

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIAS DA COMPUTAÇÃO – BACHARELADO

GERÊNCIA DE REQUISITOS COM ADOÇÃO DE PADRÕES

TIAGO WANKE MARQUES

BLUMENAU
2008

2008/2-26

TIAGO WANKE MARQUES

GERÊNCIA DE REQUISITOS COM ADOÇÃO DE PADRÕES

Trabalho de Conclusão de Curso submetido à Universidade Regional de Blumenau para a obtenção dos créditos na disciplina Trabalho de Conclusão de Curso II do curso de Ciências da Computação — Bacharelado.

Prof. Everaldo Artur Grahl, Titulação - Orientador

**BLUMENAU
2008**

2008/2-26

GERÊNCIA DE REQUISITOS COM ADOÇÃO DE PADRÕES

Por

TIAGO WANKE MARQUES

Trabalho aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II, pela banca examinadora formada por:

Presidente:

Prof. Everaldo Artur Grahl, Mestre – Orientador, FURB

Membro:

Prof. Fabiane Barreto Vavassori Benitti, Doutora – FURB

Membro:

Prof. Marcel Hugo, Mestre – FURB

Blumenau, 11 de fevereiro de 2009

Dedico este trabalho à minha família, em especial minha mãe, Sheila Wanke.

AGRADECIMENTOS

À Deus.

À minha mãe, Sheila Wanke, por todo o seu amor, dedicação e compreensão.

Ao meu pai, Nelson Marques.

Ao orientador Everaldo Artur Grahl, pelo suporte, correções e incentivo.

A Cesar Rodrigo Bagatoli, grande amigo e meu “professor” de PHP.

Aos amigos Tiago Piske e Rodrigo Hackbart, por toda ajuda durante o período de graduação, pela amizade, pelos trabalhos e diversões.

À minha namorada Mirella de Melo, por estar sempre ao meu lado me apoiando.

À empresa Naxes, na qual me proporcionou recursos para o desenvolvimento deste trabalho.

Além do meu agradecimento, toda a minha admiração ao meu avô, Artur Wanke.

Technology is the rock n'roll of our times.

Fabio Seixas

RESUMO

O presente trabalho detalha a extensão de uma ferramenta de gerência de requisitos denominada *requisite manager*, incorporando padrões de requisitos minimizando os erros cometidos durante a criação dos requisitos. Além disso, a ferramenta foi reestruturada através da recodificação no paradigma orientado a objetos.

Palavras-chave: Gerência de requisitos. Padrões de requisitos. Requisite manager.

ABSTRACT

This work details the extension of a requirements management tool called requisite manager, incorporating requirements pattern minimizing the errors during the creation of requirements. Moreover, the tool has been restructured through the recoding in the object oriented paradigm.

Key-words: Requisite management. Requisite patterns. Requisite manager.

LISTA DE ILUSTRAÇÕES

Figura 1 – Engenharia de software em camadas	16
Figura 2 – Charge que ilustra o levantamento de requisitos	19
Figura 3 – Padrões de requisitos.....	21
Quadro 1 – Exemplo de padrões de requisitos	23
Quadro 2 – Exemplo de padrões de requisitos	24
Figura 4 – Figura mostra a tela da ferramenta de gerenciamento de requisitos	25
Figura 5 - Tela de cadastro de requisitos do sistema.....	27
Figura 6 - Imagem da uma tela do software GatherSpace.....	28
Figura 7 – Diagrama de casos de uso	31
Quadro 3 – Caso de uso cadastra padrão.....	31
Quadro 4 – Caso de uso listar padrões	32
Quadro 5 – Caso de uso listar padrões	33
Quadro 6 – Caso de uso cadastra requisito.....	33
Quadro 7 – Caso de uso gera relatório de requisitos	34
Figura 8 – Diagrama de classes do domínio <i>view</i>	35
Figura 9 – Diagrama de classes do domínio HTML	36
Figura 10 – Diagrama de classes do domínio sistema.....	37
Figura 11 – Diagrama de classes do domínio conexão com o banco de dados.....	38
Figura 12 – Diagrama de classes do domínio requisitos	39
Figura 13 – Diagrama de classes do domínio padrões	40
Figura 14 – Diagrama de seqüência para cadastro de um padrão de requisito.....	41
Figura 15 – Diagrama de seqüência para cadastro de um requisito	43
Figura 16 – Diagrama de seqüência para geração de um relatório de requisitos	45
Figura 17 – Diagrama de seqüência para listar padrões de requisitos.....	46
Figura 18 – Diagrama de seqüência para visualizar a lista de exemplo de padrões de requisitos	48
Figura 19 – Modelo de entidade e relacionamento.....	50
Quadro 8 – Tabela que armazena os atributos de um tipo de requisito.....	50
Quadro 9 – Tabela que armazena as opções dos atributos de múltipla escolha	50
Quadro 10 – Tabela que armazena os exemplos dos padrões de requisitos	50
Quadro 11 – Tabela que armazena o glossário de um projeto.....	50

Quadro 12 – Tabela que armazena os padrões de requisitos.....	51
Quadro 13 – Tabela que armazena os projetos.....	51
Quadro 14 – Tabela que armazena os usuários de um projeto	51
Quadro 15 – Tabela que armazena os requisitos de um projeto.....	51
Quadro 16 – Tabela que armazena os templates do sistema	51
Quadro 17 – Tabela que armazena os tipos de requisitos de um template.....	51
Quadro 18 – Tabela que armazena os vínculos do template	51
Quadro 19 – Tabela os tipos de dados existentes no sistema	52
Quadro 20 – Tabela que armazena os tipos de requisitos.....	52
Quadro 21 – Tabela que armazena os atributos de um tipo de requisito.....	52
Quadro 22 – Tabela que armazena os usuários do sistema	52
Quadro 23 – Tabela que armazena o valor dos atributos de um requisito.....	52
Quadro 24 – Tabela que armazena os vínculos dos templates	53
Quadro 25 – Tabela que armazena os vínculos dos requisitos	53
Figura 20 – Ambiente de desenvolvimento ZEND	54
Quadro 26 – Utilização da biblioteca xajax.....	55
Quadro 27 – Utilização da biblioteca FPDF.....	55
Figura 21 – Ambiente de gerência de dados do SQLYog	56
Quadro 28 – Chamada de adicionar padrão requisito no banco dados.....	57
Quadro 29 – Arquivo HTML lista de padrões de requisitos	58
Quadro 30 – Arquivo PHP para manipulação do HTML.....	59
Quadro 31 – Arquivo PHP para manipulação do HTML.....	60
Figura 22 – Interface do arquivo CHM gerado	61
Quadro 32 – Método de criação da URL amigável.....	62
Quadro 33 – Código fonte arquivo lib e método autoload	63
Figura 23 – Tela de <i>login</i>	64
Figura 24 – Tela inicial do módulo usuário.....	64
Figura 25 – Tela de gerência de requisitos	65
Figura 26 – Tela de cadastro de requisito funcional.....	66
Figura 27 – Tela listando os padrões de requisitos funcionais	67
Figura 28 – Tela de padrão de requisito funcional de relatório.....	68
Figura 29 – Tela da lista de requisitos cadastrados	69
Figura 30 – Tela da lista de padrões de requisitos cadastrados	70
Figura 31 – Tela para adicionar um novo padrão de requisito	70

Figura 32 – Tela da lista de exemplos do padrão de requisitos de entidade ativa.....	71
Figura 33 – Tela do formulário para adicionar um novo exemplo de padrão de requisito	72
Figura 34 – Tela de opções de relatórios do sistema.....	72
Figura 35 – Tela de opções de tipo de relatórios para relatório de requisitos	73
Figura 36 – Relatório de requisitos em formato PDF.....	73
Figura 37 – Tela para escolher a matriz de rastreabilidade	74
Figura 38 – Tela de edição da matriz de rastreabilidade	74
Quadro 34 – Codificação de Marquardt (2004) para adicionar um <i>template</i>	75
Quadro 35 – Codificação atual para adicionar um <i>template</i>	76
Quadro 36 – Principais diferenciais alcançados com a execução desse trabalho.....	77
Quadro 37 – relatório de requisitos da ferramenta gerado pela ferramenta	86

LISTA DE SIGLAS

AJAX – *Asynchronous Javascript And XML*

CHM – *Microsoft Compiled HTML Help*

CMMI – *Capability Maturity Model Integrated*

CSS – *Cascading Style Sheets*

EA – *Enterprise Architect*

GRE – Gerência de Requisitos

HTML – *HyperText Markup Language*

IEEE – Institute of Electrical and Electronics Engineers

MER – Modelo Entidade e Relacionamento

PDF – *Portable Document Format*

PHP – *Hypertext Preprocessor*

RF – Requisito(s) Funcional(is)

RNF – Requisito(s) Não Funcional(is)

SGBD – Sistema de Gerenciamento de Banco de Dados

SQL – *Structured Query Language*

TCC – Trabalho de Conclusão de Curso

UML – *Unified Modeling Language*

URL – *Uniform Resource Locator*

SUMÁRIO

1 INTRODUÇÃO.....	14
1.1 OBJETIVOS DO TRABALHO	15
1.2 ESTRUTURA DO TRABALHO	15
2 FUNDAMENTAÇÃO TEÓRICA	16
2.1 ENGENHARIA DE SOFTWARE.....	16
2.2 GERÊNCIA DE REQUISITOS	17
2.3 PADRÕES DE REQUISITOS	20
2.4 TRABALHOS CORRELATOS.....	24
2.4.1 Ferramenta web para gerenciamento de requisitos de software.....	24
2.4.2 Ferramenta de gerência de requisitos de software integrada com Enterprise Architect .	26
2.4.3 Ferramenta de Apoio a Gerência de Requisitos Baseada no Modelo CMMI.....	26
2.4.4 Gatherspace	27
3 DESENVOLVIMENTO.....	29
3.1 REQUISITOS DA FERRAMENTA.....	29
3.2 ESPECIFICAÇÃO	30
3.2.1 Diagrama de caso de uso.....	30
3.2.2 Diagrama de classes	34
3.2.3 Diagramas de seqüência.....	41
3.2.4 Diagrama entidade relacionamento.....	49
3.3 IMPLEMENTAÇÃO	53
3.3.1 Tecnologias e ferramentas utilizadas	53
3.3.2 Implementação da funcionalidade de adicionar padrões	56
3.3.3 Implementação da funcionalidade de listar padrões e funcionamento do template.....	57
3.3.4 Arquivo de documentação do código fonte	60
3.3.5 URL amigável	61
3.3.6 Arquivo lib e o método autoload.....	62
3.3.7 Operacionalidade da implementação	63
3.4 RESULTADOS E DISCUSSÃO	75
4 CONCLUSÕES.....	78
4.1 EXTENSÕES	78
REFERÊNCIAS BIBLIOGRÁFICAS	79

APÊNDICE A – Relatório de requisitos.....	81
ANEXO A – Padrões de requisitos de software.....	87
ANEXO B – Padrões de requisitos fundamentais Schmidt (2008)	90

1 INTRODUÇÃO

Sommerville (2003, p. 82) afirma que em alguns casos requisito de software é como uma declaração abstrata, de alto nível, de uma função que o sistema deve fornecer ou de uma restrição do sistema. Já Pfleeger (2004, p. 111) define requisitos como uma característica do sistema ou a descrição de algo que o sistema é capaz de realizar, para atingir os seus objetivos. Segundo Paula Filho (2001, p. 5), a não especificação dos requisitos é uma situação tão absurda quanto querer resolver um problema sem escrever o respectivo enunciado: existe grande risco de resolver o problema errado. Muitas falhas podem ocorrer devido a um mau levantamento de requisitos.

Segundo Rocha (2001), uma má definição de requisitos nos estágios iniciais do processo de desenvolvimento pode resultar em altos custos de manutenção do sistema, pois diversas tarefas do ciclo de desenvolvimento de software deverão ser refeitas.

O levantamento de requisitos não é uma tarefa simples, pois deve ser tratada com muito cuidado detalhando cada peculiaridade que o sistema deve realizar. Isso é complicado levando em consideração que geralmente não são documentados pequenos detalhes que fazem grandes diferenças, talvez por parecer que não são tão importantes ou simplesmente por esquecê-los.

Uma alternativa interessante para acelerar e padronizar requisitos é a adoção de padrões. Os padrões de requisitos, segundo Decarle e Grahl (2008, p. 1), são soluções prévias consideradas boas para resolver problemas na área de engenharia de requisitos.

Quando é realizada a elicitação dos requisitos, os usuários têm a tendência natural de omitir detalhes importantes do processo, que para eles seja algo extremamente óbvio, no entanto quem realiza a elicitação dos requisitos não tem conhecimento dos detalhes do processo que se quer automatizar. Surge assim um verdadeiro problema pela frente, pois os analistas de sistemas são induzidos a ver somente o óbvio. (DERCARLE; GRAHL, 2008, p. 2).

Qualquer software de tamanho considerável, para ser desenvolvido com qualidade atingindo as necessidades do cliente, antes de ser desenvolvido deve ser cuidadosamente projetado, para isto, utiliza-se engenharia de software. Já existem ferramentas que auxiliam neste processo, porém, suas licenças geralmente são muito caras e não são feitas para serem acessadas via web.

Diante da dificuldade abordada, foi desenvolvida no Departamento de Sistemas e

Computação (DSC) da Universidade Regional de Blumenau (FURB), por Luciano Marquardt uma ferramenta web para gerenciamento de requisitos de software (MARQUARDT, 2004). A finalidade da ferramenta inclui armazenar e organizar os requisitos de software, gerar documento de requisitos seguindo os padrões internacionais e apoiar o ensino de requisitos de software. Este trabalho é uma extensão da ferramenta desenvolvida por Marquardt, para aperfeiçoar e incrementar as funcionalidades, assim como incluir os padrões de requisitos.

1.1 OBJETIVOS DO TRABALHO

O objetivo deste trabalho é acrescentar padrões de requisitos como extensão da ferramenta *Requisite Manager*, desenvolvida pelo acadêmico Luciano Marquardt no Trabalho de Conclusão de Curso (TCC) (MARQUARDT, 2004).

Como objetivos específicos têm-se:

- a) criar um repositório de padrões de requisitos definidos por Withall (WITHALL, 2007);
- b) reestruturar o código fonte melhorando sua legibilidade.

1.2 ESTRUTURA DO TRABALHO

Este trabalho encontra-se dividido em quatro capítulos. Neste capítulo é apresentada a introdução. No segundo capítulo é vista a fundamentação teórica com propósito de comunicar ao leitor todo o conhecimento necessário das áreas envolvidas no projeto como engenharia de software, gerência de requisitos, padrões de requisitos e trabalhos correlatos. No terceiro capítulo é detalhada a implementação e especificação da ferramenta. O quarto capítulo é finalizado com a conclusão, os benefícios adquiridos com a extensão da ferramenta e as suas limitações.

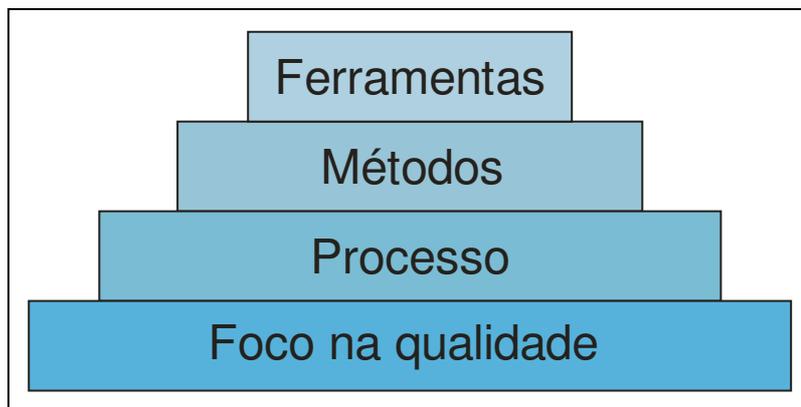
2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são relatados aspectos teóricos e conceitos de autores relacionados a engenharia de software, gerência de requisitos e padrões de requisitos que foram empregadas na realização deste trabalho. A seção 2.3 apresenta a definição de gerência de requisitos e seu propósito. Os padrões de requisitos são detalhados na seção 2.4, apresentando os padrões e seus respectivos domínios. Finalizando a fundamentação teórica na seção 2.5 com a apresentação dos trabalhos correlatos.

2.1 ENGENHARIA DE SOFTWARE

Conforme o *Institute of Electrical and Electronics Engineers* (1994) a engenharia de software é a “aplicação de uma abordagem sistemática, disciplinada e quantificável, para o desenvolvimento, operação e manutenção do software; isto é, a aplicação da engenharia ao software”. Pressman (2006, p. 17) afirma que, no entanto o que é sistemático, disciplinado e quantificável para uma equipe de software pode ser cansativo para outra. É necessária disciplina, mas também é necessária adaptabilidade e agilidade.

A engenharia de software é uma tecnologia em camadas como ilustrado na figura 1, detalhadas a seguir por Pressman (2006, p. 17-19).



Fonte: adaptado de Pressman (2006, p. 17).

Figura 1 – Engenharia de software em camadas

Toda abordagem de engenharia de software deve ter como resultado um processo de qualidade, sendo um processo de aperfeiçoamento contínuo, desenvolvendo abordagens cada vez mais efetivas para a engenharia de software. Sendo assim a base em que se apóia é o foco

na qualidade.

A camada de processo é que define quem está fazendo o quê, quando e como para se alcançar determinado objetivo. Os processos formam a base para o controle gerencial de projetos de software e estabelecem o contexto no qual os métodos técnicos são aplicados.

Os métodos abrangem um amplo conjunto de tarefas que incluem comunicação, análise de requisitos, modelagem de projeto, construção de programas, testes e manutenção. São eles que fornecem a técnica de como fazer para construir softwares, incluem atividades de modelagem e outras técnicas descritivas.

A camada de ferramentas de engenharia de software provê o apoio automatizado ou semi-automatizado para o processo e para os métodos. Quando a engenharia de software é apoiada por computador, significa que as ferramentas são integradas de modo que a informação criada por uma ferramenta possa ser usada por outra.

A prática de engenharia de software engloba conceitos, princípios, métodos e ferramentas que são aplicadas durante o processo de software. Cada projeto de engenharia de software é diferente, porém, um conjunto de princípios e tarefas genéricas aplica-se a cada atividade do processo, independentemente do projeto ou do produto. A seguir é vista a gerência de requisitos que é um dos processos pilares para a boa engenharia de software.

2.2 GERÊNCIA DE REQUISITOS

Segundo Associação para Promoção da Excelência do Software Brasileiro – Softex (2007), o propósito do processo de gerência de requisitos é gerenciar os requisitos dos produtos e componentes do projeto e identificar inconsistências entre os requisitos, os planos de projeto e os produtos de trabalho do projeto. Seu principal objetivo é controlar a evolução dos requisitos. Marquardt (2004, p. 7) define gerência de requisitos como sendo uma das atividades da engenharia de requisitos e envolve o controle das mudanças dos requisitos durante o desenvolvimento do software. Pressman (2006, p. 121) afirma que gerência de requisitos é um conjunto de atividades que ajudam a equipe do projeto a identificar, controlar e rastrear requisitos e modificações de requisitos em qualquer época, à medida que o projeto prossegue.

Conforme Paula Filho (2001, p. 88) os usuários-chaves são as pessoas capacitadas a definir os requisitos do projeto e devem estar cientes do papel essencial que desempenham na

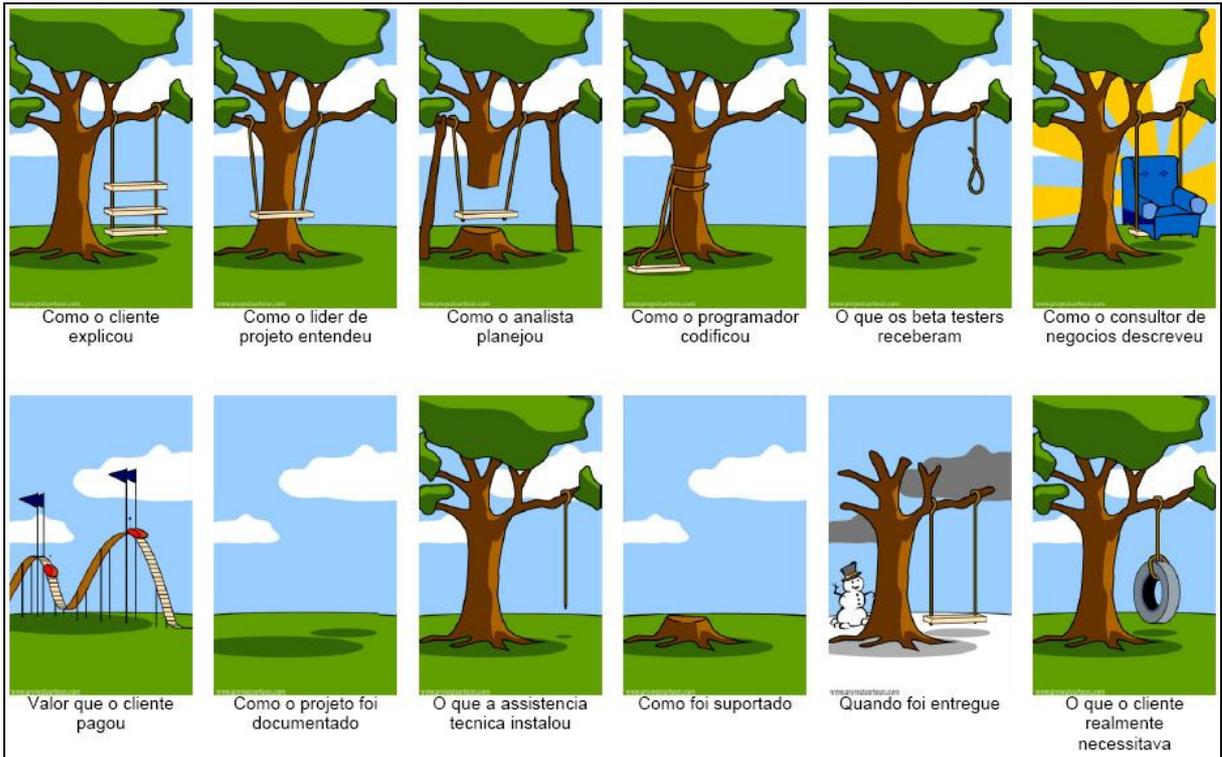
especificação dos requisitos. O custo de omissão ou especificação errônea de um requisito é 20 vezes maior se sua descoberta ou correção for realizada na fase de implementação e 200 vezes maior se for descoberta na manutenção. Mais da metade dos problemas de um software tem sua origem em erros cometidos na etapa de levantamento de requisitos (MAGELA, 2006, p. 14).

Todos os requisitos recebidos ou gerados pelo projeto são gerenciados pelo processo de Gerência de Requisitos (GRE). Também são incluídos requisitos funcionais e não-funcionais, e requisitos impostos ao projeto pela organização. A organização deve executar uma série de passos definidos e apropriados para assegurar que os requisitos definidos fornecem suporte às necessidades de planejamento e execução do projeto. Quando na etapa de elicitação de requisitos haja a participação do fornecedor de requisitos¹, os requisitos devem ser revisados para resolver questões e prevenir o mau entendimento dos mesmos. (ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO – SOFTEX, 2007, p. 1).

No mercado de hoje o cliente solicita o pedido de um sistema. No entanto, como já citado, ele tem a tendência de omitir informações que ele julgar óbvias não comunicando o sistema em detalhes, quais as atividades que deve exercer, como deve ser controlado. A figura 2 é uma charge conhecida desse problema que ilustra a realidade explicada.

A Softex (2007, p. 1) explica que o processo de gerenciamento de requisitos também deve documentar as mudanças nos requisitos e suas devidas justificativas, assim como manter a rastreabilidade bidirecional entre os requisitos e produtos de trabalho em geral identificando inconsistência no processo.

¹ Fornecedor de requisitos é a pessoa autorizada a participar da definição dos requisitos e a solicitar modificações



Fonte: Gomes (2008).

Figura 2 – Charge que ilustra o levantamento de requisitos

Uma vez identificados os requisitos, tabelas de rastreamento são desenvolvidas. Cada tabela relaciona os requisitos identificados a um ou mais aspectos do sistema ou de seu ambiente. As tabelas podem ser consultadas para entender como a modificação em um requisito afetará diferentes aspectos do sistema a ser construído. (PRESSMAN, 2006, p. 121).

Entre muitas tabelas de rastreamento possíveis estão as seguintes:

- tabela de rastreamento de características: mostra o relacionamento de requisitos com características importantes do sistema/produto observáveis pelo cliente;
- tabela de rastreamento de fontes: identifica a fonte de cada requisito;
- tabela de rastreamento de dependência: indica como os requisitos estão relacionados uns aos outros;
- tabela de rastreamento de subsistemas: caracteriza os requisitos pelo(s) subsistema(s) que eles governam;
- tabela de rastreamento de interface: mostra como os requisitos se relacionam com as interfaces internas e externas do sistema.

2.3 PADRÕES DE REQUISITOS

Decarle e Grahl (2008, p. 1) comentam que a área de requisitos de software é uma das maiores causadoras de problemas no desenvolvimento de um software. No entanto, o problema não se encontra especificamente na área e sim no processo de levantamento dos requisitos, como problemas na comunicação dos envolvidos nessa etapa, documentação incorreta dos requisitos, falta de revisão e avaliação dos requisitos e outros problemas. Com foco no problema a área de padrões de requisitos vem sendo uma solução a ser adotada para agregar maior confiabilidade a etapa do levantamento de requisitos, “Padrões de requisitos de softwares vêm sendo utilizados como alternativa para melhorar a qualidade dos documentos de requisitos.” (DECARLE; GRAHL, 2008, p. 1).

Diversas etapas durante o processo são genéricas a todos os projetos de um domínio de aplicação específico. Estas etapas podem ser classificadas como uma classe, uma função ou até um comportamento que é possível de serem reutilizadas quando se modela muitas aplicações.

A primeira idéia que se tem ao imaginar padrões de software é uma biblioteca de componentes que podem ser reutilizados para melhorar a produtividade e garantir a consistências dos sistemas, já que os componentes reutilizáveis foram testados provavelmente estão sendo utilizados em outros sistemas em produção. (DECARLE; GRAHL, 2008, p. 2).

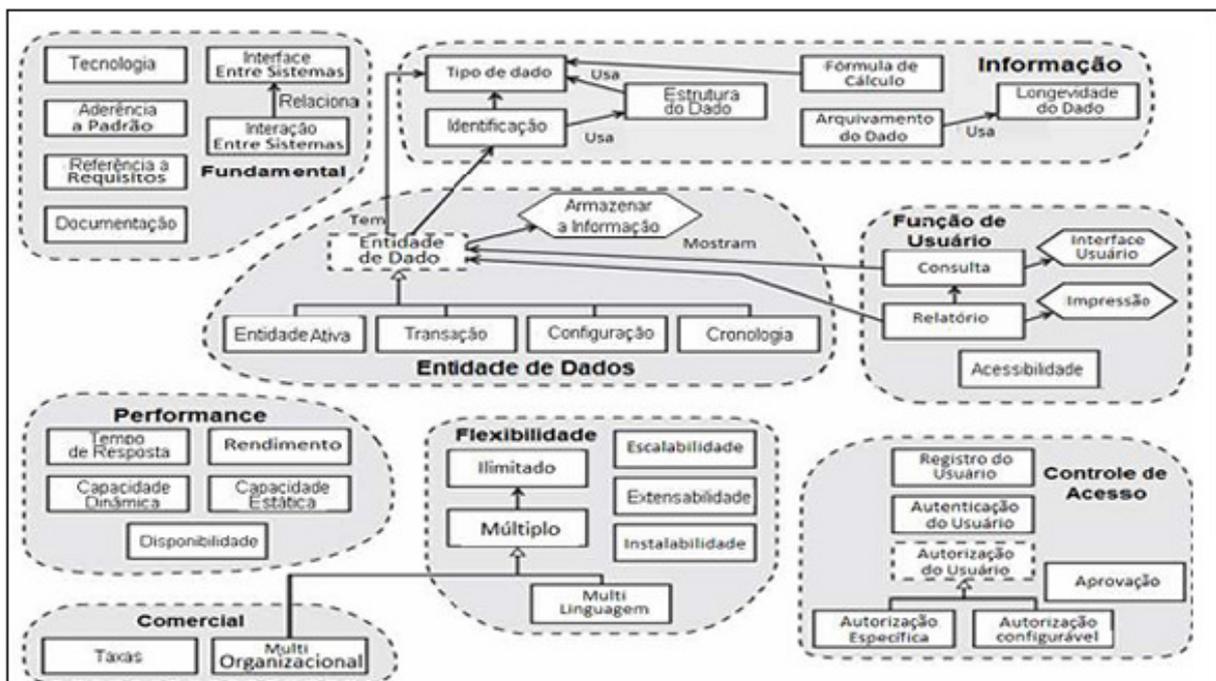
Um padrão de requisito funciona como um guia para escrever um tipo particular de requisito. O padrão explica como descrever determinado tipo de requisito, como expressá-lo e revela possíveis requisitos adicionais implícitos. Padrões de requisitos são utilizados somente para escrever um único requisito, pois cada padrão contém informações detalhadas e precisas sobre esse. Um padrão não se preocupa em representar o sistema como um todo, atendo-se apenas ao tipo de requisito específico (WHITALL, 2008).

Segundo Tagliati, Johnson e Roussos (2005) um padrão descreve um problema recorrente em um domínio de problema particular. Isso permite aos profissionais a possibilidade de reutilizar as soluções várias vezes sem necessidade de estudar um mesmo problema novamente. A idéia principal é produzir um catálogo de padrões de requisitos elegante, objetivo, extensível e reutilizável.

Padrões de requisitos são armazenados em um repositório de modo que os engenheiros de requisitos possam usar facilidades de busca para encontrá-los e reusá-los (PRESSMAN, 2006, p. 137).

Decarle e Grahl (2008, p. 3) apresentam em seu artigo um catálogo de trinta e sete

padrões de requisitos separados em oito domínios, propostos por Withall (2007) ilustrados na figura 3 e comentados a seguir.



Fonte: Decarle e Grahl (2008).

Figura 3 – Padrões de requisitos

O domínio fundamental trata de padrões relacionados a especificação da tecnologia que deve ou não ser utilizada para executar ou construir um sistema, ou então com que tecnologia o sistema deve ser compatível; especificar a aderência do sistema a um determinado padrão; especificar que requisitos externos devem ser atendidos como se fossem da especificação corrente; especificar que precisa ser produzido um determinado tipo de documento; definir detalhes de interface do sistema especificado e qualquer componente externo, com o qual irá interagir e especificar um tipo particular de interação entre sistema.

O domínio performance tem como preocupação as especificações de tempo de resposta, a capacidade de processamento simultâneo, capacidade de armazenamento, quando estará disponível e especificar um índice que o sistema deve ser capaz de executar em alguma operação de entrada ou saída.

O domínio informação foca aspectos como a definição de um item de informação deve ser apresentado ou representado, como será a identificação única das entidades de dados, como será composto um determinado item de informação, como calcular determinado tipo de valor ou determiná-lo através de alguns passos lógicos, especificar a movimentação ou cópia de dados de um local para outro e por quanto tempo um certo tipo de dado deve ser mantido ou por quanto tempo deve estar disponível.

A entidade de dados é o domínio responsável por definir um tipo de entidade, sua vida

útil, para qual as informações são armazenadas, definir eventos que ocorrerão ao iniciar uma transação, definir configurações do sistema e como registrar determinados eventos do sistema.

Informações adquiridas através de consultas, relatórios são definidas pelo domínio função do usuário, assim como a acessibilidade do sistema aos usuários com deficiências ou outra necessidade específica.

Atributos como multi-linguagem, quanto o sistema poderá expandir sem problemas, suportando o crescimento do volume de negócios, ou como poderá ser estendido por terceiros, o grau de dificuldade de instalação do sistema são definidos pelo domínio flexibilidade. Assim como a especificação de como um sistema deve se comportar para acomodar múltiplas moedas ou empresas simultaneamente.

O domínio controle de acesso prevê o registro de novos usuários, sua autenticação, ou determinadas autorizações específicas e personalizadas pelos usuários, assim como a aprovação do registro do usuário por uma pessoa responsável.

O domínio comercial especifica o tipo de estrutura organizacional que o sistema deve suportar e pagamentos de taxas ou impostos de qualquer natureza que deve ser calculado ou apresentado.

No anexo A segue um catálogo com os trinta e sete padrões de requisitos apresentados no artigo de Decarle e Grahl (2008, p. 7-9) separados em oito domínios comentados anteriormente. Alguns exemplos de padrões de requisitos propostos por Whitall (WHITALL, 2008) estão listados no quadro 1. No anexo B estão os 37 padrões de requisitos propostos por Schmidt (SCHMIDT, 2008).

