome.create();
            VOPosCondicao vo = new VOPosCondicao();
            al = (ArrayList)projeto.getAllByRelatedId(vo, idCasoDeUso);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return al;
    }

    public ArrayList getPreCondicoes(Long idCasoDeUso)
    {
        ArrayList al = new ArrayList();
        try {
            ProjetoFacadeHome projetoHome =
(ProjetoFacadeHome)ServiceLocator.getLocalHome(JNDI.PROJETO_FACADE);
            ProjetoFacade projeto = projetoHome.create();
            VOPreCondicao vo = new VOPreCondicao();
            al = (ArrayList)projeto.getAllByRelatedId(vo, idCasoDeUso);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return al;
    }

    public VOProjeto getProjeto(Long idProjeto)
    {
        VOProjeto vo = new VOProjeto();
        try {
            ProjetoFacadeHome projetoHome =
(ProjetoFacadeHome)ServiceLocator.getLocalHome(JNDI.PROJETO_FACADE);
            ProjetoFacade projeto = projetoHome.create();
            vo = projeto.findById(idProjeto);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return vo;
```

```
    }

    public ArrayList getAtores(Long idProjeto)
    {
        ArrayList al = new ArrayList();
        try {
            ProjetoFacadeHome projetoHome =
(ProjetoFacadeHome)ServiceLocator.getLocalHome(JNDI.PROJETO_FACADE);
            ProjetoFacade projeto = projetoHome.create();
            VOAtor vo = new VOAtor();
            al = (ArrayList)projeto.getAllByRelatedId(vo, idProjeto);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return al;
    }

    public ArrayList getAtoresCasoDeUso(Long idCasoDeUso)
    {
        ArrayList al = new ArrayList();
        try {
            ProjetoFacadeHome projetoHome =
(ProjetoFacadeHome)ServiceLocator.getLocalHome(JNDI.PROJETO_FACADE);
            ProjetoFacade projeto = projetoHome.create();
            al =
(ArrayList)projeto.getAllAtoresDoCasoByIdCaso(idCasoDeUso);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return al;
    }

    public ArrayList getAtoresNaoRelacionados(Long idProjeto, Long
idCasoDeUso)
    {
        ArrayList al1 = getAtores(idProjeto);
        ArrayList al2 = getAtoresCasoDeUso(idCasoDeUso);
        VOAtor vo1;
        VOAtor vo2;

        Iterator it = al2.iterator();

        while ( it.hasNext() )
        {
           vo2 = (VOAtor) it.next();
           for ( int i = al1.size() ; i > 0 ; i-- )
           {
              vo1 = (VOAtor) al1.get(i);
              if ( vo1.getNome().compareTo(vo2.getNome()) == 0 )
              {
                  al1.remove(i);
              }
           }
        }

        return al1;
    }

    public ArrayList getRequisitosFuncionais(Long idProjeto)
    {
        ArrayList al = new ArrayList();
```

```
        try {
            ProjetoFacadeHome projetoHome =
(ProjetoFacadeHome)ServiceLocator.getLocalHome(JNDI.PROJETO_FACADE);
            ProjetoFacade projeto = projetoHome.create();
            VORequisitoFuncional vo = new VORequisitoFuncional();
            al = (ArrayList)projeto.getAllByRelatedId(vo, idProjeto);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return al;
    }

    public ArrayList getRequisitosFuncionaisRelacionados(Long
idCasoDeUso)
    {
        ArrayList al = new ArrayList();
        try {
            ProjetoFacadeHome projetoHome =
(ProjetoFacadeHome)ServiceLocator.getLocalHome(JNDI.PROJETO_FACADE);
            ProjetoFacade projeto = projetoHome.create();
            al =
(ArrayList)projeto.getAllRequisitosByIdCaso(idCasoDeUso);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return al;
    }

    public ArrayList getRequisitosFuncionaisNaoRelacionados(Long
idProjeto, Long idCasoDeUso)
    {
        ArrayList al1 = getRequisitosFuncionais(idProjeto);
        ArrayList al2 =
getRequisitosFuncionaisRelacionados(idCasoDeUso);
        VORequisitoFuncional vo1;
        VORequisitoFuncional vo2;

        Iterator it = al2.iterator();

        while ( it.hasNext() )
        {
           vo2 = (VORequisitoFuncional) it.next();
           for ( int i = al1.size() ; i > 0 ; i-- )
           {
              vo1 = (VORequisitoFuncional) al1.get(i);
              if ( vo1.getNome().compareTo(vo2.getNome()) == 0 )
              {
                  al1.remove(i);
              }
           }
        }

        return al1;
    }

    public VORequisitoNaoFuncional getRequisitoNaoFuncional(Long
idRequisitoNaoFuncional)
    {
        VORequisitoNaoFuncional vo = new VORequisitoNaoFuncional();
        try {
```

```
            ProjetoFacadeHome projetoHome = (ProjetoFacadeHome)
ServiceLocator.getLocalHome(JNDI.PROJETO_FACADE);
            ProjetoFacade     projeto     = projetoHome.create();
            vo = (VORequisitoNaoFuncional) projeto.getById(vo,
idRequisitoNaoFuncional);
        } catch(Exception e) {
            e.printStackTrace();
        }
        return vo;
    }

    public ArrayList getRequisitosNaoFuncionais(Long idProjeto)
    {
        ArrayList al = new ArrayList();
        try {
            ProjetoFacadeHome projetoHome =
(ProjetoFacadeHome)ServiceLocator.getLocalHome(JNDI.PROJETO_FACADE);
            ProjetoFacade projeto = projetoHome.create();
            VORequisitoNaoFuncional vo = new VORequisitoNaoFuncional();
            vo.setIdProjeto(new Long(0));
            al = (ArrayList)projeto.getAllByRelatedId(vo, idProjeto);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return al;
    }

    public ArrayList getRequisitosNaoFuncionaisAssociados(Long
idRequisito)
    {
        ArrayList al = new ArrayList();
        try {
            ProjetoFacadeHome projetoHome =
(ProjetoFacadeHome)ServiceLocator.getLocalHome(JNDI.PROJETO_FACADE);
            ProjetoFacade projeto = projetoHome.create();
            VORequisitoNaoFuncional vo = new VORequisitoNaoFuncional();
            al = (ArrayList)projeto.getAllByRelatedId(vo, idRequisito);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return al;
    }

    public VOCasoDeUso getCasoDeUso(Long idCasoDeUso)
    {
        VOCasoDeUso vo = new VOCasoDeUso();
        try {
           ProjetoFacadeHome projetoHome = (ProjetoFacadeHome)
ServiceLocator.getLocalHome(JNDI.PROJETO_FACADE);
           ProjetoFacade     projeto     = projetoHome.create();
           vo = (VOCasoDeUso) projeto.getById(vo, idCasoDeUso);
        } catch(Exception e) {
            e.printStackTrace();
        }
        return vo;
    }

    public ArrayList getCasosDeUso(Long idProjeto)
    {
        ArrayList al = new ArrayList();
        try {
```

```
            ProjetoFacadeHome projetoHome =
(ProjetoFacadeHome)ServiceLocator.getLocalHome(JNDI.PROJETO_FACADE);
            ProjetoFacade projeto = projetoHome.create();
            VOCasoDeUso vo = new VOCasoDeUso();
            al = (ArrayList)projeto.getAllByRelatedId(vo, idProjeto);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return al;
    }

    public ArrayList getCasosDeUsoRelacionados(Long idRequisito)
    {
        ArrayList al = new ArrayList();
        try {
            ProjetoFacadeHome projetoHome =
(ProjetoFacadeHome)ServiceLocator.getLocalHome(JNDI.PROJETO_FACADE);
            ProjetoFacade projeto = projetoHome.create();
            al =
(ArrayList)projeto.getAllCasosByIdRequisito(idRequisito);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return al;
    }

    public ArrayList getCasosDeUsoNaoRelacionados(Long idProjeto, Long
idRequisito)
    {
        ArrayList al1 = getCasosDeUso(idProjeto);
        ArrayList al2 = getCasosDeUsoRelacionados(idRequisito);
        VOCasoDeUso vo1;
        VOCasoDeUso vo2;

        Iterator it = al2.iterator();

        while ( it.hasNext() )
        {
           vo2 = (VOCasoDeUso) it.next();
           for ( int i = al1.size() ; i > 0 ; i-- )
           {
              vo1 = (VOCasoDeUso) al1.get(i);
              if ( vo1.getNome().compareTo(vo2.getNome()) == 0 )
              {
                  al1.remove(i);
              }
           }
        }

        return al1;
    }

    public VOAtor getAtor(Long idAtor)
    {
        VOAtor vo = new VOAtor();
        try {
           ProjetoFacadeHome projetoHome = (ProjetoFacadeHome)
ServiceLocator.getLocalHome(JNDI.PROJETO_FACADE);
           ProjetoFacade     projeto     = projetoHome.create();
           vo = (VOAtor) projeto.getById(vo, idAtor);
        } catch(Exception e) {
```

```
            e.printStackTrace();
        }
        return vo;
    }

    public ArrayList getQuestoesEmAberto(Long idCasoDeUso)
    {
        ArrayList al = new ArrayList();
        try {
            ProjetoFacadeHome projetoHome =
(ProjetoFacadeHome)ServiceLocator.getLocalHome(JNDI.PROJETO_FACADE);
            ProjetoFacade projeto = projetoHome.create();
            VOQuestaoEmAberto vo = new VOQuestaoEmAberto();
            al = (ArrayList)projeto.getAllByRelatedId(vo, idCasoDeUso);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return al;
    }

    public VORequisitoFuncional getRequisitoFuncional(Long idRequisito)
    {
        VORequisitoFuncional vo = new VORequisitoFuncional();
        try {
           ProjetoFacadeHome projetoHome = (ProjetoFacadeHome)
ServiceLocator.getLocalHome(JNDI.PROJETO_FACADE);
           ProjetoFacade     projeto     = projetoHome.create();
           vo = (VORequisitoFuncional) projeto.getById(vo, idRequisito);
        } catch(Exception e) {
            e.printStackTrace();
        }
        return vo;
    }

    public void editAtor(VOAtor ator)
    {
        try {
            ProjetoFacadeHome projetoHome = (ProjetoFacadeHome)
ServiceLocator.getLocalHome(JNDI.PROJETO_FACADE);
            ProjetoFacade     projeto     = projetoHome.create();
            projeto.edit(ator);
         } catch(Exception e) {
             e.printStackTrace();
         }
    }

    public ArrayList getInteressadosCasoDeUso(Long idCasoDeUso)
    {
        ArrayList al = new ArrayList();
        try {
            ProjetoFacadeHome projetoHome =
(ProjetoFacadeHome)ServiceLocator.getLocalHome(JNDI.PROJETO_FACADE);
            ProjetoFacade projeto = projetoHome.create();
            VOInteressado vo = new VOInteressado();
            al = (ArrayList)projeto.getAllByRelatedId(vo, idCasoDeUso);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return al;
    }
```

```
    public ArrayList getListaCategorias()
    {
        ArrayList al = new ArrayList();
        try {
            CategoriaDeRequisitoNaoFuncionalFacadeHome categoriaHome =
(CategoriaDeRequisitoNaoFuncionalFacadeHome)ServiceLocator.getLocalHome(
JNDI.CATEGORIA_FACADE);
            CategoriaDeRequisitoNaoFuncionalFacade categoria =
categoriaHome.create();
            al = (ArrayList)categoria.findAll();
        } catch (Exception e) {
            e.printStackTrace();
        }
        return al;
    }

    public ArrayList getListaUsuarios()
    {
        ArrayList al = new ArrayList();
        try {
            UsuarioFacadeHome usuarioHome =
(UsuarioFacadeHome)ServiceLocator.getLocalHome(JNDI.USUARIO_FACADE);
            UsuarioFacade usuario = usuarioHome.create();
            al = (ArrayList)usuario.findAll();
        } catch (Exception e) {
            e.printStackTrace();
        }
        return al;
    }

    public VOUsuario getUsuario(String username)
    {
        VOUsuario vo = new VOUsuario();
        try {
            UsuarioFacadeHome usuarioHome =
(UsuarioFacadeHome)ServiceLocator.getLocalHome(JNDI.USUARIO_FACADE);
            UsuarioFacade usuario = usuarioHome.create();
            vo = (VOUsuario)usuario.findByUsername(username);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return vo;
    }

    public ArrayList getVariacoesTecnologicas(Long idCasoDeUso)
    {
        ArrayList al = new ArrayList();
        try {
            ProjetoFacadeHome projetoHome =
(ProjetoFacadeHome)ServiceLocator.getLocalHome(JNDI.PROJETO_FACADE);
            ProjetoFacade projeto = projetoHome.create();
            VOVariacaoTecnologica vo = new VOVariacaoTecnologica();
            al = (ArrayList)projeto.getAllByRelatedId(vo, idCasoDeUso);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return al;
    }

    public void excluaAtor(Long id)
    {
```

```
        try {
            ProjetoFacadeHome projetoHome =
(ProjetoFacadeHome)ServiceLocator.getLocalHome(JNDI.PROJETO_FACADE);
            ProjetoFacade projeto = projetoHome.create();
            VOAtor vo = new VOAtor();
            vo.setId(id);
            projeto.remove(vo);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public void excluaCategoria(Long id)
    {
        try {
            CategoriaDeRequisitoNaoFuncionalFacadeHome categoriaHome =
(CategoriaDeRequisitoNaoFuncionalFacadeHome)ServiceLocator.getLocalHome(
JNDI.CATEGORIA_FACADE);
            CategoriaDeRequisitoNaoFuncionalFacade categoria =
categoriaHome.create();
            categoria.destroy(id);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public void excluaCasoDeUso(Long id)
    {
        try {
            ProjetoFacadeHome projetoHome =
(ProjetoFacadeHome)ServiceLocator.getLocalHome(JNDI.PROJETO_FACADE);
            ProjetoFacade projeto = projetoHome.create();
            VOCasoDeUso vo = new VOCasoDeUso();
            vo.setId(id);
            projeto.remove(vo);
        } catch (Exception e) {
            e.printStackTrace();
        }
   }

    public void excluaProjeto(Long id)
    {
        try {
            ProjetoFacadeHome projetoHome =
(ProjetoFacadeHome)ServiceLocator.getLocalHome(JNDI.PROJETO_FACADE);
            ProjetoFacade projeto = projetoHome.create();
            projeto.destroy(id);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public void excluaRequisitoFuncional(Long id)
    {
        try {
            ProjetoFacadeHome projetoHome =
(ProjetoFacadeHome)ServiceLocator.getLocalHome(JNDI.PROJETO_FACADE);
            ProjetoFacade projeto = projetoHome.create();
            VORequisitoFuncional vo = new VORequisitoFuncional();
            vo.setId(id);
            projeto.remove(vo);
```

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

    public void excluaRequisitoNaoFuncional(Long id)
    {
        try {
            ProjetoFacadeHome projetoHome =
(ProjetoFacadeHome)ServiceLocator.getLocalHome(JNDI.PROJETO_FACADE);
            ProjetoFacade projeto = projetoHome.create();
            VORequisitoNaoFuncional vo = new VORequisitoNaoFuncional();
            vo.setId(id);
            projeto.remove(vo);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public void excluaUsuario(Long id)
    {
        try {
            UsuarioFacadeHome usuarioHome =
(UsuarioFacadeHome)ServiceLocator.getLocalHome(JNDI.USUARIO_FACADE);
            UsuarioFacade usuario = usuarioHome.create();
            usuario.destroy(id);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public VOCategoria getCategoria(Long idCategoria)
    {
        VOCategoria vo = new VOCategoria();
        try {
            CategoriaDeRequisitoNaoFuncionalFacadeHome categoriaHome =
(CategoriaDeRequisitoNaoFuncionalFacadeHome)
ServiceLocator.getLocalHome(JNDI.CATEGORIA_FACADE);
            CategoriaDeRequisitoNaoFuncionalFacade     categoria     =
categoriaHome.create();
            vo = categoria.findById(idCategoria);
        } catch(Exception e) {
            e.printStackTrace();
        }
        return vo;
    }

    public void editCategoria(VOCategoria categoria)
    {
        try {
            CategoriaDeRequisitoNaoFuncionalFacadeHome categoriaHome   =
(CategoriaDeRequisitoNaoFuncionalFacadeHome)
ServiceLocator.getLocalHome(JNDI.CATEGORIA_FACADE);
            CategoriaDeRequisitoNaoFuncionalFacade     categoriaFacade =
categoriaHome.create();
            categoriaFacade.setData(categoria);
         } catch(Exception e) {
             e.printStackTrace();
         }
    }
```

```
    public void editProjeto(VOProjeto projeto)
    {
        try {
            ProjetoFacadeHome projetoHome   = (ProjetoFacadeHome)
ServiceLocator.getLocalHome(JNDI.PROJETO_FACADE);
            ProjetoFacade     projetoFacade = projetoHome.create();
            projetoFacade.setData(projeto);
         } catch(Exception e) {
             e.printStackTrace();
         }
    }

    public void editRequisitoFuncional(VORequisitoFuncional
requisitoFuncional)
    {
        try {
            ProjetoFacadeHome projetoHome   = (ProjetoFacadeHome)
ServiceLocator.getLocalHome(JNDI.PROJETO_FACADE);
            ProjetoFacade     projetoFacade = projetoHome.create();
            projetoFacade.edit(requisitoFuncional);
         } catch(Exception e) {
             e.printStackTrace();
         }
    }

    public void editRequisitoNaoFuncional(VORequisitoNaoFuncional rnf)
    {
        try {
            ProjetoFacadeHome projetoHome   = (ProjetoFacadeHome)
ServiceLocator.getLocalHome(JNDI.PROJETO_FACADE);
            ProjetoFacade     projetoFacade = projetoHome.create();
            projetoFacade.edit(rnf);
         } catch(Exception e) {
             e.printStackTrace();
         }
    }

    public void editUsuario(VOUsuario usuario)
    {
        try {
            UsuarioBusinessFacadeHome usuarioHome   =
(UsuarioBusinessFacadeHome)
ServiceLocator.getLocalHome(JNDI.BUSINESS_FACADE_USUARIO);
            UsuarioBusinessFacade     usuarioFacade =
usuarioHome.create();
            usuarioFacade.editUsuario(usuario);
         } catch(Exception e) {
             e.printStackTrace();
         }
    }

    public void addAtor(VOAtor ator)
    {
        try {
            ProjetoFacadeHome projetoHome   = (ProjetoFacadeHome)
ServiceLocator.getLocalHome(JNDI.PROJETO_FACADE);
            ProjetoFacade     projetoFacade = projetoHome.create();
            projetoFacade.add(ator);
         } catch(Exception e) {
             e.printStackTrace();
         }
```

```
    }

    public void addCategoria(VOCategoria categoria)
    {
        try {
            CategoriaDeRequisitoNaoFuncionalFacadeHome categoriaHome   =
(CategoriaDeRequisitoNaoFuncionalFacadeHome)
ServiceLocator.getLocalHome(JNDI.CATEGORIA_FACADE);
            CategoriaDeRequisitoNaoFuncionalFacade     categoriaFacade =
categoriaHome.create();
            categoriaFacade.createCategoria(categoria);
         } catch(Exception e) {
             e.printStackTrace();
         }
    }

    public VOProjeto addProjeto(Long idUsuario, VOProjeto projeto)
    {
        VOProjeto vo = new VOProjeto();
        try {
            ProjetoFacadeHome projetoHome   = (ProjetoFacadeHome)
ServiceLocator.getLocalHome(JNDI.PROJETO_FACADE);
            ProjetoFacade     projetoFacade = projetoHome.create();
            vo = projetoFacade.createProjeto(projeto);

            VOEscritor vo2 = new VOEscritor();
            vo2.setIdUsuario(idUsuario);
            vo2.setIdProjeto(vo.getId());
            projetoFacade.add(vo2);
         } catch(Exception e) {
             e.printStackTrace();
         }
         return vo;
    }

    public VORequisitoFuncional
addRequisitoFuncional(VORequisitoFuncional requisitoFuncional)
    {
        VORequisitoFuncional vo = null;
        try {
            ProjetoFacadeHome projetoHome   = (ProjetoFacadeHome)
ServiceLocator.getLocalHome(JNDI.PROJETO_FACADE);
            ProjetoFacade     projetoFacade = projetoHome.create();
            vo =
(VORequisitoFuncional)projetoFacade.add(requisitoFuncional);
         } catch(Exception e) {
             e.printStackTrace();
         }
         return vo;
    }

    public void addRequisitoNaoFuncional(VORequisitoNaoFuncional rnf)
    {
        try {
            ProjetoFacadeHome projetoHome   = (ProjetoFacadeHome)
ServiceLocator.getLocalHome(JNDI.PROJETO_FACADE);
            ProjetoFacade     projetoFacade = projetoHome.create();
            projetoFacade.add(rnf);
         } catch(Exception e) {
             e.printStackTrace();
         }
```

```
    }

    public void addUsuario(VOUsuario usuario)
    {
        try {
            UsuarioBusinessFacadeHome usuarioHome   =
(UsuarioBusinessFacadeHome)
ServiceLocator.getLocalHome(JNDI.BUSINESS_FACADE_USUARIO);
            UsuarioBusinessFacade     usuarioFacade =
usuarioHome.create();
            usuarioFacade.createUsuario(usuario);
         } catch(Exception e) {
             e.printStackTrace();
         }
    }

    public ArrayList getProjetosAsLeitor(Long idUsuario)
    {
        ArrayList al = new ArrayList();
        try {
            UsuarioBusinessFacadeHome usuarioHome   =
(UsuarioBusinessFacadeHome)
ServiceLocator.getLocalHome(JNDI.BUSINESS_FACADE_USUARIO);
            UsuarioBusinessFacade     usuarioFacade =
usuarioHome.create();
            usuarioFacade.getProjetosAsLeitor(idUsuario);
         } catch(Exception e) {
             e.printStackTrace();
         }
         return al;
    }

    public ArrayList getProjetosAsEscritor(Long idUsuario)
    {
        ArrayList al = new ArrayList();
        try {
            UsuarioBusinessFacadeHome usuarioHome   =
(UsuarioBusinessFacadeHome)
ServiceLocator.getLocalHome(JNDI.BUSINESS_FACADE_USUARIO);
            UsuarioBusinessFacade     usuarioFacade =
usuarioHome.create();
            al =
(ArrayList)usuarioFacade.getProjetosAsEscritor(idUsuario);
            Iterator it = al.iterator();
            VOProjeto vo;
            while (it.hasNext()) {
                vo = (VOProjeto)it.next();
            }
         } catch(Exception e) {
             e.printStackTrace();
         }
         return al;
    }
}
```

AbraAtorAction.java

```
package org.hopi.web.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.hopi.web.delegate.WebDelegate;

/**
 *
 * Repassa a solicitação de busca das informações da entidade ator.
 *
 * @since 07/11/2004
 * @author Marcel Horner e Rafael Pires
 */
public class AbraAtorAction extends Action {

      public ActionForward execute(ActionMapping map, ActionForm form,
                  HttpServletRequest req, HttpServletResponse resp)
throws Exception
      {
          WebDelegate delegate = new WebDelegate();
            Long idAtor = new Long(req.getParameter("idAtor"));

            HttpSession sessao = req.getSession();

            sessao.setAttribute("ator", delegate.getAtor(idAtor));

            return map.findForward("ok");
      }
}
```

AbraCasoDeUsoAction.java

```
package org.hopi.web.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.hopi.ejb.entity.VOCasoDeUso;
import org.hopi.web.delegate.WebDelegate;

/**
 *
 * Repassa a solicitação de busca das informações da entidade caso de
uso.
 *
 * @since 23/11/2004
 * @author Marcel Horner e Rafael Pires
 */
public class AbraCasoDeUsoAction extends Action {

      public ActionForward execute(ActionMapping map, ActionForm form,
                  HttpServletRequest req, HttpServletResponse resp)
throws Exception
      {
          WebDelegate delegate = new WebDelegate();
            HttpSession sessao = req.getSession();
            Long idCasoDeUso = new Long(-1);

            if (req.getParameter("idCasoDeUso") != null)
            {
                idCasoDeUso = new Long(req.getParameter("idCasoDeUso"));
            }
            else if (sessao.getAttribute("casoDeUso") != null)
            {
                idCasoDeUso =
((VOCasoDeUso)sessao.getAttribute("casoDeUso")).getId();
            }

            VOCasoDeUso casoDeUso = delegate.getCasoDeUso(idCasoDeUso);

            sessao.setAttribute("casoDeUso", casoDeUso);
            sessao.setAttribute("atoresCasoDeUso",
delegate.getAtoresCasoDeUso(idCasoDeUso));
            sessao.setAttribute("atoresNaoRelacionados",
delegate.getAtoresNaoRelacionados(casoDeUso.getIdProjeto(),casoDeUso.get
Id()));
            sessao.setAttribute("interessadosCasoDeUso",
delegate.getInteressadosCasoDeUso(idCasoDeUso));
            sessao.setAttribute("preCondicoes",
delegate.getPreCondicoes(idCasoDeUso));
            sessao.setAttribute("posCondicoes",
delegate.getPosCondicoes(idCasoDeUso));
            sessao.setAttribute("variacoesTecnologicas",
delegate.getVariacoesTecnologicas(idCasoDeUso));
            sessao.setAttribute("questoesEmAberto",
delegate.getQuestoesEmAberto(idCasoDeUso));
```

```
            sessao.setAttribute("requisitosFuncionaisRelacionados",
delegate.getRequisitosFuncionaisRelacionados(idCasoDeUso));
            sessao.setAttribute("requisitosFuncionaisNaoRelacionados",
delegate.getRequisitosFuncionaisNaoRelacionados(casoDeUso.getIdProjeto()
,idCasoDeUso));

            return map.findForward("ok");
      }
}
```

AbraCategoriaAction.java

```
package org.hopi.web.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.hopi.web.delegate.WebDelegate;

/**
 *
 * Repassa a solicitação de busca das informações da entidade categoria.
 *
 * @author Rafael Andreis Pires (pires@inf.ufsc.br) e Marcel Horner
(mhorner@inf.ufsc.br)
 * @since 08/11/2004
 *
 */
public class AbraCategoriaAction extends Action {

      public ActionForward execute(ActionMapping map, ActionForm form,
                  HttpServletRequest req, HttpServletResponse resp)
throws Exception
      {
          WebDelegate delegate = new WebDelegate();
            Long idCategoria = new
Long(req.getParameter("idCategoria"));

            HttpSession sessao = req.getSession();

            sessao.setAttribute("categoria",
delegate.getCategoria(idCategoria));

            return map.findForward("ok");
      }
}
```

AbraProjetoAction.java

```
package org.hopi.web.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.hopi.ejb.entity.VOProjeto;
import org.hopi.web.delegate.WebDelegate;

/**
 *
 * Repassa a solicitação de busca das informações da entidade projeto.
 *
 * @since 06/11/2004
 * @author Marcel Horner e Rafael Pires
 */
public class AbraProjetoAction extends Action {

      public ActionForward execute(ActionMapping map, ActionForm form,
                  HttpServletRequest req, HttpServletResponse resp)
throws Exception
      {
          WebDelegate delegate = new WebDelegate();
            HttpSession sessao = req.getSession();
            Long idProjeto = new Long(-1);

            if (req.getParameter("idProjeto") != null)
            {
                  idProjeto = new Long(req.getParameter("idProjeto"));
            }
            else if (sessao.getAttribute("projeto") != null)
            {
                idProjeto =
((VOProjeto)sessao.getAttribute("projeto")).getId();
            }

          VOProjeto projeto = delegate.getProjeto(idProjeto);
            sessao.setAttribute("projeto", projeto);

            sessao.setAttribute("atores",
delegate.getAtores(idProjeto));
            sessao.setAttribute("requisitosFuncionais",
delegate.getRequisitosFuncionais(idProjeto));
            sessao.setAttribute("requisitosNaoFuncionais",
delegate.getRequisitosNaoFuncionais(idProjeto));
            sessao.setAttribute("casosDeUso",
delegate.getCasosDeUso(idProjeto));

            return map.findForward("ok");
      }
}
```

AbraRequisitoFuncionalAction;

```
package org.hopi.web.struts.action;
```

```
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.hopi.ejb.entity.VORequisitoFuncional;
import org.hopi.web.delegate.WebDelegate;

/**
 * Repassa a solicitação de busca das informações da entidade requisito
funcional.
 *
 * @since 21/11/2004
 * @author Marcel Horner e Rafael Pires
 */
public class AbraRequisitoFuncionalAction extends Action {

      public ActionForward execute(ActionMapping map, ActionForm form,
                  HttpServletRequest req, HttpServletResponse resp)
throws Exception
      {
          WebDelegate delegate = new WebDelegate();
            HttpSession sessao = req.getSession();
            Long idRequisitoFuncional = new Long(-1);

            if (req.getParameter("idRequisitoFuncional") != null)
            {
                idRequisitoFuncional = new
Long(req.getParameter("idRequisitoFuncional"));
            }
            else if (sessao.getAttribute("requisitoFuncional") != null)
            {
                idRequisitoFuncional =
((VORequisitoFuncional)sessao.getAttribute("requisitoFuncional")).getId(
);
            }

            VORequisitoFuncional rf =
delegate.getRequisitoFuncional(idRequisitoFuncional);

            sessao.setAttribute("requisitoFuncional", rf);
            sessao.setAttribute("casosDeUso",
delegate.getCasosDeUsoRelacionados(rf.getId()));
            sessao.setAttribute("casosDeUsoNaoRelacionados",
delegate.getCasosDeUsoNaoRelacionados(rf.getIdProjeto(), rf.getId()));
            sessao.setAttribute("requisitosNaoFuncionais",
delegate.getRequisitosNaoFuncionaisAssociados(rf.getId()));

            return map.findForward("ok");
      }
}
```

AbraRequisitoNaoFuncional.java

```
package org.hopi.web.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.hopi.ejb.entity.VORequisitoNaoFuncional;
import org.hopi.web.delegate.WebDelegate;

/**
 *
 * Repassa a solicitação de busca das informações da entidade requisito
não-funcional.
 *
 * @since 30/11/2004
 * @author Marcel Horner e Rafael Pires
 */
public class AbraRequisitoNaoFuncional extends Action {

      public ActionForward execute(ActionMapping map, ActionForm form,
                  HttpServletRequest req, HttpServletResponse resp)
throws Exception
      {
          WebDelegate delegate = new WebDelegate();
            HttpSession sessao = req.getSession();
            Long idRequisitoNaoFuncional = new Long(-1);

            if (req.getParameter("idRequisitoNaoFuncional") != null)
            {
                idRequisitoNaoFuncional = new
Long(req.getParameter("idRequisitoNaoFuncional"));
            }
            else if (sessao.getAttribute("requisitoNaoFuncional") !=
null)
            {
                idRequisitoNaoFuncional =
((VORequisitoNaoFuncional)sessao.getAttribute("requisitoNaoFuncional")).
getId();
            }

            VORequisitoNaoFuncional rnf =
delegate.getRequisitoNaoFuncional(idRequisitoNaoFuncional);

            sessao.setAttribute("requisitoNaoFuncional", rnf);
            sessao.setAttribute("categorias",
delegate.getListaCategorias());

            return map.findForward("ok");
      }
}
```

AbraRequisitoNaoFuncionalAssociado.java

```
package org.hopi.web.struts.action;
```

```
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.hopi.ejb.entity.VORequisitoNaoFuncional;
import org.hopi.web.delegate.WebDelegate;

/**
 *
 * Repassa a solicitação de busca das informações da entidade requisito
não-funcional associado ao requisito funcional.
 *
 * @since 21/11/2004
 * @author Marcel Horner e Rafael Pires
 */
public class AbraRequisitoNaoFuncionalAssociadoAction extends Action {

     public ActionForward execute(ActionMapping map, ActionForm form,
                HttpServletRequest req, HttpServletResponse resp)
throws Exception
     {
         WebDelegate delegate = new WebDelegate();
           HttpSession sessao = req.getSession();
           Long idRequisitoNaoFuncional = new Long(-1);

           if (req.getParameter("idRequisitoNaoFuncional") != null)
           {
               idRequisitoNaoFuncional = new
Long(req.getParameter("idRequisitoNaoFuncional"));
           }
           else if (sessao.getAttribute("requisitoNaoFuncional") !=
null)
           {
               idRequisitoNaoFuncional =
((VORequisitoNaoFuncional)sessao.getAttribute("requisitoNaoFuncional")).
getId();
           }

           VORequisitoNaoFuncional rnf =
delegate.getRequisitoNaoFuncional(idRequisitoNaoFuncional);

           sessao.setAttribute("requisitoNaoFuncional", rnf);
           sessao.setAttribute("categorias",
delegate.getListaCategorias());

           return map.findForward("ok");
     }
}
```

AbraUsuarioAction.java

```
package org.hopi.web.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
```

```
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.hopi.web.delegate.WebDelegate;

/**
 *
 * Repassa a solicitação de busca das informações da entidade usuário.
 *
 * @author Rafael Andreis Pires (pires@inf.ufsc.br) e Marcel Horner
(mhorner@inf.ufsc.br)
 * @since 20/11/2004
 *
 */
public class AbraUsuarioAction extends Action {

      public ActionForward execute(ActionMapping map, ActionForm form,
                  HttpServletRequest req, HttpServletResponse resp)
throws Exception
      {
          WebDelegate delegate = new WebDelegate();
            String username = new String(req.getParameter("idUsuario"));

            HttpSession sessao = req.getSession();

            sessao.setAttribute("usuario",
delegate.getUsuario(username));

            return map.findForward("ok");
      }
}
```

AcessoNaoAutorizadoAction.java

```
package org.hopi.web.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

/**
 *
 * Encaminha a apresentação da página de erro de acesso ao usuário.
 *
 * @since 30/10/2004
 * @author Marcel Horner e Rafael Pires
 */
public class AcessoNaoAutorizadoAction extends Action {

     public ActionForward execute(ActionMapping map, ActionForm form,
                  HttpServletRequest req, HttpServletResponse resp)
throws Exception
     {
          req.setAttribute("erro", "<font class='erro'>Acesso não
autorizado!<br>Para acessar a página solicitada é necessário realizar o
login no sistema.</font><br>");
          return map.findForward("ok");
     }
}
```

ExcluaAtorAction.java

```
package org.hopi.web.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.hopi.web.delegate.WebDelegate;

/**
 *
 * Repassa a solicitação de remoção de uma entidade do tipo ator.
 *
 * @author Rafael Andreis Pires (pires@inf.ufsc.br) e Marcel Horner
(mhorner@inf.ufsc.br)
 * @since 07/11/2004
 */
public class ExcluaAtorAction extends Action {

      public ActionForward execute(ActionMapping map, ActionForm form,
                  HttpServletRequest req, HttpServletResponse resp)
throws Exception
      {
          WebDelegate delegate = new WebDelegate();
            Long idAtor = new Long(req.getParameter("idAtor"));

            delegate.excluaAtor(idAtor);

            return map.findForward("ok");
      }
}
```

ExcluaCasoDeUsoAction.java

```
package org.hopi.web.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.hopi.web.delegate.WebDelegate;

/**
 *
 * Repassa a solicitação de remoção de uma entidade do tipo caso de uso.
 *
 * @author Rafael Andreis Pires (pires@inf.ufsc.br) e Marcel Horner
(mhorner@inf.ufsc.br)
 * @since 21/11/2004
 */
public class ExcluaCasoDeUsoAction extends Action {

      public ActionForward execute(ActionMapping map, ActionForm form,
                  HttpServletRequest req, HttpServletResponse resp)
throws Exception
      {
          WebDelegate delegate = new WebDelegate();
            Long idCasoDeUso = new
Long(req.getParameter("idCasoDeUso"));

            delegate.excluaCasoDeUso(idCasoDeUso);

            return map.findForward("ok");
      }
}
```

ExcluaCategoriaAction.java

```
package org.hopi.web.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.hopi.web.delegate.WebDelegate;

/**
 *
 * Repassa a solicitação de remoção de uma entidade do tipo categorias.
 *
 * @author Rafael Andreis Pires (pires@inf.ufsc.br) e Marcel Horner
(mhorner@inf.ufsc.br)
 * @since 07/11/2004
 */
public class ExcluaCategoriaAction extends Action {

     public ActionForward execute(ActionMapping map, ActionForm form,
                 HttpServletRequest req, HttpServletResponse resp)
throws Exception
     {
         WebDelegate delegate = new WebDelegate();
           Long idCategoria = new
Long(req.getParameter("idCategoria"));

           delegate.excluaCategoria(idCategoria);

           return map.findForward("ok");
     }
}
```

ExcluaProjetoAction.java

```
package org.hopi.web.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.hopi.web.delegate.WebDelegate;

/**
 *
 * Repassa a solicitação de remoção de uma entidade do tipo projeto.
 *
 * @since 07/11/2004
 * @author Marcel Horner e Rafael Pires
 */
public class ExcluaProjetoAction extends Action {

      public ActionForward execute(ActionMapping map, ActionForm form,
                  HttpServletRequest req, HttpServletResponse resp)
throws Exception
      {
          WebDelegate delegate = new WebDelegate();
            Long idProjeto = new Long(req.getParameter("idProjeto"));
            delegate.excluaProjeto(idProjeto);

            return map.findForward("ok");
      }
}
```

ExcluaRequisitoFuncionalAction.java

```
package org.hopi.web.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.hopi.web.delegate.WebDelegate;

/**
 *
 * Repassa a solicitação de remoção de uma entidade do tipo requisito
funcional.
 *
 * @author Rafael Andreis Pires (pires@inf.ufsc.br) e Marcel Horner
(mhorner@inf.ufsc.br)
 * @since 21/11/2004
 */
public class ExcluaRequisitoFuncionalAction extends Action {

      public ActionForward execute(ActionMapping map, ActionForm form,
                  HttpServletRequest req, HttpServletResponse resp)
throws Exception
      {
          WebDelegate delegate = new WebDelegate();
            Long idRequisito = new
Long(req.getParameter("idRequisitoFuncional"));

            delegate.excluaRequisitoFuncional(idRequisito);

            return map.findForward("ok");
      }
}
```

ExcluaRequisitoNaoFuncionalAction.java

```
package org.hopi.web.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.hopi.web.delegate.WebDelegate;

/**
 *
 * Repassa a solicitação de remoção de uma entidade do tipo requisito
não-funcional.
 *
 * @author Rafael Andreis Pires (pires@inf.ufsc.br) e Marcel Horner
(mhorner@inf.ufsc.br)
 * @since 30/11/2004
 */
public class ExcluaRequisitoNaoFuncionalAction extends Action {

      public ActionForward execute(ActionMapping map, ActionForm form,
                  HttpServletRequest req, HttpServletResponse resp)
throws Exception
      {
          WebDelegate delegate = new WebDelegate();
            Long idRequisitoNaoFuncional = new
Long(req.getParameter("idRequisitoNaoFuncional"));


      delegate.excluaRequisitoNaoFuncional(idRequisitoNaoFuncional);

            return map.findForward("ok");
      }
}
```

ExcluaRequisitoNaoFuncionalAssociado.java

```
package org.hopi.web.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.hopi.web.delegate.WebDelegate;

/**
 *
 * Repassa a solicitação de remoção de uma entidade do tipo requisito
não-funcional associado a requisito funcional.
 *
 * @author Rafael Andreis Pires (pires@inf.ufsc.br) e Marcel Horner
(mhorner@inf.ufsc.br)
 * @since 22/11/2004
 */
public class ExcluaRequisitoNaoFuncionalAssociadoAction extends Action {

      public ActionForward execute(ActionMapping map, ActionForm form,
                  HttpServletRequest req, HttpServletResponse resp)
throws Exception
      {
          WebDelegate delegate = new WebDelegate();
            Long idRequisitoNaoFuncionalAssociado = new
Long(req.getParameter("idRequisitoNaoFuncionalAssociado"));


      delegate.excluaRequisitoNaoFuncional(idRequisitoNaoFuncionalAssoci
ado);

            return map.findForward("ok");
      }
}
```

ExcluaUsuarioAction.java

```
package org.hopi.web.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.hopi.web.delegate.WebDelegate;

/**
 *
 * Repassa a solicitação de remoção de uma entidade do tipo usuário.
 *
 * @author Rafael Andreis Pires (pires@inf.ufsc.br) e Marcel Horner
(mhorner@inf.ufsc.br)
 * @since 07/11/2004
 */
public class ExcluaUsuarioAction extends Action {

      public ActionForward execute(ActionMapping map, ActionForm form,
                  HttpServletRequest req, HttpServletResponse resp)
throws Exception
      {
          WebDelegate delegate = new WebDelegate();
            Long idUsuario = new Long(req.getParameter("idUsuario"));

            delegate.excluaUsuario(idUsuario);

            return map.findForward("ok");
      }
}
```

ListeCategoriasAction.java

```
package org.hopi.web.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.hopi.web.delegate.WebDelegate;

/**
 *
 * Repassa a recuperação da lista de entidades do tipo categoria.
 *
 * @since 07/11/2004
 * @author Marcel Horner e Rafael Pires
 */
public class ListeCategoriasAction extends Action {

      public ActionForward execute(ActionMapping map, ActionForm form,
                  HttpServletRequest req, HttpServletResponse resp)
throws Exception
      {
          WebDelegate delegate = new WebDelegate();
            HttpSession sessao = req.getSession();
            sessao.setAttribute("categorias",
delegate.getListaCategorias());

            return map.findForward("ok");
      }
}
```

ListeUsuariosAction.java

```
package org.hopi.web.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.hopi.web.delegate.WebDelegate;

/**
 *
 * Repassa a recuperação da lista de entidades do tipo usuário.
 *
 * @since 07/11/2004
 * @author Marcel Horner e Rafael Pires
 */
public class ListeUsuariosAction extends Action {

     public ActionForward execute(ActionMapping map, ActionForm form,
                 HttpServletRequest req, HttpServletResponse resp)
throws Exception
     {
         WebDelegate delegate = new WebDelegate();
           HttpSession sessao = req.getSession();
           sessao.setAttribute("usuarios",
delegate.getListaUsuarios());

           return map.findForward("ok");
     }
}
```

LoginAction.java

```
package org.hopi.web.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.hopi.DTO.DTOSessao;
import org.hopi.ejb.entity.VOUsuario;
import org.hopi.exception.LoginInvalidoException;
import org.hopi.web.delegate.WebDelegate;
import org.hopi.web.struts.form.LoginForm;

/**
* Verifica se o nome do usuário e senha informados pelo estão corretos.
 *
 * @since 30/10/2004
 * @author Marcel Horner e Rafael Pires
 */
public class LoginAction extends Action {

      public ActionForward execute(ActionMapping map, ActionForm form,
                  HttpServletRequest req, HttpServletResponse resp)
throws Exception
      {
          WebDelegate delegate = new WebDelegate();

            LoginForm login = (LoginForm) form;
            DTOSessao meuDTO = new DTOSessao();
          VOUsuario vo = new VOUsuario();

            try {
                vo = delegate.validarLogin(login.getUsername(),
login.getPassword());
            } catch (LoginInvalidoException lie) {
                req.setAttribute("erro", "<font class='erro'>Os valores
informados para o username e password não conferem.</font><br>");
                  return map.findForward("erro");
            } catch (Exception e) {
                req.setAttribute("erro", "<font class='erro'>Ocorreu um
erro interno do sistema. Tente novamente mais tarde.</font><br>");
                  return map.findForward("erro");
            }

          meuDTO.setNomeCompleto(vo.getNome());
          meuDTO.setUsername(vo.getUsername());
          meuDTO.setId(vo.getId());
            HttpSession sessao = req.getSession();
            sessao.setAttribute("dadosSessao", meuDTO);

            return map.findForward("ok");
      }
}
```

LogoutAction.java

```
package org.hopi.web.struts.action;
```

```
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

/**
 *
 * Limpa as informações na sessão do usuário.
 *
 * @since 30/10/2004
 * @author Marcel Horner e Rafael Pires
 */
public class LogoutAction extends Action {

     public ActionForward execute(ActionMapping map, ActionForm form,
                 HttpServletRequest req, HttpServletResponse resp)
throws Exception
     {
           HttpSession sessao = req.getSession();
           sessao.invalidate();

         return map.findForward("logout");
     }
}
```

NovaCategoriaAction.java

```
package org.hopi.web.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.hopi.ejb.entity.VOCategoria;

/**
 *
 * Prepara o formulário para a inclusão da nova categoria.
 *
 * @author Rafael Andreis Pires (pires@inf.ufsc.br) e Marcel Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 */
public class NovaCategoriaAction extends Action {

      public ActionForward execute(ActionMapping map, ActionForm form,
                  HttpServletRequest req, HttpServletResponse resp)
throws Exception
      {
            HttpSession sessao = req.getSession();
            VOCategoria categoria = new VOCategoria();
            categoria.setId(new Long("-1"));
            sessao.setAttribute("categoria", categoria);

          return map.findForward("ok");
      }
}
```

NovoAtorAction.java

```
package org.hopi.web.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.hopi.ejb.entity.VOAtor;
import org.hopi.ejb.entity.VOProjeto;

/**
 *
 * Prepara o formulário para a inclusão do novo ator.
 *
 * @since 07/11/2004
 * @author Marcel Horner e Rafael Pires
 */
public class NovoAtorAction extends Action {

      public ActionForward execute(ActionMapping map, ActionForm form,
                  HttpServletRequest req, HttpServletResponse resp)
throws Exception
      {
            HttpSession sessao = req.getSession();
            VOAtor ator = new VOAtor();
            Long idProjeto = null;

            if(sessao.getAttribute("projeto") != null)
            {
                  VOProjeto projeto =
(VOProjeto)sessao.getAttribute("projeto");
                  idProjeto = projeto.getId();
            }

            ator.setId(new Long(-1));
            ator.setIdProjeto(idProjeto);
            sessao.setAttribute("ator", ator);

          return map.findForward("ok");
      }
}
```

NovoProjetoAction.java

```
package org.hopi.web.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.hopi.ejb.entity.VOProjeto;

/**
 *
 * Prepara o formulário para a inclusão do novo projeto.
 *
 * @since 21/11/2004
 * @author Marcel Horner e Rafael Pires
 */
public class NovoProjetoAction extends Action {

      public ActionForward execute(ActionMapping map, ActionForm form,
                  HttpServletRequest req, HttpServletResponse resp)
throws Exception
      {
            HttpSession sessao = req.getSession();
            VOProjeto projeto = new VOProjeto();

            projeto.setId(new Long(-1));
            sessao.setAttribute("projeto", projeto);

          return map.findForward("ok");
      }
}
```

NovoRequisitoFuncionalAction.java

```
package org.hopi.web.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.hopi.ejb.entity.VORequisitoFuncional;
import org.hopi.ejb.entity.VOProjeto;

/**
 *
 * Prepara o formulário para a inclusão do novo requisito funcional.
 *
 * @author Rafael Andreis Pires (pires@inf.ufsc.br) e Marcel Horner
(mhorner@inf.ufsc.br)
 * @since 23/11/2004
 */
public class NovoRequisitoFuncionalAction extends Action {

      public ActionForward execute(ActionMapping map, ActionForm form,
                  HttpServletRequest req, HttpServletResponse resp)
throws Exception
      {
            HttpSession sessao = req.getSession();
            VORequisitoFuncional requisitoFuncional = new
VORequisitoFuncional();
            Long idProjeto = null;

            if(sessao.getAttribute("projeto") != null)
            {
                  VOProjeto projeto =
(VOProjeto)sessao.getAttribute("projeto");
                  idProjeto = projeto.getId();
            }

            requisitoFuncional.setId(new Long(-1));
            requisitoFuncional.setIdProjeto(idProjeto);
            sessao.setAttribute("requisitoFuncional",
requisitoFuncional);

          return map.findForward("ok");
      }
}
```

NovoRequisitoNaoFuncionalAction.java

```
package org.hopi.web.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.hopi.ejb.entity.VORequisitoFuncional;
import org.hopi.ejb.entity.VORequisitoNaoFuncional;
import org.hopi.web.delegate.WebDelegate;

/**
 *
 * Prepara o formulário para a inclusão do novo requisito não-funcional.
 *
 * @author Rafael Andreis Pires (pires@inf.ufsc.br) e Marcel Horner
(mhorner@inf.ufsc.br)
 * @since 30/11/2004
 */
public class NovoRequisitoNaoFuncionalAction extends Action {

      public ActionForward execute(ActionMapping map, ActionForm form,
                  HttpServletRequest req, HttpServletResponse resp)
throws Exception
      {
          //TODO Corrigir a criação do novo requisito não funcional,
associando ao
          //projeto
          WebDelegate delegate = new WebDelegate();
            HttpSession sessao = req.getSession();
            VORequisitoNaoFuncional rnf = new VORequisitoNaoFuncional();
            Long idRequisitoFuncional = null;

            if(sessao.getAttribute("requisitoFuncional") != null)
            {
                  VORequisitoFuncional rf =
(VORequisitoFuncional)sessao.getAttribute("requisitoFuncional");
                  idRequisitoFuncional = rf.getId();
            }

            rnf.setId(new Long(-1));
            rnf.setIdRequisitoFuncional(idRequisitoFuncional);
            sessao.setAttribute("requisitoNaoFuncional", rnf);
            sessao.setAttribute("categorias",
delegate.getListaCategorias());

          return map.findForward("ok");
      }
}
```

