

Versão 1.0

Revision History

Data	Versão	Descrição	Autor
27/03/2003	1.0	Visão inicial para discussão	Carlos Eugênio

Conteúdo

1.	Introdução	5
1.1	Finalidade	5
1.2	Fatores influentes	5
1.3	Escopo	5
1.4	Definições, siglas e abreviações	5
1.5	Referências externas	6
1.6	Resumo da Visão	6
2.	Posicionamento do Projeto	6
2.1	Oportunidade de Negócio	6
2.2	Definição do Problema e solução proposta	6
2.3	Definição da Posição do Produto	7
3.	Descrição dos Interessados e Usuários	7
3.1	Visão estratégica do mercado e usuários	7
3.2	Sumário dos Representantes de Grupos de Interessados	8
3.3	Sumário dos usuários	8
3.4	Ambiente de Trabalho	9
3.5	Perfil dos Representantes de Grupos de Interessados	9
3.5.1	FUNC[...]	9
3.6	Perfil dos Usuários	9
3.6.1	FUNC[Analista XXX]	10
3.7	Necessidades chaves dos Representantes e Usuários	10
3.8	Alternativas e Competição	11
3.8.1	FUNC[....]	11
4.	Visão Geral do Produto	11
4.1	Perspectiva do Produto	11
4.2	Sumário de Funcionalidades	12
4.3	Suposições e Dependências	12
4.4	Custo e Preço	13
4.5	Licenças e Instalação do Software	13
5.	Funcionalidades do Sistema	13
5.1	FUNC[1.0]: Criação de projetos para o desenvolvimento de um ou mais aplicativos.	14
5.2	FUNC[2.0]: Definição e modelagem de diagramas UML visualmente.	14
5.3	FUNC[3.0]: Utilização de assistentes para configuração de tarefas razoavelmente complexas ou longas.	14
5.3.1	FUNC[3.1]: Assistente para a criação de projetos.	14
5.4	FUNC[4.0]: Geração automática de código a partir de diagrama de classes.	14
5.5	FUNC[5.0]: Geração automática de páginas JSP a partir da definição de classes Java Beans e de arquivos utilizando a notação do Struts.	14
5.6	FUNC[6.0]: Geração automática de classes para testes utilizando o Coiote da Apache.	14
5.7	FUNC[7.0]: Templates pré-configurados para a inserção de código pré-definido de padrões de projeto na aplicação.	14
5.8	FUNC[8.0]: Deployment automático nos containers.	14
5.9	FUNC[9.0]:	14
6.	Restrições	14

7.	Requisitos de Qualidade	14
8.	Precedência e Prioridade	14
9.	Outros requisitos do Produto	14
9.1	Padrões	14
9.2	Requisitos para instalação e funcionamento	14
9.3	Requisitos de Performance	15
9.4	Requisitos do Ambiente	15
10.	Requisitos de Documentação do Software	15
10.1	Manual do Usuário	15
10.2	Ajuda On-line	15
10.3	Guia de Instalação, Configuração e Arquivo README.	15
10.4	Rótulos, etiquetas, ícones gráficos e logos	16
11.	Appendix 1 - Feature Attributes	16
11.1	Status	20

<Sistema>

1. Introdução

A finalidade deste documento é coletar, analisar e definir necessidades de alto nível do sistema É focado nas necessidades definidas pelos solicitantes/interessados do sistema, e dos usuários finais, e o porquê destas necessidades existirem. Os detalhes de como o ... irá preencher estes requisitos e necessidades estão expostos nas especificações complementares e use-cases. A seguir são definidas de maneira resumida a finalidade do sistema, fatores que influenciaram ou deram contribuição positiva ao início do processo de desenvolvimento, o escopo do sistema e sistemas relatos, fatores que impulsionaram os solicitantes e interessados do sistema e os documentos, abreviaturas e siglas utilizadas neste documento.

[Algumas regras gerais para a definição dos itens desta seção: indicar a finalidade geral do documento, a plataforma alvo do sistema (ex.: Aplicação Web, Aplicação Stand-alone, etc.), finalidade macro do sistema (reduzidamente), resumida exposição de fatores que levaram ao desenvolvimento do sistema, indicação do sistema macro ao qual este faz parte (se houver), fatores que impulsionaram os gerentes ou usuários a solicitarem o sistema, e outros pontos chaves do sistema de maneira resumida. Se alguma abreviatura do sistema é utilizada durante o documento esta pode ser indicada aqui para aumentar a clareza do documento.]

1.1 Finalidade

O ... é um framework que permite ao usuário criar aplicações JSP utilizando um framework de classes pré-construídas e desing patterns pré-estabelecidos e implementados. A intenção é que o usuário parta dos artefatos gerados pela UML, e que só seja necessária a codificação dos métodos pertinentes dos objetos de regras de negócio e a construção das páginas JSP/HTML baseadas em frameworks também pré-estabelecidos, para que se possa fazer o deployment do sistema em um container JSP.