No campo padrão estão os padrões: relatório, consulta e entidade ativa. No campo título tem-se um modelo de título para cada um dos tipos. No campo descrição, para cada tipo de padrão, um texto que detalha sua funcionalidade.

Padrão	Título	Descrição
Relatório	<<nome do relatório>>	<p>Deve haver um relatório que mostra <<Informação para mostrar>> <<Critério de Seleção>> ordenadas por <<Seqüência de classificação>> O objetivo do presente relatório é a <<Intenção de Negócios>>.</p> <p>Para cada <<Nome tipo de item>>, o relatório deve mostrar o seguinte:</p> <ul style="list-style-type: none"> • <<Valor nome 1>> • <<Valor nome 2>> • ... <p>[Os itens a serem exibidos podem ser especificados por entrar em qualquer um dos seguintes critérios de seleção:</p> <ul style="list-style-type: none"> • <<critério de seleção 1>> • <<critério de seleção 2>>

Padrão	Título	Descrição
		<ul style="list-style-type: none"> •... Totais devem ser indicados para «totalizando níveis». [Uma nova página deve ser iniciada por «atirar níveis de páginas».] [O relatório destina-se a ser executado automaticamente «Detalhes executados automaticamente».]
Consulta	<<Nome da Consulta>>	Deve haver uma [«nome da consulta"] consulta que revela <<Informação para mostrar>>. O seu objetivo é <<Intenção de negócio>>. Para cada <<Nome da entidade>>, a consulta deve mostrar o seguinte: <ul style="list-style-type: none"> • "Informação item 1» • "Informação item 2» • ... [Os itens a serem exibidos serão listados em seqüência <<Classificar seqüência de detalhes>>] [Os itens a serem exibidos pode ser especificado por entrar qualquer um dos seguintes critérios de seleção: <ul style="list-style-type: none"> • <<Critério de seleção 1>> • <<Critério de seleção 2>> • ...] [O usuário será capaz de navegar <<Detalhes de navegação do usuário>>.] [O usuário deve ser capaz de interagir com a consulta <<Detalhes da interação do usuário>>.] [As informações mostradas são atualizadas automaticamente <<atualização automática dos detalhes>>.]
Entidade ativa	<<Nome da Entidade>>	O sistema deve armazenar as seguintes informações sobre o <<Nome da Entidade>>: <ul style="list-style-type: none"> • "Dados item 1 descrição". • ... Uma "Entidade nome" é "Entidade explicação". Cada "Entidade nome" é exclusivamente identificada por "identificador Entidade (s)".

Quadro 1 – Exemplo de padrões de requisitos

O quadro 2 apresenta para cada tipo de requisito listado no quadro 1 um respectivo exemplo.

Tipo	Título	Definição
Relatório	Relatório de clientes	Deve haver um relatório que lista todos os clientes da loja. Ordenados pelo nome do cliente. O objetivo deste relatório é listar todos os clientes da loja. Para cada cliente, o relatório deve mostrar o seguinte: <ul style="list-style-type: none"> • nome • e-mail • sexo Ao final do relatório deve mostrar o total de clientes da loja.
Consulta	Consulta de clientes	Deve haver uma consulta de clientes que revela os dados dos clientes. O seu objetivo é acessar as informações dos clientes

Tipo	Título	Definição
		cadastrados no sistema. Para cada cliente, a consulta deve mostrar o seguinte: <ul style="list-style-type: none"> • Nome • E-mail • Sexo Os itens a serem exibidos serão listados ordenados pelo nome do cliente. Os itens a serem exibidos podem ser especificados por entrar em qualquer um dos seguintes critérios de seleção: <ul style="list-style-type: none"> • Nome • E-mail • Sexo O usuário será capaz de navegar pela lista de clientes. O usuário poderá remover clientes da consulta.
Entidade Ativa	Informações Básicas sobre os clientes	O sistema deve armazenar as seguintes informações sobre o cliente: <ul style="list-style-type: none"> • ID do cliente (como definido no requisito XR99.1). • Senha. • Informações de contatos pessoais (como definido no requisito XR99.2). • Informações de cartão de crédito (tal como definido no requisito XR99.3). • Data de nascimento. • Data de inscrição. • Status (ativo, bloqueado ou encerrado). Isso nunca é mostrado ao cliente. Cada cliente é identificado unicamente pelo seu ID de cliente.

Quadro 2 – Exemplo de padrões de requisitos

2.4 TRABALHOS CORRELATOS

Nesta seção são descritos quatro trabalhos correlatos que contribuíram na definição e revisão dos textos. Todos eles focam na gerência de requisitos.

2.4.1 Ferramenta web para gerenciamento de requisitos de software

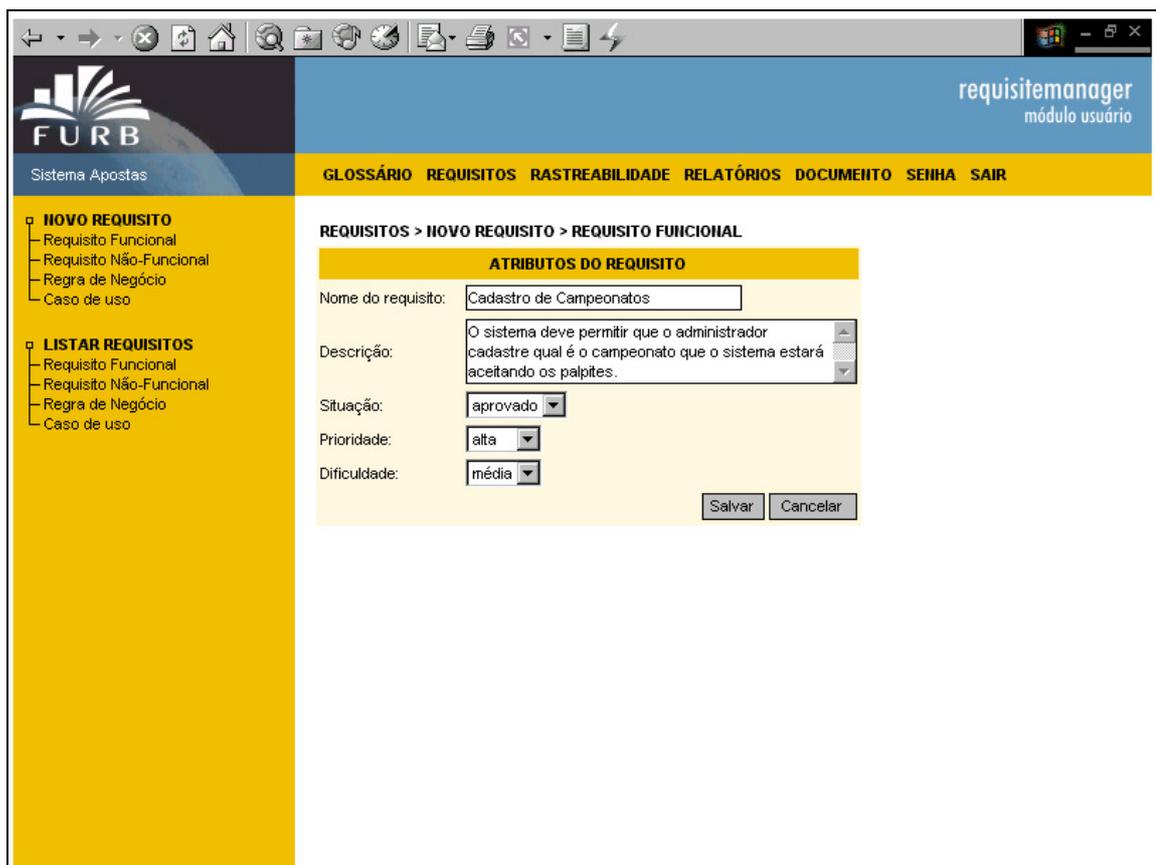
Em Marquardt (2004, p. 7) é apresentada a construção da ferramenta web *Requisite Manager* foco deste TCC. A ferramenta é essencialmente acadêmica e auxilia a aplicação dos

conceitos de gerência de requisitos em uma disciplina como a de requisitos de software (MARQUARDT, 2004, p. 29).

Segundo Marquardt (2004, p. 29) a ferramenta é dividida em três módulos:

- a) *master*: no módulo *master* é possível autenticar os acessos, são cadastrados projetos, tipos de requisitos, atributos dos tipos de requisitos, vínculos permitidos entre os tipos de requisitos e *templates*;
- b) administrador: esse módulo já está em um nível de projeto e ao ser configurado e liberado para uso não pode ser mais editado. Neste módulo são cadastrados os usuários do projeto e selecionados os tipos de requisitos, atributos e vínculos;
- c) usuário: no módulo usuário os requisitos são gerenciados, isso significa que eles podem ser cadastrados, alterados e visualizados, nesse nível de acesso. Também é possível obter relatórios do projeto no módulo de usuário.

A figura 4 apresenta uma tela da ferramenta com a listagem dos projetos e o menu do sistema.



Fonte: Marquardt (2004, p. 51).

Figura 4 – Figura mostra a tela da ferramenta de gerenciamento de requisitos

2.4.2 Ferramenta de gerência de requisitos de software integrada com Enterprise Architect

Batista (2007, p. 6) resume seu projeto como uma ferramenta para gerência de requisitos de software integrada com o Enterprise Architect (EA) que gera um documento de especificação de requisitos de software no padrão IEEE-830-1998.

Batista (2007, p. 13) explica que seu projeto propõe suprir a deficiência do EA, permitindo que se tenha o gerenciamento mais eficiente dos requisitos, focando na produção de um documento de especificação de requisitos completo.

Essa ferramenta oferece a funcionalidade de geração de uma matriz de rastreabilidade, facilitando o rastreamento dos requisitos e casos de uso, sinalizando as alterações realizadas nos elementos relacionados na matriz, visualizando assim os impactos de futuras alterações ou extensões que o projeto de software venha a sofrer. (BATISTA, 2007, p. 6).

2.4.3 Ferramenta de Apoio a Gerência de Requisitos Baseada no Modelo CMMI

Meisen (2005, p. 32) descreve sua ferramenta como uma auxiliadora para gerentes de projetos e analistas a gerenciar os requisitos de software. Meisen (2005, p. 7) também afirma que o foco principal da ferramenta está em atender algumas das recomendações do modelo *Capability Maturity Model Integrated* (CMMI) nos níveis 2 e 3.

A ferramenta é dividida em três módulos. O módulo administrativo será utilizado pela pessoa responsável pela ferramenta. Essa será a única pessoa com permissão para fazer o cadastro inicial dos projetos e o cadastro de novos usuários (MEISEN, 2005, p. 33). Um segundo módulo denominado gerencial será utilizado pelos gerentes de projetos, que são definidos pelo administrados a cada novo projeto. Segundo Meisen (2005, p. 33), os gerentes são responsáveis por manter os cadastros dos tipos de requisitos, atributos, vínculos e por manter os projetos dos quais são responsáveis.

A figura 5 mostra uma tela da ferramenta onde possui um menu e um formulário para cadastro de um novo requisito.

The screenshot shows a web browser window titled 'RW0010PI - Microsoft Internet Explorer'. The address bar shows 'http://localhost:8080/requisiteWEB/navegacao.do?target=inserirRequisito'. The page content includes a user profile for 'Mariane Meisen' with a 'Logout' button. A navigation bar contains links for 'Stakeholders', 'Entrevistas', 'Glossário', 'Requisitos', 'Solicitação de Alteração', 'Relatórios', and 'Senha'. Below this is a breadcrumb trail: 'Home >> Cadastros >> Requisito (Inclusão)', followed by buttons for 'Consultar', 'Inserir', 'Voltar', and 'Ajuda'. The main heading is 'Cadastrar Requisito'. The form fields are: 'Projeto' (text box with 'LEA' and a 'Lista' button), 'Tipo' (dropdown menu with 'Requisito Funcional'), 'Nome' (text box with 'Acesso' and 'FPA's' text box with '10'), and 'Objetivo' (text box with 'O sistema deve ter o controle de quem pode acessar'). At the bottom of the form are 'Salvar' and 'Cancelar' buttons. A footer contains 'Administrativo', 'Gerencial', and 'Análise' buttons. The status bar at the bottom shows 'Concluído' and 'Intranet local'.

Fonte: Meisen (2005, p. 67).

Figura 5 - Tela de cadastro de requisitos do sistema

2.4.4 Gatherspace

GatherSpace é um sistema administrador de requisitos que proporciona a colaboração entre a equipe de negócio e a equipe técnica no levantamento e manutenção de requisitos durante o ciclo de vida do produto (GATHERSPACE, 2008). É um software que é de baixo custo e não possui instalação.

A ferramenta serve para empresas de todos os tamanhos, de pequenas a grandes corporações. Oferece uma plataforma completa para gerenciar o processo de análise de negócio (GATHERSPACE, 2008).

Na figura 6 é apresentada a imagem da janela principal do sistema com a listagem dos requisitos criados, alguns de seus atributos e o menu do software.

GATHERSPACE.COM Help | Getting Started | Customer Service | Log Out

Select Project: **Online Magazine Store**

Requirements | Projects | Reporting | Issues | Account Center | Search

Packages | Features | Requirements | Actors | Uses Cases | Glossary | BaseLine

Features

Key	Rank	Name	Package	Priority	Assigned To	Status
			Any Package	Any	All Users	Any Status
F7	2	Feat Test1	CRM		Not assigned	Submitted
F6		Feature (with subfeatures)		High	Not assigned	Submitted
F5		System will offer CRM functionality	CRM		Not assigned	Submitted
SR4		System will render pages within 2 seconds.				
UC16		Pick up CSR (customer service requests)				
UC17		Send emails back to the customer				
F4		Marketing upsell after purchase		High	Not assigned	Released
F3	1	Purchase books online	Accounting S...		darrenlevy@..	Submitted
F2		Browse books online	Ecommerce	Medium	Not assigned	Cancelled
UC7		Manages Account				
F1		Shopping Cart Functionality	CRM		Not assigned	Submitted
SR6		Delete items from basket				
UC11		Put books in shopping cart				
UC12		Remove book from shopping cart				
UC13		Clear Shopping Cart				

Add Feature Recs 1 - 9 of 9

Copyright © 2007 GatherSpace.com | Privacy Policy

Fonte: Gatherspace (2008).

Figura 6 - Imagem da uma tela do software GatherSpace

3 DESENVOLVIMENTO

Este capítulo apresenta a análise e especificação de requisitos e a modelagem da ferramenta. Além disso, são mostrados os códigos de implementação e operacionalidade da ferramenta.

3.1 REQUISITOS DA FERRAMENTA

A seguir são apresentados os Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF) que foram melhorados ou estendidos nessa ferramenta utilizando os padrões de requisitos. Para cada requisito é apresentada a classificação do padrão de requisito utilizado.

- a) RF01 - Acesso do usuário via navegador da web: Todas funções do usuário serão acessíveis através de um navegador web.
- b) Classificação - tecnologia
- c) RF02 – Navegadores populares da web: A interface do usuário deve ser baseada na WEB e todas as funções deve funcionar plenamente com os seguintes browsers: Internet Explorer 6.0, Mozilla Firefox 3.0, e Chrome.
- d) RF01 – Cadastro de padrão de requisito: o sistema deve armazenar as seguintes informações sobre o cadastro de requisito: tipo, título, descrição. O cadastro de padrões de requisitos será utilizado para criar o repositório de padrões de requisito. Cada padrão de requisito é exclusivamente identificado pelo seu título. Classificação – entidade ativa;
- e) RF02 – Cadastro de exemplo de padrão de requisito: o sistema deve armazenar as seguintes informações sobre o exemplo de padrão de requisito: nome e descrição. Um exemplo de padrão de requisito será utilizado para exemplificar a utilização de um padrão de requisito. Cada exemplo de padrão de requisito é exclusivamente identificado por seu nome. Classificação – entidade ativa;
- f) RF03 – Utilizar padrão: deve haver uma função de utilizar um padrão de requisito para um requisito a ser criado/editado. Cada padrão de requisito utilizado deve conter a descrição. A utilização de um padrão é feita para facilitar a criação de requisitos de um projeto. Classificação – transação;

- g) RNF01 – Relatório de requisitos no formato PDF: o formato PDF deve ser usado para gerar o relatório de requisitos tanto no módulo *master* como no módulo usuário. Classificação – tecnologia;
- h) RNF02 – Utilizar AJAX, CSS e javascript: as tecnologias AJAX, *Cascading Style Sheets* (CSS) e javascript devem ser usadas para melhorar o desempenho da ferramenta como também aumentar a legibilidade do código fonte da mesma. Classificação – tecnologia;
- i) RNF03 – Documentação do código fonte: deve haver um documento de ajuda que contém a documentação de todas as classes e seus respectivos métodos utilizadas para desenvolver a ferramenta. Deve ser no formato .chm (arquivo de ajuda do windows). Ele deve ser escrito em português. Classificação – documentação.

No apêndice A segue todos os requisitos da ferramenta listados em um relatório gerado com a própria ferramenta.

3.2 ESPECIFICAÇÃO

Para definir a especificação foram utilizados diagramas da Unified Modeling Language (UML):

- a) diagrama de caso de uso;
- b) diagrama de classe;
- c) diagrama de seqüência.

Para a diagramação foi utilizada a ferramenta CASE EA.

3.2.1 Diagrama de caso de uso

No diagrama de caso de uso são apresentadas as ações que o usuário pode realizar nas funcionalidades desenvolvidas na extensão da ferramenta. Neste TCC foram incluídos somente os casos de uso incorporados ou melhorados. A figura 7 ilustra os casos de uso .

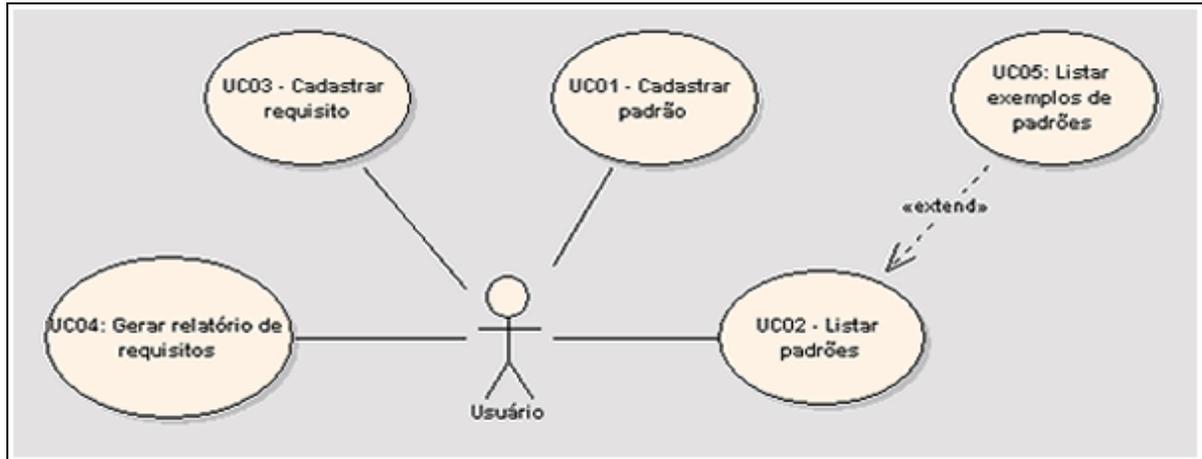


Figura 7 – Diagrama de casos de uso

O caso de uso ilustrado na figura 7 apresenta as funcionalidades estendidas e as novas funcionalidades incorporadas à ferramenta. O usuário do sistema pode interagir com quatro funcionalidades principais do sistema, sendo elas o cadastro de padrão de requisitos onde será possível cadastrar padrões de requisitos, listar padrões cadastrados, onde poderá ver a lista dos padrões cadastrados e optar por listar os exemplos de um padrão selecionado, gerar relatório de requisitos que permite o usuário gerar relatório dos requisitos do projeto que esta trabalhando e cadastrar requisito, que permite ao usuário cadastrar um requisito podendo consultar um exemplo padrão cadastrado na funcionalidade de cadastrar padrão de requisitos.

Os casos de uso cadastra padrão, listar padrões, listar exemplos de padrões, cadastra requisito e gera relatório de requisitos estão respectivamente detalhados nos quadros 3, 4, 5, 6 e 7.

Cadastrar padrão	
Pré-condição:	Usuário habilitado para o módulo usuário.
Cenário Principal:	1 – O usuário clica na opção padrões de requisitos. 2 – O usuário seleciona novo padrão. 3 – O sistema exibe o formulário para cadastrar um novo padrão de requisito. 4 – O usuário digita o tipo, título e descrição do padrão e seleciona em salvar. 5 – O sistema valida os dados preenchidos. 6 – O sistema apresenta tela de padrão adicionado com sucesso. 7 – O sistema volta à lista de padrões cadastrados.
Exceções:	Caso no passo 5 algum dado não seja informado: 5.1 – O sistema envia um alerta solicitando que todos os dados devem ser informados. 5.2 – Volta ao passo 3.
Pós-condição:	Padrão cadastrado com sucesso.

Quadro 3 – Caso de uso cadastrar padrão

Listar padrões	
Pré-condição:	Usuário habilitado para o módulo usuário.
Cenário Principal:	<p>1 – O usuário clica na opção padrões de requisitos.</p> <p>2 – O sistema apresenta a lista de padrões de requisitos.</p> <p>3 – O usuário opta por editar um padrão de requisito.</p> <p>4 – O sistema apresenta o formulário para edição do padrão apresentando os campos já preenchidos com os dados atuais.</p> <p>5 – O usuário altera os dados e salva o padrão de requisito.</p> <p>6 – O sistema valida os dados informados.</p> <p>7 – O sistema apresenta tela de padrão alterado com sucesso.</p>
Cenários Alternativos:	<p>Caso no passo 3 o usuário opte por excluir um padrão de requisito:</p> <p>3.1 – O sistema apresenta a mensagem “Tem certeza que deseja remover?”.</p> <p>3.2 – O usuário confirma a exclusão.</p> <p>3.3 – O sistema exclui o padrão de requisito.</p> <p>3.4 – Retorna ao passo 2.</p> <p>Caso no passo 3 o usuário opte por ver os exemplos do padrão de requisito:</p> <p>3.5 – O sistema apresenta a lista de exemplos do padrão de requisito selecionado.</p> <p>3.6 – Inclui o caso de uso listar exemplos de padrões.</p>
Exceções:	<p>Caso no passo 6 algum dado não seja informado:</p> <p>6.1 – O sistema envia um alerta solicitando que todos os dados devem ser informados.</p> <p>6.2 – Volta ao passo 4.</p>
Pós-condição:	Padrões visualizados com sucesso.

Quadro 4 – Caso de uso listar padrões

Listar exemplos de padrões	
Pré-condição:	Usuário habilitado para o módulo usuário.
Cenário Principal:	<p>1 – O usuário opta por adicionar um exemplo de padrão.</p> <p>2 – O sistema apresenta o formulário para cadastro exemplo de padrão.</p> <p>3 – O usuário digita os dados solicitados e clica em salvar.</p> <p>4 – O sistema valida os dados informados.</p> <p>5 – O sistema apresenta tela de exemplo de padrão adicionado com sucesso.</p>
Cenários Alternativos:	<p>Caso no passo 1 o usuário opte por editar um exemplo de padrão de requisito:</p> <p>1.1 – O sistema apresenta o formulário para edição do exemplo de padrão apresentando os campos já preenchidos com os dados atuais.</p> <p>1.2 – O usuário altera os dados e salva.</p> <p>1.3 – O sistema valida os dados informados.</p> <p>1.4 – Retorna ao passo 1.</p>

Listar exemplos de padrões	
	<p>Caso no passo 1 o usuário opte por excluir um exemplo de padrão de requisito:</p> <p>1.5 – O sistema apresenta a mensagem “Tem certeza que deseja remover?”.</p> <p>1.6 – O usuário confirma a exclusão.</p> <p>1.7 – O sistema exclui o exemplo de padrão de requisito.</p> <p>1.8 – Retorna ao passo 1.</p>
Exceções:	<p>Caso no passo 4 algum dado não seja informado:</p> <p>4.1 – O sistema envia um alerta solicitando que todos os dados devem ser informados.</p> <p>4.2 – Volta ao passo 2.</p> <p>Caso no passo 1.3 algum dado não seja informado:</p> <p>1.3.1 – O sistema envia um alerta solicitando que todos os dados devem ser informados.</p> <p>1.3.2 – Volta ao passo 1.1.</p>
Pós-condição:	Padrões visualizados com sucesso.

Quadro 5 – Caso de uso listar padrões

Cadastrar requisito	
Pré-condição:	Usuário habilitado para o módulo usuário.
Cenário Principal:	<p>1 – O usuário clica na opção requisitos.</p> <p>2 – O sistema exibe os tipos de requisitos que podem ser cadastrados, requisito funcional, requisito não funcional e regra de negócio.</p> <p>3 – O usuário seleciona clica no tipo de requisito que deseja cadastrar.</p> <p>4 – O sistema apresenta o formulário para cadastro requisito.</p> <p>5 – O usuário digita os dados solicitados e clica em salvar</p> <p>6 – O Sistema apresenta tela de requisito adicionado com sucesso.</p>
Cenários Alternativos:	<p>Caso no passo 4 o usuário opte por utilizar um padrão de requisito:</p> <p>4.1 – O sistema apresenta a lista de padrões de requisitos.</p> <p>4.2 – O usuário seleciona o padrão que deseja utilizar.</p> <p>4.3 – O sistema apresenta as informações sobre o padrão de requisitos e a opção de utilizar o padrão.</p> <p>4.4 – O usuário seleciona a opção utilizar.</p> <p>4.5 – O sistema aplica o padrão de requisito.</p> <p>4.6 – Retorna ao passo 4.</p>
Exceções:	<p>Caso no passo 5 algum dado não seja informado:</p> <p>5.1 – O sistema envia um alerta solicitando que todos os dados devem ser informados.</p> <p>5.2 – Volta ao passo 4.</p>
Pós-condição:	Requisito cadastrado com sucesso.

Quadro 6 – Caso de uso cadastrar requisito

Gerar relatório de requisitos	
Pré-condição:	O usuário estar acessando o sistema com o módulo de usuário.
Cenário Principal:	1 – O usuário clica na opção relatórios. 2 – O sistema exibe os tipos de relatórios: glossário, relatório de requisitos, relatório funcional e relatório simplificado que podem ser gerados. 3 – O usuário clica em relatório de requisitos. 4 – O sistema apresenta as opções para geração do relatório: PDF, <i>HyperText Markup Language</i> (HTML) e documento de texto. 5 – O usuário seleciona visualizar em PDF. 6 – O sistema apresenta o relatório em formato PDF.
Pós-condição:	Relatório de requisitos gerado com sucesso.

Quadro 7 – Caso de uso gerar relatório de requisitos

3.2.2 Diagrama de classes

Para facilitar o entendimento da estrutura da ferramenta foi construído um diagrama de classes com seis domínios:

- a) *view*: responsável por apresentar as telas ao usuário de acordo com os arquivos HTML;
- b) HTML: responsável por gerenciar as informações de cabeçalho, rodapé e menu em HTML de cada módulo de acesso como *master*, administrador e usuário;
- c) sistema: abrange classes padrões independentes das regra de negócio do sistema, como avisos, exceções, formatação de dados, *templates*, geração de PDF;
- d) conexão com o banco de dados: responsável pela conexão e comunicação com o SGBD;
- e) requisitos: controla as regras de negócio da aplicação em relação a gerência de requisitos, usuários e demais funcionalidades;
- f) padrões: agrega as classes de negócio responsáveis sobre os padrões de requisitos da ferramenta.

A seguir são detalhadas as classes de todos os domínios previamente citados, apresentando a responsabilidade de cada classe e os diagramas, agrupados e apresentados por domínios. A figura 8 apresenta o diagrama de classes do domínio *view*.

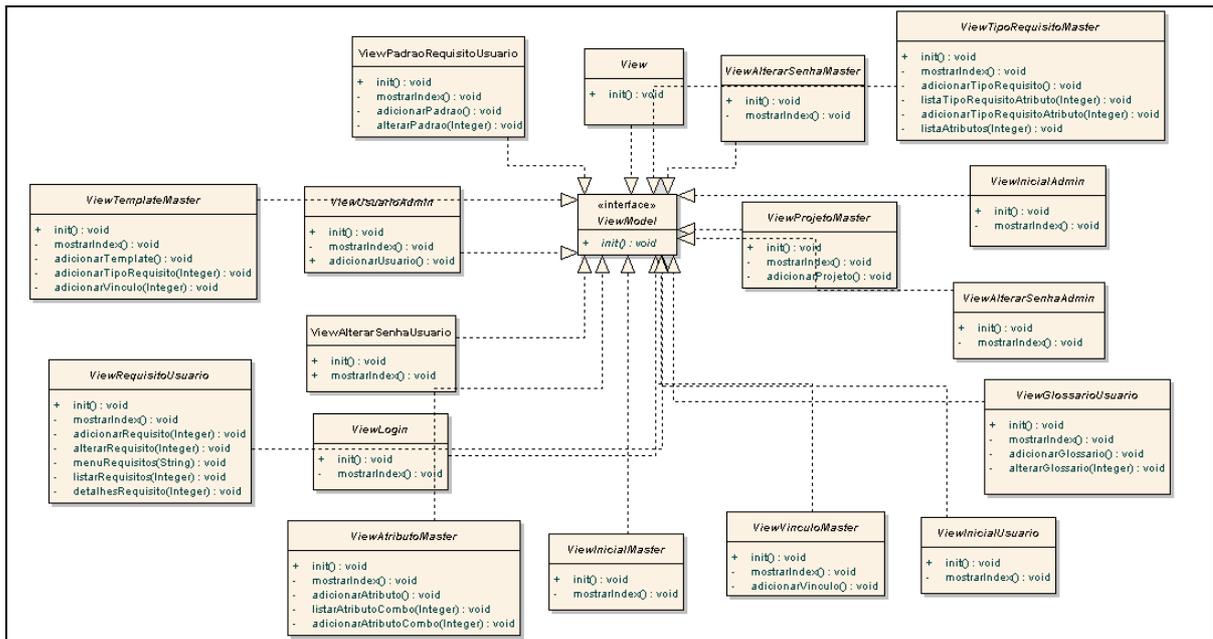


Figura 8 – Diagrama de classes do domínio *view*

O domínio *view* abrange as classes descritas a seguir:

- a) *ViewModel*: interface utilizada por todas as classes do domínio *view*. A interface *ViewModel* possui a assinatura do método *init()*. Este método é responsável por ler a *Uniform Resource Locator* (URL) informada e fazer a chamada do método da *view* necessária;
- b) *View*: primeira tela acessada pelo usuário do sistema, responsável por redirecionar para a *view* específica da área que o usuário deseja acessar de acordo com o endereço na URL;
- c) *ViewPadraoRequisitoUsuario*: monta as telas da área de padrões de requisito do módulo usuário;
- d) *ViewAlterarSenhaMaster*: apresenta as telas para alterar senha do módulo *master*;
- e) *ViewTipoRequisitoMaster*: responsável por gerenciar as telas dos tipos de requisito do módulo *master*;
- f) *ViewTemplateMaster*: constrói as telas da área de *template* do módulo *master*;
- g) *ViewUsuarioAdmin*: gerencia as telas da criação de usuário do módulo administrativo;
- h) *ViewProjetoMaster*: apresenta a área de projeto do módulo *master*;
- i) *ViewInicialAdmin*: tela inicial do módulo *master*;
- j) *ViewAlterarSenhaAdmin*: apresenta as telas da área de alterar senha do módulo administrativo;

- k) ViewRequisitoUsuario: responsável pelas telas da área de requisitos do módulo de usuário;
- l) ViewAlterarSenhaUsuario: gerencia as telas de alteração de senha do módulo de usuário;
- m) ViewLogin: tela de login do sistema;
- n) ViewAtributoMaster: constrói as telas da área de atributos do sistema no módulo *master*;
- o) ViewInicialMaster: tela inicial do módulo *master*;
- p) ViewVinculoMaster: gerencia as telas da área de vínculo do módulo *master*;
- q) ViewInicialUsuario: tela inicial do módulo usuário;
- r) ViewGlossarioUsuario: apresenta as telas do glossário do módulo usuário;
- s) ViewVinculoRequisito: apresenta as telas dos vínculos entre os requisitos.

A figura 9 apresenta o diagrama de classes do domínio HTML.

