



Universidade Federal de Pernambuco

CENTRO DE INFORMÁTICA

PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Ana Cláudia Veras Beltrão Didier

## WRE-Process: Um Processo de Engenharia de Requisitos Baseado no RUP

**ESTE TRABALHO FOI APRESENTADO À PÓS-GRADUAÇÃO  
EM CIÊNCIA DA COMPUTAÇÃO DO CENTRO DE  
INFORMÁTICA DA UNIVERSIDADE FEDERAL DE  
PERNAMBUCO COMO REQUISITO PARCIAL PARA OBTENÇÃO  
DO GRAU DE MESTRE EM CIÊNCIA DA COMPUTAÇÃO.**

ORIENTADOR: Alexandre Marcos Lins de Vasconcelos

CO-ORIENTADOR: Jaelson Freire Brelaz de Castro

RECIFE, AGOSTO DE 2003



*Às minhas princesinhas, Maria Fernanda  
e Ana Clara, e ao meu marido, Fernando,  
que fazem de nossas vidas um eterno conto  
de fadas.*

# Agradecimentos

*Acima de tudo agradeço a Deus, que sempre esteve presente na minha vida como fonte de inspiração e fortaleza, sendo meu principal apoio nos momentos de dificuldade.*

*Agradeço aos meus pais, que sempre priorizaram a educação dos filhos. Especialmente à minha mãe, que além de estar sempre ao meu lado, me apoiando e me incentivando, desempenha o incansável papel de avó/mãe, suprimindo minha ausência perante suas netas.*

*Agradeço de forma muito carinhosa a meu marido, Fernando, por me incentivar e apoiar incondicionalmente, e pela sua incessável paciência e compreensão com meu trabalho.*

*Agradeço aos meus irmãos, cunhados e sogros, que sempre estiveram presentes procurando como me ajudar.*

*Agradeço aos Professores Alexandre, Jaelson e Hermano, que sempre me incentivaram e procuraram mostrar-me os melhores caminhos a serem seguidos. E a todos os professores do Centro de Informática que contribuíram com meu aperfeiçoamento profissional através de suas orientações.*

*Agradeço ao C.E.S.A.R pela oportunidade de aplicar esse trabalho em seus projetos, especialmente a Ana Sofia Marçal, que como gerente do projeto, sempre acreditou no meu trabalho e me ajudou com sua experiência, e como amiga, sempre me deu palavras de incentivo nos momentos mais difíceis. Também não posso deixar de agradecer a todos que fizeram a equipe do projeto, demonstrando competência, comprometimento e amizade.*

*Agradeço a todos os amigos que me incentivaram com palavras de ânimo ou que, de alguma forma, contribuíram com a execução deste trabalho, principalmente a Aida Ferreira, André Didier, Elisabeth Morais, Eduardo Peixoto, Lourdes Melo, Paula Renata e Rossana Padilha.*

# Resumo

Uma das maiores dificuldades encontradas no desenvolvimento de sistemas computacionais é atender as expectativas dos usuários e clientes, de acordo com o cronograma e custo previstos. Um fator que contribui para este tipo de problema é a dificuldade em elicitar, analisar e especificar os requisitos do sistema. Este tem sido um dos pontos de grande interesse entre os pesquisadores.

Três décadas depois do surgimento da Crise do Software, ainda continuamos com grandes dificuldades para produzir um documento de requisitos e mantê-lo consistente com os outros artefatos produzidos no desenvolvimento das aplicações. Além disto, devido a globalização da economia mundial, a Internet tem se mostrado como um dos mais efetivos e atrativos meios para realização de transações comerciais. As empresas estão migrando seus sistemas corporativos para plataformas baseadas principalmente na Web, tornando o desenvolvimento cada vez mais sofisticado.

Diante deste panorama, este trabalho propõe um processo de Engenharia de Requisitos onde são consideradas características que requerem maior atenção em relação ao desenvolvimento de aplicações Web, tais como: questões sobre elicitação de requisitos, interface do usuário e maior ênfase nos requisitos não-funcionais.

O processo proposto, chamado WRE-Process (Web Requirements Engineering Process), é baseado no fluxo de Requisitos do RUP, por este ser o processo dominante no mercado. Com o desenvolvimento da proposta, procuramos criar um processo que fosse genérico o suficiente para atender diversos domínios e tipos de aplicações, mas que contemplasse necessidades das aplicações Web, como: integração entre requisitos funcionais e não-funcionais, elicitação e desenvolvimento de requisitos de interface, entre outros. Desta forma, podemos dizer a principal contribuição deste processo é fornecer um conjunto coerente de atividades e artefatos direcionados para a Engenharia de Requisitos, considerando características das aplicações Web, mas que mantém a generacidade do RUP, podendo ser aplicado a em diferentes tipos de sistemas de software.

O principal objetivo a ser alcançado com a utilização deste processo é a satisfação do cliente, possibilitando que ele receba um produto conforme suas necessidades e expectativas, dentro do

prazo e custos previstos, porém sem acarretar prejuízos à empresa fornecedora e sem sobrecarregar a equipe de desenvolvimento.

# Abstract

Achieving the users and clients expectations, regarding schedule and cost estimates for software system development, is a very difficult task. A factor that had contributed to it is the difficulty to elicit, analyse and specify the systems requirements. Nowadays, this is a strong research point.

Three decades after the software crisis emerged, we still have difficulties to write the requirements document and keep it consistent with other artifacts. Besides, due the global economy, the internet has been one of the most attractive and effective media to make commercial transactions. Many companies are migrating their coporative applications to web based platform, turning the software development more sophisticated.

In this way, as web applications have some points that request more attention than traditional applications, like requirements elicitation, user interface and more emphasis on non-functional requirements, this dissertation proposes a requirement engineering process to deal with these issues.

The proposed process, called WRE-Process (Web Requirements Engineering Process), is based on the RUP Requirements Workflow. In this work, we built a generic process to attend many applications domains, but mainly considering web applications needs, like: integrating functional requirements with non-functional requirements, elicitation and development of interface requirements, and so on. So the most important contribution of this process is to present a coherent set of activities and artifacts directed to requirements engineering, considering web applications characteristics, but still general as the RUP process is.

The most important goal to be reached is the client satisfaction, with a product that support theirs needs and spectations, on schedule and costs estimates.

# Índice

Capítulo 1	<b>Introdução</b> .....	1
1.1.	Visão Geral .....	1
1.2.	Motivação .....	2
1.3.	Abordagem do Trabalho .....	4
1.3.1.	Engenharia de Software para Web.....	5
1.3.2.	Engenharia de Requisitos.....	6
1.3.3.	RUP (Rational Unified Process).....	8
1.4.	Objetivo da Proposta .....	8
1.5.	Estrutura da Dissertação .....	10
Capítulo 2	<b>Engenharia de Requisitos</b> .....	11
2.1.	Visão Geral .....	11
2.2.	Engenharia de Requisitos .....	11
2.3.	O Documento de Requisitos .....	13
2.4.	Tipos de Requisitos .....	15
2.4.1.	Requisitos Funcionais.....	15
2.4.2.	Requisitos Não-Funcionais.....	23
2.4.3.	Requisitos Organizacionais .....	31
2.5.	O Processo de Engenharia de Requisitos .....	32
2.5.1.	Elicitação de Requisitos.....	34
2.5.2.	Análise e Negociação de Requisitos .....	36
2.5.3.	Documentação dos Requisitos .....	37
2.5.4.	Verificação e Validação de Requisitos.....	37
2.5.5.	Gerência de Requisitos .....	40
2.5.6.	Qualidade no Processo de Engenharia de Requisitos.....	48
2.6.	Engenharia de Requisitos para Web .....	50
2.6.1.	Principais Problemas .....	51
2.6.2.	Fatores Importantes.....	53
2.7.	Considerações Finais .....	59
Capítulo 3	<b>O RUP: Rational Unified Process</b> .....	60
3.1.	Visão Geral do RUP .....	60
3.1.1.	Fases do RUP .....	60
3.1.2.	Conceitos Básicos .....	61
3.1.3.	Principais Características .....	62
3.2.	Fluxos do RUP.....	64
3.3.	Fluxo de Requisitos .....	66
3.3.1.	Subfluxo: Analisar o Problema.....	67



3.3.2.	Subfluxo: Entender as Necessidades dos Stakeholders .....	68
3.3.3.	Subfluxo: Definir o Sistema .....	70
3.3.4.	Subfluxo: Gerenciar o Escopo do Sistema .....	71
3.3.5.	Subfluxo: Refinar a Definição do Sistema .....	72
3.3.6.	Subfluxo: Gerenciar Mudanças nos Requisitos .....	73
3.4.	Considerações Finais .....	74
Capítulo 4	<b>WRE-Process: Um processo para a engenharia de requisitos, baseado no RUP</b> .....	76
4.1.	Visão Geral .....	76
4.2.	Descrição do Processo WRE-Process .....	77
4.2.1.	Subfluxo: Planejar a Engenharia de Requisitos .....	79
4.2.2.	Subfluxo: Elicitar os Requisitos .....	86
4.2.3.	Subfluxo: Analisar e Documentar Requisitos .....	100
4.2.4.	Subfluxo: Definir a Interface .....	109
4.2.5.	Subfluxo: Gerenciar o Escopo do Sistema .....	119
4.2.6.	Subfluxo: Refinar a Definição do Sistema .....	121
4.2.7.	Subfluxo: Validar os Requisitos .....	123
4.2.8.	Subfluxo: Gerenciar Mudanças de Requisitos .....	127
4.3.	Considerações Finais .....	131
Capítulo 5	<b>Relato de Experiência: Sistema de Gestão de Acesso em Aplicações Web</b> .....	133
5.1.	Visão Geral .....	133
5.2.	Descrição do Sistema de Gestão de Acessos .....	134
5.3.	Descrição dos Usuários da Aplicação .....	136
5.4.	O Processo de Engenharia de Requisitos .....	137
5.4.1.	Subfluxo: Planejar Engenharia de Requisitos .....	137
5.4.2.	Subfluxo: Elicitar Requisitos .....	143
5.4.3.	Subfluxo: Analisar e Documentar Requisitos .....	158
5.4.4.	Subfluxo: Definir a Interface .....	166
5.4.5.	Subfluxo: Gerenciar Escopo .....	172
5.4.6.	Subfluxo: Refinar a Definição do Sistema .....	173
5.4.7.	Subfluxo: Validar os Requisitos .....	176
5.4.8.	Subfluxo: Gerenciar Mudanças nos Requisitos .....	179
5.5.	Considerações Finais .....	184
Capítulo 6	<b>Conclusão e Trabalhos Futuros</b> .....	186
6.1.	Principais Contribuições .....	186
6.2.	Trabalhos Relacionados .....	187
6.3.	Perspectiva de Trabalhos Futuros .....	188
6.4.	Considerações Finais .....	189
Referências	.....	191

<i>Apêndice A</i>	<i>Documento de Requisitos</i> .....	<i>198</i>
<i>Apêndice B</i>	<i>Especificação de Casos de Uso</i> .....	<i>208</i>
<i>Apêndice C</i>	<i>Requisitos de Interface</i> .....	<i>220</i>
<i>Apêndice D</i>	<i>Casos de Uso de Interface</i> .....	<i>226</i>

# Índice de Figuras

<i>Figura 1.1- Entradas e saídas do processo de ER [Som98]</i> .....	7
<i>Figura 2.1- Notações para Casos de Uso em UML</i> .....	21
<i>Figura 2.2- Classificação dos RNF, segundo Sommerville [Som98]</i> .....	25
<i>Figura 2.3 - Uma Taxonomia para RNFs [Cys01-A]</i> .....	27
<i>Figura 2.4- Exemplo de Decomposição do Grafo de RNF</i> .....	28
<i>Figura 2.5- Grafo de RNFs Decomposto até suas Operacionalizações</i> .....	29
<i>Figura 2.6- Exemplo de Grafo com Interdependências</i> .....	30
<i>Figura 2.7- Grafo de RNFs com Interdependências Analisadas</i> .....	31
<i>Figura 2.8 - O processo genérico de Engenharia de Requisitos [Som98]</i> .....	33
<i>Figura 2.9- Modelo Espiral do Processo de Engenharia de Requisitos</i> .....	34
<i>Figura 2.10- Exemplo de relacionamento entre artefatos</i> .....	42
<i>Figura 2.11 - Classificação das informações a serem rastreadas</i> .....	43
<i>Figura 2.12 - Modelo de Rastreamento para um Sistema, segundo Toranzo [Tor02-B]</i> .....	47
<i>Figura 3.1- Fluxos do RUP [Rup02]</i> .....	65
<i>Figura 3.2- Fluxo de Requisitos do RUP [Rup02]</i> .....	66
<i>Figura 3.3- Subfluxo Analisar o Problema [Rup02]</i> .....	68
<i>Figura 3.4 - Subfluxo Entender as Necessidades dos Stakeholders [Rup02]</i> .....	69
<i>Figura 3.5 - Subfluxo Definir o Sistema [Rup02]</i> .....	70
<i>Figura 3.6- Subfluxo Gerenciar o Escopo do Sistema [Rup02]</i> .....	71
<i>Figura 3.7- Subfluxo Refinar a Definição do Sistema [Rup02]</i> .....	72
<i>Figura 3.8- Subfluxo Gerenciar Mudanças nos Requisitos</i> .....	74
<i>Figura 4.1- Fluxo do Processo WRE-Process</i> .....	78
<i>Figura 4.2 - Subfluxo Planejar a Engenharia de Requisitos</i> .....	80
<i>Figura 4.3- Subfluxo Elicitar Requisitos</i> .....	87
<i>Figura 4.4 - Modelo de Casos de Uso Preliminar</i> .....	95
<i>Figura 4.5 - Grafo de RNF's com operacionalizações</i> .....	98
<i>Figura 4.6 - Subfluxo Analisar e Documentar Requisitos</i> .....	101
<i>Figura 4.7- Diagrama de Casos de Uso</i> .....	102
<i>Figura 4.8- Modelo de Casos de Uso Refletindo RNF's</i> .....	109
<i>Figura 4.9 - Subfluxo Definir a Interface</i> .....	111
<i>Figura 4.10 - Modelo de Casos de Uso de Interface</i> .....	115
<i>Figura 4.11- Subfluxo Gerenciar Escopo</i> .....	120
<i>Figura 4.12 - Subfluxo Refinar a Definição do Sistema</i> .....	122
<i>Figura 4.13- Subfluxo Validar Requisitos</i> .....	124
<i>Figura 4.14 - Subfluxo Gerenciar Mudanças nos Requisitos</i> .....	129

<i>Figura 2.1 -Diagrama de Casos de Uso Preliminar - Pacote Configuração do Sistema Gestão de Acesso .....</i>	<i>151</i>
<i>Figura 2.2- Diagrama de Casos de Uso Preliminar - Pacote Gestão de usuários.....</i>	<i>152</i>
<i>Figura 2.3- Diagrama de Casos de Uso Preliminar - Pacote Controle de Acesso .....</i>	<i>153</i>
<i>Figura 2.4 - Diagrama de Casos de Uso Preliminar - Pacote Trilha de Auditoria.....</i>	<i>154</i>
<i>Figura 2.5- Grafo de RNF's com operacionalizações .....</i>	<i>156</i>
<i>Figura 2.6-Diagrama de casos de uso refletindo RNF's - Pacote Configuração do Sistema Gestão de Acesso.....</i>	<i>161</i>
<i>Figura 2.7- Diagrama de casos de uso refletindo RNF's - Pacote Gestão de Usuários.....</i>	<i>162</i>
<i>Figura 2.8- Diagrama de casos de uso refletindo RNF's - Pacote Controle de Acesso .....</i>	<i>163</i>
<i>Figura 2.9 - Diagrama de casos de uso refletindo RNF's - Pacote Trilha de Auditoria .....</i>	<i>164</i>
<i>Figura 2.10- Modelo de Casos de Uso de Interface.....</i>	<i>168</i>
<i>Figura 2.11- Exemplo da tela de login, do protótipo de interface .....</i>	<i>171</i>
<i>Figura 2.12- Exemplo da tela de menu, no protótipo de interface.....</i>	<i>171</i>
<i>Figura 2.13- Tela Cadastrar Sistema - Protótipo de Interface .....</i>	<i>178</i>
<i>Figura 2.14 - Tela Inserir Sistema - Protótipo de Interface.....</i>	<i>178</i>
<i>Figura 2.15 - Modelo de Rastreamento do Sistema Gestão de Acessos.....</i>	<i>182</i>

# Índice de Tabelas

<i>Tabela 2.1- Modelo padrão do Documento de Requisitos, segundo IEEE/ANSI 830-1993 citado em [Som98].....</i>	<i>14</i>
<i>Tabela 2.2- Modelo do Documento de Requisitos sugerido no RUP 2002[Rup02].....</i>	<i>14</i>
<i>Tabela 2.3- Modelo do Documento de Requisitos proposto por Duran [Dur00] .....</i>	<i>14</i>
<i>Tabela 2.4-Etapas para a construção do LAL [Had99] .....</i>	<i>17</i>
<i>Tabela 2.5 - Heurísticas de Representação [Cys01-A].....</i>	<i>18</i>
<i>Tabela 2.6 - Esquema para construção de cenários, por Júlio Leite [Cys01-A] .....</i>	<i>19</i>
<i>Tabela 2.7 - Template de caso de uso proposto por Cockburn [Coc00] .....</i>	<i>22</i>
<i>Tabela 2.8 - Template de Casos de Uso proposto por Duran [Dur00].....</i>	<i>22</i>
<i>Tabela 2.9-Template de casos de uso proposto no RUP.....</i>	<i>23</i>
<i>Tabela 2.10 - Requisitos de um Sistema de Biblioteca.....</i>	<i>47</i>
<i>Tabela 2.11- Relação de Programas de um sistema de Biblioteca.....</i>	<i>48</i>
<i>Tabela 2.12- Matriz de Rastreabilidade "Representado_em" entre Requisitos e Programas .....</i>	<i>48</i>
<i>Tabela 2.13 - Processo VRU [San02].....</i>	<i>58</i>
<i>Tabela 4.1- Template para Agenda de Reuniões .....</i>	<i>82</i>
<i>Tabela 4.2- Template para Pauta de Reuniões.....</i>	<i>82</i>
<i>Tabela 4.3- Template do Plano de Gerência de Requisitos.....</i>	<i>83</i>
<i>Tabela 4.4- Exemplo de Rastreabilidade no Plano de Gerência de Requisitos.....</i>	<i>84</i>
<i>Tabela 4.5- Sugestão de tabela para organizar as classes candidatas .....</i>	<i>84</i>
<i>Tabela 4.6 - Template do Plano de Validação .....</i>	<i>85</i>
<i>Tabela 4.7 - Template do documento de visão .....</i>	<i>86</i>
<i>Tabela 4.8- Template de Ata de Reunião.....</i>	<i>88</i>
<i>Tabela 4.9 - Exemplo de Questionário de Requisitos de Interface .....</i>	<i>91</i>
<i>Tabela 4.10- Exemplo de Glossário Léxico (Extensão proposta ao glossário do RUP).....</i>	<i>93</i>
<i>Tabela 4.11- Template do Documento de Requisitos.....</i>	<i>105</i>
<i>Tabela 4.12- Template para a Especificação dos Casos de Uso .....</i>	<i>107</i>
<i>Tabela 4.13 - Template do Documento Requisitos de Interface.....</i>	<i>115</i>
<i>Tabela 4.14 - Template Especificação de Casos de Uso de Interface.....</i>	<i>116</i>
<i>Tabela 4.15- Template do documento de descrição do layout.....</i>	<i>118</i>
<i>Tabela 4.16- Template do Registro da Validação dos Requisitos .....</i>	<i>127</i>
<i>Tabela 5.1- Agenda de Reuniões .....</i>	<i>138</i>
<i>Tabela 5.2- Exemplo de Pauta de Reunião .....</i>	<i>139</i>
<i>Tabela 5.3- Expressões usadas nas matrizes de rastreamento .....</i>	<i>140</i>
<i>Tabela 5.4- Tabela para organizar as classes candidatas ao rastreamento.....</i>	<i>140</i>

<i>Tabela 5.5 - Plano de Gerência de Requisitos</i> .....	140
<i>Tabela 5.6 - Documento de Visão</i> .....	141
<i>Tabela 5.7 - Exemplo de ata de registro de reunião</i> .....	144
<i>Tabela 5.8-Ata de Reunião de Elicitação de Requisitos de Interface</i> .....	146
<i>Tabela 5.9- Glossário Léxico do Sistema Gestão de Acesso</i> .....	147
<i>Tabela 5.10 - Expressões usadas nas matrizes de rastreamento</i> .....	157
<i>Tabela 5.11- Matriz de Informações do Ambiente Externo</i> .....	157
<i>Tabela 5.12 - Matriz com os Objetivos Organizacionais</i> .....	157
<i>Tabela 5.13 - Matriz com os Objetivos do Sistema</i> .....	158
<i>Tabela 5.14 - Matriz com os Subsistemas</i> .....	158
<i>Tabela 5.15- Incremento para o Glossário Léxico da Aplicação</i> .....	159
<i>Tabela 5.16- Exemplo de Requisito</i> .....	165
<i>Tabela 5.17- Exemplo de Caso de Uso</i> .....	165
<i>Tabela 5.18 - Exemplo de Requisito de Interface</i> .....	168
<i>Tabela 5.19 - Exemplo de Caso de Uso de Interface</i> .....	169
<i>Tabela 5.20 - Exemplo do Documento de Descrição da Interface</i> .....	169
<i>Tabela 5.21- Matriz dos Requisitos do Pacote Configuração</i> .....	173
<i>Tabela 5.22-Matriz de Casos de Uso do Pacote Configuração</i> .....	173
<i>Tabela 5.23 - Matriz com os diagramas do sistema Gestão de Usuários</i> .....	173
<i>Tabela 5.24- Exemplo de Caso de Uso Detalhado</i> .....	174
<i>Tabela 5.25 - Registro de Validação de Requisitos</i> .....	179
<i>Tabela 5.26 - Classes candidatas ao rastreamento</i> .....	181
<i>Tabela 5.27- Representação Matricial do Relacionamento dep_reg_org</i> .....	183
<i>Tabela 5.28- Representação Matricial do Relacionamento dep_objetivoSistema</i> .....	183
<i>Tabela 5.29- Representação Matricial do Relacionamento Mapeado_em</i> .....	183
<i>Tabela 5.30- Representação Matricial do Relacionamento Representado_em_diag</i> .....	184

### 1.1. Visão Geral

A “Crise de Software” despertou o interesse da comunidade científica no que diz respeito à forma como desenvolver software. Através dos anos, várias ferramentas, técnicas e métodos têm sido desenvolvidos para auxiliar o processo de desenvolvimento do software. Nos anos 60, os sistemas eram desenvolvidos para resolver problemas relativamente bem estruturados e, de certa forma, razoavelmente simples. Com o aumento da complexidade e o crescimento da natureza dos problemas a serem resolvidos, o desenvolvimento de software passou a ser uma tarefa cada vez mais difícil [Sou02],[Ara01]. Atualmente, a globalização da economia e a popularização da Internet estão provocando uma grande demanda no desenvolvimento de aplicações Web. As organizações estão migrando seus sistemas críticos para uma plataforma baseada na Web e, a cada dia, surgem novos tipos de sistemas e ferramentas voltadas à Web, tornando o desenvolvimento de software mais sofisticado.

Um dos pontos de grande interesse entre os pesquisadores, desde o surgimento da crise do software, refere-se à forma de como elicitar, coletar, analisar e especificar formalmente os requisitos de um sistema [Tor02-B]. Três décadas depois, ainda continuamos com grandes dificuldades para produzir um documento de requisitos e mantê-lo consistente com outros artefatos produzidos no desenvolvimento de software.

Como as aplicações Web possuem algumas características que não são relevantes em aplicações tradicionais, e que precisam ser tratadas no processo de desenvolvimento, este trabalho propõe um processo para a realização da Engenharia de Requisitos no desenvolvimento de software, considerando estas características. Devemos salientar que o processo proposto é genérico e pode ser usado em qualquer tipo de aplicação, inclusive aquelas que não são voltadas à Web.

Este capítulo apresenta as principais motivações deste trabalho e define alguns conceitos fundamentais. Descreve o escopo e objetivos do trabalho e apresenta a estrutura da dissertação.

## 1.2. Motivação

Uma das maiores dificuldades encontradas no desenvolvimento de sistemas computacionais é atender as expectativas dos usuários e clientes, de acordo com o cronograma e custo previstos. Desta forma, as empresas de software necessitam competir num mercado que cresce, pressionado pela escassez de tempo para desenvolver os projetos e pela necessidade de oferecer maior qualidade nos produtos de modo a satisfazer seus clientes.

Em 2000 o Standish Group publicou um relatório [Joh03] referindo-se ao desenvolvimento de aproximadamente 300 mil projetos de software nos Estados Unidos, com as seguintes estatísticas:

- 28% dos projetos foram concluídos no tempo previsto e dentro dos custos esperados, com todas as características e funções originais especificadas;
- 23% dos projetos foram cancelados antes de serem concluídos;
- 49% dos projetos foram concluídos após os prazos previstos, ultrapassaram os custos estimados e apresentaram menos características e funções que as especificadas originalmente.

No mesmo relatório [Joh03], é apresentado o resultado de uma pesquisa similar, realizada em 1998 no desenvolvimento de 200 mil projetos, onde apenas 26% deles foram classificados no primeiro item acima, e os dois itens seguintes apresentaram taxas de 28% e 46%, respectivamente. A melhora do desempenho dos projetos, constatada na pesquisa de 2000, é resultado, principalmente, dos esforços realizados pelas inúmeras pesquisas desenvolvidas em todo o mundo na tentativa de solucionar os problemas mencionados.

Com o resultado das pesquisas e dos avanços tecnológicos, o mercado está cada vez mais agressivo e turbulento. Os clientes de software exigem mudanças de forma mais freqüente a fim de manterem posições competitivas no mercado. Com a globalização da economia mundial e a popularização da Internet, as aplicações Web têm se mostrado como um dos mais novos e efetivos meios para a realização de negócios. Além disto, as empresas estão migrando seus sistemas corporativos e departamentais para plataformas baseadas na Web, tendo em vista que o espaço geográfico de atuação de uma empresa pode ser em âmbito mundial. Este fato possibilita uma maior disseminação do conhecimento e outras inúmeras vantagens, mas por outro lado, do ponto de vista da Engenharia de Software, começaram a surgir preocupações que antes não existiam nas aplicações tradicionais. Por este motivo, surgiu o projeto WEST (Web Oriented Software Technology), que tem como objetivo principal, desenvolver e gerar a tecnologia



necessária para obter métodos de desenvolvimento de software para aplicações de negócios em ambientes web, congregando os resultados e esforços dos vários grupos participantes, vide [Ler03]. Esta tecnologia abordará uma visão unificada entre as estratégias de desenvolvimento de aplicações hipermídia e convencionais, direcionando os esforços para assegurar a qualidade dos produtos de software finais obtidos.

Um projeto de sucesso pode ser definido como aquele que é entregue no prazo, dentro do orçamento previsto e atendendo aos requisitos de funcionalidade e de qualidade acordados com o cliente [Pmi03]. Nesse contexto, é consenso na literatura especializada que a engenharia de requisitos é um dos aspectos mais críticos para o sucesso dos projetos de software, pois esta é a responsável por elicitar, analisar, especificar, documentar e validar os requisitos do sistema, fornecendo dados para planejar o projeto, definir seu escopo, alocar recursos e estimar custos e prazos. O relatório publicado pelo Standish Group [Joh03], descreve dez critérios de sucesso de um projeto, e entre eles podemos destacar (i) Estabelecer os “Requisitos Base” de um sistema e (ii) Desenvolver uma Metodologia Formal, que são os tópicos principais deste trabalho.