NovoRequisitoNaoFuncionalAssociadoAction.java

```
package org.hopi.web.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.hopi.ejb.entity.VORequisitoFuncional;
import org.hopi.ejb.entity.VORequisitoNaoFuncional;
import org.hopi.web.delegate.WebDelegate;

/**
 *
 * Prepara o formulário para a inclusão do novo requisito não-funcioal
associado ao requisito funcional.
 *
 * @author Rafael Andreis Pires (pires@inf.ufsc.br) e Marcel Horner
(mhorner@inf.ufsc.br)
 * @since 23/11/2004
 */
public class NovoRequisitoNaoFuncionalAssociadoAction extends Action {

      public ActionForward execute(ActionMapping map, ActionForm form,
                  HttpServletRequest req, HttpServletResponse resp)
throws Exception
      {
          WebDelegate delegate = new WebDelegate();
            HttpSession sessao = req.getSession();
            VORequisitoNaoFuncional rnf = new VORequisitoNaoFuncional();
            Long idRequisitoFuncional = null;

            if(sessao.getAttribute("requisitoFuncional") != null)
            {
                  VORequisitoFuncional rf =
(VORequisitoFuncional)sessao.getAttribute("requisitoFuncional");
                  idRequisitoFuncional = rf.getId();
            }

            rnf.setId(new Long(-1));
            rnf.setIdRequisitoFuncional(idRequisitoFuncional);
            sessao.setAttribute("requisitoNaoFuncional", rnf);
            sessao.setAttribute("categorias",
delegate.getListaCategorias());

          return map.findForward("ok");
      }
}
```

NovoUsuarioAction.java

```
package org.hopi.web.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.hopi.ejb.entity.VOUsuario;

/**
 *
 * Prepara o formulário para a inclusão do novo usuário.
 *
 * @author Rafael Andreis Pires (pires@inf.ufsc.br) e Marcel Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 */
public class NovoUsuarioAction extends Action {

      public ActionForward execute(ActionMapping map, ActionForm form,
                  HttpServletRequest req, HttpServletResponse resp)
throws Exception
      {
          VOUsuario usuario = new VOUsuario();
          usuario.setId(new Long(-1));

            HttpSession sessao = req.getSession();
            sessao.setAttribute("usuario", usuario);

          return map.findForward("ok");
      }
}
```

PrincipalAction.java

```
package org.hopi.web.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.hopi.DTO.DTOSessao;
import org.hopi.web.delegate.WebDelegate;

/**
 *
 * Recupera a lista de projetos onde o usuáro é escritor e / ou leitor.
 *
 * @since 30/10/2004
 * @author Marcel Horner e Rafael Pires
 */
public class PrincipalAction extends Action {

      public ActionForward execute(ActionMapping map, ActionForm form,
                  HttpServletRequest req, HttpServletResponse resp)
throws Exception
      {
          WebDelegate delegate = new WebDelegate();
            HttpSession sessao = req.getSession();

            DTOSessao meuDTO =
(DTOSessao)sessao.getAttribute("dadosSessao");


            sessao.setAttribute("projetosEdicao",
delegate.getProjetosAsEscritor(meuDTO.getId()));
            sessao.setAttribute("projetosVisualizacao",
delegate.getProjetosAsLeitor(meuDTO.getId()));

          return map.findForward("ok");
      }
}
```

SalveAtorAction.java

```
package org.hopi.web.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.hopi.ejb.entity.VOAtor;
import org.hopi.web.delegate.WebDelegate;
import org.hopi.web.struts.form.AtorForm;

/**
 *
 * Repassa a solicitaçõa da inclusão da entidade do tipo Ator.
 *
 * @since 07/11/2004
 * @author Marcel Horner e Rafael Pires
 */
public class SalveAtorAction extends Action {

      public ActionForward execute(ActionMapping map, ActionForm form,
                  HttpServletRequest req, HttpServletResponse resp)
throws Exception
      {

            AtorForm formAtor = (AtorForm) form;

          WebDelegate delegate = new WebDelegate();
          VOAtor ator = new VOAtor();
          ator.setId(new Long(formAtor.getId()));
          ator.setIdProjeto(new Long(formAtor.getIdProjeto()));
          ator.setNome(formAtor.getNome());
          ator.setDescricao(formAtor.getDescricao());

          if ( ator.getId().longValue() == -1 ) {
              delegate.addAtor(ator);
          } else {
              delegate.editAtor(ator);
          }

          return map.findForward("ok");
      }
}
```

SalveCategoriaAction.java

```
package org.hopi.web.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.hopi.ejb.entity.VOCategoria;
import org.hopi.web.delegate.WebDelegate;
import org.hopi.web.struts.form.CategoriaForm;

/**
 *
 * Repassa a solicitaçõa da inclusão da entidade do tipo categoria.
 *
 * @author Rafael Andreis Pires (pires@inf.ufsc.br) e Marcel Horner
(mhorner@inf.ufsc.br)
 * @since 08/11/2004
 */
public class SalveCategoriaAction extends Action {

      public ActionForward execute(ActionMapping map, ActionForm form,
                  HttpServletRequest req, HttpServletResponse resp)
throws Exception
      {
          CategoriaForm formCategoria = (CategoriaForm) form;
          HttpSession sessao = req.getSession();
          VOCategoria categoria = new VOCategoria();

          WebDelegate delegate = new WebDelegate();
          categoria.setId(new Long(formCategoria.getId()));
          categoria.setNome(formCategoria.getNome());
          categoria.setDescricao(formCategoria.getDescricao());

          if ( categoria.getId().longValue() == -1 ) {
              delegate.addCategoria(categoria);
          } else {
              delegate.editCategoria(categoria);
          }

          return map.findForward("ok");
      }
}
```

SalveProjetoAction.java

```
package org.hopi.web.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.hopi.DTO.DTOSessao;
import org.hopi.ejb.entity.VOProjeto;
import org.hopi.web.delegate.WebDelegate;
import org.hopi.web.struts.form.ProjetoForm;

/**
 *
 * Repassa a solicitaçõa da inclusão da entidade do tipo projeto.
 *
 * @author Rafael Andreis Pires (pires@inf.ufsc.br) e Marcel Horner
(mhorner@inf.ufsc.br)
 * @since 21/11/2004
 */
public class SalveProjetoAction extends Action {

      public ActionForward execute(ActionMapping map, ActionForm form,
                  HttpServletRequest req, HttpServletResponse resp)
throws Exception
      {
          ProjetoForm formProjeto = (ProjetoForm) form;
          HttpSession sessao = req.getSession();
          VOProjeto projeto = new VOProjeto();

          WebDelegate delegate = new WebDelegate();
          projeto.setId(new Long(formProjeto.getId()));
          projeto.setNome(formProjeto.getNome());

projeto.setSumarioExecutivo(formProjeto.getSumarioExecutivo());
          projeto.setDataInicial(new
Long(formProjeto.getDataInicial()));

          DTOSessao dados =
(DTOSessao)sessao.getAttribute("dadosSessao");

          if ( projeto.getId().longValue() == -1 ) {
                sessao.setAttribute("projeto",
delegate.addProjeto(dados.getId(), projeto));
                return map.findForward("novo");
          } else {
              delegate.editProjeto(projeto);
                return map.findForward("existente");
          }
      }
}
```

SalveRequisitoFuncionalAction.java

```
package org.hopi.web.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.hopi.ejb.entity.VORequisitoFuncional;
import org.hopi.web.delegate.WebDelegate;
import org.hopi.web.struts.form.RequisitoFuncionalForm;

/**
 *
 * Repassa a solicitaçõa da inclusão da entidade do tipo requisito
funcional.
 *
 * @author Rafael Andreis Pires (pires@inf.ufsc.br) e Marcel Horner
(mhorner@inf.ufsc.br)
 * @since 22/11/2004
 */
public class SalveRequisitoFuncionalAction extends Action {

      public ActionForward execute(ActionMapping map, ActionForm form,
                  HttpServletRequest req, HttpServletResponse resp)
throws Exception
      {
          RequisitoFuncionalForm formRequisitoFuncional =
(RequisitoFuncionalForm) form;
          HttpSession sessao = req.getSession();
          VORequisitoFuncional requisitoFuncional = new
VORequisitoFuncional();

          WebDelegate delegate = new WebDelegate();
          requisitoFuncional.setId(new
Long(formRequisitoFuncional.getId()));
          requisitoFuncional.setNome(formRequisitoFuncional.getNome());

requisitoFuncional.setDescricao(formRequisitoFuncional.getDescricao());
          requisitoFuncional.setIdProjeto(new
Long(formRequisitoFuncional.getIdProjeto()));
          requisitoFuncional.setOculto(new Boolean(true));

          if ( requisitoFuncional.getId().longValue() == -1 ) {
              sessao.setAttribute("requisitoFuncional",
delegate.addRequisitoFuncional(requisitoFuncional));
                return map.findForward("novo");
          } else {
              delegate.editRequisitoFuncional(requisitoFuncional);
                return map.findForward("existente");
          }
      }
}
```

SalveRequisitoNaoFuncionalAction.java

```
package org.hopi.web.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.hopi.ejb.entity.VORequisitoNaoFuncional;
import org.hopi.web.delegate.WebDelegate;
import org.hopi.web.struts.form.RequisitoNaoFuncionalForm;

/**
 *
 * Repassa a solicitaçõa da inclusão da entidade do tipo requisito não
funcional.
 *
 * @author Rafael Andreis Pires (pires@inf.ufsc.br) e Marcel Horner
(mhorner@inf.ufsc.br)
 * @since 30/11/2004
 */
public class SalveRequisitoNaoFuncionalAction extends Action {

      public ActionForward execute(ActionMapping map, ActionForm form,
                  HttpServletRequest req, HttpServletResponse resp)
throws Exception
      {
          RequisitoNaoFuncionalForm formRequisitoNaoFuncional =
(RequisitoNaoFuncionalForm) form;
          HttpSession sessao = req.getSession();
          VORequisitoNaoFuncional rnf = new VORequisitoNaoFuncional();

          WebDelegate delegate = new WebDelegate();
          rnf.setId(new Long(formRequisitoNaoFuncional.getId()));
          rnf.setNome(formRequisitoNaoFuncional.getNome());
          rnf.setDescricao(formRequisitoNaoFuncional.getDescricao());
          rnf.setObrigatorio(new Boolean(true));
          rnf.setPermanente(new Boolean(false));

          if ( rnf.getId().longValue() == -1 ) {
              delegate.addRequisitoNaoFuncional(rnf);
          } else {
              delegate.editRequisitoNaoFuncional(rnf);
          }
          return map.findForward("ok");
      }
}
```

SalveRequisitoNaoFuncionalAssociadoAction.java

```
package org.hopi.web.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.hopi.ejb.entity.VORequisitoNaoFuncional;
import org.hopi.web.delegate.WebDelegate;
import org.hopi.web.struts.form.RequisitoNaoFuncionalForm;

/**
 *
 * Repassa a solicitaçõa da inclusão da entidade do tipo requisito não-
funcional associado.
 *
 * @author Rafael Andreis Pires (pires@inf.ufsc.br) e Marcel Horner
(mhorner@inf.ufsc.br)
 * @since 23/11/2004
 */
public class SalveRequisitoNaoFuncionalAssociadoAction extends Action {

      public ActionForward execute(ActionMapping map, ActionForm form,
                  HttpServletRequest req, HttpServletResponse resp)
throws Exception
      {
          RequisitoNaoFuncionalForm formRequisitoNaoFuncional =
(RequisitoNaoFuncionalForm) form;
          HttpSession sessao = req.getSession();
          VORequisitoNaoFuncional rnf = new VORequisitoNaoFuncional();

          WebDelegate delegate = new WebDelegate();
          rnf.setId(new Long(formRequisitoNaoFuncional.getId()));
          rnf.setNome(formRequisitoNaoFuncional.getNome());
          rnf.setDescricao(formRequisitoNaoFuncional.getDescricao());
          rnf.setObrigatorio(new Boolean(true));
          rnf.setPermanente(new Boolean(false));
          rnf.setIdRequisitoFuncional(new
Long(formRequisitoNaoFuncional.getIdRequisitoFuncional()));
          rnf.setIdCategoria(new
Long(formRequisitoNaoFuncional.getIdCategoria()));

          if ( rnf.getId().longValue() == -1 ) {
              delegate.addRequisitoNaoFuncional(rnf);
          } else {
              delegate.editRequisitoNaoFuncional(rnf);
          }
          return map.findForward("ok");
      }
}
```

SalveUsuarioAction.java

```
package org.hopi.web.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.hopi.ejb.entity.VOUsuario;
import org.hopi.web.delegate.WebDelegate;
import org.hopi.web.struts.form.UsuarioForm;

/**
* Repassa a solicitaçõa da inclusão da entidade do tipo usuário.
 *
 * @author Rafael Andreis Pires (pires@inf.ufsc.br) e Marcel Horner
(mhorner@inf.ufsc.br)
 * @since 08/11/2004
 */
public class SalveUsuarioAction extends Action {

      public ActionForward execute(ActionMapping map, ActionForm form,
                  HttpServletRequest req, HttpServletResponse resp)
throws Exception
      {
          UsuarioForm formUsuario = (UsuarioForm) form;
          HttpSession sessao = req.getSession();
          VOUsuario usuario = new VOUsuario();

          try {
                WebDelegate delegate = new WebDelegate();
                usuario.setId(new Long(formUsuario.getId()));
                usuario.setNome(formUsuario.getNome());
                usuario.setUsername(formUsuario.getUsername());
                usuario.setPassword(formUsuario.getPassword());
                usuario.setMatricula(formUsuario.getMatricula());
                usuario.setTelefone(formUsuario.getTelefone());
                usuario.setEmail(formUsuario.getEmail());

usuario.setAdministrador(formUsuario.getAdministrador());

                if ( usuario.getId().longValue() == -1 ) {
                    delegate.addUsuario(usuario);
                } else {
                    delegate.editUsuario(usuario);
                }
          } catch (Exception e) {
              e.printStackTrace();
              throw new Exception(e);
          }

          return map.findForward("ok");
      }
}
```

PACKAGE ORG.HOPI.WEB.STRUTS.FORM

AtorForm.java

```
package org.hopi.web.struts.form;

import javax.servlet.http.HttpServletRequest;
import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;

/**
* Formulário para validação dos campos da entidade ator.
* @since 07/11/2004
 * @author Marcel Horner e Rafael Pires
 */
public class AtorForm extends ActionForm {
      String id;
      String idProjeto;
      String nome;
      String descricao;

      public ActionErrors validate(ActionMapping map, HttpServletRequest
req) {
            ActionErrors erros = new ActionErrors();
            boolean camposOK = true;
            try {
                  if (this.nome.trim().compareTo("") == 0) {
                        erros.add("nome", new
ActionMessage("erro.campo.nome.branco"));
                        camposOK = false;
                  }
            } catch (NullPointerException e) {
                erros.add("nome", new
ActionMessage("erro.campo.nome.branco"));
                  camposOK = false;
            }
            if (camposOK) {
                try {
                    if (this.id.trim().compareTo("") == 0) {
                        erros.add("id", new
ActionMessage("erro.campo.id.branco"));
                                camposOK = false;
                    }
                } catch (NullPointerException e) {
                    erros.add("id", new
ActionMessage("erro.campo.id.branco"));
                        camposOK = false;
                }
            }
            if (camposOK) {
                  try {
                        if (this.idProjeto.trim().compareTo("") == 0) {
                              erros.add("idProjeto", new
ActionMessage("erro.campo.idProjeto.branco"));
                        }
                  } catch (NullPointerException e) {
                        erros.add("idProjeto", new
ActionMessage("erro.campo.idProjeto.branco"));
                  }
```

```
            }
            return erros;
        }
    /**
     * @return Returns the descricao.
     */
    public String getDescricao() {
        return descricao;
    }
    /**
     * @param descricao The descricao to set.
     */
    public void setDescricao(String descricao) {
        this.descricao = descricao;
    }
    /**
     * @return Returns the id.
     */
    public String getId() {
        return id;
    }
    /**
     * @param id The id to set.
     */
    public void setId(String id) {
        this.id = id;
    }
    /**
     * @return Returns the idProjeto.
     */
    public String getIdProjeto() {
        return idProjeto;
    }
    /**
     * @param idProjeto The idProjeto to set.
     */
    public void setIdProjeto(String idProjeto) {
        this.idProjeto = idProjeto;
    }
    /**
     * @return Returns the nome.
     */
    public String getNome() {
        return nome;
    }
    /**
     * @param nome The nome to set.
     */
    public void setNome(String nome) {
        this.nome = nome;
    }
}
```

CasoDeUsoForm.java

```
package org.hopi.web.struts.form;

import javax.servlet.http.HttpServletRequest;

import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;

/**
 *
 * Formulário para validação dos campos da entidade caso de uso.
 *
 * @author Rafael Andreis Pires (pires@inf.ufsc.br) e Marcel Horner
(mhorner@inf.ufsc.br)
 * @since 23/11/2004
 */
public class CasoDeUsoForm extends ActionForm {

    private Long id;
    private Long idProjeto;
    private String referencia;
    private String nome;
    private String finalidade;
    private String visaoGeral;

     public ActionErrors validate(ActionMapping map, HttpServletRequest
req) {
           ActionErrors erros = new ActionErrors();
           boolean camposOK = true;

           try {
                 if (this.nome.trim().compareTo("") == 0) {
                       erros.add("nome", new
ActionMessage("erro.campo.nome.branco"));
                       camposOK = false;
                 }
           } catch (NullPointerException e) {
                 erros.add("nome", new
ActionMessage("erro.campo.nome.branco"));
                 camposOK = false;
           }

           return erros;
     }

}
```

CategoriaForm.java

```
package org.hopi.web.struts.form;

import javax.servlet.http.HttpServletRequest;

import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;

/**
 *
 * Formulário para validação dos campos da entidade categoria.
 *
 * @author Rafael Andreis Pires (pires@inf.ufsc.br) e Marcel Horner
(mhorner@inf.ufsc.br)
 * @since 08/11/2004
 */
public class CategoriaForm extends ActionForm {

    protected String id;
    protected String nome;
    protected String descricao;

     public ActionErrors validate(ActionMapping map, HttpServletRequest
req) {
            ActionErrors erros = new ActionErrors();
            boolean camposOK = true;

            try {
                  if (this.nome.trim().compareTo("") == 0) {
                        erros.add("nome", new
ActionMessage("erro.campo.nome.branco"));
                        camposOK = false;
                  }
            } catch (NullPointerException e) {
                  erros.add("nome", new
ActionMessage("erro.campo.nome.branco"));
                  camposOK = false;
            }

            return erros;
     }

    /**
     * @return Returns the descricao.
     */
    public String getDescricao() {
        return descricao;
    }
    /**
     * @param descricao The descricao to set.
     */
    public void setDescricao(String descricao) {
        this.descricao = descricao;
    }
    /**
     * @return Returns the id.
     */
    public String getId() {
```

```
        return id;
    }
    /**
     * @param id The id to set.
     */
    public void setId(String id) {
        this.id = id;
    }
    /**
     * @return Returns the nome.
     */
    public String getNome() {
        return nome;
    }
    /**
     * @param nome The nome to set.
     */
    public void setNome(String nome) {
        this.nome = nome;
    }
}
```

LoginForm.java

```
package org.hopi.web.struts.form;

import javax.servlet.http.HttpServletRequest;

import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;

/**
 *
 * Formulário para validação do login no sistema.
 *
 * @since 30/10/2004
 * @author Marcel Horner e Rafael Pires
 */
public class LoginForm extends ActionForm {

      String username;
      String password;

      public ActionErrors validate(ActionMapping map, HttpServletRequest
req) {
            ActionErrors erros = new ActionErrors();

            try {
                  if (this.username.trim().compareTo("") == 0) {
                        erros.add("errors", new
ActionMessage("erro.login.username.branco"));
                  }
            } catch (NullPointerException e) {
                  erros.add("errors", new
ActionMessage("erro.login.username.branco"));
            }

            try {
                  if (this.password.trim().compareTo("") == 0) {
                        erros.add("password", new
ActionMessage("erro.login.password.branco"));
                  }
            } catch (NullPointerException e) {
                  erros.add("password", new
ActionMessage("erro.login.password.branco"));
            }

            return erros;
      }
      /**
       * @return Returns the password.
       */
      public String getPassword() {
            return password;
      }
      /**
       * @param password The password to set.
       */
      public void setPassword(String password) {
            this.password = password;
      }
```

```
	/**
	 * @return Returns the username.
	 */
	public String getUsername() {
		return username;
	}
	/**
	 * @param username The username to set.
	 */
	public void setUsername(String username) {
		this.username = username;
	}
}
```

ProjetoForm.java

```
package org.hopi.web.struts.form;

import javax.servlet.http.HttpServletRequest;
import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;

/**
* Formulário para validação dos campos da entidade projeto.
 *
 * @author Rafael Andreis Pires (pires@inf.ufsc.br) e Marcel Horner
(mhorner@inf.ufsc.br)
 * @since 21/11/2004
 */
public class ProjetoForm extends ActionForm {
    protected String id;
    protected String nome;
    protected String dataInicial;
    protected String dataFinal;
    protected String sumarioExecutivo;

     public ActionErrors validate(ActionMapping map, HttpServletRequest
req) {
           ActionErrors erros = new ActionErrors();
           boolean camposOK = true;
           try {
                 if (this.nome.trim().compareTo("") == 0) {
                       erros.add("nome", new
ActionMessage("erro.campo.nome.branco"));
                       camposOK = false;
                 }
           } catch (NullPointerException e) {
                 erros.add("nome", new
ActionMessage("erro.campo.nome.branco"));
                 camposOK = false;
           }
           try {
                 if (this.dataInicial.trim().compareTo("") == 0) {
                       erros.add("nome", new
ActionMessage("erro.campo.dataInicial.branco"));
                       camposOK = false;
                 }
           } catch (NullPointerException e) {
                 erros.add("nome", new
ActionMessage("erro.campo.dataInicial.branco"));
                 camposOK = false;
           }
           return erros;
     }

    /**
     * @return Returns the dataFinal.
     */
    public String getDataFinal() {
        return dataFinal;
    }
    /**
     * @param dataFinal The dataFinal to set.
```

```
     */
    public void setDataFinal(String dataFinal) {
        this.dataFinal = dataFinal;
    }
    /**
     * @return Returns the dataInicial.
     */
    public String getDataInicial() {
        return dataInicial;
    }
    /**
     * @param dataInicial The dataInicial to set.
     */
    public void setDataInicial(String dataInicial) {
        this.dataInicial = dataInicial;
    }
    /**
     * @return Returns the id.
     */
    public String getId() {
        return id;
    }
    /**
     * @param id The id to set.
     */
    public void setId(String id) {
        this.id = id;
    }
    /**
     * @return Returns the nome.
     */
    public String getNome() {
        return nome;
    }
    /**
     * @param nome The nome to set.
     */
    public void setNome(String nome) {
        this.nome = nome;
    }
    /**
     * @return Returns the sumarioExecutivo.
     */
    public String getSumarioExecutivo() {
        return sumarioExecutivo;
    }
    /**
     * @param sumarioExecutivo The sumarioExecutivo to set.
     */
    public void setSumarioExecutivo(String sumarioExecutivo) {
        this.sumarioExecutivo = sumarioExecutivo;
    }
}
```

RequisitoFuncionalForm.java

```
package org.hopi.web.struts.form;

import javax.servlet.http.HttpServletRequest;

import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;

/**
 *
 * Formulário para validação dos campos da entidade requisito funcional.
 *
 * @author Rafael Andreis Pires (pires@inf.ufsc.br) e Marcel Horner
(mhorner@inf.ufsc.br)
 * @since 21/11/2004
 */
public class RequisitoFuncionalForm extends ActionForm {

    protected String id;
    protected String idProjeto;
    protected String nome;
    protected String descricao;
    protected Boolean oculto;

     public ActionErrors validate(ActionMapping map, HttpServletRequest
req) {
            ActionErrors erros = new ActionErrors();
            boolean camposOK = true;

            try {
                  if (this.nome.trim().compareTo("") == 0) {
                        erros.add("nome", new
ActionMessage("erro.campo.nome.branco"));
                        camposOK = false;
                  }
            } catch (NullPointerException e) {
                  erros.add("nome", new
ActionMessage("erro.campo.nome.branco"));
                  camposOK = false;
            }

            return erros;
     }

    /**
     * @return Returns the descricao.
     */
    public String getDescricao() {
        return descricao;
    }
    /**
     * @param descricao The descricao to set.
     */
    public void setDescricao(String descricao) {
        this.descricao = descricao;
    }
    /**
     * @return Returns the id.
```

```
     */
    public String getId() {
        return id;
    }
    /**
     * @param id The id to set.
     */
    public void setId(String id) {
        this.id = id;
    }
    /**
     * @return Returns the idProjeto.
     */
    public String getIdProjeto() {
        return idProjeto;
    }
    /**
     * @param idProjeto The idProjeto to set.
     */
    public void setIdProjeto(String idProjeto) {
        this.idProjeto = idProjeto;
    }
    /**
     * @return Returns the nome.
     */
    public String getNome() {
        return nome;
    }
    /**
     * @param nome The nome to set.
     */
    public void setNome(String nome) {
        this.nome = nome;
    }
    /**
     * @return Returns the oculto.
     */
    public Boolean getOculto() {
        return oculto;
    }
    /**
     * @param oculto The oculto to set.
     */
    public void setOculto(Boolean oculto) {
        this.oculto = oculto;
    }
}
```

RequisitoNaoFuncionalForm.java

```
package org.hopi.web.struts.form;

import javax.servlet.http.HttpServletRequest;

import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;

/**
 *
 * Formulário para validação dos campos da entidade requisito não-
funcional.
 *
 * @author Rafael Andreis Pires (pires@inf.ufsc.br) e Marcel Horner
(mhorner@inf.ufsc.br)
 * @since 22/11/2004
 */
public class RequisitoNaoFuncionalForm extends ActionForm {

    protected String id;
    protected String idRequisitoFuncional;
    protected String idCategoria;
    protected String nome;
    protected String descricao;
    protected Boolean obrigatorio;
    protected Boolean permanente;

     public ActionErrors validate(ActionMapping map, HttpServletRequest
req) {
            ActionErrors erros = new ActionErrors();
            boolean camposOK = true;

            try {
                  if (this.nome.trim().compareTo("") == 0) {
                        erros.add("nome", new
ActionMessage("erro.campo.nome.branco"));
                        camposOK = false;
                  }
            } catch (NullPointerException e) {
                  erros.add("nome", new
ActionMessage("erro.campo.nome.branco"));
                  camposOK = false;
            }

            try {
                  if (this.idCategoria.trim().compareTo("") == 0) {
                        erros.add("idCategoria", new
ActionMessage("erro.campo.categoria.branco"));
                        camposOK = false;
                  }
            } catch (NullPointerException e) {
                  erros.add("idCategoria", new
ActionMessage("erro.campo.categoria.branco"));
                  camposOK = false;
            }

            return erros;
     }
```

```
/**
 * @return Returns the descricao.
 */
public String getDescricao() {
    return descricao;
}
/**
 * @param descricao The descricao to set.
 */
public void setDescricao(String descricao) {
    this.descricao = descricao;
}
/**
 * @return Returns the id.
 */
public String getId() {
    return id;
}
/**
 * @param id The id to set.
 */
public void setId(String id) {
    this.id = id;
}
/**
 * @return Returns the idCategoria.
 */
public String getIdCategoria() {
    return idCategoria;
}
/**
 * @param idCategoria The idCategoria to set.
 */
public void setIdCategoria(String idCategoria) {
    this.idCategoria = idCategoria;
}
/**
 * @return Returns the idRequisitoFuncional.
 */
public String getIdRequisitoFuncional() {
    return idRequisitoFuncional;
}
/**
 * @param idRequisitoFuncional The idRequisitoFuncional to set.
 */
public void setIdRequisitoFuncional(String idRequisitoFuncional) {
    this.idRequisitoFuncional = idRequisitoFuncional;
}
/**
 * @return Returns the nome.
 */
public String getNome() {
    return nome;
}
/**
 * @param nome The nome to set.
 */
public void setNome(String nome) {
    this.nome = nome;
}
```

```
    /**
     * @return Returns the obrigatorio.
     */
    public Boolean getObrigatorio() {
        return obrigatorio;
    }
    /**
     * @param obrigatorio The obrigatorio to set.
     */
    public void setObrigatorio(Boolean obrigatorio) {
        this.obrigatorio = obrigatorio;
    }
    /**
     * @return Returns the permanente.
     */
    public Boolean getPermanente() {
        return permanente;
    }
    /**
     * @param permanente The permanente to set.
     */
    public void setPermanente(Boolean permanente) {
        this.permanente = permanente;
    }
}
```

UsuarioForm.java

```
package org.hopi.web.struts.form;

import javax.servlet.http.HttpServletRequest;

import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;

/**
 *
 * Formulário para validação dos campos da entidade usuário.
 *
 * @author Rafael Andreis Pires (pires@inf.ufsc.br) e Marcel Horner
(mhorner@inf.ufsc.br)
 * @since 20/11/2004
 */
public class UsuarioForm extends ActionForm {

    protected String id;
    protected String nome;
    protected String username;
    protected String password;
    protected String confirmaPassword;
    protected String matricula;
    protected Boolean administrador;
    protected String telefone;
    protected String email;

      public ActionErrors validate(ActionMapping map, HttpServletRequest
req) {
            ActionErrors erros = new ActionErrors();
            boolean camposOK = true;
            boolean passOK = true;

            try
            {
                  if (this.nome.trim().compareTo("") == 0)
                  {
                        erros.add("nome", new
ActionMessage("erro.campo.nome.branco"));
                        camposOK = false;
                  }
            } catch (NullPointerException e) {
                  erros.add("nome", new
ActionMessage("erro.campo.nome.branco"));
                  camposOK = false;
            }

            try
            {
                  if (this.username.trim().compareTo("") == 0)
                  {
                        erros.add("username", new
ActionMessage("erro.campo.username.branco"));
                        camposOK = false;
                  }
            } catch (NullPointerException e) {
```

```
                erros.add("username", new
ActionMessage("erro.campo.username.branco"));
                camposOK = false;
          }

          try
          {
                if (this.password.trim().compareTo("") == 0)
                {
                      erros.add("password", new
ActionMessage("erro.campo.password.branco"));
                      camposOK = false;
                      passOK = false;
                }
          } catch (NullPointerException e) {
                erros.add("password", new
ActionMessage("erro.campo.password.branco"));
                camposOK = false;
                passOK = false;
          }

          try
          {
                if (this.confirmaPassword.trim().compareTo("") == 0)
                {
                      erros.add("confirmaPassword", new
ActionMessage("erro.campo.confirmaPassword.branco"));
                      camposOK = false;
                      passOK = false;
                }
          } catch (NullPointerException e) {
                erros.add("confirmaPassword", new
ActionMessage("erro.campo.confirmaPassword.branco"));
                camposOK = false;
                passOK = false;
          }

          try
          {
              if (this.telefone.trim().compareTo("") == 0)
                {
                      erros.add("telefone", new
ActionMessage("erro.campo.telefone.branco"));
                      camposOK = false;
                }
          } catch (NullPointerException e) {
                erros.add("telefone", new
ActionMessage("erro.campo.telefone.branco"));
                camposOK = false;
          }

          try
          {
              if (this.email.trim().compareTo("") == 0)
                {
                      erros.add("email", new
ActionMessage("erro.campo.email.branco"));
                      camposOK = false;
                }
          } catch (NullPointerException e) {
```

```
                erros.add("email", new
ActionMessage("erro.campo.email.branco"));
                camposOK = false;
            }

            if (passOK)
            {
                if
(this.password.trim().compareTo(this.confirmaPassword.trim()) != 0)
                {
                    erros.add("confirmaPassword", new
ActionMessage("erro.campo.confirmaPassword.diferente"));
                    camposOK = false;
                }
            }

            //TODO Não faz muito sentido validar o campo ID.
            /*
            if (camposOK)
            {
                try
                {
                    if (this.id.trim().compareTo("") == 0)
                    {
                        erros.add("id", new
ActionMessage("erro.campo.id.branco"));
                            camposOK = false;
                    }
                } catch (NullPointerException e) {
                    erros.add("id", new
ActionMessage("erro.campo.id.branco"));
                    camposOK = false;
                }
            }
            */

            return erros;
     }

    /**
     * @return Returns the administrador.
     */
    public Boolean getAdministrador() {
        return administrador;
    }
    /**
     * @param administrador The administrador to set.
     */
    public void setAdministrador(Boolean administrador) {
        this.administrador = administrador;
    }
    /**
     * @return Returns the confirmaPassword.
     */
    public String getConfirmaPassword() {
        return confirmaPassword;
    }
    /**
     * @param confirmaPassword The confirmaPassword to set.
     */
    public void setConfirmaPassword(String confirmaPassword) {
```

```
    this.confirmaPassword = confirmaPassword;
}
/**
 * @return Returns the email.
 */
public String getEmail() {
    return email;
}
/**
 * @param email The email to set.
 */
public void setEmail(String email) {
    this.email = email;
}
/**
 * @return Returns the id.
 */
public String getId() {
    return id;
}
/**
 * @param id The id to set.
 */
public void setId(String id) {
    this.id = id;
}
/**
 * @return Returns the matricula.
 */
public String getMatricula() {
    return matricula;
}
/**
 * @param matricula The matricula to set.
 */
public void setMatricula(String matricula) {
    this.matricula = matricula;
}
/**
 * @return Returns the nome.
 */
public String getNome() {
    return nome;
}
/**
 * @param nome The nome to set.
 */
public void setNome(String nome) {
    this.nome = nome;
}
/**
 * @return Returns the password.
 */
public String getPassword() {
    return password;
}
/**
 * @param password The password to set.
 */
public void setPassword(String password) {
    this.password = password;
```

```
    }
    /**
     * @return Returns the telefone.
     */
    public String getTelefone() {
        return telefone;
    }
    /**
     * @param telefone The telefone to set.
     */
    public void setTelefone(String telefone) {
        this.telefone = telefone;
    }
    /**
     * @return Returns the username.
     */
    public String getUsername() {
        return username;
    }
    /**
     * @param username The username to set.
     */
    public void setUsername(String username) {
        this.username = username;
    }
}
```

UsuarioForm.java

```
package org.hopi.web.struts.form;

import javax.servlet.http.HttpServletRequest;

import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;

/**
 *
 * Formulário para validação dos campos da entidade usuário.
 *
 * @author Rafael Andreis Pires (pires@inf.ufsc.br) e Marcel Horner
(mhorner@inf.ufsc.br)
 * @since 20/11/2004
 */
public class UsuarioForm extends ActionForm {

    protected String id;
    protected String nome;
    protected String username;
    protected String password;
    protected String confirmaPassword;
    protected String matricula;
    protected Boolean administrador;
    protected String telefone;
    protected String email;

      public ActionErrors validate(ActionMapping map, HttpServletRequest
req) {
            ActionErrors erros = new ActionErrors();
            boolean camposOK = true;
            boolean passOK = true;

            try {
                  if (this.nome.trim().compareTo("") == 0) {
                        erros.add("nome", new
ActionMessage("erro.campo.nome.branco"));
                        camposOK = false;
                  }
            } catch (NullPointerException e) {
                  erros.add("nome", new
ActionMessage("erro.campo.nome.branco"));
                  camposOK = false;
            }

            try {
                  if (this.username.trim().compareTo("") == 0) {
                        erros.add("username", new
ActionMessage("erro.campo.username.branco"));
                        camposOK = false;
                  }
            } catch (NullPointerException e) {
                  erros.add("username", new
ActionMessage("erro.campo.username.branco"));
                  camposOK = false;
            }
```

```
            try {
                  if (this.password.trim().compareTo("") == 0) {
                        erros.add("password", new
ActionMessage("erro.campo.password.branco"));
                        camposOK = false;
                        passOK = false;
                  }
            } catch (NullPointerException e) {
                  erros.add("password", new
ActionMessage("erro.campo.password.branco"));
                  camposOK = false;
                  passOK = false;
            }

            try {
                  if (this.confirmaPassword.trim().compareTo("") == 0) {
                        erros.add("confirmaPassword", new
ActionMessage("erro.campo.confirmaPassword.branco"));
                        camposOK = false;
                        passOK = false;
                  }
            } catch (NullPointerException e) {
                  erros.add("confirmaPassword", new
ActionMessage("erro.campo.confirmaPassword.branco"));
                  camposOK = false;
                  passOK = false;
            }

            try {
                if (this.telefone.trim().compareTo("") == 0) {
                        erros.add("telefone", new
ActionMessage("erro.campo.telefone.branco"));
                        camposOK = false;
                  }
            } catch (NullPointerException e) {
                  erros.add("telefone", new
ActionMessage("erro.campo.telefone.branco"));
                  camposOK = false;
            }

            try {
                if (this.email.trim().compareTo("") == 0) {
                        erros.add("email", new
ActionMessage("erro.campo.email.branco"));
                        camposOK = false;
                  }
            } catch (NullPointerException e) {
                  erros.add("email", new
ActionMessage("erro.campo.email.branco"));
                  camposOK = false;
            }

            if (passOK) {
                  if
(this.password.trim().compareTo(this.confirmaPassword.trim()) != 0) {
                        erros.add("confirmaPassword", new
ActionMessage("erro.campo.confirmaPassword.diferente"));
                        camposOK = false;
                  }
            }
```

```
        return erros;
    }

/**
 * @return Returns the administrador.
 */
public Boolean getAdministrador() {
    return administrador;
}
/**
 * @param administrador The administrador to set.
 */
public void setAdministrador(Boolean administrador) {
    this.administrador = administrador;
}
/**
 * @return Returns the confirmaPassword.
 */
public String getConfirmaPassword() {
    return confirmaPassword;
}
/**
 * @param confirmaPassword The confirmaPassword to set.
 */
public void setConfirmaPassword(String confirmaPassword) {
    this.confirmaPassword = confirmaPassword;
}
/**
 * @return Returns the email.
 */
public String getEmail() {
    return email;
}
/**
 * @param email The email to set.
 */
public void setEmail(String email) {
    this.email = email;
}
/**
 * @return Returns the id.
 */
public String getId() {
    return id;
}
/**
 * @param id The id to set.
 */
public void setId(String id) {
    this.id = id;
}
/**
 * @return Returns the matricula.
 */
public String getMatricula() {
    return matricula;
}
/**
 * @param matricula The matricula to set.
 */
public void setMatricula(String matricula) {
```

```
        this.matricula = matricula;
    }
    /**
     * @return Returns the nome.
     */
    public String getNome() {
        return nome;
    }
    /**
     * @param nome The nome to set.
     */
    public void setNome(String nome) {
        this.nome = nome;
    }
    /**
     * @return Returns the password.
     */
    public String getPassword() {
        return password;
    }
    /**
     * @param password The password to set.
     */
    public void setPassword(String password) {
        this.password = password;
    }
    /**
     * @return Returns the telefone.
     */
    public String getTelefone() {
        return telefone;
    }
    /**
     * @param telefone The telefone to set.
     */
    public void setTelefone(String telefone) {
        this.telefone = telefone;
    }
    /**
     * @return Returns the username.
     */
    public String getUsername() {
        return username;
    }
    /**
     * @param username The username to set.
     */
    public void setUsername(String username) {
        this.username = username;
    }
}
```

PACOTE ORG.HOPI.EJB.BASE

EntityBase.java

```
package org.hopi.ejb.base;

import java.rmi.RemoteException;
import javax.ejb.CreateException;
import javax.ejb.EJBException;
import javax.ejb.EntityContext;
import javax.ejb.RemoveException;
import org.hopi.ejb.util.GeradorChaveUtil;

/**
* Super Classe dos EJBs Entitys que gerencia a criação das chaves-
primárias, além de implementar os métodos da especificação EJB.
 *
 * @author Rafael Andreis Pires (pires@inf.ufsc.br) e Marcel Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
*/

public class EntityBase {

   public Long nextID(String entidade) throws CreateException {
       return GeradorChaveUtil.proximoValor(entidade);
   }

   /**
    * ESPECIFICAÇÃO EJB
    */

   public void ejbActivate() throws EJBException, RemoteException {
   }

   public void ejbLoad() throws EJBException, RemoteException {
   }

   public void ejbPassivate() throws EJBException, RemoteException {
   }

   public void ejbRemove() throws RemoveException, EJBException,
           RemoteException {
   }

   public void ejbStore() throws EJBException, RemoteException {
   }

   public void setEntityContext(EntityContext arg0) throws EJBException,
           RemoteException {
   }

   public void unsetEntityContext() throws EJBException, RemoteException
{
   }

}
```

SessionBase.java

```
package org.hopi.ejb.base;

import java.rmi.RemoteException;

import javax.ejb.EJBException;
import javax.ejb.SessionContext;
import javax.naming.NamingException;

import org.hopi.ejb.util.ServiceLocator;

/**
 *
 * Super Classes dos EJBs Sessions que gerencia a utilização do Cache de
JNDI, além de implementar os métodos da especificação EJB.
 *
 * @author Rafael Andreis Pires (pires@inf.ufsc.br) e Marcel Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 */

public class SessionBase {

    SessionContext ctx;

    public Object lookup(String jndi) throws NamingException {
        return ServiceLocator.getLocalHome(jndi);
    }

    public void ejbActivate() throws EJBException, RemoteException {
    }

    public void ejbPassivate() throws EJBException, RemoteException {
    }

    public void ejbRemove() throws EJBException, RemoteException {
    }

    public void setSessionContext(SessionContext arg0)
        throws EJBException, RemoteException {
        ctx = arg0;
    }

}
```

PACOTE ORG.HOPI.EJB.ENTITY

AtorBean.java

```
package org.hopi.ejb.entity;

import javax.ejb.CreateException;
import javax.ejb.EntityBean;
import org.hopi.ejb.base.EntityBase;

/**
* EJB Entity da Entidade de Dados ATOR.
 *
 *  @author  Rafael  Andreis  Pires  (pires@inf.ufsc.br)  e  Marcel  Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 * @ejb.bean
 *    name = "Ator"
 *    local-jndi-name="org/hopi/Ator"
 *    view-type = "local"
 *    type = "CMP"
 *    cmp-version = "2.x"
 *    schema = "ator"
 *    primkey-field = "id"
 *
 * @ejb.value-object
 *
 *       @ejb.finder       signature       =       "java.util.Collection
findByIdProjeto(java.lang.Long idProjeto)"
 *                         query = "select Object(a) from ator a where
a.idProjeto = ?1"
*/

public abstract class AtorBean extends EntityBase implements EntityBean
{
    /**
     * @ejb.create-method
     */
    public Long ejbCreate(VOAtor dados) throws CreateException {
        setId(nextID("Ator"));
        setDados(dados);
        return dados.getId();
    }

    public void ejbPostCreate(VOAtor dados) throws CreateException {
    }

    /**
     * @ejb.interface-method
     */
    public VOAtor getDados() {
        VOAtor dados = new VOAtor();
        dados.setId(getId());
        dados.setNome(getNome());
        dados.setDescricao(getDescricao());
        dados.setIdProjeto(getIdProjeto());
        return dados;
    }

    /**
```

```
     * @ejb.interface-method
     */
    public void setDados(VOAtor dados) throws CreateException {
        setNome(dados.getNome());
        setDescricao(dados.getDescricao());
        setIdProjeto(dados.getIdProjeto());
    }

     /**
      * @ejb.persistence column-name = "ID"
      * @ejb.interface-method
      */
    public abstract Long getId();

     /**
      * @ejb.interface-method
      */
    public abstract void setId(Long id);

     /**
      * @ejb.persistence column-name = "NOME"
      * @ejb.interface-method
      */
    public abstract String getNome();

     /**
      * @ejb.interface-method
      */
    public abstract void setNome(String nome);

     /**
      * @ejb.persistence column-name = "DESCRICAO"
      * @ejb.interface-method
      */
    public abstract String getDescricao();

     /**
      * @ejb.interface-method
      */
    public abstract void setDescricao(String descricao);

     /**
      * @ejb.persistence column-name = "ID_PROJETO"
      * @ejb.interface-method
      */
      public abstract Long getIdProjeto();

      /**
       * @ejb.value-object
       */
      public abstract void setIdProjeto(Long idProjeto);

}
```

AtorDoCasoBean.java

```
package org.hopi.ejb.entity;

import javax.ejb.CreateException;
import javax.ejb.EntityBean;

import org.hopi.ejb.base.EntityBase;

/**
 *
 * EJB Entity da Entidade de Dados ATOR_DO_CASO.
 *
 *  @author  Rafael  Andreis  Pires  (pires@inf.ufsc.br)  e  Marcel  Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 * @ejb.bean
 *    name = "AtorDoCaso"
 *    local-jndi-name="org/hopi/AtorDoCaso"
 *    view-type = "local"
 *    type = "CMP"
 *    cmp-version = "2.x"
 *    schema = "atorDoCaso"
 *    primkey-field = "id"
 *
 * @ejb.value-object
 *
 *       @ejb.finder       signature       =       "java.util.Collection
findByIdAtor(java.lang.Long idAtor)"
 *                 query = "select Object(adc) from atorDoCaso adc where
adc.idAtor = ?1"
 *
 *       @ejb.finder       signature       =       "java.util.Collection
findByIdCasoDeUso(java.lang.Long idCasoDeUso)"
 *                 query = "select Object(adc) from atorDoCaso adc where
adc.idCaso = ?1"
 *
 */

public  abstract  class  AtorDoCasoBean  extends  EntityBase  implements
EntityBean {

    /**
     * @ejb.create-method
     */
    public Long ejbCreate(VOAtorDoCaso dados) throws CreateException {
        setId(nextID("AtorDoCaso"));
        setDados(dados);
        return dados.getId();
    }

    public void ejbPostCreate(VOAtorDoCaso dados) throws CreateException
{
    }

    /**
     * @ejb.interface-method
     */
    public VOAtorDoCaso getDados() {
        VOAtorDoCaso dados = new VOAtorDoCaso();
```

```
        dados.setId(getId());
        dados.setIdAtor(getIdAtor());
        dados.setIdCaso(getIdCaso());
        return dados;
    }

    /**
     * @ejb.interface-method
     */
    public void setDados(VOAtorDoCaso dados) {
       setIdAtor(dados.getIdAtor());
       setIdCaso(dados.getIdCaso());
    }

     /**
      * @ejb.persistence column-name = "ID"
      * @ejb.interface-method
      */
    public abstract Long getId();

     /**
      * @ejb.interface-method
      */
    public abstract void setId(Long id);

    /**
       * @ejb.persistence column-name = "ID_ATOR"
       * @ejb.interface-method
       */
    public abstract Long getIdAtor();

     /**
      * @ejb.interface-method
      */
    public abstract void setIdAtor(Long idAtor);

     /**
      * @ejb.persistence column-name = "ID_CASO"
      * @ejb.interface-method
      */
    public abstract Long getIdCaso();

     /**
      * @ejb.interface-method
      */
    public abstract void setIdCaso(Long idCaso);

}
```