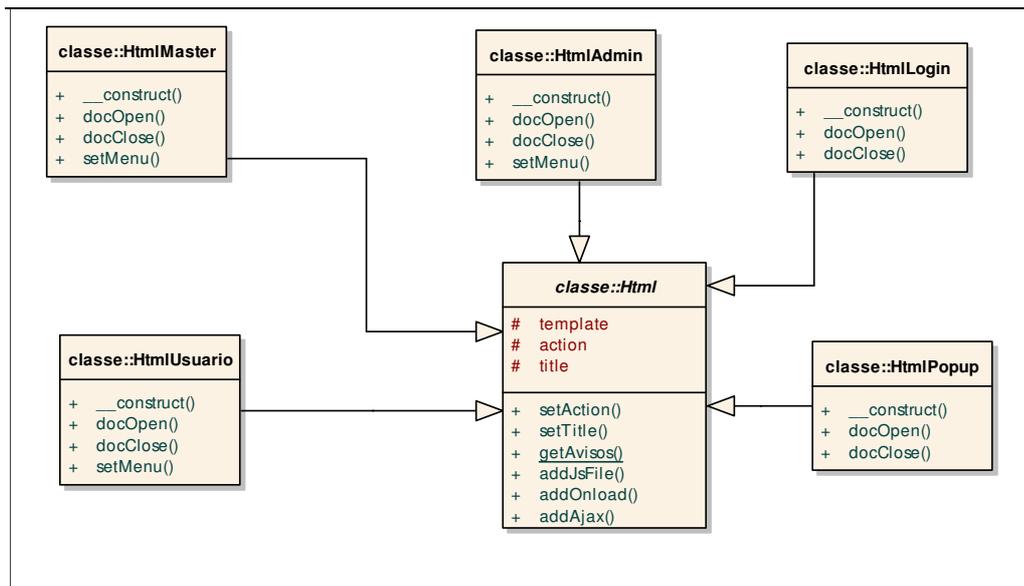


Figura 9 – Diagrama de classes do domínio HTML

O domínio HTML é chamado pelo domínio da *view* para montagem das telas padrões do layout como cabeçalho, menus e rodapés. Ele abrange as classes descritas a seguir:

- a) Html: classe com os métodos padrões para a criação do layout do sistema. Todas as classes Html devem herdar esta classe;
- b) HtmlLogin: responsável por montar o layout da página de login do sistema;
- c) HtmlUsuario: constrói o layout do módulo de usuário;
- d) HtmlMaster: monta o layout do módulo *master*;

- e) `HtmlAdmin`: monta o layout do módulo administrativo.
 f) `HtmlPoupup`: monta o layout de janelas *poupup* utilizadas no sistema.

A figura 10 apresenta o diagrama de classes domínio sistema descrevendo as classes internas independentes de regra de negócio.

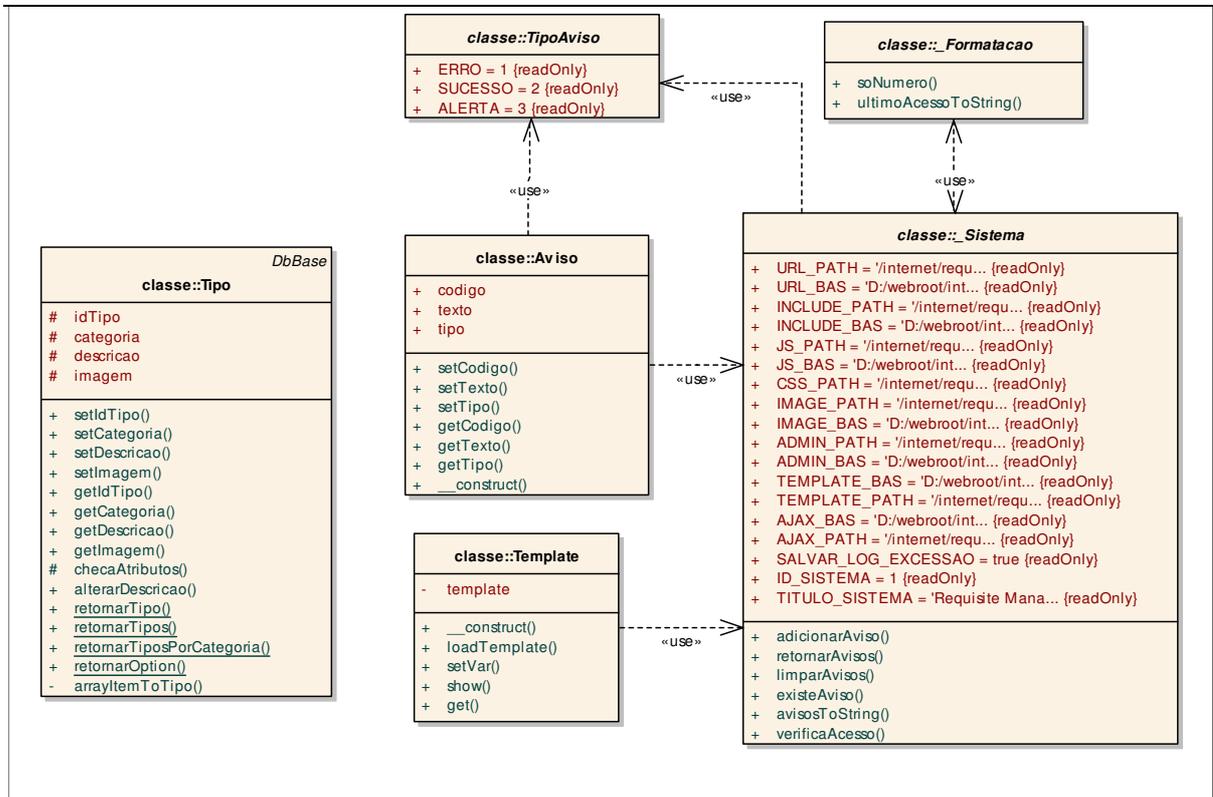


Figura 10 – Diagrama de classes do domínio sistema

O domínio sistema é um conjunto de classes padrões que gerenciam avisos, exceções, geração de PDF, formatação de dados do sistema, entre outros. As classes a seguir compõem o domínio de sistema:

- a) `Tipo`: classe responsável por gerenciar qualquer atributo que possa ser considerado um tipo, por exemplo, sexo (masculino ou feminino), tipo de usuário (administrador, cliente), entre outros;
 b) `Aviso`: gerencia os avisos de sucesso, alerta e erro do sistema;
 c) `TipoAviso`: possui constantes com os tipos de aviso do sistema (sucesso, alerta e erro);
 d) `Template`: adiciona o conteúdo do *Hypertext Preprocessor* (PHP) ao HTML;
 e) `_Formatacao`: agrega métodos de validação de formatos como emails, datas, entre outros formatos padrões utilizados em diversos sistemas;
 f) `_Sistema`: possui métodos e atributos padrões do sistema como constantes com endereços de acesso ao sistema e método de conferência de *login*;

A figura 11 apresenta o diagrama de classes do domínio conexão com o banco de dados.

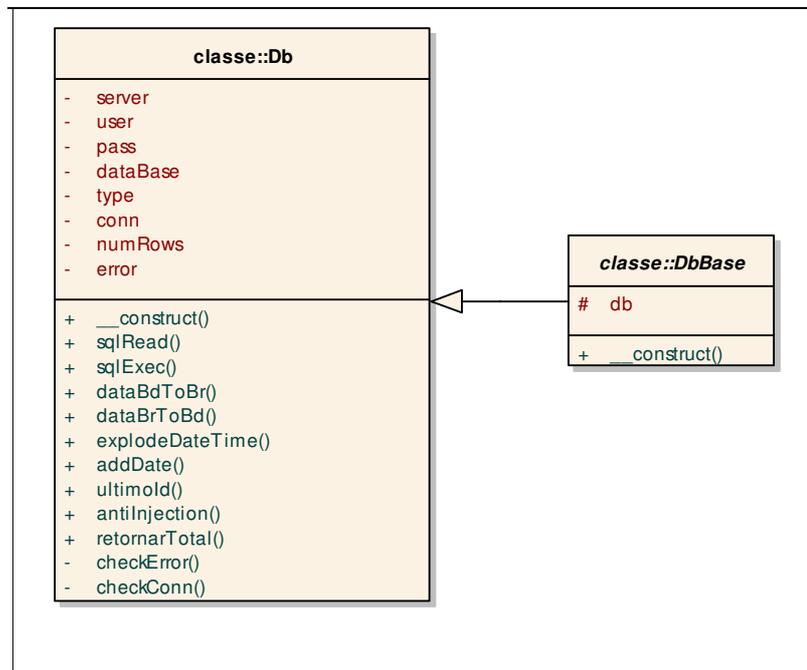


Figura 11 – Diagrama de classes do domínio conexão com o banco de dados

O domínio conexão com o banco de dados realiza a comunicação e chamadas para inserções, atualizações, exclusões e conexões com o banco de dados, possuindo as classes descritas a seguir:

- a) Db: possui todos os métodos e atributos necessários para a comunicação com o SGBD;
- b) DbBase: classe herdada pelas classes que necessitam de comunicação com o SGBD, esta classe possui apenas o construtor que inicializa a classe Db.

- b) `AtributoSelect`: opções de um atributo listados em uma caixa de seleção. Ex: `Prioridade` (alta, média e baixa);
- c) `Requisito`: requisitos criados pelo módulo usuário;
- d) `TipoRequisito`: tipos de requisitos definidos pelo módulo *master*;
- e) `ValorAtributo`: valor dos atributos definidos pelo módulo usuário;
- f) `Projeto`: projeto criado pelo módulo *master*;
- g) `TemplateProjeto`: *template* utilizado por um projeto;
- h) `Glossario`: palavras e suas definições de um projeto;
- i) `Usuario`: usuários do sistema;
- j) `Vinculo`: tipos de vínculos permitidos em um *template*;
- k) `VinculoRequisito`: os vínculos existentes entre os requisitos.

A figura 13 apresenta o diagrama de classes do domínio padrões responsáveis pelo gerenciamento dos padrões de requisitos e seus modelos.

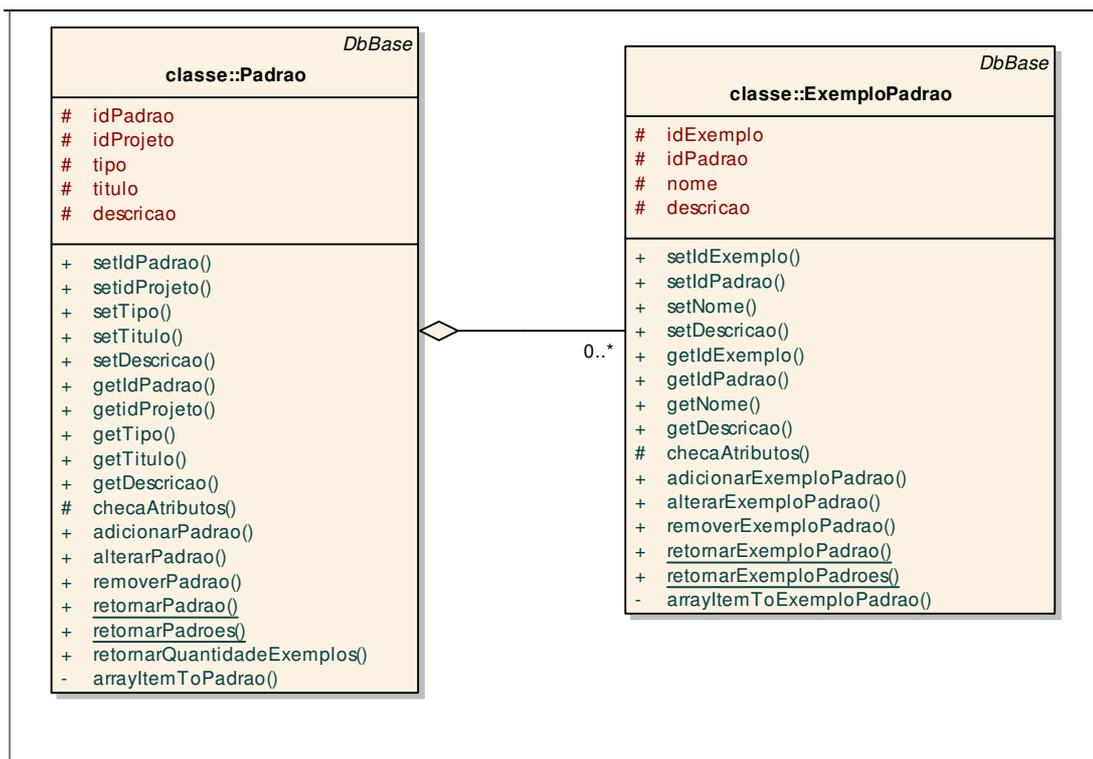


Figura 13 – Diagrama de classes do domínio padrões

O domínio de padrões compreende as classes:

- a) `Padrao`: gerencia as diversas funcionalidades dos padrões de requisitos;
- b) `ExemploPadrao`: representa os exemplos dos padrões de requisitos.

3.2.3 Diagramas de seqüência

A seguir são apresentados quatro diagramas de seqüência referentes as funcionalidades estendidas e novas funcionalidade implementadas na ferramenta. Na figura 14 é apresentado o diagrama de seqüência para cadastro de um padrão de requisito.

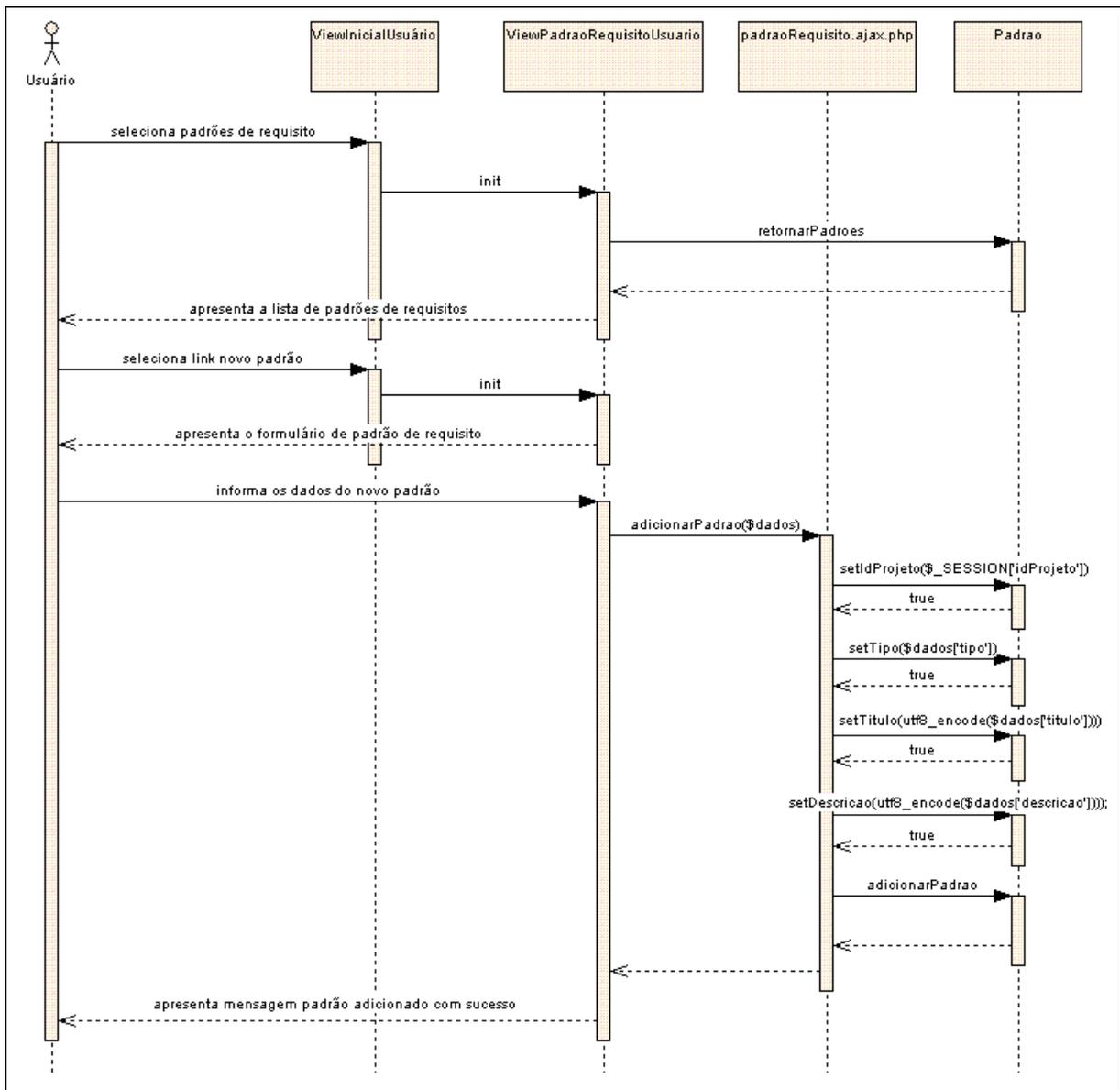


Figura 14 – Diagrama de seqüência para cadastro de um padrão de requisito

O diagrama de seqüência para o cadastro de um padrão de requisito (figura 14) mostra que ao usuário selecionar o link padrões de requisitos presente na ViewInicialUsuario, é redirecionado para ViewPadraoRequisito que chama o método `retonarPadroes` da classe `Padrao` para apresentar a lista de padrões de requisitos, assim como, a opção de novo padrão. Ao usuário selecionar a opção novo padrão é apresentado o formulário para cadastro de um

novo padrão de requisito, o usuário então informa os dados solicitados e seleciona a opção de salvar padrão de requisito. A `ViewPadraoRequisito` chama uma função *Asynchronous Javascript And XML* (AJAX) para adicionar o padrão de requisito no banco de dados, a função ajax efetua as chamadas para atribuir os valores da classe `Padrão` para efetuar a chamada do método `adicionarPadrao` que salva o novo padrão de requisito no banco de dados e apresenta a mensagem de sucesso ao usuário.

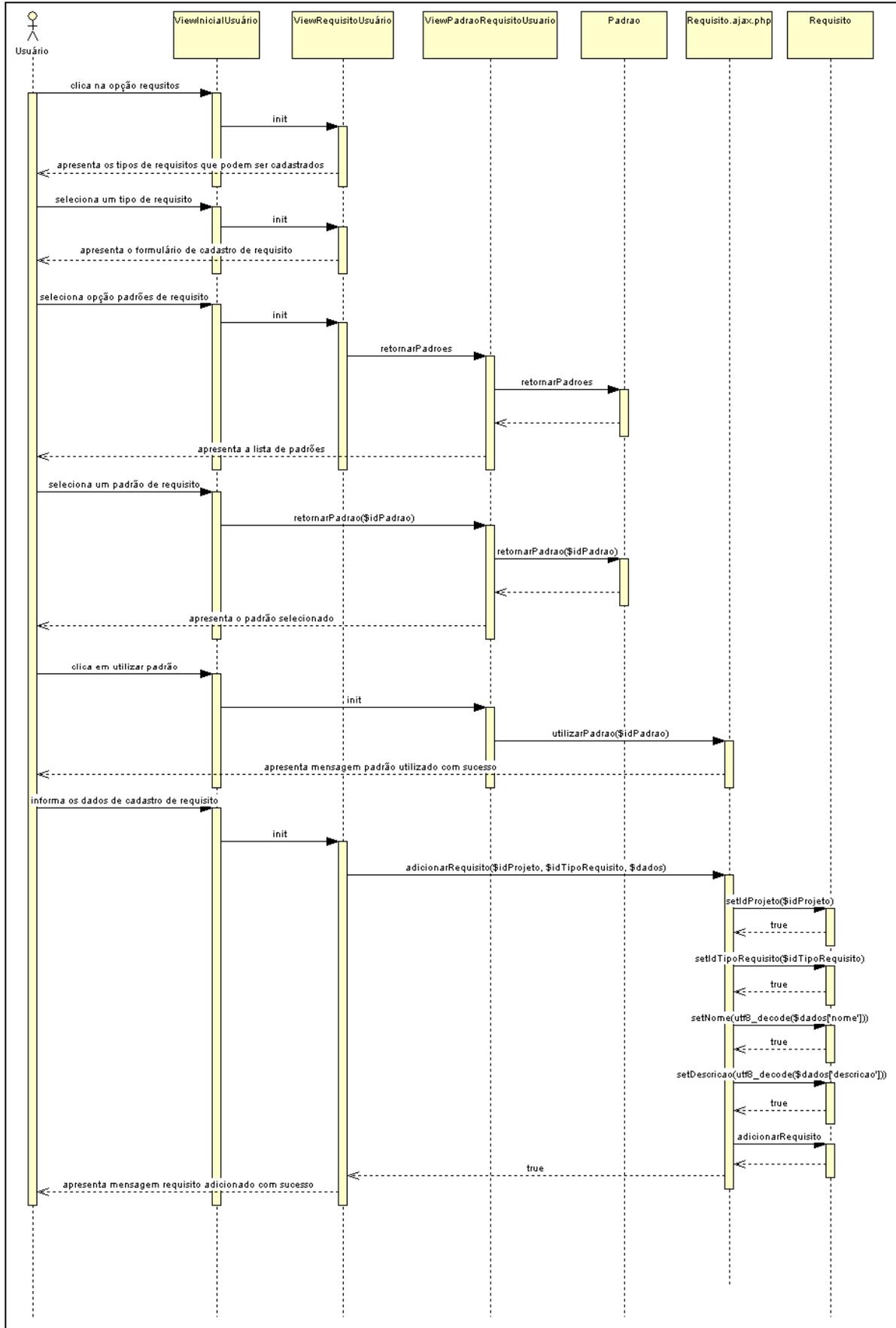


Figura 15 – Diagrama de seqüência para cadastro de um requisito

O diagrama de seqüência para o cadastro de um requisito (figura 15) mostra que ao

usuário selecionar o link requisitos presente na `ViewInicialUsuario`, é redirecionado para `ViewRequisitoUsuario` que apresenta os tipos de requisitos que podem ser cadastrados (requisito funcional, requisito não funcional, regra de negócio). Ao usuário selecionar um tipo de requisito é apresentado o formulário para cadastro de um novo requisito. O usuário então pode selecionar a opção padrões de requisitos. Assim é solicitada à classe `Padrao` que retorne todos os padrões de requisitos através do método `retornaPadroes`. Assim é apresentada a lista com os padrões de requisitos, o usuário então seleciona um padrão de requisitos para visualizar o padrão e seleciona a opção utilizar para utilizar esse padrão. A `ViewRequisitoUsuario` fará a chamada a função AJAX `utilizarPadrao` que colocará o padrão selecionado na requisição. Assim o usuário deve informar os dados do requisito e selecionar a opção para salvar o requisito, será realizada uma chamada da função AJAX `adicionarRequisito` que atribuirá os valores a classe `Requisito` e efetuará a chamada do método `adicionarRequisito` cadastrando o requisito no banco de dados e apresentando a mensagem de sucesso para esse processo.

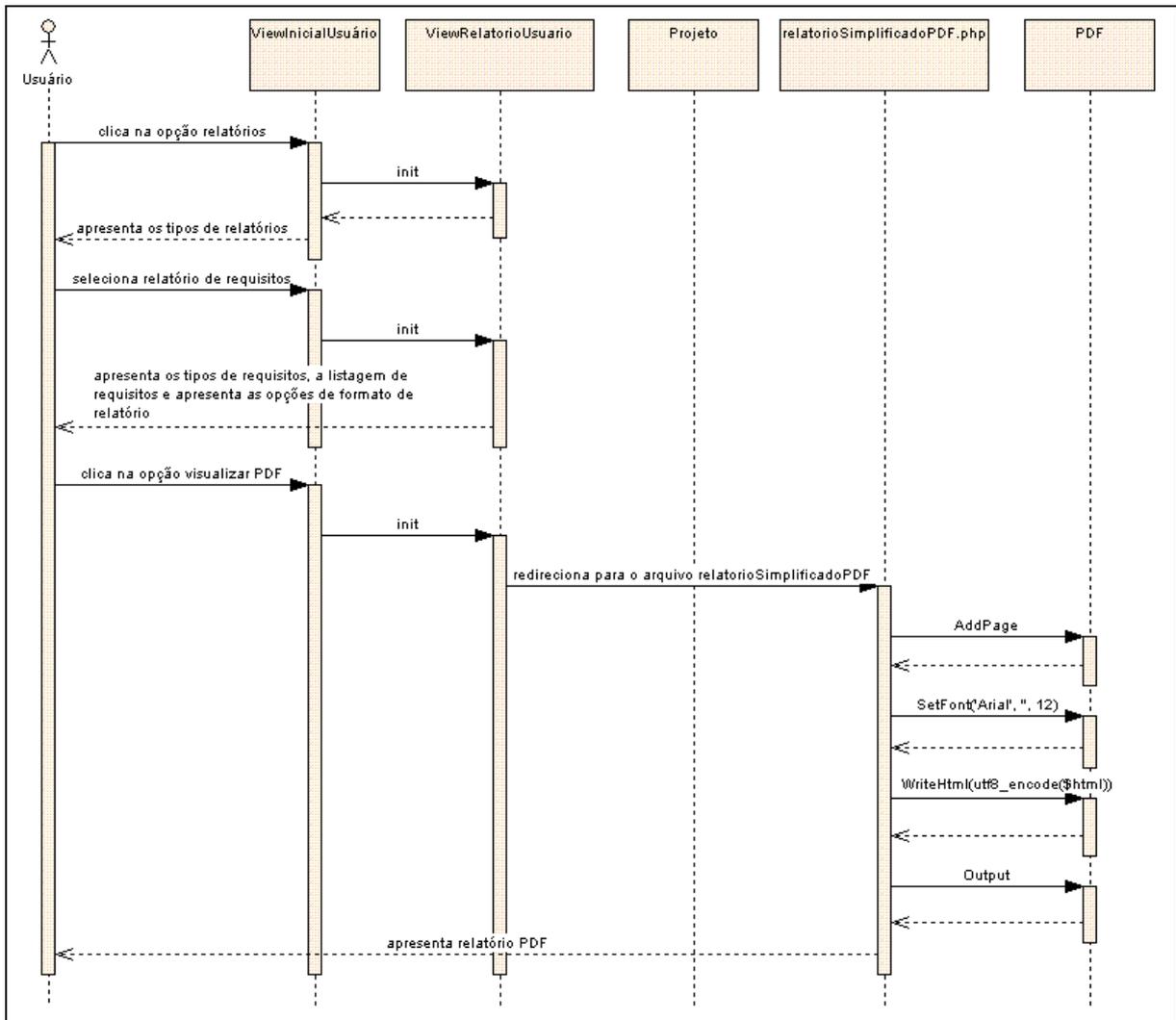


Figura 16 – Diagrama de seqüência para geração de um relatório de requisitos

O diagrama de seqüência para geração de um relatório de requisito (figura 16) mostra que ao usuário selecionar o link relatórios presente na `ViewInicialUsuario`, é redirecionado para `ViewPadraoRequisito` que apresenta os tipos de relatórios (glossário, relatório de requisitos, relatório funcional e relatório simplificado). Ao usuário selecionar o relatório de requisitos a `ViewPadraoRequisito` efetua uma chamada do método `retornarTipoRequisitos` da classe `Projeto` para apresentar os tipos de requisitos, os requisitos e as opções de formato para geração do relatório. O usuário então seleciona a opção visualizar em PDF, assim o usuário é redirecionado para o arquivo `relatorioSimplificadoPDF.php` onde é realizada a instância da classe `PDF` e chamada dos métodos `AddPage`, `SetFont`, `WriteHtml` e `Output` para geração do arquivo PDF. O resultado desse processo é a apresentação do relatório de requisitos em PDF.

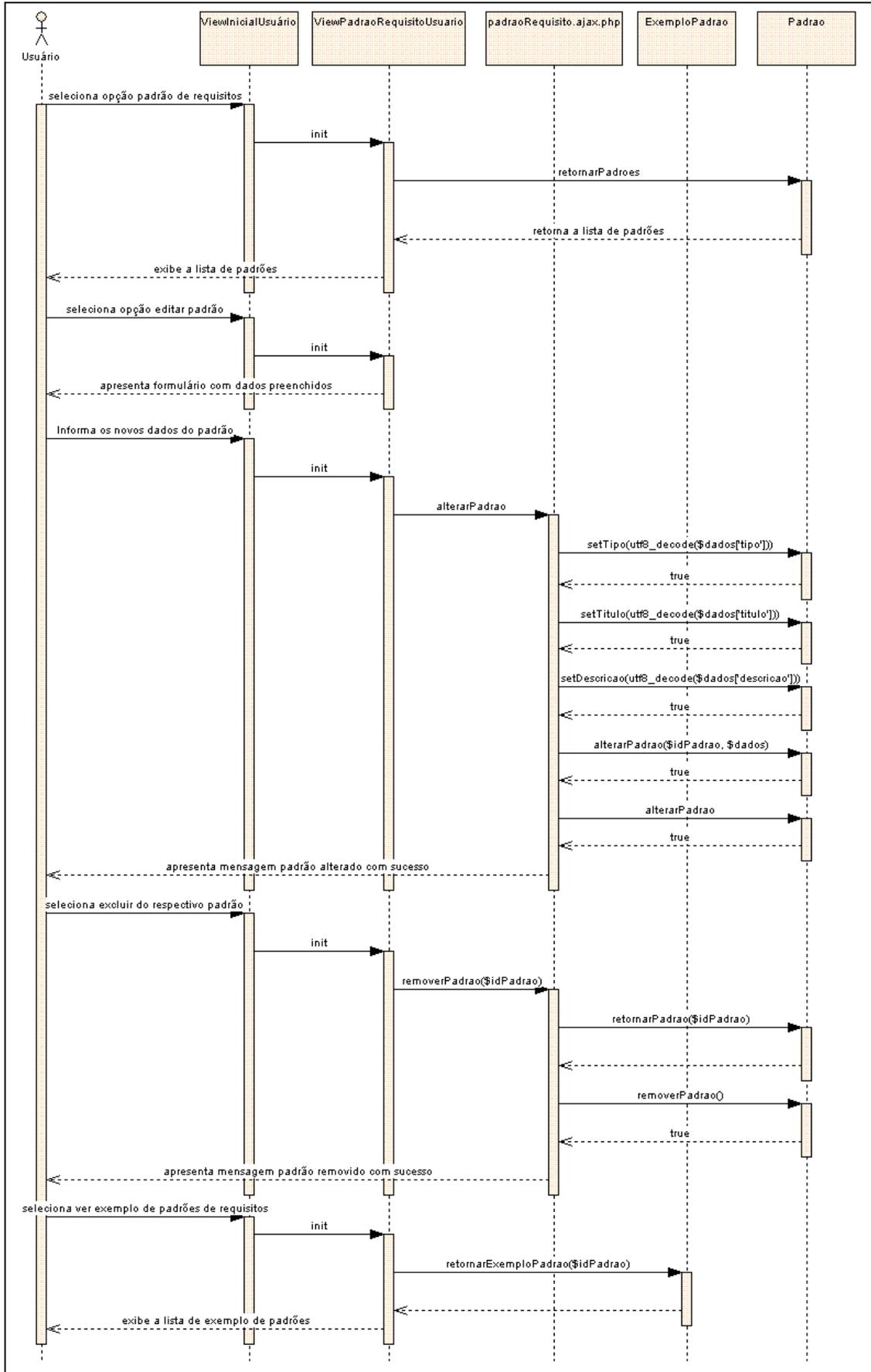


Figura 17 – Diagrama de seqüência para listar padrões de requisitos

O diagrama de seqüência para visualizar a lista de padrões de requisitos (figura 17) mostra que ao usuário selecionar o link padrões de requisitos presente na `ViewInicialUsuario`, é redirecionado para `ViewPadraoRequisitoUsuario` que chama o método `retonarPadroes` da classe `Padrao` para apresentar a lista de padrões de requisitos, assim como, as opções de editar padrão, excluir padrão e exemplo de padrão de requisito. Ao usuário selecionar a opção editar padrão é apresentado o formulário para edição do padrão de requisito, o usuário então informa os dados solicitados e seleciona a opção de alterar padrão de requisito. A `ViewPadraoRequisitoUsuário` chama uma função AJAX para alterar o padrão de requisito cadastrado, a função AJAX efetua as chamadas para atribuir os valores da classe `Padrao` para efetuar a chamada do método `alterarPadrao` que salva os novos valores do padrão de requisito no banco de dados e apresenta a mensagem de sucesso ao usuário. Caso usuário selecionar a opção excluir algum padrão de requisito a `ViewPadraoRequisitoUsuário` chama o método `retornarPadrao` e `removerPadrao` através do arquivo AJAX `padraoRequisito.ajax.php` removendo o padrão do banco de dados e apresentando a mensagem de removido com sucesso ao usuário. Caso usuário selecionar a opção ver exemplos de padrões de requisitos a `ViewPadraoRequisitoUsuário` efetua a chamada do método `retornarPadrao` e `retornarExemploPadroes` da `Padrao` para apresentar a lista de exemplos de padrões ao usuário.