Os processos de engenharia de requisitos são dominados por fatores humanos, sociais e organizacionais porque eles sempre envolvem um conjunto de partes interessadas com diferentes conhecimentos e objetivos organizacionais. As partes interessadas pelo sistema, chamadas também de *stakeholders*, podem ter uma variedade de conhecimento técnico e não técnico e de diferentes disciplinas, além de diferentes influências políticas. Exemplos de *stakeholders* são: usuários finais do sistema, patrocinadores do sistema, especialistas de domínio que possuem informações essenciais sobre o domínio da aplicação, engenheiros de software responsáveis pelo desenvolvimento, gerentes dos usuários finais do sistema, fiscais externos que verificam se o sistema satisfaz os requisitos legais, etc.

A alta competitividade das empresas decorrente da globalização tem implicado em um maior foco na melhoria da qualidade dos produtos desenvolvidos e processos utilizados. Nesse contexto, as empresas têm reconhecido a necessidade da melhoria dos processos e produtos como um aspecto relevante para a sobrevivência e permanência no mercado globalizado e competitivo. A melhoria do processo está relacionada com a modificação do processo de forma a alcançar melhoria de qualidade, redução de prazo e redução de recursos [Som98]. Referindo-se ao uso de metodologias de desenvolvimento de software, Charette, em [Char03], afirma que o RUP (Rational Unified Process, da IBM Rational) é a metodologia dominante no mercado, sendo utilizada em 51% das organizações.

Visando a melhoria do desenvolvimento de software, vários modelos para avaliação do processo de produção de software têm sido propostos por instituições no mundo inteiro. Dentre os mais utilizados, podemos citar o Capability Maturity Model (CMM), do Software Engineering Institute (SEI), o qual tem sido bastante utilizado pelas empresas de software [Cmm03]. O CMM fornece às organizações diretrizes para controlar seus processos de desenvolvimento de software, de modo a desenvolver e manter software de melhor qualidade. O CMM propõe um caminho gradual, através de níveis de maturidade da capacitação, na escala de 1 a 5, onde quanto maior a escala, mais maduro é o processo. Um conjunto de práticas-chaves da gerência de requisitos deve ser inserido no processo de uma organização para que ela atinja o nível 2 de maturidade do CMM. Diante deste fato, constata-se a importância de definir um processo de Engenharia de Requisitos (ER) no contexto da produção de software.

Um importante fator que motivou o desenvolvimento deste trabalho foi o fato do autor ser um profissional experiente nas atividades relacionadas à Engenharia de Requisitos e ter sido o Engenheiro de Requisitos responsável pelo desenvolvimento de um Sistema de Gestão de Acesso para aplicações Web, no CESAR (Centro de Estudos e Sistemas Avançados do Recife). Sendo assim, nesta dissertação procuramos buscar soluções para as dificuldades encontradas na experiência prática do desenvolvimento de uma aplicação Web, seguindo um processo baseado no RUP. Aplicamos nossa proposta sobre este sistema real para validá-la, conforme será apresentado no Capítulo 5.

Vale salientar que, até o nosso conhecimento, vários problemas no desenvolvimento de aplicações Web já foram detectados, entretanto ainda não existe nenhuma proposta que trate todos os problemas existentes no processo de Engenharia de Requisitos. Este trabalho apresenta um processo de Engenharia de Requisitos que contempla as necessidades das aplicações web e, ao mesmo tempo, estende o fluxo de requisitos do RUP para solucionar alguns problemas, que serão apresentados no Capítulo 3.

### **1.3. Abordagem do Trabalho**

Buscando a melhoria do desenvolvimento de software através de uma Engenharia de Requisitos mais efetiva que levasse em consideração aspectos existentes em aplicações Web, iniciamos nosso trabalho estudando a Engenharia de Software para Web, especializando-nos na Engenharia de Requisitos para Web. Após essa etapa inicial, aprofundamos nossos conhecimentos em Engenharia de Requisitos e nas fases do processo genérico de ER, definido

por Sommerville e Kotonya em [Som98], sempre procurando associar os objetivos e atividades com as características peculiares da Web. Analisamos trabalhos de diversos autores ligados ao Projeto West, como Amador Duran [Dur02],[Dur00]; Graciela Hadad [Had99] ; Jaelson Castro [Cas99], [Cas00-A], [Cas00-B], [Tor02-A]; Júlio Leite [Lei97], [Lei00]; Luis Cysneiros [Cys98], [Cys01-A], [Cys01-B]; Paolo Paolini [Bar01], [Pao02]; entre outros, para compor nossa proposta. Com o conhecimento do estado da arte da Engenharia de Requisitos, passamos a estudar o fluxo de requisitos do RUP, analisando-o e comparando-o com o processo genérico de ER. Com a base de conhecimento adquirida, propusemos o WRE-Process. Para avaliar a efetividade do processo proposto, relatamos a experiência obtida ao aplicá-lo sobre um projeto real. O resultado do trabalho realizado é apresentado ao longo desta dissertação.

As próximas seções apresentam alguns conceitos básicos das áreas pesquisadas neste trabalho: *Engenharia de Software para Web*, *Engenharia de Requisitos* e *RUP (Rational Unified Process)*.

### **1.3.1. ENGENHARIA DE SOFTWARE PARA WEB**

A Engenharia de Software para a Web trata todas as fases do desenvolvimento de software Web, desde a sua concepção e elaboração até a implantação, incluindo a avaliação de qualidade e manutenção contínua [Gin01]. Embora a Engenharia de Software para Web adote vários princípios da Engenharia de Software tradicional, ela incorpora novas abordagens, metodologias, ferramentas e técnicas que buscam adequar o desenvolvimento às características únicas das aplicações Web. As principais atividades desenvolvidas são [Gin01]:

- integração com sistemas legados;
- migração de sistemas legados para o ambiente Web;
- desenvolvimento de aplicações de tempo real baseadas na Web;
- testes, verificação e validação;
- avaliação da qualidade;
- gerenciamento de projeto e configuração;
- “métricas Web” – para estimar o esforço de desenvolvimento;
- especificação e análise de requisitos;
- desenvolvimento de técnicas e metodologias para sistemas Web;
- avaliação e especificação da performance das aplicações;
- atualização e manutenção de software;
- desenvolvimento centrado no usuário;

- desenvolvimento de aplicações para usuários finais.

As principais diferenças entre aplicações Web e as tradicionais referem-se a questões sobre navegação, organização da interface e implementação. Algumas dessas características são [Gin01]:

- a maioria dos sistemas web são orientados a documentos, contendo páginas web estáticas e dinâmicas;
- sistemas web são focados na aparência, exigindo criatividade na apresentação visual e a incorporação de recursos multimídia;
- a maioria dos sistemas web são dirigidos a conteúdo, portanto, desenvolver aplicações web inclui desenvolver o conteúdo a ser apresentado;
- sistemas web precisam atender às necessidades de usuários com diferentes habilidades e capacidades, complicando a interação homem-máquina, a interface com o usuário e a apresentação das informações;
- a natureza e as características da mídia da web não estão tão bem entendidas quanto a mídia dos softwares tradicionais;
- a maioria dos sistemas web precisam ser desenvolvidos em um curto espaço de tempo (Web-time), tornando difícil a adoção do mesmo nível de formalismo no planejamento e teste utilizado em sistemas tradicionais;
- Os sistemas web diferem de sistemas tradicionais quanto ao ambiente em que são disponibilizados.

As fases do ciclo de um sistema de informação necessitam ser adaptadas para melhor atenderem o desenvolvimento de aplicações Web, desde os requisitos (com grande atenção aos requisitos não-funcionais e de interface), até o tratamento de questões como audiência da aplicação e compatibilidade de ferramentas, além de preocupação excessiva com segurança. Diante desse panorama, são necessárias adaptações em processos de desenvolvimento de software já existentes, para melhor atenderem às necessidades do desenvolvimento de aplicações Web.

### **1.3.2. ENGENHARIA DE REQUISITOS**

A Engenharia de Requisitos é uma das mais importantes etapas do processo de engenharia de software. Isto decorre porque a maior parte dos problemas e, geralmente, os mais dispendiosos e de maior impacto negativo no desenvolvimento de software, são originados nas etapas iniciais do desenvolvimento [Som98]. As principais atividades do processo de Engenharia de Requisitos

são: elicitação, análise, negociação, especificação, gerenciamento e validação de requisitos. Normalmente, falhas na realização destas atividades, resultam em documentos de requisitos inconsistentes, incompletos e conseqüentemente produtos de software de baixa qualidade.

É consenso que a qualidade do produto de software, bem como a satisfação do cliente, dependem de uma completa e consistente especificação de requisitos. Se os requisitos são especificados de forma incompleta ou inconsistente, os artefatos resultantes das próximas etapas do processo de software (Projeto, Implementação e Testes) não terão a qualidade desejada. Quanto mais tarde problemas com requisitos forem detectados no processo de desenvolvimento, maior será o custo para corrigi-los.

Processo é um conjunto organizado de atividades que transformam entradas em saídas, conforme apresentado na Figura 1.1. As descrições de processos encapsulam as informações e procedimentos que permitem que estes sejam reusados [Som98].

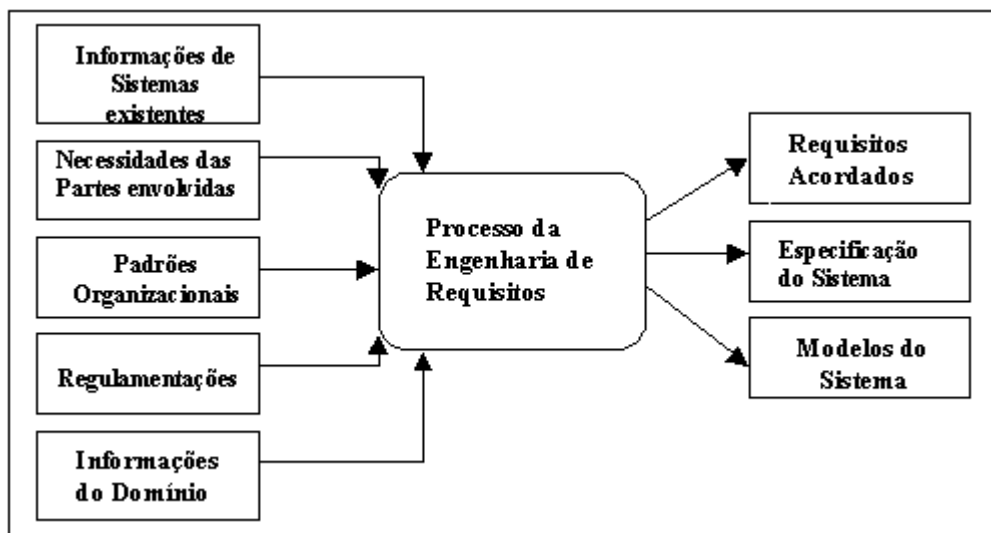


Figura 1.1- Entradas e saídas do processo de ER [Som98]

Os processos de requisitos variam radicalmente de uma organização para outra e os principais fatores que contribuem para esta variação são [Som98]:

- as tecnologias e métodos usados para executar a engenharia de requisitos variam de acordo com a maturidade de uma organização para outra;
- as técnicas aplicadas para engenharia e gerência de requisitos variam de uma organização para outra;
- a cultura de uma organização tem um importante efeito em todos os processos de negócios;

- diferentes tipos de sistemas podem necessitar de diferentes processos de engenharia de requisitos.

### **1.3.3. RUP (RATIONAL UNIFIED PROCESS)**

O Rational Unified Process (RUP) é um processo genérico de engenharia de software desenvolvido e mantido pela IBM Rational Software Corporation [Rat02], estando integrado a um *suite* de ferramentas de apoio ao desenvolvimento de software. Seu objetivo é garantir a produção de software de alta qualidade e atender às necessidades dos usuários finais, de acordo com o cronograma e o orçamento previstos. Esse processo oferece um método disciplinado para a distribuição de tarefas e responsabilidades durante o desenvolvimento do software, explicitando o quê, quando e por quem cada tarefa deve ser executada.

Ele é um processo customizável e pode ser adaptado à forma de trabalho das organizações. É disponibilizado em páginas da Web, permitindo fácil acesso aos membros da organização, além de prover *guidelines*, modelos e exemplos que facilitam sua utilização.

O RUP utiliza UML - Unified Modeling Language - como linguagem padrão para elaboração da modelagem de software orientados a objetos e possui 9 fluxos de atividades:

- Modelagem do Negócio (Business Modeling)
- Requisitos (Requirements)
- Análise e Projeto (Analysis and Design)
- Implementação (Implementation)
- Teste (Test)
- Distribuição (Deployment)
- Gerência de Projeto (Project Management)
- Gerência de Configuração e Mudanças (Configuration and Change Management)
- Configuração do Ambiente (Environment)

Neste trabalho, a versão do RUP descrita é a 2002.05.00 [Rup02] e o foco principal será no fluxo de Requisitos.

## **1.4. Objetivo do Trabalho**

Como os problemas mais dispendiosos e de maior impacto negativo no desenvolvimento de software, são originados nas etapas iniciais do desenvolvimento, isto é, na fase de Requisitos, temos como objetivo dessa dissertação propor o WRE-Process (Web Requirements Engineering

Process), um processo para a Engenharia de Requisitos em aplicações Web. Mais especificamente, este trabalho consiste na fusão entre o processo genérico da Engenharia de Requisitos com o fluxo de Requisitos do RUP – Rational Unified Process, utilizando algumas técnicas desenvolvidas no projeto WEST para a construção de aplicações Web.

O aumento da competitividade na busca por novos mercados, novas estratégias e técnicas para a disponibilização de bens e serviços têm sido criadas e usadas. Dentro deste panorama, a Internet tem se mostrado como um meio efetivo para atender estas necessidades mercadológicas. Sendo assim, a complexidade no desenvolvimento de software tem aumentado consideravelmente nos últimos anos. Como as empresas estão desenvolvendo ou migrando seus sistemas corporativos para o ambiente Web, faz-se necessário o uso de técnicas disciplinadas, novos métodos e ferramentas para desenvolvimento, implantação e avaliação destas aplicações. Para evitar uma possível crise na Web e alcançar maior sucesso no desenvolvimento de aplicações complexas estas técnicas devem levar em consideração as características especiais da nova mídia, os ambientes operacionais, os cenários e multiplicidade de perfis de usuários e o tipo, experiência e conhecimento das pessoas envolvidas no processo de construção das aplicações.

As vantagens do uso de um processo no desenvolvimento de software, descritas em [Sou02], são: ajudar a construir um produto de melhor qualidade, em termos de padrões de documentação, aceitação do usuário, manutenibilidade e consistência do software; ajudar a garantir que os requisitos do sistema sejam completamente atendidos; ajudar a gerenciar o projeto, pois possibilita um melhor controle da execução do projeto e a redução de custos de desenvolvimento; e promover a comunicação entre participantes do projeto, pela definição dos participantes essenciais e interações, bem como pela estruturação do processo como um todo.

O RUP foi escolhido devido à sua genericidade e por ser adaptável, bem como por reunir o melhor de várias técnicas modernas de desenvolvimento de software. Além disto, o RUP é uma metodologia que atende todo o ciclo de vida do desenvolvimento de software, sendo bem aceita no meio acadêmico e bastante disseminada no mercado comercial.

O WRE-Process pode ser visto como uma extensão do fluxo de requisitos do RUP, onde novas atividades foram criadas e outras adaptadas visando atender alguns princípios da engenharia de requisitos e tratar características de aplicações Web, com ênfase nos aspectos não-funcionais, de navegação e de apresentação. Entretanto, o WRE-Process é um processo genérico de ER, que atende às necessidades da Web, mas que pode ser utilizado por diversos tipos de aplicações.

## 1.5. Estrutura da Dissertação

Além deste capítulo introdutório, a dissertação está organizada da forma descrita a seguir:

- O Capítulo 2 apresenta o estado da arte da Engenharia de Requisitos, sendo descritas todas as etapas do processo genérico de ER. No decorrer deste capítulo são abordados conceitos e técnicas que serão aplicadas na proposta do WRE-Process. Este capítulo também apresenta as principais características da Engenharia de Requisitos para Web.
- O Capítulo 3 traz um resumo da metodologia de desenvolvimento de software RUP, através da apresentação de suas características, fases e fluxos. É dada ênfase no Fluxo de Requisitos, onde são detalhados seus subfluxos.
- No Capítulo 4 é apresentado o WRE-Process, um processo para o desenvolvimento da Engenharia de Requisitos de software, principalmente àqueles direcionados à web. O processo é apresentado através de subfluxos e, em cada subfluxo, são detalhadas as atividades propostas. Na descrição destas atividades são aplicados os conceitos e técnicas vistos no Capítulo 2.
- O Capítulo 5 descreve o relato de experiência da aplicação processo do proposto no desenvolvimento de um sistema para realizar a Gestão de Acessos de Usuários em Aplicações Web.
- Por fim, no Capítulo 6 apresenta as conclusões do trabalho, as contribuições trazidas por este, trabalhos relacionados e melhorias para trabalhos futuros.
- Os Apêndices A, B, C e D correspondem a documentos gerados na aplicação do processo, que são referenciados no Capítulo 5.



### 2.1. Visão Geral

Uma das maiores dificuldades encontradas no desenvolvimento de sistemas computacionais é atender as expectativas dos usuários e clientes, de acordo com o cronograma e custo previstos. Um fator que contribui para este tipo de problema é a dificuldade em elicitar, analisar e especificar os requisitos do sistema.

Este capítulo aborda boas práticas de Engenharia de Requisitos (ER) que podem reduzir os problemas existentes e minimizar seu impacto no produto final. Isso se dá através da definição dos conceitos básicos usados na Engenharia de Requisitos (ER); do detalhamento das fases do processo genérico de ER, onde são apresentados os conceitos e técnicas usadas na elaboração do processo WRE-Process; da melhoria do processo de software para a produção de produtos com qualidade. Também são descritas as características da Engenharia de Requisitos para Web, enfatizando a Interface com o usuário. Por fim, é vista a necessidade de um processo que oriente o desenvolvimento de aplicações com características Web e é realizada uma análise conclusiva do capítulo.

### 2.2. Engenharia de Requistos

Segundo Sommerville, em [Som98] e [Som03], **Requisitos** correspondem à descrição dos serviços que um sistema deve prover e às restrições sobre as quais ele deve operar; e a **Engenharia de Requisitos** é um processo que envolve todas as atividades necessárias para coletar, documentar e manter o conjunto de requisitos de um sistema. O uso do termo “Engenharia” implica que técnicas sistemáticas e repetitivas podem ser aplicadas para certificar que os requisitos de um sistema estejam completos, consistentes, relevantes, etc.

O processo de engenharia de requisitos tem como principal objetivo gerar, validar e manter a documentação dos requisitos de um sistema. Através do *Documento de Requisitos* são estabelecidos, oficialmente, os requisitos do sistema para clientes, usuários finais e desenvolvedores de software.

Como os requisitos são dinâmicos, isto é, o surgimento de novos requisitos ou mudanças nos requisitos existentes podem ocorrer, então há necessidade de gerenciar estes requisitos [Som98].

Alterações em requisitos são inevitáveis e ocorrem enquanto os requisitos estão sendo elicitados, analisados, validados e depois do sistema implantado. Os principais fatores que levam a estas mudanças são:

- erros, conflitos e inconsistências nos requisitos: Quando os requisitos são analisados e implementados, erros e inconsistências aparecem e devem ser corrigidos. Eles podem ser descobertos durante a análise e validação de requisitos ou mais tarde durante o processo de desenvolvimento;
- evolução do conhecimento do cliente/usuário final do sistema: Ao se desenvolverem os requisitos, clientes e usuários-finais desenvolvem um melhor entendimento do que eles realmente querem do sistema;
- problemas técnicos, de custo e prazo: Problemas podem ser encontrados no momento da implementação de um requisito, o que pode levar a mudanças no mesmo.
- mudanças nas prioridades dos clientes: A prioridade dos clientes pode mudar durante o desenvolvimento do sistema, como resultado de mudanças no ambiente de negócios, surgimento de novos competidores, mudanças na equipe, etc;
- mudanças ambientais: O ambiente no qual o sistema será instalado poderá mudar, de forma que os requisitos de sistema precisem ser alterados para manter compatibilidade;
- mudanças organizacionais: A organização que pretende usar o sistema pode precisar mudar sua estrutura e processos, resultando em novos requisitos do sistema.

Gerência de requisitos é o processo para gerenciar mudanças dos requisitos de um sistema. As principais preocupações do gerenciamento de requisitos são:

- gerenciar mudanças nos requisitos que foram concordados;
- gerenciar o relacionamento entre os requisitos;
- gerenciar as dependências entre os documentos de requisitos e outros documentos produzidos no processo de engenharia de sistemas.

Com o gerenciamento dos requisitos, os custos de desenvolvimento dos sistemas são reduzidos e as necessidades dos usuários são atendidas, pois eles passam a ter maior controle sobre as alterações dos requisitos do sistema. Com isto, os clientes ficam mais satisfeitos.

## 2.3. O Documento de Requisitos

O *Documento de Requisitos*, também chamado de Especificação de Requisitos de Software, é a declaração oficial do que o sistema deve contemplar, ou seja, o que é exigido dos desenvolvedores de software. O documento de requisitos tem um conjunto diversificado de usuários, abrangendo tanto os clientes do sistema (a alta gerência e usuários finais), como aqueles que são responsáveis pelo seu desenvolvimento (engenheiros e projetistas de software e engenheiros de teste).

Segundo Sommerville e Kotonya [Som98], o documento de Requisitos é um documento formal usado para comunicar os requisitos, descrevendo:

- os serviços e funções que o sistema deve prover;
- as limitações sobre as quais o sistema deve operar;
- as propriedades gerais do sistema;
- as definições de integração com outros sistemas;
- informações sobre o domínio da aplicação do sistema; Ex.: como calcular um certo tipo de computação;
- limitações nos processos usados para desenvolver o sistema;
- descrições sobre o hardware no qual o sistema irá executar; etc

Adicionalmente, deverá sempre conter um capítulo introdutório que provê um resumo do sistema, necessidades de negócio suportadas pelo sistema e um glossário que explica a terminologia usada.

Diversas organizações definiram padrões para o documento de requisitos. Segundo Sommerville, em [Som98], [Som03], o padrão mais amplamente conhecido é o IEEE/ANSI 830-1993, cuja estrutura sugerida é apresentada na Tabela 2.1.

Este padrão é genérico e pode ser aplicado em uma variada gama de documentos de requisitos. Em geral, nem todas as partes do documento são necessárias para todas as empresas, com isto, cada organização deverá adaptar o padrão de acordo com o tipo de sistema que desenvolve.

Tabela 2.1- Modelo padrão do Documento de Requisitos, segundo IEEE/ANSI 830-1993 citado em [Som98]

<ul style="list-style-type: none"><li><b>1. Introdução</b><ul style="list-style-type: none"><li>1.1 Propósito do documento de Requisitos</li><li>1.2 Escopo do produto</li><li>1.3 Definições, acrônimos e abreviações</li><li>1.4 Referencias</li><li>1.5 Resumo do resto do documento</li></ul></li><li><b>2. Descrição Geral</b><ul style="list-style-type: none"><li>2.1 Perspectiva do produto</li><li>2.2 Funções do produto</li><li>2.3 Características do usuário</li><li>2.4 Limitações gerais</li><li>2.5 Suposições e dependências</li></ul></li><li><b>3. Requisitos Específicos:</b> abrange os requisitos funcionais, não-funcionais e de interface.</li><li><b>4. Apêndices</b></li><li><b>5. Índice</b></li></ul>
---

O RUP 2002 sugere o modelo apresentado na Tabela 2.2 para o documento Especificação de Requisitos de Software [Rup02].

Tabela 2.2- Modelo do Documento de Requisitos sugerido no RUP 2002[Rup02]

<ul style="list-style-type: none"><li><b>1. Introdução</b></li><li><b>2. Descrição Geral</b></li><li><b>3. Requisitos Específicos</b><ul style="list-style-type: none"><li>3.1. Casos de Uso (Opcional)</li><li>3.2. Requisitos Suplementares (Requisitos Não-Funcionais)</li></ul></li><li><b>4. Informações de suporte (tabelas, índices, apêndices, etc).</b></li></ul>
--

Amador Duran, em sua proposta metodológica [Dur00], propõe a estrutura apresentada na Tabela 2.3 para o Documento de Requisitos. Um elemento importante deste documento é a matriz de rastreabilidade entre requisitos e objetivos, permitindo comprovar se, para todos os objetivos definidos para o sistema, há um requisito associado.

Tabela 2.3- Modelo do Documento de Requisitos proposto por Duran [Dur00]

<ul style="list-style-type: none"><li><b>1. Introdução</b></li><li><b>2. Participantes do Projeto</b></li><li><b>3. Descrição do Sistema Atual (Opcional)</b></li><li><b>4. Objetivos do Sistema</b></li><li><b>5. Catálogo de Requisitos do Sistema</b><ul style="list-style-type: none"><li>5.1 Requisitos</li><li>5.2 Requisitos Funcionais</li><li>5.3 Requisitos Não-Funcionais</li></ul></li><li><b>6. Matriz de Rastreabilidade Objetivos/Requisitos</b></li><li><b>7. Conflitos Pendentes de Resolução (Opcional)</b></li><li><b>8. Glossário de Termos (Opcional)</b></li><li><b>9. Apêndices (Opcional)</b></li></ul>
---

Naturalmente, as informações incluídas em um Documento de Requisitos dependem do tipo de software que está sendo desenvolvido e da abordagem de desenvolvimento adotada.

## **2.4. Tipos de Requisitos**

A construção de software deve considerar, além das funcionalidades essenciais que o software deve executar, o ambiente no qual ele está inserido e os aspectos relacionados com a estrutura e a política empresarial.

Assim, durante o processo de especificação dos requisitos, surge a necessidade de estabelecer o tipo de requisito que se está tratando, a fim de melhorar a compreensão das necessidades do cliente, bem como modelar melhor esta necessidade.

De forma geral, os requisitos podem ser classificados em três tipos distintos, que podem estar relacionados: *Funcionais*, *Não-Funcionais* e *Organizacionais* [Som98]. As seções seguintes detalham estes tipos de requisitos.

### **2.4.1. REQUISITOS FUNCIONAIS**

Os requisitos funcionais expressam funções ou serviços que um software deve ou pode ser capaz de executar ou fornecer. Eles descrevem as transformações a serem realizadas nas entradas de um sistema ou em um de seus componentes, a fim de que se produzam saídas.

É objetivo da Engenharia de Requisitos sistematizar o processo de definição destes requisitos de modo que seja gerado um compromisso entre os desenvolvedores e usuários/clientes, para que todos participem da geração e definição dos requisitos de um sistema. Existem diversos métodos, técnicas e ferramentas que ajudam os engenheiros de requisitos na obtenção de requisitos mais seguros e completos, permitindo compreender o problema, facilitar a obtenção das necessidades do usuário/cliente, validá-las com o usuário/cliente e descrever as especificações de requisitos.

Diversos autores propõem o uso de cenários para atingirem estes objetivos. É possível definir um cenário através de descrições de situações particulares do sistema atual ou de um sistema a ser desenvolvido, utilizando linguagem natural com estrutura simples, permitindo o entendimento do cliente/usuário [Kap00]. Existem algumas técnicas baseadas em cenários, tais como Casos de Uso, proposta por Ivar Jacobson em [Jac95] e CREWS, proposta por Jolita Ralyté e citada em [San02], que além de utilizarem a linguagem natural, também utilizam diagramas e notações gráficas. Neste trabalho, será dada maior importância aos Casos de Uso,

que têm sido adotados como integrantes do modelo padrão de linguagem de modelagem para a orientação a objetos (UML) e é uma das bases do RUP, que é orientado a casos de uso, conforme será visto no próximo capítulo.

Como parte do processo de construção de cenários, alguns autores sugerem o uso de glossários de termos, relacionados com os casos de uso. Como o glossário e os casos de uso são escritos em linguagem natural, há possibilidade de serem geradas ambigüidades e inconsistências. Para resolver este problema, Júlio Leite [Kap00] propõe a utilização de uma lista de termos efetivamente reduzida, mediante a utilização de um vocabulário bem definido do Universo de Informações (UdI), através do Léxico Ampliado da Linguagem (LAL).