[Regras para a definição da Finalidade: deve-se indicar a finalidade do sistema e quais as principais necessidades que ele supre].

1.2 Fatores influentes

Este sistema está sendo desenvolvido para atender uma solicitação da ... que necessita criar um ambiente que em seja possível facilitar e padronizar o desenvolvimento dos sistemas internos da instituição e possivelmente sistemas tercerizados. O ambiente atual da universidade torna proibitiva a criação de sistemas utilizando padrões de projeto, e a diversidade e falta de controle dos aplicativos criados impedem a manutenção facilitada dos sistemas.

[Regras para a definição dos Fatores que influenciam o desenvolvimento: apresentar de forma sucinta os principais fatores que levam ao desenvolvimento do sistema, além de fatores relevantes do cenário geral ao qual o sistema está inserido].

1.3 Escopo

O framework deve ser capaz de gerar aplicações para Web utilizando JSP e Enterprise Java Beans.

[Regras para a definição do Escopo: definir o escopo do sistema em relação ao ambiente de aplicação (Web, stand-alone, distribuída, ect). Pode-se também definir sua relação com outros sistemas].

1.4 Definições, siglas e abreviações

Todas as definições de termos, siglas e abreviações utilizadas neste documento estão detalhadas e explicadas no Glossário do projeto.

[Regras para esta subseção: pode-se definir alguns termos mais utilizados no documento, entretanto o mais comum é reportar-se ao Glossário como o local central de definição de termos, siglas e abreviações utilizadas durante todo o projeto].

1.5 Referências externas

Os documentos que este projeto referencia são os que seguem:

- Glossário do Projeto ... – Engeny Informática, 2003.
- Use Cases do Projeto ... – Engeny Informática, 2003.

[Regras para Referências externas: aqui se deve enumerar todos os documentos aos quais esta Visão faz referência. A notação que deve ser utilizada para isto é a mesma utilizada para referências bibliográficas em trabalhos acadêmicos].

1.6 Resumo da Visão

No resto deste documento estão definidos todos os tópicos relacionados com o Ele contém a identificação da posição da empresa e do software no mercado, seu público alvo, os líderes e definidores de requisitos do projeto, os usuários finais, o mercado que o aplicativo deseja atingir, a definição do projeto propriamente dita, competidores, custos e análise inicial de funcionalidades do sistema.

[Regras para o Resumo da Visão: esta parte deve ser preenchida com um breve resumo do resto do documento e pode incluir destaque a alguns fatores. Deve definir os tópicos que serão abordados no resto da visão como a identificação da posição da empresa e do software no mercado, seu público alvo, os líderes e definidores de requisitos do projeto, os usuários finais, o mercado que o aplicativo deseja atingir, a definição do projeto propriamente dita, competidores, custos e análise inicial de funcionalidades do sistema].

2. Posicionamento do Projeto

2.1 Oportunidade de Negócio

Apesar do projeto ser inicialmente focado para a ..., o sistema deve ser feito tendo em mente sua ampla utilização no mercado de desenvolvimento de sistemas. Outras empresas que podem ter interesse imediato no sistema são a

[Regras para a definição da subseção Oportunidade de Negócio: aqui são descritas as oportunidades de negócio que devem ou podem ser geradas pelo projeto].

2.2 Definição do Problema e solução proposta

O desenvolvimento de software de qualidade é uma problemática constante para as empresas envolvidas neste processo. O processo de desenvolvimento compreende várias fases. Apesar de ser uma questão pacífica, a elaboração do projeto utilizando padrões como UML, não é seguida rigidamente por todas empresas. Não obstante, desenvolvedores tendem a criar projetos cada um utilizando as ferramentas que mais conhece, propiciando o surgimento de um ambiente heterogêneo de tecnologias e de difícil manutenção. A fase de codificação é uma das fases que costuma ser uma das menos padronizadas dentro das empresas. O código gerado por um desenvolvedor não é facilmente reconhecido por outro, e pior, a falta de conhecimento teórico do mesmo, o impede de utilizar padrões de projeto bem conhecidos e que poderiam facilitar a comunicação de todos os desenvolvedores e analistas da empresa.

A passagem da fase de planejamento e elaboração, construção de use cases e diagrama de classes para a fase de codificação geralmente é feita de maneira empírica e não padronizada, o que, frequentemente, leva a sistemas que às vezes utilizam componentes mas não podem gerar componentes utilizáveis em outros sistemas.

Desta maneira, uma ferramenta que permita solucionar os problemas descritos deve prover basicamente os seguintes serviços:

- Automatização da criação do código a partir dos modelos de classes gerados pela UML.
- Geração automática de classes de suporte à persistência e a padrões de projetos conhecidos.
- Direcionamento para o analista na construção de páginas JSP.
- Automatização de testes.
- Automatização do deployment.