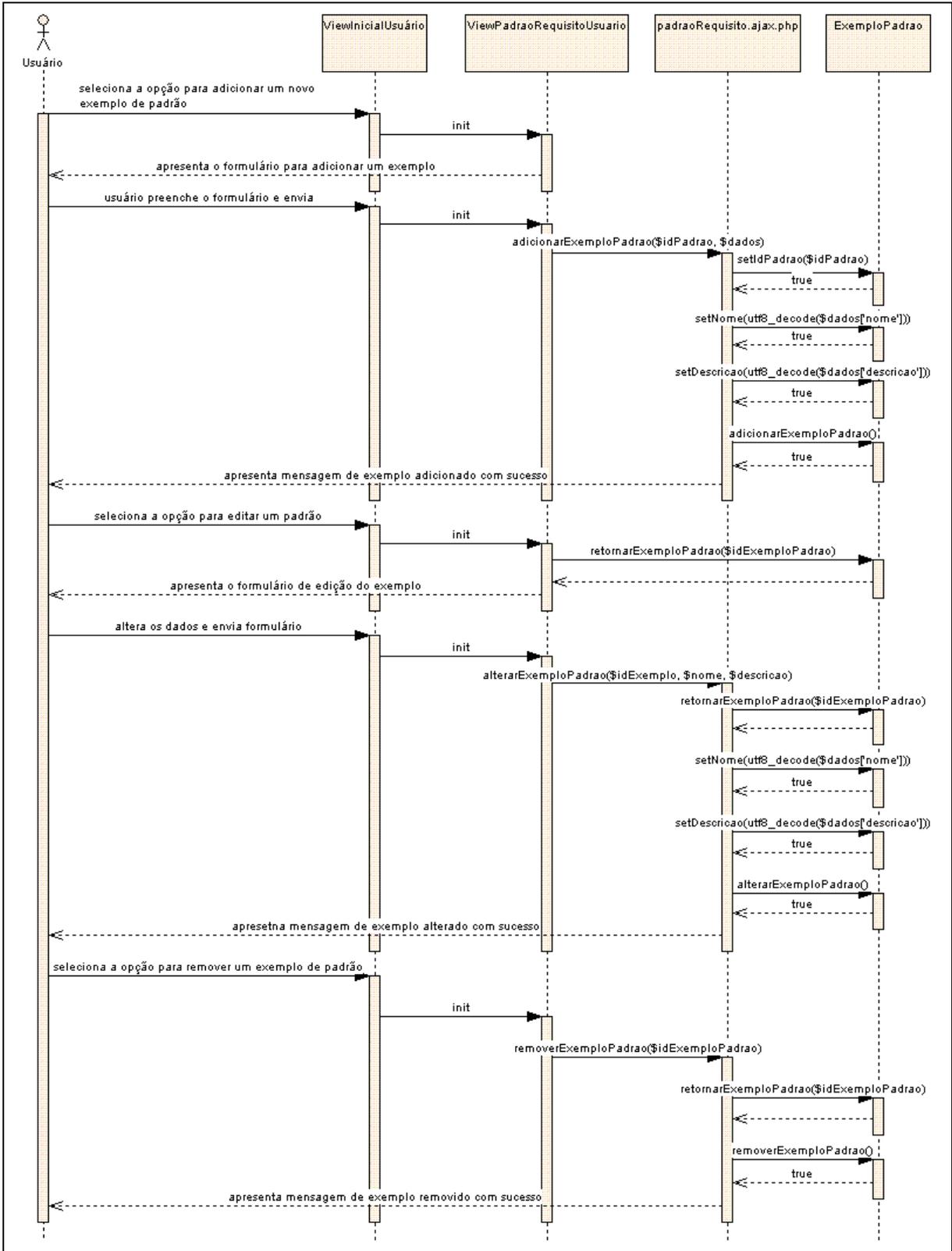


Figura 18 – Diagrama de seqüência para visualizar a lista de exemplo de padrões de requisitos

O diagrama de seqüência para visualizar a lista de exemplos de padrões de requisitos (figura 18) mostra que ao usuário selecionar o link adicionar novo exemplo presente na ViewInicialUsuario, é redirecionado para ViewPadraoRequisitoUsuario que apresenta o

formulário para adicionar um exemplo, o usuário preenche o formulário e o envia, então a `ViewPadraoRequisitoUsuario` chama a função AJAX `alterarExemploPadrao` presente no arquivo `padraoRequisito`, essa função atribui os valores a classe `ExemploPadrao` e chama o método `adicionarExemploPadrao` que efetua o cadastro do padrão no banco de dados. Ao usuário selecionar a opção editar exemplo do padrão é apresentado o formulário para edição do exemplo de padrão de requisito, o usuário então informa os dados solicitados e seleciona a opção de alterar exemplo padrão. A `ViewPadraoRequisitoUsuario` chama uma função AJAX para alterar o padrão de requisito cadastrado, a função AJAX efetua as chamadas para atribuir os valores da classe `ExemploPadrao` para efetuar a chamada do método `alterarExemploPadrao` que salva os novos valores do exemplo no banco de dados e apresenta a mensagem de sucesso ao usuário. Caso usuário selecionar a opção excluir exemplo a `ViewPadraoRequisitoUsuario` chama o método `retornarPadrao` e `removerPadrao` através do arquivo AJAX `padraoRequisito.ajax.php` removendo o exemplo do banco de dados e apresentando a mensagem de removido com sucesso ao usuário.

3.2.4 Diagrama entidade relacionamento

O DER foi aperfeiçoado fornecendo uma solução mais completa e detalhada para a ferramenta. Apresentando o relacionamento entre as tabelas, suas chaves primárias, chaves estrangeiras como também o tipo de cada campo da tabela. Todas as tabelas possuem descrição como cada um de seus campos. Foi utilizado o software EA para modelagem do DER que é apresentado na figura 19.

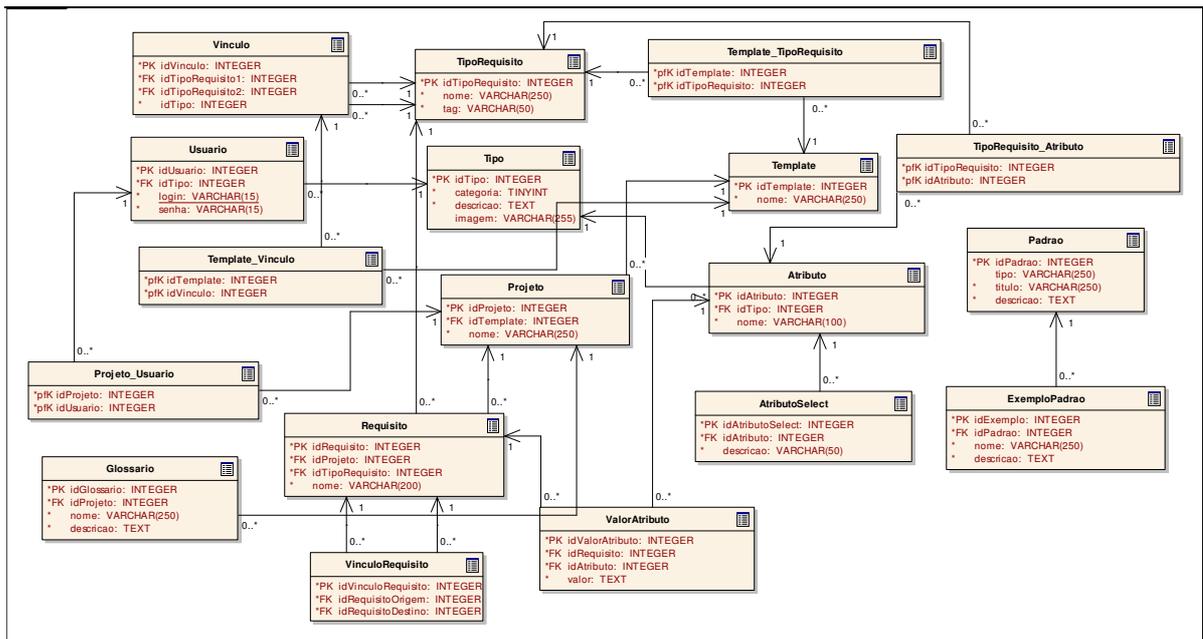


Figura 19 – Modelo de entidade e relacionamento

A seguir é apresentada uma seqüência de quadros (8 a 24) que representam as tabelas do banco de dados, detalhados com nome dos atributos, tipo e uma descrição.

Nome	Tipo	Descrição
idAtributo	INTEGER	Identificador do atributo
idTipo	INTEGER	Identificador do tipo do atributo
nome	VARCHAR	Nome do atributo.

Quadro 8 – Tabela que armazena os atributos de um tipo de requisito

Nome	Tipo	Descrição
idAtributoSelect	INTEGER	Identificador da opção
idAtributo	INTEGER	Identificador do atributo
descricao	VARCHAR	Descrição da opção

Quadro 9 – Tabela que armazena as opções dos atributos de múltipla escolha

Nome	Tipo	Descrição
idExemplo	INTEGER	Identificador do exemplo
idPadrao	INTEGER	Identificador do padrão
nome	VARCHAR	Nome do exemplo
descricao	TEXT	Descrição do exemplo

Quadro 10 – Tabela que armazena os exemplos dos padrões de requisitos

Nome	Tipo	Descrição
idGlossario	INTEGER	Identificador do item glossário
idProjeto	INTEGER	Identificador do projeto
nome	VARCHAR	Nome do item do glossário
descricao	TEXT	Descrição do item do glossário

Quadro 11 – Tabela que armazena o glossário de um projeto

Nome	Tipo	Descrição
idPadrao	INTEGER	Identificador do padrão
tipo	VARCHAR	Tipo de padrão
titulo	VARCHAR	Titulo do padrão
descricao	TEXT	Descrição do padrão

Quadro 12 – Tabela que armazena os padrões de requisitos

Nome	Tipo	Descrição
idProjeto	INTEGER	Identificador do projeto
idTemplate	INTEGER	Identificador do template
nome	VARCHAR	Nome do projeto

Quadro 13 – Tabela que armazena os projetos

Nome	Tipo	Descrição
idProjeto	INTEGER	Identificador do projeto
idUsuario	INTEGER	Identificador do usuário

Quadro 14 – Tabela que armazena os usuários de um projeto

Nome	Tipo	Descrição
idRequisito	INTEGER	Identificador do requisito
idProjeto	INTEGER	Identificador do projeto
idTipoRequisito	INTEGER	Identificador do tipo de requisito
nome	VARCHAR	Nome do requisito

Quadro 15 – Tabela que armazena os requisitos de um projeto

Nome	Tipo	Descrição
idTemplate	INTEGER	Identificador do template
nome	VARCHAR	Nome do template

Quadro 16 – Tabela que armazena os templates do sistema

Nome	Tipo	Descrição
idTemplate	INTEGER	Identificador do template
idTipoRequisito	INTEGER	Identificador do tipo de requisito

Quadro 17 – Tabela que armazena os tipos de requisitos de um template

Nome	Tipo	Descrição
idTemplate	INTEGER	Identificador do template
idVinculo	INTEGER	Identificador do vinculo

Quadro 18 – Tabela que armazena os vínculos do template

Nome	Tipo	Descrição
idTipo	INTEGER	Identificador do tipo
categoria	TINYINT	Constante que determina de qual categoria pertence este tipo. Ex: 01 = Telefone, 02 = Sexo
descricao	TEXT	Descrição do tipo
imagem	VARCHAR	Se existir imagem representando o tipo

Quadro 19 – Tabela os tipos de dados existentes no sistema

Nome	Tipo	Descrição
idTipoRequisito	INTEGER	Identificador do tipo de requisito
nome	VARCHAR	Nome do tipo de requisito
tag	VARCHAR	Tag que representa o tipo de requisito

Quadro 20 – Tabela que armazena os tipos de requisitos

Nome	Tipo	Descrição
idTipoRequisito	INTEGER	Identificador do tipo de requisito
idAtributo	INTEGER	Identificador do atributo

Quadro 21 – Tabela que armazena os atributos de um tipo de requisito

Nome	Tipo	Descrição
idUsuario	INTEGER	Identificador do usuário
idTipo	INTEGER	Identificador do tipo de usuário
login	VARCHAR	Login do usuário
senha	VARCHAR	Senha do usuário

Quadro 22 – Tabela que armazena os usuários do sistema

Nome	Tipo	Descrição
idValorAtributo	INTEGER	Identificador do valor
idRequisito	INTEGER	Identificador do requisito
idAtributo	INTEGER	Identificador do atributo
valor	TEXT	Valor do atributo

Quadro 23 – Tabela que armazena o valor dos atributos de um requisito

Nome	Tipo	Descrição
idVinculo	INTEGER	Identificador do vinculo
idTipoRequisito1	INTEGER	Identificador do tipo de requisito 1
idTipoRequisito2	INTEGER	Identificador do tipo de requisito 2
idTipo	INTEGER	Identificador do tipo de vínculo

Quadro 24 – Tabela que armazena os vínculos dos templates

Nome	Tipo	Descrição
idVinculoRequisito	INTEGER	Identificador do vinculo
idRequisitoOrigem	INTEGER	Identificador do requisito origem
idRequisitoDestino	INTEGER	Identificador do requisito destino

Quadro 25 – Tabela que armazena os vínculos dos requisitos

3.3 IMPLEMENTAÇÃO

Com base na especificação, na seção de implementação é apresentado o desenvolvimento da ferramenta, detalhando trechos de código fonte mais relevantes, operacionalidade da ferramenta, técnicas e tecnologias utilizadas.

3.3.1 Tecnologias e ferramentas utilizadas

Foram utilizadas as seguintes tecnologias e ferramentas na etapa de implementação e testes do presente trabalho:

- a) zend development environment: plataforma de desenvolvimento que oferece suporte a PHP, utilizado para codificar a ferramenta em PHP a figura 20 mostra a interface do editor;

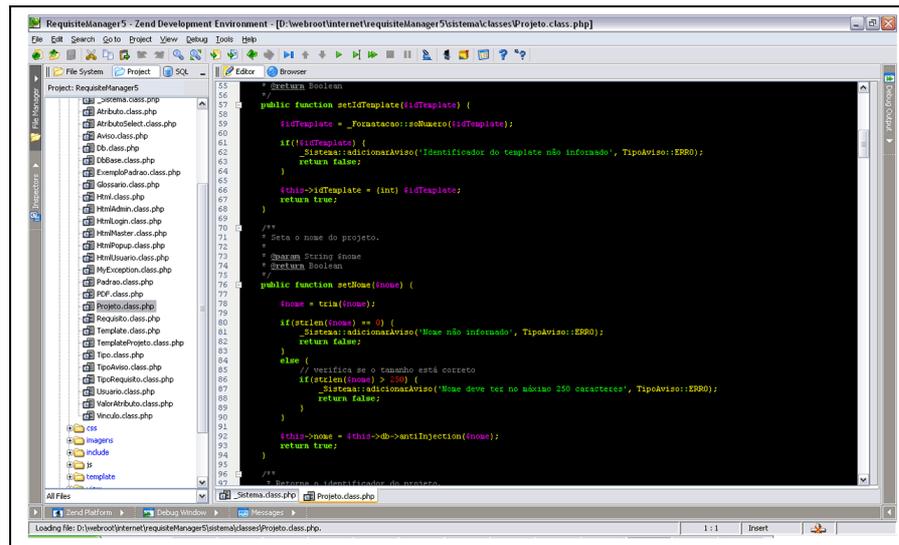


Figura 20 – Ambiente de desenvolvimento ZEND

- b) linguagem PHP: linguagem de programação interpretada conhecida em desenvolvimento web;
- c) MySQL: SGBD gratuito utilizado para gerenciar o banco de dados;
- d) javascript: linguagem de programação executada no cliente, para interagir com o browser do usuário;
- e) xajax: biblioteca de código aberto em PHP para construções de aplicativos web em AJAX² como mostra o quadro 26, onde a função em xajax mostra o processo para adicionar um usuário no sistema, recebendo um array com os dados de cadastro;

² Uso metodológico de tecnologias de comunicação assíncrona com um servidor web.

```

1. function adicionarUsuario($dados) {
2.     $objResponse = new xajaxResponse();

3.     _Sistema::limparAvisos();
4.     $usuario = new Usuario();
5.     $usuario->setLogin($dados['login']);
6.     $usuario->setSenha($dados['senha']);
7.     $usuario->setIdTipo(3);
8.
9.     if(!_Sistema::existeAviso()) {
10.         // verifica se as senha $conferem
11.         if($usuario->getSenha() != $dados['confirmarSenha']) {
12.
13.             $objResponse->alert('As senhas não são iguais.');
```

Quadro 26 – Utilização da biblioteca xajax

- f) FPDF: biblioteca gratuita destinada a geração de arquivos em formato PDF, o quadro 27 exemplifica a utilização da classe da biblioteca para gerar um arquivo PDF a partir de um arquivo HTML;

```

1. $pdf=new PDF();
2. //First page
3. $pdf->AddPage();
4. $pdf->SetFont('Arial','',12);
5. $pdf->WriteHTML(utf8_decode($html));
6. $pdf->Output();
```

Quadro 27 – Utilização da biblioteca FPDF

- g) SQLYog: software que possibilita a edição de banco de dados MySQL, utilizado

para gerenciamento dos dados armazenados no Banco de Dados, a figura 21 ilustra a interface do software;

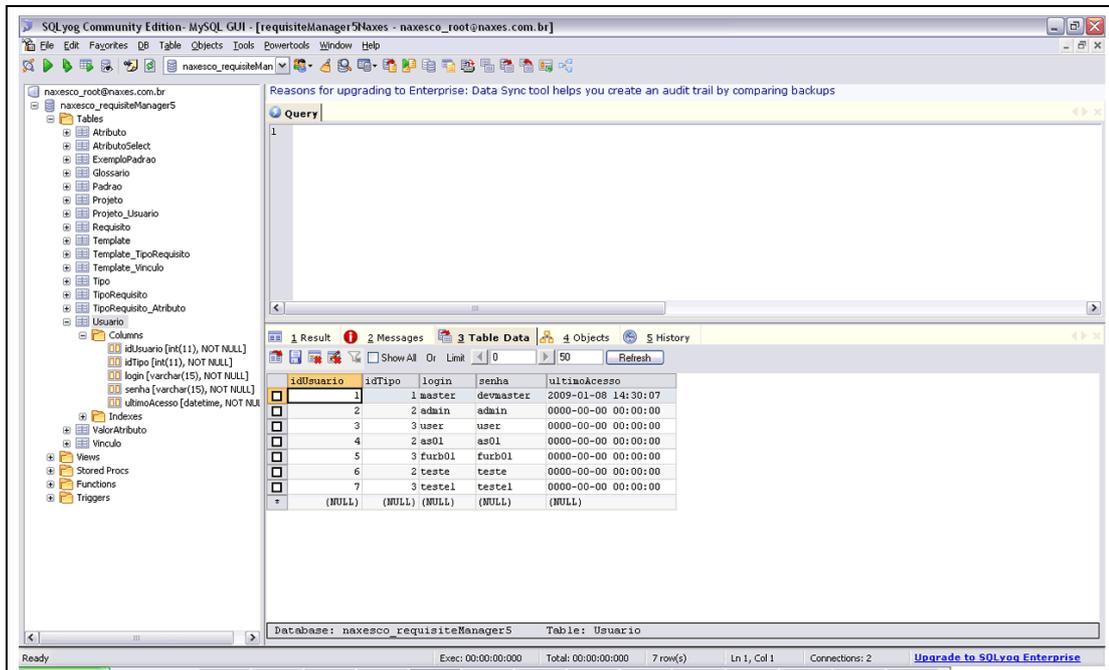


Figura 21 – Ambiente de gerência de dados do SQLYog

h) apache: servidor web livre compatível com o protocolo HTTP versão 1.1, utilizado para executar o sistema *Requisite Manager*.

3.3.2 Implementação da funcionalidade de adicionar padrões

A funcionalidade de adicionar um novo padrão de requisito ao banco de dados está detalhada em nível de código fonte no quadro 28. Essa funcionalidade utiliza a biblioteca *xajax*.

```

1. $objResponse = new xajaxResponse();
2.
3. $padrao = new Padrao();
4.
5. $padrao->setIdProjeto($_SESSION['idProjeto']);
6. $padrao->setTipo($dados['tipo']);
7. $padrao->setTitulo(utf8_encode($dados['titulo']));
8. $padrao->setDescricao(utf8_encode($dados['descricao']));
9.
10. If(!$padrao->adicionarPadrao()) {
11.     $listaAvisos = _Sistema::retornarAvisos(TipoAviso::ERRO);
12.     $objResponse->
        alert(utf8_encode(_Sistema::avisosToString($listaAvisos)));
13. } else {
14.     $objResponse->alert(utf8_encode('Padrão adicionado com
        sucesso'));
15.     $objResponse->script('self.location=""._Sistema::URL_PATH.
        'usuario/padraoRequisito"');
16. }

```

Quadro 28 – Chamada de adicionar padrão requisito no banco dados

A linha 1 cria uma instância do objeto que realiza a comunicação com o xajax. Na linha 3 está sendo instanciada a classe `Padrao`. Entre as linhas 5 e 8 são atribuído os valores para os atributos da classe `Padrao` com os valores informados pelo usuário através do formulário HTML. Na sequência é realizada a chamada do método `adicionarPadrao` que retornará `true` caso o padrão tenha sido acionado e `false` caso ocorra algum problema. Em caso de retorno `false` o próprio método `adicionarPadrao` se encarrega de adicionar na classe abstrata `_Sistema` as mensagens com os avisos de exceções que aconteceram durante o processo de adicionar o padrão de requisito. O método `alert` do xajax nas linhas 12 e 14 tem o propósito de exibir a mensagem definida no método `adicionarPadrao` para o usuário. Caso o retorno tenha sido `true` o sistema apresenta a mensagem de sucesso para o usuário e executa um script javascript através do xajax para redirecionar a página para a lista de padrões de requisitos.

3.3.3 Implementação da funcionalidade de listar padrões e funcionamento do template

São apresentados os trechos de códigos fontes utilizados de um método da classe abstrata chamada `ViewPadraoRequisitoUsuario` para mostrar a tela com a lista de padrões de requisitos.

O quadro 29 detalha um trecho do código fonte em HTML presente no arquivo “`padraoRequisito.tpl.html`”. Este HTML é utilizado pela classe `ViewPadraoRequisitoUsuario` para construir a tela de visualização dos padrões de

requisitos ao usuário. Através da utilização da classe *template* é possível separar totalmente o código PHP do código HTML como é mostrado no quadro 29 e 30.

```

1.  *=>listaPadroes
2.  <table>
3.    <tr>
4.      <td>NOME</td>
5.      <td>&nbsp;</td>
6.      <td>&nbsp;</td>
7.    </tr>
8.
9.    <!--listaPadroes-->
10.
11. </table>
12. <=*
13.
14. *=>linhaPadrao
15. <tr>
16.   <td>%s</td>
17.   <td><a href="padraoRequisito/alterar/index.php?id=%d">
18.     </a>
20.   </td>
21.   <td><a href="#" onclick="acaoRemoverAjax('xajax_removePadrao(%d)')">
22.     
24.   </a>
25.   </td>
26. </tr>
27. <=*

```

Quadro 29 – Arquivo HTML lista de padrões de requisitos

O quadro 29 mostra o HTML separado em dois blocos, utilizando o delimitador `*=>` seguido do nome do bloco é feita a abertura do bloco como pode ser visto nas linhas 1 e 14, o fechamento do bloco é delimitado por `<=*` como é representado nas linhas 12 e 25. O bloco da linha 14 chamado de `linhaPadrao` é utilizado para cada padrão a ser listado, injetando os valores das variáveis no bloco em tempo de execução, quando o bloco `linhaPadrao` for preenchido com todos os padrões de requisitos a serem listados, este é injetado no bloco `listaPadroes` ao qual é exibido como resultado desta operação.

```

1. $html = new HtmlUsuario();
2. $tpl = new Template(_Sistema::TEMPLATE_BAS .
   'usuario/padraoRequisito.tpl.html');
3. $tpl->setVar('IMAGE_PATH', _Sistema::IMAGE_PATH);
4. $tpl->setVar('URL_PATH', _Sistema::URL_PATH);
5.
6. $html->setMenu($tpl->getMenu('menu'));
7. $html->docOpen();
8. $listaPadroes = Padrao::retornarPadroes();
9. $total = count($listaPadroes);
10. $str = '';
11. $linha = $tpl->get('linhaPadrao');
12. for($i = 0; $i < $total; $i++) {
13.     $padrao = $listaPadroes[$i];
14.     $str .= sprintf($linha, $padrao->getTipo(),
15.                    $padrao->getIdPadrao(),
16.                    $padrao->getIdPadrao());
17. }
18. $tpl->setVar('listaPadroes', $str);
19. $tpl->show('listaPadroes');
20. $html->docClose();

```

Quadro 30 – Arquivo PHP para manipulação do HTML

O quadro 30 apresenta o método da classe `ViewPadraoRequisitoUsuario` que manipula o código HTML apresentado no quadro 29 para construir a lista de padrões de requisitos de acordo com os dados armazenados no banco de dados. A linha 1 instancia a classe `HtmlUsuario` utilizada para montar o cabeçalho e o rodapé da página através dos métodos `docOpen()` (cabeçalho) e `docClose()` (rodapé). A linha 2 instancia a classe `Template` que estará gerenciando o conteúdo HTML. As linhas 3 e 4 estão injetando valores nos blocos HTML comentados no quadro 29, utilizando o método `setVar()` da classe `Template`. Na linha 6 o menu está sendo adicionado ao layout da página, o HTML deste menu também é manipulado através de um arquivo *template*. Na linha 8 esta sendo retornado a lista de padrões de requisitos cadastrados no banco. Na linha 9 é adquirido o total de padrões retornados com intuito de ser utilizado na lógica da rotina. A linha 11 está atribuindo o bloco HTML chamado `linhaPadrao` comentado no quadro 29, a uma variável PHP. Na linha 12 para cada padrão de requisito retornado está injetando seus valores no bloco `linhaPadrao` armazenado em uma variável PHP. O método `sprintf` na linha 14 se encarrega de substituir do HTML todos os atributos com o símbolo `%` pelo valor passado por parâmetro. O `%d` representa que este valor deve ser substituído por um valor inteiro, no exemplo esta sendo substituído pelo identificador do tipo do padrão e pelo identificador do padrão. Após a `string` ter sido montada ela é adicionada ao HTML (linha 18) e a tela é mostrada ao usuário (linha 19).

O quadro 31 exhibe um trecho de código fonte da ferramenta de *Requisite Manager*, desenvolvida pelo acadêmico Luciano Marquardt no TCC (MARQUARDT, 2004), onde não

é feita a distinção da camada de interface (código fonte HTML) e da camada lógica (código fonte PHP), conforme comentado esse aspecto foi alterado a fim de aumenta legibilidade de ferramenta.

```

1. <tr>
2.   <td width="20"></td>
3.   <td align="right">Projeto:</td>
4.   <td width="10">&nbsp;</td>
5.   <td><select size="1" name="codigoProjeto">
6.     <? while ($linha = mysql_fetch_row($rs)) { ?>
7.       <option value="<? echo $linha[0]; ?>"><? echo $linha[1];?>
8.     <? } ?>
9.   </select></td>
10.  <td width="20"></td>
1. </tr>

```

Quadro 31 – Arquivo PHP para manipulação do HTML

3.3.4 Arquivo de documentação do código fonte

A partir do código fonte documentado, foi gerado um arquivo de extensão *Microsoft Compiled HTML Help* (CHM) para plataformas windows, para auxiliar desenvolvedores que virem a estender, ampliar ou aperfeiçoar a ferramenta, permitindo realizar consultas de todas as funcionalidades que a ferramenta possui. A figura 22 ilustra a interface do arquivo CHM gerado a partir da documentação da ferramenta.

Como pode ser visto na figura 22, ao lado esquerdo da interface encontra-se uma lista com todas as funções que a ferramenta possui, onde o usuário pode navegar e escolher a função que procura ou mesmo utilizar a busca encontrada acima desta lista para procurar por uma funcionalidade específica. No lado direito da interface do arquivo CHM gerado encontra-se a área onde é detalhada a função escolhida, com seus parâmetros e retornos.

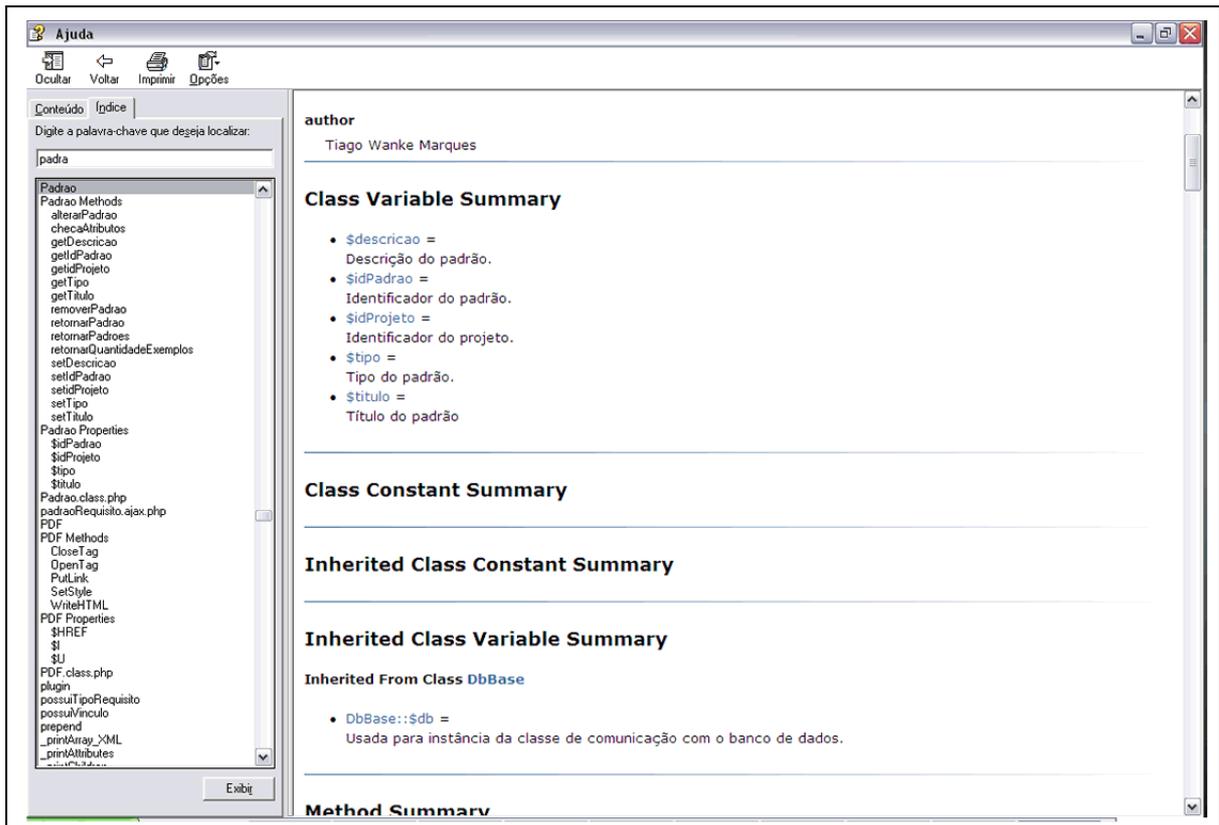


Figura 22 – Interface do arquivo CHM gerado

3.3.5 URL amigável

Durante a implementação da ferramenta um dos aspectos a ser pensado foi a utilização de URL amigável. Esse recurso permite ao desenvolvedor ou responsável configurar o texto exibido nas URL da ferramenta de acordo com sua vontade, aumentando a liberdade de personalização da ferramenta podendo associar as áreas da ferramenta a sua URL. Na prática significa exibir na URL algo como “[endereço]/padrao-requisito/adicionar” ao invés de “[endereço]/[pasta]/[arquivo].php”. Isto é possível através das classes de visualização comentadas no diagrama de classes, que através de expressões regulares identificam o conteúdo da URL e realizam a chamada da `view` que construirá a tela com o conteúdo requisitado ou mostrará um texto de página inexistente. O quadro 32 exibe o método da `view` que realiza o processo que permite a criação da URL amigável.

```

1. public function init() {
2.
3. $area = str_replace(_Sistema::URL_PATH, '', $_SERVER['REQUEST_URI']);
4.
5. switch (true) {
6.
7.     case ereg('^/?$', $area): // página inicial (login)
8.         ViewLogin::init();
9.         break;
10.    case ereg('^index.php$', $area): // página inicial (login)
11.        ViewLogin::init();
12.        break;
13.    case ereg('^master(.)*$', $area): // paginas master
14.        ViewInicialMaster::init();
15.        break;
16.    case ereg('^admin(.)*$', $area): // paginas admin
17.        ViewInicialAdmin::init();
18.        break;
19.    case ereg('^usuario(.)*$', $area): // paginas usuário
20.        ViewInicialUsuario::init();
21.        break;
22.    default:
23.        print 'Página Inexistente';
24.        break;
25. }
26. }

```

Quadro 32 – Método de criação da URL amigável

Na linha 3 está sendo atribuído a variável `area` o endereço da URL que o usuário digitou retirando o endereço principal. No exemplo: `www.naxes.com.br/contato` a variável receberá o endereço a partir do endereço padrão do sistema definido nas constantes da classe `_Sistema`, recebendo então apenas a área acessada (`contato` no exemplo). Após a variável receber a atribuição é verificado qual `view` deve ser chamada através de um `switch`. Cada condição do `switch` verifica através de expressões regulares utilizando a função nativa do PHP chamada `ereg()`. O código apresentado no quadro 32 está dentro da classe `View` no método `init()`.