A seguir detalhamos os conceitos do LAL, Cenários e Casos de Uso, que serviram como base para a proposta deste trabalho.

#### **2.4.1.1. Léxico Ampliado da Linguagem (LAL)**

O Léxico Ampliado da Linguagem (LAL) é um meta modelo projetado para ajudar a representar (e entender) os termos de linguagem usados na aplicação [Kap00]. Este modelo é centrado na idéia que uma descrição destes termos ajuda a compreender o Universo de Informações (UdI).

O principal objetivo do LAL é registrar a linguagem utilizada pelos atores do UdI, sem contudo se preocupar com a funcionalidade [Cys01-A]. O LAL do UdI é composto por entradas, onde cada entrada está associada a um símbolo (palavra ou frase) da linguagem do UdI. Cada símbolo pode possuir sinônimos e é descrito através de noções e impactos. As noções descrevem o significado e as relações fundamentais de existência do símbolo com outros símbolos (denotação). Os impactos descrevem os efeitos causados pelo uso, ou ocorrência, do símbolo no UdI. Dependendo do símbolo que descrevem, as entradas podem ser classificadas como sujeito, verbo, objeto e estado (predicativo).

Ao se descrever entradas no LAL, deve-se obedecer aos princípios de vocabulário mínimo e de circularidade. O princípio de vocabulário mínimo demanda que a utilização de símbolos externos ao LAL do UdI seja minimizada ao máximo. O princípio de circularidade implica na maximização da utilização de símbolos do LAL do UdI na descrição de símbolos. Os símbolos do LAL, que aparecem na descrição de símbolos, devem ser sublinhados. Como decorrência do princípio de circularidade, a forma natural de representação do LAL é pela utilização de uma estrutura de hipertexto. As entradas do LAL são os nós do hipertexto, enquanto que os símbolos que aparecem nas descrições de símbolos, são os elos do hipertexto.

Em [Gil00] e [Had99] é especificado o processo de construção do LAL, segundo proposta de Graciela Hadad [Had99]. Este modelo é executado em seis passos, conforme descrito na Tabela 2.4:

*Tabela 2.4-Etapas para a construção do LAL [Had99]*

<b>Passo</b>	<b>Etapa</b>	<b>Objetivo</b>
1	Entrevistas	Conhecer o vocabulário da aplicação.
2	Geração da lista de símbolos	Obter os diversos símbolos do Universo de Informações, sem se preocupar com as descrições dos mesmos.
3	Classificação dos símbolos	Sobre a lista preliminar de símbolos, obtida no passo anterior, elabora-se uma classificação específica para o domínio especificado, onde os símbolos são divididos em <b>sujeito, verbo, objeto e estado</b> .
4	Descrição dos símbolos	Descrever as noções e impactos de cada símbolo.
5	Validação com os clientes	Certificar que o conteúdo das descrições de cada símbolo correspondem à realidade do Universo de Informações.
6	Controle do LAL	Manter a consistência e homogeneidade do LAL.

Cysneiros, em [Cys01-A], descreve o processo de construção do LAL da seguinte forma:

- Inicialmente são identificados os símbolos do Universo de Informações (UdI). Para tal, deve-se identificar as palavras ou frases habitualmente usadas na linguagem do domínio da aplicação que pareçam ter um significado especial. Em geral, estas palavras ou frases são utilizadas com frequência por atores do UdI ou outras fontes de informação. Palavras ou frases que parecem sem sentido, ou fora de contexto, têm grande probabilidade de serem símbolos do UdI. As técnicas de elicitação mais adequadas para esta atividade são a observação, leitura de documentos e entrevista não estruturada. As heurísticas de identificação se resumem a procurar sujeitos, verbos, objetos e estados (predicativos) presentes em frases que surgem no UdI e que possuem um significado especial. Antes de utilizar as heurísticas, deve-se separar os períodos compostos em orações simples, transportar orações na voz passiva para a voz ativa e transformar as formas substantivas de verbos para a forma verbal correspondente. Observa-se que a transformação das formas substantivas de verbos para formas verbais correspondentes é extremamente importante para que se possa utilizar as heurísticas de construção de cenários a partir do LAL do UdI, propostas por Hadad [Had99].
- O próximo passo deste processo consiste na classificação dos símbolos. Neste passo, os símbolos são classificados em sujeitos, verbos, objetos e estados (predicativos) de frases que surgem no UdI.

- Após a classificação, estes símbolos devem ser descritos. Para realizar esta etapa é necessário criar entradas no LAL referentes aos símbolos identificados. Para criar uma entrada do LAL para um símbolo, deve-se descrever noções e impactos, seguindo as heurísticas de representação. Algumas destas heurísticas de representação são mostradas na Tabela 2.5. Nesta atividade, identificam-se os sinônimos dos símbolos descritos por entradas do LAL. Eventualmente, são identificados novos símbolos nesta atividade, bem como novas noções e impactos de outros símbolos.
- Conforme mostra a Tabela 2.5, o processo de construção do LAL é mantido sempre que surgirem alterações no Universo de Informações (UdI).

*Tabela 2.5 - Heurísticas de Representação [Cys01-A]*

<ul style="list-style-type: none"> <li>• Cada entrada tem zero ou mais sinônimos.</li> <li>• Cada entrada tem uma ou mais noções.</li> <li>• Cada entrada tem um ou mais impactos.</li> <li>• Escreva noções e impactos usando frases simples, que expressam uma única idéia.</li> <li>• A descrição de noções e impactos deve respeitar os princípios de circularidade e vocabulário mínimo.</li> <li>• Para uma entrada que descreve um símbolo classificado como sujeito: <ul style="list-style-type: none"> <li>❖ As noções devem dizer quem é o sujeito (pessoa ou coisa) do qual se diz alguma coisa;</li> <li>❖ Os impactos devem registrar as atividades que o sujeito realiza;</li> <li>❖ Os sinônimos devem conter termos que tenham o mesmo sentido do símbolo no UdI.</li> </ul> </li> <li>• Para uma entrada que descreve um símbolo classificado como verbo: <ul style="list-style-type: none"> <li>❖ As noções devem descrever sucintamente a ação e dizer quem é o sujeito (pessoa ou coisa) que realiza a ação, quando a ação acontece, e quais as informações necessárias para realizar a ação;</li> <li>❖ Os impactos devem registrar os procedimentos que fazem parte da ação, as condições/situações decorrentes da ação e outras ações que deverão ocorrer;</li> <li>❖ Os sinônimos devem registrar a forma substantivada, a voz passiva e a forma infinitiva do verbo.</li> </ul> </li> <li>• Para uma entrada que descreve um símbolo classificado como objeto: <ul style="list-style-type: none"> <li>❖ As noções devem definir o ser (pessoa ou coisa) que sofre ações;</li> <li>❖ Os impactos devem descrever as ações que podem ser aplicadas ao objeto;</li> <li>❖ Os sinônimos devem conter termos que tenham o mesmo sentido do símbolo no UdI.</li> </ul> </li> </ul>
---

### **2.4.1.2. Cenários**

Segundo Breitman [Bre00], um grande passo no processo de definição dos requisitos de uma aplicação reside no reconhecimento dos atores, entidades e ações que descrevem o macro sistema do qual esta aplicação fará parte. Cenários são uma opção para a descrição de situações do mundo real que envolvem agentes interagindo dentro de um determinado contexto. Esta interação é descrita através de ações enumeradas em um ou mais episódios. Cenários utilizam elementos conhecidos pelos clientes, facilitando tanto o processo de elicitação de requisitos quanto a validação dos mesmos.



Objetiva-se com o uso de cenários descrever as ações, em um ambiente, relacionadas a um sistema atual ou a um sistema a ser desenvolvido. Normalmente, a linguagem utilizada para estas descrições é a linguagem natural.

Cada cenário aborda um ou um pequeno número de possíveis iterações. O cenário começa com o esboço da iteração e, com a evolução do conhecimento dos requisitos são acrescentados detalhes para criar uma descrição completa dessa iteração. Em [Som03], Sommerville afirma que, de um modo geral, um cenário pode incluir:

- uma descrição do estado do sistema no início do cenário,
- uma descrição do fluxo normal de eventos,
- uma descrição do que pode sair errado e como lidar com isso,
- informações sobre outras atividades que possam estar em andamento ao mesmo tempo e,
- uma descrição do estado do sistema no final do cenário.

A Tabela 2.6 apresenta o modelo de cenários proposto por Júlio Leite [Cys01-A],[Lei00]. Nesta proposta os cenários são compostos de título, objetivo, contexto, recursos, atores, episódios e exceções. Na seção referente ao contexto, Leite considera que, pelo menos um, dos seguintes sub componentes deve estar presente: Localização Geográfica, Localização Temporal e Pré-Condições.

*Tabela 2.6 - Esquema para construção de cenários, por Júlio Leite [Cys01-A]*

1. Título: <Título do cenário, que serve também para identificá-lo>
2. Objetivo: <Meta que o cenário deve atingir, o cenário descreve o alcance de um objetivo>
3. Contexto: <Localização geográfica, temporal e estado inicial do processo (pré-condições)>
4. Recursos: <Meios de suporte, dispositivos e outras entidades passivas utilizadas pelos atores do cenário>
5. Atores: <Pessoas, sistemas ou estruturas organizacionais que desempenham algum papel no cenário>
6. Episódios: <Ações que detalham o cenário e apresentam seu comportamento.>

### **2.4.1.3. Casos de Uso**

Os casos de uso correspondem a uma poderosa técnica, baseada em cenários, para especificação de requisitos proposta por Jacobson, em 1993 [Bit03], [Bre00]. Segundo Breitman [Bre00], os cenários são instâncias de casos de uso. A notação em UML (*Unified Modeling Language*) utiliza casos de uso para descrever modelos de sistemas orientados a objetos [Bit03].

Um caso de uso descreve uma seqüência de passos/operações que um ator realiza quando interage com um sistema, visando realizar uma determinada tarefa ou alcançar um objetivo. Um ator representa qualquer elemento externo que interage com o sistema. Dessa forma, o aspecto comportamental de um sistema a ser desenvolvido pode ser descrito através de casos de uso. No entanto, a descrição dos casos de uso não trata a questão de como as interações entre o usuário e o sistema serão implementadas. Fases posteriores à etapa de engenharia de requisitos, tais como Projeto e Implementação, focam este aspecto.

Um caso de uso envolve uma situação de utilização do sistema por um ator. Nesta situação, vários caminhos podem ser seguidos dependendo do contexto na execução do sistema. Estes caminhos são os possíveis cenários do caso de uso. Considera-se que o caminho básico para realizar um caso de uso, sem problemas e sem erros em nenhum dos passos da seqüência, é denominado de cenário primário. Neste tipo de cenário, a execução dos passos para realizar a funcionalidade básica do caso de uso, é obtida com sucesso. Por outro lado, caminhos alternativos, bem como situações de erro, podem ser representados através de cenários secundários. Cenários secundários descrevem seqüências alternativas e de erros que podem ocorrer em um cenário primário associado com um caso de uso. Cenários secundários podem ser descritos separadamente ou como extensão da descrição de um cenário primário. Se um cenário secundário é bastante complexo e inclui um conjunto considerável de passos, é conveniente descrevê-lo separadamente.

A descrição de requisitos funcionais através de casos de uso apresenta algumas vantagens em relação à descrição meramente textual: são facilmente compreensíveis pelos usuários, facilitam a validação de requisitos e servem como base para a documentação dos usuários.

Os casos de uso têm uma representação gráfica chamada *Diagramas de Casos de Uso*, onde os atores são representados por pequenos bonecos e os casos de uso por elipses. A interação entre atores e casos de uso é realizada através de setas. Os *Diagramas de Casos de Uso* proporcionam uma visão global do conjunto de casos de uso de um sistema e das suas interação com os atores.

Algumas vezes convém estabelecer relacionamentos que permitam descrever diversos aspectos de comportamento entre casos de uso. Os relacionamentos apontados em UML incluem [Jac99],[Bit03]:

- **Relacionamento do tipo <<include>>**: quando for detectado no sistema um conjunto de passos comuns a vários casos de uso, pode-se criar um caso de uso com estes passos, com potencial para ser reutilizado por outros casos de uso. A idéia consiste em estabelecer que

diversos casos de uso do sistema podem fazer uso de um determinado caso de uso, quando necessário.

- **Relacionamento do tipo <<extend>>**: quando existe uma seqüência opcional ou condicional de passos que podem ser incluídas em um caso de uso. Esta seqüência de passos deve ser descrita em um caso de uso específico que poderá ser utilizado por outros casos de uso em certo ponto de sua execução. O uso do caso de uso estendido ocorre devido a uma situação de comportamento opcional ou condicional.
- **Relacionamento do tipo <<generalization>>**: generalização entre casos de uso tem o mesmo significado de generalização entre classes na orientação a objetos. Isto significa que um caso de uso “filho” herda o comportamento e estrutura do caso de uso “pai”. Considera-se que um caso de uso “filho” é uma especialização do caso de uso “pai”, podendo adicionar nova estrutura e comportamento bem como modificar o comportamento do caso de uso “pai”. Da mesma forma que é permitido o uso do mecanismo de generalização entre casos de uso, pode-se usar o relacionamento de generalização entre atores representados em diagramas de casos de uso.

O processo de construção de casos de uso inicia com a descoberta dos atores do sistema e prossegue com a descoberta do(s) caso(s) de uso associados com esses atores. O segundo passo consiste em definir os caminhos básicos (cenários primários) e, posteriormente, os caminhos alternativos (cenários secundários) para cada um dos casos de uso. O terceiro passo envolve revisar descrições de aspectos comportamentais, encontrando relacionamentos do tipo <<include>>, <<extend>> e <<generalization>>. Este processo é geralmente realizado adotando-se o princípio de desenvolvimento de software iterativo e incremental. Depois de definidos todos os casos de uso e atores do sistema, desenvolve-se um *Diagrama de Casos de Uso*, utilizando as notações UML, conforme apresentado na Figura 2.1.

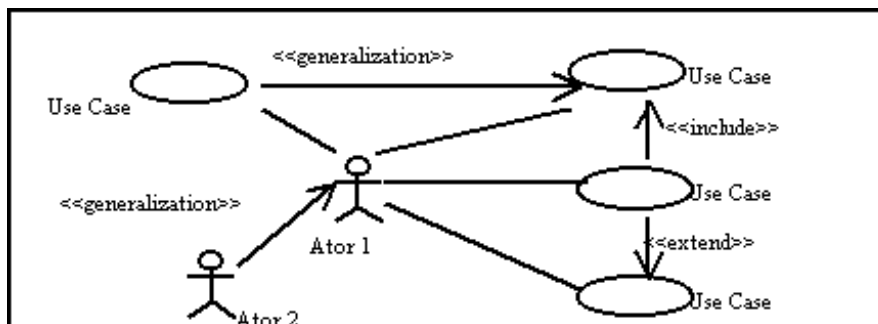


Figura 2.1- Notações para Casos de Uso em UML

Como é uma técnica amplamente aceita, diversas propostas utilizam os casos de uso concretamente. A seguir apresentamos algumas destas propostas.

Alistair Cockburn propõe, em [Coc00], o modelo (*template*) de especificação de casos de uso descrito na Tabela 2.7. Este modelo requer a definição explícita do objetivo do caso de uso bem como os níveis associados com estes objetivos. A demais informações presentes no *template* também são importantes para tornar a descrição de casos de uso o mais clara possível.

Tabela 2.7 - *Template de caso de uso proposto por Cockburn [Coc00]*

<p><b>Caso de Uso:</b> &lt;número&gt; &lt;&lt; o nome é um objetivo descrito com uma frase curta contendo um verbo na voz ativa &gt;&gt;</p> <p>-----</p> <p><b>INFORMAÇÃO CARACTERÍSTICA</b></p> <p><b>Objetivo no Contexto:</b> &lt;uma sentença mais longa do objetivo do caso de uso, caso necessário&gt;</p> <p><b>Escopo:</b> &lt;Qual sistema está sendo considerado (por exemplo, organização ou sistema computacional)&gt;</p> <p><b>Nível:</b> &lt;um dos tipos de objetivo: objetivo de contexto, objetivo de usuário ou objetivo de subfunção&gt;</p> <p><b>Pré-condições:</b> &lt;o que é necessário que já esteja satisfeito para realizar o use case &gt;</p> <p><b>Condição Final de Sucesso:</b> &lt;o que ocorre/muda após a obtenção do objetivo do use case&gt;</p> <p><b>Condição Final de Falha:</b> &lt;o que ocorre/muda se o objetivo é abandonado&gt;</p> <p><b>Ator Primário:</b> &lt;o nome do papel para o ator primário, ou descrição&gt;</p> <p>-----</p> <p><b>CENÁRIO PRINCIPAL DE SUCESSO</b></p> <p>&lt;coloque aqui os passos do cenário necessários para a obtenção do objetivo &gt;</p> <p>&lt;passo #&gt; &lt;descrição da ação &gt;</p> <p>-----</p> <p><b>EXTENSÕES</b></p> <p>&lt;coloque aqui as extensões, uma por vez, cada uma referenciando o passo associado no cenário principal &gt;</p> <p>&lt;passo alterado&gt; &lt;condição&gt; : &lt;ação ou sub-use case &gt;</p> <p>&lt;passo alterado &gt; &lt;condição&gt; : &lt;ação ou sub-use case &gt;</p> <p>-----</p> <p><b>INFORMAÇÃO RELACIONADA (opcional)</b></p> <p><b>Prioridade:</b> &lt;Quão crítico é o caso de uso para seu sistema/organização &gt;</p> <p><b>Desempenho alvo:</b> &lt;o total de tempo que este caso de uso pode demorar &gt;</p> <p><b>Frequência:</b> &lt;com que frequência espera-se que o caso de uso ocorra &gt;</p> <p><b>Caso de Uso Pai:</b> &lt;opcional, nome do caso de uso que inclui este &gt;</p> <p><b>Casos de Uso Subordinados:</b> &lt;opcional, ligações para sub-casos de uso &gt;</p> <p><b>Atores Secundários:</b> &lt;lista de outros atores(sistemas) necessários para realizar este caso de uso&gt;</p>
--

Amador Duran, em [Dur00], propõe o uso de planilhas para a especificação dos casos de uso das aplicações. O modelo (*template*) é apresentado na Tabela 2.8.

Tabela 2.8 - *Template de Casos de Uso proposto por Duran [Dur00]*

<b>RF-&lt;id&gt;</b>	<Nome descritivo do Requisito Funcional- Caso de Uso>
<b>Versão</b>	<Número da versão atual do documento>
<b>Autor</b>	Lista de autores
<b>Objetivos Associados</b>	OBJ-X <Nome do Objetivo >
<b>Requisitos Associados</b>	REQ-X <Nome do Requisito>

<b>Descrição</b>	<Descreve o comportamento do sistema com a realização do caso de uso>	
<b>Pré-Condição</b>	<Condições necessárias para a execução do caso de uso>	
<b>Seqüência Normal</b>	<i>Passo</i>	<i>Ação</i>
	<i>P&lt;n&gt;</i>	<Ação realizada no passo>
	<i>P&lt;n+1&gt;</i>	<Ação realizada no passo seguinte>
<b>Pós-Condição</b>	<Condições após a execução normal do caso de uso>	
<b>Exceções</b>	<i>Passo</i>	<i>Ação</i>
	<i>P&lt;n&gt;</i>	<Ação realizada no passo>
<b>Rendimento (tempo de execução previsto)</b>	<i>Passo</i>	<i>Tempo Máximo</i>
	<i>P&lt;n&gt;</i>	<unidade de tempo para a execução do passo do caso de uso>
<b>Freqüência Esperada</b>	<Número de vezes que o caso de uso é executado por unidade de tempo>	
<b>Importância</b>	<Importância do Caso de Uso>	
<b>Urgência</b>	<Urgência do Caso de Uso>	
<b>Estado</b>	<Estado do Caso de Uso>	
<b>Estabilidade</b>	<Estabilidade do Caso de Uso>	
<b>Comentários</b>	<Comentários adicionais>	

O Rational Unified Process (RUP) sugere um template para a especificação de casos de uso conforme apresentado na Tabela 2.9

*Tabela 2.9-Template de casos de uso proposto no RUP*

<ol style="list-style-type: none"> <li>1. <b>Identificação do Caso de Uso</b></li> <li>2. <b>Breve Descrição</b></li> <li>3. <b>Fluxos de Eventos</b> <ol style="list-style-type: none"> <li>3.1. Fluxo Básico</li> <li>3.2. Fluxos Alterantivos</li> </ol> </li> <li>4. <b>Objetivo</b></li> <li>5. <b>Requisitos Especiais (Não-Funcionais)</b></li> <li>6. <b>Pré-Condições</b></li> <li>7. <b>Pós-Condições</b></li> <li>8. <b>Extensões</b></li> </ol>
---

Apresentamos o conceito e as técnicas associadas aos requisitos funcionais. A próxima seção apresenta os requisitos não-funcionais.

## **2.4.2. REQUISITOS NÃO-FUNCIONAIS**

Os requisitos não funcionais (RNFs) não expressam nenhuma função a ser implementada em uma aplicação [Som98], mas definem condições de comportamento, restrições, ou atributos de qualidade para um software e/ou para o seu processo de desenvolvimento. Segurança, precisão, usabilidade, performance e manutenibilidade são exemplos de requisitos não-funcionais.

Segundo Cysneiros [Cys01-A], a distinção entre o que é um requisito funcional e o que é um não funcional nem sempre é clara. Parte da razão disto é que os RNFs estão sempre

relacionados a requisitos funcionais. Com base na definição acima pode-se dizer que um requisito funcional expressa algum tipo de transformação que tem lugar no software, enquanto um requisito não-funcional expressa como essa transformação irá se comportar ou que qualidades específicas ela deverá possuir.

Requisitos não-funcionais abordam importantes aspectos relacionados à qualidade de software, sendo essenciais para que softwares sejam bem sucedidos. O não tratamento destes requisitos resulta em: softwares com inconsistência e de baixa qualidade; clientes e desenvolvedores insatisfeitos; tempo e custo de desenvolvimento além dos previstos devido à necessidade de se consertar softwares que não foram desenvolvidos sob a ótica da utilização destes requisitos.

Requisitos não-funcionais são, normalmente, subjetivos pois podem ser percebidos, interpretados e conceituados de forma diferente por diferentes pessoas. É verdade que os requisitos funcionais também sofrem do problema de diferentes conteúdos advindos de diferentes pontos de vista, porém, como os requisitos não-funcionais são por natureza mais abstratos e uma vez que são comumente relacionados de maneira breve e vaga, este problema é potencializado [Cys01-A].

É importante observar que os requisitos não-funcionais freqüentemente interagem entre si, uma vez que a tentativa de satisfazer um destes requisitos pode prejudicar ou ajudar a satisfazer outros. Como vários requisitos não-funcionais possuem efeitos de natureza global nos softwares, uma solução localizada pode não ser adequada.

Diferentes formas de classificar requisitos não-funcionais têm sido propostas. Sommerville, por exemplo, classifica os requisitos não-funcionais em requisitos de produtos, processos e externos, conforme apresentado na Figura 2.2 [Som98].

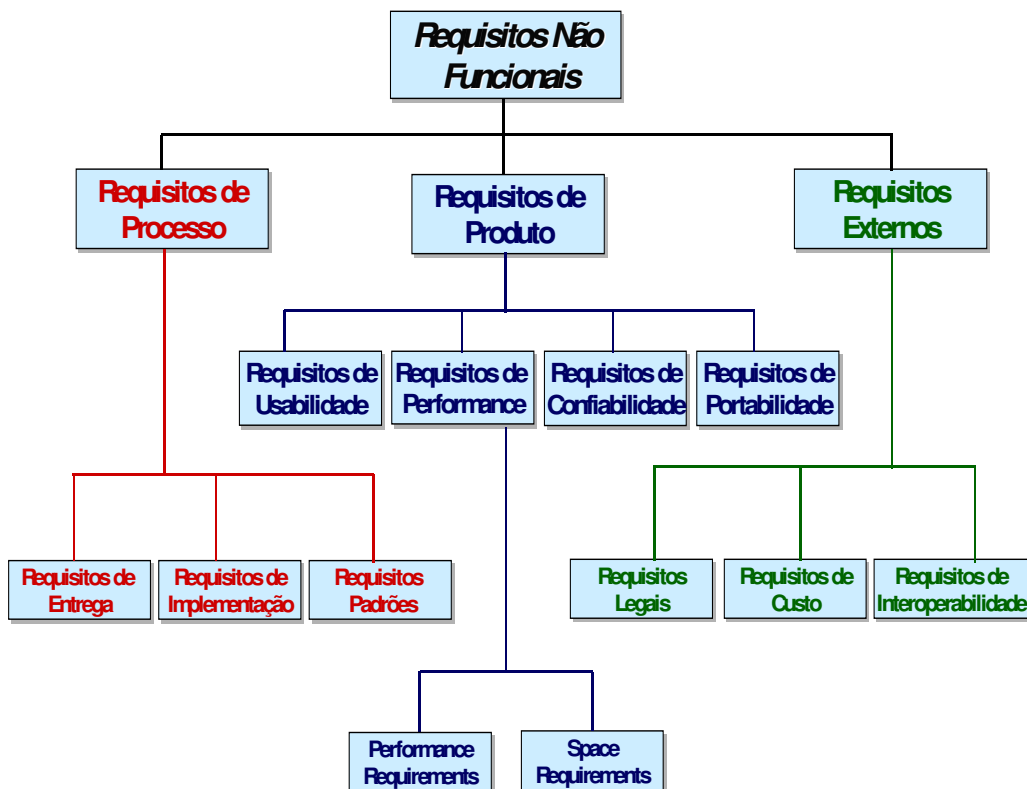


Figura 2.2- Classificação dos RNF, segundo Sommerville [Som98]

Cysneiros, em [Cys01-A], propõe uma taxonomia que classifica os RNFs em primários e específicos, conforme apresentado na Figura 2.3. RNFs primários são aqueles mais abstratos que representam propriedades como: Confiabilidade, Rastreabilidade, Precisão, Segurança. Já os RNFs específicos são decomposições que se seguem aos RNFs primários e tendem a diminuir o nível de abstração de um determinado RNF, e portanto, atingir um nível de granularidade no tratamento da informação mais detalhado.

Um exemplo desta classificação seria o RNF Primário Confiabilidade que pode ser decomposto em validação, autorização e entrega do software.

RNFs primários podem ser decompostos em outros RNFs secundários até que se chegue à operacionalização dos RNFs. Uma operacionalização de um RNF é uma ação ou informação que irá garantir a satisfação do RNF.

Segundo Cysneiros, em [Cys01-A], existem duas abordagens para o tratamento de requisitos não-funcionais:

- Abordagens orientadas a processo: integra o esforço de descrever requisitos não-funcionais durante o processo de desenvolvimento.

- Abordagens orientadas a produto: avalia o grau que o produto final atende a determinados requisitos não-funcionais.

Não existe uma abordagem melhor que a outra, elas são complementares e devem ser usadas para obter sistemas que de fato atendam aos requisitos não-funcionais dos stakeholders. Durante o estágio inicial de análise de requisitos é recomendável usar abordagens de processo, já abordagens orientadas a produto são indicadas quando os requisitos estão bem definidos e podem ser especificados em termos de funcionalidades e fatores qualitativos mensuráveis.

Neste trabalho focalizaremos a abordagem orientada a processos onde ao invés de avaliar a qualidade do produto final, a ênfase é dada em orientar o processo de desenvolvimento do sistema em relação aos requisitos não-funcionais que ele precisa atender. As decisões tomadas durante o projeto podem afetar de forma positiva ou negativa requisitos não-funcionais.

Na próxima seção será analisado o modelo **NFR-Framework**, proposto por Chung [Chu00]. É um modelo orientado a processo que fornece uma representação sistemática e global dos requisitos não-funcionais. Trata requisitos não-funcionais de forma explícita, como metas a serem atingidas, baseado na análise destes requisitos.