CasoDeUsoBean.java

```
package org.hopi.ejb.entity;

import javax.ejb.CreateException;
import javax.ejb.EntityBean;

import org.hopi.ejb.base.EntityBase;

/**
 *
 * EJB Entity da Entidade de Dados CASO_DE_USO.
 *
 *  @author  Rafael  Andreis  Pires  (pires@inf.ufsc.br)  e  Marcel  Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 * @ejb.bean
 *    name = "CasoDeUso"
 *    local-jndi-name="org/hopi/CasoDeUso"
 *    view-type = "local"
 *    type = "CMP"
 *    cmp-version = "2.x"
 *    schema = "casoDeUso"
 *    primkey-field = "id"
 *
 * @ejb.value-object
 *
 *       @ejb.finder       signature       =       "java.util.Collection
findByProjeto(java.lang.Long idProjeto)"
 *                       query = "select Object(c) from casoDeUso c where
c.idProjeto = ?1"
 *
 */

public  abstract  class  CasoDeUsoBean  extends  EntityBase  implements
EntityBean {

    /**
     * @ejb.create-method
     */
    public Long ejbCreate(VOCasoDeUso dados) throws CreateException {
        setId(nextID("CasoDeUso"));
        setDados(dados);
        return dados.getId();
    }

    public void ejbPostCreate(VOCasoDeUso dados) throws CreateException
{
    }

    /**
     * @ejb.interface-method
     */
    public VOCasoDeUso getDados() {
        VOCasoDeUso dados = new VOCasoDeUso();
        dados.setId(getId());
        dados.setReferencia(getReferencia());
        dados.setNome(getNome());
        dados.setFinalidade(getFinalidade());
        dados.setVisaoGeral(getVisaoGeral());
```

```
    dados.setIdProjeto(getIdProjeto());
    return dados;
}

/**
 * @ejb.interface-method
 */
public void setDados(VOCasoDeUso dados) {
    setReferencia(dados.getReferencia());
    setNome(dados.getNome());
    setFinalidade(dados.getFinalidade());
    setVisaoGeral(dados.getVisaoGeral());
    setIdProjeto(dados.getIdProjeto());
}

/**
 * @ejb.persistence column-name = "ID"
 * @ejb.interface-method
 */
public abstract Long getId();

/**
 * @ejb.interface-method
 */
public abstract void setId(Long id);

/**
 * @ejb.persistence column-name = "REFERENCIA"
 * @ejb.interface-method
 */
public abstract String getReferencia();

/**
 * @ejb.interface-method
 */
public abstract void setReferencia(String referencia);

/**
 * @ejb.persistence column-name = "NOME"
 * @ejb.interface-method
 */
public abstract String getNome();

/**
 * @ejb.interface-method
 */
public abstract void setNome(String nome);

/**
 * @ejb.persistence column-name = "FINALIDADE"
 * @ejb.interface-method
 */
public abstract String getFinalidade();

/**
 * @ejb.interface-method
 */
public abstract void setFinalidade(String finalidade);

/**
 * @ejb.persistence column-name = "VISAO_GERAL"
```

```
     * @ejb.interface-method
     */
    public abstract String getVisaoGeral();

    /**
     * @ejb.interface-method
     */
    public abstract void setVisaoGeral(String visaoGeral);

    /**
     * @ejb.persistence column-name = "ID_PROJETO"
     * @ejb.interface-method
     */
    public abstract Long getIdProjeto();

    /**
     * @ejb.interface-method
     */
    public abstract void setIdProjeto(Long idProjeto);

}
```

CasoDoRequisitoBean.java

```
package org.hopi.ejb.entity;

import javax.ejb.CreateException;
import javax.ejb.EntityBean;

import org.hopi.ejb.base.EntityBase;

/**
 *
 * EJB Entity da Entidade de Dados CASO_DO_REQUISITO.
 *
 *  @author  Rafael  Andreis  Pires  (pires@inf.ufsc.br)  e  Marcel  Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 * @ejb.bean
 *    name = "CasoDoRequisito"
 *    local-jndi-name="org/hopi/CasoDoRequisito"
 *    view-type = "local"
 *    type = "CMP"
 *    cmp-version = "2.x"
 *    schema = "casoDoRequisito"
 *    primkey-field = "id"
 *
 * @ejb.value-object
 *
 *       @ejb.finder       signature       =       "java.util.Collection
findByIdCasoDeUso(java.lang.Long idCasoDeUso)"
 *                  query = "select Object(cdr) from casoDoRequisito cdr
where cdr.idCaso = ?1"
 *
 *       @ejb.finder       signature       =       "java.util.Collection
findByIdRequisito(java.lang.Long idRequisito)"
 *                  query = "select Object(cdr) from casoDoRequisito cdr
where cdr.idRequisito = ?1"
 *
 */

public abstract class CasoDoRequisitoBean extends EntityBase implements
EntityBean {

    /**
     * @ejb.create-method
     */
    public     Long     ejbCreate(VOCasoDoRequisito     dados)     throws
CreateException {
        setId(nextID("CasoDoRequisito"));
        setDados(dados);
        return dados.getId();
    }

    public    void    ejbPostCreate(VOCasoDoRequisito    dados)    throws
CreateException {
    }

    /**
     * @ejb.interface-method
     */
    public VOCasoDoRequisito getDados() {
```

```
        VOCasoDoRequisito dados = new VOCasoDoRequisito();
        dados.setId(getId());
        dados.setIdCaso(getIdCaso());
        dados.setIdRequisito(getIdRequisito());
        return dados;
    }

    /**
     * @ejb.interface-method
     */
    public void setDados(VOCasoDoRequisito dados) {
        setIdCaso(dados.getIdCaso());
        setIdRequisito(dados.getIdRequisito());
    }

      /**
       * @ejb.persistence column-name = "ID"
       * @ejb.interface-method
       */
    public abstract Long getId();

      /**
       * @ejb.interface-method
       */
    public abstract void setId(Long id);

    /**
       * @ejb.persistence column-name = "ID_CASO"
       * @ejb.interface-method
       */
    public abstract Long getIdCaso();

      /**
       * @ejb.interface-method
       */
    public abstract void setIdCaso(Long idCaso);

      /**
       * @ejb.persistence column-name = "ID_REQUISITO"
       * @ejb.interface-method
       */
    public abstract Long getIdRequisito();

      /**
       * @ejb.interface-method
       */
    public abstract void setIdRequisito(Long idRequisito);

}
```

CategoriaBean.java

```
package org.hopi.ejb.entity;

import javax.ejb.CreateException;
import javax.ejb.EntityBean;

import org.hopi.ejb.base.EntityBase;

/**
 *
 * EJB Entity da Entidade de Dados CATEGORIA.
 *
 *  @author  Rafael  Andreis  Pires  (pires@inf.ufsc.br)  e  Marcel  Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 * @ejb.bean
 *    name = "Categoria"
 *    local-jndi-name="org/hopi/Categoria"
 *    view-type = "local"
 *    type = "CMP"
 *    cmp-version = "2.x"
 *    schema = "categoria"
 *    primkey-field = "id"
 *
 * @ejb.value-object
 *
 * @ejb.finder signature = "java.util.Collection findAll()"
 *                   query = "select Object(c) from categoria c"
 *
 */

public  abstract  class  CategoriaBean  extends  EntityBase  implements
EntityBean {

    /**
     * @ejb.create-method
     */
    public Long ejbCreate(VOCategoria dados) throws CreateException {
        setId(nextID("Categoria"));
        setDados(dados);
        return dados.getId();
    }

    public void ejbPostCreate(VOCategoria dados) throws CreateException
{
    }

    /**
     * @ejb.interface-method
     */
    public VOCategoria getDados() {
        VOCategoria dados = new VOCategoria();
        dados.setId(getId());
        dados.setNome(getNome());
        dados.setDescricao(getDescricao());
        return dados;
    }

    /**
```

```
     * @ejb.interface-method
     */
    public void setDados(VOCategoria dados) {
        setNome(dados.getNome());
        setDescricao(dados.getDescricao());
    }

     /**
      * @ejb.persistence column-name = "ID"
      * @ejb.interface-method
      */
    public abstract Long getId();

     /**
      * @ejb.interface-method
      */
    public abstract void setId(Long id);

     /**
      * @ejb.persistence column-name = "NOME"
      * @ejb.interface-method
      */
    public abstract String getNome();

     /**
      * @ejb.interface-method
      */
    public abstract void setNome(String nome);

     /**
      * @ejb.persistence column-name = "DESCRICAO"
      * @ejb.interface-method
      */
    public abstract String getDescricao();

     /**
      * @ejb.interface-method
      */
    public abstract void setDescricao(String descricao);

}
```

EscritorBean.java

```
package org.hopi.ejb.entity;

import javax.ejb.CreateException;
import javax.ejb.EntityBean;

import org.hopi.ejb.base.EntityBase;

/**
 *
 * EJB Entity da Entidade de Dados ESCRITOR.
 *
 *  @author  Rafael  Andreis  Pires  (pires@inf.ufsc.br)  e  Marcel  Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 * @ejb.bean
 *    name = "Escritor"
 *    local-jndi-name="org/hopi/Escritor"
 *    view-type = "local"
 *    type = "CMP"
 *    cmp-version = "2.x"
 *    schema = "escritor"
 *    primkey-field = "id"
 *
 * @ejb.value-object
 *
 *       @ejb.finder       signature       =       "java.util.Collection
findByIdUsuario(java.lang.Long idUsuario)"
 *                      query = "select Object(e) from escritor e where
e.idUsuario = ?1"
 *
 *       @ejb.finder       signature       =       "java.util.Collection
findByIdProjeto(java.lang.Long idProjeto)"
 *                      query = "select Object(e) from escritor e where
e.idProjeto = ?1"
 *
 */

public  abstract  class  EscritorBean  extends  EntityBase  implements
EntityBean {

    /**
     * @ejb.create-method
     */
    public Long ejbCreate(VOEscritor dados) throws CreateException {
        setId(nextID("Escritor"));
        setDados(dados);
        return getId();
    }

    public void ejbPostCreate(VOEscritor dados) throws CreateException {
    }

    /**
     * @ejb.interface-method
     */
    public VOEscritor getDados() {
        VOEscritor dados = new VOEscritor();
        dados.setId(getId());
```

```
        dados.setIdUsuario(getIdUsuario());
        dados.setIdProjeto(getIdProjeto());
        return dados;
    }

    /**
     * @ejb.interface-method
     */
    public void setDados(VOEscritor dados) {
        setIdUsuario(dados.getIdUsuario());
        setIdProjeto(dados.getIdProjeto());
    }

      /**
       * @ejb.persistence column-name = "ID"
       * @ejb.interface-method
       */
    public abstract Long getId();

      /**
       * @ejb.interface-method
       */
    public abstract void setId(Long id);

    /**
       * @ejb.persistence column-name = "ID_USUARIO"
       * @ejb.interface-method
       */
    public abstract Long getIdUsuario();

      /**
       * @ejb.interface-method
       */
    public abstract void setIdUsuario(Long idUsuario);

    /**
       * @ejb.persistence column-name = "ID_PROJETO"
       * @ejb.interface-method
       */
    public abstract Long getIdProjeto();

      /**
       * @ejb.interface-method
       */
    public abstract void setIdProjeto(Long idProjeto);

}
```

FluxoAlternativoBean.java

```
package org.hopi.ejb.entity;

import javax.ejb.CreateException;
import javax.ejb.EntityBean;

import org.hopi.ejb.base.EntityBase;

/**
 *
 * EJB Entity da Entidade de Dados FLUXO_ALTERNATIVO.
 *
 *  @author  Rafael  Andreis  Pires  (pires@inf.ufsc.br)  e  Marcel  Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 * @ejb.bean
 *    name = "FluxoAlternativo"
 *    local-jndi-name="org/hopi/FluxoAlternativo"
 *    view-type = "local"
 *    type = "CMP"
 *    cmp-version = "2.x"
 *    schema = "fluxoAlternativo"
 *    primkey-field = "id"
 *
 * @ejb.value-object
 *
 * @ejb.finder signature = "java.util.Collection findAll()"
 *                   query = "select Object(f) from fluxoAlternativo f"
 *
 */

public abstract class FluxoAlternativoBean extends EntityBase implements
EntityBean {

    /**
     * @ejb.create-method
     */
    public    Long    ejbCreate(VOFluxoAlternativo    dados)    throws
CreateException {
        setId(nextID("FluxoAlternativo"));
        setDados(dados);
        return dados.getId();
    }

    public   void   ejbPostCreate(VOFluxoAlternativo   dados)   throws
CreateException {
    }

    /**
     * @ejb.interface-method
     */
    public VOFluxoAlternativo getDados() {
        VOFluxoAlternativo dados = new VOFluxoAlternativo();
        dados.setId(getId());
        dados.setIdPassoInicial(getIdPassoInicial());
        dados.setDescricao(getDescricao());
        return dados;
    }
```

```
    /**
     * @ejb.interface-method
     */
    public void setDados(VOFluxoAlternativo dados) {
        setIdPassoInicial(dados.getIdPassoInicial());
        setDescricao(dados.getDescricao());
    }

    /**
     * @ejb.persistence column-name = "ID"
     * @ejb.interface-method
     */
    public abstract Long getId();

    /**
     * @ejb.interface-method
     */
    public abstract void setId(Long id);

    /**
     * @ejb.persistence column-name = "ID_PASSO_INICIAL"
     * @ejb.interface-method
     */
    public abstract Long getIdPassoInicial();

    /**
     * @ejb.interface-method
     */
    public abstract void setIdPassoInicial(Long idPassoInicial);

    /**
     * @ejb.persistence column-name = "ID_PASSO_ORIGEM"
     * @ejb.interface-method
     */
    public abstract Long getIdPassoOrigem();

    /**
     * @ejb.interface-method
     */
    public abstract void setIdPassoOrigem(Long idPassoOrigem);

    /**
     * @ejb.persistence column-name = "DESCRICAO"
     * @ejb.interface-method
     */
    public abstract String getDescricao();

    /**
     * @ejb.interface-method
     */
    public abstract void setDescricao(String descricao);

}
```

GeradorChaveBean.java

```
package org.hopi.ejb.entity;

import java.rmi.RemoteException;

import javax.ejb.CreateException;
import javax.ejb.EJBException;
import javax.ejb.EntityBean;
import javax.ejb.EntityContext;
import javax.ejb.FinderException;
import javax.ejb.RemoveException;

/**
 *
 * EJB Entity da Entidade de Dados GERADOR_CHAVE.
 *
 *  @author  Rafael  Andreis  Pires  (pires@inf.ufsc.br)  e  Marcel  Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 * @ejb.bean
 *    name = "GeradorChave"
 *    local-jndi-name="org/hopi/GeradorChave"
 *    view-type = "local"
 *    type = "CMP"
 *    cmp-version = "2.x"
 *      schema = "geradorchave"
 *    primkey-field = "entidade"
 *
 */
public abstract class GeradorChaveBean implements EntityBean {

     EntityContext ctx = null;

     // métodos de negócio
     /**
      * @ejb.home-method
      */
     public    Long    ejbHomeProximoValor(String    entidade)    throws
CreateException {
           GeradorChaveLocalHome                  home                  =
(GeradorChaveLocalHome)ctx.getEJBLocalHome();
           GeradorChaveLocal gerador = null;
           try {
                 gerador = home.findByPrimaryKey(entidade);
           }
           catch (FinderException e) {
           }
           if (gerador == null)
                       gerador = home.create(entidade, new Long(0));
           Long proximo = new Long(gerador.getValor().longValue() + 1);
           gerador.setValor(proximo);
           return proximo;
     }


     // métodos de criação
     /**
      * @ejb.create-method
      */
```

```
      public String ejbCreate(String entidade, Long valorInicial) throws
CreateException {
            setEntidade(entidade);
            setValor(valorInicial);
            return entidade;
      }

      public void ejbPostCreate(String entidade, Long valorInicial)  {
      }

      // atributos CMP
      /**
       * @ejb.persistence column-name = "entidade"
       * @ejb.interface-method
       */
      public abstract String getEntidade();

      /**
       * @ejb.interface-method
       */
      public abstract void setEntidade(String entidade);

      /**
       * @ejb.persistence column-name = "valor"
       * @ejb.interface-method
       */
      public abstract Long getValor();

      /**
       * @ejb.interface-method
       */
      public abstract void setValor(Long valor);

      // métodos da interface EntityBean
      public void ejbActivate() throws EJBException, RemoteException {
      }

      public void ejbLoad() throws EJBException, RemoteException {
      }

      public void ejbPassivate() throws EJBException, RemoteException {
      }

      public void ejbRemove()
            throws RemoveException, EJBException, RemoteException {
      }

      public void ejbStore() throws EJBException, RemoteException {
      }

      public void setEntityContext(EntityContext arg0)
                  throws EJBException, RemoteException {
            ctx = arg0;
      }

      public    void    unsetEntityContext()    throws    EJBException,
RemoteException {
            ctx = null;
      }

}
```

InteressadoBean.java

```
package org.hopi.ejb.entity;

import javax.ejb.CreateException;
import javax.ejb.EntityBean;

import org.hopi.ejb.base.EntityBase;

/**
 *
 * EJB Entity da Entidade de Dados INTERESSADO.
 *
 *  @author  Rafael  Andreis  Pires  (pires@inf.ufsc.br)  e  Marcel  Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 * @ejb.bean
 *    name = "Interessado"
 *    local-jndi-name="org/hopi/Interessado"
 *    view-type = "local"
 *    type = "CMP"
 *    cmp-version = "2.x"
 *    schema = "interessado"
 *    primkey-field = "id"
 *
 * @ejb.value-object
 *
 *       @ejb.finder       signature       =       "java.util.Collection
findByCasoDeUso(java.lang.Long idCaso)"
 *                    query = "select Object(c) from interessado c where
c.idCasoDeUso = ?1"
 *
 */

public  abstract  class  InteressadoBean  extends  EntityBase  implements
EntityBean {

    /**
     * @ejb.create-method
     */
    public Long ejbCreate(VOInteressado dados) throws CreateException {
        setId(nextID("Interessado"));
        setDados(dados);
        return dados.getId();
    }

    public     void     ejbPostCreate(VOInteressado     dados)     throws
CreateException {
    }

    /**
     * @ejb.interface-method
     */
    public VOInteressado getDados() {
        VOInteressado dados = new VOInteressado();
        dados.setId(getId());
        dados.setDescricao(getDescricao());
        dados.setIdCasoDeUso(getIdCasoDeUso());
        return dados;
    }
```

```
     /**
      * @ejb.interface-method
      */
     public void setDados(VOInteressado dados) throws CreateException {
         setDescricao(dados.getDescricao());
         setIdCasoDeUso(dados.getIdCasoDeUso());
     }

       /**
        * @ejb.persistence column-name = "ID"
        * @ejb.interface-method
        */
    public abstract Long getId();

       /**
        * @ejb.interface-method
        */
    public abstract void setId(Long id);

       /**
        * @ejb.persistence column-name = "DESCRICAO"
        * @ejb.interface-method
        */
    public abstract String getDescricao();

       /**
        * @ejb.interface-method
        */
    public abstract void setDescricao(String descricao);

    /**
     * @ejb.persistence column-name = "ID_CASO_DE_USO"
     * @ejb.interface-method
     */
    public abstract Long getIdCasoDeUso();

    /**
     * @ejb.value-object
     */
    public abstract void setIdCasoDeUso(Long idCasoDeUso);

}
```

LeitorBean.java

```
package org.hopi.ejb.entity;

import javax.ejb.CreateException;
import javax.ejb.EntityBean;

import org.hopi.ejb.base.EntityBase;

/**
 *
 * EJB Entity da Entidade de Dados LEITOR.
 *
 *  @author  Rafael  Andreis  Pires  (pires@inf.ufsc.br)  e  Marcel  Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 * @ejb.bean
 *    name = "Leitor"
 *    local-jndi-name="org/hopi/Leitor"
 *    view-type = "local"
 *    type = "CMP"
 *    cmp-version = "2.x"
 *    schema = "leitor"
 *    primkey-field = "id"
 *
 * @ejb.value-object
 *
 *       @ejb.finder       signature       =       "java.util.Collection
findByIdUsuario(java.lang.Long idUsuario)"
 *                        query  =  "select  Object(l)  from  leitor  l  where
l.idUsuario = ?1"
 *
 *       @ejb.finder       signature       =       "java.util.Collection
findByIdProjeto(java.lang.Long idProjeto)"
 *                        query  =  "select  Object(l)  from  leitor  l  where
l.idProjeto = ?1"
 *
 */

public  abstract  class  LeitorBean  extends  EntityBase  implements
EntityBean {

    /**
     * @ejb.create-method
     */
    public Long ejbCreate(VOLeitor dados) throws CreateException {
        setId(nextID("Leitor"));
        setDados(dados);
        return getId();
    }

    public void ejbPostCreate(VOLeitor dados) throws CreateException {
    }

    /**
     * @ejb.interface-method
     */
    public VOLeitor getDados() {
        VOLeitor dados = new VOLeitor();
        dados.setId(getId());
```

```
        dados.setIdUsuario(getIdUsuario());
        dados.setIdProjeto(getIdProjeto());
        return dados;
    }

    /**
     * @ejb.interface-method
     */
    public void setDados(VOLeitor dados) {
        setIdUsuario(dados.getIdUsuario());
        setIdProjeto(dados.getIdProjeto());
    }

     /**
      * @ejb.persistence column-name = "ID"
      * @ejb.interface-method
      */
    public abstract Long getId();

     /**
      * @ejb.interface-method
      */
    public abstract void setId(Long id);

    /**
      * @ejb.persistence column-name = "ID_USUARIO"
      * @ejb.interface-method
      */
    public abstract Long getIdUsuario();

     /**
      * @ejb.interface-method
      */
    public abstract void setIdUsuario(Long idUsuario);

    /**
      * @ejb.persistence column-name = "ID_PROJETO"
      * @ejb.interface-method
      */
    public abstract Long getIdProjeto();

     /**
      * @ejb.interface-method
      */
    public abstract void setIdProjeto(Long idProjeto);
}
```

PassoBean.java

```
package org.hopi.ejb.entity;

import javax.ejb.CreateException;
import javax.ejb.EntityBean;

import org.hopi.ejb.base.EntityBase;

/**
 *
 * EJB Entity da Entidade de Dados PASSO.
 *
 *  @author  Rafael  Andreis  Pires  (pires@inf.ufsc.br)  e  Marcel  Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 * @ejb.bean
 *    name = "Passo"
 *    local-jndi-name="org/hopi/Passo"
 *    view-type = "local"
 *    type = "CMP"
 *    cmp-version = "2.x"
 *    schema = "passo"
 *    primkey-field = "id"
 *
 * @ejb.value-object
 *
 *       @ejb.finder       signature       =       "java.util.Collection
findByPassoOrigem(java.lang.Long idPassoOrigem)"
 *                        query  =  "select  Object(p)  from  passo  p  where
p.idPassoOrigem = ?1"
 *
 */

public abstract class PassoBean extends EntityBase implements EntityBean
{

    /**
     * @ejb.create-method
     */
    public Long ejbCreate(VOPasso dados) throws CreateException {
        setId(nextID("Passo"));
        setDados(dados);
        return dados.getId();
    }

    public void ejbPostCreate(VOPasso dados) throws CreateException {
    }

    /**
     * @ejb.interface-method
     */
    public VOPasso getDados() {
        VOPasso dados = new VOPasso();
        dados.setId(getId());
        dados.setIdPassoOrigem(getIdPassoOrigem());
        dados.setDescricao(getDescricao());

        return dados;
    }
```

```
    /**
     * @ejb.interface-method
     */
    public void setDados(VOPasso dados) {
        setIdPassoOrigem(dados.getIdPassoOrigem());
        setDescricao(dados.getDescricao());
    }

    /**
     * @ejb.persistence column-name = "ID"
     * @ejb.interface-method
     */
    public abstract Long getId();

    /**
     * @ejb.interface-method
     */
    public abstract void setId(Long id);

    /**
     * @ejb.persistence column-name = "ID_PASSO_ORIGEM"
     * @ejb.interface-method
     */
    public abstract Long getIdPassoOrigem();

    /**
     * @ejb.interface-method
     */
    public abstract void setIdPassoOrigem(Long idPassoOrigem);

    /**
     * @ejb.persistence column-name = "DESCRICAO"
     * @ejb.interface-method
     */
    public abstract String getDescricao();

    /**
     * @ejb.interface-method
     */
    public abstract void setDescricao(String descricao);

}
```

PosCondicaoBean.java

```
package org.hopi.ejb.entity;

import javax.ejb.CreateException;
import javax.ejb.EntityBean;

import org.hopi.ejb.base.EntityBase;

/**
* EJB Entity da Entidade de Dados POS_CONDICAO.
 *
 *  @author  Rafael  Andreis  Pires  (pires@inf.ufsc.br)  e  Marcel  Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 * @ejb.bean
 *    name = "PosCondicao"
 *    local-jndi-name="org/hopi/PosCondicao"
 *    view-type = "local"
 *    type = "CMP"
 *    cmp-version = "2.x"
 *    schema = "posCondicao"
 *    primkey-field = "id"
 *
 * @ejb.value-object
 *
 *       @ejb.finder       signature       =       "java.util.Collection
findByCasoDeUso(java.lang.Long idCaso)"
 *                    query = "select Object(c) from posCondicao c where
c.idCasoDeUso = ?1"
*/

public  abstract  class  PosCondicaoBean  extends  EntityBase  implements
EntityBean {

    /**
     * @ejb.create-method
     */
    public Long ejbCreate(VOPosCondicao dados) throws CreateException {
        setId(nextID("PosCondicao"));
        setDados(dados);
        return dados.getId();
    }

    public     void     ejbPostCreate(VOPosCondicao     dados)     throws
CreateException {
    }

    /**
     * @ejb.interface-method
     */
    public VOPosCondicao getDados() {
        VOPosCondicao dados = new VOPosCondicao();
        dados.setId(getId());
        dados.setNome(getNome());
        dados.setDescricao(getDescricao());
        dados.setIdCasoDeUso(getIdCasoDeUso());
        return dados;
    }
```

```
    /**
     * @ejb.interface-method
     */
    public void setDados(VOPosCondicao dados) throws CreateException {
        setNome(dados.getNome());
        setDescricao(dados.getDescricao());
        setIdCasoDeUso(dados.getIdCasoDeUso());
    }

    /**
     * @ejb.persistence column-name = "ID"
     * @ejb.interface-method
     */
    public abstract Long getId();

    /**
     * @ejb.interface-method
     */
    public abstract void setId(Long id);

    /**
     * @ejb.persistence column-name = "ID_CASO_DE_USO"
     * @ejb.interface-method
     */
     public abstract Long getIdCasoDeUso();

     /**
      * @ejb.value-object
      */
     public abstract void setIdCasoDeUso(Long idCasoDeUso);

    /**
     * @ejb.persistence column-name = "NOME"
     * @ejb.interface-method
     */
    public abstract String getNome();

    /**
     * @ejb.interface-method
     */
    public abstract void setNome(String nome);

    /**
     * @ejb.persistence column-name = "DESCRICAO"
     * @ejb.interface-method
     */
    public abstract String getDescricao();

    /**
     * @ejb.interface-method
     */
    public abstract void setDescricao(String descricao);

}
```

Projeto.java

```
package org.hopi.ejb.entity;

import javax.ejb.CreateException;
import javax.ejb.EntityBean;

import org.hopi.ejb.base.EntityBase;

/**
 *
 * EJB Entity da Entidade de Dados PROJETO.
 *
 *  @author  Rafael  Andreis  Pires  (pires@inf.ufsc.br)  e  Marcel  Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 * @ejb.bean
 *    name = "Projeto"
 *    local-jndi-name="org/hopi/Projeto"
 *    view-type = "local"
 *    type = "CMP"
 *    cmp-version = "2.x"
 *    schema = "projeto"
 *    primkey-field = "id"
 *
 * @ejb.value-object
 *
 * @ejb.finder signature = "java.util.Collection findAll()"
 *                   query = "select Object(p) from projeto p"
 *
 */

public  abstract  class  ProjetoBean  extends  EntityBase  implements
EntityBean {


    /**
     * @ejb.create-method
     */
    public Long ejbCreate(VOProjeto dados) throws CreateException {
        setId(nextID("Projeto"));
        setDados(dados);
        return dados.getId();
    }

    public void ejbPostCreate(VOProjeto dados) throws CreateException {
    }

    /**
     * @ejb.interface-method
     */
    public VOProjeto getDados() {
        VOProjeto dados = new VOProjeto();
        dados.setId(getId());
        dados.setNome(getNome());
        dados.setDataInicial(getDataInicial());
        dados.setDataFinal(getDataFinal());
        dados.setSumarioExecutivo(getSumarioExecutivo());
        return dados;
    }
```

```
/**
 * @ejb.interface-method
 */
public void setDados(VOProjeto dados) {
    setNome(dados.getNome());
    setDataInicial(dados.getDataInicial());
    setDataFinal(dados.getDataFinal());
    setSumarioExecutivo(dados.getSumarioExecutivo());
}

/**
 * @ejb.persistence column-name = "ID"
 * @ejb.interface-method
 */
public abstract Long getId();

/**
 * @ejb.interface-method
 */
public abstract void setId(Long id);

/**
 * @ejb.persistence column-name = "NOME"
 * @ejb.interface-method
 */
public abstract String getNome();

/**
 * @ejb.interface-method
 */
public abstract void setNome(String nome);

/**
 * @ejb.persistence column-name = "DATA_INICIAL"
 * @ejb.interface-method
 */
public abstract Long getDataInicial();

/**
 * @ejb.interface-method
 */
public abstract void setDataInicial(Long dataInicial);

/**
 * @ejb.persistence column-name = "DATA_FINAL"
 * @ejb.interface-method
 */
public abstract Long getDataFinal();

/**
 * @ejb.interface-method
 */
public abstract void setDataFinal(Long dataFinal);

/**
 * @ejb.persistence column-name = "SUMARIO_EXECUTIVO"
 * @ejb.interface-method
 */
public abstract String getSumarioExecutivo();
```

```
    /**
     * @ejb.interface-method
     */
    public abstract void setSumarioExecutivo(String sumarioExecutivo);

}
```

QuestaoEmAbertoBean.java

```
package org.hopi.ejb.entity;

import javax.ejb.CreateException;
import javax.ejb.EntityBean;

import org.hopi.ejb.base.EntityBase;

/**
 *
 * EJB Entity da Entidade de Dados QUESTAO_EM_ABERTO.
 *
 *  @author  Rafael  Andreis  Pires  (pires@inf.ufsc.br)  e  Marcel  Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 * @ejb.bean
 *    name = "QuestaoEmAberto"
 *    local-jndi-name="org/hopi/QuestaoEmAberto"
 *    view-type = "local"
 *    type = "CMP"
 *    cmp-version = "2.x"
 *    schema = "questaoEmAberto"
 *    primkey-field = "id"
 *
 * @ejb.value-object
 *
 *       @ejb.finder       signature       =       "java.util.Collection
findByCasoDeUso(java.lang.Long idCaso)"
 *                      query = "select Object(c) from questaoEmAberto c
where c.idCasoDeUso = ?1"
 *
 */

public abstract class QuestaoEmAbertoBean extends EntityBase implements
EntityBean {

    /**
     * @ejb.create-method
     */
    public     Long     ejbCreate(VOQuestaoEmAberto     dados)     throws
CreateException {
        setId(nextID("QuestaoEmAberto"));
        setDados(dados);
        return dados.getId();
    }

    public    void    ejbPostCreate(VOQuestaoEmAberto    dados)    throws
CreateException {
    }

    /**
     * @ejb.interface-method
     */
    public VOQuestaoEmAberto getDados() {
        VOQuestaoEmAberto dados = new VOQuestaoEmAberto();
        dados.setId(getId());
        dados.setDescricao(getDescricao());
        dados.setIdCasoDeUso(getIdCasoDeUso());
        return dados;
```

```
    }

    /**
     * @ejb.interface-method
     */
    public void setDados(VOQuestaoEmAberto dados) throws CreateException
{
        setDescricao(dados.getDescricao());
        setIdCasoDeUso(dados.getIdCasoDeUso());
    }

    /**
     * @ejb.persistence column-name = "ID"
     * @ejb.interface-method
     */
    public abstract Long getId();

    /**
     * @ejb.interface-method
     */
    public abstract void setId(Long id);

    /**
     * @ejb.persistence column-name = "DESCRICAO"
     * @ejb.interface-method
     */
    public abstract String getDescricao();

    /**
     * @ejb.interface-method
     */
    public abstract void setDescricao(String descricao);

      /**
       * @ejb.persistence column-name = "ID_CASO_DE_USO"
       * @ejb.interface-method
       */
      public abstract Long getIdCasoDeUso();

      /**
       * @ejb.value-object
       */
      public abstract void setIdCasoDeUso(Long idCasoDeUso);

}
```

RequisitoFuncionalBean.java

```
package org.hopi.ejb.entity;

import javax.ejb.CreateException;
import javax.ejb.EntityBean;

import org.hopi.ejb.base.EntityBase;

/**
 *
 * EJB Entity da Entidade de Dados REQUISITO_FUNCIONAL.
 *
 *  @author  Rafael  Andreis  Pires  (pires@inf.ufsc.br)  e  Marcel  Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 * @ejb.bean
 *    name = "RequisitoFuncional"
 *    local-jndi-name="org/hopi/RequisitoFuncional"
 *    view-type = "local"
 *    type = "CMP"
 *    cmp-version = "2.x"
 *    schema = "requisitoFuncional"
 *    primkey-field = "id"
 *
 * @ejb.value-object
 *
 *       @ejb.finder       signature       =       "java.util.Collection
findByProjeto(java.lang.Long idProjeto)"
 *                   query = "select Object(c) from requisitoFuncional c
where c.idProjeto = ?1"
 *
 */

public   abstract   class   RequisitoFuncionalBean   extends   EntityBase
implements EntityBean {


    /**
     * @ejb.create-method
     */
    public    Long    ejbCreate(VORequisitoFuncional    dados)    throws
CreateException {
        setId(nextID("RequisitoFuncional"));
        setDados(dados);
        return dados.getId();
    }

    public   void   ejbPostCreate(VORequisitoFuncional   dados)   throws
CreateException {
    }

    /**
     * @ejb.interface-method
     */
    public VORequisitoFuncional getDados() {
        VORequisitoFuncional dados = new VORequisitoFuncional();
        dados.setId(getId());
        dados.setNome(getNome());
        dados.setDescricao(getDescricao());
```

```
        dados.setOculto(getOculto());
        dados.setIdProjeto(getIdProjeto());
        return dados;
    }

    /**
     * @ejb.interface-method
     */
    public    void    setDados(VORequisitoFuncional    dados)    throws
CreateException {
        setNome(dados.getNome());
        setDescricao(dados.getDescricao());
        setOculto(dados.getOculto());
        setIdProjeto(dados.getIdProjeto());
    }

    /**
     * @ejb.persistence column-name = "ID"
     * @ejb.interface-method
     */
    public abstract Long getId();

    /**
     * @ejb.interface-method
     */
    public abstract void setId(Long id);

    /**
     * @ejb.persistence column-name = "NOME"
     * @ejb.interface-method
     */
    public abstract String getNome();

    /**
     * @ejb.interface-method
     */
    public abstract void setNome(String nome);

    /**
     * @ejb.persistence column-name = "DESCRICAO"
     * @ejb.interface-method
     */
    public abstract String getDescricao();

    /**
     * @ejb.interface-method
     */
    public abstract void setDescricao(String descricao);

    /**
     * @ejb.persistence column-name = "OCULTO"
     * @ejb.interface-method
     */
    public abstract Boolean getOculto();

    /**
     * @ejb.interface-method
     */
    public abstract void setOculto(Boolean oculto);

    /**
```

```
     * @ejb.persistence column-name = "ID_PROJETO"
     * @ejb.interface-method
     */
     public abstract Long getIdProjeto();

     /**
      * @ejb.value-object
      */
     public abstract void setIdProjeto(Long idProjeto);

}
```

RequisitoNaoFuncionalBean.java

```
package org.hopi.ejb.entity;

import javax.ejb.CreateException;
import javax.ejb.EntityBean;
import org.hopi.ejb.base.EntityBase;

/**
* EJB Entity da Entidade de Dados REQUISITO_NAO_FUNCIONAL.
 *
 * @author Rafael Andreis Pires (pires@inf.ufsc.br) e Marcel Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 * @ejb.bean
 *    name = "RequisitoNaoFuncional"
 *    local-jndi-name="org/hopi/RequisitoNaoFuncional"
 *    view-type = "local"
 *    type = "CMP"
 *    cmp-version = "2.x"
 *    schema = "requisitoNaoFuncional"
 *    primkey-field = "id"
 *
 * @ejb.value-object
 *
 *       @ejb.finder       signature       =       "java.util.Collection
findByRequisitoFuncional(java.lang.Long idRequisito)"
 *                  query = "select Object(r) from requisitoNaoFuncional
r where r.idRequisitoFuncional = ?1"
 *
 *       @ejb.finder       signature       =       "java.util.Collection
findByProjeto(java.lang.Long idProjeto)"
 *                  query = "select Object(r) from requisitoNaoFuncional
r where r.idProjeto = ?1"
 */

public  abstract  class  RequisitoNaoFuncionalBean  extends  EntityBase
implements EntityBean {
    /**
     * @ejb.create-method
     */
    public   Long   ejbCreate(VORequisitoNaoFuncional   dados)   throws
CreateException {
        setId(nextID("RequisitoNaoFuncional"));
        setDados(dados);
        return dados.getId();
    }

    public  void  ejbPostCreate(VORequisitoNaoFuncional  dados)  throws
CreateException {
    }

    /**
     * @ejb.interface-method
     */
    public VORequisitoNaoFuncional getDados() {
        VORequisitoNaoFuncional dados = new VORequisitoNaoFuncional();
        dados.setId(getId());
        dados.setNome(getNome());
        dados.setDescricao(getDescricao());
```

```
        dados.setObrigatorio(getObrigatorio());
        dados.setPermanente(getPermanente());
        dados.setIdRequisitoFuncional(getIdRequisitoFuncional());
        dados.setIdCategoria(getIdCategoria());
        dados.setIdProjeto(getIdProjeto());
        return dados;
    }

    /**
     * @ejb.interface-method
     */
    public    void    setDados(VORequisitoNaoFuncional    dados)    throws
CreateException {
        setNome(dados.getNome());
        setDescricao(dados.getDescricao());
        setObrigatorio(dados.getObrigatorio());
        setPermanente(dados.getPermanente());
        setIdRequisitoFuncional(dados.getIdRequisitoFuncional());
        setIdCategoria(dados.getIdCategoria());
        setIdProjeto(dados.getIdProjeto());
    }

    /**
     * @ejb.persistence column-name = "ID"
     * @ejb.interface-method
     */
    public abstract Long getId();

    /**
     * @ejb.interface-method
     */
    public abstract void setId(Long id);

    /**
     * @ejb.persistence column-name = "NOME"
     * @ejb.interface-method
     */
    public abstract String getNome();

    /**
     * @ejb.interface-method
     */
    public abstract void setNome(String nome);

    /**
     * @ejb.persistence column-name = "DESCRICAO"
     * @ejb.interface-method
     */
    public abstract String getDescricao();

    /**
     * @ejb.interface-method
     */
    public abstract void setDescricao(String descricao);

    /**
     * @ejb.persistence column-name = "OBRIGATORIO"
     * @ejb.interface-method
     */
    public abstract Boolean getObrigatorio();
```

```
    /**
     * @ejb.interface-method
     */
    public abstract void setObrigatorio(Boolean obrigatorio);

    /**
     * @ejb.persistence column-name = "PERMANENTE"
     * @ejb.interface-method
     */
    public abstract Boolean getPermanente();

    /**
     * @ejb.interface-method
     */
    public abstract void setPermanente(Boolean permanente);

    /**
     * @ejb.persistence column-name = "ID_REQUISITO_FUNCIONAL"
     * @ejb.interface-method
     */
    public abstract Long getIdRequisitoFuncional();

    /**
     * @ejb.interface-method
     */
    public abstract void setIdRequisitoFuncional(Long idRequisito);

    /**
     * @ejb.persistence column-name = "ID_CATEGORIA"
     * @ejb.interface-method
     */
    public abstract Long getIdCategoria();

    /**
     * @ejb.interface-method
     */
    public abstract void setIdCategoria(Long idCategoria);

    /**
     * @ejb.persistence column-name = "ID_PROJETO"
     * @ejb.interface-method
     */
    public abstract Long getIdProjeto();

    /**
     * @ejb.interface-method
     */
    public abstract void setIdProjeto(Long idProjeto);
}
```

UsuarioBean.java

```
package org.hopi.ejb.entity;

import javax.ejb.CreateException;
import javax.ejb.EntityBean;
import org.hopi.ejb.base.EntityBase;

/**
 *
 * EJB Entity da Entidade de Dados USUARIO.
 *
 *  @author  Rafael  Andreis  Pires  (pires@inf.ufsc.br)  e  Marcel  Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 * @ejb.bean
 *    name = "Usuario"
 *    local-jndi-name="org/hopi/Usuario"
 *    view-type = "local"
 *    type = "CMP"
 *    cmp-version = "2.x"
 *    schema = "usuario"
 *    primkey-field = "id"
 *
 * @ejb.value-object
 *
 * @ejb.finder signature = "java.util.Collection findAll()"
 *                 query = "select Object(u) from usuario u"
 *
 * @ejb.finder signature = "UsuarioLocal findByUsername(java.lang.String
username)"
 *                       query  =  "select  Object(u)  from  usuario  u  where
u.username = ?1"
 *
 */

public  abstract  class  UsuarioBean  extends  EntityBase  implements
EntityBean {

    /**
     * @ejb.create-method
     */
    public Long ejbCreate(VOUsuario dados) throws CreateException {
        setId(nextID("Usuario"));
        setDados(dados);
        return dados.getId();
    }

    public void ejbPostCreate(VOUsuario dados) throws CreateException {
    }

    /**
     * @ejb.interface-method
     */
    public VOUsuario getDados() {
        VOUsuario dados = new VOUsuario();
        dados.setId(getId());
        dados.setUsername(getUsername());
        dados.setPassword(getPassword());
        dados.setMatricula(getMatricula());
```

```
    dados.setAdministrador(getAdministrador());
    dados.setNome(getNome());
    dados.setTelefone(getTelefone());
    dados.setEmail(getEmail());

    return dados;
}

/**
 * @ejb.interface-method
 */
public void setDados(VOUsuario dados) throws CreateException {
    setUsername(dados.getUsername());
    setPassword(dados.getPassword());
    setMatricula(dados.getMatricula());
    setAdministrador(dados.getAdministrador());
    setNome(dados.getNome());
    setTelefone(dados.getTelefone());
    setEmail(dados.getEmail());
}

  /**
   * @ejb.persistence column-name = "ID"
   * @ejb.interface-method
   */
public abstract Long getId();

  /**
   * @ejb.interface-method
   */
public abstract void setId(Long id);

  /**
   * @ejb.persistence column-name = "USERNAME"
   * @ejb.interface-method
   */
public abstract String getUsername();

  /**
   * @ejb.interface-method
   */
public abstract void setUsername(String username);

  /**
   * @ejb.persistence column-name = "PASSWORD"
   * @ejb.interface-method
   */
public abstract String getPassword();

  /**
   * @ejb.interface-method
   */
public abstract void setPassword(String password);

  /**
   * @ejb.persistence column-name = "MATRICULA"
   * @ejb.interface-method
   */
public abstract String getMatricula();

  /**
```

```
     * @ejb.interface-method
     */
    public abstract void setMatricula(String matricula);

    /**
     * @ejb.persistence column-name = "ADMINISTRADOR"
     * @ejb.interface-method
     */
    public abstract Boolean getAdministrador();

    /**
     * @ejb.interface-method
     */
    public abstract void setAdministrador(Boolean administrador);

    /**
     * @ejb.persistence column-name = "NOME"
     * @ejb.interface-method
     */
    public abstract String getNome();

    /**
     * @ejb.interface-method
     */
    public abstract void setNome(String nome);

    /**
     * @ejb.persistence column-name = "TELEFONE"
     * @ejb.interface-method
     */
    public abstract String getTelefone();

    /**
     * @ejb.interface-method
     */
    public abstract void setTelefone(String telefone);

    /**
     * @ejb.persistence column-name = "EMAIL"
     * @ejb.interface-method
     */
    public abstract String getEmail();

    /**
     * @ejb.interface-method
     */
    public abstract void setEmail(String email);

}
```

VariacaoTecnologicaBean.java

```
package org.hopi.ejb.entity;

import javax.ejb.CreateException;
import javax.ejb.EntityBean;

import org.hopi.ejb.base.EntityBase;

/**
 *
 * EJB Entity da Entidade de Dados VARIACAO_TECNOLOGICA.
 *
 *  @author  Rafael  Andreis  Pires  (pires@inf.ufsc.br)  e  Marcel  Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 * @ejb.bean
 *    name = "VariacaoTecnologica"
 *    local-jndi-name="org/hopi/VariacaoTecnologica"
 *    view-type = "local"
 *    type = "CMP"
 *    cmp-version = "2.x"
 *    schema = "variacaoTecnologica"
 *    primkey-field = "id"
 *
 * @ejb.value-object
 *
 *       @ejb.finder       signature       =       "java.util.Collection
findByCasoDeUso(java.lang.Long idCaso)"
 *                     query = "select Object(c) from variacaoTecnologica
c where c.idCasoDeUso = ?1"
 *
 */

public   abstract   class   VariacaoTecnologicaBean   extends   EntityBase
implements EntityBean {

    /**
     * @ejb.create-method
     */
    public    Long    ejbCreate(VOVariacaoTecnologica    dados)    throws
CreateException {
        setId(nextID("VariacaoTecnologica"));
        setDados(dados);
        return dados.getId();
    }

    public   void   ejbPostCreate(VOVariacaoTecnologica   dados)   throws
CreateException {
    }

    /**
     * @ejb.interface-method
     */
    public VOVariacaoTecnologica getDados() {
        VOVariacaoTecnologica dados = new VOVariacaoTecnologica();
        dados.setId(getId());
        dados.setDescricao(getDescricao());
        dados.setIdCasoDeUso(getIdCasoDeUso());
        return dados;
```

```
    }

    /**
     * @ejb.interface-method
     */
    public    void    setDados(VOVariacaoTecnologica    dados)    throws
CreateException {
        setDescricao(dados.getDescricao());
        setIdCasoDeUso(dados.getIdCasoDeUso());
    }

      /**
       * @ejb.persistence column-name = "ID"
       * @ejb.interface-method
       */
   public abstract Long getId();

      /**
       * @ejb.interface-method
       */
   public abstract void setId(Long id);

      /**
       * @ejb.persistence column-name = "DESCRICAO"
       * @ejb.interface-method
       */
   public abstract String getDescricao();

      /**
       * @ejb.interface-method
       */
   public abstract void setDescricao(String descricao);

      /**
       * @ejb.persistence column-name = "ID_CASO_DE_USO"
       * @ejb.interface-method
       */
      public abstract Long getIdCasoDeUso();

      /**
       * @ejb.value-object
       */
      public abstract void setIdCasoDeUso(Long idCasoDeUso);

}
```

PACOTE ORG.HOPI.EJB.ENTITY.FACADE

AtorDoCasoDeUsoFacadeBean.java

```
package org.hopi.ejb.entity.facade;

import java.util.ArrayList;
import java.util.Collection;
import java.util.Iterator;

import javax.ejb.CreateException;
import javax.ejb.EJBException;
import javax.ejb.SessionBean;

import org.hopi.ejb.base.SessionBase;
import org.hopi.ejb.entity.AtorDoCasoLocalHome;
import org.hopi.ejb.entity.AtorLocalHome;
import org.hopi.ejb.entity.CasoDeUsoLocalHome;
import org.hopi.ejb.entity.VOAtorDoCaso;
import org.hopi.ejb.util.JNDI;
import org.hopi.ejb.util.ListAssembler;

/**
 *
 * Fachada da Entidade de Dados Ator - Caso de Uso.
 *
 *  @author  Rafael  Andreis  Pires  (pires@inf.ufsc.br)  e  Marcel  Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 * @ejb.bean
 *    name      = "AtorDoCasoDeUsoFacade"
 *    jndi-name = "org/hopi/facade/AtorDoCasoDeUsoFacade"
 *    view-type = "remote"
 *      type      = "Stateless"
 */

public  class  AtorDoCasoDeUsoFacadeBean  extends  SessionBase  implements
SessionBean {

    /**
     * @ejb.create-method
     */
    public void ejbCreate() throws CreateException {
    }

     /**
      * @ejb.interface-method
      */
    public VOAtorDoCaso createAtorDoCasoDeUso(VOAtorDoCaso dados) throws
EJBException {
        try {
            AtorDoCasoLocalHome  atorDoCasoHome  =  (AtorDoCasoLocalHome)
lookup(JNDI.ATOR_DO_CASO);
            return atorDoCasoHome.create(dados).getDados();
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
```

```
      * @ejb.interface-method
      */
    public void destroy(Long id) throws EJBException {
        try {
            AtorDoCasoLocalHome  atorDoCasoHome  =  (AtorDoCasoLocalHome)
lookup(JNDI.ATOR_DO_CASO);
            atorDoCasoHome.findByPrimaryKey(id).remove();
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public Collection findByIdAtor(Long idAtor) throws EJBException {
        try {
            AtorDoCasoLocalHome                        atorDoCasoHome    =
(AtorDoCasoLocalHome) lookup(JNDI.ATOR_DO_CASO);
            CasoDeUsoLocalHome                casoHome                =
(CasoDeUsoLocalHome)  lookup(JNDI.CASO_DE_USO);
            ArrayList                casosDeUso     = new ArrayList();

            Iterator                           it                          =
ListAssembler.montaColecaoVOs(atorDoCasoHome.findByIdAtor(idAtor)).itera
tor();
            VOAtorDoCaso vo = null;
            while ( it.hasNext() )
            {
                vo = (VOAtorDoCaso) it.next();

casosDeUso.add(casoHome.findByPrimaryKey(vo.getIdCaso()).getDados());
            }
            return casosDeUso;
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public   Collection   findByIdCasoDeUso(Long   idCasoDeUso)   throws
EJBException {
        try {
            AtorDoCasoLocalHome  atorDoCasoHome  =  (AtorDoCasoLocalHome)
lookup(JNDI.ATOR_DO_CASO);
            AtorLocalHome           atorHome           =  (AtorLocalHome)
lookup(JNDI.ATOR);
            ArrayList           atores         = new ArrayList();

            Iterator                           it                          =
ListAssembler.montaColecaoVOs(atorDoCasoHome.findByIdCasoDeUso(idCasoDeU
so)).iterator();
            VOAtorDoCaso vo = null;
            while ( it.hasNext() )
            {
                vo = (VOAtorDoCaso) it.next();

atores.add(atorHome.findByPrimaryKey(vo.getIdAtor()).getDados());
            }
```

```
            return atores;
        } catch(Exception e) {
            throw new EJBException();
        }
    }

}
```