3.3.6 Arquivo `lib` e o método `autoload`

O arquivo `lib` é um arquivo padrão do *Requisite Manager* (MARQUARDT, 2004), que é incluído por todos os scripts da ferramenta. Sua finalidade principal é executar a chamada do método `_autoload` assim como efetuar a chamada de validação do usuário que está acessando algum script da ferramenta, o quadro 33 detalha o arquivo `lib`.

```

1. <?php
2.
3. session_start();
4.
5. function __autoload($className) {
6.
7.     if(strpos( $className, 'View' ) !== false) {
8.
9.         require_once('D:/webroot/internet/requisiteManager5/sistema/
           view/' . $className . '.class.php');
10.     } else {
11.         require_once('D:/webroot/internet/requisiteManager5/sistema/
           classes/' . $className . '.class.php');
12.     }
13.
14. }
15.
16. _Sistema::verificaAcesso();
17.
18. ?>

```

Quadro 33 – Código fonte arquivo lib e método autoload

Na linha 3 é iniciada a sessão utilizada pelo servidor através do método nativo do PHP `session_start()`. O método `autoload` encontra-se entre as linhas 5 e 14, o método `autoload` somente está disponível a partir da versão 5 do PHP sendo um método nativo. O benefício oferecido por esse método é permitir que o desenvolvedor não necessite mais incluir em seus scripts desenvolvidos o arquivo das classes que ele utilizar. Assim todo momento que o desenvolvedor instanciar uma classe qualquer, caso ele não tenha incluído o script da classe, o método `autoload` executa um script automático, desapropriando essa responsabilidade do desenvolvedor, além de tornar seu código mais otimizado. No arquivo `lib` ainda é realizada na linha 16 a chamada do método `_Sistema::verificaAcesso()` que verifica se o usuário está ou não registrado no sistema para acessar a área que está requisitando na ferramenta.

3.3.7 Operacionalidade da implementação

Esta seção detalha a operacionalidade do sistema ao utilizar padrões de requisitos. Ao executar a ferramenta é apresentada a tela ilustrada na figura 23.

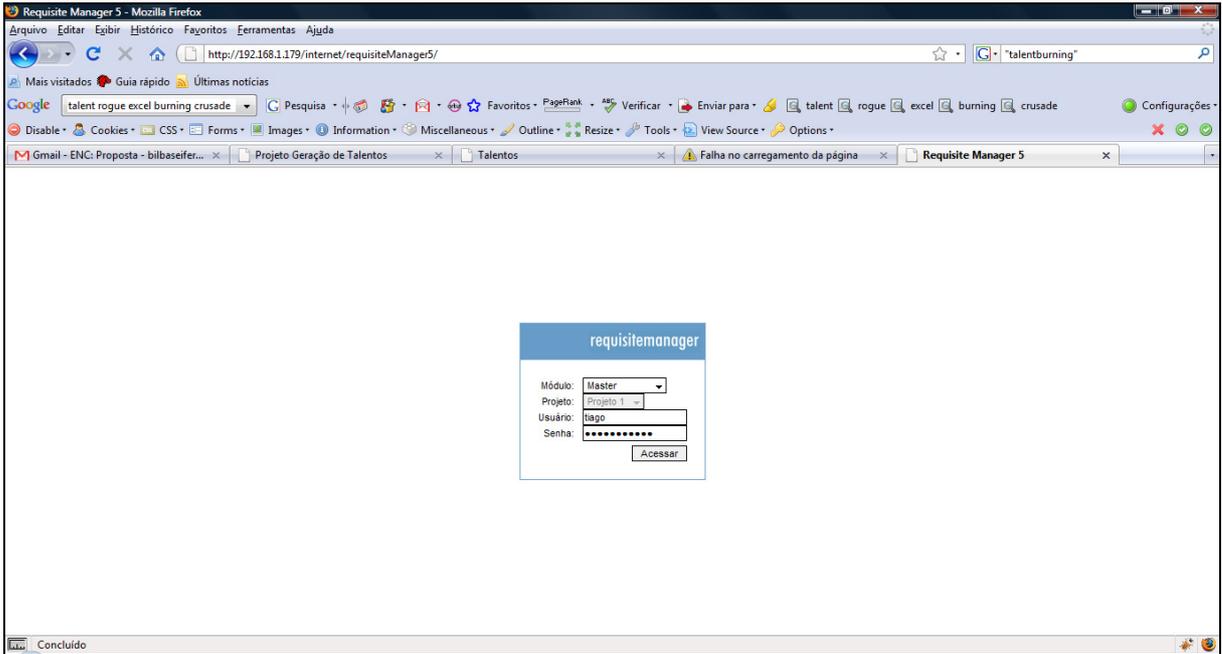


Figura 23 – Tela de login

A figura 23 ilustra a tela para efetuar *login* no sistema, onde o usuário poderá escolher o modo em que utilizará a ferramenta (usuário, administrador ou *master*) digitando na seqüência seu usuário e senha de acesso, ao clicar na opção acessar o sistema verificará os dados fornecidos, permitindo ou não a entrada do usuário na ferramenta, para explicar a operacionalidade do sistema será utilizado o módulo usuário ao acessar o sistema. A figura 24 mostra a tela inicial do módulo usuário ao acessar o sistema.

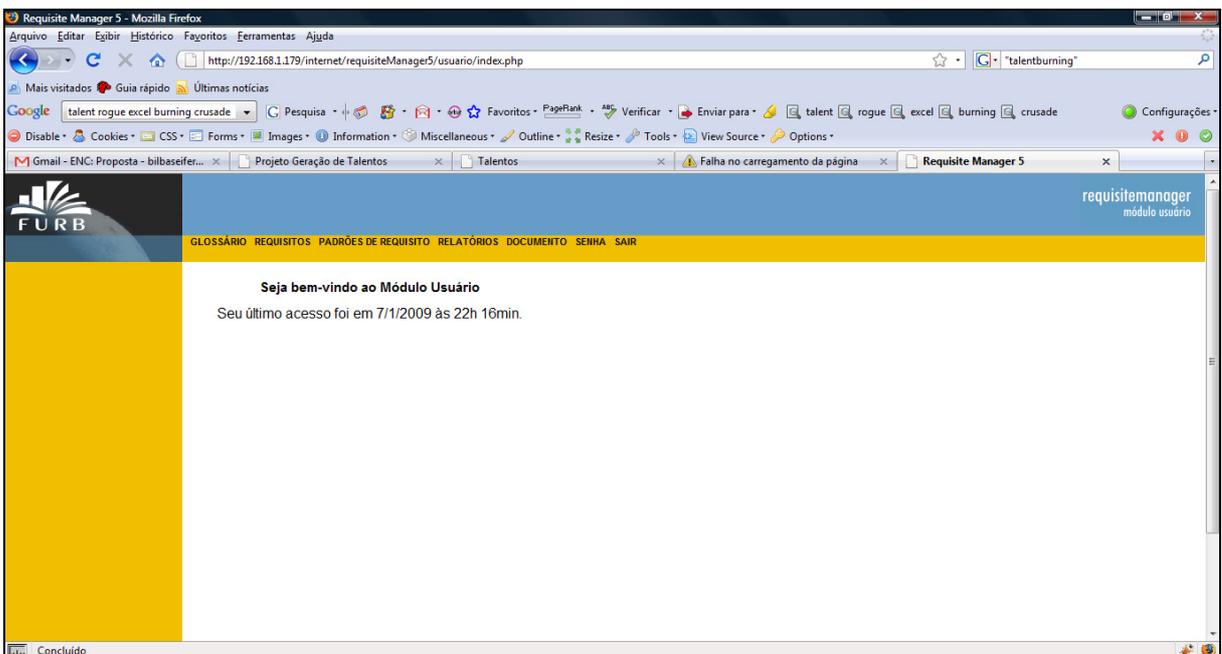


Figura 24 – Tela inicial do módulo usuário

Ao acessar o sistema selecionando o módulo usuário é apresentada a tela ilustrada na figura 24, nessa tela o usuário tem a opção de cadastrar um novo requisito, para isso deve-se

clicar na opção requisitos, encontrada no menu do topo do site. A figura 25 apresenta a tela de gerência de requisitos.

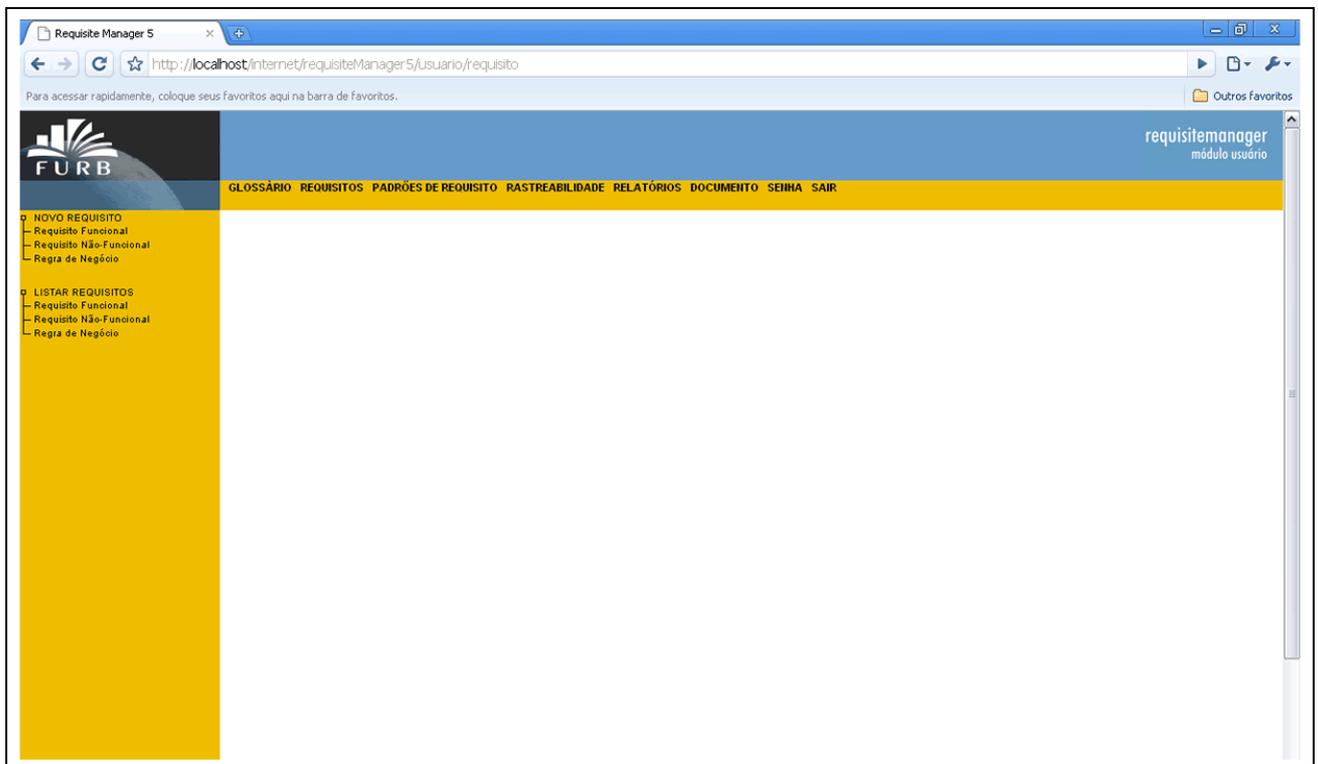


Figura 25 – Tela de gerência de requisitos

Após acessar o menu requisitos, é exibido o menu lateral categorizado em novo requisito e listar requisitos, como apresentado na figura 25. Os tipos de requisitos apresentados abaixo do menu novo requisito são os tipos de requisitos que podem ser cadastrados no projeto. Ao acessar qualquer um destes tipos de requisito é apresentado o formulário para cadastro do tipo de requisito acessado. Os tipos de requisitos apresentados abaixo do menu listar requisitos, se acessados, apresentam a lista dos requisitos cadastrados correspondente ao tipo de requisitos acessado.

Acessando a opção de cadastro de requisito funcional, é apresentada a tela da figura 26.

The screenshot shows a web browser window titled 'Requisite Manager 5' with the URL 'http://localhost/internet/requisiteManager5/usuario/requisito/adicionar/index.php?id=1'. The page features a blue header with the FURB logo and a navigation menu with items: GLOSSÁRIO, REQUISITOS, PADRÕES DE REQUISITO, RELATÓRIOS, DOCUMENTO, SEIHA, and SAIR. A yellow sidebar on the left contains two main sections: 'NOVO REQUISITO' with sub-items 'Requisito Funcional', 'Requisito Não Funcional', and 'Regra de Negócio'; and 'LISTAR REQUISITOS' with the same sub-items. The main content area is titled 'REQUISITOS > NOVO REQUISITO > REQUISITO FUNCIONAL' and 'Padrões de requisitos'. A yellow box labeled 'ATRIBUTOS' contains the following form fields: 'Nome:' with the value 'Relatório de requisitos'; 'Descrição:' with a text area containing 'é apresentado:' followed by a bulleted list: 'Nome do tipo de requisitos', 'Requisitos do tipo de requisito', and 'Para cada requisitos é apresentado:' followed by another bulleted list: 'Nome do requisitos', 'Descrição do requisito', and 'Atributos do requisito'; and 'Prioridade:' with a dropdown menu set to 'Alta'. At the bottom of the form are 'Salvar' and 'Cancelar' buttons.

Figura 26 – Tela de cadastro de requisito funcional

Na tela representada através da figura 26, o usuário se desejar, pode selecionar a opção padrões de requisito, antes de realizar o cadastro. Se o usuário não desejar realizar o cadastro de um novo requisito, pode optar por cancelar, voltando então à tela inicial de requisitos, apresentada na figura 25. Caso o usuário selecione a opção padrões de requisitos o sistema apresenta a tela com os padrões de requisitos funcionais existentes, como ilustrado na figura 27.

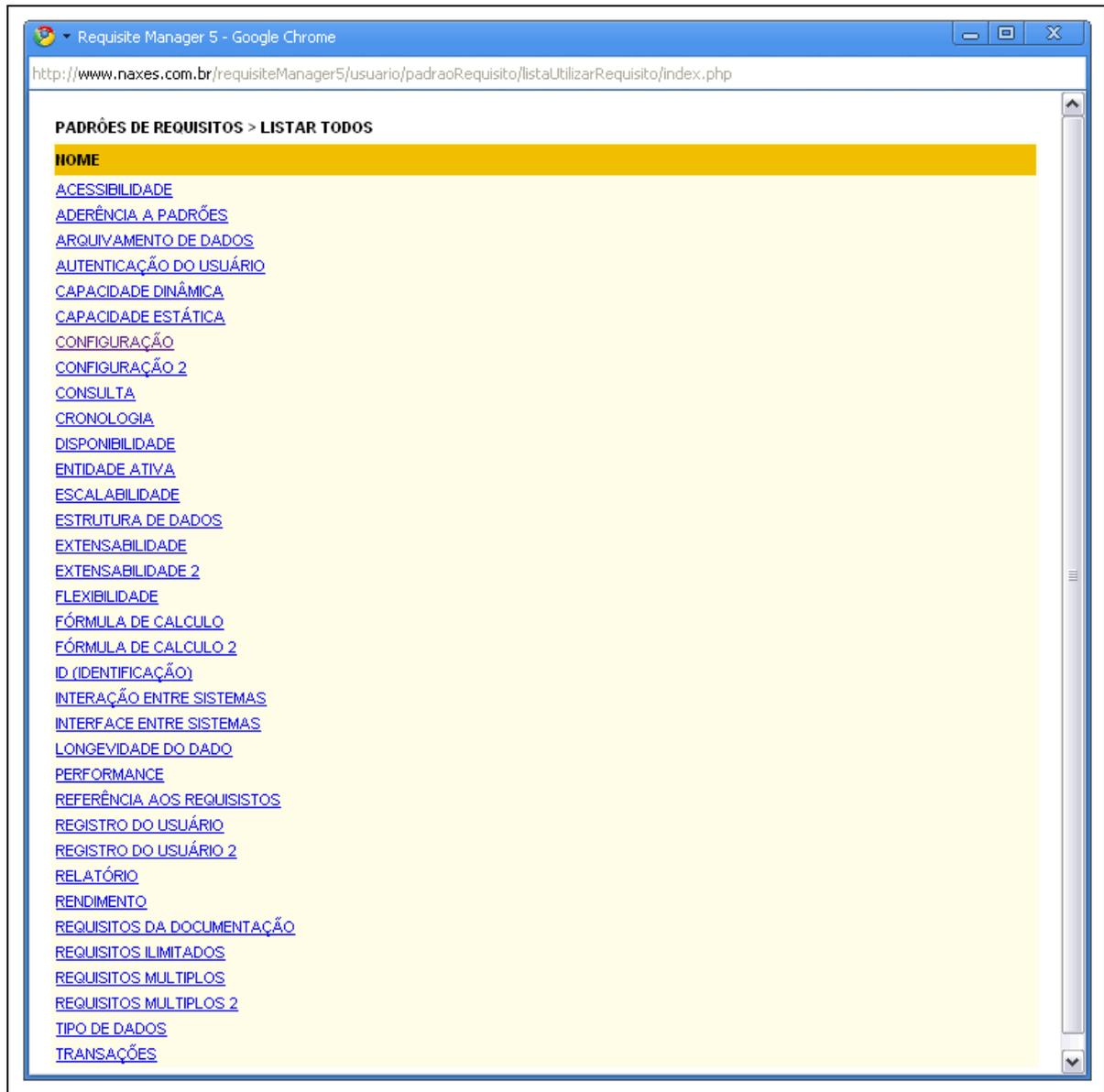


Figura 27 – Tela listando os padrões de requisitos funcionais

O usuário optará então por algum padrão de requisito funcional listado na figura 27. Ao selecionar o padrão é apresentada uma tela conforme a figura 28, onde a imagem representa a seleção do tipo de padrão de requisito funcional de relatório.



Figura 28 – Tela de padrão de requisito funcional de relatório

A figura 28 demonstra ao usuário como proceder ao se adicionar um requisito, conforme especificado pelo padrão e apresentando alguns exemplos. Também é possível utilizar o padrão de requisito, nessa opção o padrão é incluído diretamente no requisito a ser cadastrado. Após verificar ou incluído o padrão desejado, o usuário pode realizar o cadastro de um novo requisito, de forma que caso haja algum erro durante o processo de cadastro, uma mensagem de erro é apresentada ao usuário, mantendo-o na tela de cadastro. Caso não ocorram problemas durante o cadastro, uma mensagem de sucesso é apresentada e é retornada ao usuário a tela inicial de requisitos.

Para realizar o cadastro de um requisito não funcional ou regra de negócio o processo segue a mesma lógica do cadastro apresentado em um requisito funcional, de modo que são

utilizados os mesmos padrões para ambos os requisitos.

O usuário tem a opção de visualizar a lista de requisitos funcionais, não funcionais e regras de negócio, permitindo ao usuário a visualização do requisito, sua exclusão ou edição, conforme apresentado na figura 29.

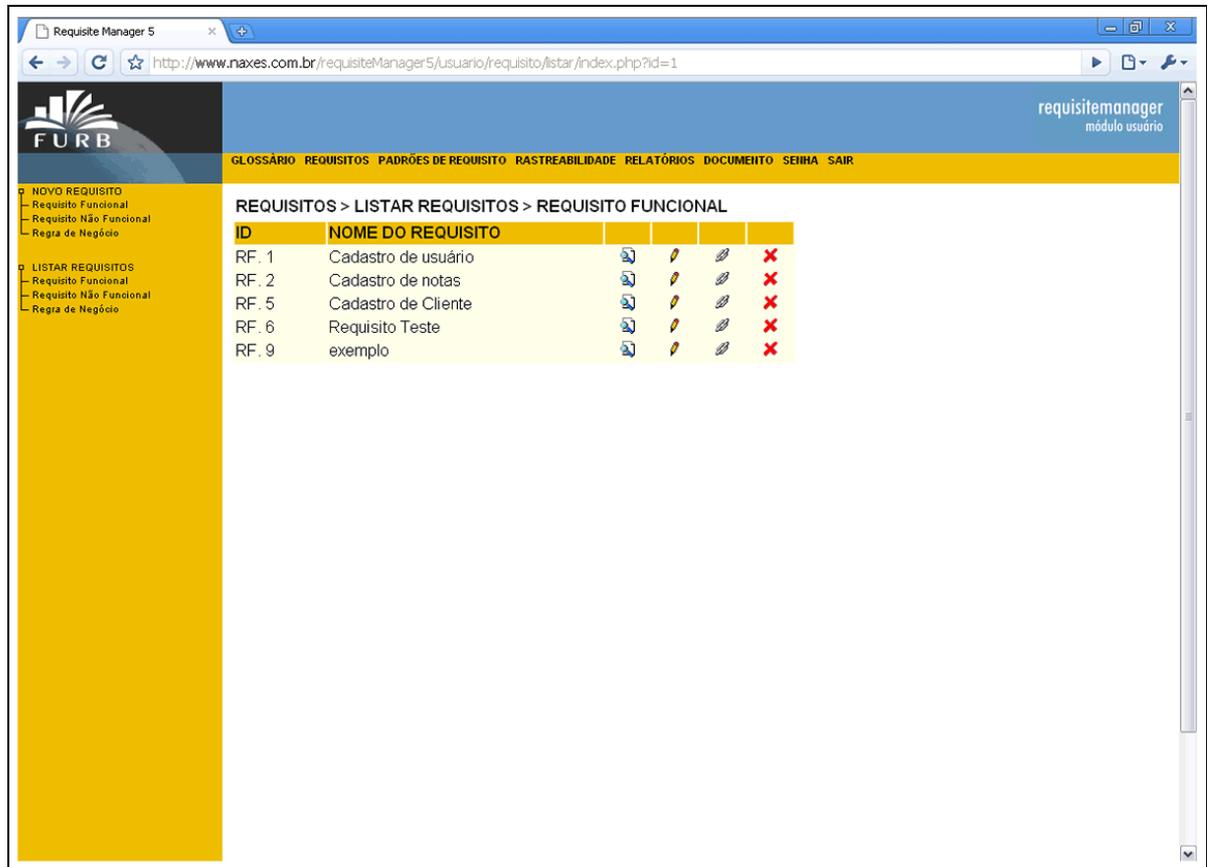


Figura 29 – Tela da lista de requisitos cadastrados

Caso o usuário opte por editar um requisito, é apresentado o formulário de edição dos dados atuais do requisito a ser modificado. O processo de edição de um requisito ocorre da mesma forma que sua adição. Ao selecionar a opção de exclusão de um requisito, uma mensagem é apresentada para verificar se o usuário realmente deseja excluir o requisito. Caso confirme a exclusão do requisito, o mesmo é removido.

No menu do topo da ferramenta além de requisitos é apresentado um menu chamado padrões de requisitos, onde é listado os padrões já cadastrados no sistema, além de oferecer as opções para adicionar um novo padrão, editar um padrão já existente, excluir um padrão e visualizar a lista de exemplos de cada padrão, conforme apresentado na figura 30.

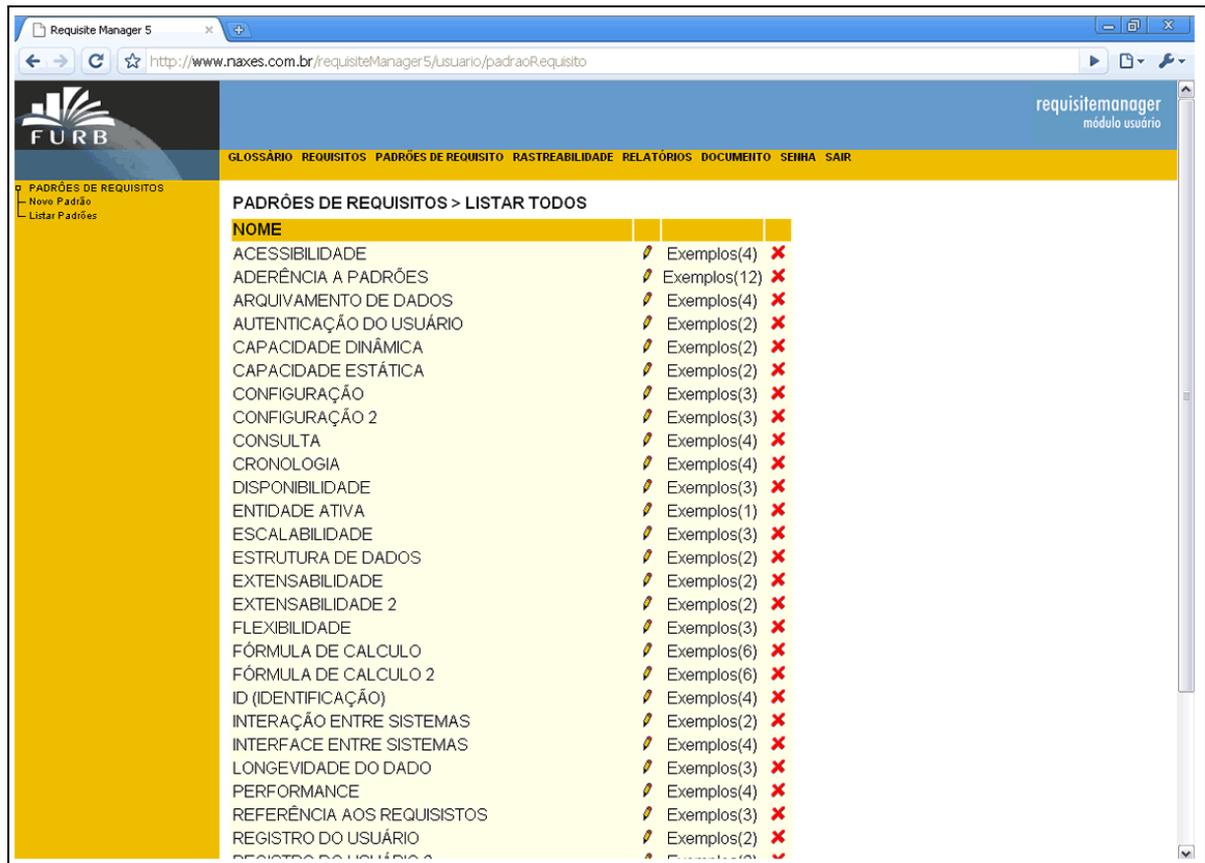


Figura 30 – Tela da lista de padrões de requisitos cadastrados

Ao acessar a opção para adicionar um novo padrão é apresentada a tela da figura 31. Após preencher os campos, o usuário deve optar pela opção salvar para cadastrar o novo padrão. Caso opte por excluir um padrão, uma mensagem é apresentada para confirmar a decisão do usuário. Caso confirme a exclusão do padrão, o mesmo é removido.

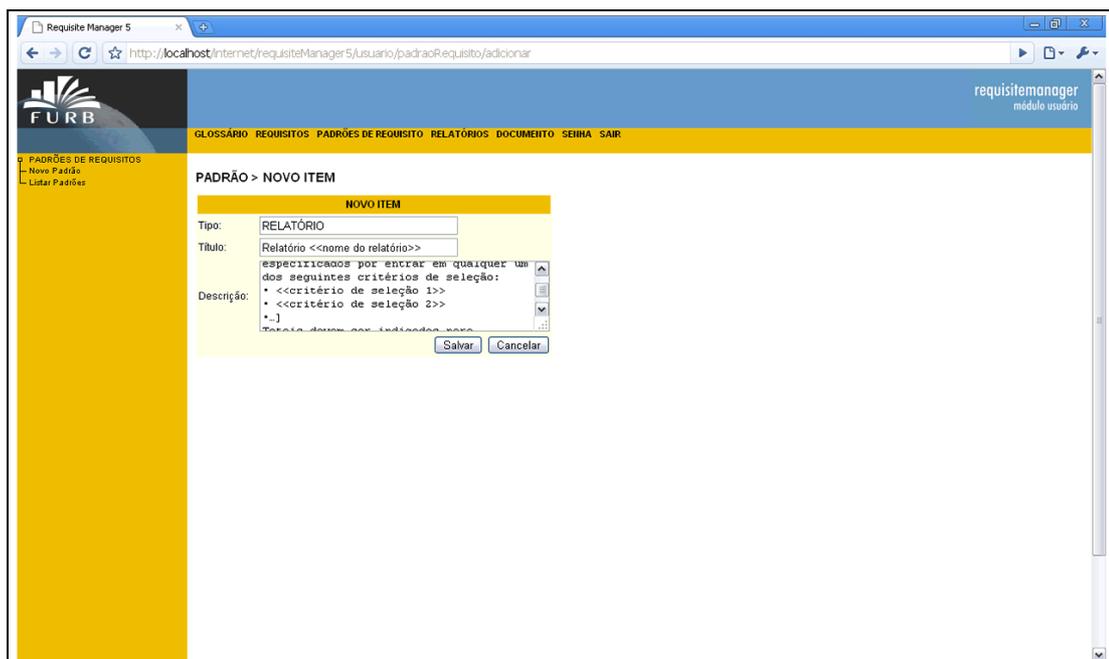


Figura 31 – Tela para adicionar um novo padrão de requisito

Ao selecionar a opção de edição de um padrão de requisito é apresentada uma tela com o formulário dos dados do padrão de requisito já preenchidos, sendo o mesmo processo da adição de um novo padrão de requisito.

Ao acessar os exemplos de padrões de requisitos é apresentada a tela da figura 32, onde se tem as opções de cadastrar, editar e remover algum exemplo. Caso o usuário opte por excluir algum exemplo a ferramenta apresenta uma mensagem para confirmar a exclusão, se confirmado o exemplo é removido.

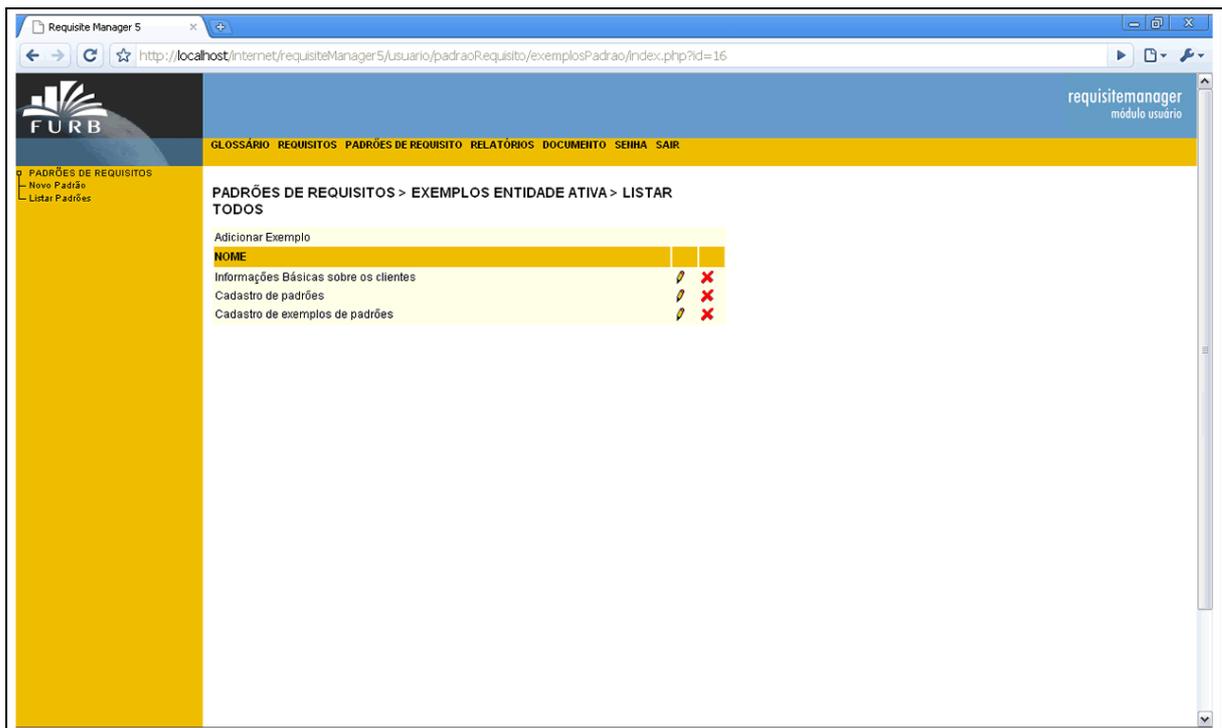


Figura 32 – Tela da lista de exemplos do padrão de requisitos de entidade ativa

A figura 33 representa o formulário para adicionar um novo exemplo para o padrão de requisito. O processo de edição de um exemplo do padrão de requisito é o mesmo de adição do exemplo, sendo que no processo de edição é apresentado o formulário com os dados preenchido.