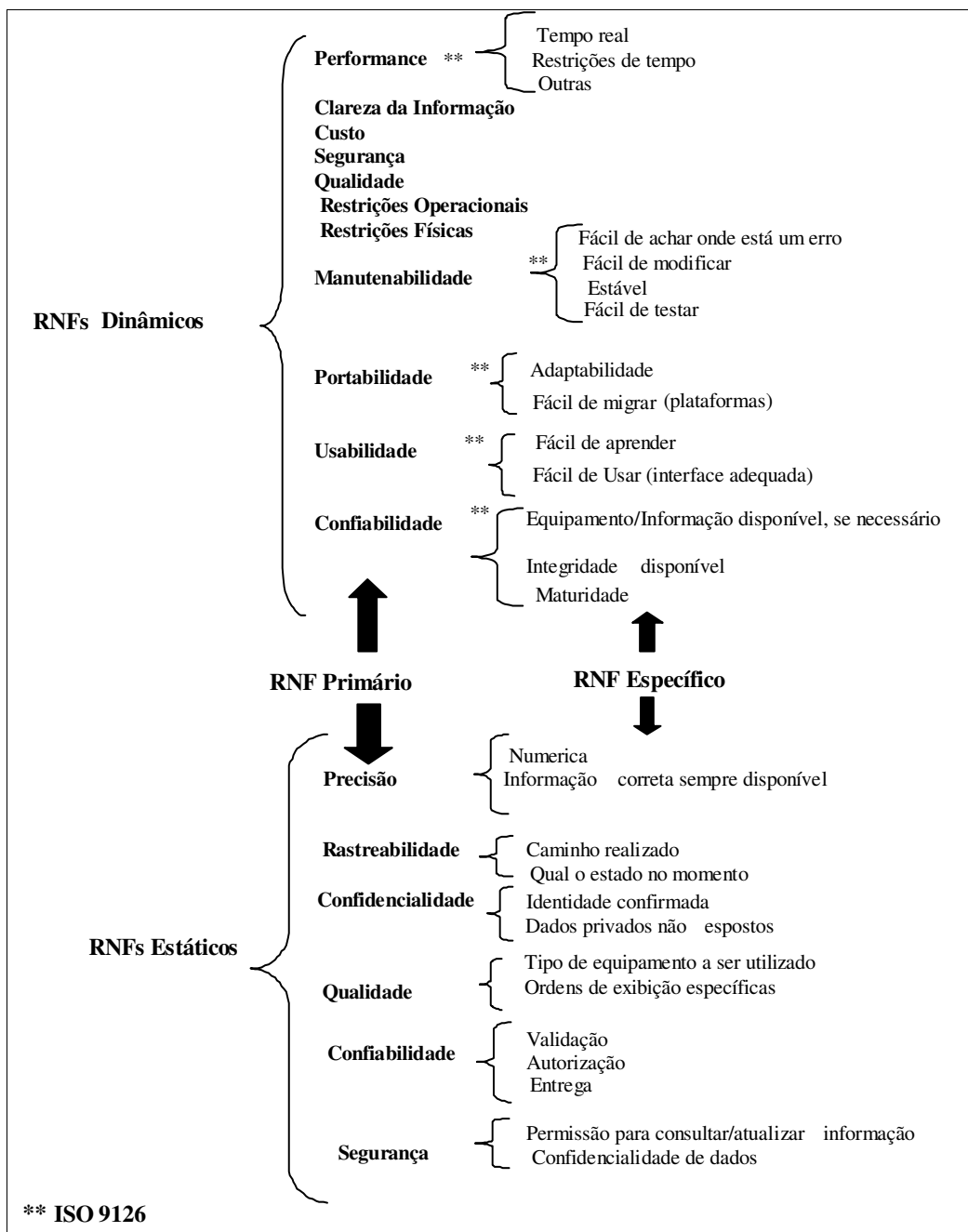


Figura 2.3 - Uma Taxonomia para RNFs [Cys01-A]

### 2.4.2.1. O NFR-Framework

A elicitação de Requisitos Não-Funcionais (RNFs) demanda o uso de uma notação para representá-los de modo que a informação não se perca no tempo. Permite tratar os RNFs de uma maneira organizada, facilitando lidar com as interdependências. Como a idéia básica é que os RNFs sejam elicitados desde as primeiras etapas do processo de desenvolvimento de software, há necessidade de representar estes RNFs.

Segundo o framework, os RNFs devem ser representados como metas (chamados *softgoals*), a serem satisfeitas, ou seja, eles são metas que precisam ser elaboradas, clarificadas e priorizadas. Cada meta é decomposta em subseqüentes submetas até que são encontradas as operacionalizações<sup>1</sup>. Para auxiliar a encontra estas operacionalizações, o engenheiro de requisitos pode utilizar catálogos, que são artefatos que armazenam conhecimento sobre requisitos não-funcionais, acumulados em experiências anteriores [Chu00].

Para representarmos os RNF's, será utilizado o grafo de RNF's proposto por Chung [Chu00]. Neste grafo, as metas são chamadas *softgoals* e são representadas por pequenas nuvens. A Figura 2.4, extraída de [Chu00], mostra um exemplo da decomposição do RNF *Segurança* aplicado à conta corrente como parte de um estudo de caso de automação bancária.

Ao decompor um RNF, são verificadas quais as decomposições possíveis para o mesmo. No caso do RNF *Segurança*, por exemplo, integridade, confidencialidade e disponibilidade podem ser possíveis decomposições desta meta. Neste exemplo pode-se ver que o RNF *Segurança*, quando analisado sob a ótica de contas correntes, apresenta 3 possíveis decomposições em *Integridade*, *Confidencialidade* e *Disponibilidade*. Percebe-se ainda, que a submeta *integridade* é novamente decomposta em sub-metas de *completude* e *precisão*. As outras submetas não são aqui decompostas para tornar o grafo simples de ser visualizado como um exemplo. Em uma abordagem completa todas seriam decompostas.

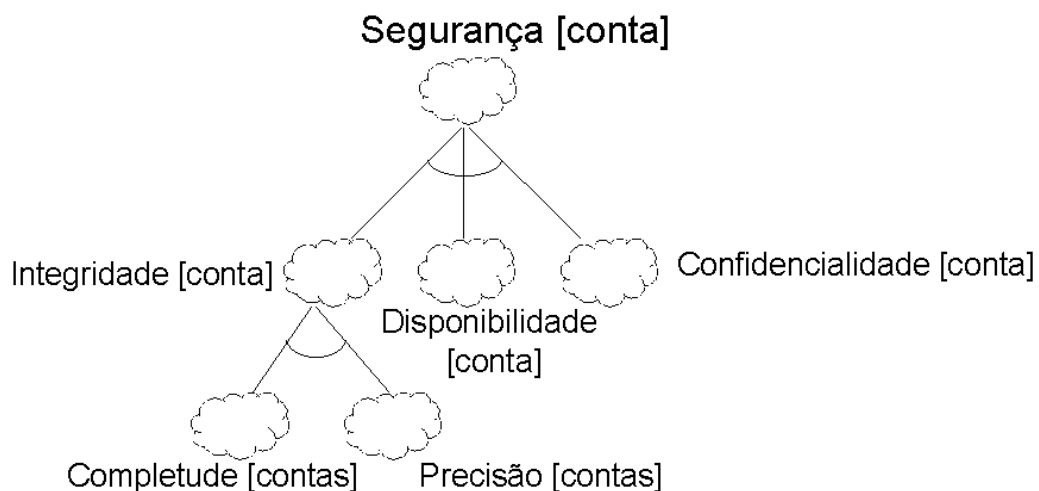


Figura 2.4- Exemplo de Decomposição do Grafo de RNF

<sup>1</sup> Uma operacionalização corresponde a ações ou atributos que claramente identifiquem o que é necessário para satisfazer a meta principal.

A Figura 2.5[Chu00], mostra um exemplo de refinamento até encontrarmos as operacionalizações necessárias e suficientes. Deve-se observar que a submeta decomposta é a de *Confidencialidade*. Esta, por sua vez, é refinada na submeta *Autorizar Acesso*, para acesso às informações sobre a conta. Vê-se então, que esta submeta é novamente decomposta em *Validação com Regras de Elegibilidade, Identificação de Usuários e Autenticação de Acesso de Usuário*, onde esta última é ainda decomposta em *Usar P.I.N., Comparar Assinatura e Pedir Identificação Adicional*. No contexto do desenvolvimento deste sistema, estes últimos refinamentos foram considerados suficientes para expressar as ações e dados necessários para satisfazer ao RNF *Segurança*. O momento de parar a decomposição é uma decisão pessoal e não há regras fixas que determinem um limite.

É importante observar que todas as três submetas filhas de *Autorizar Acesso* estão ligadas por um arco. Isto denota que as três são contribuições do tipo “E”, logo, para que a submeta seja satisfeita todas as três filhas terão de ser satisfeitas também.

Por sua vez, a submeta *Autenticação de Acesso* possui outras três submetas filhas que são ligadas por um arco duplo, denotando uma contribuição do tipo “OU”, assim, se uma das três filhas for satisfeita, a submeta imediatamente superior será satisfeita.

Neste novo grafo aparece ainda um outro RNF: *Performance*, que também é decomposto até que as operacionalizações *Usar formato descomprimido e Usar indexação* sejam encontradas.

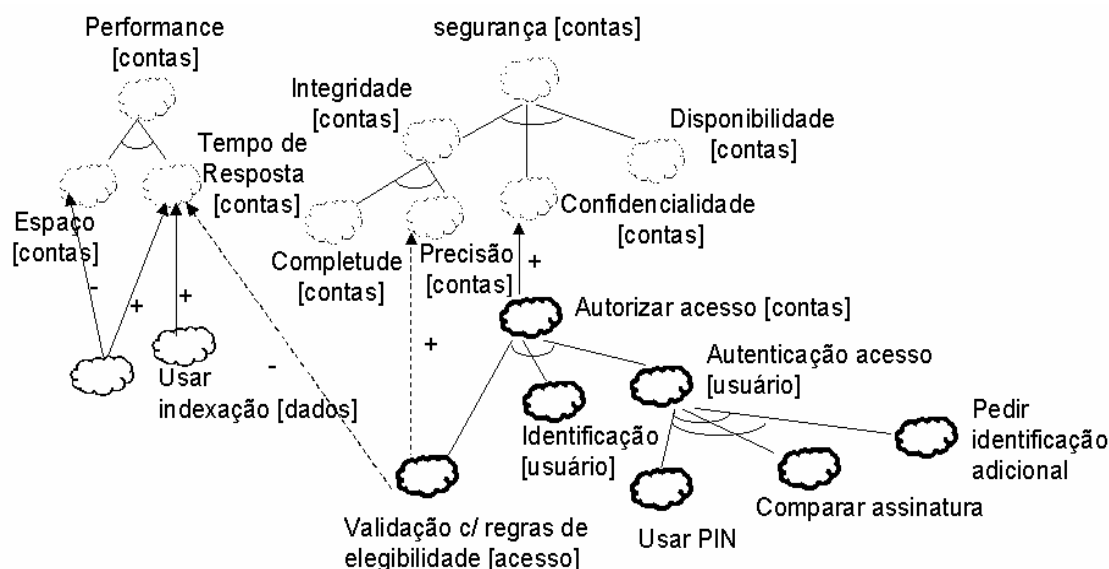


Figura 2.5- Grafo de RNFs Decomposto até suas Operacionalizações

Uma análise mais detalhada deste grafo apresenta algumas interdependências entre submetas, algumas positivas representadas com um sinal “+” e outras negativas, representadas com o sinal “-” A Figura 2.6 mostra estas interdependências. Observa-se que a operacionalização *Validar Acesso* com regras de elegibilidade contribui positivamente para a satisfação da submeta *Precisão* e negativamente para *Tempo de Resposta*, do RNF *Performance*. Outro ponto visto é um impacto negativo da operacionalização *Pedir Identificação Adicional* no RNF *Usabilidade - Amigável* no acesso a contas correntes.

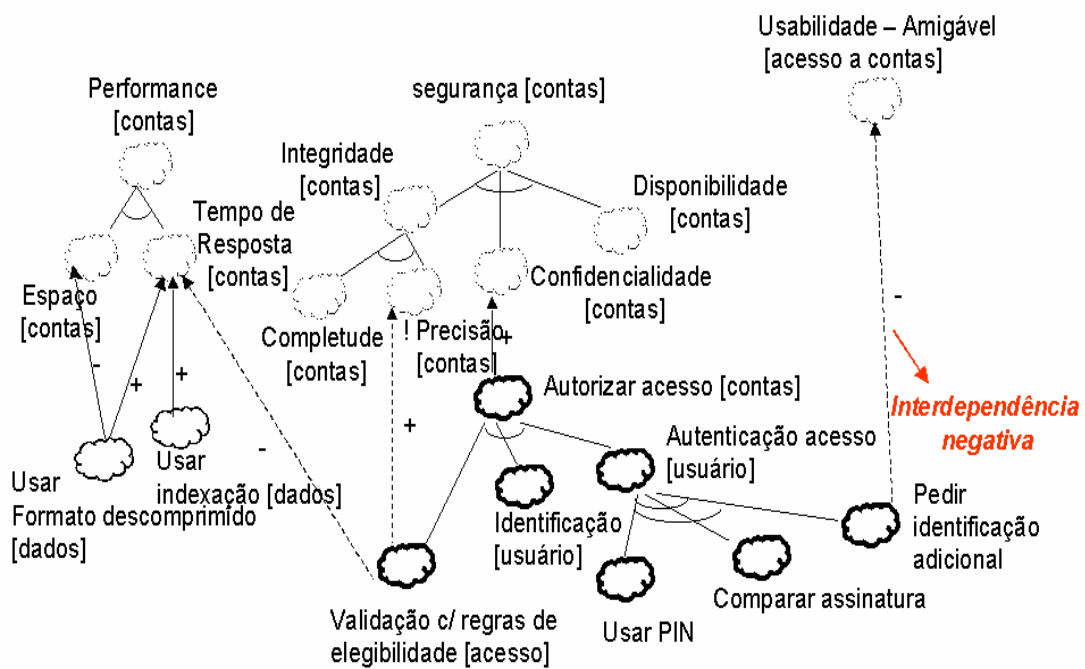


Figura 2.6- Exemplo de Grafo com Interdependências

Uma vez que as interdependências negativas são detectadas, é necessário tratar os conflitos que elas representam, ou seja, avaliar quais metas e submetas deverão não ser satisfeitas ou satisfeitas apenas parcialmente, ou ainda, se apesar das contribuições negativas todas as metas e submetas podem ser mantidas. A Figura 2.7 mostra o grafo após o engenheiro de requisitos responsável ter tomado as decisões. Neste grafo o símbolo “√” representa que a operacionalização será implementada enquanto o “X”, representa operacionalizações rejeitadas devido a seus impactos negativos em outras metas de maior relevância para a aplicação.

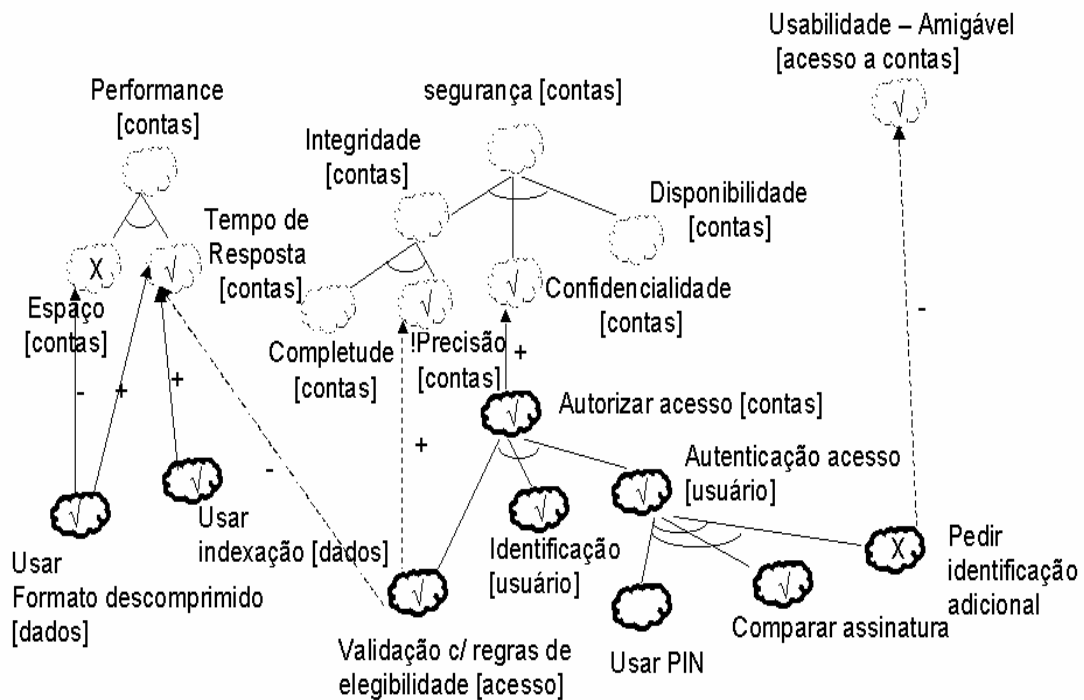


Figura 2.7- Grafo de RNFs com Interdependências Analisadas

### 2.4.3. REQUISITOS ORGANIZACIONAIS

Requisitos organizacionais correspondem aos objetivos organizacionais a serem atingidos com a construção de um novo software. No processo de construção de software, é necessário modelar e analisar os interesses dos stakeholders e como eles podem ser atendidos.

A captura de requisitos é uma fase crítica no desenvolvimento de software visto que ela não lida apenas com os conhecimentos técnicos, mas também com questões organizacionais, gerenciais, econômicas e sociais. Sendo assim, já é um consenso que a especificação de requisitos deve incluir, não apenas especificações de software, mas também modelos de negócios e outros tipos de informações que descrevam o contexto sobre o qual a aplicação irá operar [Cas00-B].

Existe uma preocupação na engenharia de requisitos que é elicitar e especificar os requisitos para um novo sistema de software da melhor forma possível. O sucesso de sistemas de software depende, entre outros aspectos, do grau de satisfação dos objetivos e estratégias organizacionais traçados pela empresa, haja vista que os softwares visam atender ou colaborar na obtenção de metas e objetivos previamente definidos pela organização. Além disso, o trabalho de engenheiros de requisitos e demais desenvolvedores é facilitado a partir do momento em que há um pleno conhecimento do ambiente organizacional. Este conhecimento inclui: comportamento,

motivações, intenções e objetivos dos atores da organização bem como descrições técnicas dos processos organizacionais e de que forma atores participam, integram e conduzem estes processos.

Várias técnicas têm sido propostas com o intuito de modelar ambientes organizacionais. Entre as principais técnicas existem *i\**, proposta por Eric Yu [Yu95], a Técnica de Bubenko [Bub93] e a Técnica KAOS [Dar93]. Pretende-se, com uso destas técnicas, representar o conhecimento a respeito da organização e permitir que este conhecimento seja utilizado e integrado efetivamente no processo de engenharia de requisitos. Neste trabalho, não serão apresentadas descrições das técnicas de modelagem organizacional por não serem escopo dessa dissertação.

## **2.5. O Processo de Engenharia de Requisitos**

Um processo é um conjunto de passos, parcialmente ordenados, constituídos por atividades, métodos, práticas e transformações, usadas para atingir uma meta [Pad01]. A etapa de requisitos de um processo de construção de software reúne as atividades que visam obter o enunciado completo, claro e preciso dos requisitos de um produto de software. Estes requisitos devem ser levantados pela equipe do projeto em conjunto com representantes do cliente, usuários chave e outros especialistas da área da aplicação. O conjunto de técnicas empregadas para levantar, detalhar, documentar e validar os requisitos de um produto formam a Engenharia de Requisitos. A realização de uma Engenharia de Requisitos efetiva é um passo fundamental para o desenvolvimento de um bom produto, onde a satisfação do usuário deve ser o principal objetivo a ser atingido [Ubi01].

Uma descrição completa do processo de requisitos deve incluir as atividades que são executadas, o responsável por cada uma das atividades, suas entradas e saídas e as ferramentas usadas para suportar a engenharia de requisitos.

Neste trabalho, detalharemos o processo genérico de Engenharia de Requisitos, proposto por Kotonya e Sommerville, em [Som98], que descreve as atividades Elicitação, Análise, Documentação e Validação como sendo as fases do processo de requisitos, conforme apresentado na Figura 2.8. A seleção deste processo como âncora para nossa pesquisa deve-se ao fato do mesmo ser amplamente aceito na comunidade acadêmica, sendo, inclusive referenciado em diversos trabalhos, artigos e livros publicados.

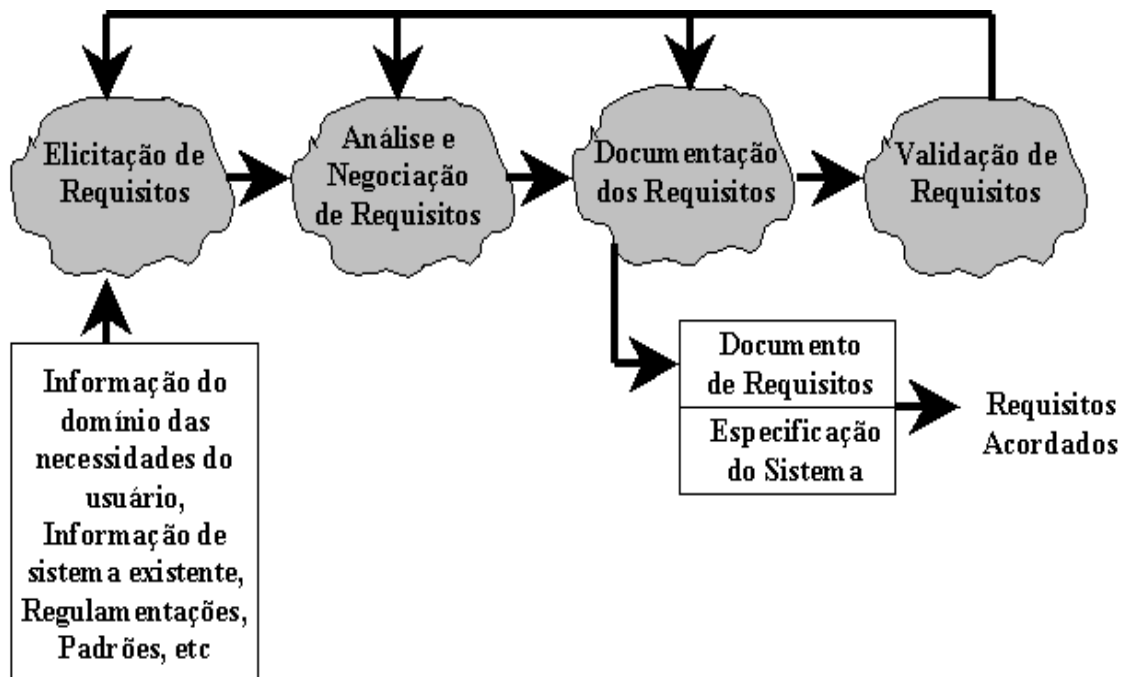


Figura 2.8 - O processo genérico de Engenharia de Requisitos [Som98]

A Figura 2.9 mostra essas atividades representadas num modelo espiral, no qual cada atividade do processo é repetida até que seja tomada a decisão de que o documento de requisitos pode ser aceito. Além disso, as mudanças de requisitos são parte do processo de Gerenciamento de Requisitos. Apesar dessas atividades serem normalmente descritas independentemente e numa ordem particular, na prática, elas consistem de processos iterativos e inter-relacionados que podem cobrir todo o ciclo de vida do desenvolvimento de sistemas de software.

As subseções seguintes detalham as fases do processo genérico de Engenharia de Requisitos.

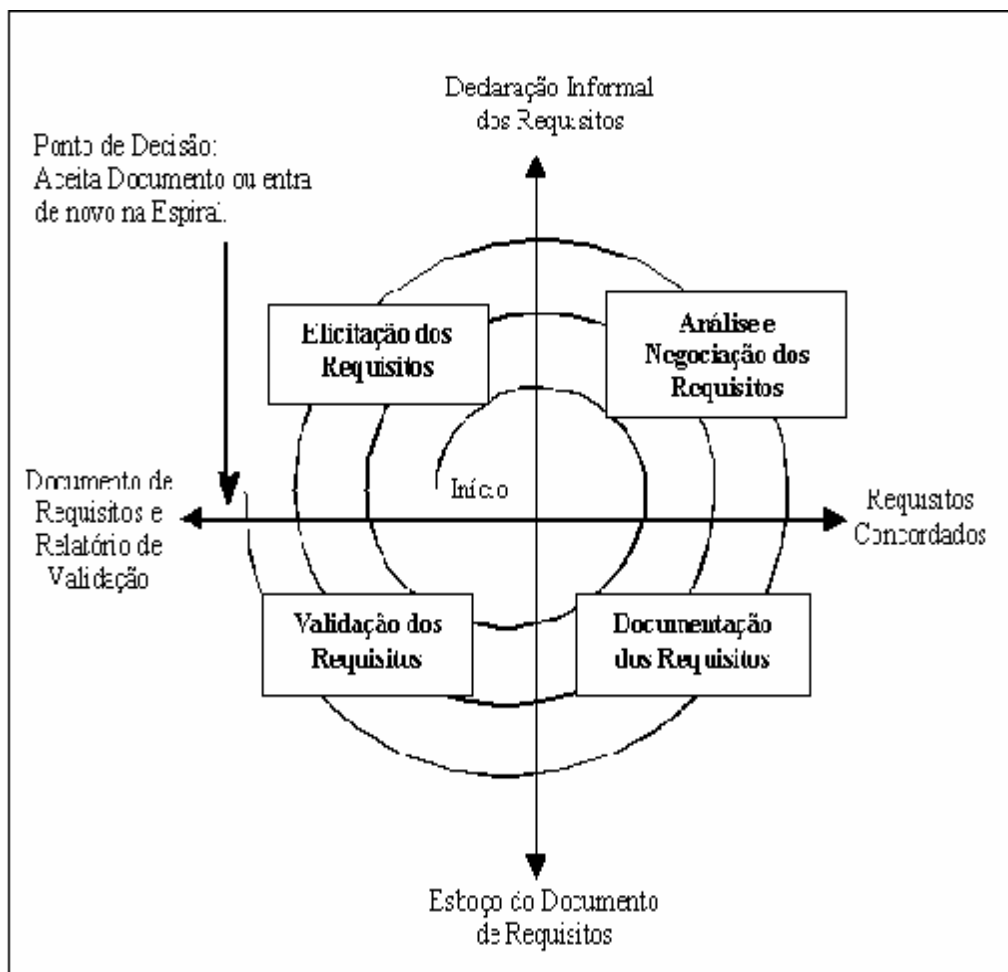


Figura 2.9- Modelo Espiral do Processo de Engenharia de Requisitos

### 2.5.1. ELICITAÇÃO DE REQUISITOS

Elicitar Requisitos é a atividade relacionada com a identificação dos requisitos do sistema, a partir de:

- comunicação com clientes, usuários finais e demais interessados no desenvolvimento do sistema;
- análise da organização, do domínio da aplicação e dos processos organizacionais;
- consultas aos documentos existentes e pesquisas de mercado.

Assim, cabe à elicitação a tarefa de identificar os fatos que compõem os requisitos do sistema, de forma a prover o mais correto entendimento do que é esperado do sistema de software e identificar as necessidades das diferentes classes de usuários [Sha99].



Neste processo de descoberta, o foco principal está no cliente/usuário que requisita e/ou utilizará o sistema. As maiores dificuldades que surgem não são computacionais, mas de comunicação, pois não é trivial extrair do usuário suas reais necessidades.