[Regras para a subseção Definição do problema e solução proposta: aqui é necessária a indicação do problema em poucas frases, os usuários atingidos pelo problema, o impacto deste problema para os usuários e o que uma aplicação que pretende resolver este problema deve implementar]

2.3 Definição da Posição do Produto

O ... é um produto que visa suprir a deficiência de similares em prover um meio claro e fácil para um analista partir dos modelos de classe gerados na fase de análise e preencher as regras de negócio dos objetos sem se preocupar com todas as problemáticas inerentes à segurança, transações, persistência e padrões de projeto.

O público alvo é composto, em sua maioria, por empresas que pretendem desenvolver sistemas voltados para a Internet utilizando a plataforma Java, JSP e Enterprise Java Beans como plataforma padrão. Inicialmente o principal interessado é a

[Regras para a subseção Definição da Posição do Produto: é imprescindível que se defina a localização do produto no mercado, com uma diferenciação principal e que justifique o desenvolvimento do mesmo. Deve conter o público alvo, a necessidade deste público, a categoria em que este produto se encaixa, os principais benefícios que levariam este público alvo a comprar o produto, seus principais concorrentes e seu diferencial. Esta definição serve para orientar todos os envolvidos no projeto de maneira decisiva, fazendo com que todos saibam o que estão desenvolvendo, o porquê, para quem e contra quem, ou o que, estão lutando].

3. Descrição dos Interessados e Usuários

[Regras para a Descrição dos Interessados e Usuários: é necessária a identificação de todos os interessados e usuários no processo de Modelagem. Esta seção provê um retrato dos interessados e usuários do sistema e os problemas que eles desejam ver solucionados com o projeto. Provê um suporte e justificativa para os requerimentos do projeto].

3.1 Visão estratégica do mercado e usuários

[Regras para a Visão estratégica do Mercado e usuários: sumário dos motivos que levaram ao desenvolvimento do software. Descrever resumidamente a posição do mercado alvo no contexto local e do país, bem como a possibilidade de utilização em outros mercados. Estimar o tamanho do mercado bem como sua capacidade de crescimento baseado no número previsto de usuários, a quantidade de dinheiro que os usuários estariam dispostos a pagar pelo produto. Sumarizar as tendências do mercado e dos concorrentes. Tentar responder as questões: Qual a reputação da empresa neste mercado e perante os usuários. Como deveria ser esta reputação? Como este produto pode ajudar a alcançar estes objetivos?].

A ... apontou como problemas principais nos softwares que desenvolve os seguintes:

- A padronização da modelagem codificação e implementação do software.

- A utilização de padrões de projeto conhecidos.
- Tempo gasto na recodificação de objetos já prontos.
- O alto preço cobrado pelas ferramentas já existentes no mercado similares como ferramentas de modelagem, desenvolvimento e codificação do software, além de implementação do mesmo.

Um dos requisitos básicos do Framework ...

Em uma estimativa inicial de duas empresas estariam interessadas imediatamente na aquisição do Framework e potencialmente mais duas outras empresas estariam interessadas a médio e longo prazo no mercado baiano.

Além do custo do software uma outra fonte de retorno do investimento é o suporte técnico necessário para as ferramentas ou Open Source utilizadas pelo Framework. A estimativa do que uma empresa teria que pagar mensalmente para ter o suporte

3.2 Sumário dos Representantes de Grupos de Interessados

[Aqui devem ser enumerados os agentes Representantes de Grupos Interessados no desenvolvimento do sistema, que grupo de usuários eles representam e quais suas atividades no desenvolvimento do sistema].

Nome	Representa	Atribuições
...	Direção da Este é o solicitante do sistema, tendo uma visão sólida do que o sistema deve fazer, sendo suas indicações de alta importância.	Certificar-se que o sistema apresentado está de acordo com o proposto.
...	Coordenação de Desenvolvimento e Projetos???	????

3.3 Sumário dos usuários

[Apresentar uma lista de todos os usuários que podem ser identificados para o sistema].

Nome	Descrição	Representante
Analista de Desenvolvimento	Este é o usuário final do sistema. O desenvolvimento do Framework é todo dirigido para facilitar o trabalho deste profissional, sendo de extrema importância captar suas opiniões no decorrer do desenvolvimento.	Analista???
Analista de Modelagem	Este usuário estará procedendo à modelagem do sistema na ferramenta projetada para modelagem UML.	Analista???
Gerente de Projeto	Responsável por organizar, dividir tarefas e acompanhar o desenvolvimento dos	Luíza???

	sistemas na empresa.	
--	----------------------	--

3.4 Ambiente de Trabalho

[Detalhar o ambiente de trabalho do usuário final do sistema. Deve-se descrever o contexto em que o usuário irá utilizar o sistema, e os principais fatores que podem atrapalhar ou facilitar a utilização do sistema neste ambiente].