CasoDeUsoDoRequisitoFuncionalFacadeBean.java

```
package org.hopi.ejb.entity.facade;

import java.util.ArrayList;
import java.util.Collection;
import java.util.Iterator;

import javax.ejb.CreateException;
import javax.ejb.EJBException;
import javax.ejb.SessionBean;

import org.hopi.ejb.base.SessionBase;
import org.hopi.ejb.entity.CasoDeUsoLocalHome;
import org.hopi.ejb.entity.CasoDoRequisitoLocalHome;
import org.hopi.ejb.entity.RequisitoFuncionalLocalHome;
import org.hopi.ejb.entity.VOCasoDoRequisito;
import org.hopi.ejb.util.JNDI;
import org.hopi.ejb.util.ListAssembler;

/**
 *
 * Fachada da Entidade de Dados Caso de Uso - Requisito Funcional.
 *
 *  @author  Rafael  Andreis  Pires  (pires@inf.ufsc.br)  e  Marcel  Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 * @ejb.bean
 *    name      = "CasoDoRequisitoFacade"
 *    jndi-name = "org/hopi/facade/CasoDoRequisitoFacade"
 *    view-type = "remote"
 *      type      = "Stateless"
 */

public class CasoDeUsoDoRequisitoFuncionalFacadeBean extends SessionBase
implements SessionBean {

    /**
     * @ejb.create-method
     */
    public void ejbCreate() throws CreateException {
    }

      /**
       * @ejb.interface-method
       */
    public   VOCasoDoRequisito   createCasoDoRequisito(VOCasoDoRequisito
dados) throws EJBException {
        try {
            CasoDoRequisitoLocalHome        casoDoRequisitoHome        =
(CasoDoRequisitoLocalHome) lookup(JNDI.CASO_DO_REQUISITO);
            return casoDoRequisitoHome.create(dados).getDados();
        } catch(Exception e) {
            throw new EJBException();
        }
    }

      /**
       * @ejb.interface-method
       */
```

```
    public void destroy(Long id) throws EJBException {
        try {
            CasoDoRequisitoLocalHome        casoDoRequisitoHome        =
(CasoDoRequisitoLocalHome) lookup(JNDI.CASO_DO_REQUISITO_FACADE);
            casoDoRequisitoHome.findByPrimaryKey(id).remove();
        } catch(Exception e) {
            throw new EJBException();
        }
    }

      /**
       * @ejb.interface-method
       */
    public   Collection   findByIdRequisito(Long   idRequisito)   throws
EJBException {
        try {
            CasoDoRequisitoLocalHome        casoDoRequisitoHome        =
(CasoDoRequisitoLocalHome) lookup(JNDI.CASO_DO_REQUISITO);
            CasoDeUsoLocalHome            casoHome                    =
(CasoDeUsoLocalHome)       lookup(JNDI.CASO_DE_USO);
            ArrayList                     casosDeUso             = new
ArrayList();

            Iterator                       it                         =
ListAssembler.montaColecaoVOs(casoDoRequisitoHome.findByIdRequisito(idRe
quisito)).iterator();
            VOCasoDoRequisito vo = null;
            while ( it.hasNext() )
            {
                vo = (VOCasoDoRequisito) it.next();

casosDeUso.add(casoHome.findByPrimaryKey(vo.getIdCaso()).getDados());
            }
            return casosDeUso;
        } catch(Exception e) {
            throw new EJBException();
        }
    }

      /**
       * @ejb.interface-method
       */
    public   Collection   findByIdCasoDeUso(Long   idCasoDeUso)   throws
EJBException {
        try {
            CasoDoRequisitoLocalHome            casoDoRequisitoHome   =
(CasoDoRequisitoLocalHome)    lookup(JNDI.CASO_DO_REQUISITO);
            RequisitoFuncionalLocalHome  requisitoHome                =
(RequisitoFuncionalLocalHome) lookup(JNDI.REQUISITO_FUNCIONAL);
            ArrayList                       requisitos            = new
ArrayList();

            Iterator                        it                        =
ListAssembler.montaColecaoVOs(casoDoRequisitoHome.findByIdCasoDeUso(idCa
soDeUso)).iterator();
            VOCasoDoRequisito vo = null;
            while ( it.hasNext() )
            {
                vo = (VOCasoDoRequisito) it.next();
```

```
requisitos.add(requisitoHome.findByPrimaryKey(vo.getIdRequisito()).getDa
dos());
            }
            return requisitos;
        } catch(Exception e) {
            throw new EJBException();
        }
    }

}
```

CasoDeUsoFacadeBean.java

```
package org.hopi.ejb.entity.facade;

import java.util.Collection;

import javax.ejb.CreateException;
import javax.ejb.EJBException;
import javax.ejb.SessionBean;

import org.hopi.ejb.base.SessionBase;
import org.hopi.ejb.entity.CasoDeUsoLocalHome;
import org.hopi.ejb.entity.InteressadoLocalHome;
import org.hopi.ejb.entity.PosCondicaoLocalHome;
import org.hopi.ejb.entity.PreCondicaoLocalHome;
import org.hopi.ejb.entity.QuestaoEmAbertoLocalHome;
import org.hopi.ejb.entity.VOCasoDeUso;
import org.hopi.ejb.entity.VOFluxoAlternativo;
import org.hopi.ejb.entity.VOInteressado;
import org.hopi.ejb.entity.VOPasso;
import org.hopi.ejb.entity.VOPosCondicao;
import org.hopi.ejb.entity.VOPreCondicao;
import org.hopi.ejb.entity.VOQuestaoEmAberto;
import org.hopi.ejb.entity.VOVariacaoTecnologica;
import org.hopi.ejb.entity.VariacaoTecnologicaLocalHome;
import org.hopi.ejb.util.JNDI;
import org.hopi.ejb.util.ListAssembler;

/**
 *
 *  Fachada da Entidade de Dados Caso de Uso e seus atributos
relacionados.
 *
 *  @author Rafael Andreis Pires (pires@inf.ufsc.br) e Marcel Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 * @ejb.bean
 *    name      = "CasoDeUsoFacade"
 *    jndi-name = "org/hopi/facade/CasoDeUsoFacade"
 *    view-type = "remote"
 *      type      = "Stateless"
 */

public class CasoDeUsoFacadeBean extends SessionBase implements
SessionBean {

    /**
     * @ejb.create-method
     */
    public void ejbCreate() throws CreateException {
    }

     /**
      * @ejb.interface-method
      */
    public VOCasoDeUso createCasoDeUso(VOCasoDeUso dados) throws
EJBException {
        try {
            CasoDeUsoLocalHome    casoHome    =    (CasoDeUsoLocalHome)
lookup(JNDI.CASO_DE_USO);
```

```
            return casoHome.create(dados).getDados();
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public void setData(VOCasoDeUso dados) throws EJBException {
        try {
            CasoDeUsoLocalHome    casoHome    =    (CasoDeUsoLocalHome)
lookup(JNDI.CASO_DE_USO);
            casoHome.findByPrimaryKey(dados.getId()).getDados();
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public void destroy(Long idCasoDeUso) throws EJBException {
        try {
            CasoDeUsoLocalHome    casoHome    =    (CasoDeUsoLocalHome)
lookup(JNDI.CASO_DE_USO);
            casoHome.findByPrimaryKey(idCasoDeUso).remove();
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public    Collection    getAllByIdProjeto(Long    idProjeto)    throws
EJBException {
        try {
            CasoDeUsoLocalHome    casoHome    =    (CasoDeUsoLocalHome)
lookup(JNDI.CASO_DE_USO);
            return
ListAssembler.montaColecaoVOs(casoHome.findByProjeto(idProjeto));
        } catch(Exception e) {
            throw new EJBException();
        }
    }

    /**************
     * PRE CONDICAO
     **************/

     /**
      * @ejb.interface-method
      */
    public  VOPreCondicao  addPreCondicao(VOPreCondicao  dados)  throws
EJBException {
        try{
            PreCondicaoLocalHome   preHome   =   (PreCondicaoLocalHome)
lookup(JNDI.PRE_CONDICAO);
            return preHome.create(dados).getDados();
        } catch(Exception e) {
```

```
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public void editPreCondicao(VOPreCondicao dados) throws EJBException
{
        try {
            PreCondicaoLocalHome   preHome   =   (PreCondicaoLocalHome)
lookup(JNDI.PRE_CONDICAO);
            preHome.findByPrimaryKey(dados.getId()).setDados(dados);
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public void removePreCondicao(Long id) throws EJBException {
        try {
            PreCondicaoLocalHome   preHome   =   (PreCondicaoLocalHome)
lookup(JNDI.PRE_CONDICAO);
            preHome.findByPrimaryKey(id).remove();
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public Collection getPreCondicoes(Long idCaso) throws EJBException {
        try {
            PreCondicaoLocalHome   preHome   =   (PreCondicaoLocalHome)
lookup(JNDI.PRE_CONDICAO);
            return
ListAssembler.montaColecaoVOs(preHome.findByCasoDeUso(idCaso));
        } catch(Exception e) {
            throw new EJBException();
        }
    }

    /**************
     * POS CONDICAO
     **************/

     /**
      * @ejb.interface-method
      */
    public  VOPosCondicao  addPosCondicao(VOPosCondicao  dados)  throws
EJBException {
        try {
            PosCondicaoLocalHome   posHome   =   (PosCondicaoLocalHome)
lookup(JNDI.POS_CONDICAO);
            return posHome.create(dados).getDados();
        } catch(Exception e) {
            throw new EJBException();
        }
```

```
    }

     /**
      * @ejb.interface-method
      */
    public void editPosCondicao(VOPosCondicao dados) throws EJBException
{
        try {
            PosCondicaoLocalHome   posHome   =   (PosCondicaoLocalHome)
lookup(JNDI.POS_CONDICAO);
            posHome.findByPrimaryKey(dados.getId()).setDados(dados);
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public void removePosCondicao(Long id) throws EJBException {
        try {
            PosCondicaoLocalHome   posHome   =   (PosCondicaoLocalHome)
lookup(JNDI.POS_CONDICAO);
            posHome.findByPrimaryKey(id).remove();
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public Collection getPosCondicoes(Long idCaso) throws EJBException {
        try {
            PosCondicaoLocalHome   posHome   =   (PosCondicaoLocalHome)
lookup(JNDI.POS_CONDICAO);
            return
ListAssembler.montaColecaoVOs(posHome.findByCasoDeUso(idCaso));
        } catch(Exception e) {
            throw new EJBException();
        }
    }

    /**************
     * INTERESSADO
     **************/

     /**
      * @ejb.interface-method
      */
    public  VOInteressado  addInteressado(VOInteressado  dados)  throws
EJBException {
        try {
            InteressadoLocalHome            interessadoHome            =
(InteressadoLocalHome) lookup(JNDI.INTERESSADO);
            return interessadoHome.create(dados).getDados();
        } catch(Exception e) {
            throw new EJBException();
        }
    }
```

```
     /**
      * @ejb.interface-method
      */
    public void editInteressado(VOInteressado dados) throws EJBException
{
        try {
            InteressadoLocalHome            interessadoHome            =
(InteressadoLocalHome) lookup(JNDI.INTERESSADO);

interessadoHome.findByPrimaryKey(dados.getId()).setDados(dados);
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public void removeInteressado(Long id) throws EJBException {
        try {
            InteressadoLocalHome            interessadoHome            =
(InteressadoLocalHome) lookup(JNDI.INTERESSADO);
            interessadoHome.findByPrimaryKey(id).remove();
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public Collection getInteressados(Long idCaso) throws EJBException {
        try {
            InteressadoLocalHome            interessadoHome            =
(InteressadoLocalHome) lookup(JNDI.INTERESSADO);
            return
ListAssembler.montaColecaoVOs(interessadoHome.findByCasoDeUso(idCaso));
        } catch(Exception e) {
            throw new EJBException();
        }
    }

    /**************
     * QUESTAO EM ABERTO
     **************/

     /**
      * @ejb.interface-method
      */
    public VOQuestaoEmAberto addQuestaoEmAberto(VOQuestaoEmAberto dados)
throws EJBException {
        try {
            QuestaoEmAbertoLocalHome            questaoHome            =
(QuestaoEmAbertoLocalHome) lookup(JNDI.QUESTAO_EM_ABERTO);
            return questaoHome.create(dados).getDados();
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
```

```
      * @ejb.interface-method
      */
    public void editQuestaoEmAberto(VOQuestaoEmAberto dados) throws
EJBException {
        try {
            QuestaoEmAbertoLocalHome           questaoHome           =
(QuestaoEmAbertoLocalHome) lookup(JNDI.QUESTAO_EM_ABERTO);
            questaoHome.findByPrimaryKey(dados.getId()).setDados(dados);
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public void removeQuestaoEmAberto(Long id) throws EJBException {
        try {
            QuestaoEmAbertoLocalHome           questaoHome           =
(QuestaoEmAbertoLocalHome) lookup(JNDI.QUESTAO_EM_ABERTO);
            questaoHome.findByPrimaryKey(id).remove();
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public   Collection   getQuestoesEmAberto(Long   idCaso)   throws
EJBException {
        try {
            QuestaoEmAbertoLocalHome           questaoHome           =
(QuestaoEmAbertoLocalHome) lookup(JNDI.QUESTAO_EM_ABERTO);
            return
ListAssembler.montaColecaoVOs(questaoHome.findByCasoDeUso(idCaso));
        } catch(Exception e) {
            throw new EJBException();
        }
    }

    /**************
     * VARIACAO TECNOLOGICA
     **************/

     /**
      * @ejb.interface-method
      */
    public                                        VOVariacaoTecnologica
addVariacaoTecnologica(VOVariacaoTecnologica dados) throws EJBException
{
        try {
            VariacaoTecnologicaLocalHome         variacaoHome         =
(VariacaoTecnologicaLocalHome) lookup(JNDI.VARIACAO_TECNOLOGICA);
            return variacaoHome.create(dados).getDados();
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
```

```
      * @ejb.interface-method
      */
    public void editVariacaoTecnologica(VOVariacaoTecnologica dados)
throws EJBException {
        try {
            VariacaoTecnologicaLocalHome variacaoHome =
(VariacaoTecnologicaLocalHome) lookup(JNDI.VARIACAO_TECNOLOGICA);

variacaoHome.findByPrimaryKey(dados.getId()).setDados(dados);
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public void removeVariacaoTecnologica(Long id) throws EJBException {
        try {
            VariacaoTecnologicaLocalHome variacaoHome =
(VariacaoTecnologicaLocalHome) lookup(JNDI.VARIACAO_TECNOLOGICA);
            variacaoHome.findByPrimaryKey(id).remove();
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public Collection getVariacoesTecnologicas(Long idCaso) throws
EJBException {
        try {
            VariacaoTecnologicaLocalHome variacaoHome =
(VariacaoTecnologicaLocalHome) lookup(JNDI.VARIACAO_TECNOLOGICA);
            return
ListAssembler.montaColecaoVOs(variacaoHome.findByCasoDeUso(idCaso));
        } catch(Exception e) {
            throw new EJBException();
        }
    }

    /**************
     * PASSO
     **************/

     /**
      * @ejb.interface-method
      */
    public VOPasso addPasso(VOPasso dados) throws EJBException {
        try {
            FluxoFacadeHome fluxoHome = (FluxoFacadeHome)
lookup(JNDI.FLUXO_FACADE);
            FluxoFacade fluxo = fluxoHome.create();
            return fluxo.addPasso(dados);
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
```

```
       * @ejb.interface-method
       */
    public void editPasso(VOPasso dados) throws EJBException {
        try {
            FluxoFacadeHome      fluxoHome      =      (FluxoFacadeHome)
lookup(JNDI.FLUXO_FACADE);
            FluxoFacade     fluxo     = fluxoHome.create();
            fluxo.editPasso(dados);
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
       * @ejb.interface-method
       */
    public void removePasso(Long idPasso) throws EJBException {
        try {
            FluxoFacadeHome      fluxoHome      =      (FluxoFacadeHome)
lookup(JNDI.FLUXO_FACADE);
            FluxoFacade     fluxo     = fluxoHome.create();
            fluxo.removePasso(idPasso);
        } catch(Exception e) {
            throw new EJBException();
        }
    }

    /**************
     * FLUXO ALTERNATIVO
     **************/

     /**
       * @ejb.interface-method
       */
    public   VOFluxoAlternativo   addFluxoAlternativo(VOFluxoAlternativo
dados) throws EJBException {
        try {
            FluxoFacadeHome      fluxoHome      =      (FluxoFacadeHome)
lookup(JNDI.FLUXO_FACADE);
            FluxoFacade     fluxo     = fluxoHome.create();
            return fluxo.addFluxoAlternativo(dados);
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
       * @ejb.interface-method
       */
    public  void  editFluxoAlternativo(VOFluxoAlternativo  dados)  throws
EJBException {
        try {
            FluxoFacadeHome      fluxoHome      =      (FluxoFacadeHome)
lookup(JNDI.FLUXO_FACADE);
            FluxoFacade     fluxo     = fluxoHome.create();
            fluxo.editFluxoAlternativo(dados);
        } catch(Exception e) {
            throw new EJBException();
        }
    }
```

```
     /**
      * @ejb.interface-method
      */
    public void removeFluxoAlternativo(Long idFluxoAlternativo) throws
EJBException {
        try {
            FluxoFacadeHome      fluxoHome      =      (FluxoFacadeHome)
lookup(JNDI.FLUXO_FACADE);
            FluxoFacade     fluxo     = fluxoHome.create();
            fluxo.removeFluxoAlternativo(idFluxoAlternativo);
        } catch(Exception e) {
            throw new EJBException();
        }
    }

    /**************
     * FLUXO COMPLETO
     **************/

     /**
      * @ejb.interface-method
      */
    public   Collection   getCompleteFluxo(Long   idPassoInicial)   throws
EJBException {
        try
        {
            FluxoFacadeHome    fluxoHome           =    (FluxoFacadeHome)
lookup(JNDI.FLUXO_FACADE);
            FluxoFacade     fluxoFacade = fluxoHome.create();
            return fluxoFacade.getCompleteFluxo(idPassoInicial);
        } catch(Exception e) {
            throw new EJBException();
        }
    }

}
```

CategoriaDeRequisitoNaoFuncionalFacadeBean.java

```
package org.hopi.ejb.entity.facade;

import java.util.Collection;

import javax.ejb.CreateException;
import javax.ejb.EJBException;
import javax.ejb.SessionBean;

import org.hopi.ejb.base.SessionBase;
import org.hopi.ejb.entity.CategoriaLocalHome;
import org.hopi.ejb.entity.VOCategoria;
import org.hopi.ejb.util.JNDI;
import org.hopi.ejb.util.ListAssembler;

/**
 *
 * Fachada da Entidade de Dados Categoria de Requisito não Funcional.
 *
 *  @author  Rafael  Andreis  Pires  (pires@inf.ufsc.br)  e  Marcel  Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 * @ejb.bean
 *    name      = "CategoriaDeRequisitoNaoFuncionalFacade"
 *                                                  jndi-name            =
"org/hopi/facade/CategoriaDeRequisitoNaoFuncionalFacade"
 *    view-type = "remote"
 *      type      = "Stateless"
 */

public    class    CategoriaDeRequisitoNaoFuncionalFacadeBean    extends
SessionBase implements SessionBean {

    /**
     * @ejb.create-method
     */
    public void ejbCreate() throws CreateException {
    }

     /**
      * @ejb.interface-method
      */
    public   VOCategoria   createCategoria(VOCategoria   dados)   throws
EJBException {
        try {
            CategoriaLocalHome   categoriaHome   =   (CategoriaLocalHome)
lookup(JNDI.CATEGORIA);
            return categoriaHome.create(dados).getDados();
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public void setData(VOCategoria dados) throws EJBException {
        try {
```

```
            CategoriaLocalHome   categoriaHome   =   (CategoriaLocalHome)
lookup(JNDI.CATEGORIA);

categoriaHome.findByPrimaryKey(dados.getId()).setDados(dados);
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public void destroy(Long id) throws EJBException {
        try {
            CategoriaLocalHome   categoriaHome   =   (CategoriaLocalHome)
lookup(JNDI.CATEGORIA);
            categoriaHome.findByPrimaryKey(id).remove();
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public VOCategoria findById(Long id) throws EJBException {
        try {
            CategoriaLocalHome   categoriaHome   =   (CategoriaLocalHome)
lookup(JNDI.CATEGORIA);
            return categoriaHome.findByPrimaryKey(id).getDados();
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public Collection findAll() throws EJBException {
        try {
            CategoriaLocalHome   categoriaHome   =   (CategoriaLocalHome)
lookup(JNDI.CATEGORIA);
            return
ListAssembler.montaColecaoVOs(categoriaHome.findAll());
        } catch(Exception e) {
            e.printStackTrace();
            throw new EJBException();
        }
    }

}
```

EscritorFacadeBean.java

```
package org.hopi.ejb.entity.facade;

import java.util.ArrayList;
import java.util.Collection;
import java.util.Iterator;

import javax.ejb.CreateException;
import javax.ejb.EJBException;
import javax.ejb.FinderException;
import javax.ejb.SessionBean;

import org.hopi.ejb.base.SessionBase;
import org.hopi.ejb.entity.EscritorLocalHome;
import org.hopi.ejb.entity.ProjetoLocalHome;
import org.hopi.ejb.entity.UsuarioLocalHome;
import org.hopi.ejb.entity.VOEscritor;
import org.hopi.ejb.entity.VOProjeto;
import org.hopi.ejb.util.JNDI;
import org.hopi.ejb.util.ListAssembler;

/**
 *
 *  Fachada da Entidade de Dados Usuario com direito de escrita -
Projeto.
 *
 *  @author Rafael Andreis Pires (pires@inf.ufsc.br) e Marcel Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 * @ejb.bean
 *    name      = "EscritorFacade"
 *    jndi-name = "org/hopi/facade/EscritorFacade"
 *    view-type = "remote"
 *     type      = "Stateless"
 */

public   class   EscritorFacadeBean   extends   SessionBase   implements
SessionBean {

    /**
     * @ejb.create-method
     */
    public void ejbCreate() throws CreateException {
    }

     /**
      * @ejb.interface-method
      */
    public    VOEscritor    createEscritor(VOEscritor    dados)    throws
EJBException {
        try {
            EscritorLocalHome    escritorHome    =    (EscritorLocalHome)
lookup(JNDI.ESCRITOR);
            return escritorHome.create(dados).getDados();
        } catch(Exception e) {
            throw new EJBException();
        }
    }
```

```
     /**
      * @ejb.interface-method
      */
    public void removeEscritor(Long idEscritor) throws EJBException {
        try {
            EscritorLocalHome    escritorHome    =    (EscritorLocalHome)
lookup(JNDI.ESCRITOR);
            escritorHome.findByPrimaryKey(idEscritor).remove();
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public    Collection    findByIdProjeto(Long    idProjeto)    throws
EJBException {
        try {
            EscritorLocalHome    escritorHome    =  (EscritorLocalHome)
lookup(JNDI.ESCRITOR);
            UsuarioLocalHome    usuarioHome    =    (UsuarioLocalHome)
lookup(JNDI.USUARIO);
            ArrayList        usuarios    = new ArrayList();

            Iterator                          it                          =
ListAssembler.montaColecaoVOs(escritorHome.findByIdProjeto(idProjeto)).i
terator();
            VOEscritor vo = null;
            while ( it.hasNext() )
            {
                vo = (VOEscritor) it.next();

usuarios.add(usuarioHome.findByPrimaryKey(vo.getIdUsuario()).getDados())
;
            }
            return usuarios;
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public    Collection    findByIdUsuario(Long    idUsuario)    throws
EJBException {
        try {
            EscritorLocalHome    escritorHome    =  (EscritorLocalHome)
lookup(JNDI.ESCRITOR);
            ProjetoLocalHome    projetoHome    =    (ProjetoLocalHome)
lookup(JNDI.PROJETO);
            ArrayList        projetos    = new ArrayList();

            int i = 0;
            Iterator                          it                          =
ListAssembler.montaColecaoVOs(escritorHome.findByIdUsuario(idUsuario)).i
terator();
            VOEscritor vo = null;
            while ( it.hasNext() )
            {
```

```
                i++;
                vo = (VOEscritor) it.next();
                try {
                     VOProjeto                projeto                =
projetoHome.findByPrimaryKey(vo.getIdProjeto()).getDados();
                     projetos.add(projeto);
                } catch (FinderException fe) {
                    // Não adiciona projeto na lista
                }
            }
            return projetos;
        } catch(Exception e) {
            e.printStackTrace();
            throw new EJBException();
        }
    }

}
```

FluxoFacadeBean.java

```
package org.hopi.ejb.entity.facade;

import java.util.ArrayList;
import java.util.Collection;

import javax.ejb.CreateException;
import javax.ejb.EJBException;
import javax.ejb.FinderException;
import javax.ejb.SessionBean;

import org.hopi.ejb.base.SessionBase;
import org.hopi.ejb.entity.FluxoAlternativoLocalHome;
import org.hopi.ejb.entity.PassoLocalHome;
import org.hopi.ejb.entity.VOFluxoAlternativo;
import org.hopi.ejb.entity.VOPasso;
import org.hopi.ejb.util.JNDI;

/**
 *
 * Fachada das Entidades de Dados Passo e Fluxo Alternativo.
 *
 * @author Rafael Andreis Pires (pires@inf.ufsc.br) e Marcel Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 * @ejb.bean
 *    name      = "FluxoFacade"
 *    jndi-name = "org/hopi/facade/FluxoFacade"
 *    view-type = "remote"
 *     type      = "Stateless"
 */

public class FluxoFacadeBean extends SessionBase implements SessionBean
{

    /**
     * @ejb.create-method
     */
    public void ejbCreate() throws CreateException {
    }

     /**
      * @ejb.interface-method
      */
    public VOPasso addPasso(VOPasso dados) throws EJBException {
        try {
            PassoLocalHome      passoHome      =      (PassoLocalHome)
lookup(JNDI.PASSO);
            return passoHome.create(dados).getDados();
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public   VOFluxoAlternativo   addFluxoAlternativo(VOFluxoAlternativo
dados) throws EJBException {
```

```
        try {
            FluxoAlternativoLocalHome           fluxoAltHome           =
(FluxoAlternativoLocalHome) lookup(JNDI.FLUXO_ALTERNATIVO);
            return fluxoAltHome.create(dados).getDados();
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public void editPasso(VOPasso dados) throws EJBException {
        try {
            PassoLocalHome       passoHome       =       (PassoLocalHome)
lookup(JNDI.PASSO);
            passoHome.findByPrimaryKey(dados.getId()).setDados(dados);
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public  void  editFluxoAlternativo(VOFluxoAlternativo  dados)  throws
EJBException {
        try {
            FluxoAlternativoLocalHome           fluxoAltHome           =
(FluxoAlternativoLocalHome) lookup(JNDI.FLUXO_ALTERNATIVO);

fluxoAltHome.findByPrimaryKey(dados.getId()).setDados(dados);
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public void removePasso(Long idPasso) throws EJBException {
        try {
            PassoLocalHome       passoHome       =       (PassoLocalHome)
lookup(JNDI.PASSO);
            passoHome.findByPrimaryKey(idPasso).remove();
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public  void  removeFluxoAlternativo(Long  idFluxoAlternativo)  throws
EJBException {
        try {
            FluxoAlternativoLocalHome           fluxoAltHome           =
(FluxoAlternativoLocalHome) lookup(JNDI.FLUXO_ALTERNATIVO);
            fluxoAltHome.findByPrimaryKey(idFluxoAlternativo).remove();
        } catch(Exception e) {
            throw new EJBException();
```

```
        }
    }

     /**
      * @ejb.interface-method
      */
    public Collection getCompleteFluxo(Long idPassoInicial) throws
EJBException {
        ArrayList fluxo = new ArrayList();
        try {
            PassoLocalHome passoHome = (PassoLocalHome)
lookup(JNDI.PASSO);

            VOPasso vo =
passoHome.findByPrimaryKey(idPassoInicial).getDados();
            fluxo.add(vo);
            while ( true ) {
                vo = (VOPasso)
passoHome.findByPassoOrigem(vo.getId()).iterator().next();
                fluxo.add(vo);
            }
        } catch(FinderException fe) {
            return fluxo;
        } catch(Exception e) {
            throw new EJBException();
        }
    }

}
```

LeitorFacadeBean.java

```
package org.hopi.ejb.entity.facade;

import java.util.ArrayList;
import java.util.Collection;
import java.util.Iterator;

import javax.ejb.CreateException;
import javax.ejb.EJBException;
import javax.ejb.SessionBean;

import org.hopi.ejb.base.SessionBase;
import org.hopi.ejb.entity.LeitorLocalHome;
import org.hopi.ejb.entity.ProjetoLocalHome;
import org.hopi.ejb.entity.UsuarioLocalHome;
import org.hopi.ejb.entity.VOLeitor;
import org.hopi.ejb.util.JNDI;
import org.hopi.ejb.util.ListAssembler;

/**
 *
 * Fachada da Entidade de Dados Usuario com direito apenas de leitura -
Projeto.
 *
 *  @author  Rafael  Andreis  Pires  (pires@inf.ufsc.br)  e  Marcel  Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 * @ejb.bean
 *    name      = "LeitorFacade"
 *    jndi-name = "org/hopi/facade/LeitorFacade"
 *    view-type = "remote"
 *    type      = "Stateless"
 */

public class LeitorFacadeBean extends SessionBase implements SessionBean
{

    /**
     * @ejb.create-method
     */
    public void ejbCreate() throws CreateException {
    }

     /**
      * @ejb.interface-method
      */
    public VOLeitor createLeitor(VOLeitor dados) throws EJBException {
        try {
            LeitorLocalHome      leitorHome      =      (LeitorLocalHome)
lookup(JNDI.LEITOR);
            return leitorHome.create(dados).getDados();
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
```

```
    public void removeLeitor(Long idLeitor) throws EJBException {
        try {
            LeitorLocalHome      leitorHome      =      (LeitorLocalHome)
lookup(JNDI.LEITOR);
            leitorHome.findByPrimaryKey(idLeitor).remove();
        } catch(Exception e) {
            throw new EJBException();
        }
    }

      /**
       * @ejb.interface-method
       */
    public    Collection    findByIdProjeto(Long    idProjeto)    throws
EJBException {
        try {
            LeitorLocalHome       leitorHome       =    (LeitorLocalHome)
lookup(JNDI.LEITOR);
            UsuarioLocalHome     usuarioHome     =     (UsuarioLocalHome)
lookup(JNDI.USUARIO);
            ArrayList        usuarios    = new ArrayList();

            Iterator                          it                          =
ListAssembler.montaColecaoVOs(leitorHome.findByIdProjeto(idProjeto)).ite
rator();
            VOLeitor vo = null;
            while ( it.hasNext() )
            {
                vo = (VOLeitor) it.next();

usuarios.add(usuarioHome.findByPrimaryKey(vo.getIdUsuario()).getDados())
;
            }
            return usuarios;
        } catch(Exception e) {
            throw new EJBException();
        }
    }

      /**
       * @ejb.interface-method
       */
    public    Collection    findByIdUsuario(Long    idUsuario)    throws
EJBException {
        try {
            LeitorLocalHome       leitorHome       =    (LeitorLocalHome)
lookup(JNDI.LEITOR);
            ProjetoLocalHome     projetoHome     =     (ProjetoLocalHome)
lookup(JNDI.PROJETO);
            ArrayList        projetos    = new ArrayList();

            Iterator                          it                          =
ListAssembler.montaColecaoVOs(leitorHome.findByIdUsuario(idUsuario)).ite
rator();
            VOLeitor vo = null;
            while ( it.hasNext() )
            {
                vo = (VOLeitor) it.next();

projetos.add(projetoHome.findByPrimaryKey(vo.getIdUsuario()).getDados())
;
```

```
            }
            return projetos;
        } catch(Exception e) {
            throw new EJBException();
        }
    }

}
```

ProjetoFacadeBean.java

```
package org.hopi.ejb.entity.facade;

import java.util.Collection;

import javax.ejb.CreateException;
import javax.ejb.EJBException;
import javax.ejb.SessionBean;

import org.hopi.ejb.base.SessionBase;
import org.hopi.ejb.entity.AtorLocalHome;
import org.hopi.ejb.entity.ProjetoLocalHome;
import org.hopi.ejb.entity.RequisitoNaoFuncionalLocalHome;
import org.hopi.ejb.entity.VOAtor;
import org.hopi.ejb.entity.VOAtorDoCaso;
import org.hopi.ejb.entity.VOCasoDeUso;
import org.hopi.ejb.entity.VOCasoDoRequisito;
import org.hopi.ejb.entity.VOEscritor;
import org.hopi.ejb.entity.VOFluxoAlternativo;
import org.hopi.ejb.entity.VOInteressado;
import org.hopi.ejb.entity.VOLeitor;
import org.hopi.ejb.entity.VOPasso;
import org.hopi.ejb.entity.VOPosCondicao;
import org.hopi.ejb.entity.VOPreCondicao;
import org.hopi.ejb.entity.VOProjeto;
import org.hopi.ejb.entity.VOQuestaoEmAberto;
import org.hopi.ejb.entity.VORequisitoFuncional;
import org.hopi.ejb.entity.VORequisitoNaoFuncional;
import org.hopi.ejb.entity.VOVariacaoTecnologica;
import org.hopi.ejb.util.JNDI;
import org.hopi.ejb.util.ListAssembler;

/**
 *
 * Fachada da Entidade de Dados Projeto e seus atributos relacionados.
 *
 *  @author  Rafael  Andreis  Pires  (pires@inf.ufsc.br)  e  Marcel  Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 * @ejb.bean
 *    name      = "ProjetoFacade"
 *    jndi-name = "org/hopi/facade/ProjetoFacade"
 *    view-type = "remote"
 *      type      = "Stateless"
 */

public   class   ProjetoFacadeBean   extends   SessionBase   implements
SessionBean {

    /**
     * @ejb.create-method
     */
    public void ejbCreate() throws CreateException {
    }

      /**
       * @ejb.interface-method
       */
```

```
    public VOProjeto createProjeto(VOProjeto dados) throws EJBException
{
        try {
            ProjetoLocalHome     projetoHome     =     (ProjetoLocalHome)
lookup(JNDI.PROJETO);
            return projetoHome.create(dados).getDados();
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public VOProjeto findById(Long id) throws EJBException {
        try {
            ProjetoLocalHome     projetoHome     =     (ProjetoLocalHome)
lookup(JNDI.PROJETO);
            return projetoHome.findByPrimaryKey(id).getDados();
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public void setData(VOProjeto dados) throws EJBException {
        try {
            ProjetoLocalHome     projetoHome     =     (ProjetoLocalHome)
lookup(JNDI.PROJETO);
            projetoHome.findByPrimaryKey(dados.getId()).setDados(dados);
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public void destroy(Long idProjeto) throws EJBException {
        try {
            ProjetoLocalHome     projetoHome     =     (ProjetoLocalHome)
lookup(JNDI.PROJETO);
            projetoHome.findByPrimaryKey(idProjeto).remove();
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public Object add(Object dados) throws EJBException {
        try
        {
            if ( VOAtor.class.isInstance(dados) ) {
                AtorLocalHome      atorHome      =      (AtorLocalHome)
lookup(JNDI.ATOR);
                return atorHome.create((VOAtor)dados).getDados();
            }
```

```
            if ( VORequisitoFuncional.class.isInstance(dados) ) {
                RequisitoFuncionalFacadeHome       requisitoHome       =
(RequisitoFuncionalFacadeHome) lookup(JNDI.REQUISITO_FUNCIONAL_FACADE);
                RequisitoFuncionalFacade           requisito           =
requisitoHome.create();
                return
requisito.createRequisitoFuncional((VORequisitoFuncional)dados);
            }
            if ( VORequisitoNaoFuncional.class.isInstance(dados) ) {
                RequisitoFuncionalFacadeHome       requisitoHome       =
(RequisitoFuncionalFacadeHome) lookup(JNDI.REQUISITO_FUNCIONAL_FACADE);
                RequisitoFuncionalFacade           requisito           =
requisitoHome.create();
                return
requisito.addRequisitoNaoFuncional((VORequisitoNaoFuncional)dados);
            }
            if ( VOLeitor.class.isInstance(dados) ) {
                LeitorFacadeHome    leitorHome    =    (LeitorFacadeHome)
lookup(JNDI.LEITOR_FACADE);
                LeitorFacade     leitor = leitorHome.create();
                return leitor.createLeitor((VOLeitor)dados);
            }
            if ( VOEscritor.class.isInstance(dados) ) {
                EscritorFacadeHome  escritorHome  =  (EscritorFacadeHome)
lookup(JNDI.ESCRITOR_FACADE);
                EscritorFacade     escritor = escritorHome.create();
                return escritor.createEscritor((VOEscritor)dados);
            }

            CasoDeUsoFacadeHome    casoHome    =    (CasoDeUsoFacadeHome)
lookup(JNDI.CASO_DE_USO_FACADE);
            CasoDeUsoFacade     caso     = casoHome.create();
            if ( VOCasoDeUso.class.isInstance(dados) ) {
                return caso.createCasoDeUso((VOCasoDeUso)dados);
            }
            if ( VOPosCondicao.class.isInstance(dados) ) {
                return caso.addPosCondicao((VOPosCondicao)dados);
            }
            if ( VOPreCondicao.class.isInstance(dados) ) {
                return caso.addPreCondicao((VOPreCondicao)dados);
            }
            if ( VOInteressado.class.isInstance(dados) ) {
                return caso.addInteressado((VOInteressado)dados);
            }
            if ( VOQuestaoEmAberto.class.isInstance(dados) ) {
                return
caso.addQuestaoEmAberto((VOQuestaoEmAberto)dados);
            }
            if ( VOVariacaoTecnologica.class.isInstance(dados) ) {
                return
caso.addVariacaoTecnologica((VOVariacaoTecnologica)dados);
            }
            if ( VOPasso.class.isInstance(dados) ) {
                return caso.addPasso((VOPasso)dados);
            }
            if ( VOFluxoAlternativo.class.isInstance(dados) ) {
                return
caso.addFluxoAlternativo((VOFluxoAlternativo)dados);
            }

            throw new Exception("VO de tipo desconhecido");
```

```
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public void edit(Object dados) throws EJBException {
        try
        {
            if ( VOAtor.class.isInstance(dados) ) {
                AtorLocalHome      atorHome      =      (AtorLocalHome)
lookup(JNDI.ATOR);
                VOAtor vo = (VOAtor) dados;
                atorHome.findByPrimaryKey(vo.getId()).setDados(vo);
                return;
            }
            if ( VORequisitoFuncional.class.isInstance(dados) ) {
                RequisitoFuncionalFacadeHome       requisitoHome       =
(RequisitoFuncionalFacadeHome) lookup(JNDI.REQUISITO_FUNCIONAL_FACADE);
                RequisitoFuncionalFacade           requisito           =
requisitoHome.create();
                requisito.setData((VORequisitoFuncional)dados);
                return;
            }
            if ( VORequisitoNaoFuncional.class.isInstance(dados) ) {
                RequisitoFuncionalFacadeHome       requisitoHome       =
(RequisitoFuncionalFacadeHome) lookup(JNDI.REQUISITO_FUNCIONAL_FACADE);
                RequisitoFuncionalFacade           requisito           =
requisitoHome.create();

requisito.editRequisitoNaoFuncional((VORequisitoNaoFuncional)dados);
                return;
            }

            CasoDeUsoFacadeHome    casoHome    =    (CasoDeUsoFacadeHome)
lookup(JNDI.CASO_DE_USO_FACADE);
            CasoDeUsoFacade     caso     = casoHome.create();
            if ( VOCasoDeUso.class.isInstance(dados) ) {
                caso.setData((VOCasoDeUso)dados);
                return;
            }
            if ( VOPosCondicao.class.isInstance(dados) ) {
                caso.editPosCondicao((VOPosCondicao)dados);
                return;
            }
            if ( VOPreCondicao.class.isInstance(dados) ) {
                caso.editPreCondicao((VOPreCondicao)dados);
                return;
            }
            if ( VOInteressado.class.isInstance(dados) ) {
                caso.editInteressado((VOInteressado)dados);
                return;
            }
            if ( VOQuestaoEmAberto.class.isInstance(dados) ) {
                caso.editQuestaoEmAberto((VOQuestaoEmAberto)dados);
                return;
            }
            if ( VOVariacaoTecnologica.class.isInstance(dados) ) {
```

```
caso.editVariacaoTecnologica((VOVariacaoTecnologica)dados);
                return;
            }
            if ( VOPasso.class.isInstance(dados) ) {
                caso.editPasso((VOPasso)dados);
                return;
            }
            if ( VOFluxoAlternativo.class.isInstance(dados) ) {
                caso.editFluxoAlternativo((VOFluxoAlternativo)dados);
                return;
            }

            throw new Exception("VO de tipo desconhecido");
        } catch(Exception e) {
            e.printStackTrace();
            throw new EJBException();
        }
    }

      /**
       * @ejb.interface-method
       */
    public void remove(Object dados) throws EJBException {
        try
        {
            if ( VOAtor.class.isInstance(dados) ) {
                AtorLocalHome      atorHome      =      (AtorLocalHome)
lookup(JNDI.ATOR);
                VOAtor vo = (VOAtor) dados;
                atorHome.findByPrimaryKey(vo.getId()).remove();
                return;
            }
            if ( VORequisitoFuncional.class.isInstance(dados) ) {
                RequisitoFuncionalFacadeHome       requisitoHome       =
(RequisitoFuncionalFacadeHome) lookup(JNDI.REQUISITO_FUNCIONAL_FACADE);
                RequisitoFuncionalFacade           requisito           =
requisitoHome.create();

requisito.destroy(((VORequisitoFuncional)dados).getId());
                return;
            }
            if ( VORequisitoNaoFuncional.class.isInstance(dados) ) {
                RequisitoFuncionalFacadeHome       requisitoHome       =
(RequisitoFuncionalFacadeHome) lookup(JNDI.REQUISITO_FUNCIONAL_FACADE);
                RequisitoFuncionalFacade           requisito           =
requisitoHome.create();

requisito.removeRequisitoNaoFuncional(((VORequisitoNaoFuncional)dados).g
etId());
                return;
            }
            if ( VOLeitor.class.isInstance(dados) ) {
                LeitorFacadeHome    leitorHome    =    (LeitorFacadeHome)
lookup(JNDI.LEITOR_FACADE);
                LeitorFacade     leitor = leitorHome.create();
                leitor.removeLeitor(((VOLeitor)dados).getId());
                return;
            }
            if ( VOEscritor.class.isInstance(dados) ) {
```

```
                EscritorFacadeHome escritorHome = (EscritorFacadeHome)
lookup(JNDI.ESCRITOR_FACADE);
                EscritorFacade     escritor = escritorHome.create();
                escritor.removeEscritor(((VOEscritor)dados).getId());
                return;
            }

            CasoDeUsoFacadeHome    casoHome    =    (CasoDeUsoFacadeHome)
lookup(JNDI.CASO_DE_USO_FACADE);
            CasoDeUsoFacade     caso     = casoHome.create();
            if ( VOCasoDeUso.class.isInstance(dados) ) {
                caso.destroy(((VOCasoDeUso)dados).getId());
                return;
            }
            if ( VOPosCondicao.class.isInstance(dados) ) {
                caso.removePosCondicao(((VOPosCondicao)dados).getId());
                return;
            }
            if ( VOPreCondicao.class.isInstance(dados) ) {
                caso.removePreCondicao(((VOPreCondicao)dados).getId());
                return;
            }
            if ( VOInteressado.class.isInstance(dados) ) {
                caso.removeInteressado(((VOInteressado)dados).getId());
                return;
            }
            if ( VOQuestaoEmAberto.class.isInstance(dados) ) {

caso.removeQuestaoEmAberto(((VOQuestaoEmAberto)dados).getId());
                return;
            }
            if ( VOVariacaoTecnologica.class.isInstance(dados) ) {

caso.removeVariacaoTecnologica(((VOVariacaoTecnologica)dados).getId());
                return;
            }
            if ( VOPasso.class.isInstance(dados) ) {
                caso.removePasso(((VOPasso)dados).getId());
                return;
            }
            if ( VOFluxoAlternativo.class.isInstance(dados) ) {

caso.removeFluxoAlternativo(((VOFluxoAlternativo)dados).getId());
                return;
            }

            throw new Exception("VO de tipo desconhecido");
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public Collection getAllByRelatedId(Object dados, Long id) throws
EJBException {
        try
        {
            if ( VOAtor.class.isInstance(dados) ) {
```

```
                AtorLocalHome      atorHome      =      (AtorLocalHome)
lookup(JNDI.ATOR);
                return
ListAssembler.montaColecaoVOs(atorHome.findByIdProjeto(id));
            }
            if ( VORequisitoFuncional.class.isInstance(dados) ) {
                RequisitoFuncionalFacadeHome       requisitoHome       =
(RequisitoFuncionalFacadeHome) lookup(JNDI.REQUISITO_FUNCIONAL_FACADE);
                RequisitoFuncionalFacade           requisito           =
requisitoHome.create();
                return requisito.findByIdProjeto(id);
            }
            if ( VORequisitoNaoFuncional.class.isInstance(dados) ) {
                if                                                     (
((VORequisitoNaoFuncional)dados).idProjetoHasBeenSet() )
                {
                   RequisitoNaoFuncionalLocalHome        reqNao        =
(RequisitoNaoFuncionalLocalHome) lookup(JNDI.REQUISITO_NAO_FUNCIONAL);
                   return
ListAssembler.montaColecaoVOs(reqNao.findByProjeto(id));
                } else {
                   RequisitoFuncionalFacadeHome     requisitoHome     =
(RequisitoFuncionalFacadeHome) lookup(JNDI.REQUISITO_FUNCIONAL_FACADE);
                   RequisitoFuncionalFacade         requisito         =
requisitoHome.create();
                   return requisito.getRequisitosNaoFuncionais(id);
                }
            }
            if ( VOLeitor.class.isInstance(dados) ) {
                LeitorFacadeHome    leitorHome    =    (LeitorFacadeHome)
lookup(JNDI.LEITOR_FACADE);
                LeitorFacade     leitor = leitorHome.create();
                return leitor.findByIdProjeto(id);
            }
            if ( VOEscritor.class.isInstance(dados) ) {
                EscritorFacadeHome  escritorHome  =  (EscritorFacadeHome)
lookup(JNDI.ESCRITOR_FACADE);
                EscritorFacade     escritor = escritorHome.create();
                return escritor.findByIdProjeto(id);
            }

            CasoDeUsoFacadeHome    casoHome    =    (CasoDeUsoFacadeHome)
lookup(JNDI.CASO_DE_USO_FACADE);
            CasoDeUsoFacade     caso     = casoHome.create();
            if ( VOCasoDeUso.class.isInstance(dados) ) {
                return caso.getAllByIdProjeto(id);
            }
            if ( VOPosCondicao.class.isInstance(dados) ) {
                return caso.getPosCondicoes(id);
            }
            if ( VOPreCondicao.class.isInstance(dados) ) {
                return caso.getPreCondicoes(id);
            }
            if ( VOInteressado.class.isInstance(dados) ) {
                return caso.getInteressados(id);
            }
            if ( VOQuestaoEmAberto.class.isInstance(dados) ) {
                return caso.getQuestoesEmAberto(id);
            }
            if ( VOVariacaoTecnologica.class.isInstance(dados) ) {
                return caso.getVariacoesTecnologicas(id);
```

```
            }
            throw new Exception("VO de tipo desconhecido");
        } catch(Exception e) {
            e.printStackTrace();
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public Object getById(Object dados, Long id) throws EJBException {
        try
        {
            if ( VOAtor.class.isInstance(dados) ) {
                AtorLocalHome      atorHome      =      (AtorLocalHome)
lookup(JNDI.ATOR);
                VOAtor vo = atorHome.findByPrimaryKey(id).getDados();
                return vo;
            }
            if ( VORequisitoFuncional.class.isInstance(dados) ) {
                RequisitoFuncionalFacadeHome       requisitoHome       =
(RequisitoFuncionalFacadeHome) lookup(JNDI.REQUISITO_FUNCIONAL_FACADE);
                RequisitoFuncionalFacade           requisito           =
requisitoHome.create();
                return requisito.findById(id);
            }
            if ( VORequisitoNaoFuncional.class.isInstance(dados) ) {
                RequisitoFuncionalFacadeHome       requisitoHome       =
(RequisitoFuncionalFacadeHome) lookup(JNDI.REQUISITO_FUNCIONAL_FACADE);
                RequisitoFuncionalFacade           requisito           =
requisitoHome.create();
                return requisito.findRequisitoNaoFuncionalById(id);
            }


            CasoDeUsoFacadeHome    casoHome    =    (CasoDeUsoFacadeHome)
lookup(JNDI.CASO_DE_USO_FACADE);
            CasoDeUsoFacade     caso     = casoHome.create();
            if ( VOCasoDeUso.class.isInstance(dados) ) {
                return caso.findById(id);
            }
            if ( VOPosCondicao.class.isInstance(dados) ) {
                return caso.findPosCondicaoById(id);
            }
            if ( VOPreCondicao.class.isInstance(dados) ) {
                return caso.findPreCondicaoById(id);
            }
            if ( VOInteressado.class.isInstance(dados) ) {
                return caso.findInteressadoById(id);
            }
            if ( VOQuestaoEmAberto.class.isInstance(dados) ) {
                return caso.findQuestaoEmAbertoById(id);
            }
            if ( VOVariacaoTecnologica.class.isInstance(dados) ) {
                return caso.findVariacaoTecnologicaById(id);
            }

            throw new Exception("VO de tipo desconhecido");
        } catch(Exception e) {
            throw new EJBException();
```

```
        }


    }

      /**
       * @ejb.interface-method
       */
    public VOAtorDoCaso linkAtorAndCaso(VOAtorDoCaso dados) throws
EJBException {
        try {
            AtorDoCasoDeUsoFacadeHome            facadeHome            =
(AtorDoCasoDeUsoFacadeHome) lookup(JNDI.ATOR_DO_CASODEUSO_FACADE);
            AtorDoCasoDeUsoFacade     facade     = facadeHome.create();
            return facade.createAtorDoCasoDeUso(dados);
        } catch(Exception e) {
            throw new EJBException();
        }
    }

      /**
       * @ejb.interface-method
       */
    public void unlinkAtorAndCaso(Long idAtorDoCaso) throws EJBException
{
        try {
            AtorDoCasoDeUsoFacadeHome            facadeHome            =
(AtorDoCasoDeUsoFacadeHome) lookup(JNDI.ATOR_DO_CASODEUSO_FACADE);
            AtorDoCasoDeUsoFacade     facade     = facadeHome.create();
            facade.destroy(idAtorDoCaso);
        } catch(Exception e) {
            throw new EJBException();
        }
    }

      /**
       * @ejb.interface-method
       */
    public    VOCasoDoRequisito    linkCasoAndRequisito(VOCasoDoRequisito
dados) throws EJBException {
        try {
            CasoDoRequisitoFacadeHome            facadeHome            =
(CasoDoRequisitoFacadeHome) lookup(JNDI.CASO_DO_REQUISITO_FACADE);
            CasoDoRequisitoFacade     facade     = facadeHome.create();
            return facade.createCasoDoRequisito(dados);
        } catch(Exception e) {
            throw new EJBException();
        }
    }

      /**
       * @ejb.interface-method
       */
    public  void  unlinkCasoAndRequisito(Long  idCasoDoRequisito)  throws
EJBException {
        try {
            CasoDoRequisitoFacadeHome            facadeHome            =
(CasoDoRequisitoFacadeHome) lookup(JNDI.CASO_DO_REQUISITO_FACADE);
            CasoDoRequisitoFacade     facade     = facadeHome.create();
            facade.destroy(idCasoDoRequisito);
        } catch(Exception e) {
```

```
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public    void    getCompleteFluxo(Long    idPassoInicial)    throws
EJBException {
        try {
            FluxoFacadeHome      facadeHome      =      (FluxoFacadeHome)
lookup(JNDI.FLUXO_FACADE);
            FluxoFacade     facade     = facadeHome.create();
            facade.getCompleteFluxo(idPassoInicial);
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public   Collection   getAllAtoresDoCasoByIdCaso(Long   idCasoDeUso)
throws EJBException {
        try {
            AtorDoCasoDeUsoFacadeHome             atorHome             =
(AtorDoCasoDeUsoFacadeHome) lookup(JNDI.ATOR_DO_CASODEUSO_FACADE);
            AtorDoCasoDeUsoFacade ator = atorHome.create();
            return ator.findByIdCasoDeUso(idCasoDeUso);
        } catch (Exception e) {
            throw new EJBException(e);
        }
    }

     /**
      * @ejb.interface-method
      */
    public   Collection   getAllCasoDeUsoByIdAtor(Long   idAtor)   throws
EJBException {
        try {
            AtorDoCasoDeUsoFacadeHome             atorHome             =
(AtorDoCasoDeUsoFacadeHome) lookup(JNDI.ATOR_DO_CASODEUSO_FACADE);
            AtorDoCasoDeUsoFacade ator = atorHome.create();
            return ator.findByIdAtor(idAtor);
        } catch (Exception e) {
            throw new EJBException(e);
        }
    }

     /**
      * @ejb.interface-method
      */
    public   Collection   getAllRequisitosByIdCaso(Long   idCaso)   throws
EJBException {
        try {
            CasoDoRequisitoFacadeHome            facadeHome            =
(CasoDoRequisitoFacadeHome) lookup(JNDI.CASO_DO_REQUISITO_FACADE);
            CasoDoRequisitoFacade     facade     = facadeHome.create();
            return facade.findByIdCasoDeUso(idCaso);
        } catch (Exception e) {
            throw new EJBException(e);
```

```
        }
    }

     /**
      * @ejb.interface-method
      */
    public Collection getAllCasosByIdRequisito(Long idRequisito) throws
EJBException {
        try {
            CasoDoRequisitoFacadeHome            facadeHome            =
(CasoDoRequisitoFacadeHome) lookup(JNDI.CASO_DO_REQUISITO_FACADE);
            CasoDoRequisitoFacade     facade     = facadeHome.create();
            return facade.findByIdRequisito(idRequisito);
        } catch (Exception e) {
            throw new EJBException(e);
        }
    }

}
```