Existem diversas técnicas para a elicitación de requisitos e a escolha da técnica apropriada depende do tempo e dos recursos disponíveis. Algumas das técnicas existentes na literatura são [Som98]:

- **Entrevistas** – técnica de elicitación muito utilizada, pois é a forma de comunicação mais natural entre as pessoas. São identificadas três fases para a realização de entrevistas: preparação, realização e análise da entrevista [Som98];
- **JAD (Joint Application Development, Desenvolvimento Conjunto de Aplicações)** – É uma alternativa a entrevistas individuais, que acontece em um conjunto de reuniões em grupo durante um período de 2 a 4 dias. Estas reuniões ajudam os clientes e usuários finais a formularem os problemas e explorar possíveis soluções. Esta técnica tem quatro princípios: (i) dinâmica de grupo, (ii) uso de ajudas visuais para melhorar a comunicação, (iii) manter um processo organizado e racional e (iv) produção da documentação no decorrer da reunião [Dur00];
- **Brainstorming** - Como o JAD, também é uma técnica de elicitación em grupo. Seu objetivo é a produção de idéias e soluções em um ambiente livre de críticas. A diferença entre JAD e Brainstorming é que este último exige menos estruturação e organização que o primeiro [Dur00];
- **Cenários** – Usa cenários para representar as tarefas que os usuários executam normalmente e aquelas que eles desejam executar [Lei97]. Os usuários finais simulam sua interação com o sistema através da descrição de cenários, onde eles explicam, ao engenheiro de requisitos, o que estão fazendo e que informação eles necessitam do sistema para executar a tarefa descrita no cenário [Som98];
- **Prototipagem** – Desenvolve-se um protótipo da aplicação que é utilizado para elicitación requisitos. É muito usado quando há um alto grau de incerteza ou quando é necessário um rápido feedback dos usuários [Som98];
- **Etnografia e Análise Social** – surgiram como uma alternativa para as técnicas tradicionais [Som98]. Este tipo de técnica tem como base a observação, descrição e análise detalhadas das práticas de trabalho em uma organização. Facilita a familiarização dos engenheiros de requisitos com o ambiente e linguagem dos usuários, propiciando o

desenvolvimento dos documentos de requisitos com menos ambigüidades, incompreensões e incompletudes [Som98];

- ***Outras Técnicas tradicionais*** – incluem o uso de questionários e análise da documentação existente [Som98].

O processo de elicitación é uma atividade complexa e quase sempre requer uma quantidade considerável de tempo e recursos.

Amador Duran, em [Dur00], apresenta uma proposta metodológica para a elicitación de requisitos, onde sete tarefas são contempladas:

- obter informação sobre o domínio do problema e sistema atual,
- preparar e realizar as sessões de elicitación/negociação,
- identificar/Revisar os objetivos do sistema,
- identificar/Revisar os requisitos de informação (organizacionais),
- identificar/Revisar os requisitos funcionais,
- identificar/Revisar os requisitos não-funcionais,
- priorizar objetivos e requisitos.

As atividades propostas por Duran são fundamentais em um processo de Engenharia de Requisitos. Vê-se que é necessário ter uma metodologia para elicitación de requisitos, de forma a garantir que o sistema final atenda às expectativas e necessidades dos usuários.

### **2.5.2. ANÁLISE E NEGOCIAÇÃO DE REQUISITOS**

Os requisitos são analisados detalhadamente e os representantes dos grupos de stakeholders devem negociá-los de forma a chegarem a um consenso.

As atividades de análise e negociação de requisitos são, na maioria das vezes, executadas de forma paralela ou intercalada. Durante o processo de análise, os requisitos são analisados, a fim de detectar inconsistências, omissões e redundâncias. Na análise, a preocupação está em descobrir os requisitos que realmente são necessários e que o usuário deseja.

Durante o processo de análise de requisitos são identificados os problemas com os requisitos, tais como: requisitos conflitantes, ambíguos e duplicados. A partir dos requisitos com problemas, os usuários/clientes devem discutir, priorizar e negociar esses requisitos até que cheguem a um consenso sobre as alterações e simplificações necessárias. O processo de negociação de requisitos tenta resolver conflitos entre usuários/clientes sem necessariamente comprometer a satisfação dos objetivos de cada um. Em geral, os modelos de negociação

identificam as principais necessidades de cada usuário, onde são estabelecidas prioridades aos requisitos. Em seguida, os requisitos são analisados com a finalidade de se tentar garantir que os mais críticos sejam atendidos prioritariamente.

### **2.5.3. DOCUMENTAÇÃO DOS REQUISITOS**

Os requisitos são documentados e formalizados em um documento de requisitos. Como já foi apresentado na Seção 2.3, este documento deve ser entendível por todos os envolvidos no processo da engenharia de requisitos, já que ele representa um contrato entre usuários e desenvolvedores. O documento de requisitos é o meio através do qual é possível descrever as restrições quanto às características do produto e quanto ao processo de desenvolvimento, à interface com outras aplicações, à descrição sobre o domínio e às informações de suporte ao conhecimento do problema.

### **2.5.4. VERIFICAÇÃO E VALIDAÇÃO DE REQUISITOS**

A validação de requisitos tem o objetivo de mostrar se os requisitos realmente definem o que os usuários desejam do sistema [Som03]. Nesta atividade, os requisitos são cuidadosamente avaliados, com ênfase na consistência e completude. Possíveis problemas devem ser identificados antes que o documento produzido sirva como base para o desenvolvimento do sistema. Através da validação de requisitos certifica-se que o documento de requisitos é consistente com as necessidades dos usuários e clientes [Som98].

A validação de requisitos é importante porque a ocorrência de erros no documento de requisitos pode levar a grandes custos relacionados ao retrabalho. O custo de fazer uma modificação no sistema devido a um problema com requisitos, é muito maior que reparar erros de projeto ou codificação. A razão disto é que uma mudança nos requisitos, em geral, implica em mudanças no projeto e implementação. [Som03].

Durante o processo de validação dos requisitos, diferentes tipos de verificação devem ser realizadas sobre os requisitos documentados. Segundo Duran [Dur00], por verificação se entende o conjunto de atividades relacionadas com a qualidade da especificação dos requisitos. Sommerville, em [Som03], destaca os seguintes tipos de verificações:

- **Verificação de Validade:** Verifica se o conjunto de requisitos representa a solução esperada pela comunidade de usuários.

- **Verificação de Consistência:** Verifica se há conflito entre os requisitos no documento de requisitos.
- **Verificação de Completude:** Verifica se o documento de requisitos contempla todas as funções e restrições exigidas pelos usuários do sistema.
- **Verificação de Realismo:** Verifica se os requisitos realmente podem ser implementados, considerando o orçamento e prazo previstos.
- **Facilidade de Verificação:** Verifica se os requisitos são escritos de modo que possam ser verificados.

Outro grande desafio durante a validação de requisitos é demonstrar que a especificação dos requisitos do sistema está correta. Existem várias técnicas que podem ser aplicadas (em conjunto ou individualmente) para suportar a validação de requisitos [Som98],[Som03]:

- **Revisões dos requisitos:** Consiste na revisão dos requisitos por um grupo de pessoas que analisam, discutem e apontam caminhos para solucionar os problemas encontrados. A partir dessas descobertas, pode-se adotar medidas que solucionem os conflitos. Problemas típicos encontrados nas revisões incluem: falta de completude das descrições dos requisitos, ambigüidade e a não obediência a padrões da organização.
- **Prototipagem:** protótipos da interface da aplicação são desenvolvidos com o intuito de permitir uma melhor representação dos requisitos de um sistema. Com o desenvolvimento de protótipos, os usuários e clientes podem ter uma idéia antecipada de como o sistema executável funcionará. É observado que a técnica de prototipagem geralmente é usada para ajudar nas atividades de elicitação e análise de requisitos. Contudo, ela também pode ser usada para validar os requisitos. Após o documento de requisitos já ter sido definido e acordado, pode-se aperfeiçoar o protótipo desenvolvido na elicitação e análise e utilizá-lo para validar os requisitos, com um auxílio mais efetivo dos usuários/clientes.
- **Geração de Casos de Testes:** Podem ser desenvolvidos casos de teste para os requisitos, já na fase de validação de requisitos. Pode-se adotar, por exemplo, técnicas baseadas em cenários para elicitar e analisar requisitos e criar casos de teste para os cenários desenvolvidos. Se surgirem dificuldades para criar os casos de teste bem como para executá-los através de simulação, há uma grande probabilidade de existirem problemas com os requisitos. É menos oneroso e traumático descobrir estes problemas na atividade

de validação do processo de engenharia de requisitos do que em etapas finais do processo de desenvolvimento de software.

- **Validação de modelos:** geralmente, quando documentamos requisitos, os mesmos são descritos através de linguagem natural e diagramas. Tipicamente, cria-se um documento em linguagem natural descrevendo os requisitos do sistema. Para detalhar melhor o funcionamento do sistema são desenvolvidos modelos do sistema. O desenvolvimento destes modelos está associado à abordagem de desenvolvimento de software adotada. Estes modelos podem ser modelos de fluxo de dados, modelos de dados semânticos ou modelos inerentes ao paradigma de desenvolvimento, por exemplo, orientação a objetos ou métodos formais. Estes modelos precisam ser validados para demonstrar que os mesmos são consistentes e representam os reais requisitos dos usuários.
- **Walkthroughs:** É uma técnica de revisão de produtos, associada com inspeções dos documentos [Dur00]. O objetivo de uma sessão de *walkthrough* é encontrar conflitos no produto que está sendo revisado e colher soluções alternativas entre os participantes. Como descreve Duran [Dur00], existem três fases principais nesta técnica: (i) Preparação: os participantes recebem antecipadamente o material a ser revisado, (ii) Revisão: sessão onde o produto é analisado, com detalhes, e os participantes levantam questões oportunas, (iii) Conclusão: Com o término da reunião de revisão, são gerados documentos formais com informações referente á sessão ocorrida. Durante a reunião, o Engenheiro de Requisitos apresenta cada requisito ao grupo com o objetivo de gerar discussões sobre os mesmos para que problemas possam ser identificados. Pode-se distinguir os seguintes tipos de participantes (ou papéis) em uma sessão de walkthrough:
  - **Autor:** Responsável pela elaboração do documento a ser avaliado na reunião. Deve assegurar-se que o documento não contenha falhas ortográficas, nem defeitos que possam desviar a atenção dos participantes durante a sessão. Deve estar presente para esclarecer as dúvidas que venham a surgir e entender o que deve ser incorporado ou alterado no documento original.
  - **Apresentador:** Responsável por organizar e dirigir a sessão. Deve entregar a documentação aos revisores (com antecedência), eleger o moderador e o registrador, apresentar uma visão geral do produto e resumir o resultado da última reunião de validação, caso já tenha ocorrido. Em alguns casos, o autor e apresentador podem ser a mesma pessoa.

- *Moderador*: Responsável por reconduzir a reunião, caso a mesma desvie do seu foco principal. Ao término da sessão, deve questionar aos participantes suas opiniões sobre o resultado da reunião.
- *Registrador*: Responsável por registrar a hora de início e término da sessão, as questões apresentadas pelos revisores, os conflitos e ações propostas.
- *Revisores*: Principais participantes da reunião, normalmente são os clientes e usuários do produto final. Antes da reunião eles devem revisar, cuidadosamente, a documentação recebida e anotar os comentários a serem expostos na reunião.

Duran, em [Dur00], apresenta uma proposta metodológica para a validação de requisitos, que é realizada mediante a combinação das técnicas *walkthroughs* e *prototipagem de interface de usuário*. Esta técnica permite uma validação conjunta, de forma que os requisitos guiam a sessão de validação com a apresentação do protótipo referente a cada funcionalidade da aplicação. Esta proposta contempla a realização de três atividades:

1. ***Validar Requisitos Organizacionais e Funcionais***: através da combinação de *walkthrough* de casos de uso e protótipos de interface com usuários, ajuda aos clientes e usuários revisarem a documentação de requisitos.
2. ***Validar Requisitos Não-Funcionais***: os clientes e usuários, com a ajuda dos engenheiros de requisitos, revisam os requisitos não-funcionais para esclarecer as possíveis dúvidas.
3. ***Gerar Versão do Documento de Requisitos***: Após a solução dos conflitos, deve-se chegar a um acordo entre cliente e desenvolvedores para que possa ser gerada uma versão do documento de requisitos.

### **2.5.5. GERÊNCIA DE REQUISITOS**

Gerenciar requisitos significa poder escrever e acompanhar um requisito ao longo de todo o processo de desenvolvimento, garantindo que os documentos de requisitos são mais confiáveis e gerenciáveis. Esta atividade é uma das mais importantes do processo de engenharia de requisitos. Ela tem como objetivo principal o controle e gerenciamento de mudanças nos requisitos [Cas99].

Um dos principais problemas do desenvolvimento de software está na volatilidade dos requisitos, visto que eles estão sempre em mutação: ou por questões de melhoria das funções do

sistema, ou por solicitações do usuário/cliente, ou devido a correções de erros encontrados ou, ainda, por mudanças sociais e econômicas. Estas mudanças podem gerar efeitos colaterais e se propagar negativamente para outras partes do software. Por causa disto, é fundamental o controle dos requisitos descobertos e analisados no processo de engenharia de requisitos para que as mudanças sejam adequadamente tratadas e seus impactos corretamente avaliados. Isto implica na necessidade de rastrear os requisitos ao longo de todos os artefatos produzidos no processo de engenharia de software. Este rastreamento de requisitos deve ocorrer em ambos os sentidos, dos requisitos iniciais para os artefatos desenvolvidos e dos artefatos para as fontes que originaram os requisitos.

De forma resumida pode-se descrever os objetivos do gerenciamento de requisitos como:

- gerenciar mudanças para os requisitos acordados;
- gerenciar o relacionamento entre requisitos;
- gerenciar as dependências entre o documento de requisitos e os demais documentos produzidos no processo de engenharia de requisitos.

O gerenciamento de mudanças de requisitos estabelece que informações são coletadas e, para cada alteração proposta, são avaliados e julgados os custos e benefícios da alteração.

Os principais benefícios alcançados ao em realizar uma gerência de requisitos efetiva são:

- execução das alterações de acordo com suas prioridades;
- controle sobre as alterações realizadas;
- facilidade em descobrir quais os requisitos e outros documentos impactados com uma solicitação de mudança;
- informações estatísticas como: requisitos mais alterados, os mais dependentes e fáceis de serem alterados e histórico de alterações;
- informações necessárias são coletadas para alteração de um requisito;
- estimativas sobre custo e tempo para realização de uma alteração são realizadas de forma mais confiável.

No processo de gerenciamento de requisitos, os requisitos devem ser unicamente identificados. Esta identificação é o meio adotado para poder rastrear e avaliar os impactos advindos de mudanças. Para grandes sistemas, nos quais o número de requisitos a serem gerenciados é muito grande, é necessário que os mesmos sejam armazenados em uma base de dados e sejam registradas as ligações entre os requisitos relacionados. Verifica-se que, na grande

maioria dos sistemas, faz-se necessário o uso de ferramentas que apóiem o processo de gerenciamento de requisitos.

Toranzo [Tor02-B] afirma que o rastreamento de requisitos pode estabelecer relacionamentos entre artefatos de diferentes ou da mesma fase de desenvolvimento. Isso introduz as noções de Rastreamento vertical e Rastreamento horizontal. O rastreamento vertical é um relacionamento entre artefatos de diferentes fases do processo de desenvolvimento. Por exemplo, um relacionamento entre um requisito e um programa. O rastreamento horizontal é um relacionamento entre artefatos de uma mesma fase do processo de desenvolvimento. Por exemplo, um relacionamento entre dois requisitos.

A Figura 2.10 apresenta um exemplo de relacionamento entre os modelos gerados no desenvolvimento de um projeto. Segundo Spence, em [Spe03], estes relacionamentos entre os modelos provêm um nível de rastreabilidade que é implícita ao processo. Se quisermos adotar princípios de rastreabilidade de requisitos, então é necessário integrar os requisitos, itens de modelos e outros itens de rastreabilidade em uma hierarquia de rastreabilidade, tornando o rastreamento “visível e explícito”.

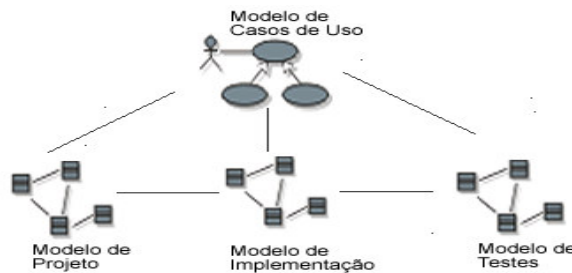


Figura 2.10- Exemplo de relacionamento entre artefatos

Mudanças nos requisitos são inevitáveis no processo de desenvolvimento. A experiência nas três últimas décadas indica que fazer mudanças sem visibilidade sobre os efeitos colaterais pode contribuir para uma pobre estimativa do esforço, degradação do projeto de software e uma desativação prematura do sistema [Tor02-B]. Para evitar esses problemas é recomendável analisar o impacto das mudanças, baseando-se nos relacionamentos existentes entre os artefatos do software e identificando os que são afetados pela introdução da proposta de uma mudança.

Diversas pesquisas têm sido realizadas na área de rastreamento de requisitos. Sendo assim, durante o planejamento e desenvolvimento de um projeto, deve-se decidir que técnica de rastreamento é mais adequada para o sistema. A seguir apresentaremos a técnica de Toranzo [Tor02-B], que foi utilizada para a definição da proposta de rastreamento dessa dissertação. O



diferencial na utilização desta técnica é a possibilidade de rastrear qualquer tipo de artefato gerado no desenvolvimento da aplicação. Além disto, ela permite rastrear os requisitos analisando seus relacionamentos com diversos tipos de informações e artefatos e fornece um processo simples para identificar as classes e relacionamentos que podem compor o modelo de rastreamento de um projeto.

### 2.5.5.1. Rastreamento: Técnica de Toranzo

A técnica de Toranzo, citada em [Tor02-A] e [Tor02-B], corresponde a uma proposta para melhorar o rastreamento de requisitos através de um conjunto de diretrizes a serem seguidas.

Inicialmente, as informações são classificadas em quatro níveis: Ambiental, Organizacional, Gerencial e Desenvolvimento, conforme apresentado na Figura 2.11. A partir daí, são identificadas as informações que podem afetar o sistema, os objetivos, as estratégias e as regras de negócios que devem ser rastreados. O passo seguinte é incluir as informações da gerência de projetos no modelo de rastreamento. As diretrizes seguintes orientam a identificar os subsistemas, identificar os requisitos, identificar os diagramas, identificar os programas e identificar os testes. Com isto, as informações são integradas e são estabelecidos os relacionamentos entre elas. A partir daí, é definida uma matriz para cada relacionamento do modelo, onde podem ser definidas diversas matrizes de rastreabilidade, tanto entre requisitos, como entre os requisitos e os diversos artefatos gerados.

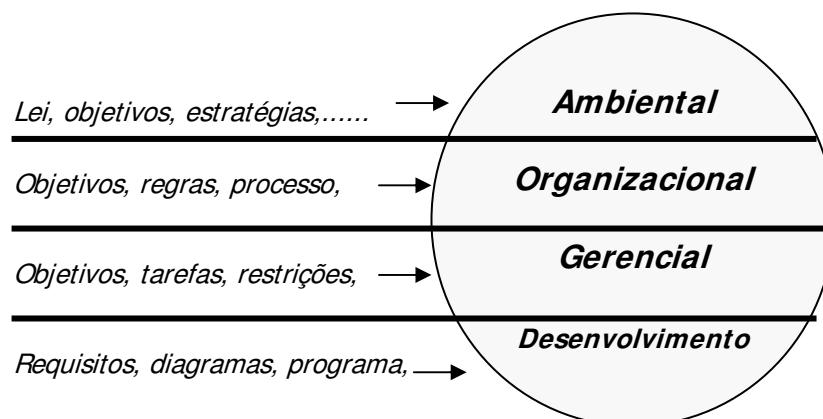


Figura 2.11 - Classificação das informações a serem rastreadas

As vantagens obtidas ao utilizar esta técnica no rastreamento de requisitos de software, são:

1. As informações a serem rastreadas são estruturadas em níveis de informações: ambiental, organizacional, gerencial e desenvolvimento.
2. Define um conjunto de tipos de relacionamentos para melhorar a semântica de um modelo de rastreamento. A proposta dos tipos de relacionamentos visa facilitar o entendimento do significado dos relacionamentos entre as classes do modelo de rastreamento. Em lugar, de delegar a responsabilidade ao usuário para definir o significado dos relacionamentos, através da nomeação dos mesmos, esta técnica define o significado e atribui uma notação para cada um dos tipos de relacionamentos do meta-modelo.
3. Desenvolve um meta-modelo que visa unificar os meta-modelos existentes. Este meta-modelo identifica e restringe os relacionamentos que podem ser expressos em um modelo de rastreamento.
4. Propõe um modelo intermediário de rastreamento de requisitos, o que evita que um usuário comece a construir seu modelo de rastreamento a partir do zero.
5. Propõe um processo para desenvolver um modelo de rastreamento. Este trabalho fornece um processo simples para identificar as classes e relacionamentos que podem compor o modelo de rastreamento de um projeto.
6. Permite a realização de rastreamento bi-direcional, horizontal e vertical, ou seja, todos os artefatos gerados no desenvolvimento do projeto podem ser rastreados.

Segundo Toranzo, o processo para construir um modelo de rastreamento deve contemplar as seguintes recomendações:

1. Informar as expressões e convenções usadas nas matrizes de rastreamento: identificar os símbolos usados.
2. Identificar classes no nível de informação ambiental: identificar as classes que podem afetar o sistema, no nível ambiental. Para satisfazer esta recomendação a seguinte diretriz deve ser seguida:
  - ✓ Diretriz 1: Identificar as informações que podem afetar o sistema.
3. Identificar as classes no nível de informação organizacional: Para satisfazer esta recomendação a seguinte diretriz deve ser seguida:
  - ✓ Diretriz 2: Identificar os objetivos, estratégias ou regras de negócios que serão rastreados.

4. Identificar classes no nível de informação gerencial: Para satisfazer esta recomendação a seguinte diretriz deve ser seguida:
  - ✓ Diretriz 3: Incluir classes de informação da gerência de projeto no modelo de rastreamento.
5. Identificar classes no nível de informação de desenvolvimento: Para satisfazer esta recomendação as seguintes diretrizes devem ser seguidas:
  - ✓ Diretriz 4: Identificar subsistemas.
  - ✓ Diretriz 5: Identificar requisitos.
  - ✓ Diretriz 6: Identificar diagramas.
  - ✓ Diretriz 7: Identificar programas.
  - ✓ Diretriz 8: Identificar testes.
6. Organizar e estruturar classes candidatas: Para satisfazer esta recomendação as seguintes diretrizes devem ser seguidas:
  - ✓ Diretriz 9: Remover as classes de informação irrelevantes.
  - ✓ Diretriz 10: Integrar as classes com o mesmo significado.
  - ✓ Diretriz 11: Integrar novas classes.
7. Estabelecer relacionamentos: Para satisfazer esta recomendação as seguintes diretrizes devem ser seguidas:
  - ✓ Diretriz 12: Organizar as classes.
  - ✓ Diretriz 13: Estabelecer relacionamentos.
8. Identificar as propriedades sobre as classes: Para satisfazer esta recomendação a seguinte diretriz deve ser seguida:
  - ✓ Diretriz 14: Recomendar atributos para as classes identificadas.
9. Definir e preencher as matrizes de rastreamento: Para satisfazer esta recomendação a seguinte diretriz deve ser seguida:
  - ✓ Diretriz 15: Definir uma matriz para cada relacionamento do modelo.
10. Validar as matrizes de rastreamento.

O termo *Modelo de Rastreamento* é um diagrama de classes conceitual, que representa coleções de informações rastreadas no processo de desenvolvimento. Por exemplo, a classe Requisito representa todos os requisitos do sistema. A classe Diagrama representa todos os tipos de diagramas que modelam os requisitos do sistema. Os relacionamentos entre as classes

conceituais representa as possíveis matrizes de rastreabilidade a serem produzidas (Por exemplo: o relacionamento “*Representado\_em*”, entre as classes Requisito e Programa, da Figura 2.12).

O termo *Modelo Intermediário* de rastreamento é um modelo de rastreamento que não é um modelo básico nem um modelo avançado. Ele possui todas as possíveis classes e relacionamentos existentes nos diferentes modelos de rastreamento particulares. Os principais objetivos do modelo intermediário de rastreamento são: servir como um ponto de partida para identificação, discussão e construção de um modelo de rastreamento; fornecer uma maior compreensão do meta-modelo de rastreamento de requisitos porque o próprio modelo intermediário é um exemplo de instanciação do meta-modelo; realçar a importância dos tipos de relacionamentos do meta-modelo para incrementar o significado e entendimento dos modelos intermediário e particulares de um projeto; e entender como problemas e soluções aconteceram no processo de software.

A Figura 2.12 apresenta um modelo de rastreamento, seguindo a técnica de Toranzo, para um sistema de biblioteca. Neste modelo, as classes candidatas a serem rastreadas são relacionadas entre si. A partir do modelo gerado e das informações que podem ser rastreadas, chega-se às matrizes de rastreabilidade.

Um exemplo prático de matriz de rastreabilidade é a tabela gerada através do relacionamento *Representado\_em*, entre as classes Requisitos e Programa. Inicialmente são identificados os requisitos, como mostra a Tabela 2.10. Depois são listados os programas, conforme apresenta na Tabela 2.11. Através do relacionamento, chega-se à matriz de rastreabilidade identificada na Tabela 2.12. Note que nesta última, foram identificados os programas e procedimentos principais que implementam cada requisito.

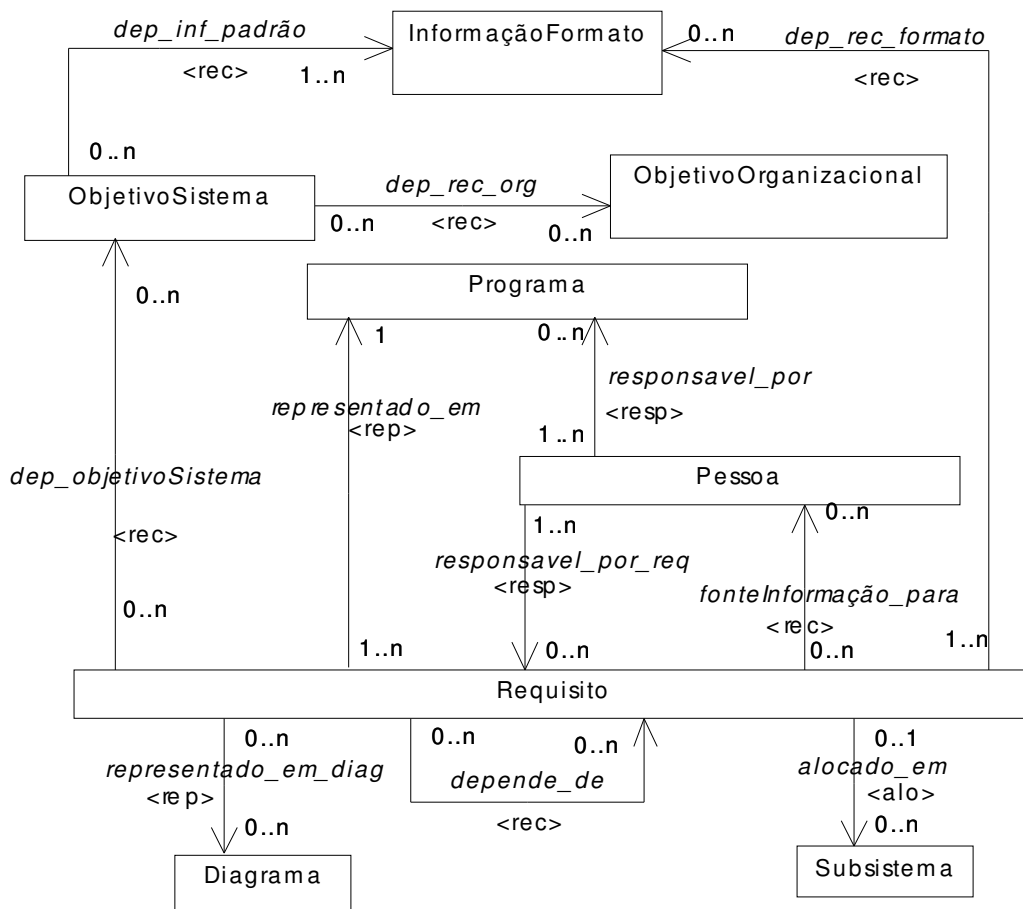


Figura 2.12 - Modelo de Rastreamento para um Sistema, segundo Toranzo [Tor02-B]

Tabela 2.10 - Requisitos de um Sistema de Biblioteca

[REQ-1]	O sistema deverá permitir a inclusão de usuários com os seguintes campos de informação: Código de Pessoa, Nome Completo, Endereço (Código do Endereço, Endereço, Tipo de endereço, Bairro, Cidade, UF, País, CEP, Caixa postal, Telefone, Fax), sobrenome, sexo, data de nascimento, nacionalidade (Tabela do Sistema), RG, Órgão Expedidor, Tipo de documento (carteira ou cédula), Data de expedição do documento, número da carteira, E-mail, biblioteca de origem.
[REQ-2]	O sistema deverá permitir a inclusão de dados referentes ao usuário caso este seja um operador do sistema de bibliotecas com o seguinte campo de formação: Nível de acesso.
[REQ-3]	O sistema deve permitir a consulta dos dados do usuário pelos campos: Nome, Sobrenome ou N° de identificação. O retorno da consulta mostra todos os dados do usuário.
[REQ-4]	O sistema deve permitir alterar os dados do usuário. O único campo que não pode ser alterado é o Código da Pessoa Física.