[Detalhar o ambiente de trabalho de cada usuário:

- *Número de pessoas que estão envolvidas para realizar a tarefa? Este número está mudando ou existe a possibilidade de alteração?*
- *Qual o tempo que o usuário leva para completar a tarefa? É satisfatório? Pode ser alterado?*
- *Qualquer restrição do ambiente: se o usuário trabalha com um notebook em constante locomoção, acessa o sistema em lugares diferentes do da empresa, utiliza via Internet com conexão via Celular, etc.?*
- *Qual a plataforma que o usuário utiliza hoje? Qual sistema operacional e tipo de equipamento? Quais as plataformas que ele deve usar em um futuro próximo?*
- *Quais são as outras aplicações que o usuário utiliza? É necessário que o sistema tenha que se integrar com elas?*

Aqui podem ser incluídos os usuários identificados no Modelo de Negócio do cliente].

3.5 Perfil dos Representantes de Grupos de Interessados

[Descrever cada Representante de Grupo de Interessados preenchendo as informações na tabela abaixo para cada Representante. Lembrar que os tipos de representantes podem ser os mais variados como usuários do sistema, um departamento de vendas, um grupo de desenvolvedores:]

3.5.1FUNC[...]

Representação	Direção da ...
Descrição	Este grupo é o mais importante no cenário atual pois é o que define como o sistema deverá parecer para ser considerado um sucesso.
Tipo de Experiência	É o que possui mais experiência e pode ser considerado um Guru ou Expert
Responsabilidades	Deve acompanhar todo o processo do desenvolvimento, verificando a adequação ao que foi proposto.
Critério de sucesso.	Para este interessado o sistema teve sucesso em seu desenvolvimento se tiver as características que ele próprio estipulou.
Produtos	<ul style="list-style-type: none"> ▪ Integração com os diagramas provindos da modelagem utilizando UML fazendo com que o desenvolvedor só precise preencher os métodos de regra de negócio para a aplicação funcionar ▪ Integração com o Struts e implementação do modelo MVC. ▪ Deployment automático da aplicação desenvolvida. ▪ Utilização de Design Patterns no código gerado pelo Framework.
Comentários e problemas	Este é o solicitante do sistema e pagador e, como tal, todas as suas considerações devem estar implementadas.

3.6 Perfil dos Usuários

[Descrever cada usuário do sistema preenchendo a tabela abaixo. Lembrar-se que os tipos de usuários também são diversos como experts e novatos. Um expert pode requer uma interface com

opções sofisticadas, que o sistema trabalhe em várias plataformas, enquanto um novato necessita de um sistema simples, sem muitas opções nas telas e fácil de usar na forma de assistentes].

3.6.1FUNC[Analista XXX

Representante	Quem é o representante deste usuário?
Descrição	Descreva brevemente este tipo de usuário.
Tipo	Qualifique o usuário com novato, médio ou expert. Incluir suas habilidades frente a um computador e ao serviço que pretende fazer utilizando o sistema.
Responsabilidades	Listar as principais responsabilidades do usuário ao utilizar o sistema (Gerar relatórios, fazer entrada de dados, etc).
Critério de Sucesso	Como o usuário define o sucesso do sistema? Como o usuário seria recompensado?
Produtos	O que o usuário produz? Para quem?
Comentários e Problemas	Especificar os problemas que podem advir da interface deste usuário com o sistema. Quais seriam as melhorias a serem implementadas para que este usuário fosse recompensado pelo uso do sistema?

3.7 Necessidades chaves dos Representantes e Usuários

3.7.1FUNC[Padronização dos modelos

Os analistas da ... não adotam uma padronização na descrição dos modelos criados para as aplicações. Cada grupo de trabalho cria sua aplicação baseado em sua experiência, resultando em um ambiente de aplicativos difícil de manter. Existe uma tentativa para padronizar o desenvolvimento, entretanto, devido ao diferente grau de experiência dos desenvolvedores, é difícil adotar uma solução única para todos os grupos de trabalho.

A ... pretende adotar a modelagem baseada em Orientação a Objetos e a UML para descrever os modelos como padrão para todos os projetos. Além disto a padronização deve englobar a linguagem Java, JSP e Servlets, EJB e padrões de projeto.

3.7.2FUNC[Dificuldade na criação de objetos e classes baseados em modelos

Como resultado do diferente grau de conhecimento dos analistas existe a dificuldade de alguns em criar classes utilizando os padrões estabelecidos.

Assim, é necessário que exista uma ferramenta que automatize o processo de desenvolvimento, permitindo que a partir de modelos criados em uma ferramenta de design de classes, seja gerada a estrutura das classes padronizadas e que os analistas tenham o mínimo de trabalho, ou seja, de implementar os métodos dos objetos de negócio. Todo o código para o suporte a persistência, deployment no EJB container, deployment das páginas JSP no Web container deve ser gerado automaticamente pela ferramenta.