RequisitoFuncionalFacadeBean.java

```
package org.hopi.ejb.entity.facade;

import java.util.Collection;

import javax.ejb.CreateException;
import javax.ejb.EJBException;
import javax.ejb.SessionBean;

import org.hopi.ejb.base.SessionBase;
import org.hopi.ejb.entity.RequisitoFuncionalLocalHome;
import org.hopi.ejb.entity.VORequisitoFuncional;
import org.hopi.ejb.entity.VORequisitoNaoFuncional;
import org.hopi.ejb.util.JNDI;
import org.hopi.ejb.util.ListAssembler;

/**
 *
 * Fachada da Entidade de Dados RequisitoFuncional.
 *
 *  @author  Rafael  Andreis  Pires  (pires@inf.ufsc.br)  e  Marcel  Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 * @ejb.bean
 *    name      = "RequisitoFuncionalFacade"
 *    jndi-name = "org/hopi/facade/RequisitoFuncionalFacade"
 *    view-type = "remote"
 *      type      = "Stateless"
 */

public class RequisitoFuncionalFacadeBean extends SessionBase implements
SessionBean {

    /**
     * @ejb.create-method
     */
    public void ejbCreate() throws CreateException {
    }

      /**
       * @ejb.interface-method
       */
    public                                              VORequisitoFuncional
createRequisitoFuncional(VORequisitoFuncional dados) throws EJBException
{
        try {
            RequisitoFuncionalLocalHome         requisitoHome         =
(RequisitoFuncionalLocalHome) lookup(JNDI.REQUISITO_FUNCIONAL);
            return requisitoHome.create(dados).getDados();
        } catch(Exception e) {
            throw new EJBException();
        }
    }

      /**
       * @ejb.interface-method
       */
    public VORequisitoFuncional findById(Long id) throws EJBException {
        try {
```

```
            RequisitoFuncionalLocalHome          requisitoHome          =
(RequisitoFuncionalLocalHome) lookup(JNDI.REQUISITO_FUNCIONAL);
            return requisitoHome.findByPrimaryKey(id).getDados();
        } catch(Exception e) {
            throw new EJBException();
        }
    }


     /**
      * @ejb.interface-method
      */
    public void setData(VORequisitoFuncional dados) throws EJBException
{
        try {
            RequisitoFuncionalLocalHome          requisitoHome          =
(RequisitoFuncionalLocalHome) lookup(JNDI.REQUISITO_FUNCIONAL);

requisitoHome.findByPrimaryKey(dados.getId()).setDados(dados);
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public void destroy(Long id) throws EJBException {
        try {
            RequisitoFuncionalLocalHome          requisitoHome          =
(RequisitoFuncionalLocalHome) lookup(JNDI.REQUISITO_FUNCIONAL);
            requisitoHome.findByPrimaryKey(id).remove();
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public    Collection    findByIdProjeto(Long    idProjeto)    throws
EJBException {
        try {
            RequisitoFuncionalLocalHome          requisitoHome          =
(RequisitoFuncionalLocalHome) lookup(JNDI.REQUISITO_FUNCIONAL);
            return
ListAssembler.montaColecaoVOs(requisitoHome.findByProjeto(idProjeto));
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public                                          VORequisitoNaoFuncional
addRequisitoNaoFuncional(VORequisitoNaoFuncional      dados)      throws
EJBException {
        try
        {
```

```
            RequisitoNaoFuncionalFacadeHome      requisitoNaoHome      =
(RequisitoNaoFuncionalFacadeHome)
lookup(JNDI.REQUISITO_NAO_FUNCIONAL_FACADE);
            RequisitoNaoFuncionalFacade          requisitoNao          =
requisitoNaoHome.create();
            return requisitoNao.createRequisitoNaoFuncional(dados);
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public void editRequisitoNaoFuncional(VORequisitoNaoFuncional dados)
throws EJBException {
        try
        {
            RequisitoNaoFuncionalFacadeHome      requisitoNaoHome      =
(RequisitoNaoFuncionalFacadeHome)
lookup(JNDI.REQUISITO_NAO_FUNCIONAL_FACADE);
            RequisitoNaoFuncionalFacade          requisitoNao          =
requisitoNaoHome.create();
            requisitoNao.setData(dados);
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public             void             removeRequisitoNaoFuncional(Long
idRequisitoNaoFuncional) throws EJBException {
        try
        {
            RequisitoNaoFuncionalFacadeHome      requisitoNaoHome      =
(RequisitoNaoFuncionalFacadeHome)
lookup(JNDI.REQUISITO_NAO_FUNCIONAL_FACADE);
            RequisitoNaoFuncionalFacade          requisitoNao          =
requisitoNaoHome.create();
            requisitoNao.destroy(idRequisitoNaoFuncional);
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public   Collection   getRequisitosNaoFuncionais(Long   idRequisito)
throws EJBException {
        try
        {
            RequisitoNaoFuncionalFacadeHome      requisitoNaoHome      =
(RequisitoNaoFuncionalFacadeHome)
lookup(JNDI.REQUISITO_NAO_FUNCIONAL_FACADE);
            RequisitoNaoFuncionalFacade          requisitoNao          =
requisitoNaoHome.create();
            return requisitoNao.findByIdRequisito(idRequisito);
        } catch(Exception e) {
```

```
            throw new EJBException();
        }
    }

      /**
       * @ejb.interface-method
       */
    public   VORequisitoNaoFuncional   findRequisitoNaoFuncionalById(Long
idRequisitoNaoFuncional) throws EJBException {
        try
        {
            RequisitoNaoFuncionalFacadeHome      requisitoNaoHome      =
(RequisitoNaoFuncionalFacadeHome)
lookup(JNDI.REQUISITO_NAO_FUNCIONAL_FACADE);
            RequisitoNaoFuncionalFacade          requisitoNao          =
requisitoNaoHome.create();
            return requisitoNao.findById(idRequisitoNaoFuncional);
        } catch(Exception e) {
            throw new EJBException();
        }
    }

}
```

RequisitoNaoFuncionaFacadeBean.java

```
package org.hopi.ejb.entity.facade;

import java.util.Collection;

import javax.ejb.CreateException;
import javax.ejb.EJBException;
import javax.ejb.SessionBean;

import org.hopi.ejb.base.SessionBase;
import org.hopi.ejb.entity.RequisitoNaoFuncionalLocalHome;
import org.hopi.ejb.entity.VORequisitoNaoFuncional;
import org.hopi.ejb.util.JNDI;
import org.hopi.ejb.util.ListAssembler;

/**
 *
 * Fachada da Entidade de Dados Requisito não Funcional.
 *
 *  @author  Rafael  Andreis  Pires  (pires@inf.ufsc.br)  e  Marcel  Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 * @ejb.bean
 *    name      = "RequisitoNaoFuncionalFacade"
 *    jndi-name = "org/hopi/facade/RequisitoNaoFuncionalFacade"
 *    view-type = "remote"
 *      type      = "Stateless"
 */

public   class   RequisitoNaoFuncionalFacadeBean   extends   SessionBase
implements SessionBean {

    /**
     * @ejb.create-method
     */
    public void ejbCreate() throws CreateException {
    }

      /**
       * @ejb.interface-method
       */
    public                                          VORequisitoNaoFuncional
createRequisitoNaoFuncional(VORequisitoNaoFuncional    dados)    throws
EJBException {
        try {
            RequisitoNaoFuncionalLocalHome      requisitoNaoHome      =
(RequisitoNaoFuncionalLocalHome) lookup(JNDI.REQUISITO_NAO_FUNCIONAL);
            return requisitoNaoHome.create(dados).getDados();
        } catch(Exception e) {
            throw new EJBException();
        }
    }

      /**
       * @ejb.interface-method
       */
    public    void    setData(VORequisitoNaoFuncional    dados)    throws
EJBException {
        try {
```

```
            RequisitoNaoFuncionalLocalHome      requisitoNaoHome      =
(RequisitoNaoFuncionalLocalHome) lookup(JNDI.REQUISITO_NAO_FUNCIONAL);

requisitoNaoHome.findByPrimaryKey(dados.getId()).setDados(dados);
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public void destroy(Long id) throws EJBException {
        try {
            RequisitoNaoFuncionalLocalHome      requisitoNaoHome      =
(RequisitoNaoFuncionalLocalHome) lookup(JNDI.REQUISITO_NAO_FUNCIONAL);
            requisitoNaoHome.findByPrimaryKey(id).remove();
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public   Collection   findByIdRequisito(Long   idRequisito)   throws
EJBException {
        try {
            RequisitoNaoFuncionalLocalHome      requisitoNaoHome      =
(RequisitoNaoFuncionalLocalHome) lookup(JNDI.REQUISITO_NAO_FUNCIONAL);
            return
ListAssembler.montaColecaoVOs(requisitoNaoHome.findByRequisitoFuncional(
idRequisito));
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public VORequisitoNaoFuncional findById(Long id) throws EJBException
{
        try {
            RequisitoNaoFuncionalLocalHome      requisitoNaoHome      =
(RequisitoNaoFuncionalLocalHome) lookup(JNDI.REQUISITO_NAO_FUNCIONAL);
            return requisitoNaoHome.findByPrimaryKey(id).getDados();
        } catch(Exception e) {
            throw new EJBException(e);
        }
    }

}
```

UsuarioFacadeBean.java

```
package org.hopi.ejb.entity.facade;

import java.util.Collection;

import javax.ejb.CreateException;
import javax.ejb.EJBException;
import javax.ejb.FinderException;
import javax.ejb.SessionBean;

import org.hopi.ejb.base.SessionBase;
import org.hopi.ejb.entity.UsuarioLocalHome;
import org.hopi.ejb.entity.VOUsuario;
import org.hopi.ejb.util.JNDI;
import org.hopi.ejb.util.ListAssembler;

/**
 *
 * Fachada da Entidade de Dados Usuario.
 *
 *  @author  Rafael  Andreis  Pires  (pires@inf.ufsc.br)  e  Marcel  Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 * @ejb.bean
 *    name      = "UsuarioFacade"
 *    jndi-name = "org/hopi/facade/UsuarioFacade"
 *    view-type = "remote"
 *      type      = "Stateless"
 */

public   class   UsuarioFacadeBean   extends   SessionBase   implements
SessionBean {

    /**
     * @ejb.create-method
     */
    public void ejbCreate() throws CreateException {
    }

     /**
      * @ejb.interface-method
      */
    public VOUsuario createUsuario(VOUsuario dados) throws EJBException
{
        try {
            UsuarioLocalHome     usuarioHome     =     (UsuarioLocalHome)
lookup(JNDI.USUARIO);
            return usuarioHome.create(dados).getDados();
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public void setData(VOUsuario dados) throws EJBException {
        try {
```

```
            UsuarioLocalHome     usuarioHome     =     (UsuarioLocalHome)
lookup(JNDI.USUARIO);
            usuarioHome.findByPrimaryKey(dados.getId()).setDados(dados);
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public void destroy(Long id) throws EJBException {
        try {
            UsuarioLocalHome     usuarioHome     =     (UsuarioLocalHome)
lookup(JNDI.USUARIO);
            usuarioHome.findByPrimaryKey(id).remove();
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public    VOUsuario    findByUsername(String    username)    throws
FinderException, EJBException {
        try {
            UsuarioLocalHome     usuarioHome     =     (UsuarioLocalHome)
lookup(JNDI.USUARIO);
            return usuarioHome.findByUsername(username).getDados();
        } catch(FinderException fe) {
            throw new FinderException();
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public   Collection   findProjetosAsLeitor(Long   idUsuario)   throws
EJBException {
        try {
            LeitorFacadeHome     leitorHome     =     (LeitorFacadeHome)
lookup(JNDI.LEITOR_FACADE);
            LeitorFacade     leitor     = leitorHome.create();
            return leitor.findByIdUsuario(idUsuario);
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public  Collection  findProjetosAsEscritor(Long  idUsuario)  throws
EJBException {
        try {
            EscritorFacadeHome   escritorHome   =   (EscritorFacadeHome)
lookup(JNDI.ESCRITOR_FACADE);
            EscritorFacade     escritor     = escritorHome.create();
```

```
            return escritor.findByIdUsuario(idUsuario);
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public Collection findAll() throws EJBException {
        try {
            UsuarioLocalHome     usuarioHome     =     (UsuarioLocalHome)
lookup(JNDI.USUARIO);
            return ListAssembler.montaColecaoVOs(usuarioHome.findAll());
        } catch(Exception e) {
            e.printStackTrace();
            throw new EJBException();
        }
    }

}
```

PACOTE ORG.HOPI.EJB.SESSION

UsuarioBusinessFacadeBean.java

```
package org.hopi.ejb.session;

import java.util.Collection;

import javax.ejb.EJBException;
import javax.ejb.FinderException;
import javax.ejb.SessionBean;

import org.hopi.ejb.base.SessionBase;
import org.hopi.ejb.entity.VOUsuario;
import org.hopi.ejb.entity.facade.UsuarioFacade;
import org.hopi.ejb.entity.facade.UsuarioFacadeHome;
import org.hopi.ejb.util.JNDI;
import org.hopi.ejb.util.ListAssembler;
import org.hopi.exception.LoginInvalidoException;

/**
 *
 * Classe de negócio que gerencia as informações relativas a Usuarios.
 *
 *  @author  Rafael  Andreis  Pires  (pires@inf.ufsc.br)  e  Marcel  Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 * @ejb.bean
 *    name      = "UsuarioBusinessFacade"
 *    jndi-name = "org/hopi/facade/UsuarioBusinessFacade"
 *    view-type = "remote"
 *      type      = "Stateless"
 *
 */

public  class  UsuarioBusinessFacadeBean  extends  SessionBase  implements
SessionBean {

    /**
     * @ejb.create-method
     */
    public void ejbCreate() throws EJBException {
    }

     /**
      * @ejb.interface-method
      */
    public VOUsuario createUsuario(VOUsuario dados) throws EJBException
{
        try {
            UsuarioFacadeHome    usuarioHome    =    (UsuarioFacadeHome)
lookup(JNDI.USUARIO_FACADE);
            UsuarioFacade     facade      = usuarioHome.create();
            return facade.createUsuario(dados);
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
```

```
      * @ejb.interface-method
      */
    public void editUsuario(VOUsuario dados) throws EJBException {
        try {
            UsuarioFacadeHome    usuarioHome    =    (UsuarioFacadeHome)
lookup(JNDI.USUARIO_FACADE);
            UsuarioFacade     facade      = usuarioHome.create();
            facade.setData(dados);
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public void removeUsuario(Long id) throws EJBException {
        try {
            UsuarioFacadeHome    usuarioHome    =    (UsuarioFacadeHome)
lookup(JNDI.USUARIO_FACADE);
            UsuarioFacade     facade      = usuarioHome.create();
            facade.destroy(id);
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public     VOUsuario     validarLogin(VOUsuario     dados)     throws
LoginInvalidoException, EJBException {
        String passwordReal;
        String nome = null;
        VOUsuario vo = new VOUsuario();
        try {
            UsuarioFacadeHome    usuarioHome    =    (UsuarioFacadeHome)
lookup(JNDI.USUARIO_FACADE);
            UsuarioFacade     facade      = usuarioHome.create();
            vo = facade.findByUsername(dados.getUsername());
            passwordReal = vo.getPassword();
        } catch(FinderException e) {
            throw new LoginInvalidoException();
        } catch(Exception e) {
            throw new EJBException();
        }
        if ( dados.getPassword().compareTo(passwordReal) != 0 ) {
            throw new LoginInvalidoException();
        }
        return vo;
    }

     /**
      * @ejb.interface-method
      */
    public Collection findAll() throws EJBException {
        try {
            UsuarioFacadeHome    usuarioHome    =    (UsuarioFacadeHome)
lookup(JNDI.USUARIO_FACADE);
            UsuarioFacade     facade      = usuarioHome.create();
            return ListAssembler.montaColecaoVOs(facade.findAll());
```

```
        } catch(Exception e) {
            throw new EJBException();
        }
    }

     /**
      * @ejb.interface-method
      */
    public Collection getProjetosAsLeitor(Long idUsuario) throws
EJBException
    {
        try {
            UsuarioFacadeHome usuarioHome = (UsuarioFacadeHome)
lookup(JNDI.USUARIO_FACADE);
            UsuarioFacade usuarioFacade = usuarioHome.create();
            return usuarioFacade.findProjetosAsLeitor(idUsuario);
        } catch (Exception e) {
            throw new EJBException(e);
        }
    }

     /**
      * @ejb.interface-method
      */
    public Collection getProjetosAsEscritor(Long idUsuario) throws
EJBException
    {
        try {
            UsuarioFacadeHome usuarioHome = (UsuarioFacadeHome)
lookup(JNDI.USUARIO_FACADE);
            UsuarioFacade usuarioFacade = usuarioHome.create();
            return usuarioFacade.findProjetosAsEscritor(idUsuario);
        } catch (Exception e) {
            throw new EJBException(e);
        }
    }
}
```

PACOTE ORG.HOPI.EJB.UTIL

GeradorChaveUtil.java

```
package org.hopi.ejb.util;

import javax.ejb.CreateException;
import javax.ejb.EJBException;
import javax.naming.NamingException;

import org.hopi.ejb.entity.GeradorChaveLocalHome;

/**
 *
 * Classe que abstrai a criação de novas chaves primárias.
 *
 *  @author  Rafael  Andreis  Pires  (pires@inf.ufsc.br)  e  Marcel  Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 */

public class GeradorChaveUtil {

     public   static   Long   proximoValor(String   entidade)   throws
CreateException {
           GeradorChaveLocalHome home;
           try {
                 home             =             (GeradorChaveLocalHome)
ServiceLocator.getLocalHome(JNDI.GERADOR_CHAVE);
           }
           catch (NamingException e) {
                 throw new EJBException(e);
           }
           return home.proximoValor(entidade);
     }

 }
```

JNDI.java

```
package org.hopi.ejb.util;

/**
 *
 * Centraliza os JNDIs para que possam ser chamados globalmente de forma
estática.
 *
 * @author Rafael Andreis Pires (pires@inf.ufsc.br) e Marcel Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 */

public interface JNDI {

    public static final String entityLocalHome = "org/hopi/";

    public static String ATOR                       = entityLocalHome +
"Ator";
    public static String CASO_DE_USO                = entityLocalHome +
"CasoDeUso";
    public static String CATEGORIA                  = entityLocalHome +
"Categoria";
    public static String FLUXO_ALTERNATIVO          = entityLocalHome +
"FluxoAlternativo";
    public static String GERADOR_CHAVE              = entityLocalHome +
"GeradorChave";
    public static String INTERESSADO                = entityLocalHome +
"Interessado";
    public static String PASSO                      = entityLocalHome +
"Passo";
    public static String POS_CONDICAO               = entityLocalHome +
"PosCondicao";
    public static String PRE_CONDICAO               = entityLocalHome +
"PreCondicao";
    public static String PROJETO                    = entityLocalHome +
"Projeto";
    public static String QUESTAO_EM_ABERTO          = entityLocalHome +
"QuestaoEmAberto";
    public static String REQUISITO_FUNCIONAL        = entityLocalHome +
"RequisitoFuncional";
    public static String REQUISITO_NAO_FUNCIONAL  = entityLocalHome  +
"RequisitoNaoFuncional";
    public static String USUARIO                    = entityLocalHome +
"Usuario";
    public static String VARIACAO_TECNOLOGICA       = entityLocalHome +
"VariacaoTecnologica";
    public static String LEITOR                     = entityLocalHome +
"Leitor";
    public static String ESCRITOR                   = entityLocalHome +
"Escritor";
    public static String CASO_DO_REQUISITO          = entityLocalHome +
"CasoDoRequisito";
    public static String ATOR_DO_CASO               = entityLocalHome +
"AtorDoCaso";

    public static String entityFacade = entityLocalHome + "facade/";
```

```
    public static String USUARIO_FACADE                 = entityFacade +
"UsuarioFacade";
    public static String LEITOR_FACADE                  = entityFacade +
"LeitorFacade";
    public static String ESCRITOR_FACADE                = entityFacade +
"EscritorFacade";
    public static String PROJETO_FACADE                 = entityFacade +
"ProjetoFacade";
    public static String ATOR_DO_CASODEUSO_FACADE       = entityFacade +
"AtorDoCasoDeUsoFacade";
    public static String CASO_DE_USO_FACADE             = entityFacade +
"CasoDeUsoFacade";
    public static String FLUXO_FACADE                   = entityFacade +
"FluxoFacade";
    public static String CASO_DO_REQUISITO_FACADE       = entityFacade +
"CasoDoRequisitoFacade";
    public static String REQUISITO_FUNCIONAL_FACADE     = entityFacade +
"RequisitoFuncionalFacade";
    public static String REQUISITO_NAO_FUNCIONAL_FACADE = entityFacade +
"RequisitoNaoFuncionalFacade";
    public static String CATEGORIA_FACADE               = entityFacade +
"CategoriaDeRequisitoNaoFuncionalFacade";

    public static String businessFacade = "org/hopi/facade/";

    public  static  String  BUSINESS_FACADE_USUARIO  =  businessFacade  +
"UsuarioBusinessFacade";

}
```

ListaAssembler.java

```
package org.hopi.ejb.util;

import java.lang.reflect.Method;
import java.util.ArrayList;
import java.util.Collection;
import java.util.Iterator;

import javax.ejb.EJBException;
import javax.ejb.EJBLocalObject;

/**
 *
 * Esta classe tem a função de montar a lista de VOs de uma dada coleção
de EJBs.
 *
 * @author Rafael Andreis Pires (pires@inf.ufsc.br) e Marcel Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 */

public class ListAssembler {

     public static Collection montaColecaoVOs(Collection colecaoEjbs,
String getData) {
           Collection colecaoVOs = new ArrayList();
           Method getVO = null;
           Iterator it = colecaoEjbs.iterator();
           Class[] arrayClasses = new Class[0];
           Object[] arrayObjetos = new Object[0];
           try {
                 while (it.hasNext()) {
                       EJBLocalObject ejb = (EJBLocalObject)it.next();
                       if (getVO == null)
                             getVO  =  ejb.getClass().getMethod(getData,
arrayClasses);
                       Object vo = getVO.invoke(ejb, arrayObjetos);
                       colecaoVOs. add(vo);
                 }
           }
           catch (Exception e) {
                 throw new EJBException(e);
           }
           return colecaoVOs;
     }

     public static Collection montaColecaoVOs(Collection colecaoEjbs) {
           return montaColecaoVOs(colecaoEjbs, "getData");
     }

}
```

ServiceLocator.java

```
package org.hopi.ejb.util;

import java.util.HashMap;
import java.util.Map;

import javax.naming.InitialContext;
import javax.naming.NamingException;

/**
 *
 * Esta classe implementa o design pattern "service locator":
 * Um cache de referências a interfaces home na JVM
 *
 *  @author  Rafael  Andreis  Pires  (pires@inf.ufsc.br)  e  Marcel  Horner
(mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 */

public class ServiceLocator {

      private static InitialContext ctx = null;
      private static Map cacheLocal  = new HashMap();
      private static Map cacheRemoto = new HashMap();

      public    static    InitialContext    getInitialContext()    throws
NamingException {
            if (ctx == null) {
                  ctx = new InitialContext();
            }
            return ctx;
      }

      public    static    Object    getRemoteHome(String    jndi)    throws
NamingException {
            InitialContext ctx = getInitialContext();
            return ctx.lookup(jndi);
      }

      public    static    Object    getLocalHome(String    jndi)    throws
NamingException {
            InitialContext ctx = getInitialContext();
            return ctx.lookup(jndi);
      }

}
```

PACOTE ORG.HOPI.EXCEPTION

LoginInvalidoException.java

```
package org.hopi.exception;

/**
 *
 * Exceção a ser gerado caso o username/password informados no login sejam incorretos.
 *
 * @author Rafael Andreis Pires (pires@inf.ufsc.br) e Marcel Horner (mhorner@inf.ufsc.br)
 * @since 15/Out/2004
 *
 */

public class LoginInvalidoException extends Exception {

    public LoginInvalidoException() {
        super();
    }

    public LoginInvalidoException(String msg) {
        super(msg);
    }

}
```

FOLDER WEBAPP/WEB-INF

struts-config.xml

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

<!DOCTYPE struts-config PUBLIC
          "-//Apache   Software   Foundation//DTD   Struts   Configuration
1.1//EN"
          "http://jakarta.apache.org/struts/dtds/struts-config_1_1.dtd">

<struts-config>
      <form-beans>
            <form-bean name="ator"
type="org.hopi.web.struts.form.AtorForm"/>
            <form-bean name="casoDeUso"
type="org.hopi.web.struts.form.CasoDeUsoForm"/>
            <form-bean name="categoria"
type="org.hopi.web.struts.form.CategoriaForm"/>
            <form-bean name="login"
type="org.hopi.web.struts.form.LoginForm"/>
            <form-bean name="projeto"
type="org.hopi.web.struts.form.ProjetoForm"/>
            <form-bean name="requisitoFuncional"
type="org.hopi.web.struts.form.RequisitoFuncionalForm"/>
            <form-bean                          name="requisitoNaoFuncional"
type="org.hopi.web.struts.form.RequisitoNaoFuncionalForm"/>
            <form-bean name="usuario"
type="org.hopi.web.struts.form.UsuarioForm"/>
      </form-beans>

    <action-mappings>

      <!--### ACESSO NÃO AUTORIZADO #########################-->
            <action
                  path="/acessoNaoAutorizado"

      type="org.hopi.web.struts.action.AcessoNaoAutorizadoAction">
                  <forward name="ok" path="/login.jsp"/>
            </action>

      <!--### LOGIN ##############################################-->
            <action path="/login"
                        type="org.hopi.web.struts.action.LoginAction"
                        name="login"
                        scope="request"
                        input="/login.jsp">
                  <forward name="ok" path="/principal.do"/>
                  <forward name="erro" path="/login.jsp"/>
            </action>

      <!--### PRINCIPAL ##########################################-->
            <action
                  path="/principal"
                  type="org.hopi.web.struts.action.PrincipalAction">
                  <forward name="ok" path="/principal.jsp"/>
            </action>

      <!--### LOGOUT ##############################################-->
            <action
                  path="/logout"
```

```
            type="org.hopi.web.struts.action.LogoutAction">
            <forward name="logout" path="/logout.jsp"/>
      </action>

<!--### USUÁRIOS ##########################################-->
      <action
            path="/listeUsuarios"
            type="org.hopi.web.struts.action.ListeUsuariosAction">
            <forward name="ok" path="/listeUsuarios.jsp"/>
      </action>
      <action
            path="/abraUsuario"
            type="org.hopi.web.struts.action.AbraUsuarioAction">
            <forward name="ok" path="/usuario.jsp"/>
      </action>
      <action
            path="/novoUsuario"
            type="org.hopi.web.struts.action.NovoUsuarioAction">
            <forward name="ok" path="/usuario.jsp"/>
      </action>
      <action path="/salveUsuario"
                  name="usuario"
                  scope="request"

type="org.hopi.web.struts.action.SalveUsuarioAction"
                  input="/usuario.jsp">
            <forward name="ok" path="/listeUsuarios.do"/>
      </action>
      <action
            path="/excluaUsuario"
            type="org.hopi.web.struts.action.ExcluaUsuarioAction">
            <forward name="ok" path="/listeUsuarios.do"/>
      </action>

<!--### CATEGORIAS ########################################-->
      <action
            path="/listeCategorias"

type="org.hopi.web.struts.action.ListeCategoriasAction">
            <forward name="ok" path="/listeCategorias.jsp"/>
      </action>
      <action
            path="/abraCategoria"
            type="org.hopi.web.struts.action.AbraCategoriaAction">
            <forward name="ok" path="/categoria.jsp"/>
      </action>
      <action
            path="/novaCategoria"
            type="org.hopi.web.struts.action.NovaCategoriaAction">
            <forward name="ok" path="/categoria.jsp"/>
      </action>
      <action path="/salveCategoria"
                  name="categoria"
                  scope="request"

type="org.hopi.web.struts.action.SalveCategoriaAction"
                  input="/categoria.jsp">
            <forward name="ok" path="/listeCategorias.do"/>
      </action>
      <action
            path="/excluaCategoria"
```

```
type="org.hopi.web.struts.action.ExcluaCategoriaAction">
            <forward name="ok" path="/listeCategorias.do"/>
      </action>

<!--### PROJETOS ##########################################-->
      <action
            path="/abraProjeto"
            type="org.hopi.web.struts.action.AbraProjetoAction">
            <forward name="ok" path="/projeto.jsp"/>
      </action>
      <action
            path="/novoProjeto"
            type="org.hopi.web.struts.action.NovoProjetoAction">
            <forward name="ok" path="/projeto.jsp"/>
      </action>
      <action
            path="/excluaProjeto"
            type="org.hopi.web.struts.action.ExcluaProjetoAction">
            <forward name="ok" path="/principal.do"/>
      </action>
      <action path="/salveProjeto"
                  name="projeto"
                  scope="request"

type="org.hopi.web.struts.action.SalveProjetoAction"
                  input="/projeto.jsp">
            <forward name="existente" path="/principal.do"/>
            <forward name="novo" path="/projeto.jsp"/>
      </action>

<!--### ATORES ############################################-->
      <action
            path="/abraAtor"
            type="org.hopi.web.struts.action.AbraAtorAction">
            <forward name="ok" path="/ator.jsp"/>
      </action>
      <action
            path="/excluaAtor"
            type="org.hopi.web.struts.action.ExcluaAtorAction">
            <forward name="ok" path="/abraProjeto.do"/>
      </action>
      <action path="/salveAtor"
                  name="ator"
                  scope="request"
                  type="org.hopi.web.struts.action.SalveAtorAction"
                  input="/ator.jsp">
            <forward name="ok" path="/abraProjeto.do"/>
      </action>
      <action
            path="/novoAtor"
            type="org.hopi.web.struts.action.NovoAtorAction">
            <forward name="ok" path="/ator.jsp"/>
      </action>

<!--### REQUISITOS FUNCIONAIS ##############################-->
      <action
            path="/abraRequisitoFuncional"

type="org.hopi.web.struts.action.AbraRequisitoFuncionalAction">
            <forward name="ok" path="/requisitoFuncional.jsp"/>
```

```
            </action>
            <action
                  path="/excluaRequisitoFuncional"

      type="org.hopi.web.struts.action.ExcluaRequisitoFuncionalAction">
                  <forward name="ok" path="/abraProjeto.do"/>
            </action>
            <action path="/salveRequisitoFuncional"
                        name="requisitoFuncional"
                        scope="request"

      type="org.hopi.web.struts.action.SalveRequisitoFuncionalAction"
                        input="/requisitoFuncional.jsp">
                  <forward name="existente" path="/abraProjeto.do"/>
                  <forward                                        name="novo"
path="/abraRequisitoFuncional.do"/>
            </action>
            <action
                  path="/novoRequisitoFuncional"

      type="org.hopi.web.struts.action.NovoRequisitoFuncionalAction">
                  <forward name="ok" path="/requisitoFuncional.jsp"/>
            </action>

      <!--### REQUISITOS NÃO-FUNCIONAIS ##########################-->
            <action
                  path="/abraRequisitoNaoFuncional"

      type="org.hopi.web.struts.action.AbraRequisitoNaoFuncionalAction">
                  <forward name="ok" path="/requisitoNaoFuncional.jsp"/>
            </action>
            <action
                  path="/novoRequisitoNaoFuncional"

      type="org.hopi.web.struts.action.NovoRequisitoNaoFuncionalAction">
                  <forward name="ok" path="/requisitoNaoFuncional.jsp"/>
            </action>
            <action path="/salveRequisitoNaoFuncional"
                        name="requisitoNaoFuncional"
                        scope="request"

      type="org.hopi.web.struts.action.SalveRequisitoNaoFuncionalAction"
                        input="/requisitoNaoFuncional.jsp">
                  <forward                                          name="ok"
path="/abraRequisitoNaoFuncional.do"/>
            </action>
            <action
                  path="/excluaRequisitoNaoFuncional"

      type="org.hopi.web.struts.action.ExcluaRequisitoNaoFuncionalAction
">
                  <forward name="ok" path="/abraProjeto.do"/>
            </action>

      <!--### REQUISITOS NÃO-FUNCIONAIS ASSOCIADOS ###############-->
            <action
                  path="/abraRequisitoNaoFuncionalAssociado"

      type="org.hopi.web.struts.action.AbraRequisitoNaoFuncionalAssociad
oAction">
                  <forward name="ok" path="/abraRequisitoFuncional.do"/>
```

```
            </action>
            <action
                  path="/novoRequisitoNaoFuncionalAssociado"

      type="org.hopi.web.struts.action.NovoRequisitoNaoFuncionalAssociad
oAction">
                  <forward name="ok" path="/requisitoNaoFuncional.jsp"/>
            </action>
            <action path="/salveRequisitoNaoFuncionalAssociado"
                        name="requisitoNaoFuncional"
                        scope="request"

      type="org.hopi.web.struts.action.SalveRequisitoNaoFuncionalAssocia
doAction"
                        input="/requisitoNaoFuncional.jsp">
                  <forward name="ok" path="/abraRequisitoFuncional.do"/>
            </action>
            <action
                  path="/excluaRequisitoNaoFuncionalAssociado"

      type="org.hopi.web.struts.action.ExcluaRequisitoNaoFuncionalAssoci
adoAction">
                  <forward name="ok" path="/abraRequisitoFuncional.do"/>
            </action>

      <!--### CASOS DE USO ######################################-->
            <action
                  path="/abraCasoDeUso"
                  type="org.hopi.web.struts.action.AbraCasoDeUsoAction">
                  <forward name="ok" path="/casoDeUso.jsp"/>
            </action>
            <action
                  path="/excluaCasoDeUso"

      type="org.hopi.web.struts.action.ExcluaCasoDeUsoAction">
                  <forward name="ok" path="/abraProjeto.do"/>
            </action>

      <!--#####################################################-->
    </action-mappings>

    <message-resources parameter="Mensagens"/>
</struts-config>
```

web.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--
<!DOCTYPE web-app
  PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.4//EN"
  "http://java.sun.com/j2ee/dtds/web-app_2_4.dtd">
-->

<web-app version="2.4">
  <display-name>HOPI</display-name>

  <!-- Standard Action Servlet Configuration (with debugging) -->
  <servlet>
    <servlet-name>action</servlet-name>
    <servlet-class>org.apache.struts.action.ActionServlet</servlet-
class>
    <init-param>
      <param-name>config</param-name>
      <param-value>/WEB-INF/struts-config.xml</param-value>
    </init-param>
    <init-param>
      <param-name>debug</param-name>
      <param-value>2</param-value>
    </init-param>
    <init-param>
      <param-name>detail</param-name>
      <param-value>2</param-value>
    </init-param>
    <load-on-startup>2</load-on-startup>
  </servlet>


  <!-- Standard Action Servlet Mapping -->
  <servlet-mapping>
    <servlet-name>action</servlet-name>
    <url-pattern>*.do</url-pattern>
  </servlet-mapping>

</web-app>
```