Tabela 2.11- Relação de Programas de um sistema de Biblioteca

Nomes de unidades que implementam os requisitos do Sistema de Biblioteca	
[PRG-1]	FcadUsuario
[PRG-3]	FconsUsuario
[PRG-4]	FcadUsuario
[PRG-5]	FcadOperador
[PRG-7]	FcadObra
[PRG-8]	FcadObra
[PRG-12]	FcadEmprestimo
[PRG-14]	FcadReserva
[PRG-15]	FrelatorioObra
[PRG-19]	FrelatorioFicha
[PRG-21]	FimprimeLombadaObra
[PRG-22]	FcadClassificacao
[PRG-26]	FcadAutor
[PRG-27]	FconsAutor

Tabela 2.12- Matriz de Rastreabilidade "Representado\_em" entre Requisitos e Programas

Representado _em: <rep> →	Programas
[REQ-1]	<imp,U: fCadUsuario; P: TF_CadUsuario.Frame_Botoes_Cadastro_CadUsuarioBtNovoClick( Sender: TObject)>
[REQ-2]	<imp,U: fCadOperador; P: TF_CadOperador.Frame_Botoes_Cadastro_CadUsuarioBtEditarClick( Sender: TObject)>
[REQ-3]	<imp,U: fConsUsuario; P: TF_ConsUsuario.EditPesquisaChange(Sender: TObject)>
[REQ-4]	<imp,U: fCadUsuario; P: TF_CadUsuario.Frame_Botoes_Cadastro_CadUsuarioBtEditarClick( Sender: TObject)>

## 2.5.6. QUALIDADE NO PROCESSO DE ENGENHARIA DE REQUISITOS

A melhoria do processo está relacionada com a modificação do processo de forma a alcançar melhoria de qualidade, redução de prazo e redução de recursos [Som98]. Existem quatro questões importantes que devem ser analisadas quando se planeja realizar melhoria nos processos existentes:

- quais são os problemas com os processos atuais;
- quais são os objetivos a serem alcançados com a melhoria do processo;

- como introduzir melhorias no processo para alcançar estes objetivos;
- como a melhoria do processo poderá ser controlada e gerenciada.

Os processos de Engenharia de Requisitos podem ter diversos tipos de problemas que resultam em problemas nos requisitos do sistema e, conseqüentemente, no próprio software. Sommerville, em [Som98], identifica os problemas com os processos de Engenharia de Requisitos:

- o processo não identifica, nem leva em conta, as reais necessidades dos diferentes tipos de *stakeholders* envolvidos;
- as necessidades do negócio não são consideradas;
- a falta de gerenciamento dos requisitos;
- a falta de definição de responsabilidades;
- os problemas de comunicação dos stakeholders;
- o planejamento longo demais e baixa qualidade dos documentos de requisitos.

Um fator importante que afeta a melhoria dos processos de software é o nível de maturidade do processo em uma organização. A maturidade do processo de uma organização corresponde ao grau de definição dos seus processos, como eles são controlados e a existência de suporte sistemático tanto humano como baseado em computador.

O modelo de maturidade CMM (*Capability Maturity Model*) do SEI (*Software Engineering Institute*) [Cmm03] é uma proposta para avaliação da maturidade do processo de software de empresas de desenvolvimento. Este modelo avalia as organizações na escala de 1 a 5, onde quanto maior a escala, mais maduro é o processo. A idéia básica do CMM é que as organizações devem avaliar seus processos e promoverem mudanças nos mesmos para atingirem níveis mais altos de maturidade. Os cinco níveis de maturidade do CMM são:

1. **Nível inicial:** As empresas têm um processo não disciplinado e fica a cargo dos indivíduos tanto a escolha das técnicas de desenvolvimento a serem usadas como o gerenciamento do processo.
2. **Nível repetível:** As empresas possuem os procedimentos básicos de gerenciamento de custo e prazo. Provavelmente serão capazes de fazerem previsões consistentes de custo e escalonamento para projetos na mesma área de aplicação.
3. **Nível definido:** O processo de software, tanto das atividades de gerenciamento como de engenharia, está documentado, padronizado, e integrado aos padrões de processo de software para toda a organização.

4. **Nível gerenciado:** Medições detalhadas tanto do processo como da qualidade do produto são coletadas e usadas para controlar o processo.
5. **Nível otimizado:** As empresas possuem uma estratégia de melhoria contínua do processo, baseada nos objetivos adotados para medição.

Em cada nível, um conjunto de práticas para Áreas Chave do Processo (ACP) deve ser definido e, uma vez tenha sido introduzido na organização, diz-se que ela atingiu o nível de maturidade e pode passar a se preocupar em atingir o próximo nível. Para que o nível 2 seja atingido, um conjunto de práticas chaves da gerência de requisitos deve ser inserido no processo, conforme será apresentado na subseção a seguir.

O propósito da ACP *Gerência de Requisitos* é entender as necessidades e expectativas do cliente, determinando exatamente o que deve ser contemplado no projeto do software e o que se espera receber como resultado, não se preocupando neste momento como o requisito será projetado e implementado. Em outras palavras, esta ACP objetiva manter um acordo entre quem solicita e quem desenvolve o software, estabelecendo um entendimento comum entre os participantes do projeto com relação aos requisitos do software contratado.

Apesar do CMM estabelecer a ACP *Gerência de Requisitos*, ele não descreve um processo de definição de requisitos. As organizações que desejarem realizar uma gerência efetiva de requisitos devem definir e documentar seu próprio processo de engenharia de requisitos para auxiliar na elicitação, documentação, validação e gerência dos mesmos.

## 2.6. Engenharia de Requisitos para Web

Segundo Holck [Hol01], ao desenvolver aplicações Web voltadas ao público geral, o grupo de usuários é bastante diferente daquele voltado aos sistemas internos das organizações, pois estes usuários nem sempre podem ser identificados através de um relacionamento com a organização e os usuários não são obrigados a usar a aplicação.

Os métodos tradicionais de desenvolvimento de sistemas concentram-se, inicialmente nos requisitos funcionais (o que o sistema deve fazer?), já nas aplicações web os requisitos não funcionais são analisados primeiro, pois se a aplicação não atender a alguns destes requisitos, é possível que os usuários nem tentem executar as funcionalidades às quais ela se propõe [Hol01]. Outra dificuldade encontrada é na definição dos requisitos funcionais, pois se os possíveis usuários de um web site não sabem o que ele pode oferecer, então não há como informar suas expectativas, diferentemente dos sistemas onde os usuários especificam e validam seus requisitos.



Holck ainda cita em [Hol01] a dificuldade para a realização dos testes da aplicação e a necessidade de novos padrões de cooperação entre os componentes técnicos usados no desenvolvimento (Por. Ex: Bancos de Dados, Servidores de Aplicação, Componentes de Segurança, etc.).

A maioria das metodologias de desenvolvimento de software assume, implicitamente, que existe uma “organização do usuário”. Assim, os usuários devem ser identificados e seus requisitos em relação ao sistema devem ser elicitados. Pode-se também observar os processos adotados pelos usuários antes da implantação do sistema e tentar reproduzir esses processos na aplicação. Outro ponto bastante enfatizado pelas metodologias de desenvolvimento é o impacto que a implantação do sistema poderá causar na organização. Essas considerações não se aplicam diretamente a sistemas web dirigidos ao público em geral, pois, na maioria das vezes, é praticamente impossível descobrir quem são os usuários do sistema. Não existe uma “organização do usuário”. Os usuários de um site aberto, por exemplo, encontram-se, normalmente, bastante dispersos, tornando difícil descobrir o que eles esperam do sistema e analisar o impacto que o sistema porventura possa causar em suas organizações.

Como a Engenharia de Requisitos para Web segue os princípios da Engenharia de Requisitos tradicional, surgem alguns problemas que devem ser tratados ao desenvolver aplicações web. Estes problemas serão discutidos na seção 2.6.1

Existem também dois fatores relevantes em relação aos requisitos das aplicações web: a grande importância dos Requisitos Não-Funcionais e o tratamento dos requisitos de interface. Estes fatores serão apresentados na seção 2.6.2.2.

## **2.6.1. PRINCIPAIS PROBLEMAS**

Nas subseções seguintes serão discutidos alguns problemas encontrados nas fases da Engenharia de Requisitos no desenvolvimento de aplicações web.

### **2.6.1.1. Elicitação de Requisitos**

No desenvolvimento de aplicações Web os principais problemas encontrados nesta fase são:

- o cliente, normalmente, tem em mente apenas as metas que ele deseja atingir, mas não as funcionalidades que a aplicação deve oferecer [Hol01]. Por esse motivo, algumas metodologias como, por exemplo, o Rational Unified Process [Rup02], sugerem que seja dada maior ênfase à modelagem do negócio como forma de descobrir os requisitos;

- existem requisitos não-funcionais que o cliente não conhece, tais como cores, largura de banda etc. por serem estritamente técnicos;
- os usuários não sabem o que desejam da aplicação por não saberem o que ela pode oferecer [Hol01];
- existe a necessidade de elicitare requisitos diferentes daqueles presentes em sistemas tradicionais. Exemplos desses novos requisitos são os requisitos de navegação/estrutura, apresentação (layout, design), conteúdo etc. [Jul01].

Uma dificuldade na captura de requisitos de sistemas web é a falta de *feedback* do usuário. No desenvolvimento tradicional de software, o novo sistema é normalmente imposto aos usuários, que se vêem obrigados a utilizá-lo. Isso cria, de certa maneira, um compromisso dos usuários com o sistema, estimulando-os a apresentar sugestões de como o sistema pode ser melhorado ou como eles gostariam que o sistema funcionasse. Sistemas web não podem, em sua maioria, ser impostos aos usuários. Geralmente, não existe nenhum compromisso dos usuários com a melhoria do sistema, ou seja, os usuários não estão engajados no desenvolvimento. Algumas técnicas têm sido utilizadas para tentar obter feedback do usuário em relação ao sistema como, por exemplo, disponibilizar um e-mail ou um formulário para sugestões e avaliação do sistema. Não há, porém, estudos que indiquem que essas técnicas são realmente efetivas.

### **2.6.1.2. Análise e Negociação de Requisitos**

No desenvolvimento de aplicações Web a diversidade de tipos de requisitos pode trazer dificuldades para a fase de análise e negociação. Para lidar com esse problema, Coda [Cod98] sugere a classificação dos requisitos nas seguintes categorias: conteúdo, estrutura, acesso e layout. Essa divisão propicia a modularização da análise entre os diferentes profissionais da equipe de desenvolvimento.

### **2.6.1.3. Documentação dos Requisitos**

A documentação de características próprias de aplicações web, como navegação, interatividade, personalização de informação e elementos multimídia, é dificultada pela ausência de uma notação padronizada. O documento de requisitos precisa estar muito bem estruturado e seguir um padrão pré-estabelecido, uma vez que será o principal elemento de comunicação de uma equipe cujos membros possuem as mais diversas formações.

#### **2.6.1.4. Verificação e Validação dos Requisitos**

Para aplicações web, dirigidas ao público geral, uma das principais dificuldades relacionadas à etapa de validação é devido a falta de conhecimento sobre os usuários, tornando difícil a obtenção do *feedback* sobre o sistema. Algumas soluções são sugeridas, como por exemplo:

- disponibilizar protótipos para avaliação do sistema e do comportamento do usuário ao usá-lo [Rup02];
- disponibilizar um e-mail para coletar possíveis sugestões dos usuários;
- escolher um grupo de usuários beta para avaliação da aplicação antes de torná-la disponível publicamente, a exemplo dos softwares de prateleira.

#### **2.6.2. FATORES IMPORTANTES**

Nesta seção abordaremos dois fatores relevantes no desenvolvimento de aplicações Web: os *Requisitos Não-Funcionais* e a *Interface com o usuário*.

##### **2.6.2.1. Requisitos Não-Funcionais**

No desenvolvimento de aplicações Web os RNFs desempenham um papel crítico. Problemas na elicitação deste tipo de requisito podem trazer conseqüências desastrosas, visto que estes requisitos estão entre os mais caros e difíceis de corrigir, uma vez que um sistema tenha sido implementado. Por este motivo, será dada forte ênfase aos Requisitos Não Funcionais neste trabalho.

Esses requisitos, na maior parte dos casos, também estão presentes nas aplicações tradicionais mas, nas aplicações web, eles adquirem maior relevância e visibilidade [Pin03]:

- *Segurança*: a segurança de um sistema web é de extrema importância, não apenas do ponto de vista técnico, mas do próprio negócio. Falhas de segurança podem abalar a reputação da empresa e trazer conseqüências desastrosas para o negócio. A segurança envolve fatores como política de privacidade e segurança nas transações. Quanto à política de privacidade, é necessário deixar claro que as informações fornecidas pelos usuários não serão utilizadas de forma indevida. A segurança das transações é uma questão técnica que tem sido resolvida através do uso de técnicas como criptografia e controle de acesso. O usuário tende a se sentir mais seguro efetuando transações em sites de empresas que ele já conhece. Uma forma de aumentar a confiança que o usuário tem

na aplicação é fornecer informações sobre a empresa e alguma garantia de indenização caso algo dê errado [Pin03];

- *Usabilidade*: em aplicações web, a usabilidade é um fator de extrema relevância. Uma complicação adicional é a impossibilidade de realizar um treinamento com todos usuários. Outra dificuldade é a grande diversidade do público alcançado pelas aplicações web. Sabe-se que usuários se comportam de modo semelhante quando interagindo com certas páginas web. A questão é como identificar esse comportamento e como identificar os vários grupos (perfis) de usuários e associar um comportamento padrão a cada um deles [Pin03];
- *Performance*: uma característica presente na maioria dos usuários é a impaciência em relação à velocidade com que a aplicação web é carregada. A questão da performance está ligada a fatores técnicos como tamanho dos arquivos, tipo de conexão, número de usuários, quantidade de informações manipuladas, etc. [Pin03];
- *Consistência*: uma preocupação em sites que contêm um grande número de informações é que não deve haver informações conflitantes [Pin03];
- *Objetividade*: a dificuldade em encontrar informações é um dos fatores que podem levar o usuário a desistir de utilizar a aplicação [Pin03];
- *Escalabilidade*: na história da Internet, não são raros os casos de empresas que atraíram um grande número de usuários para seus sites em um curto espaço de tempo. Como é muito difícil prever o número de usuários de uma aplicação web, o sistema deve ser o mais escalável possível e deve-se descobrir, através de simulações, como o sistema deve se comportar no caso de um grande aumento no número de usuários [Pin03];
- *Disponibilidade*: uma aplicação web deve funcionar vinte e quatro horas por dia, sete dias por semana. Para alcançar essa meta, faz-se necessário o uso de hardware e software de qualidade; [Hol01]
- *Manutenibilidade*: a questão da manutenibilidade de sistemas web está diretamente ligada à disponibilidade, já que a manutenção do sistema deve ser feita sem que o serviço fique indisponível [Hol01];
- *Atualidade*: as informações presentes em um site devem estar sempre atualizadas, sob pena de causar má impressão aos usuários. Deve-se disponibilizar informações sobre quando a página foi escrita, colocada na web e atualizada. Deve-se deixar claro quando os dados (gráficos, tabelas, pesquisas) foram obtidos [Pin03];

- *Utilidade*: uma aplicação web deve agregar o maior número possível de facilidades para seus usuários. Essas facilidades podem ser desde serviços prestados pela própria aplicação até links para outros sites que forneçam o serviço ou informação. Muitas vezes, entretanto, existe um conflito entre o que o usuário acha que é útil e o que os stakeholders do site desejam disponibilizar. Um link para o site de concorrentes pode ser considerado de grande utilidade pelo usuário mas não é interessante em termos do negócio [Pin03];
- *Personalização*: a enorme diversidade de usuários de uma aplicação web faz com que nenhuma solução adotada agrade a todos. Uma solução para resolver esse problema é a incorporação de um sistema de personalização da aplicação. Deve-se identificar as características desejáveis e inaceitáveis para cada tipo de usuário, como ele se comporta quando interagindo com o site, que informações ele gostaria de receber, que produtos ele costuma comprar etc. A partir dessas informações, a aplicação deve ser configurada de acordo com o usuário que a estiver utilizando [Pin03];

### 2.6.2.2. Interface com o Usuário

Segundo Landauer [Lan95], 80% do custo total de desenvolvimento são gastos com a manutenção. E 80% dos custos de manutenção de software ocorrem devido a problemas na interação entre os usuários e o sistema, e não a problemas técnicos. Ele também afirma que 64% destes custos estão relacionados com problemas de usabilidade. Esta situação reforça a importância dos requisitos de interface no desenvolvimento de software. Isto significa que o desenvolvimento de software é uma atividade multidisciplinar, implicando na participação de profissionais de diferentes áreas, não apenas engenheiros de computação, mas também das áreas de design, marketing e psicologia, por exemplo.

Em [Med00], Medeiros, apresenta um modelo conceitual para o projeto de interface homem-máquina, baseado na análise das tarefas dos usuários (“*Task Analysis*”<sup>3</sup>). Do ponto de vista de projeto de interface, as tarefas correspondem ao que os usuários devem executar ao usarem o sistema. Assim o objetivo desta análise é identificar os modelos mentais e descrever suas representações de forma a revelar conceitos e características do processo de execução da tarefa no sistema. Em geral, o processo de análise de tarefas consiste, inicialmente, em obter as

---

<sup>3</sup> Análise de Tarefas: uma forma de análise do trabalho humano, que especifica os objetivos a serem alcançados, ou como eles devem ser atingidos.

descrições das atividades dos usuários através da utilização de técnicas de elicitação (entrevistas, questionários, etc.), seguida pela análise dos requisitos elicitados, objetivando compreender:

- os objetivos dos usuários;
- a decomposição hierárquica das tarefas em sub-tarefas;
- os objetos e conceitos bem conhecidos dos usuários e suas aplicações durante a execução das tarefas;
- os procedimentos adotados pelos usuários para atingirem seus objetivos;
- as pré-condições para a aplicação destes procedimentos;
- as possíveis incoerências e incompletudes;
- fatores críticos, erros e problemas possíveis.

Com a obtenção destas informações, Medeiros sugere que a análise de tarefas pode ser usada para suportar o projeto em três momentos: especificação do sistema, especificação de interface e elaboração do manual do usuário.

Segundo, John Shiple, em [Shi03], antes de construirmos a interface de uma aplicação web, ou website, deve-se:

- identificar qual o objetivo da aplicação, depois qual a audiência da mesma, ou seja, avaliar a experiência dos usuários e definir seus papéis;
- definir os cenários da interface, ou seja, com base nos papéis encontrados ao avaliar a audiência, devem ser identificados os cenários de uso da interface por cada tipo de usuário (papéis) e então, escritas as histórias de como será a interação entre o usuário e a aplicação;
- analisar interfaces, ou sites, similares e/ou concorrentes;
- identificar o conteúdo visual através da análise dos requisitos funcionais;
- organizar o conteúdo e definir a estrutura do site;
- definir a navegabilidade do site;
- definir os templates que descrevem as páginas html;
- definir o protótipo da aplicação.

Baresi, Garzotto e Paolini, em [Bar01], propõem o Framework W2000 para projeto de aplicações Web. Este framework é uma integração entre UML e HDM – Hypermedia Design Model e consiste em: definir estereótipos e customização de diagramas para caracterizar HDM com UML; criar diretrizes para usar UML com o objetivo de especificar alguns dos aspectos dinâmicos e operacionais de aplicações Web; e refinar diagramas de Casos de Uso para descrever

requisitos de alto nível, relacionados a aspectos funcionais e navegacionais. Neste framework, cada atividade produz um modelo (representado por diagramas de UML), que descreve alguns aspectos da aplicação.

A análise de requisitos inicia com a identificação das diversas categorias (papéis) de usuários que devem interagir com a aplicação. Esta identificação dos papéis ajuda a organizar os caminhos navegacionais e as operações a serem realizadas pelos usuários. Por exemplo, para um site de uma operadora de cartão de crédito: *Usuários Genéricos* navegam sobre as informações públicas; *Clientes* realizam consultas em seus cartões (saldo, extrato, etc), alteram senha e solicitam alterações cadastrais, segunda via de cartão e inclusão de dependentes; e *Futuros Clientes* submetem propostas e consultam suas propostas.

No Framework W2000 também são produzidos o Modelo de Casos de Uso Funcionais, que identifica as principais funções e as associa com os papéis de usuários, e o Modelo de Casos de Uso Navegacional, onde as capacidades de navegação são associadas a cada papel. Os usuários genéricos que navegam pela aplicação web, mas que não são representados por atores do sistema por não executarem nenhuma funcionalidade da aplicação, aparecem apenas no Modelo de Casos de Uso Navegacional porque eles apenas navegam através das páginas públicas, não executando nenhuma ação em particular.

Juan Sánchez, propõe em [San02] um processo automático de geração dos componentes da interface de uma aplicação baseado em XML (*Extensible Markup Language*) que possibilita gerar as interfaces em diversas ferramentas de desenvolvimento, como Java, VisualBasic, Delphi, etc.

O método proposto por Sanchez chama-se VRU (*Validación de Requisitos de Usuário*) e pode ser integrado aos métodos de desenvolvimento de software que utilizem casos de uso, para identificar requisitos funcionais, e alguma variante dos diagramas de seqüência, para descobrir as interações internas do sistema.

Sanchez, em [San02], utiliza *Message Sequence Chart (MSC)* para descrever e especificar, de um modo formal, os casos de uso da aplicação, do ponto de vista das interações internas, que devem incluir informações necessárias para gerar os formulários de interface com o usuário, associados a cada caso de uso.

A Tabela 2.13 apresenta os principais componentes do processo VRU. Durante o ciclo de vida de desenvolvimento são encontradas as seguintes fases:

- *Análise Preliminar*: Esta fase é composta pelo fluxo de trabalho Análise Externa e corresponde ao início do projeto, onde são executadas três atividades: (i) Interiorizar o Projeto, onde um especialista do domínio da aplicação elabora um dicionário com a terminologia utilizada e um documento inicial de requisitos. (ii) Criar Perfil de Usuário, onde são descritos os possíveis papéis de usuários que deverão interagir com a aplicação. (iii) Elaborar Escopo do Sistema, que contextualiza a aplicação na organização.
- *Análise detalhada*: O objetivo desta fase é reescrever o documento de requisitos inicial acrescentando modelos gráficos e textuais consistentes com UML. Os fluxos de trabalho desta fase são: (i) a Análise Funcional, onde são criados os modelos funcionais da aplicação; (ii) a Análise Interna, onde são refinados e estruturados os requisitos através de Diagramas de Classes, Diagramas de *MSC* (*Message Sequence Chart* - sintetizam os casos de uso) e Diagramas de *MSC* Etiquetados (com informação da interface do usuário); (iii) a Geração onde são realizadas atividades que permitem a geração da especificação do comportamento do sistema e o modelo de apresentação da interface com usuários.
- *Validação*: O objetivo desta fase é validar os requisitos com os usuários através da interface gerada na fase anterior.

Tabela 2.13 - Processo VRU [San02]

Etapa do Ciclo de Vida	Fluxo de Trabalho	Atividades	Artefatos
<b>Análise Preliminar</b>	Análise Externa	Interiorizar Projeto	Dicionário
			Requisitos Informais
		Criar Perfil de Usuário	Mapa de papéis de usuários
		Elaborar Escopo do Sistema	Modelo de domínio
			Diagrama de Contexto
<b>Análise Detalhada</b>	Análise Funcional	Criar Modelo Funcional	Atores
			Casos de Uso
			Serviços Funcionais
	Análise Interna	Modelar Objetos	Diagramas de Classes
		Sintetizar Casos de Uso	Diagramas MSC
		Etiquetar MSC's	Diagramas MSC Etiquetados
	Geração	Gerar Interface	Modelo de Apresentação
		Gerar Dinâmica	Diagramas de Transição
<b>Validação</b>	Animação	Executar a Interface	

Analisando o exposto acima, concluímos que as aplicações de software requerem interfaces com usuários aptas a suportar uma grande variedade de tarefas, em diversos contextos. Os componentes de interface para aplicações interativas estão adquirindo, a cada dia, mais



importância nos meios acadêmico e industrial. Isto ocorre porque existem diversas aplicações que executam tarefas similares, mas os usuários tendem a escolher aquelas que são mais fáceis de usar e, conseqüentemente, mais eficientes e produtivas, além de reduzir as necessidades de treinamento.

## **2.7. Considerações Finais**

A Engenharia de Requisitos deve ser realizada em toda as fases do ciclo de desenvolvimento do software, pois a “voz do cliente” deve ser ouvida desde a concepção inicial do sistema até a sua entrega. Qualquer metodologia deve tratar este fato e prever a possibilidade de requisitos serem modificados, inseridos ou excluídos.

A volatilidade do contexto dos requisitos também é um fator a ser considerado, pois ninguém pode, durante a análise dos requisitos, assumir verdades absolutas. Por este motivo, a mudança de requisitos deve ser vista como um fato natural no desenvolvimento. Axel van Lamsweerde, em [Lam93], faz uma distinção entre metas e requisitos. Metas são os objetivos que o sistema deve atingir, ou seja, elas são imutáveis pois são os objetivos do serviço propriamente dito. Já os requisitos representam uma das possíveis maneiras de alcançar os objetivos, logo eles podem ser modificados durante o desenvolvimento da aplicação.

Com base nas considerações a cima e nos conceitos e técnicas descritos neste capítulo, concentraremos nossos esforços no desenvolvimento de um processo genérico para a engenharia de requisitos, que além de refletir os resultados mais recentes de pesquisas na área de requisitos, também atenda as características das aplicações web.

No próximo capítulo será abordado o Rational Unified Process (RUP), que é um processo já validado nas comunidades acadêmica e industrial, para então propormos um modelo para engenharia de requisitos, com base neste processo.

## Capítulo 3 O RUP: RATIONAL UNIFIED PROCESS

---

### 3.1. Visão Geral do RUP

O Rational Unified Process (RUP) é um processo genérico de engenharia de software desenvolvido e mantido pela IBM Rational Software Corporation [Rat02] que cobre todo o ciclo de desenvolvimento de software e provê uma abordagem disciplinada para atribuir tarefas e responsabilidades durante o desenvolvimento de software dentro de uma organização. Seu principal objetivo é assegurar a produção de software de alta qualidade, que atenda às expectativas do seu usuário final dentro do cronograma e do orçamento previstos.

Este processo vem sendo muito bem aceito nas organizações de desenvolvimento de software, por possuir características que o tornam um processo prático, pois ele é um processo customizável e pode ser adaptado à forma de trabalho das organizações, é disponibilizado em páginas da Web permitindo fácil acesso aos membros da organização, além de prover *guidelines*, modelos e exemplos que facilitam sua utilização.