[Listar os principais problemas com as soluções existentes apresentadas pelo Representante. Elucide as seguintes questões para cada problema:

- *Quais são as razões para este problema?*
- *Como ele está sendo resolvido neste momento?*
- *Quais são as soluções que o representante deseja?*

Preencher a tabela a seguir para definir as prioridades de necessidades e soluções].

Necessidade	Prioridade	Solução atual	Soluções propostas
Padronização de modelos	Alta	Nenhum	Incorporação ao Framework de uma ferramenta de modelagem UML.
Criação de Classes baseadas nos modelos	Alta	Nenhum	Criar uma função automatizada de geração de código para as classes a partir dos modelos de classes da UML
Deployment facilitado	Alta	Nenhum	Automatizar o processo de deployment das aplicações JSP utilizando as classes do Framework assim como EJB.

3.8 Alternativas e Competição

[Identificar as alternativas que os representantes enxergam. Pode ser um produto de um competidor, desenvolver o produto ou simplesmente manter as coisas do jeito que estão. Listar qualquer escolha que exista de um produto concorrente ou que possa ser desenvolvido. Incluir as maiores vantagens e desvantagens de cada competidor como definido pelo representante].

3.8.1 FUNC[...]

Empresa:

Tipo de Software: Framework para o desenvolvimento de aplicações java.

Vantagens:

- ???

Desvantagens:

- ???

4. Visão Geral do Produto

[Esta seção provê uma visão de alto nível das funcionalidades do sistema, as suas iterações com outras aplicações e configurações do sistema.]

4.1 Perspectiva do Produto

[Regras para a Perspectiva do Produto: esta seção deve situar o produto em relação a outros produtos produzidos pela empresa. Se este produto não tem relação nenhuma com outros produtos deve ser esclarecido aqui. Se ele é parte de um contexto maior, deve ser exposto como ele se relaciona com os outros sistemas. Uma maneira fácil de mostrar esta iteração é através de blocos de sistemas].

O ..., deve visar em última instância ser um Framework para desenvolvimento de aplicações Java para a Web que suporte a geração automática de várias partes do sistema e de sua implementação. Futuramente ele deve integra-se com os sistemas de controle de processos gerenciais e o de modelagem UML que devem ser desenvolvidos pela empresa.

É importante iniciar o desenvolvimento de uma aplicação de acompanhamento gerencial do desenvolvimento utilizando o Framework. Isto tornará a aplicação mais atraente para gerentes de informática buscando ter um maior controle sobre todo o processo de desenvolvimento, assim como um acompanhamento ativo das atividades dos usuários.

4.2 Sumário de Funcionalidades

[Regras para o Sumário de Funcionalidades: esta seção lista os maiores benefícios do sistema sem entrar em detalhes como eles serão implementados. Esta lista deve poder ser compreendida por todos envolvidos no projeto, inclusive os usuários. Uma simples tabela mostrando os benefícios primários do sistema é o suficiente aqui].

[Sistema]	
Benefício para o usuário	Funcionalidades previstas
Analistas com pouca experiência de programação devem poder utilizar o Framework através de assistentes sem a necessidade de um aprofundamento em Java, JSP e EJB	Os assistentes conduzem o usuário passo a passo na criação da aplicação.
O aprendizado é rápido e de fácil retenção	As apresentações exemplos, o guia de perguntas mais freqüentes, e a Ajuda do usuário ajudam a entender como trabalhar com o Framework e as possíveis aplicações do sistema.
Usuários avançados podem melhorar o funcionamento da ferramenta configurando o Framework para agir de forma específica em certas situações.	Para os usuários avançados estará disponível toda a configuração do sistema, inclusive edição de arquivos ocultos de implementação como os descriptors EJB, web.xml e server.xml do Web container e o build.xml do Ant.
Melhor visualização do processo de desenvolvimento.	Aos gerentes será possível visualizar os modelos e classes geradas pelos analistas para acompanharem o processo.

4.3 Suposições e Dependências

[Regras para a seção Suposições e Dependências: esta seção serve para mostrar os fatores que podem levar a mudanças significativas neste documento de Visão do Sistema. Por exemplo, o suporte a um sistema operacional específico pode não estar incluso na Visão por não existir no momento em que ela foi confeccionada. Caso o suporte seja necessário a Visão deve mudar].

A infra-estrutura de software e hardware específicos para utilizar o EJP será especificada mais adiante.

4.4 Custo e Preço

[Regras para a seção de Custos e Preço: apresentar todos os fatores que influenciam direta ou indiretamente no custo do produto, indicando a quantidade e a estimativa de custo para cada usuário. Se necessário, indicar os custos variantes a depender dos usuários. Exemplos de definição destes custos são as mídias para distribuição, os manuais, equipamentos, custos com suporte e outros que sejam necessários].

O ... deverá ser distribuído em CDROM. O custo aproximado de embalagem é de ... por unidade. Outra maneira de distribuição prevista é pela Internet, mas este tópico será alvo de um estudo futuro pelo problema da segurança.