FOLDER WEBAPP

ator.jsp

```
<%@ page import="org.hopi.DTO.DTOSessao"%>
<%@ page import="org.hopi.ejb.entity.VOProjeto"%>
<%@ page import="org.hopi.ejb.entity.VOAtor"%>
<%@ taglib uri="WEB-INF/lib/struts/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="WEB-INF/lib/struts/struts-html.tld" prefix="html" %>
<%@ taglib uri="WEB-INF/lib/struts/struts-bean.tld" prefix="bean" %>

<logic:present name="dadosSessao" scope="session">
<html>
      <head>
    <title> HOPI - Ator </title>
            <link         href="xstylesheet.css"         rel="stylesheet"
type="text/css">
            <script language="JavaScript">
                  function popUpAjudaAtor()
                  {
                        ajudaAtor     =     window.open("ajudaAtor.jsp",
"AjudaAtor",
'toolbar=0,scrollbars=0,location=0,statusbar=0,menubar=0,resizable=1,wid
th=500,height=200');
                  }

                  function popUpSobre()
                  {
                        sobre  =  window.open("sobre.jsp",  "SobreoHOPI",
'toolbar=0,scrollbars=0,location=0,statusbar=0,menubar=0,resizable=1,wid
th=500,height=200');
                  }

                  function cancelar()
                  {
                        document.myForm.action = "abraProjeto.do";
                        document.myForm.submit();
                  }

                  function salvar()
                  {
                        document.myForm.action = "salveAtor.do";
                        document.myForm.submit();
                  }
            </script>
      </head>

      <body>
            <form method="post" name="myForm">
            <html:hidden name="ator" property="id"/>
            <html:hidden name="ator" property="idProjeto"/>
            <table class="titulo">
                  <!--
##########################################################-->
                  <!--# TITULO: HOPI e informações sobre a página aberta
#-->
                  <!--
##########################################################-->
                  <tr>
                        <td>
                              <table class="transparente">
```

```
                                    <tr height="50">
                                    <td width="10%">
                                                <font     class="titulo">
HOPI </font>
                                    </td>

                                    <td width="90%">
                                                <table
class="transparente">
                                                <tr
class="linhaItemSecao">
                                                      <td
class="colunaTipoItemSecao">
                                                            <font
class="miniVerdeEscuro">Usuário Logado: </font>
                                                      </td>

                                                      <td
class="colunaValorItemSecao">
                                                            <font
class="menu">

      <bean:write            scope="session"            name="dadosSessao"
property="nomeCompleto"/>
                                                            </font>
                                                      </td>
                                                </tr>

                                                <tr
class="linhaItemSecao">
                                                      <td
class="colunaTipoItemSecao">
                                                            <font
class="miniVerdeEscuro">Projeto Selecionado: </font>
                                                      </td>

                                                      <td
class="colunaValorItemSecao">
                                                            <font
class="menu">

      <bean:write scope="session" name="projeto" property="nome"/>
                                                            </font>
                                                      </td>
                                                </tr>
                                                </table>
                                    </td>
                                    </tr>
                              </table>
                        </td>
                  </tr>
                  <!--
##########################################################-->
                  <!--#  NAVEGAÇÃO:  apresenta  cada  etapa  da  análise  e
projeto #-->
                  <!--
##########################################################-->
                  <tr>
                        <td>
                              <table class="navegacaoOculta">
                                    <tr class="contornoNavegacao">
```

```
                                    <td colspan="5"></td>
                              </tr>

                              <tr class="linhaNavegacao">
                                    <td width=2%></td>

                                    <td class="etapas">
                                    <font
class="navegacaoEtapasMiniSelecionada">
                                                 Planejamento
                                    </font>
                                    <font class="navegacaoEtapas">
                                                 1 2
                                          </font>
                                    </td>

                                    <td width=3%></td>

                                    <td class="etapas">
                                    <font
class="navegacaoEtapasMini">
                                                 Análise
                                    </font>
                                    <font class="navegacaoEtapas">
                                                 1 2 3
                                          </font>
                                    </td>

                                    <td width=3%></td>

                                    <td class="etapas">
                                    <font
class="navegacaoEtapasMini">
                                                 Projeto
                                    </font>
                                    <font class="navegacaoEtapas">
                                                 1 2 3
                                          </font>
                                    </td>

                                    <td width="32%"></td>
                              </tr>

                              <tr class="contornoNavegacao">
                                    <td colspan="5"></td>
                              </tr>
                        </table>
                  </td>
            </tr>
            <!--
##########################################################-->
            <!--#   MENUS:   apresenta   menus   de   acesso   rápido
#-->
            <!--
##########################################################-->
            <tr>
                  <td>
                        <table class="menu">
                              <tr>
                                    <td>
                                          <font class="menu">
```

```
                                                       <a
class='linkMenu' href='principal.do'>Principal</a> |
                                                       <a
class='linkMenu' href='abraProjeto.do'>Projeto</a> |
                                                       <a
class='linkMenu' href='listeCategorias.do'>Categorias</a> |
                                                       <a
class='linkMenu' href='listeUsuarios.do'>Usuários</a> |
                                                       <a
class='linkMenu' href="javascript:popUpAjudaAtor()">Ajuda</a> |
                                                       <a
class='linkMenu' href="javascript:popUpSobre()">Sobre</a> |
                                                       <a
class='linkMenu' href='logout.do'>Logout</a>
                                                            </font>
                                                      </td>
                                                </tr>
                                          </table>
                                    </td>
                              </tr>
                              <!--
##########################################################-->
                              <!--# CADASTRO: apresenta o cadastro para entrada dos
dados #-->
                              <!--
##########################################################-->
                              <tr height="10"><td></td></tr>
                              <tr>
                                    <td>
                                          <table class="transparente">
                                                <tr>
                                                <td>
                                                      <font
class="nomePagina">Ator</font>
                                                </td>
                                                </tr>
                                          </table>
                                    </td>
                              </tr>
                              <tr height="15"><td></td></tr>
                              <tr>
                                    <td>
                                    <table class="transparente">
                                                <tr width="100%">
                                                      <td width="25%"></td>
                                                      <td width="75%">
                                                            <html:errors/>
                                                      </td>
                                                </tr>
                                                <tr width="100%">
                                                      <td width="25%"></td>
                                                      <td width="75%">
                                                      <font
class="normal">Nome</font>
                                                      </td>
                                                </tr>
                                                <tr width="100%">
                                                      <td></td>
                                                      <td>
                                                            <html:text
styleClass="edit" name="ator" property="nome" size="40"/>
```

```
                                    </td>
                                </tr>
                                <tr                    height="15"><td
colspan="3"></td></tr>
                                <tr>
                                    <td></td>
                                    <td colspan="2">
                                    <font
class="normal">Descrição</font>
                                    </td>
                                </tr>
                                <tr>
                                    <td></td>
                                    <td colspan="2">
                                        <html:textarea
styleClass="edit" name="ator" property="descricao" cols="70" rows="5"/>
                                    </td>
                                </tr>
                            </table>
                        </td>
                    </tr>
                    <tr height="15"><td></td></tr>
                    <tr>
                        <td>
                        <table class="transparente">
                                <tr>
                                <td align="right" width="72%">
                                        <input     class="normal"
type="button"                                          value="CANCELAR"
onClick="javascript:cancelar()">  
                                        <input     class="normal"
type="button" value="SALVAR" onClick="javascript:salvar()">
                                    </td>
                                    <td width="28%"></td>
                                </tr>
                            </table>
                        </td>
                    </tr>
                    <tr height="7"><td></td></tr>
                </table>
                </form>
        </body>
</html>
</logic:present>
<logic:notPresent name="dadosSessao" scope="session">
        <meta               http-equiv="refresh"               content="0;
URL=acessoNaoAutorizado.do">
</logic:notPresent>
```

casoDeUso.jsp

```
<%@ page import="java.util.ArrayList"%>
<%@ page import="org.hopi.ejb.entity.VOCasoDeUso"%>
<%@ page import="org.hopi.ejb.entity.VOPasso"%>
<%@ page import="org.hopi.ejb.entity.VOAtor"%>
<%@ page import="org.hopi.ejb.entity.VOInteressado"%>
<%@ page import="org.hopi.ejb.entity.VOPreCondicao"%>
<%@ page import="org.hopi.ejb.entity.VOPosCondicao"%>
<%@ page import="org.hopi.ejb.entity.VOVariacaoTecnologica"%>
<%@ page import="org.hopi.ejb.entity.VOQuestaoEmAberto"%>
<%@ page import="org.hopi.ejb.entity.VORequisitoFuncional"%>

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

<logic:present name="dadosSessao" scope="session">

<html>
      <head>
      <title>
            HOPI - Caso de Uso
      </title>
            <link         href="xstylesheet.css"         rel="stylesheet"
type="text/css">
            <script language="JavaScript">
                  function popUpAjudaCasoDeUso()
                  {
                        ajudaCasoDeUso                                    =
window.open("ajudaCasoDeUso.jsp",                        "AjudaCasoDeUso",
'toolbar=0,scrollbars=0,location=0,statusbar=0,menubar=0,resizable=1,wid
th=500,height=200');
                  }

                  function popUpSobre()
                  {
                        sobre  =  window.open("sobre.jsp",  "SobreoHOPI",
'toolbar=0,scrollbars=0,location=0,statusbar=0,menubar=0,resizable=1,wid
th=500,height=200');
                  }

                  function cancelar()
                  {
                      document.myForm.action = "principal.do";
                      document.myForm.submit();
                  }

                  function salvar()
                  {
                      document.myForm.action = "salveProjeto.do";
                      document.myForm.submit();
                  }

                  function novoAtor()
                  {
                      document.myForm.action = "novoAtor.do";
                      document.myForm.submit();
                  }

                  function abraAtor(id)
```

```
                    {
                        document.myForm.idAtor.value = id;
                        document.myForm.action = "abraAtor.do";
                        document.myForm.submit();
                    }

                    function excluaAtorDoCasoDeUso(id)
                    {
                        document.myForm.idAtor.value = id;
                        document.myForm.action                                =
"excluaAtorDoCasoDeUso.do";
                        document.myForm.submit();
                    }

                    function novoInteressado()
                    {
                        document.myForm.action = "novoInteressado.do";
                        document.myForm.submit();
                    }

                    function abraInteressado(id)
                    {
                        document.myForm.idInteressado.value = id;
                        document.myForm.action = "abraInteressado.do";
                        document.myForm.submit();
                    }

                    function excluaInteressado(id)
                    {
                        document.myForm.idInteressado.value = id;
                        document.myForm.action = "excluaInteressado.do";
                        document.myForm.submit();
                    }

                    function novaPreCondicao()
                    {
                        document.myForm.action = "novaPreCondicao.do";
                        document.myForm.submit();
                    }

                    function abraPreCondicao(id)
                    {
                        document.myForm.idPreCondicao.value = id;
                        document.myForm.action = "abraPreCondicao.do";
                        document.myForm.submit();
                    }

                    function excluaPreCondicao(id)
                    {
                        document.myForm.idPreCondicao.value = id;
                        document.myForm.action = "excluaPreCondicao.do";
                        document.myForm.submit();
                    }

                    function novaPosCondicao()
                    {
                        document.myForm.action = "novaPosCondicao.do";
                        document.myForm.submit();
                    }

                    function abraPosCondicao(id)
```

```
                {
                    document.myForm.idPosCondicao.value = id;
                    document.myForm.action = "abraPosCondicao.do";
                    document.myForm.submit();
                }

                function excluaPosCondicao(id)
                {
                    document.myForm.idPosCondicao.value = id;
                    document.myForm.action = "excluaPosCondicao.do";
                    document.myForm.submit();
                }

                function novaVariacaoTecnologica()
                {
                    document.myForm.action                              =
"novaVariacaoTecnologica.do";
                    document.myForm.submit();
                }

                function abraVariacaoTecnologica(id)
                {
                    document.myForm.idVariacaoTecnologica.value = id;
                    document.myForm.action                              =
"abraVariacaoTecnologica.do";
                    document.myForm.submit();
                }

                function excluaVariacaoTecnologica(id)
                {
                    document.myForm.idVariacaoTecnologica.value = id;
                    document.myForm.action                              =
"excluaVariacaoTecnologica.do";
                    document.myForm.submit();
                }

                function novaQuestaoEmAberto()
                {
                    document.myForm.action = "novaQuestaoEmAberto.do";
                    document.myForm.submit();
                }

                function abraQuestaoEmAberto(id)
                {
                    document.myForm.idQuestaoEmAberto.value = id;
                    document.myForm.action = "abraQuestaoEmAberto.do";
                    document.myForm.submit();
                }

                function excluaQuestaoEmAberto(id)
                {
                    document.myForm.idQuestaoEmAberto.value = id;
                    document.myForm.action                              =
"excluaQuestaoEmAberto.do";
                    document.myForm.submit();
                }

                function novoRequisitoFuncionalRelacionado()
                {
                    document.myForm.action                              =
"novoRequisitoFuncionalRelacionado.do";
```

```
                    document.myForm.submit();
                }

                function abraRequisitoFuncionalRelacionado(id)
                {

document.myForm.idRequisitoFuncionalRelacionado.value = id;
                    document.myForm.action                              =
"abraRequisitoFuncionalRelacionado.do";
                    document.myForm.submit();
                }

                function excluaRequisitoFuncionalRelacionado(id)
                {

document.myForm.idRequisitoFuncionalRelacionado.value = id;
                    document.myForm.action                              =
"excluaRequisitoFuncionalRelacionado.do";
                    document.myForm.submit();
                }
            </script>
      </head>

      <body>
            <form method="post" name="myForm">
            <html:hidden name="casoDeUso" property="id"/>
            <table class="titulo">
                  <!--
##########################################################-->
                  <!--# TITULO: HOPI e informações sobre a página aberta
#-->
                  <!--
##########################################################-->
                  <tr>
                        <td>
                              <table class="transparente">
                                    <tr height="50">
                                    <td width="10%">
                                                <font class="titulo">
                                                      HOPI
                                                </font>
                                    </td>

                                    <td width="90%">
                                                <table
class="transparente">
                                                <tr
class="linhaItemSecao">
                                                      <td
class="colunaTipoItemSecao">
                                                            <font
class="miniVerdeEscuro">
                                                                  Usuário
Logado:
                                                            </font>
                                                      </td>

                                                      <td
class="colunaValorItemSecao">
                                                            <font
class="menu">
```

```
      <bean:write            scope="session"            name="dadosSessao"
property="nomeCompleto"/>
                                                              </font>
                                                           </td>
                                                     </tr>

                                                     <tr
class="linhaItemSecao">
                                                         <td
class="colunaTipoItemSecao">
                                                               <font
class="miniVerdeEscuro">
                                                                     Projeto
Selecionado:
                                                               </font>
                                                         </td>

                                                         <td
class="colunaValorItemSecao">
                                                               <font
class="menu">

      <bean:write scope="session" name="projeto" property="nome"/>
                                                               </font>
                                                         </td>
                                                   </tr>
                                                   </table>
                                    </td>
                                    </tr>
                              </table>
                        </td>
                  </tr>

                  <!--
############################################################-->
                  <!--# NAVEGAÇÃO: apresenta cada etapa da análise e
projeto #-->
                  <!--
############################################################-->
                  <tr>
                        <td>
                              <table class="navegacao">
                                    <tr class="contornoNavegacao">
                                          <td colspan="5"></td>
                                    </tr>

                                    <tr class="linhaNavegacao">
                                          <td width=2%></td>

                                          <td class="etapas">
                                          <font
class="navegacaoEtapasMini">
                                                 Planejamento
                                          </font>
                                          <font
class="navegacaoEtapasRealizadas">
                                                 1
                                          </font>
                                          <font
class="navegacaoEtapaSelecionada">
```

```
                                                2
                                          </font>
                                    </td>

                                    <td width=3%></td>

                                    <td class="etapas">
                                    <font
class="navegacaoEtapasMini">
                                                 Análise
                                    </font>
                                    <font class="navegacaoEtapas">
                                                 1 2 3
                                          </font>
                                    </td>

                                    <td width=3%></td>

                                    <td class="etapas">
                                    <font
class="navegacaoEtapasMini">
                                                 Projeto
                                    </font>
                                    <font class="navegacaoEtapas">
                                                 1 2 3
                                          </font>
                                    </td>

                                    <td width="32%"></td>
                              </tr>
                              <tr class="contornoNavegacao">
                                    <td colspan="5"></td>
                              </tr>
                        </table>
                  </td>
            </tr>

            <!--
########################################################-->
            <!--#   MENUS:   apresenta   menus   de   acesso   rápido
#-->
            <!--
########################################################-->
            <tr>
                  <td>
                        <table class="menu">
                              <tr>
                                    <td>
                                          <font class="menu">
                                                 <a
class='linkMenu' href='principal.do'>Principal</a> |
                                                 <a
class='linkMenu' href='abraProjeto.do'>Projeto</a> |
                                                 <a
class='linkMenu' href='listeCategorias.do'>Categorias</a> |
                                                 <a
class='linkMenu' href='listeUsuarios.do'>Usuários</a> |
                                                 <a
class='linkMenu'
href="javascript:popUpAjudaCasoDeUso()">Ajuda</a> |
```

```
                                                       <a
class='linkMenu' href="javascript:popUpSobre()">Sobre</a> |
                                                       <a
class='linkMenu' href='logout.do'>Logout</a>
                                                </font>
                                          </td>
                                    </tr>
                              </table>
                        </td>
                  </tr>

                  <!--
##############################################################-->
                  <!--# CADASTRO: apresenta o cadastro para entrada dos
dados #-->
                  <!--
##############################################################-->
                  <tr height="10"><td></td></tr>
                  <tr>
                        <td>
                              <table class="transparente">
                                    <tr>
                                    <td>
                                          <font class="nomePagina">
                                                Caso de Uso
                                          </font>
                                    </td>
                                    </tr>
                              </table>
                        </td>
                  </tr>
                  <tr height="15"><td></td></tr>

                  <!--
#################################################################-->
                  <!--# NOME, FINALIDADE, VISÃO GERAL e BOTÕES SALVAR E
CANCELAR    #-->
                  <!--
#################################################################-->
                  <tr>
                        <td>
                        <table class="transparente">
                                    <tr>
                                          <td width="100%" colspan="2">
                                                <html:errors/>
                                          </td>
                                    </tr>
                                    <tr>
                                          <td width="50%">
                                          <font class="normal">
                                                Nome
                                          </font>
                                          </td>
                                          <td width="50%"></td>
                                    </tr>

                                    <tr>
                                          <td>
                                                <html:text
styleClass="edit" name="casoDeUso" property="nome" size="59"/>
                                          </td>
```

```
                                    <td></td>
                              </tr>

                              <tr                   height="15"><td
colspan="2"></td></tr>

                              <tr>
                                    <td>
                                          <font class="normal">
                                                Finalidade
                                          </font>
                              </td>
                              <td>
                                    <font class="normal">
                                          Visão Geral
                                    </font>
                              </td>
                              </tr>

                              <tr>
                                    <td width="50%">
                                          <html:textarea
name="casoDeUso" property="finalidade" cols="70" rows="5"/>
                              </td>

                              <td width="50%">
                                          <html:textarea
name="casoDeUso" property="visaoGeral" cols="70" rows="5"/>
                              </td>
                              </tr>

                              <tr height="10"><td></td></tr>
                              <tr align="right">
                              <td colspan=2 width="100%">
                                          <input     class="normal"
type="button" value="CANCELAR">  
                                          <input     class="normal"
type="button" value="SALVAR">      
                                    </td>
                              </tr>
                              <tr height="10"><td></td></tr>
                        </table>
                  </td>
            </tr>

            <!--
##########################################################################
#############-->
            <!--# ATORES e INTERESSADOS: apresenta a lista de
atores e interessados no caso de uso  #-->
            <!--
##########################################################################
#############-->
            <tr height="10"><td></td></tr>
            <tr>
                  <td>
                  <table class="transparente">
                              <tr>
                                    <td width="50%">
                                          <font class="normal">
                                                Atores
```

```
                                          </font>
                                    </td>
                                    <td width="50%">
                                          <font class="normal">
                                          Interessados
                                          </font>
                                          </td>
                                    </tr>
                              </table>
                        </td>
                  </tr>
                  <tr>
                        <td>
                        <table class="transparente">
                                    <tr>
                                          <td width="39%">
                                          <select   class="combobox"
style="width : 300">
                                                <logic:present
name="atores" scope="session">
                                                <logic:iterate
id="ator" name="atoresNaoRelacionados" type="VOAtor" indexId="i">

      <option><bean:write name="ator" property="nome"/></option>
                                                </logic:iterate>
                                                </logic:present>
                                          </select>
                                    </td>

                                    <td width="11%">
                                          <input          class="pequeno"
type="button" value="INCLUIR">
                                    </td>

                                    <td width="40%">
                                          <input       class="edit"
type="text" size="48">
                                    </td>

                                    <td width="10%">
                                          <input    class="pequeno"
type="button" value="INCLUIR">
                                          </td>
                                    </tr>
                              </table>
                        </td>
                  </tr>
                  <tr>
                        <td>
                        <table class="transparente">
                                    <%
                                    ArrayList atores = new ArrayList();
                                    ArrayList    interessados    =    new
ArrayList();

                                    if
(session.getAttribute("atoresCasoDeUso") != null)
                                    {
                                          atores                        =
(ArrayList)session.getAttribute("atoresCasoDeUso");
                                    }
```

```
                                    if
(session.getAttribute("interessadosCasoDeUso") != null)
                                    {
                                          interessados                  =
(ArrayList)session.getAttribute("interessadosCasoDeUso");
                                    }

                                    String atorId = "";
                                    String atorNome = "";
                                    String interessadoId = "";
                                    String interessadoDescricao = "";

                                    VOAtor ator;
                                    VOInteressado interessado;

                                    boolean linhaAtor = true;
                                    boolean linhaInteressado = true;

                                    int max = 0;

                                    if          (atores.size()          >
interessados.size())
                                    {
                                          max = atores.size();
                                    }
                                    else
                                    {
                                          max = interessados.size();
                                    }

                                    for (int i=0; i<max; i++)
                                    {
                                          if (i < atores.size())
                                          {
                                                ator                    =
(VOAtor)atores.get(i);
                                                atorId                  =
ator.getId().toString();
                                                atorNome                =
ator.getNome();
                                          }
                                          else
                                          {
                                                atorId = "";
                                                atorNome = "";
                                                linhaAtor = false;

                                          }

                                          if (i < interessados.size())
                                          {
                                                interessado             =
(VOInteressado)interessados.get(i);
                                                interessadoId           =
interessado.getId().toString();
                                                interessadoDescricao    =
interessado.getDescricao();
                                          }
                                          else
                                          {
```

```
                                          interessadoId = "";

                                          interessadoDescricao    =
"";
                                          linhaInteressado = false;
                                    }

                                    if (i % 2 == 0)
                                    {
                                          %>
                                          <tr class="linhaPar">
                                          <%
                                    }
                                    else
                                    {
                                    %>
                                          <tr class="linhaImpar">
                                          <%
                                    }


                                    if (linhaAtor)
                                    {
                                          %>
                                          <td width="43%">
                                                <font
class="lista">
                                                      <%=atorNome%>
                                                </font>
                                          </td>
                                          <td width="4%">
                                                <font
class="lista">
                                                      <a
href="javascript:abraAtor('<%=atorId%>')">
                                                            <img
src="xfig-seta.gif"    alt="Editar"    onMouseOver="javascript:src='xfig-
setaOver.gif'" onMouseOut="javascript:src='xfig-seta.gif'">
                                                      </a>

                                                       
                                                      <a
href="javascript:excluaAtor('<%=atorId%>')">
                                                            <img
src="xfig-xis.gif"    alt="Excluir"    onMouseOver="javascript:src='xfig-
xisOver.gif'" onMouseOut="javascript:src='xfig-xis.gif'">
                                                      </a>
                                                </font>
                                          </td>
                                          <%
                                    }
                                    else
                                    {
                                          %>
                                          <td            width="43%"
bgcolor="#FFFFFF"></td>
                                          <td             width="4%"
bgcolor="#FFFFFF"></td>
                                          <%
                                    }
                                    %>
```

```
                                    <td                         width="3%"
bgcolor="#FFFFFF"></td>
                                          <%
                                    if (linhaInteressado)
                                    {
                                          %>
                                                <td width="44%">
                                                      <font
class="lista">

      <%=interessadoDescricao%>
                                                      </font>
                                                </td>
                                                <td width="4.1%">
                                                      <font
class="lista">
                                                            <a
href="javascript:abraInteressado('<%=interessadoId%>')">
                                                                  <img
src="xfig-seta.gif"    alt="Editar"    onMouseOver="javasript:src='xfig-
setaOver.gif'" onMouseOut="javascript:src='xfig-seta.gif'">
                                                            </a>

                                                             
                                                            <a
href="javascript:excluaInteressado('<%=interessadoId%>')">
                                                                  <img
src="xfig-xis.gif"    alt="Excluir"    onMouseOver="javascript:src='xfig-
xisOver.gif'" onMouseOut="javascript:src='xfig-xis.gif'">
                                                            </a>
                                                      </font>
                                                </td>
                                                <%
                                          }
                                          else
                                          {
                                                %>
                                                <td            width="44%"
bgcolor="#FFFFFF"></td>
                                                <td             width="4%"
bgcolor="#FFFFFF"></td>
                                                <%
                                          }
                                          %>
                                          <td                   width="2%"
bgcolor="white"></td>
                                    </tr>
                                    <%
                                    }
                                    %>
                              </table>
                        </td>
                  </tr>
                  <tr height="15"><td></td></tr>

                  <!--
##########################################################################
#########-->
                  <!--# PRÉ e PÓS-CONDIÇÕES: apresenta a lista de pré e
pós-condições do caso de uso  #-->
```

```
                <!--
##########################################################################
#########-->
                <tr>
                      <td>
                      <table class="transparente">
                                  <tr>
                                        <td width="50%">
                                              <font class="normal">
                                                    Pré-Condições
                                              </font>
                                  </td>
                                  <td width="50%">
                                        <font class="normal">
                                        Pós-Condições
                                  </font>
                                  </td>
                                  </tr>
                            </table>
                      </td>
                </tr>
                <tr>
                      <td>
                      <table class="transparente">
                                  <tr>
                                        <td width="39%">
                                              <input       class="edit"
type="text" size="47">
                                  </td>

                                  <td width="11%">
                                        <input          class="pequeno"
type="button" value="INCLUIR">
                                  </td>

                                  <td width="40%">
                                              <input       class="edit"
type="text" size="48">
                                  </td>
                                  <td width="10%">
                                              <input    class="pequeno"
type="button" value="INCLUIR">
                                        </td>
                                  </tr>
                            </table>
                      </td>
                </tr>
                <tr>
                      <td>
                      <table class="transparente">
                                  <%
                                  ArrayList    preCondicoes    =    new
ArrayList();
                                  ArrayList    posCondicoes    =    new
ArrayList();

                                  if
(session.getAttribute("preCondicoes") != null)
                                  {
                                        preCondicoes                =
(ArrayList)session.getAttribute("preCondicoes");
```

```
                                    }

                                    if
(session.getAttribute("posCondicoes") != null)
                                    {
                                          posCondicoes                    =
(ArrayList)session.getAttribute("posCondicoes");
                                    }

                                    String preCondicaoId = "";
                                    String preCondicaoNome = "";
                                    String posCondicaoId = "";
                                    String posCondicaoNome = "";

                                    VOPreCondicao preCondicao;
                                    VOPosCondicao posCondicao;

                                    boolean linhaPreCondicao = true;
                                    boolean linhaPosCondicao = true;

                                    max = 0;

                                    if       (preCondicoes.size()       >
posCondicoes.size())
                                    {
                                          max = preCondicoes.size();
                                    }
                                    else
                                    {
                                          max = posCondicoes.size();
                                    }

                                    for (int i=0; i<max; i++)
                                    {
                                          if (i < preCondicoes.size())
                                          {
                                                preCondicao             =
(VOPreCondicao)preCondicoes.get(i);
                                                preCondicaoId           =
preCondicao.getId().toString();
                                                preCondicaoNome         =
preCondicao.getNome();
                                          }
                                          else
                                          {
                                                preCondicaoId = "";
                                                preCondicaoNome = "";
                                                linhaPreCondicao = false;

                                          }

                                          if (i < posCondicoes.size())
                                          {
                                                posCondicao             =
(VOPosCondicao)posCondicoes.get(i);
                                                posCondicaoId           =
posCondicao.getId().toString();
                                                posCondicaoNome         =
posCondicao.getNome();
                                          }
                                          else
```

```
                                        {
                                              posCondicaoId = "";

                                              posCondicaoNome   =   "";

                                              linhaPosCondicao = false;
                                        }

                                        if (i % 2 == 0)
                                        {
                                              %>
                                              <tr class="linhaPar">
                                              <%
                                        }
                                        else
                                        {
                                        %>
                                              <tr class="linhaImpar">
                                              <%
                                        }


                                        if (linhaPreCondicao)
                                        {
                                              %>
                                              <td width="43%">
                                                    <font
class="lista">

      <%=preCondicaoNome%>
                                                    </font>
                                              </td>
                                              <td width="4%">
                                                    <font
class="lista">
                                                          <a
href="javascript:abraPreCondicao('<%=preCondicaoId%>')">
                                                                <img
src="xfig-seta.gif"    alt="Editar"    onMouseOver="javascript:src='xfig-
setaOver.gif'" onMouseOut="javascript:src='xfig-seta.gif'">
                                                          </a>

                                                           
                                                          <a
href="javascript:excluaPreCondicao('<%=preCondicaoId%>')">
                                                                <img
src="xfig-xis.gif"    alt="Excluir"    onMouseOver="javascript:src='xfig-
xisOver.gif'" onMouseOut="javascript:src='xfig-xis.gif'">
                                                          </a>
                                                    </font>
                                              </td>
                                              <%
                                        }
                                        else
                                        {
                                              %>
                                              <td            width="43%"
bgcolor="#FFFFFF"></td>
                                              <td             width="4%"
bgcolor="#FFFFFF"></td>
                                              <%
```

```
                                        }
                                        %>
                                  <td                          width="3%"
bgcolor="#FFFFFF"></td>
                                        <%
                                  if (linhaPosCondicao)
                                  {
                                        %>
                                              <td width="44%">
                                                    <font
class="lista">

      <%=posCondicaoNome%>
                                                    </font>
                                              </td>
                                              <td width="4.1%">
                                                    <font
class="lista">
                                                          <a
href="javascript:abraPosCondicao('<%=posCondicaoId%>')">
                                                                <img
src="xfig-seta.gif"    alt="Editar"    onMouseOver="javasript:src='xfig-
setaOver.gif'" onMouseOut="javascript:src='xfig-seta.gif'">
                                                          </a>

                                                           
                                                          <a
href="javascript:excluaPosCondicao('<%=posCondicaoId%>')">
                                                                <img
src="xfig-xis.gif"    alt="Excluir"    onMouseOver="javascript:src='xfig-
xisOver.gif'" onMouseOut="javascript:src='xfig-xis.gif'">
                                                          </a>
                                                    </font>
                                              </td>
                                              <%
                                        }
                                        else
                                        {
                                              %>
                                              <td            width="44%"
bgcolor="#FFFFFF"></td>
                                              <td             width="4%"
bgcolor="#FFFFFF"></td>
                                              <%
                                        }
                                        %>
                                        <td                   width="2%"
bgcolor="white"></td>
                                  </tr>
                                  <%
                                  }
                                  %>
                            </table>
                      </td>
                </tr>
                <tr height="15"><td></td></tr>

                <!--
#########################################################################
############-->
```

```
                <!--# VARIAÇÕES TECNOLÓGICAS e QUESTÕES EM ABERTO:
apresenta a lista destes elementos  #-->
                <!--
##########################################################################
############--->
                <tr>
                      <td>
                      <table class="transparente">
                                  <tr>
                                        <td width="50%">
                                              <font class="normal">
                                                    Variações
Tecnológicas
                                              </font>
                                  </td>
                                  <td width="50%">
                                        <font class="normal">
                                        Questões em Aberto
                                        </font>
                                  </td>
                                  </tr>
                            </table>
                      </td>
                </tr>
                <tr>
                      <td>
                      <table class="transparente">
                                  <tr>
                                        <td width="39%">
                                              <input       class="edit"
type="text" size="47">
                                  </td>

                                  <td width="11%">
                                        <input          class="pequeno"
type="button" value="INCLUIR">
                                  </td>

                                  <td width="40%">
                                              <input       class="edit"
type="text" size="48">
                                  </td>
                                  <td width="10%">
                                              <input    class="pequeno"
type="button" value="INCLUIR">
                                        </td>
                                  </tr>
                            </table>
                      </td>
                </tr>
                <tr>
                      <td>
                      <table class="transparente">
                                  <%
                                  ArrayList variacoesTecnologicas = new
ArrayList();
                                  ArrayList   questoesEmAberto   =   new
ArrayList();

                                  if
(session.getAttribute("variacoesTecnologicas") != null)
```

```
                                    {
                                          variacoesTecnologicas         =
(ArrayList)session.getAttribute("variacoesTecnologicas");
                                    }

                                    if
(session.getAttribute("questoesEmAberto") != null)
                                    {
                                          questoesEmAberto              =
(ArrayList)session.getAttribute("questoesEmAberto");
                                    }

                                    String variacaoTecnologicaId = "";
                                    String variacaoTecnologicaNome = "";
                                    String questaoEmAbertoId = "";
                                    String questaoEmAbertoNome = "";

                                    VOVariacaoTecnologica
variacaoTecnologica;
                                    VOQuestaoEmAberto questaoEmAberto;

                                    boolean   linhaVariacaoTecnologica   =
true;
                                    boolean linhaQuestaoEmAberto = true;

                                    max = 0;

                                    if   (variacoesTecnologicas.size()   >
questoesEmAberto.size())
                                    {
                                          max                           =
variacoesTecnologicas.size();
                                    }
                                    else
                                    {
                                          max = questoesEmAberto.size();
                                    }

                                    for (int i=0; i<max; i++)
                                    {
                                          if             (i             <
variacoesTecnologicas.size())
                                          {
                                                variacaoTecnologica     =
(VOVariacaoTecnologica)variacoesTecnologicas.get(i);
                                                variacaoTecnologicaId   =
variacaoTecnologica.getId().toString();
                                                variacaoTecnologicaNome =
variacaoTecnologica.getDescricao();
                                          }
                                          else
                                          {
                                                variacaoTecnologicaId   =
"";
                                                variacaoTecnologicaNome =
"";
                                                linhaVariacaoTecnologica
= false;
                                          }
```

```
                                        if             (i             <
questoesEmAberto.size())
                                        {
                                              questaoEmAberto          =
(VOQuestaoEmAberto)questoesEmAberto.get(i);
                                              questaoEmAbertoId        =
questaoEmAberto.getId().toString();
                                              questaoEmAbertoNome      =
questaoEmAberto.getDescricao();
                                        }
                                        else
                                        {
                                              questaoEmAbertoId  =  "";

                                              questaoEmAbertoNome = "";

                                              linhaQuestaoEmAberto     =
false;
                                        }

                                        if (i % 2 == 0)
                                        {
                                              %>
                                              <tr class="linhaPar">
                                              <%
                                        }
                                        else
                                        {
                                        %>
                                              <tr class="linhaImpar">
                                              <%
                                        }


                                        if (linhaVariacaoTecnologica)
                                        {
                                              %>
                                              <td width="43%">
                                                    <font
class="lista">

      <%=variacaoTecnologicaNome%>
                                                    </font>
                                              </td>
                                              <td width="4%">
                                                    <font
class="lista">
                                                          <a
href="javascript:abraVariacaoTecnologica('<%=variacaoTecnologicaId%>')">
                                                                <img
src="xfig-seta.gif"    alt="Editar"    onMouseOver="javascript:src='xfig-
setaOver.gif'" onMouseOut="javascript:src='xfig-seta.gif'">
                                                          </a>

                                                           
                                                          <a
href="javascript:excluaVariacaoTecnologica('<%=variacaoTecnologicaId%>')
">
                                                                <img
src="xfig-xis.gif"    alt="Excluir"    onMouseOver="javascript:src='xfig-
xisOver.gif'" onMouseOut="javascript:src='xfig-xis.gif'">
```

```
                                                      </a>
                                                </font>
                                          </td>
                                          <%
                                    }
                                    else
                                    {
                                          %>
                                          <td            width="43%"
bgcolor="#FFFFFF"></td>
                                          <td             width="4%"
bgcolor="#FFFFFF"></td>
                                          <%
                                    }
                                    %>
                              <td                         width="3%"
bgcolor="#FFFFFF"></td>
                                    <%
                              if (linhaQuestaoEmAberto)
                              {
                                    %>
                                          <td width="44%">
                                                <font
class="lista">

      <%=questaoEmAbertoNome%>
                                          </font>
                                          </td>
                                          <td width="4.1%">
                                                <font
class="lista">
                                                      <a
href="javascript:abraQuestaoEmAberto('<%=questaoEmAbertoId%>')">
                                                            <img
src="xfig-seta.gif"    alt="Editar"    onMouseOver="javasript:src='xfig-
setaOver.gif'" onMouseOut="javascript:src='xfig-seta.gif'">
                                                      </a>

                                                       
                                                      <a
href="javascript:excluaQuestaoEmAberto('<%=questaoEmAbertoId%>')">
                                                            <img
src="xfig-xis.gif"    alt="Excluir"    onMouseOver="javascript:src='xfig-
xisOver.gif'" onMouseOut="javascript:src='xfig-xis.gif'">
                                                      </a>
                                                </font>
                                          </td>
                                          <%
                                    }
                                    else
                                    {
                                          %>
                                          <td            width="44%"
bgcolor="#FFFFFF"></td>
                                          <td             width="4%"
bgcolor="#FFFFFF"></td>
                                          <%
                                    }
                                    %>
                                          <td                   width="2%"
bgcolor="white"></td>
```

```
                                    </tr>
                                    <%
                                    }
                                    %>
                              </table>
                        </td>
                  </tr>
                  <tr height="15"><td></td></tr>

                  <!--
##########################################################################
########################-->
                  <!--# REQUISITOS RELACIONADOS: apresenta a lista dos
requisitos relacionados com o caso de uso    #-->
                  <!--
##########################################################################
########################-->
                  <tr>
                        <td>
                        <table class="transparente">
                              <tr>
                                    <td width="50%">
                                          <font class="normal">
                                                Requisitos
Funcionais Relacionados
                                          </font>
                              </td>
                                    <td width="50%"></td>
                              </tr>
                        </table>
                        </td>
                  </tr>
                  <tr>
                        <td>
                        <table class="transparente">
                              <tr>
                                    <td width="39%">
                                          <select   class="combobox"
style="width : 300">
                                                <logic:present
name="requisitosFuncionaisNaoRelacionados" scope="session">
                                                <logic:iterate
id="requisitoFuncionalNaoRelacionado"
name="requisitosFuncionaisNaoRelacionados"   type="VORequisitoFuncional"
indexId="i">

      <option><bean:write         name="requisitoFuncionalNaoRelacionado"
property="nome"/></option>
                                                </logic:iterate>
                                                </logic:present>
                                          </select>
                              </td>

                              <td width="11%">
                                    <input          class="pequeno"
type="button" value="INCLUIR">
                              </td>

                              <td width="50%" colspan="2"></td>
                              </tr>
                        </table>
```

```
                        </td>
                  </tr>
                  <tr>
                        <td>
                        <table class="transparente">
                                    <%
                                    ArrayList
requisitosFuncionaisRelacionados = new ArrayList();

                                    if
(session.getAttribute("requisitosFuncionaisRelacionados") != null)
                                    {

      requisitosFuncionaisRelacionados                                    =
(ArrayList)session.getAttribute("requisitosFuncionaisRelacionados");
                                    }

                                    String
requisitoFuncionalRelacionadoId = "";
                                    String
requisitoFuncionalRelacionadoNome = "";

                                    VORequisitoFuncional
requisitoFuncionalRelacionado;

                                    for             (int             i=0;
i<requisitosFuncionaisRelacionados.size(); i++)
                                    {
                                          requisitoFuncionalRelacionado =
(VORequisitoFuncional)requisitosFuncionaisRelacionados.get(i);
                                          requisitoFuncionalRelacionadoId
= requisitoFuncionalRelacionado.getId().toString();

      requisitoFuncionalRelacionadoNome                                   =
requisitoFuncionalRelacionado.getNome();

                                          if (i % 2 == 0)
                                          {
                                                %>
                                                <tr class="linhaPar">
                                                <%
                                          }
                                          else
                                          {
                                          %>
                                                <tr class="linhaImpar">
                                                <%
                                          }


                                          %>
                                          <td width="43%">
                                                <font class="lista">

      <%=requisitoFuncionalRelacionadoNome%>
                                                </font>
                                          </td>
                                          <td width="4%">
                                                <font class="lista">
```

```
                                                <a
href="javascript:abraRequisitoFuncionalRelacionado('<%=requisitoFunciona
lRelacionadoId%>')">
                                                      <img
src="xfig-seta.gif"    alt="Editar"    onMouseOver="javascript:src='xfig-
setaOver.gif'" onMouseOut="javascript:src='xfig-seta.gif'">
                                                </a>

                                                 
                                                <a
href="javascript:excluaRequisitoFuncionalRelacionado('<%=requisitoFuncio
nalRelacionadoId%>')">
                                                      <img
src="xfig-xis.gif"    alt="Excluir"    onMouseOver="javascript:src='xfig-
xisOver.gif'" onMouseOut="javascript:src='xfig-xis.gif'">
                                                </a>
                                          </font>
                                    </td>
                                    <td                 width="53%"
bgcolor="#FFFFFF" colspan="4"></td>
                              </tr>
                              <%
                              }
                              %>
                        </table>
                  </td>
            </tr>
            <tr height="15"><td></td></tr>

            <!--
##################################################################-->
            <!--# PASSOS: apresenta os passos do fluxo principal
#-->
            <!--
##################################################################-->
            <tr>
                  <td>
                  <table class="transparente">
                              <tr>
                                    <td>
                                          <font class="normal">
                                                Fluxo Principal
                                          </font>
                                    </td>
                              </tr>
                        </table>
                  </td>
            </tr>

            <tr>
                  <td>
                  <table class="transparente">
                              <tr>
                                    <td width="90%">
                                          <input       class="edit"
type="text" size="114">
                              </td>
                              <td width="10%">
                                          <input    class="pequeno"
type="button" value="INCLUIR">
                                    </td>
```

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

            <tr>
                  <td>
                  <table class="transparente">
                              <tr class="linhaImpar">
                              <td align="right" width="3%">
                                          <font class="lista">
                                                1.
                                          </font>
                              </td>
                                    <td width="87%">
                                          <font class="lista">
                                                Descrição        do
primeiro passo.
                                          </font>
                                    </td>
                                    <td width="8%">
                                          <font class="lista">
                                                <img     src="xfig-
v.gif">  <img src="xfig-a.gif">  
                                                <img     src="xfig-
seta.gif">  <img src="xfig-xis.gif">
                                          </font>
                                    </td>
                              <td                        width="2%"
bgcolor="#FFFFFF"></td>
                              </tr>

                              <tr class="linhaPar">
                              <td align="right">
                                          <font class="lista">
                                                2.
                                          </font>
                              </td>
                                    <td>
                                          <font class="lista">
                                                Descrição        do
segundo passo.
                                          </font>
                                    </td>
                                    <td>
                                          <font class="lista">
                                                <img     src="xfig-
v.gif">  <img src="xfig-a.gif">  
                                                <img     src="xfig-
seta.gif">  <img src="xfig-xis.gif">
                                          </font>
                                    </td>
                              <td bgcolor="#FFFFFF"></td>
                              </tr>

                              <tr class="linhaImpar">
                              <td align="right">
                                          <font class="lista">
                                                3.
                                          </font>
                              </td>
```

```
                                    <td>
                                          <font class="lista">
                                                Descrição        do
terceiro passo.
                                          </font>
                                    </td>
                                    <td>
                                          <font class="lista">
                                                <img     src="xfig-
v.gif">  <img src="xfig-a.gif">  
                                                <img     src="xfig-
seta.gif">  <img src="xfig-xis.gif">
                                          </font>
                                    </td>
                              <td bgcolor="#FFFFFF"></td>
                              </tr>
                              <tr class="linhaPar">
                              <td align="right">
                                          <font class="lista">
                                                4.
                                          </font>
                              </td>
                                    <td>
                                          <font class="lista">
                                                Descrição do quarto
passo.
                                          </font>
                                    </td>
                                    <td>
                                          <font class="lista">
                                                <img     src="xfig-
v.gif">  <img src="xfig-a.gif">  
                                                <img     src="xfig-
seta.gif">  <img src="xfig-xis.gif">
                                          </font>
                                    </td>
                              <td bgcolor="#FFFFFF"></td>
                              </tr>
                        </table>
                  </td>
            </tr>
            <tr height="15"><td></td></tr>

            <!--
#################################################################-->
            <!--#  VARIANTES:  apresenta  as  variantes  do  fluxo
principal      #-->
            <!--
#################################################################-->
            <tr>
                  <td>
                  <table class="transparente">
                              <tr>
                                    <td>
                                          <font class="normal">
                                                Variantes
                                          </font>
                                    </td>
                              </tr>
                        </table>
                  </td>
```

```
                </tr>
                <tr>
                  <td>
                    <table class="transparente">
                              <tr class="linhaImpar">
                                <td width="4%" align="right">
                                             <font class="lista">
                                                   3.1.
                                             </font>
                                </td>
                                    <td width="88%">
                                             <font class="lista">
                                                   Nome   da   primeira
variante do passo três.
                                             </font>
                                         </td>
                                         <td width="6%">
                                           <font class="lista">
                                             <img             src="xfig-
p.gif">  
                                               <img          src="xfig-
seta.gif">  <img src="xfig-xis.gif">
                                             </font>
                                         </td>
            <td width="2%" bgcolor="#FFFFFF"></td>
                                 </tr>

                               <tr class="linhaPar">
                                 <td align="right"></td>
                                    <td>
                                             <font class="lista">
                                                   3.1.1. Descrição do
primeiro passo da primeira variante do passo três.
                                             </font>
                                         </td>
                                         <td>
                                           <font class="lista">

    
                                             <img          src="xfig-
seta.gif">  <img src="xfig-xis.gif">
                                             </font>
                                         </td>
            <td bgcolor="#FFFFFF"></td>
                                 </tr>

                               <tr class="linhaImpar">
                                 <td align="right"></td>
                                    <td>
                                             <font class="lista">
                                                   3.1.2. Descrição do
segundo passo da primeira variante do passo três.
                                             </font>
                                         </td>
                                         <td>
                                           <font class="lista">

    
                                             <img          src="xfig-
seta.gif">  <img src="xfig-xis.gif">
                                             </font>
```

```
                                    </td>
            <td bgcolor="#FFFFFF"></td>
                                </tr>
                              <tr class="linhaPar">
                                <td align="right">
                                            <font class="lista">
                                                  3.2.
                                            </font>
                                </td>
                                    <td>
                                            <font class="lista">
                                                  Nome   da   segunda
variante do passo três.
                                            </font>
                                        </td>
                                        <td>
                                          <font class="lista">
                                            <img             src="xfig-
p.gif">  
                                               <img          src="xfig-
seta.gif">  <img src="xfig-xis.gif">
                                            </font>
                                        </td>
            <td bgcolor="#FFFFFF"></td>
                                </tr>
                              <tr class="linhaImpar">
                                <td align="right"></td>
                                    <td>
                                            <font class="lista">
                                                  3.2.1. Descrição do
primeiro passo da segunda variante do passo três.
                                            </font>
                                        </td>
                                        <td>
                                          <font class="lista">

    
                                               <img          src="xfig-
seta.gif">  <img src="xfig-xis.gif">
                                            </font>
                                        </td>
            <td bgcolor="#FFFFFF"></td>
                                </tr>
                            </table>
                    </td>
                </tr>
                <tr height="15"><td></td></tr>

      <!--
##########################################################################
##########-->
           <!--# FLUXOS ALTERNATIVOS: apresenta os fluxos alternativos
do fluxo principal       #-->
      <!--
##########################################################################
##########-->
                <tr>
                  <td>
                    <table class="transparente">
                      <tr>
                                    <td>
```

```
                                        <font     class="normal">Fluxos
Alternativos</font>
                                    </td>
                                </tr>
                                </table>
                            </td>
                        </tr>
                        <tr>
                          <td>
                            <table class="transparente">
                                    <tr class="linhaImpar">
                                      <td width="4%" align="right">
                                                  <font class="lista">
                                                        2.a.
                                                  </font>
                                      </td>
                                          <td width="86%">
                                                  <font class="lista">
                                                        Nome da exceção que
origina o primeiro fluxo alternativo do passo dois.
                                                  </font>
                                        </td>
                                        <td width="8%">
                                          <font class="lista">

        
                                            <img         src="xfig-
seta.gif">  <img src="xfig-xis.gif">
                                                  </font>
                                        </td>
                    <td width="2%" bgcolor="#FFFFFF"></td>
                                    </tr>

                                    <tr class="linhaPar">
                                      <td align="right"></td>
                                          <td>
                                                  <font class="lista">
                                                        2.a.1. Descrição da
primeira ação corretiva do primeiro fluxo alternativo do passo dois.
                                                  </font>
                                        </td>
                                        <td>
                                          <font class="lista">

        
                                            <img         src="xfig-
seta.gif">  <img src="xfig-xis.gif">
                                                  </font>
                                        </td>
                    <td bgcolor="#FFFFFF"></td>
                                    </tr>

                                    <tr class="linhaImpar">
                                      <td align="right"></td>
                                          <td>
                                                  <font class="lista">
                                                        2.a.2. Descrição da
segunda ação corretiva do primeiro fluxo alternativo do passo dois.
                                                  </font>
                                        </td>
                                        <td>
```

```
                                        <font class="lista">

        
                                            <img          src="xfig-
seta.gif">  <img src="xfig-xis.gif">
                                        </font>
                                    </td>
              <td bgcolor="#FFFFFF"></td>
                                </tr>
                              <tr class="linhaPar">
                                <td align="right"></td>
                                    <td>
                                        <font class="lista">
                                              2.a.3.  Finalização:
Retorna para o passo 3.
                                        </font>
                                    </td>
                                    <td>
                                      <font class="lista">

        
                                            <img          src="xfig-
seta.gif">  <img src="xfig-xis.gif">
                                        </font>
                                    </td>
              <td bgcolor="#FFFFFF"></td>
                                </tr>
                              <tr class="linhaImpar">
                                <td align="right">
                                        <font class="lista">
                                              2.b.
                                        </font>
                                </td>
                                    <td>
                                        <font class="lista">
                                              Nome da exceção que
origina o segundo fluxo alternativo do passo dois.
                                        </font>
                                    </td>
                                    <td>
                                      <font class="lista">
                                        <img              src="xfig-
p.gif">  
                                        <img              src="xfig-
f.gif"> 
                                            <img          src="xfig-
seta.gif">  <img src="xfig-xis.gif">
                                        </font>
                                    </td>
              <td bgcolor="#FFFFFF"></td>
                                </tr>
                              <tr class="linhaPar">
                                <td align="right"></td>
                                    <td>
                                        <font class="lista">
                                              2.b.1. Descrição da
primeira ação corretiva do segundo fluxo alternativo do passo dois.
                                        </font>
                                    </td>
                                    <td>
                                      <font class="lista">
```

```
        
                                                <img          src="xfig-
seta.gif">  <img src="xfig-xis.gif">
                                              </font>
                                            </td>
              <td bgcolor="#FFFFFF"></td>
                                </tr>
                              <tr class="linhaImpar">
                                <td align="right">
                                              <font class="lista">
                                                    4.a.
                                              </font>
                                </td>
                                    <td>
                                              <font class="lista">
                                                    Nome da exceção que
origina o primeiro fluxo alternativo do passo quatro.
                                              </font>
                                          </td>
                                          <td>
                                            <font class="lista">

        
                                                <img          src="xfig-
seta.gif">  <img src="xfig-xis.gif">
                                              </font>
                                          </td>
              <td bgcolor="#FFFFFF"></td>
                                </tr>
                              <tr class="linhaPar">
                                <td align="right"></td>
                                    <td>
                                              <font class="lista">
                                                    4.a.1. Descrição da
primeira ação corretiva do primeiro fluxo alternativo do passo quatro.
                                              </font>
                                          </td>
                                          <td>
                                            <font class="lista">

        
                                                <img          src="xfig-
seta.gif">  <img src="xfig-xis.gif">
                                              </font>
                                          </td>
              <td bgcolor="#FFFFFF"></td>
                                </tr>
                              <tr class="linhaImpar">
                                <td align="right"></td>
                                    <td>
                                              <font class="lista">
                                                    4.a.2.  Finalização:
O caso de uso é abortado.
                                              </font>
                                          </td>
                                          <td>
                                            <font class="lista">

        
```

```
                                                    <img          src="xfig-
seta.gif">  <img src="xfig-xis.gif">
                                                 </font>
                                              </td>
              <td bgcolor="#FFFFFF"></td>
                                 </tr>
                             </table>
                        </td>
                   </tr>
                   <tr height="15"><td></td></tr>

            </table>
            </form>
      </body>
</html>

</logic:present>

<logic:notPresent name="dadosSessao" scope="session">
      <meta                http-equiv="refresh"                content="0;
URL=acessoNaoAutorizado.do">
</logic:notPresent>
```

categoria.jsp

```
<%@ page import="org.hopi.DTO.DTOSessao"%>
<%@ page import="org.hopi.ejb.entity.VOCategoria"%>
<%@ taglib uri="WEB-INF/lib/struts/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="WEB-INF/lib/struts/struts-html.tld" prefix="html" %>
<%@ taglib uri="WEB-INF/lib/struts/struts-bean.tld" prefix="bean" %>

<logic:present name="dadosSessao" scope="session">
<html>
      <head>
      <title> HOPI - Categoria de Requisito Não-Funcional </title>
            <link         href="xstylesheet.css"         rel="stylesheet"
type="text/css">
            <script language="JavaScript">
                  function popUpAjudaCategoria()
                  {
                        ajudaCategoria                                    =
window.open("ajudaCategoria.jsp",                      "AjudaCategoria",
'toolbar=0,scrollbars=0,location=0,statusbar=0,menubar=0,resizable=1,wid
th=500,height=200');
                  }

                  function popUpSobre()
                  {
                        sobre  =  window.open("sobre.jsp",  "SobreoHOPI",
'toolbar=0,scrollbars=0,location=0,statusbar=0,menubar=0,resizable=1,wid
th=500,height=200');
                  }

                  function cancelar()
                  {
                        document.myForm.action = "listeCategorias.do";
                        document.myForm.submit();
                  }

                  function salvar()
                  {
                        document.myForm.action = "salveCategoria.do";
                        document.myForm.submit();
                  }
            </script>
      </head>

      <body>
      <form name="myForm" mathod="post">
            <html:hidden        styleClass="edit"        name="categoria"
property="id"/>
            <table class="titulo">
                  <!--
##########################################################-->
                  <!--# TITULO: HOPI e informações sobre a página aberta
#-->
                  <!--
##########################################################-->
                  <tr>
                        <td>
                              <table class="transparente">
                                    <tr height="50">
                                    <td width="10%">
```

```
                                                <font     class="titulo">
HOPI </font>
                                    </td>

                                    <td width="90%">
                                                <table
class="transparente">
                                                <tr
class="linhaItemSecao">
                                                      <td
class="colunaTipoItemSecao">
                                                            <font
class="miniVerdeEscuro">Usuário Logado: </font>
                                                      </td>

                                                      <td
class="colunaValorItemSecao">
                                                            <font
class="menu">