Este capítulo apresenta as principais características deste processo e descreve, de forma detalhada, o Fluxo de Requisitos, apresentando as atividades realizadas e os artefatos produzidos em cada subfluxo; haja vista o foco deste trabalho ser a produção de um processo de Engenharia de Requisitos, que tanto contemple as características das aplicações Web, como boas práticas que têm sido pesquisadas na área de Engenharia de Requisitos.

#### 3.1.1. FASES DO RUP

As atividades do RUP contemplam desde o planejamento do projeto, até os processos de teste e gerência da configuração do software. O processo é composto por quatro fases (concepção, elaboração, construção e transição), onde cada uma possui objetivos específicos [Rup02].

Na fase de concepção, deve-se estabelecer o escopo e a viabilidade econômica do projeto, incluindo uma visão operacional, o critério de aceitação e o que deve ser esperado do produto; a discriminação dos Casos de Uso críticos do sistema e restrições; a exibição e/ou demonstração de pelo menos uma arquitetura candidata; a estimativa do custo total e do prazo para o projeto inteiro; a identificação dos potenciais riscos e a preparação do ambiente de suporte para o projeto.

Na fase de elaboração, o objetivo é eliminar os principais riscos e estabelecer uma arquitetura estável para determinar custo e prazo para o restante do processo de desenvolvimento. As atividades principais desta fase são as seguintes: definir e validar a arquitetura, estabelecer um entendimento sólido dos casos de uso mais críticos que dirigem as decisões de planejamento e de arquitetura, criar planos de iteração detalhados para a fase de construção, refinar a arquitetura e selecionar componentes a serem utilizados no projeto.

Na fase de construção, um produto completo é desenvolvido até que esteja pronto para ser entregue ao cliente. Os objetivos desta fase são: minimizar custos de desenvolvimento, otimizando recursos e evitando re-trabalho; criação de versões usáveis do sistema (alfa, beta, e outras versões de teste); conclusão da análise, projeto, desenvolvimento e teste de todas as funcionalidades requeridas; desenvolvimento iterativo e incremental de um produto que esteja pronto para a transição para sua comunidade de usuários; decidir se o software e os usuários estão prontos para a aplicação ser implantada; obtenção de um certo grau de paralelismo no trabalho das equipes de desenvolvimento.

Na fase de transição uma versão beta do sistema é disponibilizada, bem como documentações e treinamento. Os objetivos desta fase são: executar beta testes para validar o sistema com relação às expectativas dos usuários; executar beta testes relativos à integração com sistemas legados; realizar treinamento de usuários; obter o aceite dos *stakeholders* de que o produto está consistente com o critério de aceitação. As atividades realizadas nesta fase são: execução dos planos de implantação, finalização do material de suporte ao usuário final, criação de versões do produto, obtenção de feedback do usuário, melhoramento do produto baseado no feedback, disponibilização dos produtos para os usuários finais.

### **3.1.2. CONCEITOS BÁSICOS**

Nesta seção apresentamos alguns elementos que são utilizados na descrição do RUP [Rup02]:

- *Responsáveis (workers)*: um responsável define o comportamento e as responsabilidades de um indivíduo (ou conjunto de indivíduos). Eles representam um papel realizado por indivíduos dentro do projeto e definem como eles devem conduzir seus trabalhos. É importante saber que uma pessoa específica pode desempenhar diferentes papéis ao longo do processo.

- *Atividades*: uma atividade é algo que um responsável realiza e que provê um resultado significativo para o contexto do projeto. Possui um propósito bem-definido e geralmente é expressa em termos da criação ou atualização de artefatos. Toda atividade é atribuída a um responsável específico.
- *Artefatos*: um artefato é um item produzido ao longo do processo e serve de entrada e saída para as atividades: os responsáveis usam artefatos para realizarem as atividades e produzem artefatos ao longo da realização dessas atividades. Um artefato pode ter várias formas: pode ser um modelo que contém outros artefatos, pode ser um elemento de um modelo (como uma classe, em um Diagrama de Classes), um documento, um código-fonte ou um executável.
- *Fluxos de atividades (workflows)*: um fluxo de atividades descreve uma seqüência de atividades que produz algum resultado de valor e mostra as interações entre os responsáveis, representado no RUP através de um diagrama de atividade de UML, sendo que cada atividade deste diagrama corresponde a um subfluxo.
- *Subfluxo*: são usados para estruturar os fluxos de atividades e representam um agrupamento de atividades, responsáveis envolvidos, artefatos de entrada e artefatos produzidos. Fornecem um alto nível de abstração na apresentação de um fluxo.
- *Guias (guidelines)*: são regras, recomendações ou heurísticas que dão suporte às atividades e aos artefatos. Podem estar associados a uma atividade ou a um grupo de atividades, dando dicas práticas de como conduzir essas atividades. Ou podem estar associados a um artefato, descrevendo como criar, avaliar e usar o artefato. Os guias também servem para garantir a qualidade dos artefatos (em forma de checklists) ou para realizar revisão de atividades.
- *Templates*: são modelos ou protótipos de artefatos, que são usados para construir os artefatos. Estão sempre associados a uma ferramenta (templates do Word, do Project, etc.).

### **3.1.3. PRINCIPAIS CARACTERÍSTICAS**

O paradigma do RUP defende forte comunicação entre a equipe, a antecipação da solução de problemas críticos que possam vir a surgir, e o direcionamento do projeto orientado às documentações (artefatos) geradas. O RUP é um processo iterativo e incremental, orientado a casos de uso e centrado na arquitetura. Estes três aspectos são tratados nas próximas subseções.

Outras características que podem ser citadas são: orientado a objetos, ênfase na análise de riscos e o uso de UML para modelagem visual.

### **3.1.3.1. Iterativo e Incremental**

A abordagem iterativa propicia um entendimento gradativo do problema, através de sucessivos refinamentos e incremento da solução efetiva durante os múltiplos ciclos. Além disso, permite uma maior flexibilidade para acomodar novos requisitos ou solicitações de mudanças dos objetivos, bem como identificar e resolver os riscos mais cedo, contribuindo para o gerenciamento mais efetivo do projeto [Cam02].

Segundo Souza, em [Sou02], como o desenvolvimento de um produto de software comercial pode durar meses, é prático que o trabalho seja dividido em iterações. Cada iteração refere-se a passos no fluxo de desenvolvimento, e incrementos na evoluções do produto. Sucessivas iterações constroem os artefatos de desenvolvimento a partir do estado que eles foram deixados ao final da iteração anterior. Em toda iteração, o projetista identifica e especifica os Casos de Uso relevantes, cria um projeto usando a arquitetura selecionada com guia, implementa o projeto em componentes e verifica se os componentes satisfazem estes Casos de Uso. Se a iteração alcança suas metas, o desenvolvimento então, procede para a próxima iteração.

Os benefícios de um processo iterativo controlado são os seguintes: redução do risco de custos com despesas em um único incremento; redução do risco de atrasos no planejamento; aceleração do ritmo de todos os esforços de desenvolvimento porque os desenvolvedores trabalham mais efetivamente em direção a um resultado claro; reconhecimento de uma realidade freqüentemente ignorada – as necessidades do usuário e os requisitos correspondentes não podem ser completamente definidos de uma vez.

### **3.1.3.2. Orientado a Casos de Uso**

Orientado a Casos de Uso significa que os casos de uso direcionam todo o processo de desenvolvimento: os requisitos são especificados através deles; durante a análise, projeto e implementação os casos de uso são realizados, isto é, é definido como eles irão acontecer; é a partir do modelo de casos de uso do sistema que os testes são planejados e realizados; por fim, os casos de uso são usados pelo gerente do projeto no planejamento e acompanhamento das iterações. Além disto, os casos de uso apresentam meios de expressar os problemas de forma que

os mesmos sejam compreendidos por diversos tipos de *stakeholders*: clientes, usuários, desenvolvedores e arquitetos [Sou02].

Uma possível interação entre um usuário e o sistema pode ser representada por um Caso de Uso. Um Caso de Uso é a descrição de uma parte de funcional do sistema que retorna ao usuário um resultado de valor observável, como foi visto na seção 2.4.1.3. Os Casos de Uso representam os requisitos funcionais e juntos formam o Modelo de Casos de Uso que descreve todas as funcionalidades do sistema.

### 3.1.3.3. Centrado na Arquitetura

O RUP defende que a arquitetura do sistema seja definida o mais cedo possível, pois ter uma arquitetura robusta definida já nas fases iniciais do projeto, facilita o desenvolvimento paralelo, minimiza retrabalho, e aumenta a probabilidade do reuso de componentes [Cam02]. E essa arquitetura serve de guia para todo o projeto e implementação.

O conceito de arquitetura de software engloba os aspectos estáticos e dinâmicos mais significativos do sistema. A arquitetura é influenciada por diversos fatores, tais como a plataforma de software, blocos de construção disponíveis para reuso, considerações de implantação, sistemas legados e requisitos não-funcionais. A arquitetura é uma visão do projeto como um todo, que torna visível as características mais importantes do mesmo.

## 3.2. Fluxos do RUP

O RUP possui nove fluxos, executados ao longo das quatro fases, sendo seis de processo e três de suporte, conforme apresentado na Figura 3.1 [Rup02]. Os fluxos de processo são os seguintes:

- *Modelagem do Negócio (Business Modeling)*: envolve o entendimento da estrutura e dinâmica da organização cliente, garantindo que clientes, usuários e desenvolvedores tenham a mesma visão da organização para a qual será feito o desenvolvimento.
- *Requisitos (Requirements)*: envolve a definição dos requisitos do sistema de software e de como gerenciar escopo e mudanças de requisitos.
- *Análise e Projeto (Analysis and Design)*: envolve a tradução dos requisitos em uma especificação na linguagem de modelagem visual UML (Unified Modelling Language).
- *Implementação (Implementation)*: envolve o desenvolvimento de código: classes, objetos, etc., teste de unidades e integração de subsistemas.

- *Teste (Test)*: envolve a verificação do sistema como um todo, com testes de integração e conformidade com os requisitos especificados.
- *Distribuição (Deployment)*: envolve o empacotamento, distribuição, instalação e treinamento de usuários, assim como o planejamento e condução de testes beta.

Os fluxos de suporte compreendem atividades necessárias para apoiar a execução dos fluxos de processo. São eles:

- *Gerência de Projeto (Project Management)*: envolve o gerenciamento de riscos, planejamento e acompanhamento do projeto.
- *Gerência de Configuração e Mudanças (Configuration and Change Management)*: envolve o gerenciamento dos artefatos gerados durante o desenvolvimento.
- *Ambiente (Environment)*: envolve a organização do ambiente de trabalho para a equipe do projeto e a configuração do RUP para o projeto.

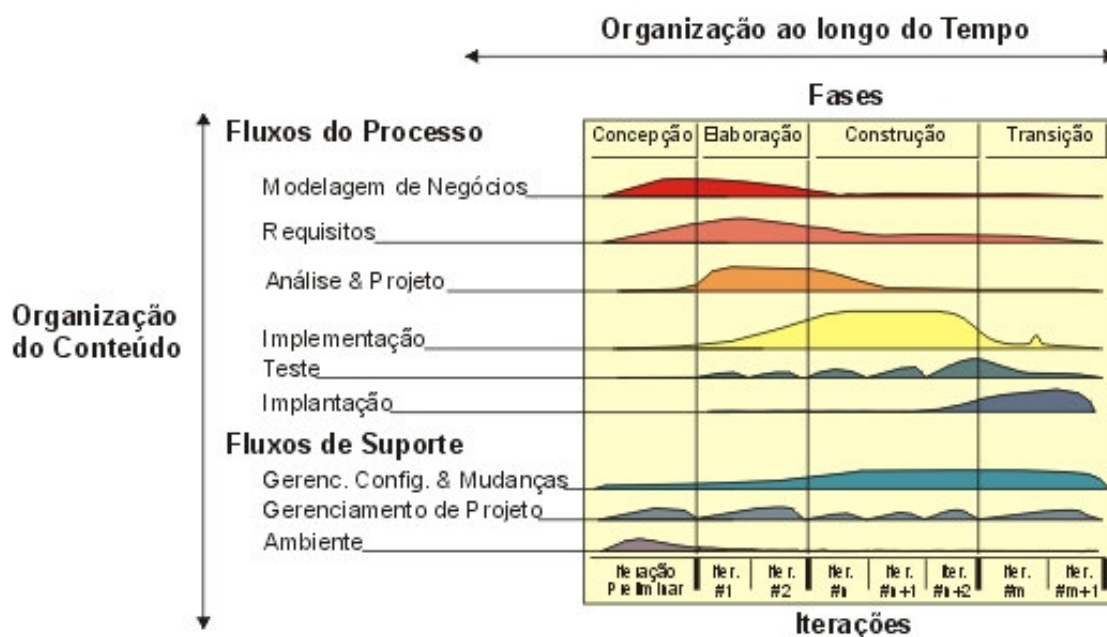


Figura 3.1- Fluxos do RUP [Rup02]

Cada fluxo do processo é definido em termos de atividades, responsáveis e artefatos. De modo que cada atividade possui um ou mais responsáveis e pode receber como entrada, ou gerar como saída, artefatos específicos.

Como o foco deste trabalho é no processo de **Engenharia de Requisitos**, a próxima seção detalha as características do *Fluxo de Requisitos* do RUP 2002 [Rup02]. Este fluxo relaciona-se com outros fluxos da seguinte forma: o fluxo de *Modelagem de Negócio* fornece um contexto organizacional para o sistema de software; o fluxo de *Análise & Projeto* tem como entrada

primária os requisitos; o fluxo de *Testes* baseia-se no Modelo de Casos de Uso e no documento de Especificação Complementar para o planejamento dos testes do sistema.

### 3.3. Fluxo de Requisitos

Os objetivos deste fluxo, apresentado na Figura 3.2, são os seguintes:

- estabelecer e manter um acordo com os clientes e outros *stakeholders* sobre o que o sistema deve fazer;
- proporcionar aos desenvolvedores do sistema um melhor entendimento dos requisitos do sistema;
- definir os limites do sistema;
- fornecer uma base para o planejamento de conteúdos técnicos das iterações;
- fornecer uma base para estimação de custos e tempo para o desenvolvimento do sistema;
- definir as interfaces gráficas do sistema, focando nas necessidades e objetivos dos usuários.

Para alcançar estas metas, é importante, em primeiro lugar, entender a definição e o escopo do problema que será resolvido com o sistema.

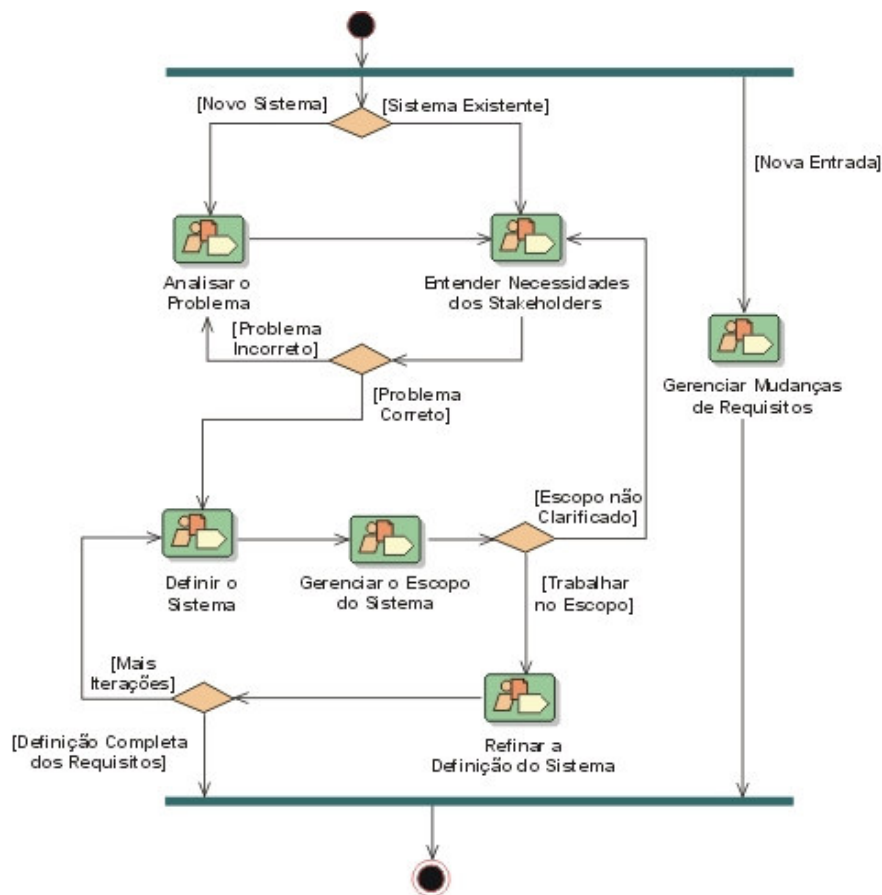


Figura 3.2- Fluxo de Requisitos do RUP [Rup02]



Analisando o fluxo apresentado na Figura 3.2, verifica-se que o primeiro passo a ser feito é *Analisar o Problema*, cujo objetivo é ter a certeza que todos os envolvidos entendem e concordam com o problema que está-se tentando resolver com o novo sistema. Além disto, devem ser identificados os stakeholders, definidos os limites do sistema e identificadas as restrições a ele impostas.

Após analisar o problema, é necessário elicitare compreender as informações dos stakeholders, através do subfluxo *Entender as Necessidades dos Stakeholders*.

Estando o problema entendido, parte-se então para o subfluxo *Definir o Sistema*, ou seja, alinhar o entendimento do problema com a equipe de projeto, fazer uma análise de alto nível dos requisitos coletados e produzir modelos e documentos formais dos requisitos elicitados.

No subfluxo *Gerenciar o Escopo do Sistema*, são determinados características e requisitos que devem ser tratados na iteração atual e definidos o conjunto de casos de uso que representam as funcionalidades centrais do sistema.

Parte-se então para o subfluxo *Refinar a Definição do Sistema* onde são detalhadas as especificações de requisitos do software e as especificações dos casos de uso.

O subfluxo *Gerenciar Mudanças de Requisitos* é responsável por avaliar e gerenciar as solicitações de mudanças nos requisitos.

Nas seções seguintes, são detalhados todos os subfluxos do Fluxo de Requisitos do RUP.

### **3.3.1. SUBFLUXO: ANALISAR O PROBLEMA**

A Figura 3.3 apresenta o detalhamento do subfluxo *Analisar o Problema*, mostrando as atividades, responsáveis e artefatos de entrada e saída.

Os objetivos desse subfluxo são:

- chegar a um acordo sobre o problema a ser resolvido;
- identificar os Stakeholders;
- definir as fronteiras do sistema;
- identificar as restrições impostas ao sistema.

O primeiro passo a ser dado ao analisar o problema é ter a certeza que todas as partes envolvidas concordam com o que é o problema. Para evitar mal entendido, é importante que todos compreendam a terminologia usada, por este motivo deve ser definido um glossário com os termos usados ao longo do projeto.

Ao planejar a Gerência de Requisitos, é definido o Plano de Gerência de Requisitos que descreve os tipos de requisitos e a documentação de requisitos a ser produzida, especificando informações e mecanismos de controles que devem ser usados para medir, reportar e controlar as mudanças ocorridas requisitos.

O primeiro artefato que relata a análise do problema é o documento de visão, que define uma “visão do produto” para os stakeholders, em termos de suas principais necessidades, provê uma visão de alto nível dos requisitos do sistema, sendo usado como uma base contratual para o detalhamento dos requisitos técnicos.

É importante observar que alguns artefatos produzidos no fluxo de Modelagem de Negócios, como o Modelo de Casos de Uso de Negócios e o Modelo de Objetos de Negócios, são usados com entrada no Fluxo de Requisitos, para auxiliar a encontrar atores e casos de uso e gerar o Modelo de Casos de Uso preliminar, cujo objetivo principal é a identificação dos atores.

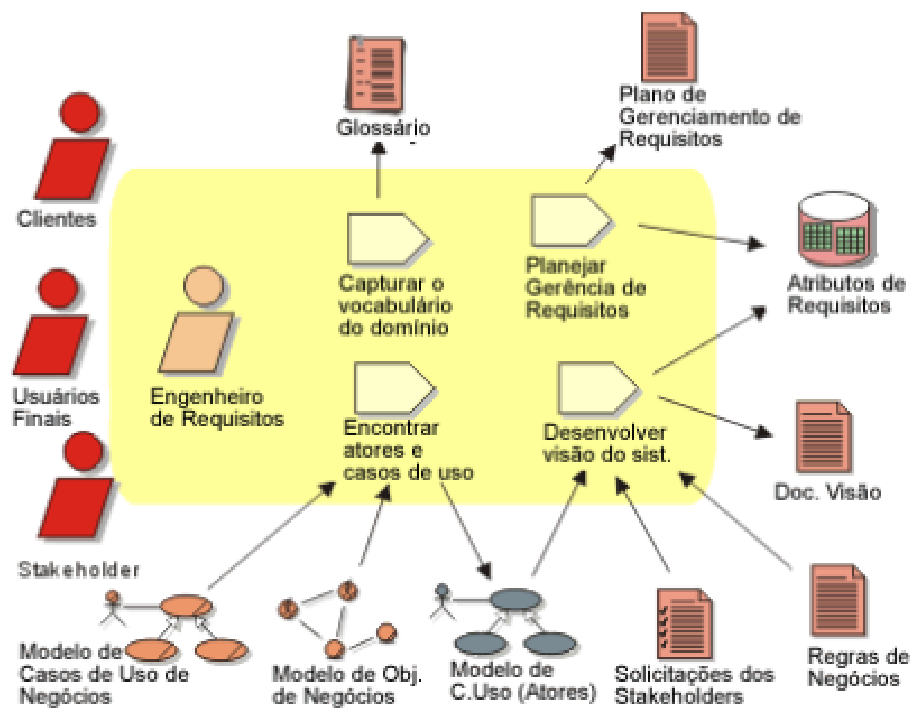


Figura 3.3- Subfluxo Analisar o Problema [Rup02]

### 3.3.2. SUBFLUXO: ENTENDER AS NECESSIDADES DOS STAKEHOLDERES

O objetivo deste subfluxo é coletar e elicitar as informações dos stakeholders e entender as suas necessidades. A apresenta os detalhes do Subfluxo *Entender as Necessidades dos Stakeholders*. Há uma interação com o fluxo *Gerência de Configuração e Mudança* para que,

com o atendimento às Solicitações de Mudança, sejam controladas e gerenciadas as alterações nos artefatos.

A atividade chave deste subfluxo é *Elicitar as Necessidades dos Stakeholders*, cujos objetivos são: entender quem são os stakeholders envolvidos no projeto, coletar os requisitos que devem ser atendidos e priorizar os requisitos. O documento Solicitações dos Stakeholders é refinado com a realização dessa atividade. Após a elicitação, a atividade Desenvolver Visão do Sistema é executada, com isto o documento de Visão e os Atributos de Requisitos são refinados.

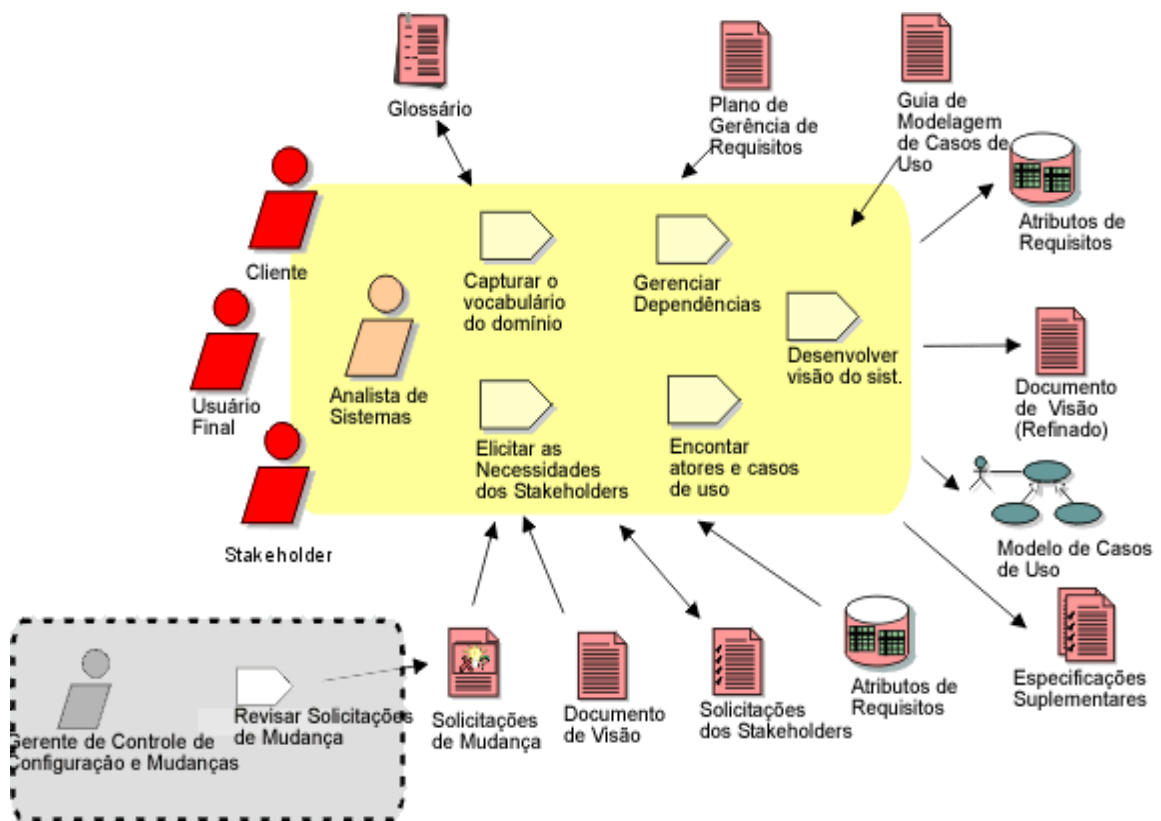


Figura 3.4 - Subfluxo Entender as Necessidades dos Stakeholders [Rup02]

Os objetivos da atividade *Encontrar Atores e Casos de Uso* são: esboçar as atividades do sistema, definir o que deve ser tratado pelo sistema e o que deve ser deixado de fora, definir quem e o quê deverá interagir com o sistema, dividir o modelo em pacotes com atores e casos de uso e desenvolver, brevemente, o modelo de casos de uso. Os artefatos gerados nesta atividade são: versão inicial do modelo de casos de uso, versão inicial da especificação de casos de uso e especificações suplementares.

Ao *Gerenciar Dependências* são analisados os atributos e a rastreabilidade dos requisitos para que as mudanças nos requisitos sejam gerenciadas. Os artefatos gerados nesta atividade são:

a atualização do documento de atributos de requisitos, atualização do documento de visão e atualização do plano de gerência de requisitos.

Outro importante artefato refinado neste subfluxo é o *Glossário* dos termos usados no domínio da aplicação.

### 3.3.3. SUBFLUXO: DEFINIR O SISTEMA

Os principais objetivos deste subfluxo são: alinhar todo o time de desenvolvimento do projeto no entendimento do problema, realizar uma análise de alto nível sobre os requisitos elicitados, refinar a visão do sistema e refinar o modelo de casos de uso, bem como a especificação dos casos de uso. A Figura 3.5 apresenta os detalhes do Subfluxo *Definir o Sistema*.