Os custos com propaganda e divulgação devem ser levantados mais adiante e inseridos neste ponto.

4.5 Licenças e Instalação do Software

[Regras para a seção Licenças e Instalação do Software: nesta parte devem ser indicadas as formas previstas de licença do software, indicando restrições de segurança e rede. A maneira de instalação é um fator que também deve ser levado em conta, pois pode ensejar a necessidade de ter-se um produto a parte para a instalação do sistema].

A licença do software poderá ser adquirida por unidade ou por um pacote econômico. A quantidade de licenças adquiridas em conjunto deve afetar o preço do produto em ordem aritmética inversa.

5. Funcionalidades do Sistema

[Listar e descrever resumidamente as funcionalidades do sistema. As funcionalidades são as funções em alto nível que o sistema deve ter para gerar benefícios para os usuários. Cada funcionalidade é um serviço que o sistema deve prover ao usuário e que geralmente requer uma série de entradas para que produza o efeito desejado. Uma funcionalidade do sistema pode ou não ser mapeada diretamente para uma solicitação do Grupo de Representantes de Usuários ou de um Usuário. Todas as requisições devem ter uma funcionalidade que a atenda.

O nível de detalhe na Visão Geral do Sistema deve ser mantido o suficiente para explicar a funcionalidade sem fazer com que todas as pessoas que tenham acesso à Visão deixem de entender tal funcionalidade]

- 5.1 FUNC[1.0]: Criação de projetos para o desenvolvimento de um ou mais aplicativos.
- 5.2 FUNC[2.0]: Definição e modelagem de diagramas UML visualmente.
- 5.3 FUNC[3.0]: Utilização de assistentes para configuração de tarefas razoavelmente complexas ou longas.
- 5.3.1 FUNC[3.1]: Assistente para a criação de projetos.
- 5.4 FUNC[4.0]: Geração automática de código a partir de diagrama de classes.
- 5.5 FUNC[5.0]: Geração automática de páginas JSP a partir da definição de classes Java Beans e de arquivos utilizando a natação do Struts.
- 5.6 FUNC[6.0]: Geração automática de classes para testes utilizando o Coiote da Apache.
- 5.7 FUNC[7.0]: Templates pré-configurados para a inserção de código pré-definido de padrões de projeto na aplicação.
- 5.8 FUNC[8.0]: Deployment automático nos containers.
- 5.9 FUNC[9.0]:

6. Restrições

[Regras para a definição de Restrições: se há restrições no desenvolvimento do projeto estas devem ser notadas aqui. Vale notar qualquer restrição de desenvolvimento ou restrições externas que foram impostas pelo ambiente].

O ... deve ser completamente desenvolvido utilizando softwares livres. O ambiente em que deve ser executado é notadamente o Linux, porém ele deve poder se portátil para o Windows.

7. Requisitos de Qualidade

[Regras para a definição dos requisitos de qualidade: nesta seção são mostrados os requisitos de qualidade que não puderam ser definidos na seção de Funcionalidades do Sistema como performance, robustez, tolerância a erros, usabilidade e outras características similares].

8. Precedência e Prioridade

9. Outros requisitos do Produto

[Listar os outros requisitos do produto que não foram capturados anteriormente neste documento].

9.1 Padrões

[Regras para a seção Padrões: listar todos os padrões que o software deve ser compilante. Podem incluir padrões legais e regulamentais, padrões de protocolos e meios de comunicação (TCP/IP, ISDN), plataformas de hardware e sistemas operacionais e padrões de qualidade (ISO)].

O ... deve conter ferramentas que possam ser executados tanto no sistema operacional o Linux quanto no ambiente Windows. O container Web deve ser o Tomcat, e o container para Enterprise Java Beans deve ser o JBoss.

9.2 Requisitos para instalação e funcionamento

[Regras para a definição dos requisitos para instalação e funcionamento: nesta seção apresentar todos os requisitos necessários para dar suporte ao funcionamento do sistema. Pode incluir sistemas operacionais alvos ou tipos de hardware suportados, configurações em registro ou

arquivos de inicialização, periféricos e softwares que acompanham o produto].

O ambiente de funcionamento do ... é o que segue:

- Sistema operacional o Linux para o Tomcat.
- Sistema operacional o Linux par JBoss.
- Sistema operacional o Linux ou o Windows para a ferramenta de desenvolvimento
- Sistema operacional o Linux ou Windows para ferramenta de modelagem em UML.

9.3 Requisitos de Performance

9.4 Requisitos do Ambiente

[Este tópico é opcional pois para a maioria dos softwares já está definido pelo ambiente operacional que está rodando. Para outros produtos ou processos pode ser de extrema importância abrange-lo].

10. Requisitos de Documentação do Software

[Nesta seção devem ser descritos todos os documentos que necessitam ser gerados durante o processo de desenvolvimento para que o produto seja bem sucedido].