      <bean:write            scope="session"            name="dadosSessao"
property="nomeCompleto"/>
                                                            </font>
                                                      </td>
                                                </tr>
                                                </table>
                                    </td>
                                    </tr>
                              </table>
                        </td>
                  </tr>
                  <!--
############################################################-->
                  <!--# NAVEGAÇÃO: apresenta cada etapa da análise e
projeto #-->
                  <!--
############################################################-->
                  <tr>
                        <td>
                              <table class="navegacaoOculta">
                                    <tr class="contornoNavegacao">
                                          <td colspan="5"></td>
                                    </tr>
                                    <tr class="contornoNavegacao">
                                          <td colspan="5"></td>
                                    </tr>
                              </table>
                        </td>
                  </tr>

                  <!--
############################################################-->
                  <!--#   MENUS:   apresenta   menus   de   acesso   rápido
#-->
                  <!--
############################################################-->
                  <tr>
                        <td>
                              <table class="menu">
                                    <tr>
                                          <td>
```

```
                                                <font class="menu">
                                                       <a
class='linkMenu' href='principal.do'>Principal</a> |
                                                       <a
class='linkMenu' href='listeCategorias.do'>Categorias</a> |
                                                       <a
class='linkMenu' href='listeUsuarios.do'>Usuarios</a> |
                                                       <a
class='linkMenu'
href="javascript:popUpAjudaCategoria()">Ajuda</a> |
                                                       <a
class='linkMenu' href="javascript:popUpSobre()">Sobre</a> |
                                                       <a
class='linkMenu' href='logout.do'>Logout</a>
                                                </font>
                                          </td>
                                    </tr>
                              </table>
                        </td>
                  </tr>
                  <!--
##########################################################-->
                  <!--# CADASTRO: apresenta o cadastro para entrada dos
dados #-->
                  <!--
##########################################################-->
                  <tr height="10"><td></td></tr>
                  <tr>
                        <td>
                              <table class="transparente">
                                    <tr>
                                    <td>
                                          <font
class="nomePagina">Categoria de Requisito Não-Funcional</font>
                                    </td>
                                    </tr>
                              </table>
                        </td>
                  </tr>
                  <tr height="15"><td></td></tr>
                  <tr>
                        <td>
                              <table class="transparente">
                                    <tr width="100%">
                                          <td width="25%"></td>
                                          <td width="50%">
                                                <html:errors/>
                                          </td>
                                          <td width="25%"></td>
                                    </tr>
                                    <tr width="100%">
                                          <td></td>
                                          <td>
                                          <font
class="normal">Nome</font>
                                          </td>
                                          <td></td>
                                    </tr>
                                    <tr width="100%">
                                          <td></td>
                                          <td>
```

```
                                          <html:text
styleClass="edit" name="categoria" property="nome" size="40"/>
                                    </td>
                                    <td></td>
                              </tr>
                              <tr                    height="15"><td
colspan="3"></td></tr>
                              <tr>
                                    <td></td>
                                    <td>
                                    <font
class="normal">Descrição</font>
                                    </td>
                                    <td></td>
                              </tr>
                              <tr>
                                    <td></td>
                                    <td>
                                          <html:textarea
styleClass="edit"   name="categoria"   property="descricao"   cols="70"
rows="5"/>
                                    </td>
                                    <td></td>
                              </tr>
                        </table>
                  </td>
            </tr>
            <tr height="15"><td></td></tr>
            <tr>
                  <td>
                        <table class="transparente">
                              <tr>
                                    <td align="right" width="72%">
                                          <input     class="normal"
type="button"                                    value="CANCELAR"
onClick="javascript:cancelar()">  
                                          <input     class="normal"
type="button" value="SALVAR" onClick="javascript:salvar()">
                                    </td>
                                    <td width="28%"></td>
                              </tr>
                        </table>
                  </td>
            </tr>
            <tr height="7"><td></td></tr>
      </table>
</form>
</body>
</html>
</logic:present>

<logic:notPresent name="dadosSessao" scope="session">
      <meta               http-equiv="refresh"               content="0;
URL=acessoNaoAutorizado.do">
</logic:notPresent>
```

listeCategorias.jsp

```
<%@ page import="org.hopi.ejb.entity.VOCategoria"%>
<%@ taglib uri="WEB-INF/lib/struts/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="WEB-INF/lib/struts/struts-bean.tld" prefix="bean" %>
```

```
<logic:present name="dadosSessao" scope="session">

<html>
      <head>
            <title> HOPI - Login </title>
            <link         href="xstylesheet.css"         rel="stylesheet"
type="text/css">
            <script language="JavaScript">
                  function popUpAjudaListaCategorias()
                  {
                        ajudaListaCategorias                              =
window.open("ajudaListaCategorias.jsp",          "AjudaListaCategorias",
'toolbar=0,scrollbars=0,location=0,statusbar=0,menubar=0,resizable=1,wid
th=500,height=200');
                  }

                  function popUpSobre()
                  {
                        sobre  =  window.open("sobre.jsp",  "SobreoHOPI",
'toolbar=0,scrollbars=0,location=0,statusbar=0,menubar=0,resizable=1,wid
th=500,height=200');
                  }

                  function novaCategoria()
                  {
                      document.myForm.action = "novaCategoria.do";
                      document.myForm.submit();
                  }

                  function abraCategoria(id)
                  {
                      document.myForm.idCategoria.value = id;
                      document.myForm.action = "abraCategoria.do";
                      document.myForm.submit();
                  }

                  function excluaCategoria(id)
                  {
                      document.myForm.idCategoria.value = id;
                      document.myForm.action = "excluaCategoria.do";
                      document.myForm.submit();
                  }
            </script>
      </head>

      <body>
            <form name="myForm" method="post">
            <input type="hidden" name="idCategoria">
                  <table class="titulo">
                        <!--
##########################################################-->
                        <!--# TITULO: HOPI e informações sobre a página
aberta   #-->
                        <!--
##########################################################-->
                        <tr>
                              <td>
                                    <table class="transparente">
                                          <tr height="50">
                                                <td width="10%">
```

```
                                                <font
class="titulo"> HOPI </font>
                                          </td>

                                          <td width="90%">
                                                <table
class="transparente">
                                                      <tr
class="linhaItemSecao">
                                                            <td
class="colunaTipoItemSecao">

      <font class="miniVerdeEscuro">Usuário Logado: </font>
                                                            </td>

                                                            <td
class="colunaValorItemSecao">

      <font class="menu">

      <bean:write            scope="session"            name="dadosSessao"
property="nomeCompleto"/>

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

                  <!--
############################################################-->
                  <!--# NAVEGAÇÃO: apresenta cada etapa da análise
e projeto #-->
                  <!--
############################################################-->
                  <tr>
                        <td>
                              <table class="navegacaoOculta">
                                    <tr class="contornoNavegacao">
                                          <td colspan="5"></td>
                                    </tr>

                                    <tr class="contornoNavegacao">
                                          <td colspan="5"></td>
                                    </tr>
                              </table>
                        </td>
                  </tr>

            <!--
##########################################################-->
                  <!--#  MENUS:  apresenta  menus  de  acesso  rápido
#-->
                  <!--
##########################################################-->
                  <tr>
                        <td>
```

```
                                    <table class="menu">
                                          <tr>
                                                <td>
                                                      <font class="menu">
                                                             <a
class='linkMenu' href='principal.do'>Principal</a> |
                                                             <a
class='linkMenu' href='listeUsuarios.do'>Usuarios</a> |
                                                             <a
class='linkMenu'
href="javascript:popUpAjudaListaCategorias()">Ajuda</a> |
                                                             <a
class='linkMenu' href="javascript:popUpSobre()">Sobre</a> |
                                                             <a
class='linkMenu' href='logout.do'>Logout</a>
                                                      </font>
                                                </td>
                                          </tr>
                                    </table>
                              </td>
                        </tr>

                  <!--
##############################################################-->
                        <!--# CADASTRO: apresenta o cadastro para entrada
dos dados #-->
                        <!--
##############################################################-->
                        <tr height="10"><td></td></tr>
                        <tr>
                              <td>
                                    <table class="transparente">
                                          <tr>
                                          <td>
                                                      <font
class="nomePagina">Lista de Categorias de Requisito Não-Funcional</font>
                                          </td>
                                          </tr>
                                    </table>
                              </td>
                        </tr>
                        <tr height="15"><td></td></tr>
                        <tr>
                              <td>
                                    <table class="transparente">
                                          <tr width="100%">
                                                <td width="25%"></td>
                                                <td width="42%">
                                                <font
class="normal">Nome</font>
                                                </td>
                                                <td           width="43%"
colspan="3">
                                                      <input
class="pequeno"              type="button"             value="INCLUIR"
onClick="javascript:novaCategoria()">
                                                </td>
                                          </tr>
                                    </table>
                              </td>
                        </tr>
```

```
                    <tr>
                          <td>
                          <table class="transparente">
                                      <logic:present
name="categorias" scope="session">
                                      <logic:iterate   id="categoria"
name="categorias" type="VOCategoria" indexId="i">
                                            <%
                                            if (i.intValue() % 2 ==
0)
                                            {
                                                  %>
                                                  <tr
class="linhaPar">
                                                  <%
                                            }
                                            else
                                            {
                                                  %>
                                                  <tr
class="linhaImpar">
                                                  <%
                                            }
                                            %>
                                                  <td     width="25%"
bgcolor="#FFFFFF"></td>
                                                  <td width="45%">
                                                        <font
class="lista">

      <bean:write name="categoria" property="nome"/>
                                                        </font>
                                                  </td>
                                                  <td      width="5%"
colspan="3">
                                                        <font
class="lista">
                                                              <a
href="javascript:abraCategoria('<bean:write            name="categoria"
property="id"/>')">

      <img               src="xfig-seta.gif"              alt="Editar"
onMouseOver="javascript:src='xfig-setaOver.gif'"
onMouseOut="javascript:src='xfig-seta.gif'">
                                                              </a>

                                                               
                                                              <a
href="javascript:excluaCategoria('<bean:write          name="categoria"
property="id"/>')">

      <img               src="xfig-xis.gif"              alt="Excluir"
onMouseOver="javascript:src='xfig-xisOver.gif'"
onMouseOut="javascript:src='xfig-xis.gif'">
                                                              </a>

                                                        </font>
                                                  </td>
                                                  <td     width="25%"
bgcolor="#FFFFFF"></td>
```

```
                                                    </tr>
                                                </logic:iterate>
                                                </logic:present>
                                            </table>
                                        </td>
                                    </tr>
                                    <tr height="7"><td></td></tr>
                                </table>
                            </form>
                        </body>
</html>

</logic:present>

<logic:notPresent name="dadosSessao" scope="session">
      <meta                http-equiv="refresh"                content="0;
URL=acessoNaoAutorizado.do">
</logic:notPresent>
```

listeUsuarios.jsp

```
<%@ page import="org.hopi.ejb.entity.VOUsuario"%>
<%@ taglib uri="WEB-INF/lib/struts/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="WEB-INF/lib/struts/struts-bean.tld" prefix="bean" %>

<logic:present name="dadosSessao" scope="session">

<html>
      <head>
      <title>
            HOPI - Login
      </title>
            <link         href="xstylesheet.css"         rel="stylesheet"
type="text/css">

            <script language="JavaScript">
                  function popUpAjudaListaUsuarios()
                  {
                        ajudaListaUsuarios                                    =
window.open("ajudaListaUsuarios.jsp",              "AjudaListaUsuarios",
'toolbar=0,scrollbars=0,location=0,statusbar=0,menubar=0,resizable=1,wid
th=500,height=200');
                  }

                  function popUpSobre()
                  {
                        sobre  =  window.open("sobre.jsp",  "SobreoHOPI",
'toolbar=0,scrollbars=0,location=0,statusbar=0,menubar=0,resizable=1,wid
th=500,height=200');
                  }

                  function novoUsuario()
                  {
                      document.myForm.action = "novoUsuario.do";
                      document.myForm.submit();
                  }

                  function abraUsuario(id)
                  {
                      document.myForm.idUsuario.value = id;
                      document.myForm.action = "abraUsuario.do";
                      document.myForm.submit();
                  }

                  function excluaUsuario(id)
                  {
                      document.myForm.idUsuario.value = id;
                      document.myForm.action = "excluaUsuario.do";
                      document.myForm.submit();
                  }
            </script>
      </head>

      <body>
            <form name="myForm" method="post">
            <input type="hidden" name="idUsuario">
                  <table class="titulo">
                        <!--
#########################################################-->
```

```
                        <!--# TITULO: HOPI e informações sobre a página
aberta   #-->
                        <!--
########################################################-->
                        <tr>
                              <td>
                                    <table class="transparente">
                                          <tr height="50">
                                          <td width="10%">
                                                      <font
class="titulo">
                                                            HOPI
                                                      </font>
                                          </td>

                                          <td width="90%">
                                                      <table
class="transparente">
                                                      <tr
class="linhaItemSecao">
                                                            <td
class="colunaTipoItemSecao">
                                                                  <font
class="miniVerdeEscuro">

      Usuário Logado:
                                                                  </font>
                                                            </td>

                                                            <td
class="colunaValorItemSecao">
                                                                  <font
class="menu">

      <bean:write name="dadosSessao" property="nomeCompleto"/>
                                                                  </font>
                                                            </td>
                                                      </tr>
                                                      </table>
                                          </td>
                                          </tr>
                                    </table>
                              </td>
                        </tr>

                        <!--
########################################################-->
                        <!--# NAVEGAÇÃO: apresenta cada etapa da análise
e projeto #-->
                        <!--
########################################################-->
                        <tr>
                              <td>
                                    <table class="navegacaoOculta">
                                          <tr class="contornoNavegacao">
                                                <td colspan="5"></td>
                                          </tr>

                                          <tr class="contornoNavegacao">
                                                <td colspan="5"></td>
                                          </tr>
```

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

                  <!--
##########################################################-->
                  <!--# MENUS: apresenta menus de acesso rápido
#-->
                  <!--
##########################################################-->
                  <tr>
                        <td>
                              <table class="menu">
                                    <tr>
                                          <td>
                                                <font class="menu">
                                                 <a
class='linkMenu' href='principal.do'>Principal</a> |
                                                 <a
class='linkMenu' href='listeCategorias.do'>Categorias</a> |
                                                 <a
class='linkMenu'
href="javascript:popUpAjudaListaUsuarios()">Ajuda</a> |
                                                 <a
class='linkMenu' href="javascript:popUpSobre()">Sobre</a> |
                                                 <a
class='linkMenu' href='logout.do'>Logout</a>
                                                </font>
                                          </td>
                                    </tr>
                              </table>
                        </td>
                  </tr>

                  <!--
##########################################################-->
                  <!--# CADASTRO: apresenta o cadastro para entrada
dos dados #-->
                  <!--
##########################################################-->
                  <tr height="10"><td></td></tr>
                  <tr>
                        <td>
                              <table class="transparente">
                                    <tr>
                                    <td>
                                          <font class="nomePagina">
                                                Lista de Usuários
                                          </font>
                                    </td>
                                    </tr>
                              </table>
                        </td>
                  </tr>
                  <tr height="15"><td></td></tr>
                  <tr>
                        <td>
                        <table class="transparente">
                                    <tr width="100%">
                                          <td width="25%">
                                          </td>
```

```
                                                <td width="42%">
                                                <font
class="normal">Nome</font>
                                                </td>
                                                <td            width="43%"
colspan="3">
                                                <input    class="pequeno"
type="button" value="INCLUIR" onClick="javascript:novoUsuario()">
                                                </td>
                                          </tr>
                                    </table>
                              </td>
                        </tr>

                        <tr>
                              <td>
                              <table class="transparente">
                                          <logic:present  name="usuarios"
scope="session">
                                          <logic:iterate     id="usuario"
name="usuarios" type="VOUsuario" indexId="i">
                                                <%
                                                if  (i.intValue()  %  2  ==
0)
                                                {
                                                      %>
                                                      <tr
class="linhaPar">
                                                      <%
                                                }
                                                else
                                                {
                                                      %>
                                                      <tr
class="linhaImpar">
                                                      <%
                                                }
                                                %>
                                                      <td     width="25%"
bgcolor="#FFFFFF"></td>
                                                      <td width="45%">
                                                            <font
class="lista">

      <bean:write name="usuario" property="nome"/>
                                                            </font>
                                                      </td>
                                                      <td       width="5%"
colspan="3">
                                                            <font
class="lista">
                                                                  <a
href="javascript:abraUsuario('<bean:write                 name="usuario"
property="username"/>')">

      <img                src="xfig-seta.gif"                alt="Editar"
onMouseOver="javascript:src='xfig-setaOver.gif'"
onMouseOut="javascript:src='xfig-seta.gif'">
                                                                  </a>

                                                                   
```

```
                                                                    <a
href="javascript:excluaUsuario('<bean:write                name="usuario"
property="id"/>')">

      <img                src="xfig-xis.gif"                alt="Excluir"
onMouseOver="javascript:src='xfig-xisOver.gif'"
onMouseOut="javascript:src='xfig-xis.gif'">
                                                                    </a>

                                                                </font>
                                                          </td>
                                                          <td      width="25%"
bgcolor="#FFFFFF"></td>
                                                    </tr>
                                               </logic:iterate>
                                               </logic:present>
                                          </table>
                                     </td>
                                </tr>
                                <tr height="7"><td></td></tr>
                           </table>
                      </form>
      </body>
</html>

</logic:present>

<logic:notPresent name="dadosSessao" scope="session">
      <meta                http-equiv="refresh"                content="0;
URL=acessoNaoAutorizado.do">
</logic:notPresent>
```

login.jsp

```
<%@ taglib uri="../WEB-INF/lib/struts/struts-html.tld" prefix="html" %>
<html>
      <head>
            <title> HOPI - Login </title>
            <link         href="xstylesheet.css"         rel="stylesheet"
type="text/css">

            <script language="JavaScript">
                  function popUpAjudaLogin()
                  {
                        ajudaLogin    =    window.open("ajudaLogin.jsp",
"AjudaLogin",
'toolbar=0,scrollbars=0,location=0,statusbar=0,menubar=0,resizable=1,wid
th=500,height=200');
                  }

                  function popUpSobre()
                  {
                        sobre  =  window.open("sobre.jsp",  "SobreoHOPI",
'toolbar=0,scrollbars=0,location=0,statusbar=0,menubar=0,resizable=1,wid
th=500,height=200');
                  }
            </script>
      </head>

      <body>
            <html:form method="post" action="login.do" focus="username">

                  <%
                        String                    erro                    =
(String)request.getAttribute("erro");
                        if (erro == null)
                        {
                              erro = "";
                        }
                  %>

                  <table class="titulo">
                        <!--
########################################################-->
                        <!--# TITULO: HOPI e informações sobre a página
aberta   #-->
                        <!--
########################################################-->
                        <tr>
                              <td>
                                    <table class="transparente">
                                          <tr height="50">
                                                <td width="10%">
                                                      <font
class="titulo"> HOPI </font>
                                                </td>
                                                <td width="90%"></td>
                                          </tr>
                                    </table>
                              </td>
                        </tr>
```

```
                        <!--
############################################################-->
                        <!--# NAVEGAÇÃO: apresenta cada etapa da análise
e projeto #-->
                        <!--
############################################################-->
                        <tr>
                              <td>
                                    <table class="navegacaoOculta">
                                          <tr class="contornoNavegacao">
                                                <td colspan="5"></td>
                                          </tr>

                                          <tr class="contornoNavegacao">
                                                <td colspan="5"></td>
                                          </tr>
                                    </table>
                              </td>
                        </tr>

                        <!--
############################################################-->
                        <!--#  MENUS:  apresenta  menus  de  acesso  rápido
#-->
                        <!--
############################################################-->
                        <tr>
                              <td>
                                    <table class="menu">
                                          <tr>
                                                <td>
                                                      <font class="menu">
                                                             <a
class='linkMenu' href="javascript:popUpAjudaLogin()">Ajuda</a> |
                                                             <a
class='linkMenu' href="javascript:popUpSobre()">Sobre</a>
                                                      </font>
                                                </td>
                                          </tr>
                                    </table>
                              </td>
                        </tr>

                        <!--
############################################################-->
                        <!--# CADASTRO: apresenta o cadastro para entrada
dos dados #-->
                        <!--
############################################################-->
                        <tr height="10"><td></td></tr>
                        <tr>
                              <td>
                                    <table class="transparente">
                                          <tr>
                                                <td>
                                                <font
class="nomePagina">Login</font>
                                          </td>
                                          </tr>
                                    </table>
                              </td>
```

```
                        </tr>
                        <tr height="15"><td></td></tr>
                        <tr>
                              <td>
                              <table class="transparente">
                                          <tr width="100%">
                                                <td width="25%"></td>
                                                <td width="75%">
                                                      <%= erro %>
                                                      <html:errors/>
                                                </td>
                                          </tr>
                                          <tr width="100%">
                                                <td width="25%"></td>
                                                <td width="75%">
                                              <font
class="normal">Username</font>
                                                </td>
                                          </tr>
                                          <tr width="100%">
                                                <td width="25%"></td>
                                                <td width="75%">
                                                      <input  class="edit"
type="text" name="username" size="58">
                                                </td>
                                          </tr>
                                          <tr              height="15"><td
colspan="3"></td></tr>
                                          <tr>
                                                <td></td>
                                                <td>
                                                <font
class="normal">Password</font>
                                                </td>
                                          </tr>
                                          <tr>
                                                <td></td>
                                                <td>
                                                      <input  class="edit"
type="password" name="password" size="58">
                                                </td>
                                          </tr>
                                    </table>
                              </td>
                        </tr>
                        <tr height="15"><td></td></tr>
                        <tr>
                              <td>
                              <table class="transparente">
                                          <tr>
                                          <td align="right" width="72%">
                                                      <input
class="normal" type="submit" value="ENTRAR">
                                                </td>
                                                <td width="28%"></td>
                                          </tr>
                                    </table>
                              </td>
                        </tr>
                        <tr height="7"><td></td></tr>
                  </table>
```

```
            </html:form>
      </body>
</html>
```

logout.jsp

```
<%@       taglib       uri="http://jakarta.apache.org/struts/tags-html"
prefix="html" %>
<html>
      <head>
            <title> HOPI - Logout </title>
            <link         href="xstylesheet.css"         rel="stylesheet"
type="text/css">

            <script language="JavaScript">
                  function popUpAjudaLogout()
                  {
                        ajudaLogout    =    window.open("ajudaLogout.jsp",
"AjudaLogout",
'toolbar=0,scrollbars=0,location=0,statusbar=0,menubar=0,resizable=1,wid
th=500,height=200');
                  }

                  function popUpSobre()
                  {
                        sobre  =  window.open("sobre.jsp",  "SobreoHOPI",
'toolbar=0,scrollbars=0,location=0,statusbar=0,menubar=0,resizable=1,wid
th=500,height=200');
                  }
            </script>
      </head>

      <body>
            <html:form action="login.do" focus="username">
                  <%
                        String                    erro                    =
(String)request.getAttribute("erro");
                        if (erro == null)
                        {
                              erro = "";
                        }
                  %>

                  <table class="titulo">
                        <!--
############################################################-->
                        <!--# TITULO: HOPI e informações sobre a página
aberta   #-->
                        <!--
############################################################-->
                        <tr>
                              <td>
                                    <table class="transparente">
                                          <tr height="50">
                                                <td width="10%">
                                                      <font
class="titulo"> HOPI </font>
                                                </td>
                                                <td width="90%"></td>
                                          </tr>
                                    </table>
                              </td>
                        </tr>
```

```
                        <!--
############################################################-->
                        <!--# NAVEGAÇÃO: apresenta cada etapa da análise
e projeto #-->
                        <!--
############################################################-->
                        <tr>
                              <td>
                                    <table class="navegacaoOculta">
                                          <tr class="contornoNavegacao">
                                                <td colspan="5"></td>
                                          </tr>

                                          <tr class="contornoNavegacao">
                                                <td colspan="5"></td>
                                          </tr>
                                    </table>
                              </td>
                        </tr>

                        <!--
############################################################-->
                        <!--#  MENUS:  apresenta  menus  de  acesso  rápido
#-->
                        <!--
############################################################-->
                        <tr>
                              <td>
                                    <table class="menu">
                                          <tr>
                                                <td>
                                                      <font class="menu">
                                                       <a
class='linkMenu' href="login.jsp">Login</a> |
                                                       <a
class='linkMenu' href="javascript:popUpAjudaLogout()">Ajuda</a> |
                                                       <a
class='linkMenu' href="javascript:popUpSobre()">Sobre</a>
                                                      </font>
                                                </td>
                                          </tr>
                                    </table>
                              </td>
                        </tr>

                        <!--
############################################################-->
                        <!--# CADASTRO: apresenta o cadastro para entrada
dos dados #-->
                        <!--
############################################################-->
                        <tr height="10"><td></td></tr>
                        <tr>
                              <td>
                                    <table class="transparente">
                                          <tr>
                                                <td>
                                                <font
class="nomePagina">Logout</font>
                                          </td>
                                          </tr>
```

```
                                    </table>
                              </td>
                        </tr>
                        <tr height="15"><td></td></tr>
                        <tr>
                              <td>
                              <table class="transparente">
                                          <tr>
                                                <td>
                                                      <font
class="normal">Você realizou o logout com sucesso.</font>
                                                </td>
                                          </tr>
                                          <tr>
                                                <td>
                                                      <font
class="normal">Muito obrigado por utilizar o sistema!</font>
                                                </td>
                                          </tr>
                                    </table>
                              </td>
                        </tr>
                        <tr height="7"><td></td></tr>
                  </table>
            </html:form>
      </body>
</html>
```

principal.jsp

```
<%@ page import="org.hopi.DTO.DTOSessao"%>
<%@ page import="org.hopi.ejb.entity.VOProjeto"%>
<%@ page import="java.util.ArrayList"%>
<%@ taglib uri="WEB-INF/lib/struts/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="WEB-INF/lib/struts/struts-bean.tld" prefix="bean" %>

<logic:present name="dadosSessao" scope="session">

<html>
      <head>
          <title> HOPI - Login </title>
            <link         href="xstylesheet.css"         rel="stylesheet"
type="text/css">

            <script language="JavaScript">
                  function popUpAjudaPrincipal()
                  {
                        ajudaPrincipal                                    =
window.open("ajudaPrincipal.jsp",                      "AjudaPrincipal",
'toolbar=0,scrollbars=0,location=0,statusbar=0,menubar=0,resizable=1,wid
th=500,height=200');
                  }

                  function popUpSobre()
                  {
                        sobre  =  window.open("sobre.jsp",  "SobreoHOPI",
'toolbar=0,scrollbars=0,location=0,statusbar=0,menubar=0,resizable=1,wid
th=500,height=200');
                  }

                  function abraProjeto(id)
                  {
                      document.myForm.idProjeto.value = id;
                      document.myForm.action = 'abraProjeto.do';
                      document.myForm.submit();
                  }

                  function excluaProjeto(id)
                  {
                      document.myForm.idProjeto.value = id;
                      document.myForm.action = 'excluaProjeto.do';
                      document.myForm.submit();
                  }

                  function novoProjeto()
                  {
                        document.myForm.action = 'novoProjeto.do';
                        document.myForm.submit();
                  }
            </script>
      </head>

      <body>
      <form method="post" name="myForm">
            <input type="hidden" name="idProjeto">
            <table class="titulo">
                  <!--
#######################################################-->
```

```
                  <!--# TITULO: HOPI e informações sobre a página aberta
#-->
                  <!--
##########################################################-->
                  <tr>
                        <td>
                              <table class="transparente">
                                    <tr height="50">
                                    <td width="10%">
                                                <font     class="titulo">
HOPI </font>
                                          </td>

                                    <td width="90%">
                                                <table
class="transparente">
                                                <tr
class="linhaItemSecao">
                                                      <td
class="colunaTipoItemSecao">
                                                            <font
class="miniVerdeEscuro">Usuário Logado: </font>
                                                            </td>

                                                      <td
class="colunaValorItemSecao">
                                                            <font
class="menu">

     <bean:write           scope="session"           name="dadosSessao"
property="nomeCompleto"/>
                                                            </font>
                                                      </td>
                                                </tr>
                                                </table>
                                    </td>
                                    </tr>
                              </table>
                        </td>
                  </tr>

                  <!--
##########################################################-->
                  <!--#  NAVEGAÇÃO:  apresenta  cada  etapa  da  análise  e
projeto #-->
                  <!--
##########################################################-->
                  <tr>
                        <td>
                              <table class="navegacaoOculta">
                                    <tr class="contornoNavegacao">
                                          <td colspan="5"></td>
                                    </tr>

                                    <tr class="contornoNavegacao">
                                          <td colspan="5"></td>
                                    </tr>
                              </table>
                        </td>
                  </tr>
```

```
                  <!--
##########################################################-->
                  <!--#   MENUS:   apresenta   menus   de   acesso   rápido
#-->
                  <!--
##########################################################-->
                  <tr>
                        <td>
                              <table class="menu">
                                    <tr>
                                          <td>
                                                <font class="menu">
                                                       <a
class='linkMenu' href='listeCategorias.do'>Categorias</a> |
                                                       <a
class='linkMenu' href='listeUsuarios.do'>Usuários</a> |
                                                       <a
class='linkMenu'
href="javascript:popUpAjudaPrincipal()">Ajuda</a> |
                                                       <a
class='linkMenu' href="javascript:popUpSobre()">Sobre</a> |
                                                       <a
class='linkMenu' href='logout.do'>Logout</a>
                                                </font>
                                          </td>
                                    </tr>
                              </table>
                        </td>
                  </tr>

                <!--
##########################################################-->
                  <!--# CADASTRO: apresenta o cadastro para entrada dos
dados #-->
                <!--
##########################################################-->
                  <tr height="10"><td></td></tr>
                  <tr>
                        <td>
                              <table class="transparente">
                                    <tr>
                                    <td>
                                          <font
class="nomePagina">Principal</font>
                                       </td>
                                    </tr>
                              </table>
                        </td>
                  </tr>
                  <tr height="15"><td></td></tr>
                  <tr>
                        <td>
                        <table class="transparente">
                              </tr>
                                          <td width="39%">
                                                <font
class="normal">Projetos para Edição</font>
                                    </td>

                                    <td width="11%">
```

```
                                          <input      class="pequeno"
type="button" value="INCLUIR" onClick="javascript:novoProjeto()">
                                    </td>

                                    <td width="40%">
                                          <font
class="normal">Projetos para Visualização</font>
                                    </td>

                                    <td width="10%"></td>
                              </tr>
                        </table>
                  </td>
            </tr>
            <tr>
                  <td>
                  <table class="transparente">
                        <%
                        ArrayList          projetosEdicao          =
(ArrayList)session.getAttribute("projetosEdicao");
                        ArrayList      projetosVisualizacao      =
(ArrayList)session.getAttribute("projetosVisualizacao");

                        String projetoEdicaoId   = "";
                        String projetoEdicaoNome = "";
                        String projetoVisualizacaoNome = "";

                        VOProjeto projetoEdicao;
                        VOProjeto projetoVisualizacao;

                        boolean linhaProjetoEdicao = true;
                        boolean linhaProjetoVisualizacao = true;

                        int max = 0;

                        if         (projetosEdicao.size()        >
projetosVisualizacao.size())
                        {
                              max = projetosEdicao.size();
                        }
                        else
                        {
                              max = projetosVisualizacao.size();
                        }

                        for (int i=0; i<max; i++)
                        {
                              if (i < projetosEdicao.size())
                              {
                                    projetoEdicao                =
(VOProjeto)projetosEdicao.get(i);
                                    projetoEdicaoNome            =
projetoEdicao.getNome();
                                    projetoEdicaoId              =
projetoEdicao.getId().toString();
                              }
                              else
                              {
                                    projetoEdicaoNome = "";
                                    linhaProjetoEdicao = false;
```

```
                                    }

                                    if (i < projetosVisualizacao.size())
                                    {
                                          projetoVisualizacao           =
(VOProjeto)projetosVisualizacao.get(i);
                                          projetoVisualizacaoNome       =
projetoVisualizacao.getNome();
                                    }
                                    else
                                    {
                                          projetoVisualizacaoNome = "";

                                          linhaProjetoVisualizacao      =
false;
                                    }

                                    if (i % 2 == 0)
                                    {
                                          %>
                                          <tr class="linhaPar">
                                          <%
                                    }
                                    else
                                    {
                                    %>
                                          <tr class="linhaImpar">
                                          <%
                                    }

                                    if (linhaProjetoEdicao)
                                    {
                                          %>
                                          <td width="43%">
                                                <font class="lista">

     <%=projetoEdicaoNome%>
                                                </font>
                                          </td>
                                          <td width="4%">
                                                <font class="lista">
                                                      <a
href="javascript:abraProjeto('<%=projetoEdicaoId%>')">
                                                            <img
src="xfig-seta.gif"    alt="Editar"    onMouseOver="javascript:src='xfig-
setaOver.gif'" onMouseOut="javascript:src='xfig-seta.gif'">
                                                      </a>

                                                       
                                                      <a
href="javascript:excluaProjeto('<%=projetoEdicaoId%>')">
                                                            <img
src="xfig-xis.gif"    alt="Excluir"    onMouseOver="javascript:src='xfig-
xisOver.gif'" onMouseOut="javascript:src='xfig-xis.gif'">
                                                      </a>
                                                </font>
                                          </td>
                                          <%
                                    }
                                    else
                                    {
```

```
                                          %>
                                          <td                  width="43%"
bgcolor="#FFFFFF"></td>
                                          <td                   width="4%"
bgcolor="#FFFFFF"></td>
                                          <%
                                    }
                                    %>
                              <td width="3%" bgcolor="#FFFFFF"></td>
                                    <%
                              if (linhaProjetoVisualizacao)
                              {
                                    %>
                                          <td width="46%">
                                                <font class="lista">

      <%=projetoVisualizacaoNome%>
                                          </font>
                                          </td>
                                          <td width="2.1%">
                                                <font class="lista">
                                                      <img     src="xfig-
seta.gif">
                                          </font>
                                          </td>
                                          <%
                                    }
                                    else
                                    {
                                          %>
                                          <td                   width="46%"
bgcolor="#FFFFFF"></td>
                                          <td                  width="2.1%"
bgcolor="#FFFFFF"></td>
                                          <%
                                    }
                                    %>
                                    <td width="2%" bgcolor="white"></td>
                              </tr>
                              <%
                              }
                              %>

                              </table>
                        </td>
                  </tr>
                  <tr height="7"><td></td></tr>
            </table>
      </form>
      </body>
</html>

</logic:present>

<logic:notPresent name="dadosSessao" scope="session">
<meta http-equiv="refresh" content="0; URL=acessoNaoAutorizado.do">
</logic:notPresent>
```

projeto.jsp

```
<%@ page import="java.util.ArrayList"%>
<%@ page import="org.hopi.DTO.DTOSessao"%>
<%@ page import="org.hopi.ejb.entity.VOProjeto"%>
<%@ page import="org.hopi.ejb.entity.VOAtor"%>
<%@ page import="org.hopi.ejb.entity.VORequisitoFuncional"%>
<%@ page import="org.hopi.ejb.entity.VORequisitoNaoFuncional"%>
<%@ page import="org.hopi.ejb.entity.VOCasoDeUso"%>
<%@ taglib uri="WEB-INF/lib/struts/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="WEB-INF/lib/struts/struts-html.tld" prefix="html" %>
<%@ taglib uri="WEB-INF/lib/struts/struts-bean.tld" prefix="bean" %>

<logic:present scope="session" name="dadosSessao">

<logic:present scope="session" name="projeto">
<html>
      <head>
      <title>
            HOPI - Projeto
      </title>
            <link         href="xstylesheet.css"         rel="stylesheet"
type="text/css">

            <script language="JavaScript">
                  function popUpAjudaProjeto()
                  {
                        ajudaProjeto   =   window.open("ajudaProjeto.jsp",
"AjudaProjeto",
'toolbar=0,scrollbars=0,location=0,statusbar=0,menubar=0,resizable=1,wid
th=500,height=200');
                  }

                  function popUpSobre()
                  {
                        sobre  =  window.open("sobre.jsp",  "SobreoHOPI",
'toolbar=0,scrollbars=0,location=0,statusbar=0,menubar=0,resizable=1,wid
th=500,height=200');
                  }

                  function novoAtor()
                  {
                      document.myForm.action = "novoAtor.do";
                      document.myForm.submit();
                  }

                  function abraAtor(id)
                  {
                      document.myForm.idAtor.value = id;
                      document.myForm.action = "abraAtor.do";
                      document.myForm.submit();
                  }

                  function excluaAtor(id)
                  {
                      document.myForm.idAtor.value = id;
                      document.myForm.action = "excluaAtor.do";
                      document.myForm.submit();
                  }

                  function novoRequisitoFuncional()
```

```
                {
                    document.myForm.action                                  =
"novoRequisitoFuncional.do";
                    document.myForm.submit();
                }

                function abraRequisitoFuncional(id)
                {
                    document.myForm.idRequisitoFuncional.value = id;
                    document.myForm.action                                  =
"abraRequisitoFuncional.do";
                    document.myForm.submit();
                }

                function excluaRequisitoFuncional(id)
                {
                    document.myForm.idRequisitoFuncional.value = id;
                    document.myForm.action                                  =
"excluaRequisitoFuncional.do";
                    document.myForm.submit();
                }

                function novoRequisitoNaoFuncional()
                {
                    document.myForm.action                                  =
"novoRequisitoNaoFuncional.do";
                    document.myForm.submit();
                }

                function abraRequisitoNaoFuncional(id)
                {
                    document.myForm.idRequisitoFuncional.value = id;
                    document.myForm.idRequisitoNaoFuncional.value = id;
                    document.myForm.action                                  =
"abraRequisitoNaoFuncional.do";
                    document.myForm.submit();
                }

                function excluaRequisitoNaoFuncional(id)
                {
                    document.myForm.idRequisitoNaoFuncional.value = id;
                    document.myForm.action                                  =
"excluaRequisitoNaoFuncional.do";
                    document.myForm.submit();
                }

                function novoCasoDeUso()
                {
                    document.myForm.action = "novoCasoDeUso.do";
                    document.myForm.submit();
                }

                function abraCasoDeUso(id)
                {
                    document.myForm.idCasoDeUso.value = id;
                    document.myForm.action = "abraCasoDeUso.do";
                    document.myForm.submit();
                }

                function excluaCasoDeUso(id)
                {
```

```
                    document.myForm.idCasoDeUso.value = id;
                    document.myForm.action = "excluaCasoDeUso.do";
                    document.myForm.submit();
                }

                function cancelar()
                {
                    document.myForm.action = "principal.do";
                    document.myForm.submit();
                }

                function salvar()
                {
                    document.myForm.action = "salveProjeto.do";
                    document.myForm.submit();
                }
            </script>
      </head>

      <body>
      <form name="myForm" method="post">
            <html:hidden name="projeto" property="id"/>
            <input type="hidden" name="idAtor">
            <input type="hidden" name="idRequisitoFuncional">
            <input type="hidden" name="idRequisitoNaoFuncional">
            <input type="hidden" name="idCasoDeUso">
            <table class="titulo">
            <!--############################################################-->
                  <!--# TITULO: HOPI e informações sobre a página aberta #-->
            <!--############################################################-->
                  <tr>
                        <td>
                              <table class="transparente">
                                    <tr height="50">
                                    <td width="10%">
                                                <font     class="titulo">HOPI </font>
                                          </td>
                                          <td width="90%">
                                                <table class="transparente">
                                                <tr class="linhaItemSecao">
                                                      <td class="colunaTipoItemSecao">
                                                            <font class="miniVerdeEscuro">Usuário Logado: </font>
                                                      </td>

                                                      <td class="colunaValorItemSecao">
                                                            <font class="menu">

      <bean:write            scope="session"           name="dadosSessao" property="nomeCompleto"/>
                                                            </font>
                                                      </td>
```

```
                                                </tr>
                                                </table>
                                    </td>
                                    </tr>
                              </table>
                        </td>
                  </tr>
                  <!--
##########################################################-->
                  <!--# NAVEGAÇÃO: apresenta cada etapa da análise e
projeto #-->
                  <!--
##########################################################-->
                  <tr>
                        <td>
                              <table class="navegacao">
                                    <tr class="contornoNavegacao">
                                          <td colspan="5"></td>
                                    </tr>

                                    <tr class="linhaNavegacao">
                                          <td width=2%></td>

                                          <td class="etapas">
                                          <font
class="navegacaoEtapasMiniSelecionada">
                                                 Planejamento
                                          </font>
                                          <font class="navegacaoEtapas">
                                             1 2
                                          </font>
                                          </td>

                                          <td width=3%></td>

                                          <td class="etapas">
                                          <font
class="navegacaoEtapasMini">
                                                 Análise
                                          </font>
                                          <font class="navegacaoEtapas">
                                                 1 2 3
                                          </font>
                                          </td>

                                          <td width=3%></td>

                                          <td class="etapas">
                                          <font
class="navegacaoEtapasMini">
                                                 Projeto
                                          </font>
                                          <font class="navegacaoEtapas">
                                                 1 2 3
                                          </font>
                                          </td>

                                          <td width="32%"></td>
                                    </tr>

                                    <tr class="contornoNavegacao">
```

```
                                    <td colspan="5"></td>
                              </tr>
                        </table>
                  </td>
            </tr>
            <!--
##########################################################-->
            <!--#   MENUS:   apresenta   menus   de   acesso   rápido
#-->
            <!--
##########################################################-->
            <tr>
                  <td>
                        <table class="menu">
                              <tr>
                                    <td>
                                          <font class="menu">
                                                 <a
class='linkMenu' href='principal.do'>Principal</a> |
                                                 <a
class='linkMenu' href='listeCategorias.do'>Categorias</a> |
                                                 <a
class='linkMenu' href='listeUsuarios.do'>Usuários</a> |
                                                 <a
class='linkMenu' href="javascript:popUpAjudaProjeto()">Ajuda</a> |
                                                 <a
class='linkMenu' href="javascript:popUpSobre()">Sobre</a> |
                                                 <a
class='linkMenu' href='logout.do'>Logout</a>
                                          </font>
                                    </td>
                              </tr>
                        </table>
                  </td>
            </tr>
            <!--
##########################################################-->
            <!--# CADASTRO: apresenta o cadastro para entrada dos
dados #-->
            <!--
##########################################################-->
            <tr height="10"><td></td></tr>
            <tr>
                  <td>
                        <table class="transparente">
                              <tr>
                              <td>
                                    <font
class="nomePagina">Projeto</font>
                              </td>
                              </tr>
                        </table>
                  </td>
            </tr>
            <tr height="15"><td></td></tr>
            <tr>
                  <td>
                  <table class="transparente">
                        <tr width="100%">
                                    <td                width="100%"
colspan="3"></td>
```

```
                                          <html:errors/>
                                    </td>
                              </tr>

                              <tr width="100%">
                                    <td width="50%">
                                    <font
class="normal">Nome</font>
                                    </td>
                                    <td width="25%">
                                          <font  class="normal">Data
de Início</font>
                                    </td>
                                    <td width="25%">
                                          <font  class="normal">Data
de Conclusão</font>
                                    </td>
                              </tr>

                              <tr width="100%">
                                    <td>
                                          <html:text
styleClass="edit" name="projeto" property="nome" size="58"/>
                                    </td>
                                    <td>
                                          <html:text
styleClass="edit"   name="projeto"   property="dataInicial"   size="19"
maxlength="19"/>
                                    </td>
                                    <td>
                                          <html:text
styleClass="edit"    name="projeto"    property="dataFinal"    size="19"
maxlength="19"/>
                                    </td>
                              </tr>
                        </table>
                  </td>
            </tr>
            <tr height="15"><td></td></tr>
            <tr>
                  <td>
                  <table class="transparente">
                              <tr>
                                    <td width="50%">
                                          <font
class="normal">Sumário Executivo</font>
                                    </td>

                                    <logic:notEqual  name="projeto"
property="id" value="-1">
                                          <td width="40%">
                                                <font
class="normal">Atores</font>
                                          </td>
                                          <td width="10%">
                                                <input
class="pequeno"              type="button"              value="INCLUIR"
onClick="javascript:novoAtor()">
                                          </td>
                                    </logic:notEqual>
```

```
                                        <logic:equal     name="projeto"
property="id" value="-1">
                                              <td            width="50%"
colspan="2"></td>
                                        </logic:equal>
                                  </tr>
                            </table>
                      </td>
                </tr>

                <tr>
                      <td>
                      <table class="transparente">
                                  <tr valign="top">
                                        <td width="49.1%" valign="top">
                                              <html:textarea
styleClass="edit"  name="projeto"  property="sumarioExecutivo"  cols="70"
rows="5"/>
                                        </td>

                                        <logic:notEqual  name="projeto"
property="id" value="-1">
                                              <td width="48%"   >
                                                    <table
class="transparente">

      <logic:present name="atores" scope="session">

      <logic:iterate id="ator" name="atores" type="VOAtor" indexId="i">
                                                                        <%
                                                                        if
(i.intValue() % 2 == 0)
                                                                        {

      %>

      <tr class="linhaPar">

      <%
                                                                        }
                                                                        else
                                                                        {

      %>

      <tr class="linhaImpar">

      <%
                                                                        }
                                                                        %>


      <td width="90%">

      <font class="lista">

            <bean:write name="ator" property="nome"/>

      </font>

      </td>
```

```
      <td width="10%">

      <font class="lista">

            <a     href="javascript:abraAtor('<bean:write     name="ator"
property="id"/>')">

                  <img          src="xfig-seta.gif"          alt="Editar"
onMouseOver="javascript:src='xfig-setaOver.gif'"
onMouseOut="javascript:src='xfig-seta.gif'">

            </a>

             

            <a    href="javascript:excluaAtor('<bean:write    name="ator"
property="id"/>')">

                  <img          src="xfig-xis.gif"          alt="Excluir"
onMouseOver="javascript:src='xfig-xisOver.gif'"
onMouseOut="javascript:src='xfig-xis.gif'">

            </a>

      </font>

      </td>
                                                                        </tr>

      </logic:iterate>

      </logic:present>
                                                      </table>
                                                </td>
                                                <td width="2.9%"></td>
                                          </logic:notEqual>

                                          <logic:equal     name="projeto"
property="id" value="-1">
                                                <td width="48%"></td>
                                          </logic:equal>
                                    </tr>
                              </table>
                        </td>
                  </tr>

                  <logic:notEqual  name="projeto"  property="id"  value="-
1">
                        <tr height="15"><td></td></tr>
                        <tr>
                              <td>
                              <table class="transparente">
                                          <td width="39%">
                                                <font
class="normal">Requisitos Funcionais</font>
                                    </td>

                                    <td width="11%">
```

```
                                        <input          class="pequeno"
type="button"                                           value="INCLUIR"
onClick="javascript:novoRequisitoFuncional()">
                                  </td>

                                  <td width="40%">
                                        <font class="normal">Requisitos
Não-Funcionais</font>
                                  </td>

                                  <td width="10%">
                                              <input    class="pequeno"
type="button" value="INCLUIR">
                                        </td>
                                  </table>
                            </td>
                      </tr>
                      <tr>
                            <td>
                            <table class="transparente">
                                  <%
                                  ArrayList   requisitosFuncionais   =
(ArrayList)session.getAttribute("requisitosFuncionais");
                                  ArrayList  requisitosNaoFuncionais  =
(ArrayList)session.getAttribute("requisitosNaoFuncionais");

                                  String requisitoFuncionalId = "";
                                  String requisitoFuncionalNome = "";
                                  String requisitoNaoFuncionalId = "";
                                  String   requisitoNaoFuncionalNome   =
"";

                                  VORequisitoFuncional
requisitoFuncional;
                                  VORequisitoNaoFuncional
requisitoNaoFuncional;

                                  boolean   linhaRequisitoFuncional   =
true;
                                  boolean  linhaRequisitoNaoFuncional  =
true;

                                  int max = 0;

                                  if   (requisitosFuncionais.size()   >
requisitosNaoFuncionais.size())
                                  {
                                        max                           =
requisitosFuncionais.size();
                                  }
                                  else
                                  {
                                        max                           =
requisitosNaoFuncionais.size();
                                  }

                                  for (int i=0; i<max; i++)
                                  {
                                        if             (i             <
requisitosFuncionais.size())
                                        {
```

```
                                                requisitoFuncional      =
(VORequisitoFuncional)requisitosFuncionais.get(i);
                                                requisitoFuncionalId    =
requisitoFuncional.getId().toString();
                                                requisitoFuncionalNome  =
requisitoFuncional.getNome();
                                          }
                                          else
                                          {
                                                requisitoFuncionalId    =
"";
                                                requisitoFuncionalNome  =
"";
                                                linhaRequisitoFuncional =
false;
                                          }

                                          if             (i             <
requisitosNaoFuncionais.size())
                                          {
                                                requisitoNaoFuncional   =
(VORequisitoNaoFuncional)requisitosNaoFuncionais.get(i);
                                                requisitoNaoFuncionalId =
requisitoNaoFuncional.getId().toString();
                                                requisitoNaoFuncionalNome
= requisitoNaoFuncional.getNome();
                                          }
                                          else
                                          {
                                                requisitoNaoFuncionalId =
"";
                                                requisitoNaoFuncionalNome
= "";

      linhaRequisitoNaoFuncional = false;
                                          }

                                          if (i % 2 == 0)
                                          {
                                                %>
                                                <tr class="linhaPar">
                                                <%
                                          }
                                          else
                                          {
                                          %>
                                                <tr class="linhaImpar">
                                                <%
                                          }


                                          if (linhaRequisitoFuncional)
                                          {
                                                %>
                                                <td width="43%">
                                                      <font
class="lista">

      F<%=requisitoFuncionalId%>. <%=requisitoFuncionalNome%>
                                                      </font>
                                                </td>
```

```
                                          <td width="4%">
                                                <font
class="lista">
                                                      <a
href="javascript:abraRequisitoFuncional('<%=requisitoFuncionalId%>')">
                                                            <img
src="xfig-seta.gif"    alt="Editar"    onMouseOver="javascript:src='xfig-
setaOver.gif'" onMouseOut="javascript:src='xfig-seta.gif'">
                                                      </a>

                                                       
                                                      <a
href="javascript:excluaRequisitoFuncional('<%=requisitoFuncionalId%>')">
                                                            <img
src="xfig-xis.gif"    alt="Excluir"    onMouseOver="javascript:src='xfig-
xisOver.gif'" onMouseOut="javascript:src='xfig-xis.gif'">
                                                      </a>
                                                </font>
                                          </td>
                                          <%
                                    }
                                    else
                                    {
                                          %>
                                          <td            width="43%"
bgcolor="#FFFFFF"></td>
                                          <td             width="4%"
bgcolor="#FFFFFF"></td>
                                          <%
                                    }
                                    %>
                              <td                         width="3%"
bgcolor="#FFFFFF"></td>
                                    <%
                              if (linhaRequisitoNaoFuncional)
                              {
                                    %>
                                          <td width="44%">
                                                <font
class="lista">

     NF<%=requisitoNaoFuncionalId%>. <%=requisitoNaoFuncionalNome%
>
                                                </font>
                                          </td>
                                          <td width="4.1%">
                                                <font
class="lista">
                                                      <a
href="javascript:abraRequisitoNaoFuncional('<%=requisitoNaoFuncionalId%>
')">
                                                            <img
src="xfig-seta.gif"    alt="Editar"    onMouseOver="javascript:src='xfig-
setaOver.gif'" onMouseOut="javascript:src='xfig-seta.gif'">
                                                      </a>

                                                       
                                                      <a
href="javascript:excluaRequisitoNaoFuncional('<%=requisitoNaoFuncionalId
%>')">
```

```
                                                                  <img
src="xfig-xis.gif"    alt="Excluir"    onMouseOver="javascript:src='xfig-
xisOver.gif'" onMouseOut="javascript:src='xfig-xis.gif'">
                                                                  </a>
                                                            </font>
                                                      </td>
                                                      <%
                                                }
                                                else
                                                {
                                                      %>
                                                      <td            width="44%"
bgcolor="#FFFFFF"></td>
                                                      <td           width="4.1%"
bgcolor="#FFFFFF"></td>
                                                      <%
                                                }
                                                %>
                                                <td                   width="2%"
bgcolor="white"></td>
                                          </tr>
                                          <%
                                          }
                                          %>
                                          </table>
                                    </td>
                              </tr>
                              <tr height="17"><td></td></tr>

                              <tr>
                                    <td>
                                    <table class="transparente">
                                                <td width="39%">
                                                      <font  class="normal">Caso
de Uso</font>
                                          </td>

                                          <td width="11%">
                                                <input          class="pequeno"
type="button" value="INCLUIR" onClick="javascript:novoCasoDeUso()">
                                          </td>

                                          <td width="40%">
                                          </td>

                                          <td width="10%">
                                                </td>
                                          </table>
                                    </td>
                              </tr>
                              <tr>
                                    <td>
                                    <table class="transparente">
                                          <%
                                          ArrayList        casosDeUso        =
(ArrayList)session.getAttribute("casosDeUso");

                                          String casoDeUsoId = "";
                                          String casoDeUsoReferencia = "";
                                          String casoDeUsoNome = "";
```

```
                                    VOCasoDeUso casoDeUso;
                                    for  (int  i=0;  i<casosDeUso.size();
i++)
                                    {
                                          casoDeUso                     =
(VOCasoDeUso)casosDeUso.get(i);
                                          casoDeUsoId                   =
casoDeUso.getId().toString();
                                          casoDeUsoReferencia           =
casoDeUso.getReferencia();
                                          casoDeUsoNome                 =
casoDeUso.getNome();

                                          if (i % 2 == 0)
                                          {
                                                %>
                                                <tr class="linhaPar">
                                                <%
                                          }
                                          else
                                          {
                                          %>
                                                <tr class="linhaImpar">
                                                <%
                                          }


                                          %>
                                          <td width="43%">
                                                <font class="lista">

     UC<%=casoDeUsoId%>. <%=casoDeUsoNome%>
                                                </font>
                                          </td>
                                          <td width="4%">
                                                <font class="lista">
                                                      <a
href="javascript:abraCasoDeUso('<%=casoDeUsoId%>')">
                                                            <img
src="xfig-seta.gif"    alt="Editar"    onMouseOver="javascript:src='xfig-
setaOver.gif'" onMouseOut="javascript:src='xfig-seta.gif'">
                                                      </a>

                                                       
                                                      <a
href="javascript:excluaCasoDeUso('<%=casoDeUsoId%>')">
                                                            <img
src="xfig-xis.gif"    alt="Excluir"    onMouseOver="javascript:src='xfig-
xisOver.gif'" onMouseOut="javascript:src='xfig-xis.gif'">
                                                      </a>
                                                </font>
                                          </td>
                                    <td                           width="3%"
bgcolor="#FFFFFF"></td>
                                          <td                    width="44%"
bgcolor="#FFFFFF"></td>
                                          <td                   width="4.1%"
bgcolor="#FFFFFF"></td>
                                          <td                     width="2%"
bgcolor="white"></td>
```

```
                                    </tr>
                                    <%
                                    }
                                    %>
                                    </table>
                              </td>
                        </tr>
                        <tr height="17"><td></td></tr>
                  </logic:notEqual>

                  <tr>
                        <td>
                        <table class="transparente">
                                    <tr>
                                    <td align="right" width="98%">
                                                <input      class="normal"
type="button"                                          value="CANCELAR"
onClick="javascript:cancelar()">  
                                                <input      class="normal"
type="button" value="SALVAR" onClick="javascript:salvar()">
                                          </td>
                                          <td width="2%"></td>
                                    </tr>
                              </table>
                        </td>
                  </tr>
                  <tr height="7"><td></td></tr>
            </table>
      </form>
      </body>
</html>

</logic:present>

<logic:notPresent name="projeto" scope="session">
<meta http-equiv="refresh" content="0; URL=acessoNaoAutorizado.do">
</logic:notPresent>

</logic:present>

<logic:notPresent name="dadosSessao" scope="session">
<meta http-equiv="refresh" content="0; URL=acessoNaoAutorizado.do">
</logic:notPresent>
```