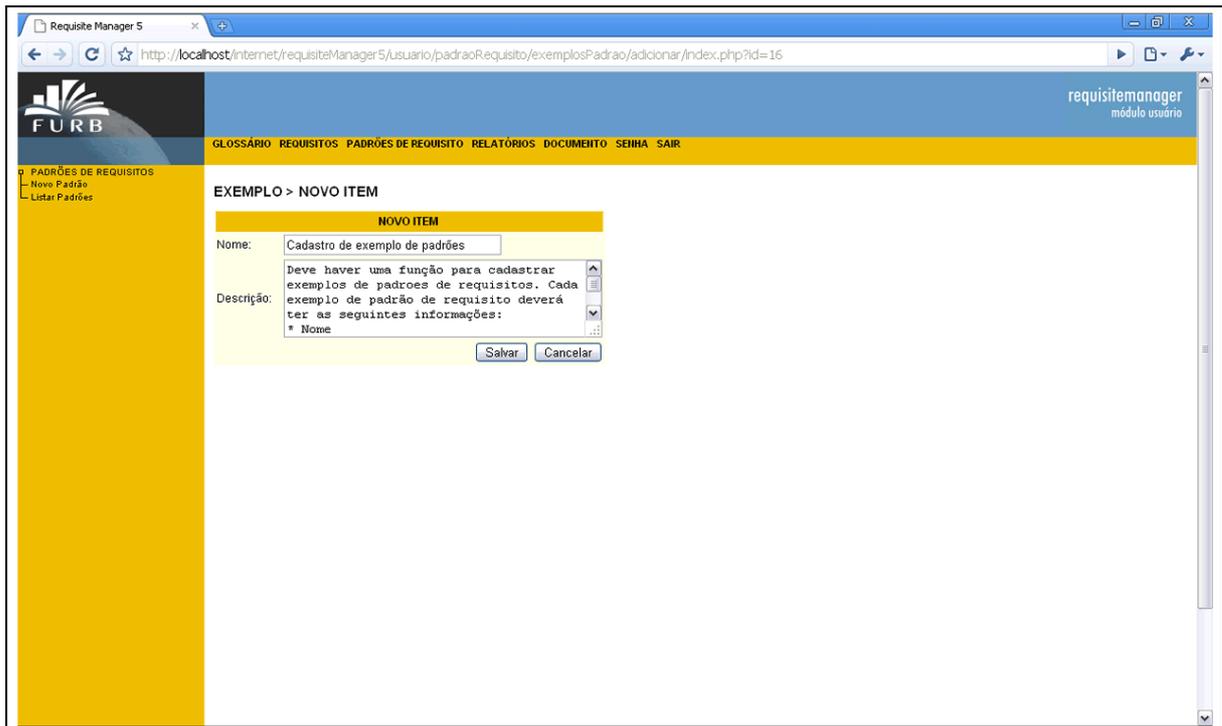


Figura 33 – Tela do formulário para adicionar um novo exemplo de padrão de requisito

A figura 34 mostra a tela para geração de relatórios, na opção de relatórios do sistema. Nessa etapa o usuário deve escolher o tipo de relatório desejado.

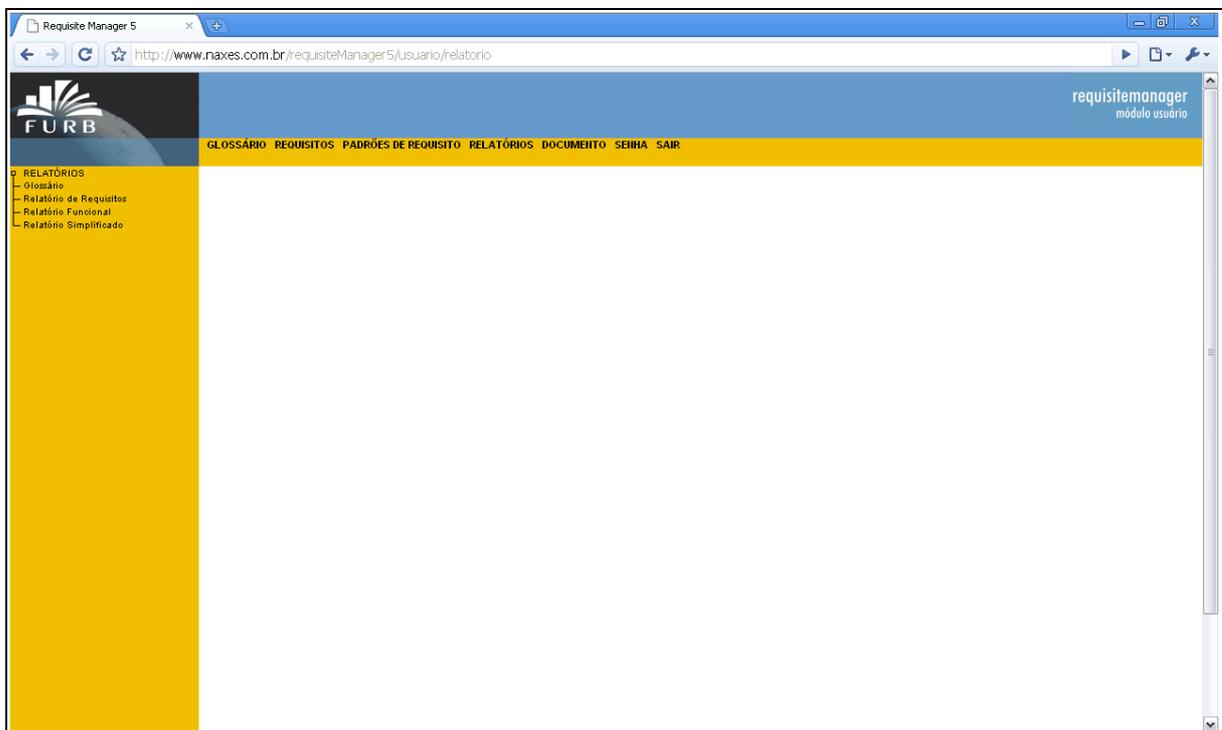


Figura 34 – Tela de opções de relatórios do sistema

Caso o usuário opte por gerar relatório de requisitos o sistema apresenta a tela com as opções de geração para esse tipo de relatório, como mostra a figura 35.

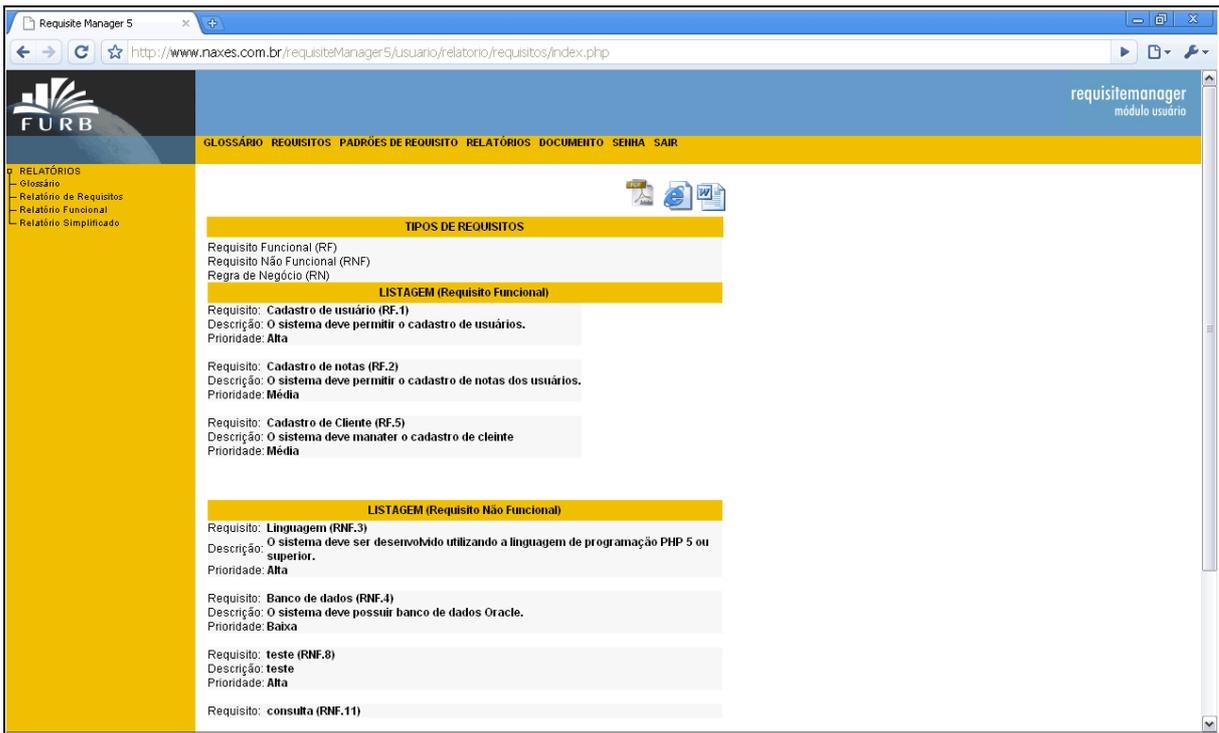


Figura 35 – Tela de opções de tipo de relatórios para relatório de requisitos

Caso o usuário opte por gerar o relatório de requisitos em PDF na opção do ícone PDF, o sistema apresenta uma nova página com o arquivo do relatório no formato PDF como é exemplificado na figura 36.

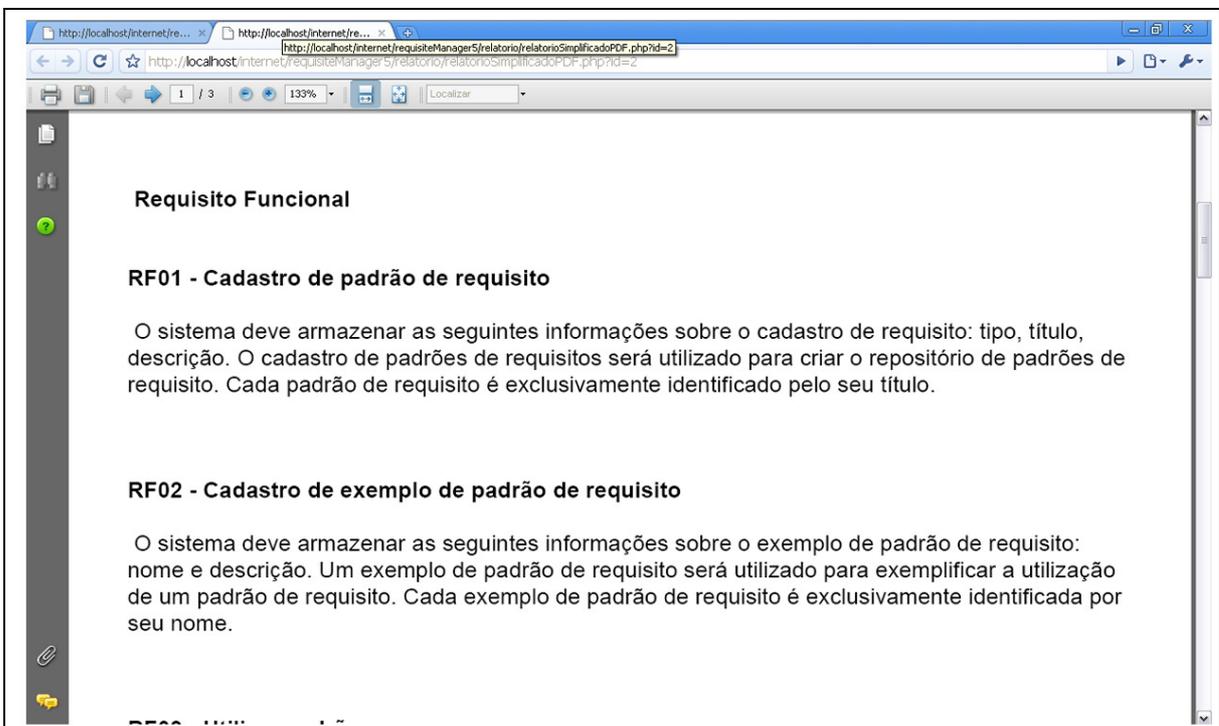


Figura 36 – Relatório de requisitos em formato PDF

Para editar a matriz de rastreabilidade do projeto, o usuário deve acessar no menu superior o item rastreabilidade, que irá apresentar a tela da figura 37.

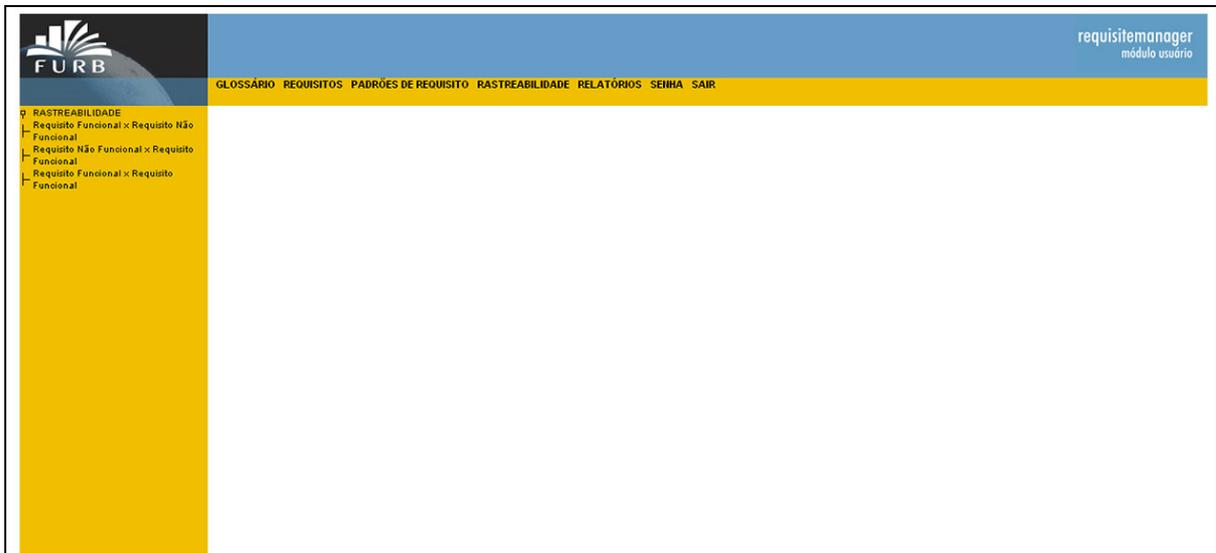


Figura 37 – Tela para escolher a matriz de rastreabilidade

O menu direito apresenta os tipos de vínculos permitidos entre os requisitos, para criar a matriz deve-se então escolher o tipo de vínculo desejado e será apresentada a matriz de rastreabilidade (figura 38).

The screenshot shows the 'requisitemanager' application interface with the 'RASTREABILIDADE' matrix editing screen. The title bar says 'requisitemanager módulo usuário'. Below this is a navigation bar with the following items: GLOSSÁRIO, REQUISITOS, PADRÕES DE REQUISITO, RASTREABILIDADE, RELATÓRIOS, SEIHA, SAIR. On the left side, there is a sidebar menu with the following options: RASTREABILIDADE, Requisito Funcional x Requisito Não Funcional, Requisito Não Funcional x Requisito Funcional, and Requisito Funcional x Requisito Funcional. The 'RASTREABILIDADE' option is currently selected. The main content area shows the title 'RASTREABILIDADE > Requisito Funcional x Requisito Funcional' and a table with the following columns and rows:

	RF01 - Cadastro de padrão de requisito	RF02 - Cadastro de exemplo de padrão de requisito	RF03 - Utilizar padrão	RF04 - Cadastro de requisito	RF05 - Cadastro de glossário	RF06 - Cadastro de atributo	RF07 - Vínculo entre os tipos de requisito	RF08 - Acesso de usuário por módulo	RF09 - Relatório	RF10 - Detalhes dos itens do glossários	RF11 - Detalhes dos requisitos	RF12 - Detalhes dos padrões de requisitos	RF13 - Detalhes dos exemplos de padrões de requisitos	RF Tip de req
RF01 - Cadastro de padrão de requisito	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
RF02 - Cadastro de exemplo de padrão de requisito	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
RF03 - Utilizar padrão	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
RF04 - Cadastro de requisito	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura 38 – Tela de edição da matriz de rastreabilidade

Para criar os vínculos entre os requisitos basta selecionar os vínculos desejados e eles são salvos automaticamente. Quando um requisito é vinculado a outro, ao usuário tentar remove-lo será apresentada uma mensagem de alerta informando que o requisito possui vínculos, assim o usuário deve informar se realmente quer remover o requisito.

3.4 RESULTADOS E DISCUSSÃO

Foi possível verificar que a falta de documentação e legibilidade de um código pode prejudicar muito o desenvolvimento de um sistema. Trabalhar utilizando o paradigma de orientação a objetos e documentação garante maior segurança do que está sendo feito como uma maior facilidade para a extensão da ferramenta.

Os quadros 34 e 35 exemplificam a melhoria aplicada na reestruturação do código fonte melhorando sua legibilidade e manutenção.

```

1. <?
2. include "../sessao.php";
3. include "../conn.php";
4.
5. $nomeTemplate = $_HTTP_POST_VARS["nomeTemplate"];
6. $sql = "select count(*) from template where nm_template like
   '$nomeTemplate'";
7. $rs = mysql_query($sql, $conn);
8. $linha = mysql_fetch_row($rs);
9.
10. if ($linha[0]==0) {
11.   $sql = "INSERT INTO template (cd_template, nm_template) VALUES
   (NULL, '$nomeTemplate')";
12.   mysql_query($sql, $conn);
13.   header("location: listarTemplates.php");
14.   exit;
15. }
16. ?>
17. <script language="javascript">
18.   window.alert("Já existe um template com este nome!\nInforme
   outro nome e tente novamente.");
19.   history.go(-1);
20. </script>

```

Quadro 34 – Codificação de Marquardt (2004) para adicionar um *template*

Ambos os quadros 34 e 35 mostram a funcionalidade de adicionar um *template*. Entretanto, o quadro 34 mostra a codificação antiga da ferramenta e o quadro 35 mostra a codificação atual da ferramenta com objetivo de melhorar sua legibilidade e facilitar a sua manutenção. Nota-se que nas linhas 6 a 8 do quadro 34 é realizada a conexão com o banco de dados no próprio processo de adicionar um *template*, já no quadro 35 essa conexão está implícita no método `adicionarTemplateProjeto` da linha 9, poupando o desenvolvedor do processo de conexão e inserção no banco quando o foco principal é adicionar um *template*. Esse é um benefício oferecido pelo paradigma de orientação a objetos. No quadro 35 com a utilização da biblioteca `xajax` é possível trabalhar em um único arquivo com código PHP e

javascript. Um exemplo pode ser visto na comparação das linhas 17 a 20 do quadro 34 onde é necessário a abertura do javascript para exibir uma mensagem ao usuário. Já no quadro 35 a linha 13 efetua a exibição dessa mensagem com o chamado de um único método, o que facilita o entendimento do código.

```

1. function adicionarTemplate($dados) {
2.     $objResponse = new xajaxResponse();
3.     _Sistema::limparAvisos();
4.     $nome = $dados['nome'];
5.
6.     $template = new TemplateProjeto();
7.     $template->setNome(utf8_decode($nome));
8.
9.     if(!$template->adicionarTemplateProjeto()) {
10.         $objResponse->alert(utf8_encode(_Sistema::avisosToString(
11.             _Sistema::retornarAvisos(TipoAviso::ERRO))););
12.         return $objResponse;
13.     } else {
14.         $objResponse->alert('Template adicionado com sucesso.');
```

```

14.         $objResponse->script('self.location = "' .
15.             _Sistema::URL_PATH . 'master/template"');
16.     }
17.     return $objResponse;
18. }
19. $xajax->registerFunction('adicionarTemplate');
```

Quadro 35 – Codificação atual para adicionar um *template*

A possibilidade de utilização de padrões de requisitos para auxiliar na criação dos requisitos é de grande valia no momento em que se está definindo as funcionalidades de um sistema. Com eles, evitam-se detalhes que muitas vezes passam despercebidos na elicitação dos requisitos. Criar exemplos para os padrões e possibilitar a consulta deles no momento do levantamento de requisitos também é essencial, pois assim consegue-se verificar com mais segurança e os detalhes que precisam ser esclarecidos.

Os principais diferenciais entre a ferramenta estendida e ferramenta anterior estão apresentadas no quadro 36.

DIFERENCIAIS ALCANÇADOS		
	Ferramenta anterior	Ferramenta extendida
Orientação a objetos	Não	Sim
Utilização de AJAX	Não	Sim
Gerência de requisitos	Sim	Sim
Padrões de requisitos	Não	Sim
Execução na web	Sim	Sim
Documentação do código fonte	Não	Sim
Distinção da camada lógica e de interface	Não	Sim

Quadro 36 – Principais diferenciais alcançados com a execução desse trabalho

A seguir encontra-se a definição de cada item do quadro 36:

- a) orientação a objetos: foi utilizado esse paradigma que define uma área de negócio através de objetos e desenvolve o sistema por meio de interações entre esses objetos (troca de mensagens), identificando qual o melhor conjunto de objetos para representar o sistema;
- b) AJAX: utilizada esta tecnologia para tornar as páginas mais interativas com o usuário;
- c) gerência de requisitos: possibilidade de adicionar, alterar e excluir requisitos no projeto, como também os atributos dos requisitos;
- d) padrões de requisitos: possibilidade da criação e utilização de um repositório de padrões de requisitos por qualquer usuário da ferramenta;
- e) execução web: sistema disponível a qualquer momento e acesso via web;
- f) documentação do código fonte: documentação de todas as funcionalidades do sistema a nível de desenvolvedor;
- g) distinção da camada lógica e de interface: utilizado conceito de separação do código de interface e lógico facilitando a legibilidade.

4 CONCLUSÕES

Os objetivos propostos para este trabalho foram desenvolvidos com êxito. A ferramenta incorpora o conceito de padrões de requisitos, a partir de um repositório de padrões de requisitos permitindo qualquer usuário do sistema utilizá-los na criação de seus requisitos. A criação de um repositório de padrões de requisito permite consultar, criar, editar, remover e utilizar padrões de requisitos. Inicialmente já possui cadastrados 37 padrões de requisitos propostos por Whitall (WHITALL, 2008). Para a descrição dos requisitos deste trabalho foram usados os próprios padrões de requisitos demonstrando a potencialidade e padronização.

Outro objetivo desenvolvido foi a reestruturação de todo o código fonte do sistema levando em consideração a legibilidade e documentação para a geração do arquivo no formato .chm para consultas e manutenções futuras. Na reestruturação foi utilizado o paradigma de orientação a objetos, além de tecnologias como o AJAX que permite acrescentar performance na aplicação e CSS que facilita a manutenção da interface do sistema agregando também legibilidade nesse aspecto.

Não foram realizadas melhorias na interface da ferramenta estendida, o que é considerado uma limitação deste trabalho.

4.1 EXTENSÕES

Sugestão de extensões para a presente ferramenta, gerando trabalhos futuros são:

- a) implementar heurísticas de Nielsen para melhorar a interface aplicando padrões de usabilidade;
- b) incorporar novos tipos de relatórios a ferramenta como relatórios de padrões de requisitos;
- c) criação de um novo nível de acesso para o cliente do projeto que valida os requisitos de um projeto e comenta sobre os mesmos.

REFERÊNCIAS BIBLIOGRÁFICAS

ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO – SOFTEX. **MPS.BR – guia geral**, jun. 2007. Disponível em: <<http://www.softex.br>>. Acesso em: 25 ago. 2007.

BATISTA, Raphael M. **Ferramenta de gerenciamento de requisitos de software integrada com enterprise architect**. 2007. 67 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

DECARLE, Luiz S.; GRAHL, Everaldo A. **Experiência prática de aplicação de padrões de requisitos de software**. 2008. 9 f. Artigo de Conclusão de Curso (Especialização em Gestão de Desenvolvimento de Software), Instituto Catarinense de Pós-Graduação, Blumenau.

GATHERSPACE. **Agile project management, requirements management**. Santa Monica, 2008. Disponível em: <<http://www.gatherspace.com/>>. Acesso em: 31 jul. 2008.

GOMES, Wagner. **Engenharia de software**. [S.l.], 2008. Disponível em: <<http://wagnergomes.wordpress.com/category/engenharia-de-software/>>. Acesso em: 02 nov. 2008.

INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. **Standards collection: software engineering**. New York: NY, 1994.

MAGELA, Rogério. **Engenharia de software aplicada**. Rio de Janeiro: Alta books, 2006.

MARQUARDT, Luciano. **Ferramenta web para gerenciamento de requisitos de software**. 2004. 86 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

MEISEN, Mariane. **Ferramenta de apoio a gerência de requisitos baseada no modelo CMMI**. 2005. 87 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

PAULA FILHO, Wilson de P. **Engenharia de software: fundamentos métodos e padrões**. 2. ed. Rio de Janeiro: LTC – Livros Técnicos e Científicos, 2001.

PFLEEGER, Shari L. **Engenharia de software: teoria e prática**. 2. ed. Tradução Dino Franklin. Revisão técnica Ana Regina Cavalcanti da Rocha. São Paulo: Prentice Hall, 2004.

PRESSMAN, Roger S. **Engenharia de software**. 6. ed. Tradução Rosângela D. Penteado. Revisão Técnica Fernão Stella R. Germano, José Carlos Maldonato e Paulo Cesar Masiero. São Paulo: McGraw-Hill, 2006.

SCHMIDT, Juliana. **Aplicação de padrões**. 2008. 92 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

SOMMERVILLE, Ian. **Engenharia de software**. 6. ed. Tradução André Maurício de Andrade Ribeiro. Revisão técnica Kechi Hiramã. São Paulo: Addison Wesley, 2003.

TAGLIATI, V. Luca; JOHNSON, Roger; ROUSSOS, George. **Requirements analysis Evolution through patterns**. [S.l.], [2005?], Disponível em: <http://www.dcs.bbk.ac.uk/~gr/pdf/seke07_requirement_patterns_2.pdf>. Acesso em: 08 jan. 2009.

WHITALL, Stephen. **Software requirements patterns**. [S.l.], 2008, Disponível em: <http://www.withallyourequire.com/software_requirements_patterns.pdf>. Acesso em: 08 jan. 2009.

_____. **Software requirement patterns**. Washington: Microsoft Press, 2007.

APÊNDICE A – Relatório de requisitos

O quadro 37 apresenta o relatório de todos os requisitos da ferramenta gerado pela própria ferramenta.

RELATÓRIO DE REQUISITOS SIMPLIFICADO

Projeto: Projeto TCC

Usuários: admin (Administrador)user (Usuário)

Glossário

Padrão de Requisito: O padrão explica como descrever determinado tipo de requisito, como expressá-lo e revela possíveis requisitos adicionais implícitos. Padrões de requisitos são utilizados somente para escrever um único requisito. Um padrão não se preocupa em representar o sistema como um todo, atendo-se apenas ao tipo de requisito específico.

Gerência de requisitos: Gerência de requisitos é gerenciar os requisitos dos produtos e componentes do projeto e identificar inconsistências entre os requisitos, os planos de projeto e os produtos de trabalho do projeto, seu principal objetivo é controlar a evolução dos requisitos.

Requisito Funcional

RF01 - Cadastro de padrão de requisito

O sistema deve armazenar as seguintes informações sobre o cadastro de requisito: tipo, título e descrição. O cadastro de padrões de requisitos será utilizado para criar o repositório de padrões de requisito. Cada padrão de requisito é exclusivamente identificado pelo seu título.

Classificação - Entidade Ativa

RF02 - Cadastro de exemplo de padrão de requisito

O sistema deve armazenar as seguintes informações sobre o exemplo de padrão de requisito: nome e descrição. Um exemplo de padrão de requisito será utilizado para exemplificar a utilização de um padrão de requisito. Cada exemplo de padrão de requisito é exclusivamente identificado por seu nome.

Classificação - Entidade Ativa

RF03 - Utilizar padrão

Deve haver uma função de utilizar um padrão de requisito para um requisito a ser cadastrado. Cada padrão de requisito utilizado deve conter a descrição. A utilização de um padrão é feita para facilitar a criação de requisitos de um projeto.

Classificação - Transação

RF04 - Cadastro de requisito

O sistema deve armazenar as seguintes informações sobre o requisito: nome, descrição e atributos correspondentes ao tipo de requisito.

Um requisito é utilizado para explicar uma determinada característica do sistema a ser desenvolvido. Cada requisito é exclusivamente identificado pelo seu nome.

Classificação - Entidade Ativa

RF05 - Cadastro de glossário

O sistema deve armazenar as seguintes informações sobre o glossário: palavra e descrição. Um glossário é utilizado para explicar determinadas palavras técnicas ou da regra de negócio. Cada glossário é exclusivamente identificado pela palavra.

Classificação - Entidade Ativa

RF06 - Cadastro de atributo

O sistema deve armazenar as seguintes informações sobre o atributo: nome e tipo (combo box, texto ou caixa de texto).

Um atributo é utilizado por um ou mais tipos de requisitos. Cada atributo é exclusivamente identificado por seu nome.

Classificação - Entidade Ativa

RF07 - Vínculo entre os tipos de requisito

Deve haver uma função de criar um vínculo entre dois tipos de requisitos quaisquer. Cada vínculo deve conter as seguintes informações: tipo de requisito origem e tipo de requisito destino. Um vínculo é utilizado para permitir o usuário criar uma matriz de rastreabilidade entre os tipos de requisitos que possuem vínculo. Cada vínculo é exclusivamente identificado por seu tipo de requisito destino e origem e pelo template a qual o vínculo pertence.

Classificação - Configuração

RF08 - Acesso de usuário por módulo

Os módulos master, administrador e usuário devem ser capazes de auto-autenticar (login) pelo login e senha Informados pelo usuário.

Classificação - Autenticação do Usuário

RF09 - Relatório

Deve haver relatórios que mostram informações sobre os requisitos de cada projeto criado. O objetivo dos relatórios é de gerar um documento com todas as informações sobre o projeto criado.

RF10 - Detalhes dos itens do glossários

Cada item do glossário deverá ter as seguintes informações: nome do item e descrição.
Classificação - Estrutura do dado

RF11 - Detalhes dos requisitos

Cada requisito cadastrado deverá ter os seguintes itens de informação: tipo do requisito, nome do requisito, versão do requisito, login da pessoa que fez a alteração do requisito, descrição do requisito e todos os demais atributos e seus respectivos valores cadastrados pelo usuário.
Classificação - Estrutura do dado

RF12 - Detalhes dos padrões de requisitos

Cada padrão de requisito cadastrados deverá ter os seguintes itens de informações: nome do padrão de requisito, título do padrão de requisito e descrição do padrão de requisito.
Classificação - Estrutura do dado

RF13 - Detalhes dos exemplos de padrões de requisitos

Cada exemplo de padrão de requisito cadastrado deverá ter os seguintes itens de informações: nome e descrição.
Classificação - Estrutura do dado

RF14 - Tipos de requisito

Deve ser possível especificar os tipos de requisitos que serão permitidos em um projeto com o objetivo de deixar restrito ao usuário master definir os tipos de requisitos de um projeto. Trata-se de um valor que corresponde ao nome do tipo de requisito (por exemplo, Requisito Funcional). Os tipos de requisitos podem ser modificados apenas se nenhum requisito do tipo que deseja ser modificado foi criado.
Classificação - Configuração

RF15 - Atributos dos tipos de requisitos

Deve ser possível especificar os atributos que os tipos de requisitos terão com o objetivo de deixar restrito ao usuário master definir os atributos de um tipo de requisito. Trata-se de um valor que corresponde ao nome do atributo e seu tipo (por exemplo, Prioridade uma caixa de texto com as opções de Alta, Baixo e Média). Os atributos podem ser modificados apenas se nenhum requisito com este atributo foi criado.

Classificação - Configuração

RF16 - Versão dos requisitos

Cada alteração feita nos requisitos serão automaticamente gravadas. Para cada uma, serão registrados o seguinte: login do usuário, data de alteração, nome do requisito, sua descrição e seus atributos.

Classificação - Cronologia

RF17 - Registro de administradores

Um usuário master deve ser capaz de registrar um ou mais administradores para o projeto criado com as seguintes informações: login e senha.

Classificação - Registro de Usuário

RF18 - Registro de usuários

Um usuário administrador deve ser capaz de registrar um ou mais usuários para seu projeto com as seguintes informações: login e senha.

Classificação - Registro de Usuário

RF19 - Alterar senha

Um usuário master, administrador ou usuário deve ser capaz de alterar sua própria senha. Ao alterar a sua senha o usuário deve informar: a sua senha atual, a sua nova senha, duas vezes.