10.1 Manual do Usuário

[Regras para a seção Manual do usuário: indicar as características desejadas para o manual do usuário bem como restrições de encadernação. Pode-se considerar todos os fatores como o tamanho, o tipo de audiência e etc.]

O ... deve ser acompanhado de um manual do usuário contendo as seções seguintes:

- Instalação e configuração do Framework.
- Instalação e configuração do Tomcat.
- Instalação e configuração do JBoss.
- Instalação e configuração do Argo UML.
- Instalação e configuração da ferramenta de desenvolvimento.
- Criação de aplicações utilizando o Framework.
- Outras seções como diz a manuais de usuário.

10.2 Ajuda On-line

[Regra para a definição da Ajuda On-line: nesta seção identificar a necessidade da Ajuda On-line do usuário, bem como a maneira que será abordada sua criação.]

Deve ser disponibilizada uma ajuda on line para o A ajuda on line deve cumprir os mesmos tópicos do manual do usuário.

10.3 Guia de Instalação, Configuração e Arquivo README.

[A definição de um guia de instalação e configuração do software é importante numa solução completa. Alguns usuários já estão acostumados a ver tais guias e a ler os arquivos README. Todas as seções deste arquivo podem ser contempladas como “Quais são as novidades desta versão do software”, “Compatibilidade com Versões Anteriores”, “Bugs conhecidos e maneiras

de contorna-los”].

Devem ser disponibilizados também um guia de instalação separadamente do manual do usuário. Este guia deve estar encadernado da mesma forma que o manual.

10.4 Rótulos, etiquetas, ícones gráficos e logos

[A DEFINIR]

11. Appendix 1 - Feature Attributes

[Features should be given attributes that can be used to evaluate, track, prioritize and manage the product items proposed for implementation. All requirement types and attributes should be outlined in the Requirements Management Plan, however you may wish to list and briefly describes the attributes for features that have been chosen. Following subsections represent a set of suggested feature attributes.]

Tipo de Documento	Descrição	Tipo de Requisito Default
Requisição de um Representante de um Grupo de Interessados (GIRQ)	Solicitações chaves de Representantes de Grupos de Interessados.	Requisição de um Representante de um Grupo de Interessados (GIRQ)
Visão Geral do Projeto (VGP)	Documento que apresenta uma visão geral do projeto, sua condição atual e capacidades. Deve estar sempre atualizado para refletir a situação atual.	Funcionalidade (FUNC)
Especificação de um Use-Case (EUC)	Descrição e elaboração de um Use Case.	Use Case (UC)
Glossário (GLS)	Usado para capturar o vocabulário comum do projeto.	Item de Glossário (TERMO)
Especificação Moderna de um Requisito do Software (EMRS)	Especificação de Requisitos específicos para o desenvolvimento do código do software.	Requisito de Software (RS)
Plano de Gerenciamento de Requisitos (PGR)	Este documento descreve os requisitos e estratégias específicas para o Gerenciamento e Desenvolvimento do Plano de Gerenciamento de Requisitos.	Plano de Gerenciamento de Requisitos (PGR) [Requisito default. Raramente é utilizado] .

Tipo de Requisito	Descrição	Atributos possíveis
Requisição de um Representante de um Grupo de Interessados (GIRQ)	Uma requisição de qualquer tipo feita por um Representante de um Grupo de Interessados (GI). Pode ser, por exemplo, uma requisição de alteração no sistema, de uma nova funcionalidade, de um melhoramento, mudança de requisito ou um defeito.	Prioridade, Status, Custo, Dificuldade, Estabilidade, Atribuído a.

Funcionalidade (FUNC)	As funcionalidades são serviços que o sistema oferece ao usuário. Todas as requisições de usuários ou grupo de interessados devem ser mapeadas para uma funcionalidade. Podem existir funcionalidades no sistema que não têm uma solicitação direta de um usuário.	Prioridade, Status, <Planned Iteration???, <Actual Iteration???, Dificuldade, Estabilidade, Atribuído a, Origem, <Rationale???, Custo, Solicitação de Melhoramento, Defeito.
Use Case (UC)	Uma descrição do comportamento do sistema em termos de sequência de ações. Um Use Case deve prover uma satisfação comprovada de uma funcionalidade a um ator.	Propriedade, Prioridade, <Affects Architecture???, <Planned Iteration???, <Actual Iteration???, Atribuído a, Ranque???, Testado, Status, Dificuldade, Estabilidade, Custo, Solicitação de Melhoramento, Defeito.
Item de Glossário (TERMO)	Um termo utilizado no vocabulário geral do projeto.	
Requisito de Software (RS)	Uma especificação de um comportamento do sistema que possa ser observado externamente, por exemplo, entradas do sistema, saídas do sistema, funções do sistema, atributos do sistema ou atributos do ambiente do sistema.	Prioridade, Status, Dificuldade, Estabilidade, Atribuído a, Custo, Solicitação de Melhoramento, Defeito.