requisitoFuncional.jsp

```
<%@ page import="org.hopi.ejb.entity.VOCasoDeUso"%>
<%@ page import="org.hopi.ejb.entity.VORequisitoNaoFuncional"%>
<%@ taglib uri="WEB-INF/lib/struts/struts-html.tld" prefix="html" %>
<%@ taglib uri="WEB-INF/lib/struts/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="WEB-INF/lib/struts/struts-bean.tld" prefix="bean" %>

<logic:present name="dadosSessao" scope="session">

<html>
      <head>
            <title>
                  HOPI - Requisito Funcional
            </title>
            <link         href="xstylesheet.css"         rel="stylesheet"
type="text/css">
            <script language="JavaScript">
                  function popUpAjudaRequisitoFuncional()
                  {
                        ajudaProjeto                                        =
window.open("ajudaRequisitoFuncional.jsp",    "AjudaRequisitoFuncional",
'toolbar=0,scrollbars=0,location=0,statusbar=0,menubar=0,resizable=1,wid
th=500,height=200');
                  }

                  function popUpSobre()
                  {
                        sobre  =  window.open("sobre.jsp",  "SobreoHOPI",
'toolbar=0,scrollbars=0,location=0,statusbar=0,menubar=0,resizable=1,wid
th=500,height=200');
                  }

                  function novoCasoDeUsoRelacionado(id)
                  {
                        document.myForm.idCasoDeUsoRelacionado.value    =
id;
                        document.myForm.action                          =
"incluaCasoDeUsoRelacionado.do";
                        document.myForm.submit();
                  }

                  function abraCasoDeUsoRelacionado(id)
                  {
                        document.myForm.idCasoDeUsoRelacionado.value    =
id;
                        document.myForm.action = "abraCasoDeUso.do";
                        document.myForm.submit();
                  }

                  function excluaCasoDeUsoRelacionado(id)
                  {
                        document.myForm.idCasoDeUsoRelacionado.value    =
id;
                        document.myForm.action                          =
"excluaCasoDeUsoRelacionado.do";
                        document.myForm.submit();
                  }

                  function novoRequisitoNaoFuncionalAssociado(id)
                  {
```

```
                        document.myForm.idRequisitoNaoFuncional.value   =
id;
                        document.myForm.action                          =
"novoRequisitoNaoFuncionalAssociado.do";
                        document.myForm.submit();
                  }

                  function abraRequisitoNaoFuncionalAssociado(id)
                  {
                        document.myForm.idRequisitoNaoFuncional.value   =
id;
                        document.myForm.action                          =
"abraRequisitoNaoFuncionalAssociado.do";
                        document.myForm.submit();
                  }

                  function excluaRequisitoNaoFuncionalAssociado(id)
                  {
                        document.myForm.idRequisitoNaoFuncional.value   =
id;
                        document.myForm.action                          =
"excluaRequisitoNaoFuncionalAssociado.do";
                        document.myForm.submit();
                  }

                  function cancelar()
                  {
                      document.myForm.action = "abraProjeto.do";
                      document.myForm.submit();
                  }

                  function salvar()
                  {
                      document.myForm.action                            =
"salveRequisitoFuncional.do";
                        document.myForm.submit();
                  }
            </script>
      </head>

      <body>
            <form name="myForm" method="post">
                  <html:hidden name="requisitoFuncional" property="id"/>
                  <html:hidden                  name="requisitoFuncional"
property="idProjeto"/>
                  <input type="hidden" name="idRequisitoNaoFuncional">

                  <table class="titulo">
                        <!--
############################################################-->
                        <!--# TITULO: HOPI e informações sobre a página
aberta   #-->
                        <!--
############################################################-->
                        <tr>
                              <td>
                                    <table class="transparente">
                                          <tr height="50">
                                          <td width="10%">
                                                      <font
class="titulo">
```

```
                                                        HOPI
                                                    </font>
                                                </td>

                                            <td width="90%">
                                                    <table
class="transparente">
                                                    <tr
class="linhaItemSecao">
                                                        <td
class="colunaTipoItemSecao">
                                                            <font
class="miniVerdeEscuro">

      Usuário Logado:
                                                            </font>
                                                        </td>

                                                        <td
class="colunaValorItemSecao">
                                                            <font
class="menu">

      <bean:write            scope="session"            name="dadosSessao"
property="nomeCompleto"/>
                                                            </font>
                                                        </td>
                                                    </tr>

                                                    <tr
class="linhaItemSecao">
                                                        <td
class="colunaTipoItemSecao">
                                                            <font
class="miniVerdeEscuro">

      Projeto Selecionado:
                                                            </font>
                                                        </td>

                                                        <td
class="colunaValorItemSecao">
                                                            <font
class="menu">

      <bean:write scope="session" name="projeto" property="nome"/>
                                                            </font>
                                                        </td>
                                                    </tr>

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

                                    <!--
############################################################-->
                                    <!--# NAVEGAÇÃO: apresenta cada etapa da análise
e projeto #-->
```

```
                        <!--
##########################################################-->
                        <tr>
                              <td>
                                    <table class="navegacaoOculta">
                                          <tr class="contornoNavegacao">
                                                <td colspan="5"></td>
                                          </tr>

                                          <tr class="linhaNavegacao">
                                                <td width=2%></td>

                                                <td class="etapas">
                                                <font
class="navegacaoEtapasMini">

       Planejamento
                                                </font>
                                                <font
class="navegacaoEtapaSelecionada">
                                                             1
                                                      </font>
                                                <font
class="navegacaoEtapas">
                                                             2
                                                      </font>
                                                </td>

                                                <td width=3%></td>

                                                <td class="etapas">
                                                <font
class="navegacaoEtapasMini">
                                                             Análise
                                                </font>
                                                <font
class="navegacaoEtapas">
                                                             1 2 3
                                                      </font>
                                                </td>

                                                <td width=3%></td>

                                                <td class="etapas">
                                                <font
class="navegacaoEtapasMini">
                                                             Projeto
                                                </font>
                                                <font
class="navegacaoEtapas">
                                                             1 2 3
                                                      </font>
                                                </td>

                                                <td width="32%"></td>
                                          </tr>

                                          <tr class="contornoNavegacao">
                                                <td colspan="5"></td>
                                          </tr>
                                    </table>
```

```
                        </td>
                  </tr>

                  <!--
##########################################################-->
                  <!--# MENUS: apresenta menus de acesso rápido
#-->
                  <!--
##########################################################-->
                  <tr>
                        <td>
                              <table class="menu">
                                    <tr>
                                          <td>
                                                <font class="menu">
                                                       <a
class='linkMenu' href='principal.do'>Principal</a> |
                                                       <a
class='linkMenu' href='abraProjeto.do'>Projeto</a> |
                                                       <a
class='linkMenu' href='listeCategorias.do'>Categorias</a> |
                                                       <a
class='linkMenu' href='listeUsuarios.do'>Usuários</a> |
                                                       <a
class='linkMenu'
href="javascript:popUpAjudaRequisitoFuncional()">Ajuda</a> |
                                                       <a
class='linkMenu' href="javascript:popUpSobre()">Sobre</a> |
                                                       <a
class='linkMenu' href='logout.do'>Logout</a>
                                                </font>
                                          </td>
                                    </tr>
                              </table>
                        </td>
                  </tr>

                  <!--
##############################################################-->
                  <!--# CADASTRO: apresenta o cadastro para entrada
dos dados #-->
                  <!--
##############################################################-->
                  <tr height="10"><td></td></tr>
                  <tr>
                        <td>
                              <table class="transparente">
                                    <tr>
                                    <td>
                                          <font class="nomePagina">
                                                Requisito Funcional
                                          </font>
                                    </td>
                                    </tr>
                              </table>
                        </td>
                  </tr>
                  <tr height="15"><td></td></tr>
                  <tr>
                        <td>
                        <table class="transparente">
```

```
                                    <tr width="100%">
                                          <td width="5%"></td>
                                          <td            width="90%"
colspan="3">
                                                <html:errors/>
                                          </td>
                                          <td width="5%"></td>
                                    </tr>
                                    <tr width="100%">
                                          <td width="5%"></td>
                                          <td width="44%">
                                          <font class="normal">
                                                Nome
                                          </font>
                                          </td>
                                          <td width="2%"></td>
                                          <td width="44%"></td>
                                          <td width="5%"></td>
                                    </tr>
                                    <tr width="100%">
                                          <td></td>
                                          <td>
                                                <html:text
styleClass="edit" name="requisitoFuncional" property="nome" size="50"/>
                                          </td>
                                          <td></td>
                                          <td>
                                          <html:checkbox
name="requisitoFuncional" property="oculto"/>
                                          <font class="normal">
                                                Marque     se     o
requisito funcional for oculto
                                          </font>
                                          </td>
                                          <td></td>
                                    </tr>
                              </table>
                        </td>
                  </tr>
                  <tr height="15"><td></td></tr>

                  <tr>
                        <td>
                        <table class="transparente">
                                    <tr width="100%">
                                          <td width="5%"></td>
                                          <td width="44%">
                                          <font class="normal">
                                                Descrição
                                          </font>
                                          </td>

                                          <logic:notEqual
name="requisitoFuncional" property="id" value="-1">
                                                <td
width="2%"></td>
                                                <td width="44%">
                                                <font
class="normal">
                                                      Casos  de  Uso
Relacionados
```

```
                                                      </font>
                                                      </td>
                                                      <td
width="5%"></td>
                                                </logic:notEqual>

                                                <logic:equal
name="requisitoFuncional" property="id" value="-1">
                                                      <td     width="51%"
colspan="3"></td>
                                                </logic:equal>
                                          </tr>
                                          <tr width="100%" valign="top">
                                                <td></td>
                                                <td>
                                                      <html:textarea
name="requisitoFuncional" property="descricao" cols="60" rows="5"/>
                                                </td>
                                                <logic:notEqual
name="requisitoFuncional" property="id" value="-1">
                                                      <td></td>
                                                      <td>
                                                      <table>
                                                                  <tr
cellspacing="0" width="100%">

      <td width="84%">

      <select class="combobox" cols="50">

            <logic:present name="casosDeUso" scope="session">

            <logic:iterate                                  id="casoDeUso"
name="casosDeUsoNaoRelacionados" type="VOCasoDeUso" indexId="i">

                  <option><bean:write                   name="casoDeUso"
property="nome"/></option>

            </logic:iterate>

            </logic:present>

      </select>
                                                                  </td>

                                                                  <td
width="16%">

      <input class="pequeno" type="button" value="INCLUIR">
                                                                  </td>
                                                                  </tr>
                                                            </table>

                                                      <table
cellspacing="0" width="100%">

      <logic:present name="casosDeUso" scope="session">

      <logic:iterate id="casoDeUso" name="casosDeUso" type="VOCasoDeUso"
indexId="i">
```

```
      <%
      if (i.intValue() % 2 == 0)
                                                                        {
      %>
      <tr class="linhaPar">
      <%
                                                                        }
      else
                                                                        {
      %>
      <tr class="linhaImpar">
      <%
                                                                        }
      %>
      <td width="89%">
            <font class="lista">
                  <bean:write name="casoDeUso" property="nome"/>
            </font>
      </td>
      <td width="11%">
            <font class="lista">
                  <a          href="javascript:abraCasoDeUso('<bean:write
name="casoDeUso" property="id"/>')">
                        <img       src="xfig-seta.gif"       alt="Editar"
onMouseOver="javascript:src='xfig-setaOver.gif'"
onMouseOut="javascript:src='xfig-seta.gif'">
                  </a>
                   
                  <a        href="javascript:excluaCasoDeUso('<bean:write
name="casoDeUso" property="id"/>')">
                        <img       src="xfig-xis.gif"       alt="Excluir"
onMouseOver="javascript:src='xfig-xisOver.gif'"
onMouseOut="javascript:src='xfig-xis.gif'">
                  </a>
            </font>
```

```
      </td>

      </tr>

      </logic:iterate>

      </logic:present>
                                                            </table>
                                                      </td>
                                                      <td></td>
                                                </logic:notEqual>

                                                <logic:equal
name="requisitoFuncional" property="id" value="-1">
                                                      <td      width="51%"
colspan="3"></td>
                                                </logic:equal>
                                          </tr>
                                    </table>
                              </td>
                        </tr>
                        <tr height="15"><td></td></tr>

                        <logic:notEqual         name="requisitoFuncional"
property="id" value="-1">
                              <tr>
                                    <td>
                                    <table class="transparente">
                                          <tr width="100%">
                                                <td
width="5%"></td>
                                                <td width="82%">
                                                <font
class="normal">

      Requisitos Não-Funcionais Associados
                                                      </font>
                                                </td>
                                                <td width="8%">
                                                <input
class="pequeno"               type="button"              value="INCLUIR"
onClick="javascript:novoRequisitoNaoFuncionalAssociado()">
                                                </td>
                                                <td
width="5%"></td>
                                                </tr>
                                          </table>
                                    </td>
                              </tr>
                              <tr>
                                    <td>
                                    <table class="transparente">
                                          <tr
class="linhaCabecalho">
                                                <td      width="5%"
bgcolor="#FFFFFF"></td>
                                                <td width="25%">
                                                      <font
class="cabecalho">
                                                            Nome
```

```
                                                      </font>
                                                </td>
                                                <td width="14%">
                                                      <font
class="cabecalho">

      Categoria
                                                      </font>
                                                </td>
                                                <td width="41%">
                                                      <font
class="cabecalho">

      Descrição
                                                      </font>
                                                </td>
                                                <td width="3%">
                                                      <font
class="cabecalho">

        O
                                                      </font>
                                                </td>
                                                <td width="3%">
                                                      <font
class="cabecalho">
                                                             P
                                                      </font>
                                                </td>

                                                <td
width="4%"></td>
                                                      <td       width="5%"
bgcolor="#FFFFFF"></td>
                                          </tr>

                                          <logic:present
name="requisitosNaoFuncionais" scope="session">
                                          <logic:iterate
id="requisitoNaoFuncional"               name="requisitosNaoFuncionais"
type="VORequisitoNaoFuncional" indexId="i">
                                                <%
                                                if  (i.intValue()  %
2 == 0)
                                                {
                                                      %>
                                                      <tr
class="linhaPar">
                                                      <%
                                                }
                                                else
                                                {
                                                      %>
                                                      <tr
class="linhaImpar">
                                                      <%
                                                }
                                                %>

                                                      <td
bgcolor="#FFFFFF"></td>
```

```
                                                                <td>
                                                                      <font
class="lista">

      <bean:write name="requisitoNaoFuncional" property="nome"/>
                                                                      </font>
                                                                </td>
                                                                <td>
                                                                      <font
class="lista">

      <bean:write name="requisitoNaoFuncional" property="idCategoria"/>
                                                                      </font>
                                                                </td>
                                                                <td>
                                                                      <font
class="lista">

      <bean:write name="requisitoNaoFuncional" property="descricao"/>
                                                                      </font>
                                                                </td>
                                                                <td
align="center">
                                                                      <font
class="cabecalho">

      <logic:equal  name="requisitoNaoFuncional"  property="obrigatorio"
value="true">

      <img     src="xfig-true.gif"     alt="Requisito     Não-Funcional
Obrigatório">

      </logic:equal>

      <logic:equal  name="requisitoNaoFuncional"  property="obrigatorio"
value="false">

      <img src="xfig-false.gif" alt="Requisito Não-Funcional Desejável">

      </logic:equal>
                                                                      </font>
                                                                </td>
                                                                <td>
                                                                      <font
class="cabecalho">

      <logic:equal   name="requisitoNaoFuncional"   property="permanente"
value="true">

      <img src="xfig-true.gif" alt="Requisito Não-Funcional Permanente">

      </logic:equal>

      <logic:equal   name="requisitoNaoFuncional"   property="permanente"
value="false">

      <img     src="xfig-false.gif"     alt="Requisito     Não-Funcional
Transitório">

      </logic:equal>
                                                                      </font>
```

```
                                                                  </td>
                                                                  <td>
                                                                        <font
class="lista">

      <a
href="javascript:abraRequisitoNaoFuncionalAssociado('<bean:write
name="requisitoNaoFuncional" property="id"/>')">

      <img                src="xfig-seta.gif"                alt="Editar"
onMouseOver="javascript:src='xfig-setaOver.gif'"
onMouseOut="javascript:src='xfig-seta.gif'">

      </a>

       

      <a
href="javascript:excluaRequisitoNaoFuncionalAssociado('<bean:write
name="requisitoNaoFuncional" property="id"/>')">

      <img                src="xfig-xis.gif"                alt="Excluir"
onMouseOver="javascript:src='xfig-xisOver.gif'"
onMouseOut="javascript:src='xfig-xis.gif'">

      </a>
                                                                        </font>
                                                                  </td>
                                                                  <td
width="5%" bgcolor="#FFFFFF"></td>
                                                            </tr>
                                                      </logic:iterate>
                                                      </logic:present>
                                                </table>
                                          </td>
                                    </tr>
                                    <tr height="15"><td></td></tr>
                              </logic:notEqual>
                              <tr>
                                    <td>
                                    <table class="transparente">
                                                <tr>
                                                <td align="right" width="95%">
                                                      <input
class="normal"              type="button"              value="CANCELAR"
onClick="javascript:cancelar()">  
                                                      <input
class="normal"               type="button"               value="SALVAR"
onClick="javascript:salvar()">
                                                </td>
                                                <td width="5%"></td>
                                                </tr>
                                          </table>
                                    </td>
                              </tr>
                              <tr height="7"><td></td></tr>
                        </table>
                  </form>
      </body>
</html>
```

```
</logic:present>

<logic:notPresent name="dadosSessao" scope="session">
      <meta                http-equiv="refresh"                content="0;
URL=acessoNaoAutorizado.do">
</logic:notPresent>
```

requisitoNaoFuncional.jsp

```
<%@ page import="org.hopi.ejb.entity.VORequisitoNaoFuncional"%>
<%@ taglib uri="WEB-INF/lib/struts/struts-html.tld" prefix="html" %>
<%@ taglib uri="WEB-INF/lib/struts/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="WEB-INF/lib/struts/struts-bean.tld" prefix="bean" %>

<logic:present name="dadosSessao" scope="session">

<html>
      <head>
    <title> HOPI - Requisito Não-Funcional Associado </title>
           <link         href="xstylesheet.css"         rel="stylesheet"
type="text/css">
           <script language="JavaScript">
                 function popUpAjudaRequisitoNaoFuncional()
                 {
                       ajudaAtor                                          =
window.open("ajudaRequisitoNaoFuncional.jsp",
"AjudaRequisitoNaoFuncional",
'toolbar=0,scrollbars=0,location=0,statusbar=0,menubar=0,resizable=1,wid
th=500,height=200');
                 }

                 function popUpSobre()
                 {
                       sobre  =  window.open("sobre.jsp",  "SobreoHOPI",
'toolbar=0,scrollbars=0,location=0,statusbar=0,menubar=0,resizable=1,wid
th=500,height=200');
                 }

                 function cancelar()
                 {
                       document.myForm.action                             =
"abraRequisitoNaoFuncionalAssociado.do";
                       document.myForm.submit();
                 }

                 function salvar()
                 {
                       document.myForm.action                             =
"salveRequisitoNaoFuncionalAssociado.do";
                       document.myForm.submit();
                 }
           </script>
     </head>

     <body>
           <form method="post" name="myForm">
           <html:hidden  styleClass="edit"  name="requisitoNaoFuncional"
property="id"/>
           <html:hidden  styleClass="edit"  name="requisitoNaoFuncional"
property="idRequisitoFuncional"/>
           <table class="titulo">
                 <!--
########################################################-->
                 <!--# TITULO: HOPI e informações sobre a página aberta
#-->
                 <!--
########################################################-->
                 <tr>
```

```
                        <td>
                              <table class="transparente">
                                    <tr height="50">
                                    <td width="10%">
                                                <font class="titulo">
                                                      HOPI
                                                </font>
                                    </td>

                                    <td width="34%">
                                                <table
class="transparente">
                                                <tr
class="linhaItemSecao">
                                                      <td      width="53%"
align="right">
                                                            <font
class="miniVerdeEscuro">
                                                                  Usuário
Logado:
                                                            </font>
                                                      </td>


                                                            <td
width="47%" align="left">
                                                            <font
class="menu">

      <bean:write            scope="session"            name="dadosSessao"
property="nomeCompleto"/>
                                                            </font>
                                                      </td>
                                                </tr>

                                                <tr
class="linhaItemSecao">
                                                      <td align="right">
                                                            <font
class="miniVerdeEscuro">
                                                                  Projeto
Selecionado:
                                                            </font>
                                                      </td>

                                                      <td align="left">
                                                            <font
class="menu">

      <bean:write scope="session" name="projeto" property="nome"/>
                                                            </font>
                                                      </td>
                                                </tr>

                                                <tr
class="linhaItemSecao">
                                                      <td align="right">
                                                            <font
class="miniVerdeEscuro">

      Requisito Selecionado:
                                                            </font>
```

```
                                                      </td>

                                                      <td align="left">
                                                            <font
class="menu">

      <bean:write        scope="session"        name="requisitoFuncional"
property="nome"/>
                                                            </font>
                                                      </td>
                                                </tr>
                                          </table>
                                    </td>

                                    <td width="56%"></td>
                                    </tr>
                              </table>
                        </td>
                  </tr>

                  <!--
############################################################-->
                  <!--# NAVEGAÇÃO: apresenta cada etapa da análise e
projeto #-->
                  <!--
############################################################-->
                  <tr>
                        <td>
                              <table class="navegacaoOculta">
                                    <tr class="contornoNavegacao">
                                          <td colspan="5"></td>
                                    </tr>

                                    <tr class="linhaNavegacao">
                                          <td width=2%></td>

                                          <td class="etapas">
                                          <font
class="navegacaoEtapasMini">
                                                       Planejamento
                                          </font>
                                          <font
class="navegacaoEtapaSelecionada">
                                                       1
                                                </font>
                                          <font class="navegacaoEtapas">
                                                      2
                                                </font>
                                          </td>

                                          <td width=3%></td>

                                          <td class="etapas">
                                          <font
class="navegacaoEtapasMini">
                                                       Análise
                                          </font>
                                          <font class="navegacaoEtapas">
                                                       1 2 3
                                                </font>
                                          </td>
```

```
                                          <td width=3%></td>

                                          <td class="etapas">
                                          <font
class="navegacaoEtapasMini">
                                                 Projeto
                                          </font>
                                          <font class="navegacaoEtapas">
                                                 1 2 3
                                          </font>
                                          </td>

                                          <td width="32%"></td>
                                    </tr>

                                    <tr class="contornoNavegacao">
                                          <td colspan="5"></td>
                                    </tr>
                              </table>
                        </td>
                  </tr>

                  <!--
##########################################################-->
                  <!--#   MENUS:   apresenta   menus   de   acesso   rápido
#-->
                  <!--
##########################################################-->
                  <tr>
                        <td>
                              <table class="menu">
                                    <tr>
                                          <td>
                                                <font class="menu">
                                                 <a
class='linkMenu' href='principal.do'>Principal</a> |
                                                 <a
class='linkMenu' href='abraProjeto.do'>Projeto</a> |
                                                 <a
class='linkMenu' href='listeCategorias.do'>Categorias</a> |
                                                 <a
class='linkMenu' href='listeUsuarios.do'>Usuários</a> |
                                                 <a
class='linkMenu'
href="javascript:popUpAjudaRequisitoNaoFuncional()">Ajuda</a> |
                                                 <a
class='linkMenu' href="javascript:popUpSobre()">Sobre</a> |
                                                 <a
class='linkMenu' href='logout.do'>Logout</a>
                                                </font>
                                          </td>
                                    </tr>
                              </table>
                        </td>
                  </tr>

                  <!--
##########################################################-->
                  <!--# CADASTRO: apresenta o cadastro para entrada dos
dados #-->
```

```
                  <!--
##########################################################-->
                  <tr height="10"><td></td></tr>
                  <tr>
                        <td>
                              <table class="transparente">
                                    <tr>
                                    <td>
                                          <font class="nomePagina">
                                                Requisito Não-Funcional
                                          </font>
                                    </td>
                                    </tr>
                              </table>
                        </td>
                  </tr>
                  <tr height="15"><td></td></tr>
                  <tr>
                        <td>
                        <table class="transparente">
                                    <tr width="100%">
                                          <td width="25%"></td>
                                          <td width="75%">
                                                <html:errors/>
                                          </td>
                                    </tr>
                                    <tr width="100%">
                                          <td width="25%"></td>
                                          <td width="75%">
                                          <font class="normal">
                                                Nome
                                          </font>
                                          </td>
                                    </tr>
                                    <tr width="100%">
                                          <td></td>
                                          <td>
                                                <html:text
styleClass="edit"      name="requisitoNaoFuncional"      property="nome"
size="58"/>
                                          </td>
                                    </tr>
                                    <tr                   height="15"><td
colspan="3"></td></tr>
                                    <tr>
                                          <td></td>
                                          <td colspan="2">
                                          <font class="normal">
                                                Descrição
                                          </font>
                                          </td>
                                    </tr>
                                    <tr>
                                          <td></td>
                                          <td colspan="2">
                                                <html:textarea
styleClass="edit"    name="requisitoNaoFuncional"    property="descricao"
cols="70" rows="5"/>
                                          </td>
                                    </tr>
```

```
                                    <tr                    height="15"><td
colspan="3"></td></tr>
                                    <tr>
                                          <td></td>
                                          <td>
                                          <font class="normal">
                                                Categoria
                                          </font>
                                          </td>
                                    </tr>
                                    <tr>
                                          <td></td>
                                          <td>
                                                <html:select
name="requisitoNaoFuncional" property="idCategoria">
                                                      <html:options
collection="categorias" property="id" labelProperty="nome"/>
                                                </html:select>

                                          </td>
                                    </tr>
                                    <tr                    height="15"><td
colspan="3"></td></tr>
                                    <tr>
                                          <td></td>
                                          <td>
                                          <html:checkbox
name="requisitoNaoFuncional" property="obrigatorio"/>
                                          <font class="normal">
                                                      Marque    se    for
obrigatório
                                          </font>
                                          </td>
                                    </tr>
                                    <tr>
                                          <td></td>
                                          <td>
                                          <html:checkbox
name="requisitoNaoFuncional" property="permanente"/>
                                          <font class="normal">
                                                Marque se for permanente
                                          </font>
                                          </td>
                                    </tr>
                              </table>
                        </td>
                  </tr>
                  <tr height="15"><td></td></tr>
                  <tr>
                        <td>
                        <table class="transparente">
                                    <tr>
                                    <td align="right" width="72%">
                                                <input      class="normal"
type="button"                                         value="CANCELAR"
onClick="javascript:cancelar()">  
                                                <input      class="normal"
type="button" value="SALVAR" onClick="javascript:salvar()">
                                          </td>
                                          <td width="28%"></td>
                                    </tr>
```

```
                              </table>
                        </td>
                  </tr>
                  <tr height="7"><td></td></tr>
            </table>
            </form>
      </body>
</html>

</logic:present>

<logic:notPresent name="dadosSessao" scope="session">
      <meta                http-equiv="refresh"                content="0;
URL=acessoNaoAutorizado.do">
</logic:notPresent>
```

usuario.jsp

```
<%@ page import="org.hopi.DTO.DTOSessao"%>
<%@ page import="org.hopi.ejb.entity.VOUsuario"%>
<%@ taglib uri="WEB-INF/lib/struts/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="WEB-INF/lib/struts/struts-html.tld" prefix="html" %>
<%@ taglib uri="WEB-INF/lib/struts/struts-bean.tld" prefix="bean" %>

<logic:present name="dadosSessao" scope="session">
<html>
      <head>
      <title>
            HOPI - Usuário
      </title>
            <link         href="xstylesheet.css"         rel="stylesheet"
type="text/css">
            <script language="JavaScript">
                  function popUpAjudaUsuario()
                  {
                        ajudaUsuario   =   window.open("ajudaUsuario.jsp",
"AjudaUsuario",
'toolbar=0,scrollbars=0,location=0,statusbar=0,menubar=0,resizable=1,wid
th=500,height=200');
                  }

                  function popUpSobre()
                  {
                        sobre  =  window.open("sobre.jsp",  "SobreoHOPI",
'toolbar=0,scrollbars=0,location=0,statusbar=0,menubar=0,resizable=1,wid
th=500,height=200');
                  }

                  function cancelar()
                  {
                        document.myForm.action = "listeUsuarios.do";
                        document.myForm.submit();
                  }

                  function salvar()
                  {
                        document.myForm.action = "salveUsuario.do";
                        document.myForm.submit();
                  }
            </script>
      </head>

      <body>
            <form method="post" name="myForm">
            <html:hidden name="usuario" property="id"/>
                  <table class="titulo">
                        <!--
##########################################################-->
                        <!--# TITULO: HOPI e informações sobre a página
aberta   #-->
                        <!--
##########################################################-->
                        <tr>
                              <td>
                                    <table class="transparente">
                                          <tr height="50">
                                          <td width="10%">
```

```
                                                <font
class="titulo">
                                                      HOPI
                                                </font>
                                          </td>

                                              <td width="90%">
                                                <table
class="transparente">
                                                <tr
class="linhaItemSecao">
                                                      <td
class="colunaTipoItemSecao">
                                                            <font
class="miniVerdeEscuro">

      Usuário Logado:
                                                            </font>
                                                      </td>

                                                      <td
class="colunaValorItemSecao">
                                                            <font
class="menu">

      <bean:write            scope="session"           name="dadosSessao"
property="nomeCompleto"/>
                                                            </font>
                                                      </td>
                                                </tr>
                                                </table>
                                          </td>
                                          </tr>
                                    </table>
                              </td>
                        </tr>

                        <!--
############################################################-->
                        <!--# NAVEGAÇÃO: apresenta cada etapa da análise
e projeto #-->
                        <!--
############################################################-->
                        <tr>
                              <td>
                                    <table class="navegacaoOculta">
                                          <tr class="contornoNavegacao">
                                                <td colspan="5"></td>
                                          </tr>

                                          <tr class="contornoNavegacao">
                                                <td colspan="5"></td>
                                          </tr>
                                    </table>
                              </td>
                        </tr>

                        <!--
############################################################-->
                        <!--#  MENUS:  apresenta  menus  de  acesso  rápido
#-->
```

```
                        <!--
##########################################################-->
                        <tr>
                              <td>
                                    <table class="menu">
                                          <tr>
                                                <td>
                                                      <font class="menu">
                                                             <a
class='linkMenu' href='principal.do'>Principal</a> |
                                                             <a
class='linkMenu' href='listeCategorias.do'>Categorias</a> |
                                                             <a
class='linkMenu' href="javascript:popUpAjudaUsuario()">Ajuda</a> |
                                                             <a
class='linkMenu' href="javascript:popUpSobre()">Sobre</a> |
                                                             <a
class='linkMenu' href='logout.do'>Logout</a>
                                                      </font>
                                                </td>
                                          </tr>
                                    </table>
                              </td>
                        </tr>

                        <!--
############################################################-->
                        <!--# CADASTRO: apresenta o cadastro para entrada
dos dados #-->
                        <!--
############################################################-->
                        <tr height="10"><td></td></tr>
                        <tr>
                              <td>
                                    <table class="transparente">
                                          <tr>
                                          <td>
                                                <font class="nomePagina">
                                                      Usuário
                                                </font>
                                          </td>
                                          </tr>
                                    </table>
                              </td>
                        </tr>
                        <tr height="15"><td></td></tr>
                        <tr>
                              <td>
                              <table class="transparente">
                                          <tr width="100%">
                                                <td width="13%"></td>
                                                <td           width="76%"
colspan="3">
                                                      <html:errors/>
                                                </td>
                                                <td width="13%"></td>
                                          </tr>
                                          <tr width="100%">
                                                <td width="13%"></td>
                                                <td width="30%">
                                                <font class="normal">
```

```
                                                Nome
                                          </font>
                                          </td>
                                          <td width="4%"></td>
                                          <td width="30%">
                                          <font class="normal">
                                                Matrícula
                                          </font>
                                          </td>
                                          <td width="13%"></td>
                                    </tr>
                                    <tr>
                                          <td></td>
                                          <td>
                                                <html:text
styleClass="edit" name="usuario" property="nome" size="40"/>
                                          </td>
                                          <td></td>
                                          <td>
                                                <html:text
styleClass="edit" name="usuario" property="matricula" size="40"/>
                                          </td>
                                          <td width="13%"></td>
                                    </tr>
                                    <tr height="15"><td></td></tr>


                                    <tr width="100%">
                                          <td width="13%"></td>
                                          <td width="30%">
                                          <font class="normal">
                                                Username
                                          </font>
                                          </td>
                                          <td width="4%"></td>
                                          <td width="30%"></td>
                                          <td width="13%"></td>
                                    </tr>
                                    <tr>
                                          <td></td>
                                          <td>
                                                <html:text
styleClass="edit" name="usuario" property="username" size="40"/>
                                          </td>
                                          <td></td>
                                          <td></td>
                                          <td width="13%"></td>
                                    </tr>
                                    <tr height="15"><td></td></tr>

                                    <tr>
                                          <td></td>
                                          <td>
                                          <font class="normal">
                                                Password
                                          </font>
                                          </td>
                                          <td></td>
                                          <td>
                                          <font class="normal">
                                                Confirme o Password
```

```
                                          </font>
                                          </td>
                                          <td width="13%"></td>
                                    </tr>
                                    <tr>
                                          <td></td>
                                          <td>
                                                <html:password
styleClass="edit" name="usuario" property="password" size="40"/>
                                          </td>
                                          <td></td>
                                          <td>
                                                <input
type="password" class="edit" name="confirmaPassword" size="40">
                                          </td>
                                          <td></td>
                                    </tr>
                                    <tr height="15"><td></td></tr>
                                    <tr>
                                          <td></td>
                                          <td>
                                          <font class="normal">
                                                Telefone
                                          </font>
                                          </td>
                                          <td></td>
                                          <td>
                                          <font class="normal">
                                                E-mail
                                          </font>
                                          </td>
                                          <td></td>
                                    </tr>
                                    <tr width="100%">
                                          <td></td>
                                          <td>
                                                <html:text
styleClass="edit" name="usuario" property="telefone" size="40"/>
                                          </td>
                                          <td></td>
                                          <td>
                                                <html:text
styleClass="edit" name="usuario" property="email" size="40"/>
                                          </td>
                                          <td width="13%"></td>
                                    </tr>
                                    <tr height="15"><td></td></tr>
                                    <tr>
                                          <td></td>
                                          <td colspan="4">
                                                <html:checkbox
name="usuario" property="administrador"/>
                                                <font
class="normal">
                                                Marque se o usuário
for administrador
                                          </font>
                                          </td>
                                    </tr>
                              </table>
                        </td>
```

```
                        </tr>
                        <tr height="15"><td></td></tr>
                        <tr>
                              <td>
                              <table class="transparente">
                                          <tr>
                                          <td align="right" width="85%">
                                                      <input
class="normal"               type="button"              value="CANCELAR"
onClick="javascript:cancelar()">  
                                                      <input
class="normal"                type="button"               value="SALVAR"
onClick="javascript:salvar()">
                                                </td>
                                                <td width="15%"></td>
                                          </tr>
                                    </table>
                              </td>
                        </tr>
                        <tr height="7"><td></td></tr>
                  </table>
            </form>
      </body>
</html>
</logic:present>

<logic:notPresent name="dadosSessao" scope="session">
      <meta               http-equiv="refresh"               content="0;
URL=acessoNaoAutorizado.do">
</logic:notPresent>
```

FOLDER . (raiz)

Packaging-build.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project default="_generation_" name="Packaging Generator">
<target name="_generation_" depends="N400004,N400015"/>
<target name="N400004" description="hopiEJB.jar">
<mkdir dir="runtime"/>
<jar destfile="runtime/hopiEJB.jar">
<zipfileset dir="bin-ejb"/>
</jar>
</target>
<target name="N400015" description="hopiWEB.war">
<mkdir dir="runtime"/>
<jar destfile="runtime/hopiWEB.war">
<zipfileset dir="webapp"/>
<zipfileset dir="bin" prefix="WEB-INF/classes" includes="**/*.class"/>
</jar>
</target>
</project>
```

xdoclet-build.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project default="_generation_" name="XDoclet Generator">
<property file="xdoclet-build.properties"/>
<path      id="xdoclet.classpath"><pathelement      location="C:/jboss-
3.2.5/server/default/deploy/jbossweb-tomcat50.sar/jsp-api.jar"/>
<pathelement    location="C:/jboss-3.2.5/server/default/deploy/jbossweb-
tomcat50.sar/servlet-api.jar"/>
<pathelement          location="C:/jboss-3.2.5/server/default/lib/jboss-
j2ee.jar"/>
<pathelement location="C:/jboss-3.2.5/server/default/lib/log4j.jar"/>
<pathelement location="bin"/>
<pathelement location="bin-ejb"/>
<pathelement location="webapp/WEB-INF/lib/antlr.jar"/>
<pathelement location="webapp/WEB-INF/lib/commons-beanutils.jar"/>
<pathelement location="webapp/WEB-INF/lib/commons-collections.jar"/>
<pathelement location="webapp/WEB-INF/lib/commons-digester.jar"/>
<pathelement location="webapp/WEB-INF/lib/commons-fileupload.jar"/>
<pathelement location="webapp/WEB-INF/lib/commons-logging.jar"/>
<pathelement location="webapp/WEB-INF/lib/commons-validator.jar"/>
<pathelement location="webapp/WEB-INF/lib/jakarta-oro.jar"/>
<pathelement location="webapp/WEB-INF/lib/struts.jar"/>

<fileset
dir="/c:/eclipse/plugins/org.jboss.ide.eclipse.xdoclet.core_1.3.30/">
<include name="*.jar"/>
</fileset>
</path>
<target name="_generation_" depends="N400004"/>
<target name="N400004" description="EJB">
<taskdef                                  classpathref="xdoclet.classpath"
classname="xdoclet.modules.ejb.EjbDocletTask"
name="ejbdoclet"/><ejbdoclet destDir="gen-src" >
<homeinterface>
</homeinterface>
<localinterface>
</localinterface>
<localhomeinterface>
</localhomeinterface>
<remoteinterface>
</remoteinterface>
<deploymentdescriptor   xmlencoding="ISO-8859-1"   destDir="gen-src/META-
INF" >
</deploymentdescriptor>
<fileset dir="src-ejb"  includes="**/*Bean.java"  excludes="" >
</fileset>
<valueobject pattern="VO{0}" >
</valueobject>
<jboss  Version="3.2"  xmlencoding="ISO-8859-1"    destDir="gen-src/META-
INF" >
</jboss>
</ejbdoclet></target>
</project>
```

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

Artigo sobre o HOPI

Marcel Horner
Rafael de Andreis Pires

Com a finalidade de atender o fraco suporte oferecido pelas ferramentas de modelagem de sistemas existentes no que diz respeito a requisitos, bem como o relacionamento dos mesmos com casos de uso, é que está sendo proposta uma nova ferramenta CASE, o HOPI.

Antes de iniciar as atividades relativas ao desenvolvimento foi feito um levantamento sobre as opções de mercado existentes, para comprovar a afirmação feita a respeito da ineficiência destas com relação as etapas iniciais de definição e análise. Foram testadas aproximadamente 20 (vinte) ferramentas CASE, sendo que destas apenas uma apresentava a idéia proposta inicialmente, mas de forma muito precária, o que tornava claro que o propósito da nova ferramenta desenvolvida era realmente pertinente.

Uma questão relevante sobre o produto final levantada ainda na fase de estudo do mercado foi a opção de criar a ferramenta como um sistema web, que oferecesse suporte ao trabalho coletivo em projetos, não se limitando apenas a uma aplicação desktop monousuário.

O desenvolvimento do trabalho apresenta a definição, análise, projeto e implementação do sistema. Esta definição especifica quais funcionalidades o sistema deve atender, através do levantamento de requisitos funcionais e não

funcionais, como também o processo que deve ser realizado para este fim. A análise identifica quais entidades compõem a solução em computador e como é o relacionamento entre elas; nesta etapa o principal artefato gerado é o modelo conceitual. O projeto gera os diagramas de classe, que são os principais guias utilizados na atividade de implementação. A implementação consiste na construção, de fato, da ferramenta a partir de todo material gerado nas fases anteriores.

O HOPI foi estruturado de acordo com a proposta MVC, utilizando-se de J2EE que implementa vários padrões (design patterns) para a resolução de problemas típicos no desenvolvimento de sistemas. A utilização de EJB's foi feita nas camadas de negócio e de persistência seguindo o modelo CMP. A camada web ficou por conta do framework Struts que mapeia todo o fluxo de navegação do sistema. A IDE de desenvolvimento utilizada foi o Eclipse. A aplicação roda em um servidor de aplicações, no caso o JBoss, que provê também o banco de dados, Hypersonic.

O mecanismo de funcionamento geral do sistema consiste em listar elementos e, através de atalhos encontrados ao lado de cada item, realizar operações como editar ou remover registros. Podemos observar que as informações envolvidas nas etapas de análise de requisitos e definição de casos de uso são devidamente armazenadas, assim como os relacionamentos entre estes elementos. Estas características compreendem as funcionalidades esperadas para a ferramenta proposta, o que nos permite concluir que a mesma atendeu aos objetivos esperados.

Florianópolis, 08 de dezembro de 2004