Classificação - Registro de Usuário

RF20 - Nunca exibir senha

Senhas nunca devem ser exibidas na tela quando estão sendo digitadas ou em qualquer outro momento.

Classificação - Registro de Usuário

RF21 - Autenticação de usuário

Um usuário deve ser capaz de auto-autenticar (login), e deve fazê-lo antes que ele possa acessar

qualquer função do sistema.
Classificação - Autenticação do Usuário

RF22 - Usuário desconectado

Um usuário deve ser capaz de tirar sua autenticação (log out). Depois de retirar sua senha, o sistema deve tratá-los como se não tivessem feito a autenticação antes - e que respeita todas as ações posteriores como sendo realizada por um usuário desconhecido.
Classificação - Autenticação do Usuário

RF23 - Mostrar última vez que conectou

Quando o usuário tem sua autenticação com sucesso, a data e hora que ele ficou conectado na última vez será exibida. O objetivo deste requisito é dar a oportunidade de descobrir se um impostor está agindo em seu nome nos últimos acessos ao sistema.
Classificação - Autenticação do Usuário

RF24 - Ocultar informações inacessíveis

Locais de onde as funções que são normalmente selecionados (por exemplo, nos menus e botões ou hiperlinks) não devem exibir funções as quais o atual usuário não tem acesso.
Classificação - Autorização Configurável

RF25 - Matriz de rastreabilidade

Deve ser possível criar uma matriz de rastreabilidade a partir dos vínculos criados pelo usuário master. A matriz é utilizada para mostrar os requisitos que possuem ligações e alertar quais vínculos um requisito possui caso o usuário queira removê-lo.
Classificação - Transação

Requisito Não Funcional

RNF01 - Relatório de requisitos no formato PDF

O formato PDF deve ser usado para gerar o relatório de requisitos tanto no módulo master como no módulo usuário.
Classificação - Tecnologia

RNF02 - Utilizar ajax, css e Javascript

As tecnologias AJAX, Cascading Style Sheets CSS e javascript devem ser usadas para melhorar o desempenho da ferramenta como também aumentar a legibilidade do código fonte da mesma.

Classificação - Tecnologia

RNF03 - Documentação do código fonte

Deve haver um documento de ajuda que contém a documentação de todas as classes e seus respectivos métodos utilizadas para desenvolver a ferramenta. Deve ser no formato chm (arquivo de ajuda do windows). Ele deve ser escrito em português.

Classificação - Documentação

RNF04 - Acesso do usuário via navegador da web

Todas funções do usuário serão acessíveis através de um navegador web.

Classificação - tecnologia

RNF05 - Navegadores populares da web

A interface do usuário deve ser baseada na WEB e todas as funções deve funcionar plenamente com os seguintes browsers: Internet Explorer 6.0, Mozilla Firefox 3.0, e Chrome.

Classificação - Tecnologia

RNF06 - Servidor utilizado

O servidor a ser utilizado para rodar o sistema deve ser o Apache versão 2 ou superior.

Classificação - Tecnologia

Quadro 37 – Relatório de requisitos da ferramenta gerado pela ferramenta

ANEXO A – Padrões de requisitos de software

Este anexo apresenta o catálogo com os trinta e sete padrões de requisitos propostos por Whitall (WHITALL, 2008) detalhados no artigo de Decarle e Grahl (2008, p. 7-9) categorizados em oito domínios.

Domínio fundamental:

- a) tecnologia (especificar a tecnologia que deve ou não ser utilizada para construir ou executar o sistema, ou com que tecnologia o sistema deve ser capaz de interagir ou ser compatível);
- b) aderência a padrão (especificar que o sistema deve ser aderente a um determinado padrão);
- c) referência a requisitos (especificar que alguns ou todos os requisitos de uma especificação externa devem ser atendidos como se fossem requisitos presentes na especificação corrente);
- d) documentação (especificar que um determinado tipo de documentação necessita ser produzido);
- e) interface entre sistemas (especificar detalhes básicos de uma interface entre o sistema a ser especificado e qualquer sistema ou componente externo, com o qual necessita interagir);
- f) interação entre sistemas (especificar um tipo particular de interação empregado através de uma interface entre sistemas).

Domínio performance:

- a) tempo de resposta (especificar quanto tempo o sistema pode levar para responder a uma solicitação);
- b) capacidade dinâmica (especificar a capacidade de processamento simultâneo do sistema);
- c) capacidade estática (especificar a capacidade de armazenamento do sistema, normalmente num banco de dados);
- d) disponibilidade (definir quando o sistema está disponível para os usuários);
- e) rendimento (especificar um índice ou taxa na qual o sistema ou uma interface entre sistema deve ser capaz de executar algum tipo de processamento de entrada ou saída).

Domínio informação:

- a) tipo de dado (definir como um item de informação para um propósito específico de

negócio é representado ou apresentado);

- b) identificação (definir como será a identificação única das entidades de dados);
- c) estrutura do dado (definir como será composto um determinado item de informação);
- d) fórmula de cálculo (especificar como calcular um determinado tipo de valor ou como determinar o valor através de alguns passos lógicos);
- e) arquivamento do dado (especificar a movimentação ou cópia de dados de um local de armazenamento para outro);
- f) longevidade do dado (especificar por quanto tempo um certo tipo de dado deve ser mantido ou por quanto tempo deve estar disponível).

Domínio entidade de dados:

- a) entidade ativa (definir um tipo de entidade para a qual as informações são armazenadas e que tem uma vida útil, ou seja, é criada, pode ser modificada várias vezes e eventualmente encerrada);
- b) transação (definir um tipo de evento durante a existência de uma entidade ativa e/ou uma função para iniciar uma transação);
- c) configuração (definir valores de parâmetros que controlam como o sistema se comporta);
- d) cronologia (especificar como determinados eventos do sistema devem ser registrados).

Domínio função do usuário:

- a) consulta (definir uma consulta em tela que mostra informações específicas ao usuário);
- b) relatório (definir um relatório que mostra informações específicas ao usuário);
- c) acessibilidade (especificar a extensão em que um sistema ou parte dele deve ser acessível por pessoas com certo tipo de deficiência ou outra necessidade específica);

Domínio flexibilidade:

- a) escalabilidade (especificar como um sistema será capaz de expandir sem problemas, normalmente para acomodar o crescimento do volume de negócios);
- b) extensibilidade (ordenar que um aspecto específico do sistema a ser construído pode ser facilmente estendido pelo acréscimo de um software extra – plug-in);
- c) instalabilidade (especificar o grau de facilidade para instalar ou atualizar um sistema ou parte dele);

- d) multi-linguagem (especificar que um sistema é capaz de apresentar sua interface em mais de uma língua);
- e) múltiplo (especificar que um sistema deve acomodar múltiplas empresas e moedas ao mesmo tempo);
- f) ilimitado (especificar uma maneira particular em que o sistema não deve ser limitado a um ambiente empresarial).

Domínio controle de acesso:

- a) registro do usuário (especificar como novos usuários são registrados);
- b) autenticação do usuário (especificar que uma pessoa deve ser reconhecida antes de acessar qualquer função não pública ou que não possa permanecer anônima);
- c) autorização específica (especificar que um conjunto de usuários está autorizado ou não, a fazer ou ver certas coisas);
- d) autorização configurável (especificar que a autorização dos usuários é configurável);
- e) aprovação (especificar que uma determinada ação deve ser aprovada por uma segunda pessoa antes de ocorrer).

Domínio comercial:

- a) unidade multi-organizacional (especificar um tipo de estrutura organizacional que o sistema deve ser capaz de suportar);
- b) taxas (especificar qualquer taxa, pagamento ou imposto que o sistema deve calcular ou apresentar).

ANEXO B – Padrões de requisitos fundamentais Schmidt (2008)

PADRÕES DE REQUISITOS FUNDAMENTAIS

Padrões de Requisitos - Interface Entre Sistemas

Modelo (s)

RESUMO	DEFINIÇÃO
<<Nome da Interface >> Interface (<<Interface ID>>)	<p>Deverá haver uma interface bem definida (chamado <<Interface ID>>) entre <<Componente 1>> e <<Componente 2>>.</p> <p>Finalidades:</p> <ol style="list-style-type: none"> 1. <<Objetivo da Interface 1>>. 2. <p>Interações entre as interfaces podem ser iniciada por <<Iniciando componente (s)>>.</p> <p>As definições de interfaces são de responsabilidade do proprietário por <<Proprietário das Organizações de Interface >>.</p> <p>(Essa interface deve cumprir versão <<Versão Padrão>> do <<Nome Padrão>>, cuja definição pode ser encontrada em <<Localização Padrão>>).</p> <p>(<<Descrição da Tecnologia>>).</p>

Exemplo (s)

RESUMO	DEFINIÇÃO
Interface de Monitor de Alarme (i5)	<p>Deverá haver uma interface bem definida (chamada i5) entre o sistema e um monitor com alarme externo. É invocado somente pelo sistema. Apropriação desta interface é no âmbito do sistema.</p> <p>O seu objetivo consiste em notificar o pessoal adequado de qualquer evento que é relevante para eles (que normalmente significa apenas problemas graves).</p>

Padrões de Requisitos - Interação Entre Sistemas

Modelos

Resumo	Definição
<<Sumário interativo>>	O <<nome da interface>> interface (Interface "ID") é a <<descrição da finalidade da interação>> (que passa, no mínimo, as seguintes informações: <<Informação para passar>>).

Exemplos

Resumo	Definição
Monitor do alarme deve elevar o despertador	A interface monitor do alarme (i6) deve permitir que o sistema aumente o alarme (isto é, para transmitir uma mensagem de alarme e causar a todas as pessoas apropriadas para serem notificadas). O pedido de alarme elevado deve incluir, no mínimo, as seguintes informações: <ul style="list-style-type: none"> • ID da mensagem • Mensagem texto • Hora da ocorrência • Alarme aumentou por (pessoa ou processo nome) O monitor de alarme deve responder com um aviso (ou um erro indicando que a resposta tem sido incapaz de dar o alarme).
Status da interface warehouse	A interface entre o sistema e o warehouse(i2) devem fornecer a capacidade de verificar o estado operacional tanto da interface e do warehouse do próprio sistema.

Padrão de Requisitos - Tecnologia

Modelo

Resumo	Definição
<<Sumário de Tecnologia>>	<<descrição da tecnologia>> deve ser usada para <<utilização da tecnologia>>. (<<Versão da declaração da tecnologia>>) <<Motivação da declaração>>.
<<Sumário de Tecnologia>>	O sistema deve usar <<descrição da tecnologia>> para <<Utilização tecnológica>>(<<Versão da declaração da tecnologia>>) <<Motivação da declaração>>.

Exemplos

Resumo	Definição
Acesso do usuário via navegador da web	Todas funções do usuário serão acessíveis através de um navegador web.
Navegadores populares da web	<p>A interface de usuário deve ser baseada na web e todas as funções devem funcionar plenamente com todos os browsers populares da web /combinações de sistema operacional familiar (no máximo oito combinações), todos nomeados por uma pessoa designada (que se espera que venha a ser o gerenciador de marketing).</p> <p>Todas as versões de qualquer tipo de navegador da web que tenham sido a última versão, em qualquer momento nos últimos dois anos serão suportadas, com exceção de qualquer versão que foi ultrapassada dentro de um mês de sua liberação.</p> <p>A lista de navegadores populares usados / combinações de sistema operacional podem ser modificados periodicamente, mas não mais do que quatro vezes por ano. Quando um navegador / sistema operacional é adicionado à lista, o apoio deve fornecer um prazo de dois meses para se atualizarem.</p>
Navegador web Internet Explorer	A interface de usuário será baseada na web e todas as funções devem funcionar plenamente com o navegador web Microsoft Internet Explorer. Todas as versões que foram a versão mais recente, em qualquer momento nos últimos dois anos serão suportadas, com exceção de qualquer versão que foi ultrapassada dentro de um mês de sua liberação.
Sistema operacional utilizado	Os componentes do servidor do sistema são executados em um sistema operacional confiável e com boa escalabilidade.
Sistema Operacional Windows	O software deve ser escrito para rodar em hardware executando uma versão do sistema operacional Microsoft Windows.
Documentação em Word	Toda documentação que está escrito como documentos (por oposição às páginas da web) serão escrito usando o Microsoft Word.

Padrões de Requisitos - Aderência a Padrão

Modelos

Resumo	Definição
Obedecer ao padrão <<nome padrão>>	<p>O sistema deve obedecer (partes <<lista de partes padronizada>> da), <<descrição padrão>> (a fim de <<finalidades padrões>>).</p> <p><<versão da declaração padrão>>.</p> <p>Fonte: <<Norma localização>>.</p>

Exemplo (s)

Exemplos de uma Indústria específica:

Resumo	Definição
Privacidade HIPAA	<p>O sistema deve proteger a privacidade de todas as suas informações privadas, em conformidade com a regra da privacidade os E.U. de acordo com a Portabilidade e confiabilidade do Health Insurance (HIPAA), de 1996.</p> <p>Informação protegida sobre saúde é qualquer informação sobre um indivíduo que diz respeito à sua saúde (privacidade), a prestação de cuidados de saúde para eles ou para o pagamento dos cuidados de saúde.</p> <p>Fonte: http://www.hhs.gov/ocr/hipaa</p>

Exemplos exigindo o cumprimento das leis:

Resumo	Definição
SOX	<p>O sistema deve obedecer o E.U.A. Sarbanes-Oxley Act, de 2002 (comumente referido como SOX), durante sua gravação, protegendo de modificação imprópria, monitorando e acompanhando a confiabilidade de todas as atividades de dentro do sistema que têm conseqüências financeiras.</p> <p>O objetivo deste requisito é a exigência de ter a certeza de que a informação financeira consubstanciados no sistema reflete fielmente os negócios efetuados, bem como a fornecer meios pelos quais auditores podem verificar a exatidão de tais informações.</p> <p>Fonte: http://frwebgate.access.gpo.gov/cgi-bin/Getdoc.cgi?Dbname=107_cong_bills & docid = f:h3763enr.tst.pdf</p>

Exemplos de normas das empresas padronizadas:

Resumo	Definição
Normas de codificação da empresa	<p>Todos os softwares devem ser escritos de acordo com as normas da empresa de codificação para a linguagem de programação em que está escrito.</p> <p>Sempre que uma nova linguagem de programação é introduzida, um adequado padrão de codificação será adotado para ele.</p>
Orientações para empresa de estilo web	<p>Todas as páginas produzidas pelo sistema ou escrita em associação com ele devem estar em conformidade com as diretrizes da Empresa web.</p>

Exemplos de normas técnicas:

Resumo	Definição
ISO 639	Linguagens faladas serão identificadas internamente no sistema usando códigos definidos no ISO 639, norma internacional para linguagem natural. (Note-se que este requisito não se aplica a exibição de escolhas de idioma para os usuários – para o qual é preferível usar os nomes das linguagens). Fonte: http://www.loc.gov/standards/iso639-2/langhome.html
X.509	Certificados digitais utilizados para a autenticação do usuário devem cumprir X.509, o certificado para frameworks para a chave pública e atributo utilizam o padrão ITU-T. Note-se que este requisito não torna obrigatória a utilização de certificados digitais, mas apenas que, se forem utilizados, eles devem ser X.509-compliant. Fonte: http://www.itu.int/ITU-T/asn1/database/itu-t/x/x509/1997/

Padrões de Requisitos - Referência a Requisitos

Modelos

Resumo	Definição
Requisitos aplicáveis <<descrição do domínio>>	O sistema deve satisfazer <<os requisitos que se aplicam>> especificados na <<Especificação da versão>> da <<Especificação do nome>>, que reside em <<Especificação do local>>. <<Prioridade declaração>>.

Exemplos:

Resumo	Definição
Todos os requisitos comuns aplicáveis.	O sistema deve preencher todos os requisitos definidos na versão 2.0 da especificação dos requisitos comuns, que reside em x:\Specs\Common\CommonReqsV2.0.doc. Todos os requisitos referenciados têm prioridade 1.
Requisitos básicos de segurança aplicáveis.	O sistema deve satisfazer todas as características dos requisitos básicas de segurança, (SR1.1-SR1.11) especificados na versão 1,3 do "Especificação dos Requisitos de Segurança", que reside em x:\Specs\Security\SecurityReqsV1.3.doc.
Requisitos do controle de acesso aplicados.	O sistema deve satisfazer todos os requisitos do controle de acesso (SR2.1-SR2.9) especificados na versão 1.3 ao "Especificação de Requisitos de Segurança", que reside em x:\Specs\Security\AccessControlReqsV1.3.doc.

Padrões de Requisitos - Documentação

Resumo	Definição
<<Nome do tipo do documento>>	Deve haver um <<Nome do tipo do Documento>> que contém a <<Descrição do Documento>>. (Deve ser sob a forma de <<Formato de Documento>>.) (Este deve cumprir com o <<Padrão do nome do documento>>.) (Ele deve ser escrito em <<nome da linguagem>> linguagem.)

Exemplos:

Resumo	Definição
Ajuda Online	Deve haver ajuda on-line para cada função online do sistema. Ajuda on-line para as funções disponíveis para clientes que devem ser destinadas a eles; ajuda on-line para outras funções devem ser destinadas a usuários internos. Para cada função, a ajuda on-line deve descrever como utilizar essa função, de tal forma que um usuário com pouca experiência deve ser capaz de usá-la como pretende.
Guia do usuário	Deve haver um guia de usuário para todas as funções disponíveis para os usuários internos. A ajuda on-line pode ser a base para satisfazer este requisito, desde que possa ser consolidada em um formulário claro e que possa ser impresso.
Explicações da mensagem de erro	Deve haver um conjunto de explicações das mensagens de erro. Deve conter uma explicação sobre o significado de cada mensagem de erro que satisfaça qualquer um dos seguintes critérios: <ul style="list-style-type: none"> a. Um usuário inexperiente que não consegue deduzir o significado completo. b. Indicação de erro para um problema grave. c. Não há mais o que explicar sobre o erro. Quando necessária cada explicação deve também descrever a forma de corrigir ou outra maneira de responder à mensagem, e (se possível) explicar como o componente do sistema originou o erro.

PADRÕES DE REQUISITOS AS INFORMAÇÕES

Padrões de Requisitos - Tipos de Dados

Modelos

Resumo	Definição
Tipo de Dado <<Nome do tipo de dado>>	<<Nome do tipo de dado>>, que são usados para <<Finalidade do Tipo de Dado>>, deve ser do formulário <<Formulário Tipo de Dado>>. (<<Declaração (descrição) do formato da exibição>>).(<<Restrições das declarações>>). (<<Manipulação especial da declaração>>.)

Exemplos

Resumo	Definição
Tamanho do endereço do email	O sistema permitirá endereços de e-mail de até 60 caracteres.
Formulário do número do telefone	Os números de telefones devem ter a seguinte forma: AA-LLLL-NNNN xEEEE Onde AA é o código da área LLLL é a localização NNNN é o número individual e EEEE é a extensão.

Padrões de Requisitos - Estrutura do Dado

Modelos

Resumo	Definição
<<Nome da estrutura de dados>>	<<Descrição do tipo de dado>> deverá ter os seguintes itens de informação: <ul style="list-style-type: none"> • <<descrição do dado do item 1>> • ...

Exemplos

Resumo	Definição
Detalhes do nome pessoal	Os detalhes do nome pessoal deverão mostrar os seguintes itens de informação: <ul style="list-style-type: none"> • Nome dado • Segundo nome • Sobrenome • Iniciais • Título
Detalhes do contato pessoal	Os detalhes do contato para uma pessoa deverão conter os seguintes itens de informação: <ul style="list-style-type: none"> • Detalhes do nome pessoal (como definido no item anterior) • Endereço • Número do telefone do trabalho • Número do telefone residencial • Número do telefone celular • Número do fax • Número do Pager (usados somente para empregados) • Endereço de email

Padrões de Requisitos - Identificação

Modelos

Resumo	Definição
<<Nome do dono da entidade>> (<<nome ID>>) ID	Cada "Nome do dono da entidade" deve ter um ID único que está sob a forma de <<forma ID>> atribuído pelo <<Como atribuir>>. (<<Mostrar formato>>) (Cada <<Nome ID>> deve ser único <<escopo da singularidade>>.) (<<Classificar o pedido da declaração".) (<<Reutilização das condições de declaração".)

Exemplos

Resumo	Definição
Número do cliente, com verificação do dígito.	Cada cliente deve ter uma identificação única por um ID do cliente que é sob a forma de um número sequencialmente acrescido no dígito de verificação que é verificado utilizando o <<Nome do algoritmo>> algoritmo (como explicado em <<algoritmo local>>).
Pedido ID	Cada pedido deve ter identificação única por um ID que está na forma de um número do cliente que será acrescido de um número sequencialmente atribuído para esse cliente, iniciando em um para o primeiro cliente. Os IDs dos pedidos serão apresentados na seguinte forma “<<Número do cliente>>-<<Número do Pedido>>”. "10762-1").
ID do funcionário	Cada funcionário deve ter um ID funcionário que tenha um número de cinco algarismos atribuído externamente (e inseridos manualmente quando os detalhes do funcionário são inseridos primeiro). Cada ID do funcionário deve ser único dentro do âmbito de aplicação do sistema, mesmo que um trabalhador se afasta o seu ID do empregado não devem ser reutilizados para outro trabalhador.

Padrões de Requisitos - Fórmula de Cálculo

Modelos

Resumo	Definição
Cálculo do <<Valor do nome>>	<<Descrição do valor>> deve ser calculado da seguinte forma: <<Valor do Nome> = <<fórmula>> Onde <<Nome da variável 1>> é <<Descrição da variável 1>>; <<Nome da variável 2>> é <<Descrição da variável 2>>; ... (<<Aperfeiçoamento do Cálculo>>.) (<<Limitações de aplicabilidade>>.) (<<Referencias>>.) (Por exemplo, <<Exemplo>>.)
Determinação do <<Valor do nome>>	<<Descrição do valor>> deverá ser determinada como mostra a seguir: 1. <<Descrição do passo 1>>. 2. <<Descrição do passo 2>>. ... (<<Limitações de aplicabilidade>>.) (<<Referencias>>.) (Por exemplo, <<Exemplo>>.)

Exemplos

Resumo	Definição
Cálculo principal para um novo depósito	O saldo principal de um depósito é calculado com o saldo velho principal acrescido dos juros acumulados desde o último juro que foi pago.
Cálculo de juros simples	Juro simples durante um período não superior a um ano deve ser calculado do seguinte modo: $\text{Juros} = \text{taxa de juro principal} * \text{período em dias/dias no ano} * 100$ Onde: Principal é o montante monetário com base no juro que foi ganho; Taxa de juro é a taxa aplicável; Prazo em dias é o número de dias no valor dos juros, calculados de acordo com a seguinte exigência; Dias do ano é o número de dias do ano para o cálculo a ser executado, calculado como por o próximo requisito mais um.

Padrões de Requisitos - Longevidade do Dado

Modelo

Resumo	Definição
Armazenar <<Descrição do dado>> <<Forma de armazenamento>> para <<Duração>>	<<Descrição do dado>> deverá ser armazenada <<Forma de armazenamento>> para <<reter a duração>> do <<início da armazenagem>>. (Quando os dados são ilegíveis para a remoção, <<ação de deadline>>.)

Exemplos

Resumo	Definição
Ordem de armazenamento online por 90 dias	Ordem de armazenamento online deverá manter-se por durante 90 dias a contar da data em que o pedido foi enviado. Os pedidos não serão visíveis para os clientes após esta data, mas eles podem continuar a ser armazenados online. O objetivo é para que os detalhes de um pedido estejam facilmente disponíveis em caso de problema com a entrega.

Padrões de Requisitos - Arquivamento do Dado

Resumo	Definição
Arquivamento <<síntese de dados>>	<<Descrição dos dados>> devem ser (movida)/(copiada) de <<dados de origem>> para <<dados de destino>> <<frequência>>. <<Descrição inicial>>. (O objetivo é para <<arquivamento do objetivo>>.)

Exemplos

Resumo	Definição
Armazenamento do pedido do cliente	<p>O pedido do cliente e todos os detalhes pertinentes de cada pedido devem ser ilegíveis para ser transferido para um meio de armazenamento off-line, número de dias configurável (prevê-se que seja da ordem de 90 dias) após ser transferido totalmente, na próxima vez, alterar a ordem de arquivamento do processo para ser executado posteriormente. Os resultados deverão ser mantidos indefinidamente em mídia de armazenamento off-line.</p> <p>Este requisito não faz qualquer correção sobre ordem de arquivamento do processo quando é iniciada, que pode ser manual ou automático.</p> <p>O objetivo deste requisito é reduzir a quantidade de dados transacionais, no sistema on-line (para melhorar o desempenho) e para reduzir o impacto negativo do acesso não autorizado aos dados on-line.</p>

PADRÕES DE REQUISITOS DA ENTIDADE DE DADOS**Padrões de Requisitos - Entidade Ativa**

Modelos

Resumo	Definição
<< Nome da Entidade >>	<p>O sistema deve armazenar as seguintes informações sobre o <<Nome da Entidade>>:</p> <ul style="list-style-type: none"> • "Dados item 1 descrição". • ... <p>Uma "Entidade nome" é "Entidade explicação". Cada "Entidade nome" é exclusivamente identificada por "identificador Entidade (s)". (<<Detalhes das entidades geradas>>.)</p>

Exemplo(s)

Resumo	Definição
Informações básicas sobre os clientes	<p>O sistema deve armazenar as seguintes informações sobre o cliente:</p> <ul style="list-style-type: none"> • ID do cliente (como definido no requisito XR99.1). • Senha. • Informações de contatos pessoais (como definido no requisito XR99.2). • Informações de cartão de crédito (tal como definido no requisito XR99.3). • Data de nascimento. • Data de inscrição. • Status (ativo, bloqueado ou encerrado). Isso nunca é mostrado ao cliente. <p>Cada cliente é identificado unicamente pelo seu ID de cliente.</p>

Padrões de Requisitos - Transações

Modelo

Resumo	Definição
<<Nome da Transação>>	<p>Deve haver uma função de criar uma <<Nome da Transação>> transação para um <<Dono do nome da entidade ativa>>. Cada <<Nome da Transação>> deve conter as seguintes informações:</p> <ul style="list-style-type: none"> • <<Dados item 1 descrição>>. • ... <p>Um <<Nome da Transação>> é <<Explicação da Transação>>. Cada <<Nome da Transação>> é exclusivamente identificada por um <<Identificador(s) da Transação>>.</p> <p>Um <<Nome da Transação>> considera-se que aconteceu <<Transação acontece em tempo descrição>>. (<<Declaração da longevidade da transação>>).</p>

Exemplo

Resumo	Definição
Conta ajuste	<p>Deve ser possível postar um ajuste (débito ou crédito) para a conta de um cliente selecionado. Um ajuste deve conter as seguintes informações:</p> <ul style="list-style-type: none"> • ID do cliente • Quantia do ajuste • Razão do ajuste – Texto livre destinado a explicar as razões pelas quais o ajuste foi levantado. <p>Um único ID é automaticamente atribuído a cada conta do ajuste. Espera-se que a autoridade para utilizar este requisito seja restrito a poucos funcionários.</p>

Padrões de Requisitos - Configuração

Modelos

Resumo	Definição
<<Nome do valor da configuração>>	Deve ser possível especificar << Nome do valor da configuração >> com o objetivo de << do valor propósito>>. Trata-se de um valor <<descrição do tipo de dados>> (por exemplo, <<valor representante>>) e é a <<descrição do nível de configuração>>. (Este valor somente pode ser modificado <<Mudança da descrição do tempo>>.)
<<Configuração do nome da entidade>>	O sistema deve armazenar as seguintes informações sobre uma <<Configuração do nome da entidade>> com o objetivo de <<Objetivo da Entidade>>: <ul style="list-style-type: none"> • <<Configuração da descrição do valor 1>>. •... Cada <<Configuração do nome da entidade>> é exclusivamente identificada pelo <<Identificados da Entidade>>. (Essa entidade pode ser modificada apenas <<Mudança da descrição do tempo>>.)

Exemplos

Resumo	Definição
Moeda local	Deve ser possível designar o sistema da moeda local. Este valor não pode ser alterado depois que o sistema estiver ativo.
Limite de retirada de dinheiro	Deve ser possível especificar o limite de saque de dinheiro, o que é um valor monetário expresso em moeda local que determina o valor máximo de uma operação de retirada em dinheiro que um usuário pode executar. Esse valor pode ser configurado para: (1) um trabalhador; (2) um empregado; (3) um departamento; e (4) um sistema padrão, de tal modo que se não especificar em um destes níveis a que nível seguinte deve ser utilizado. É facultativa a todos os níveis exceto a um sistema amplo. E pode ser mudada a qualquer momento em qualquer nível.

Padrões de Requisitos - Cronologia

Modelo

Resumo	Definição
Registro <<Resumo tipo de ocorrência>>	<p>Cada <<descrição do tipo ocorrência>> serão automaticamente gravado. Para cada uma, serão registrados os seguintes:</p> <ul style="list-style-type: none"> • Data e hora • <<Ocorrência detalhe 1>> • <<Ocorrência detalhe 2>> • ... <p>(Cada caso deve ser tratado como tendo uma gravidade de <<descrição da gravidade>>.)</p>

Exemplos

Resumo	Definição
Armazenar toda ordem dos eventos	<p>A cada evento que ocorre durante a vida do sistema devem ser armazenados na ordem e de forma permanente. Para cada caso, no mínimo, as seguintes informações devem ser registradas:</p> <ul style="list-style-type: none"> • Tipo de evento • Detalhes da mudança (que irá variar de um tipo de evento para o outro) • Data e hora da mudança • ID do usuário da pessoa que fez a mudança (se pertinente) • ID do usuário da pessoa que aprovou a mudança (se pertinente) <p>Note que não há nenhuma implicação que todos os eventos são armazenados no mesmo caminho ou no mesmo lugar.</p>
Registro do acesso aos dados	<p>Cada acesso de um usuário com sucesso, para um item de dados devem ser registrados.</p> <p>A informação para ser registrada deve incluir a data e hora do acesso, o ID do usuário, e os detalhes dos dados que foi solicitado.</p> <p>(Note que acessos sem sucessos são coberto pelo próximo requisito, e não são registradas novamente sob os auspícios deste requisito -para evitar a gravação do mesmo mais de duas vezes.)</p>

PADRÕES DE REQUISITOS PARA FUNÇÃO DO USUÁRIO

Padrões de Requisitos - Consulta

Exemplos

Resumo	Definição
Consulta <<nome da consulta>>	<p>Deve haver uma consulta (<<nome da consulta">) que revela <<informação para mostrar>>. O seu objetivo é <<intenção de negócio>>.</p> <p>Para cada <<Nome da entidade>>, a consulta deve mostrar o seguinte:</p> <ul style="list-style-type: none"> • <<Informação item 1>> • <<Informação item 2>> • ... <p>(Os itens a serem exibidos serão listados em seqüência <<Classificar seqüência de detalhes>>)</p> <p>(Os itens a serem exibidos podem ser especificados por entrar qualquer um dos seguintes critérios de seleção:</p> <ul style="list-style-type: none"> • <<Critério de seleção 1>> • <<Critério de seleção 2>> • ...) <p>(O usuário será capaz de navegar <<Detalhes de navegação do usuário>>.) (O usuário deve ser capaz de interagir com a consulta <<Detalhes da interação do usuário>>.)</p> <p>(As informações mostradas são atualizadas automaticamente <<atualização automática dos detalhes>>.)</p>

Exemplos

Resumo	Definição
Consulta da transação financeira	<p>Deve haver uma consulta que mostra as transações financeiras de um cliente selecionado em um determinado período. Assim, deve apresentar uma linha de detalhes para cada operação financeira feita pelo cliente, no período de tempo, em ordem cronológica inversa. Cada linha de detalhes deverá mostrar:</p> <ul style="list-style-type: none"> • Data e hora da transação • Descrição da transação • Número de transações <p>A consulta deve também mostrar o número total de transações efetuadas pelos clientes no período. O usuário deve poder escolher qualquer operação mostrada e ver todos os detalhes.</p>
Consulta atualizada geral	<p>O grande objetivo desta consulta é exibir novamente o próprio usuário, sem a ação do usuário de atualizar os valores, para manter atualidade o seu conteúdo. Este requisito não tem uma regra como dever ocorrer (em particular, se ela reexibir periodicamente ou se os valores devem ser transmitidos sempre que ele mudar), nem definir o que entende por “atualizar”.</p> <p>Se a consulta for atualizada periodicamente, em vez de dinamicamente mantendo-se a mesma data:</p> <ul style="list-style-type: none"> • A taxa de atualização deve ser configurável (espera-se um tempo da ordem de 30 segundos.) • O usuário deve ter a possibilidade de solicitar de imediato uma atualização manualmente.