Atributo	Descrição	Tipo	Lista de Valores Possíveis	Tipo de Requisito
Prioridade	A prioridade de um requisito é um atributo que define sua ordem de precedência de implementação perante os outros.	Lista de Valores	Indispensável	FUNC, UC,RS, PGR, GIRQ
			Recomendado	
			Possível	
			Descartado	
Status		Lista de Valores	Proposto	FUNC, UC,RS, PGR, GIRQ
			Aprovado	
			Incorporado	
			Validado	
Planned Iteration???		Inteiro	n/a	FUNC, UC
Actual Iteration???		Inteiro	n/a	FUNC, UC
Dificuldade		Lista de	Alta	FUNC,PGR,RS,GIRQ

		Valores	Média	
			Baixa	
Estabilidade		Lista de Valores	Alta	FUNC,PGR,RS,GIRQ
			Média	
			Baixa	
Atribuído a		Texto	n/a	FUNC,PGR,RS,GIRQ
Origem		Lista de Valores	Suporte	FUNC
			Parceiros	
			Competidores	
			Grandes Consumidores	
Argumento		Texto	n/a	FUNC
Custo		Moeda	n/a	FUNC,PGR,RS,GIRQ
Solicitação de Melhoramento		Texto	n/a	FUNC,RS
Defeito		Texto	n/a	FUNC,RS
Propriedade		Lista de Valores	Nome	UC
			Breve Descrição	
			Sequência Típica de Eventos	
			Sequência Alternativa de Eventos	
			Requisito Especial	
			Pré-Condição	
			Pós-Condição	
Affects Architecture???		Booleano	Verdadeiro/Falso	UC
Ranke		Inteiro	n/a	UC
Teste		Booleano	Verdadeiro/Falso	UC

Valor	Para o Atributo	Descrição
Indispensável	Prioridade	Significa que este requisito é indispensável ao software e não pode deixar de ser implementado.

Recomendado	Prioridade	O requisito foi recomendado para ser implementado no software. Primeiro deve ser feito a análise de viabilidade, ou seja, se é possível de ser implementado.
Possível	Prioridade	O requisito tem possibilidade de ser implementado no software. Entretanto não há uma forte indicação que isto acontecerá.
Descartado	Prioridade	Não há nenhuma possibilidade deste requisito ser implementado no software.
Proposto	Status	O requisito foi proposto e ainda não foi aprovado ou incorporado ao software.
Aprovado	Status	O requisito foi aprovado para ser incorporado ao software.
Incorporado	Status	Significa que o requisito já foi incorporado ao software.
Validado	Status	Significa que o requisito foi incorporado ao software e sua funcionalidade foi validada pelos testes e usuários finais.
Alta	Dificuldade	Significa que o requisito tem uma alta dificuldade de implementação no software.
Média	Dificuldade	Significa que a dificuldade de implementação deste requisito é média.
Baixa	Dificuldade	Significa que a dificuldade de implementação deste requisito é média.
Alta	Estabilidade	O requisito foi implementado e está estável após um período longo de utilização
Média	Estabilidade	O requisito foi implementado, porém em algumas circunstâncias apresenta bugs ou não funciona como desejado.
Baixa	Estabilidade	O requisito foi implementado e está apresentando vários problemas que devem ser corrigidos.
Suporte	Origem	A origem deste requisito foi através de contato que o usuário fez com a empresa.
Parceiros	Origem	Este requisito teve como origem um contato de um Parceiro ou empresa Coligada.
Competidor	Origem	Significa que o requisito foi identificado no software de algum competidor.
Grandes Consumidores	Origem	Este tipo de origem de requisito tem tratamento diferenciado, pois envolve consumidores que dão grande retorno financeiro à empresa.

Nome	Propriedade	Identifica o nome deste Use Case.
Breve Descrição	Propriedade	Registro de uma breve descrição do Use Case, apresentando em linhas gerais seu contexto, suas fronteiras e atores envolvidos.
Sequência Típica de Eventos	Propriedade	Listagem da sequência típicas de eventos de um Use Case
Sequência Alternativa de Eventos	Propriedade	Lista uma alternativa de sequência de eventos identificada na sequência típica de eventos.
Requisito Especial	Propriedade	????
Pré-Condição	Propriedade	Uma condição que deve ser satisfeita para que o Use Case seja iniciado.
Pós-Condição	Propriedade	Condições que o sistema se encontra após a sequência de eventos.

11.1 Status

[Set after negotiation and review by the project management team. Tracks progress during definition of the project baseline.]

Proposed	Used to describe features that are under discussion but have not yet been reviewed and accepted by the "official channel," such as a working group consisting of representatives from the project team, product management and user or customer community.
Approved	Capabilities that are deemed useful and feasible and have been approved for implementation by the official channel.
Incorporated	Features incorporated into the product baseline at a specific point in time.