Padrões de Requisitos - Relatório

Modelos

Resumo	Definição
Relatório <<nome do relatório>>	<p>Deve haver um relatório que mostra <<mostrar informação >> <<Critério de Seleção>> ordenadas por <<Seqüência de classificação>> O objetivo do presente relatório é a <<Intenção de Negócios>>.</p> <p>Para cada <<Nome tipo de item>>, o relatório deve mostrar o seguinte:</p> <ul style="list-style-type: none"> • <<Nome Valor 1>> • << Nome Valor 2>> • ... <p>(Os itens a serem exibidos podem ser especificados por entrar em qualquer um dos seguintes critérios de seleção:</p> <ul style="list-style-type: none"> • <<critério de seleção 1>> • <<critério de seleção 2>> • ...) <p>Totais devem ser indicados para <<totalizando níveis>>. (Uma nova página deve ser iniciada por << níveis de páginas>>.)</p> <p>(O relatório destina-se a ser executado automaticamente <<Detalhes executados automaticamente>>.)</p>

Exemplos

Resumo	Definição
Relatório cambial	<p>Deve haver um relatório que lista todos os negócios feitos no câmbio de um período selecionado (com a secretária para quem trabalha o requerente).</p> <p>O relatório deve ostentar as seguintes informações para cada moeda estrangeira:</p> <ul style="list-style-type: none"> • Tipo de negócio • Montante do negócio (incluindo a moeda) • Data e hora • Taxa de câmbio • Nome da organização com a qual a operação foi colocada. <p>Ao final do relatório deve mostrar totais conforme a organização e o total geral para o número de negócios, e o montante do negócio.</p>

Padrões de Requisitos - Acessibilidade

Discussão

Resumo	Definição
Seção 508 da Lei de Reabilitação	O sistema deve ser acessível às pessoas com deficiência, de acordo com a Seção 508 da Lei de Reabilitação os U.S.A., conforme alterada, 1998--comumente designado por "Seção 508." (Seção 508 aplica-se a todos os sistemas e sites de desenvolvidos web, comprados ou executados por qualquer órgão do governo federal dos U.S., apesar de existirem algumas exceções.) Fonte: http://www.section508.govb
Lei de Deficiência dos norte-americanos	O sistema deve ser acessível às pessoas com deficiência, de acordo com a lei de Deficiência dos norte-americanos, 1990. Fonte: http://www.usdoj.gov/crt/ada/statute.html
Lei da discriminação dos deficientes no Reino Unido	O sistema deve ser acessível às pessoas com deficiência, de acordo com a Lei de discriminação dos deficientes do Reino Un de 1995. Fonte: http://www.opsi.gov.uk/acts/acts1995/1995050.htm
Guia com o conteúdo de acessibilidades na web	Todas as páginas da web exibidas pelo sistema (incluindo todas as páginas de documentação estática) devem ter o guia com o conteúdo de acessibilidade da web, versão 1,0, produzido pelo World Wide Web Consortium. Fonte: http://www.w3.org/TR/WAI-WEBCONTENT

Modelo

Resumo	Definição
Acessíveis a pessoas com necessidades especiais <<nome da necessidade especial>>	Uma <<parte do sistema>> devem ser acessíveis a pessoas com necessidades especiais <<descrição da necessidade especial>>(dependendo do <<grau do suporte>>) (Estima-se que <<Porcentagem afetada>> % dos usuários são susceptíveis para os beneficiários.) (Esta pertence à <<cláusula da lei>>.)

REQUISITOS PADRÕES DE PERFORMANCE

Requisitos Padrões - Tempo de Resposta

Modelo

Resumo	Definição
Tempo de resposta <<tipo de operação>>	Cada <<tipo de operação>> dever ter um tempo de resposta de no máximo <<período de tempo tolerável>> de <<início do limite do tempo>> para << fim do limite do tempo>> (quando se usa <<indicativo de configuração do hardware>>. Este valor é baseado em <<Justificativa>>. (Este requisito não se aplica aos <<casos excepcionais>>.)

Exemplo(s)

Aqui estão alguns exemplos de requisitos de tempo de resposta com metas quantitativas:

Resumo	Definição
Consulta do tempo de resposta	Qualquer consulta deve completar a exibição dos seus resultados, a partir do momento em que o usuário apresenta o pedido, o mais tardar no tempo de 4 segundo para a exibição de uma página de referência simples no mesmo local. Este valor é baseado em indicativos de testes anedóticos que os usuários começam a perder a paciência logo após esse tempo. Este requisito não se aplica as consultas em grandes volumes de dados onde critérios de seleção arbitrários são permitidos.
Transação de comutação do tempo	O tempo médio para a realização da operação para mudar de rota de um pedido do cliente para um serviço deve ser inferior a 300 milissegundos. (Este valor foi calculado como um vigésimo do tempo em que um cliente aceitaria para esperar uma típica operação. Um aceitável tempo por uma típica operação é tido como 6 segundo baseada em valores relativos a temporização das transações EFTPOS.)

Aqui estão alguns exemplos dos requisitos de tempo de resposta que não foram mencionadas nas metas quantitativas:

Resumo	Definição
Tempo de resposta do usuário nunca deve exceder	Nenhuma função usuário deve ter um tempo de resposta média no sistema operacional normal que para o usuário seria razoável considerar excessivo para esse tipo de função.
Tempo de emissão do cartão de identificação	Deve ser possível emitir um cartão de identificação do trabalhador num tempo suficientemente rápido após o pedido enquanto se aguarda a entrega. Dez minutos são considerados como aceitável o tempo de esperar para os efeitos desta exigência, apesar disto não deve ser tratada como uma barreira entre bom e mau. Se a entrega for mais rápida aumentará a satisfação do usuário com o sistema, e um longo tempo diminuirá a satisfação.

Padrões de Requisitos - Rendimento

Modelo(s)

Resumo	Definição
Taxa <<tipo de vazão>>	<<Parte do sistema>> devem ser capazes de lidar com o <<tipo de objetivo da vazão>> operações a uma taxa de pelo menos <<quantidade de throughput>> por <<Unidade de período de tempo>> (quando se usa <<indicativo de configuração hardware>>).

Exemplo

Resumo	Definição
Ordem da taxa de inscrição	O sistema inicial deve ser capaz de lidar com a entrada de ordens por parte dos clientes a uma taxa de pelo menos 10 por segundo. Contingência não tenha sido adicionado; esta taxa representa a real demanda esperada. Veja o sistema de modelo de calibragem para obter detalhes de como se chegou a este valor. Situa-se na <<tamanho do modelo local>>.

Padrões de Requisitos - Capacidade Dinâmica

Exemplos

Resumo	Definição
Capacidade simultânea <<tipo de entidade>>	O sistema deve ser capaz satisfazer <<contador de entidade>> simultânea ao <<tipo de entidade>> <<declaração condição da entidade>> (<<declaração da duração do pico>>). (<<declaração do tempo de execução>>.) (<<declaração do período de pico da concessão >>).

Exemplos

Resumo	Definição
Capacidade simultânea de cliente	O sistema deve acomodar 100 clientes ativos e autenticados simultaneamente. Um usuário é considerado como ativo no caso de terem apresentado um pedido para o sistema nos últimos cinco minutos.
Capacidade máxima de cliente	O sistema deve acomodar 200 clientes ativos e autenticados simultaneamente quando, bilhetes para um concerto popular ir a venda meia-hora antes da hora publicada venda até duas horas depois. A definição de clientes ativos é concedida na condição anterior. Durante um concerto popular primeira venda máxima, é aceitável para serviços secundários oferecidos pelo site (incluindo os que envolvem grandes downloads ou o streaming de áudio ou vídeo) para ser encerrada. Também é aceitável impedir usuários internos de acessar quaisquer funções que envolvem tratamento intensivo.

Padrões de Requisitos - Capacidade Estática

Modelo

Resumo	Definição
Capacidade total do <<tipo de entidade>>	O sistema deve ser capaz de lidar com um mínimo de <<contador de entidade>> <<tipo de entidade>>. <<critérios de inclusão de entidade>> <<declaração do realização do calendário>>.)

Exemplo(s)

Resumo	Definição
Capacidade inicial dos clientes	O sistema deve ser capaz de lidar com um mínimo de 5.0000 clientes após a instalação inicial.
Eventual capacidade do cliente	O sistema deve eventualmente ser capaz de lidar com um mínimo de 1.000.000 de clientes. Este valor abrange apenas os clientes que tenham acessado o site nos últimos três meses ou colocado dentro de uma ordem dos últimos doze meses. Não se espera que esse nível de negócio seja atingido antes de dois anos após a aplicação inicial.

Padrões de Requisitos - Disponibilidade

Modelos

Resumo	Definição
Disponibilidade <<extensão>>	O sistema deve estar normalmente disponíveis para os usuários <<descrição da disponibilidade da extensão>> (salvo em circunstâncias excepcionais de uma frequência e duração não superior ao <<tolerado inatividade qualificativo>>). “Normalmente disponível” deve ser entendida como <<significado de disponibilidade>>.

Exemplos

Resumo	Definição
Disponibilidade 7am ate 7pm	O sistema deve estar disponível a todos os usuários das 7 da manhã às 7 pm em dias úteis (isto é, durante a semana que não tem feriado), salvo em circunstâncias excepcionais de uma frequência e duração de forma a não exceder os definidos em outros requisitos. “Disponíveis” entende-se que todos os usuários são funções operacionais.
Disponibilidade das funções dinâmicas da web	As funções dinâmica da web site da companhia devem estar disponíveis aos visitantes 24 horas por dia, todos os dias do ano, exceto para a imprevistos não superior a 1 hora por semana (médias de cada trimestre), acrescido de paralisação programada de forma a não exceder uma falta por mês por um período máximo de 4 horas para ser realizado no momento de uma semana de baixa atividade no site. “Funções dinâmicas” são aqueles que exigem a participação ativa do sistema web shop (por exemplo, para o local ou consulta sobre as encomendas).

REQUISITOS PADRÕES DE FLEXIBILIDADE

Padrões de Requisitos - Escalabilidade

Modelo

Resumo	Definição
Escalabilidade <<aspectos da escalabilidade>>	O sistema deve ser escalável para acomodar um número ilimitado de <<Tipo de item a ser escalável>> <<Indicativo de alto volume negócios>>. (<<A facilidade de expansão da declaração>>.) (<<Motivação declaração>>.)

Exemplo(s)

Resumo	Definição
Escalabilidade do cliente	O sistema deve ser escalável para acomodar o crescimento do número ilimitado de clientes (prospectivamente a várias centenas de milhares). A motivação para este requisito é a de servir para o crescimento futuro do negócio.
Escalabilidade da distribuição do escritório	O sistema deve ser escalável para permitir a sua utilização por um número ilimitado de escritórios da companhia distribuídos. (Eventualmente poderá haver mais de uma centena de escritórios.) A motivação para este requisito é permitir que o sistema possa ser utilizado em toda a empresa e em todas suas filiais (incluindo as futuras aquisições).

Padrões de Requisitos - Extensibilidade

Modelos

Resumo	Definição
Extensibilidade <<Aspecto do sistema>>	Deve ser possível expandir o <<aspecto do sistema>>, através do desenvolvimento e <<ligação com>> um módulo de software extra. A introdução de qualquer tipo de módulo não deve exigir mudanças fundamentais para o software do sistema para permitir a sua introdução. <<Facilidade de extensibilidade>>. (<<Detalhes de Configuração>>.)
Agrupar o <<tipo de subcomponente>>	Deve ser possível adicionar um novo <<tipo de subcomponente>> através do desenvolvimento da <<ligação com>> o software necessário para apoiá-lo. Um novo <<tipo de subcomponente>> não deve exigir mudanças fundamentais para o software do sistema para permitir a sua introdução. <<Facilidade de extensibilidade>>. (<<Detalhes de Configuração>>.)

Exemplos

Resumo	Definição
Agrupar novos métodos de notificação	Deve ser possível adicionar um novo método de notificação de usuários através do desenvolvimento e da <<ligação com>> o software necessário para apoiar esse método. Um novo método de notificação não deve exigir mudanças para o software do sistema para permitir a sua introdução.
Agrupar novos métodos de pagamento	Deve ser possível adicionar um novo método pelo qual os clientes podem pagar pelo desenvolvimento e <<ligação com>> o software necessário para suportar esse método. Um novo método de pagamento não deve exigir mudanças para o software do sistema para permitir a sua introdução.

Padrões de Requisitos - Ilimitados

Exemplos

Resumo	Definição
Não é especificado um <<único nome do ambiente>>	O sistema deve ser adequado para ser utilizado por qualquer organização que tenha <<Condições adequadas>>. <<Declaração da motivação>>. Por exemplo, <<Aspectos da variação da declaração (ões)>>.

Exemplos

Resumo	Definição
Não é específico para Acme	O sistema deve ser adequado para ser utilizado por qualquer empresa do mesmo ramo de negócio, seguindo as mesmas práticas empresariais e residem no mesmo estado em Acme Corporation. A motivação é permitir a venda deste sistema para empresas do mesmo ramo, embora ele deve ser reconhecido que o sistema é principalmente para uso interno pelo Acme Corporation. Por exemplo, o nome de "Acme Corporation Inc." e um texto com valores semelhantes não devem ter o código fonte em qualquer lugar.
Apropriado para escritórios no mundo inteiro da Acme	O sistema deve ser adequado para ser utilizado por qualquer escritório da Acme Corporation em qualquer país em que opera Acme. Assume-se que todos os usuários do sistema falam Inglês, de forma que tradução em outros idiomas da interface do usuário do sistema, relatórios, documentação e outros materiais não é necessário. Acme tem escritórios na América do Norte e Sul, Europa e Ásia.

Padrões de Requisitos - Múltiplos

Modelo

Resumo	Definição
Multi <<Nome-Multi>>	Especificar que um sistema deve acomodar múltiplas empresas e moedas ao mesmo tempo O sistema deve suportar diversos <<Descrição tipo de múltiplos>>. <<Declaração de extensão>>. (<<Número de casos declaração>>.) (<<Limitações da declaração>>.)
Prestação de ser multi-<<nome multiness>> (no futuro)	O sistema deve prever a futura introdução de suporte para múltiplos <<Multiness tipo descrição>>. <<Declaração de extensão>>. (<<Limitações declaração>>.)

Padrões de Requisitos - Multi-Linguagem

Modelo

Resumo	Definição
Multi-linguagem<<qualificador do escopo>>	O sistema deve suportar múltiplas linguagens. <<Declaração de medida>>. [<<Número de casos declarados>>] [<<Limitação das declarações>>]
Prestação [para <<qualificador do escopo>>] para se tornar multi-linguagem	O sistema deve ser disponível para introdução de suporte a múltiplas linguagens futuramente. << declaração de medida >>. [<< Limitações das declarações >>]

Exemplos

Resumo	Definição
Interface multi-linguagem do usuário cliente	O sistema deve permitir a interface do usuário cliente ser disponível em múltiplas linguagens. Cada usuário deve nomear qual a faixa de línguas suportadas eles desejam utilizar, e cada parte da informação mostrada para eles pelo sistema deve estar naquela língua. Isto inclui telas <i>prompt</i> , mensagens, gráficos, áudio e vídeo que contenha conteúdo de linguagem específica. É antecipado que não será suportado mais de três línguas. Línguas que utilizam um <i>multi-byte character set need</i> não serão suportadas. Este requisito não se aplica aos usuários de interface de funções usadas apenas pelos empregados (que é necessária apenas em uma língua).
Disposição para ser multi-linguagem	O sistema deve ter a disposição para suportar múltiplas linguagens futuramente. A disposição deve incluir pelo menos o seguinte: <ol style="list-style-type: none"> 1. A estrutura do banco de dados deve suportar multi-linguagem de tal forma que não seja necessário adicionar novas colunas a tabelas ou a reposição de qualquer tabela por uma ou mais tabelas. 2. Um usuário deve ter a permissão de nomear sua linguagem preferida quando acessar seus detalhes pessoais.

Padrões de Requisitos - Instalabilidade

Modelo

Resumo	Definição
<< Parte do sistema >> instalabilidade	Deve ser possível para << parte do sistema >> ser instalada por << pessoa que irá instalar >>. << Declaração de fácil instalação >>. [<< Declaração de instalação média >>.]

Exemplos

Resumo	Definição
Instalabilidade da aplicação do cliente	Deve ser possível para a aplicação da loja <i>web</i> do cliente ser instalada através de um cliente que não possuía nenhum conhecimento especial. O sistema de instalação deve ser conveniente e envolvendo pouca informação de entrada pelo usuário. A aplicação do cliente deve ser baixada através de um <i>web</i> site do serviço.
Instalabilidade do sistema principal	Deve ser possível para o software do sistema principal (servidor) ser instalado através de um componente de sistema administrativo que não tenha conhecimento anterior do sistema ou através de terceiros que utilizarem o produto (mas com quem é familiar com o sistema operacional da máquina que deve ser instalada). O <i>software</i> deve ser instalado através de um aparelho de armazenamento popular de capacidade média (como CD).

PADRÕES DE REQUISITOS PARA CONTROLE DE ACESSO

Padrões de Requisitos - Registro do Usuário

Modelo

Resumo	Definição
Auto-registro <<classe usuário>>	Uma pessoa deve ser auto-registar-se como uma <<classe usuário>>, por <<descrição do processo de inscrição>>. Devem ser solicitado a digitar as seguintes informações pessoais: <ul style="list-style-type: none"> • <<Detalhe do Usuário 1>> • <<Detalhe do Usuário 2>>...
Registro da <<classe usuário>>	Deve ser possível registrar uma pessoa como uma <<classe usuário>>, pela <<descrição do processo de inscrição>>. As seguintes informações devem ser inscritas sobre eles: <ul style="list-style-type: none"> • <<Detalhe do Usuário 1>> • <<Detalhe do Usuário 2>>...

Exemplos

Resumo	Definição
Registro do cliente	<p>Um visitante do site da web deve ser capaz de auto-registar-se como um cliente, ao entrar os seguintes detalhes:</p> <ul style="list-style-type: none"> • Cadastrar o ID do usuário * • Cadastrar a senha (que deve ser inscrita duas vezes) * • Detalhes do nome, como descrito no requisito <<Req't ID>> * • Endereço * • Endereço de e-mail * • Número de telefone • Sexo • Idioma preferido <p>* Todos os itens sinalizados com um asterisco são obrigatórios.</p>
Registro do empregado	<p>Deve ser possível um usuário autorizado se registrar com uma pessoa empregada, e deve cadastrar os seguintes detalhes:</p> <ul style="list-style-type: none"> • Nome completo (primeiro, meio e apelidos) • Diminutivo do nome (por exemplo, "Bill") • Título do Trabalho • Companhia endereço de e-mail • Companhia telefônica - número e extensão • Endereço residencial • Telefone residencial* • Número do telefone celular * • Sexo • Cargo pretendido <p>Toda esta informação é obrigatória, exceto os assinalados com um asterisco (*).</p> <p>O empregado deve ser automaticamente alocado no próximo número disponível do empregado, que será exibido para o usuário.</p> <p>Uma senha inicial é gerada para os novos empregados baseado nos conhecimentos sobre eles (incluindo o seu número empregado). A forma desta senha deve ser suficientemente simples para facilitar a comunicação do trabalhador sem divulgar o seu valor real, apesar de uma forma específica real não está mandada por esta exigência. (Por exemplo, poderia ser “sobrenome acrescido de nome.”) Ao efetuar o login pela primeira vez, o empregado deve ser forçado a mudar a sua senha.</p>

Padrões de Requisitos - Autenticação do Usuário

Modelo

Resumo	Definição
Autenticação/login <<classe usuário>>	A <<classe usuário>> deve ser capaz de auto-autenticar (login) pelas <<Etapas de autenticação>>. (<<Iniciado por descrição>>.)

Exemplos

Resumo	Definição
Autenticação do usuário	<p>Um usuário deve ser capaz de auto-autenticar (login), e deve fazê-lo antes que eles possam acessar qualquer função ou informações que não sejam publicamente ou anonimamente acessadas.</p> <p>O nível de segurança oferecido pelo mecanismo usado para autenticar um usuário específico (ou classe de usuários) deve ser adequada ao grau de sensibilidade e de acesso a eles (ou seja, o montante dos prejuízos que poderia infligir um mal impostor). É aceitável a utilização de diferentes mecanismos de login para diferentes classes dos usuários.</p> <p>Os clientes e funcionários devem ser mantidos seguros na medida que um cliente não será capaz de efetuar o login como um empregado entrando em um ID do empregado e senha do usuário em um cliente conectado.</p>
Autenticação do cliente	<p>Um cliente deve ser capaz de auto-autenticar (login), digitando seu ID de usuário e senha. Eles podem escolher fazer login em qualquer momento, mas que não tivessem registrado, quando eles tentam uma ação para a qual a sua identidade deve ser conhecida, estas deve ser solicitado a efetuar login e não permitir que prossiga com a ação até que ele o faça.</p> <p>A identidade de cada cliente deve ser determinada antes que eles podem dar início ou visualizar transações. Isso será atingido pelo cliente digitando seu ID de usuário e senha.</p>

Padrões de Requisitos - Autorização Configurável

Apresentação e Usabilidade

Resumo	Definição
Ocultar funções inacessíveis	Locais de onde as funções que são normalmente selecionados (por exemplo, nos menus e botões ou hiperlinks) não devem exibir funções as quais o atual usuário não tem acesso.
Desativar o controle de funções inacessíveis	Locais de onde funções são normalmente selecionados (por exemplo, nos menus e nos botões ou hiperlinks), serão exibidos em uma forma desativada, funções essa para o qual o atual usuário não tem acesso.

Delegando autoridade

Resumo	Definição
Delegar autoridade	<p>O empregado deve ter a possibilidade de conceder a outro funcionário competente para agir em seu nome.</p> <p>O delegado deve ser capazes de nomear privilégios que o delegado pode exercer em seu nome, incluindo a fixação de limites relacionados com o acesso.</p> <p>O objetivo deste requisito é de permitir que um assistente pessoal possa desempenhar tarefas administrativas em substituição do gerente para a qual trabalha.</p>
Assumir autoridade delegada	<p>Um trabalhador que tem a autoridade para agir em nome de outro trabalhador deve ser capaz de indicar ao sistema o que quer fazer. Qualquer uma das suas ações deve exercer o Imprimatur de ambos os usuários (feito por B em nome de A).</p> <p>Quando tiverem terminado, eles serão capazes de indicar que tenham deixado de agir sobre o outro nome do empregado.</p> <p>Note que assumindo autoridade delegada pode conceder privilégios que um usuário que não têm normalmente.</p>

Prevenir lacunas

Resumo	Definição
Não é possível falsificar a autenticação	<p>Não será possível em qualquer parte do sistema de fazer qualquer ação em nome de um usuário a menos que uma parte do sistema de autenticação do usuário tenha sido autenticado.</p> <p>O objetivo deste requisito é de ser capaz de proteger contra desvios de controles normais e perguntar a um componente do sistema para fazer alguma coisa diretamente. Isso se aplica em especial ao servidor de processos (por exemplo, um processo que verifica a identidade do usuário e outro processo que não pode ser explícito).</p>

Padrões de Requisitos - Autorização Específica

Resumo	Definição
Acesso <<Resumo dos Privilégios>>	<<Descrição dos Privilégios>> deve (não) estar acessível a <<descrição de regra de Acesso >>.
Acesso <<Resumo dos Privilégios>>	Uma <<Tipo de usuário>> deve (não) ser capaz de <<Descrição dos privilégios>>.

Exemplos

Categoria 1, regra universal de negação padrão:

Resumo	Definição
Negação de acesso padrão	Um usuário não deve ter acesso para qualquer função ou informação ou outro sistema de recursos a menos que tenha sido concedida a autorização explicitamente, ou a menos que tenha sido designado acesso público. No caso da informação designada como acesso público, este deve ser considerado apenas a capacidade de visualizar as informações, a menos que especificado explicitamente de outra maneira.

Categoria 2, funções:

Resumo	Definição
Acesso somente quando conectado	Um usuário não deve ter acesso às funções não públicas ou informação se não tiver logado ou autenticado.
Acesso de um visitante casual limitado	Um visitante casual para o nosso site (que não foi autenticado como um cliente) terá apenas acesso limitado. Eles não devem ser sequer capazes de iniciar qualquer função que envolve dinheiro (tais como a colocação de um pedido).

Categoria 3, as ações no âmbito das funções:

Resumo	Definição
Manutenção do cliente	Cada tipo de ações que podem ser realizadas dentro das funções de manutenção do cliente deve ser assunto para separar os acessos privilégios. <<Tipo de ação>> deve incluir alterações de endereço, e limite de crédito.

Categoria 4, dados:

Resumo	Definição
Serviços Administrativos tem acesso a empresas clientes	Os funcionários do serviço administrativo devem ter o mesmo acesso a cada um dos seus clientes da empresas como têm os seus próprios trabalhadores.
Empresa acessar as informações financeiras	Companhia de informação financeira deve ter acesso somente os membros do departamento financeiro e os veteranos da direção. Para efeitos deste requisito, <<empresa de informação financeira>>, os valores referentes ao desempenho geral da empresa, e outras informações de natureza contabilística; pedido do cliente e informações sobre pagamento não são classificados como empresa de informação financeira.
Consultas e relatórios não mostram dados inacessíveis	Nenhuma consulta ou relatório deve mostrar os dados para o qual o usuário atual não tem acesso. Se for o caso, os dados inacessíveis devem ser "filtrados". Isto deve ser feito de tal modo que as informação ficarão resumidas (totais, médias e assim por diante) sendo consistente com os dados que é mostrado.

Categorias 2 e 4 combinadas, ambas funções e dados:

Resumo	Definição
Configuração da manutenção do acesso	Somente serão permitidos funcionários nomeados para modificar os parâmetros de configuração, e então, só em áreas expressamente designados. Por exemplo, um gerente de financiamento susceptíveis de serem autorizados a modificar apenas parâmetros relacionados com o financiamento.
Empresa pode executar relatórios dos agentes	A empresa será capaz de executar para seu próprio uso todos os relatórios disponíveis para seus agentes de vendas, para mostrar a informação para qualquer agente selecionado.

Categoria 5, limites:

Resumo	Definição
Limite de restituição ao cliente	O empregado deve ter a possibilidade de aprovar uma restituição para um cliente até (e inclusive) a restituição limite estabelecido para eles.

Categoria 6, tempo:

Resumo	Definição
Iniciar operações somente durante tempo determinado	O empregado deve ter a possibilidade de iniciar operações durante horas do dia estabelecidas. Deve ser possível especificar essas horas do dia para cada trabalhador, mas se não tiverem sido fixados para um trabalhador, um intervalo de horas configurável padrão será usado.

Categoria 7, acesso remoto:

Resumo	Definição
Acesso remoto do empregado	O empregado deve ser capaz de acessar o sistema de fora da empresa para instalações caso seja autorizada (e não outro qualquer).

Categoria 8, força de autenticação

Resumo	Definição
Reduzido acesso ao empregado, sem cartão de identificação	O empregado que tenha um cartão de identificação, mas que sem ele, nos logs durante essa sessão não deve ser capaz de iniciar ou aprovar operações financeiras.

Categoria 9, transferência:

Resumo	Definição
Acessar dados de log como o acesso aos dados por dados originais	O acesso aos dados armazenados em um log deve ser restringido a pelo menos ao mesmo grau como o acesso aos dados originais. Por exemplo, se um usuário está autorizado a ver os detalhes do cliente apenas para uma empresa, eles não serão capazes de ver detalhes de um log de entrada sobre um cliente associada com outra companhia.
Acesso à consulta de informação	Informações que satisfaz uma consulta por um usuário devem ser filtrada para excluir qualquer coisa que o usuário não tem permissão para visualizar.
Acesso aos comentários de documentos	Os comentários feitos em qualquer documento devem ser organizados com o mesmo controle sobre o acesso com o próprio documento.

Categoria 10, regras operacionais:

Resumo	Definição
Não é possível aumentar seus próprios privilégios	Nenhum usuário será capaz de modificar os seus próprios privilégios de acesso. Em particular, nenhum usuário deve ter a possibilidade de aumentar os seus próprios privilégios.
Ver apenas suas próprias ordens	O sistema deve permitir que um cliente possa visualizar apenas as ordens criadas por ele, e não visualizar as transações feitas por outros clientes.

Categoria 11, proibições das coberturas:

Resumo	Definição
Não autenticado, sem controle ou acesso bloqueado	<p>Qualquer atividade comercial normal necessária para o funcionamento do sistema deve ser objeto de controle de acesso a todas as exigências do presente documento. Este requisito exige, em particular:</p> <ol style="list-style-type: none"> 1. Baixo nível de acesso a qualquer base de dados não devem ser exigidas. Isto inclui consultas SQL e qualquer aplicação que permita o acesso equivalente a consultas SQL. 2. Acesso a linha de comando será restrito ao próximo requisito. <p>Se ou quando este requisito é satisfeito por todos os sistemas em execução em uma determinada máquina do servidor, então qualquer tipo de acesso acima descrito como "não será necessário" pode e deve ser fortalecido a "não será permitida."</p> <p>(Mesmo que, para o funcionamento de outros sistemas, os usuários deverão ser concedido ao acesso não controlado a este tipo de requerimento visa prevenir as políticas que devem insistir para que esses mecanismos não podem ser utilizados em relação a este sistema.)</p> <p>Este requisito não se aplica às medidas necessárias a fim de resolver graves problemas de sistema, instalação ou reconfiguração. No entanto, nesses casos, esta exigência deve ser flexibilizada apenas na medida em que seja necessário para obter a tarefa realizada. (Para compensar a redução da segurança em tais situações, é recomendável que sejam aplicados controles manuais adicionais - tais como a estreita supervisão do pessoal quando eles empreender essas tarefas.)</p>

Padrões de Requisitos - Aprovação

Modelo

Resumo	Definição
Aprovação do <<Nome da ação>>	<p>A <<Descrição da ação>> deve <<Homologar as circunstâncias>> ser aprovado por <<descrição do aprovador>>.</p> <p>(<<Declaração de aprovação prontidão>>.)</p> <p>(<<Mecanismo de declaração de aprovação>>.)</p> <p>(Caso a aprovação seja negado <<Descrição da ação rejeitada>>.)</p>

Exemplos

Resumo	Definição
Aprovação da retirada de grandes quantias em dinheiro	Qualquer retirada em dinheiro maior do que o limite de retirada deve ser aprovado por outro empregado em cujo limite de retirada se enquadra. A aprovação é concedida mediante a retirada em função do contador da máquina que permite ao segundo empregado digitar seu ID e senha e, em seguida, verificar a aprovação. O segundo empregado deve também ter a capacidade de rejeitar a retirada, o que resulta em ser cancelada.
Férias do empregado sujeitas à aprovação	Cada pedido de férias de mais de dois dias deverá ser aprovado pelo departamento de recursos humanos.

PADRÕES DE REQUISITOS COMERCIAIS**Padrões de Requisitos - Unidade Multi-Organizacional**

Modelo

Resumo	Definição
Multi-<<Nome do tipo de unidade>>	O sistema deve apoiar múltiplas << Nome do tipo de unidade>> (por <<Nome do Pai do tipo de unidade>>). (Para efeitos do presente caderno, uma << Nome do tipo de unidade>> é <<definição do tipo de unidade>>.) (<<Características declaração>>.) (<<Número de casos declaração>>.)
<<Estrutura síntese>> estrutura organizacional	Deve ser possível definir uma estrutura organizacional <<Estrutura descrição>>. (<<Características declaração>>.)

Exemplos

Resumo	Definição
Multi-empresa	O sistema deve ser capaz de suportar múltiplas empresas simultaneamente. Para efeitos do presente caderno, uma empresa independente é um negócio jurídico em cujo nome o sistema realiza tratamento. Cada empresa terá seus próprios empregados que utilizam o sistema, eles só são da companhia para a qual eles trabalham. (As especificações deste são cobertas por outras exigências.) Uma instalação do sistema poderia ser convidada para acomodar até uma dúzia de empresas.

Padrões de Requisitos - Taxas

Resumo	Definição
<<Nome das Taxas e impostos>> taxas / impostos	O sistema calculará um <<nome das taxas e impostos>> taxa / imposto como uma <<Base>> sobre <<Origem>> desde que <<Estado>>. Ela é paga por <<pagador>> para <<Receptor>> <<Quando cobradas>>. O (taxa) / (impostos) taxa será determinada por <<Taxa / Imposto de Renda determinantes>>. O sistema é responsável pelo <<Sistema de responsabilidade>>. Fonte: <<referência>>.

Exemplos

Resumo	Definição
Taxa de transação	O sistema calculará automaticamente e cobrará uma taxa sobre a transação de cada cliente no modelo para o qual uma transação é definida como taxa a pagar. A taxa de remuneração será definida para cada tipo de transação.
Comissões vendas	O sistema calculará a comissão de vendas a pagar a cada agente de vendas feito por ele. É calculado como uma porcentagem do valor de cada venda (excluindo despesas postais / entrega e quantidades de seguros). A comissão de vendas é paga pelo operador da rede de agentes. O sistema não deve gerar automaticamente os pagamentos aos agentes.