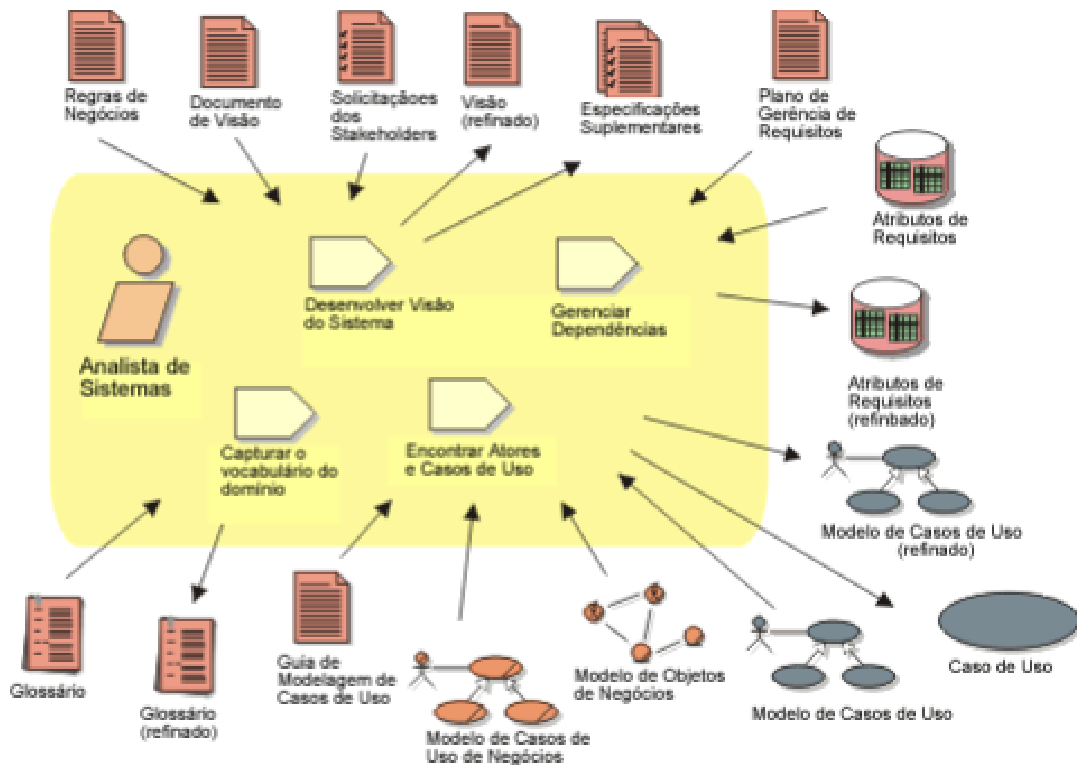


Figura 3.5 - Subfluxo Definir o Sistema [Rup02]

As atividades realizadas nos subfluxos *Analisar o Problema* e *Entender as Necessidades dos Stakeholders* representam uma iteração “preliminar”, onde os conceitos chaves do sistema são definidos e versões iniciais dos documentos de *Visão*, *Modelo de Casos de Uso* e *Atributos de Casos de Uso* são produzidas. Ao *Definir o Sistema*, o foco principal está em completar a identificação dos atores e casos e expandir as definições dos requisitos, incluindo os requisitos não-funcionais no documento de Especificações Suplementares.

### 3.3.4. SUBFLUXO: GERENCIAR O ESCOPO DO SISTEMA

Os principais objetivos deste subfluxo são: priorizar e refinar os requisitos que devem ser incluídos na iteração atual do projeto, definir o conjunto de casos de uso relevantes para a iteração e definir que atributos de requisitos e rastreamentos devem ser mantidos.

O escopo de um projeto é definido pelo conjunto de requisitos alocados a ele, bem como o conjunto de recursos disponíveis para o desenvolvimento (tempo, pessoas e finanças). Assim, gerenciar o escopo do projeto é uma atividade contínua que requer o desenvolvimento iterativo e incremental, onde o escopo é dividido em partes menores, tornando o gerenciamento mais efetivo.

Ao realizar a atividade *Priorizar Casos de Uso* são identificados os cenários e casos de uso que representam as funcionalidades centrais do sistema e aqueles que definem a arquitetura do sistema. Os atributos de requisitos são atualizados, é gerada uma versão inicial do documento da arquitetura do sistema e o glossário é revisado. A Figura 3.6 apresenta detalhes deste subfluxo.

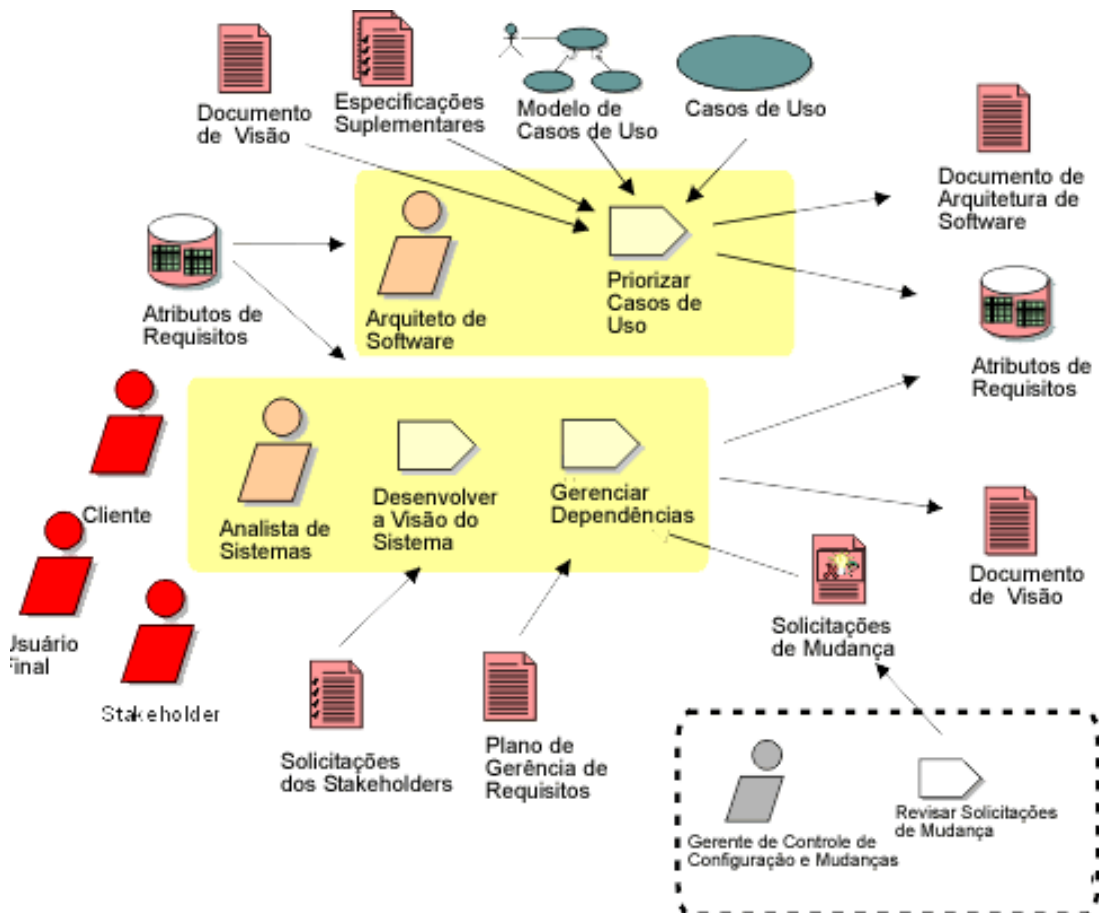


Figura 3.6- Subfluxo Gerenciar o Escopo do Sistema [Rup02]

### 3.3.5. SUBFLUXO: REFINAR A DEFINIÇÃO DO SISTEMA

Os principais objetivos deste subfluxo são: detalhar os fluxos de eventos dos casos de uso, detalhar as especificações suplementares, descrever a especificação de requisitos de software e modelar o protótipo de interface. Na Figura 3.7 podem ser vistos detalhes deste subfluxo, vale a pena observar que com o resultado de sua execução tem-se um profundo entendimento das funcionalidades do sistema, descritas através de casos de uso, especificações suplementares e elementos de interface com o usuário. Também podem ser descritos os requisitos de software, se necessário, em adição aos casos de uso e especificações suplementares.

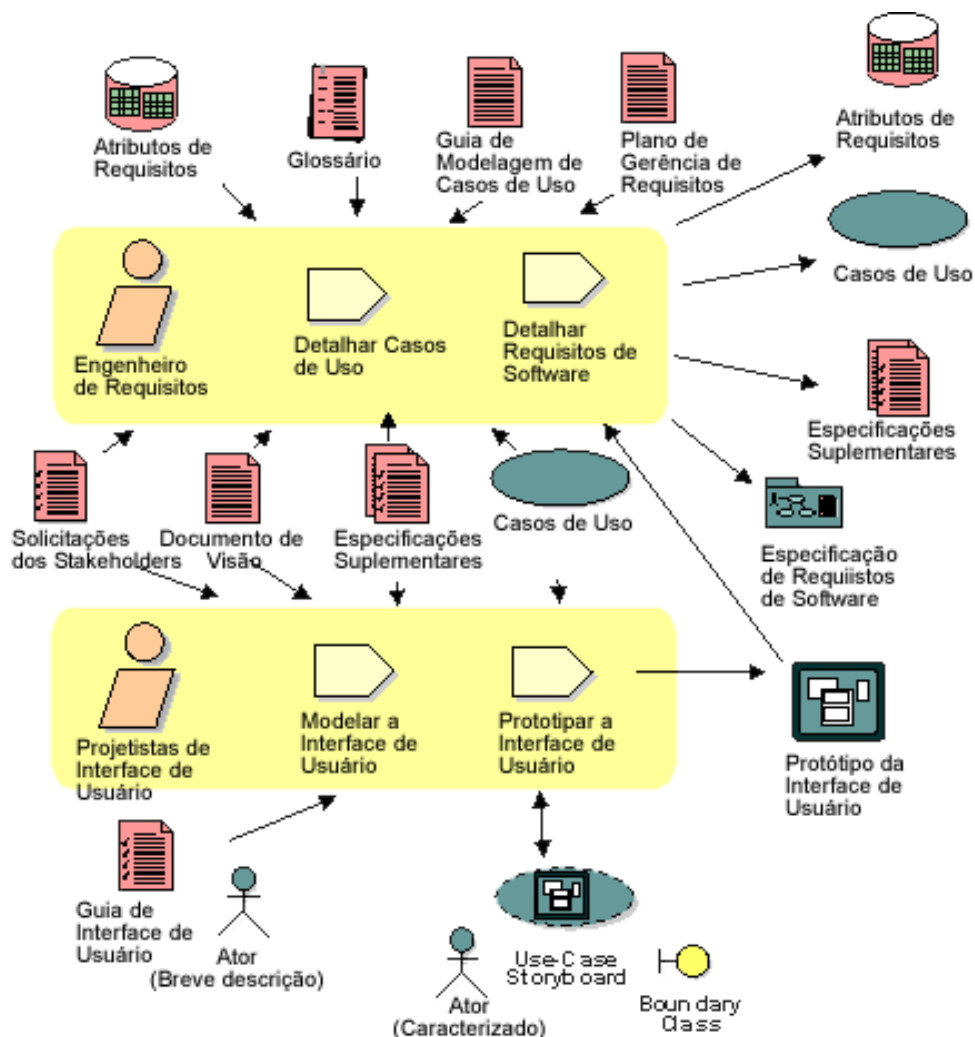


Figura 3.7- Subfluxo Refinar a Definição do Sistema [Rup02]

O refinamento da definição do sistema inicia com o detalhamento dos casos de uso, onde os fluxos dos casos de uso priorizados são descritos em detalhes, de forma que sejam entendíveis pelos usuários.

A atividade *Detalhar os Requisitos de Software* tem o objetivo de coletar, detalhar e organizar o conjunto de artefatos que descrevem os requisitos de software do sistema. Os artefatos gerados após a execução desta atividade são: a especificação dos requisitos de software, a atualização das especificações suplementares e dos atributos de requisitos. Para a execução deste subfluxo, é muito importante trabalhar junto com os usuários do sistema para modelar e desenvolver o protótipo de interface do usuário. A atividade *Modelar a interface do usuário* tem o objetivo de construir um modelo de interface, preocupado, principalmente, com a usabilidade do sistema. A atividade *Prototipar a interface do usuário* tem o objetivo de criar o protótipo da interface com o usuário.

### **3.3.6. SUBFLUXO: GERENCIAR MUDANÇAS NOS REQUISITOS**

Os principais objetivos deste subfluxo são: avaliar formalmente as solicitações de mudança e determinar seu impacto no conjunto de requisitos existentes, estruturar o modelo de casos de uso, estabelecer os atributos dos requisitos e definir rastreabilidade, e verificar se os requisitos definidos estão de acordo com a visão que os usuários têm do sistema. A apresenta detalhes deste subfluxo.

A atividade *Estruturar o modelo de casos de uso* tem como propósitos: extrair o comportamento dos casos de uso que podem ser vistos como casos de usos abstratos, ou seja os relacionamentos de extensão e inclusão, os fluxos alternativos e de exceção, além de possíveis atores abstratos que definem papéis usados por diversos atores. Os artefatos atualizados nesta atividade são: especificação de casos de uso, modelo de casos de uso e pacotes de casos de uso.

Como mudanças nos requisitos podem impactar nos documentos e modelos produzidos nas demais fases de desenvolvimento do software (análise e projeto, implementação, testes, etc), é muito importante gerenciar estas mudanças e rastrear os requisitos. A atividade *Gerenciar dependências* tem o objetivo de usar os atributos e a rastreabilidade dos requisitos do projeto para gerenciar o escopo e as mudanças nos requisitos.

Outra importante atividade deste subfluxo é *Revisar os Requisitos*, onde, através de reuniões de revisão com usuários e clientes, são validados se os requisitos correspondem às suas expectativas.

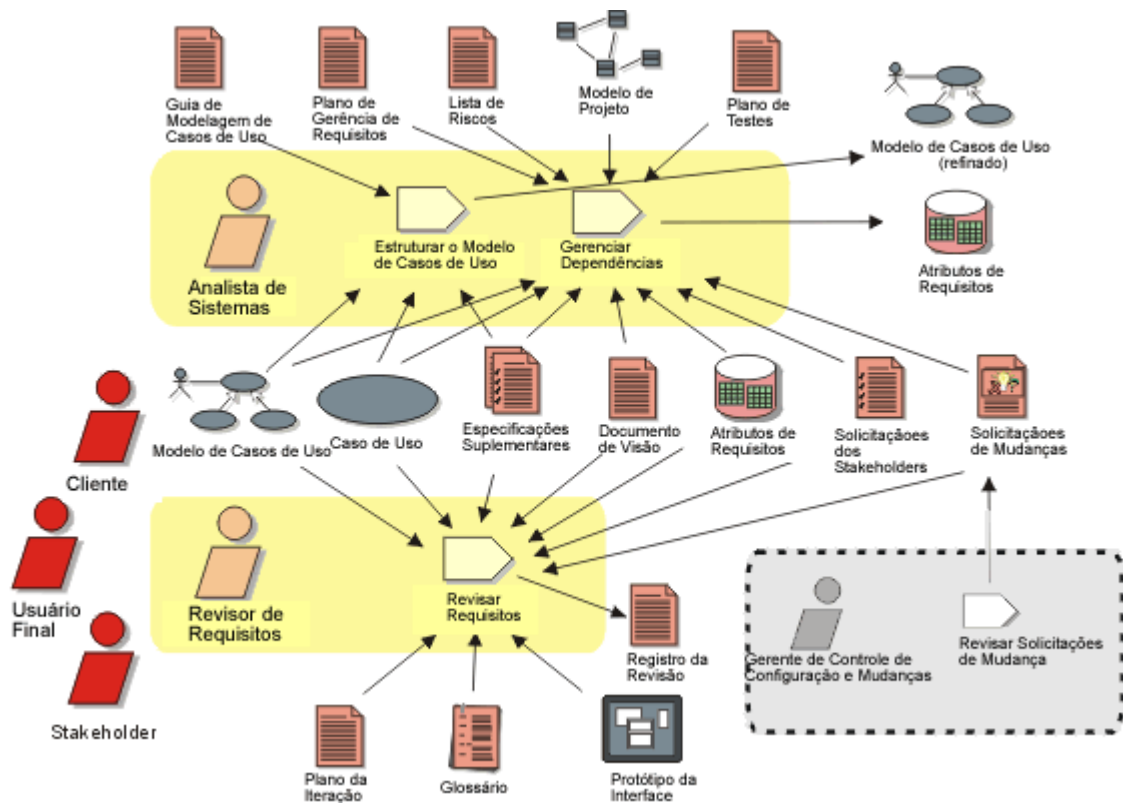


Figura 3.8- Subfluxo Gerenciar Mudanças nos Requisitos

### 3.4. Considerações Finais

Observando o fluxo de requisitos do RUP, podemos fazer algumas considerações a respeito dos artefatos e atividades gerados nos subfluxos que podem ser alterados ou estendidos para atender as diretrizes do Processo Genérico de Engenharia de Requisitos, tornando-o mais efetivo, e considerar características que são fundamentais em projetos direcionados à Web.

Sabe-se que a realização da atividade elicitar os requisitos não é simples. Juntamente com a elicitação, é necessário analisar os requisitos para identificar inconsistências e negociar um acordo sobre o conjunto de requisitos. Analisando as atividades e artefatos do fluxo de requisitos do RUP, sentimos a falta de sistematizar os registros das informações colhidas durante a elicitação e negociação de requisitos. Embora existam diversas técnicas de elicitação (entrevistas, observação e análise social, etnografia, etc), para todas elas há necessidade de manter um histórico de observações realizadas e quem as detectou. Por exemplo, para o caso de entrevistas ou reuniões com clientes, é fundamental documentar atas de reuniões, de forma estruturada para que sirvam como base para o documento de requisitos, além de resguardar os desenvolvedores de eventuais mal-entendidos, especialmente quando há negociação dos



requisitos. Nas aplicações Web, a grande variedade de tipos de usuários, com diferentes experiências e habilidades, torna-se um fator crítico para a elicitación de requisitos

Embora o RUP aborde os requisitos não-funcionais através do documento de especificações suplementares, nota-se que é necessário desenvolver uma estratégia que permita verificar a integração entre os requisitos não-funcionais e funcionais, para que as interdependências entre estes tipos de requisitos possam ser analisadas, diminuindo a possibilidade de futuros conflitos entre as visões não-funcional e funcional do sistema. Além disto, alguns requisitos não-funcionais são cruciais para aplicações Web e precisam ser tratados desde a fase de requisitos, como segurança, performance e usabilidade, por exemplo.

Uma atividade existente em um processo de engenharia de requisitos, mas que não é amplamente especificada no RUP é a validação dos requisitos, onde desenvolvedores e clientes devem concentrar-se na análise do documento de requisitos em relação à sua completude, consistência e precisão e verificar se o mesmo satisfaz, em todos os aspectos, às expectativas de seus usuários e clientes. Na literatura existem diversas técnicas de validação que podem ser aplicadas.

Outro fator importante nas aplicações, principalmente àquelas direcionadas à Web, reside na análise de requisitos de interface com o usuário, onde devem ser elicitadas e definidas as estruturas navegacionais necessárias pelos diferentes usuários da aplicação.

Os pontos supracitados serão tratados no próximo capítulo, onde é definido um processo baseado no RUP, que atende às diretrizes do processo genérico de Engenharia de Requisitos e é aplicável, também, às soluções Web.

## Capítulo 4 WRE-PROCESS: UM PROCESSO PARA A ENGENHARIA DE REQUISITOS, BASEADO NO RUP

---

### 4.1. Visão Geral

Diante da importância que os Requisitos têm em todo o ciclo de vida de desenvolvimento de um software, este capítulo apresenta um processo de Engenharia de Requisitos, baseado no RUP, que contempla as necessidades das aplicações Web. Vale salientar que embora o RUP seja um processo de larga utilização na indústria e adaptável a diferentes tipos de aplicações, este capítulo apresenta um processo para Gerência de Requisitos, mais efetivo que aquele abordado no fluxo de Requisitos do RUP, pois contempla soluções para as questões críticas abordadas no capítulo anterior. Além disto, este modelo considera algumas características do desenvolvimento de aplicações Web, as quais requerem maior atenção em relação ao desenvolvimento de aplicações tradicionais, como: elicitação de requisitos, interface do usuário e a importância dos requisitos não-funcionais.

O processo proposto, chamado *WRE-Process (Web Requirements Engineering Process)*, aborda o desenvolvimento iterativo, sendo guiado pela produção de artefatos. Também são fornecidos templates auto-explicativos que devem ser utilizados pelos desenvolvedores para apoiar e conduzir a realização das atividades que estão sob a sua responsabilidade. Com o desenvolvimento da proposta, procuramos criar um processo que fosse genérico o suficiente para atender diversos domínios e tipos de aplicações, mas que contemplasse necessidades das aplicações web, como: integração entre requisitos funcionais e não-funcionais; elicitação e desenvolvimento de requisitos de interface, entre outros. Desta forma, podemos dizer que a principal contribuição deste processo é fornecer um conjunto coerente de atividades e artefatos direcionados para a Engenharia de Requisitos, considerando características das aplicações Web, mas que mantém a genericidade do RUP, podendo ser aplicado em diferentes tipos de sistemas de software. Além disto, como o trabalho também foi fundamentado em uma revisão bibliográfica da Engenharia de Requisitos e na experiência prática do autor, são propostas algumas atividades e artefatos que aprimoram o processo de Engenharia de Requisitos proposto no RUP.

O principal objetivo a ser alcançado com a utilização deste modelo é a satisfação do cliente, possibilitando que ele receba um produto conforme suas necessidades e expectativas, dentro do prazo e custos previstos, porém sem acarretar prejuízos à empresa fornecedora e sem sobrecarregar a equipe de desenvolvimento.

Como procuramos propor um processo completo para ser customizado a toda e qualquer aplicação, é importante que a equipe de desenvolvimento faça uma análise inicial do processo como um todo e avalie que atividades e artefatos devem ser instanciados para o projeto, de acordo com suas necessidades.

## 4.2. Descrição do Processo WRE-Process

Conforme já mencionado no capítulo anterior, alguns pontos críticos foram identificados no fluxo de Requisitos do RUP. Para resolver estas questões e direcionar a aplicabilidade do RUP aos sistemas desenvolvidos para plataformas Web, propomos o **WRE-Process (Web Requirements Engineering Process)**, um processo para a Engenharia de Requisitos, baseado no RUP, contemplando suas características de adequabilidade aos diversos tipos de aplicações.

Fazendo uma comparação entre o modelo de processo genérico de engenharia de requisitos, apresentado na Figura 2.8, visto no Capítulo 2, e o Fluxo de Requisitos proposto pelo RUP, mostrado na Figura 3.2, verificamos que o RUP não utiliza a nomenclatura padrão do processo genérico, de Sommerville e Kotonya [Som98]. Assim, os subfluxos *Analisar o Problema* e *Entender as Necessidades dos Stakeholders* do RUP correspondem à fase de **Elicitação de Requisitos** do padrão genérico. Enquanto os subfluxos *Definir o Sistema*, *Gerenciar Escopo* e *Refinar a Definição do Sistema* correspondem à **Análise, Negociação e Documentação de Requisitos**. Embora exista, no RUP, uma atividade chamada *Revisar Requisitos*, que é executada no subfluxo Gerenciar Mudanças de Requisitos, observamos que a **Validação dos Requisitos**, no RUP, não recebe a mesma ênfase do processo genérico, onde há uma fase específica para validar requisitos.

A Figura 4.1 representa o fluxo proposto, através de um diagrama de atividades de UML, onde cada atividade do diagrama corresponde a um subfluxo. Este subfluxos encapsulam as atividades e artefatos que devem ser produzidos durante sua execução.

Comparando a Figura 4.1 com a Figura 3.2, que representa o Fluxo de Requisitos do RUP, verifica-se que sugerimos alterações nos nomes de alguns subfluxos para que eles passem a contar com uma descrição mais aproximada daquela usada no processo genérico da Engenharia de Requisitos.

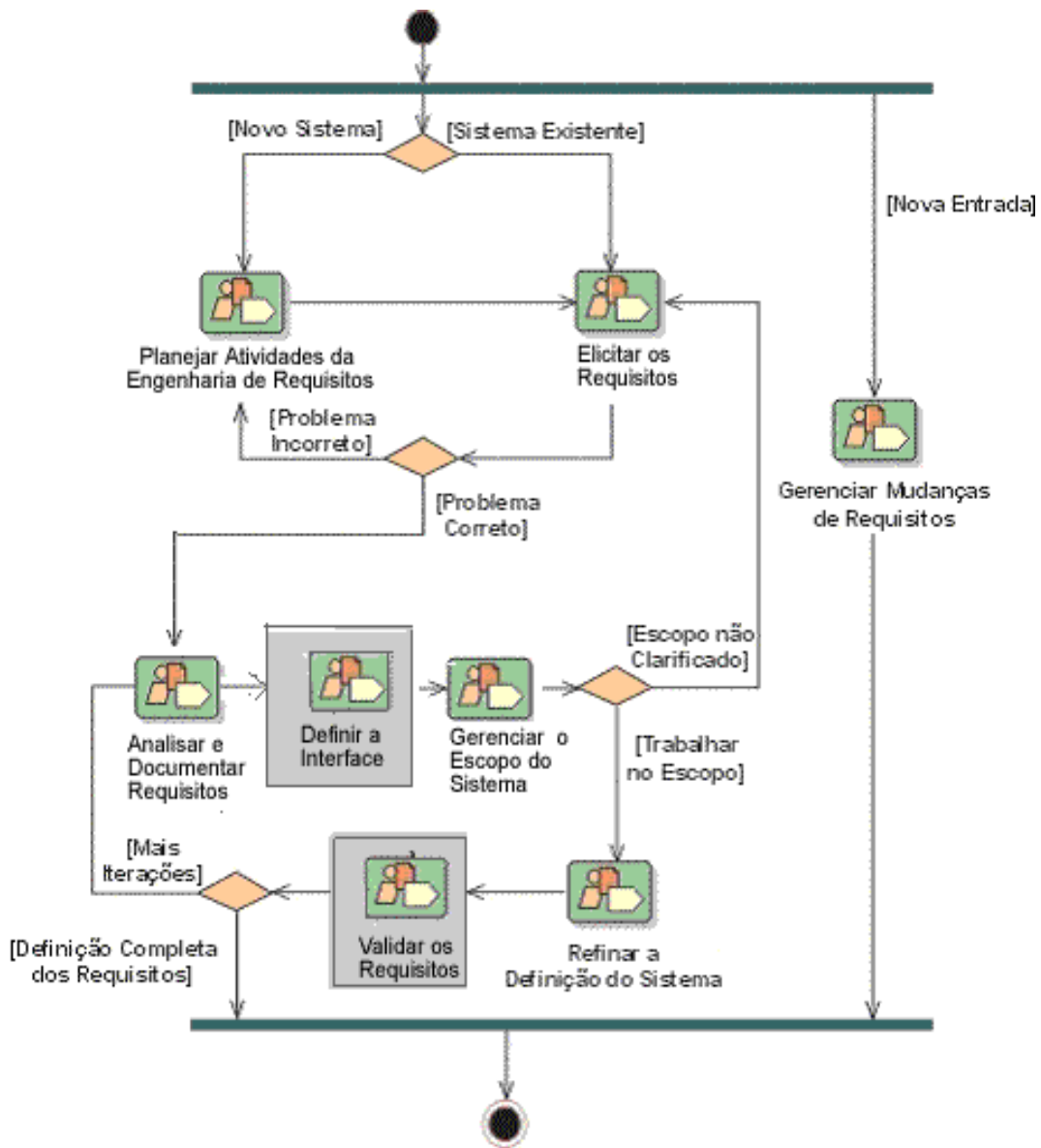


Figura 4.1- Fluxo do Processo WRE-Process

*Analisar o Problema* foi renomeado para **Planejar Atividades da Engenharia de Requisitos** haja vista as atividades executadas neste subfluxo corresponderem a atividades de planejamento. *Entender as Necessidades dos Stakeholders* passou a ser chamado **Elicitar Requisitos** e *Definir o Sistema* foi renomeado para **Analisar e Documentar Requisitos**, conforme as fases do processo genérico. Dois novos subfluxos estão sendo acrescentados nessa proposta: **Definir a Interface** e **Validar Requisitos**. A justificativa para o primeiro fluxo é o fato que, em determinados tipos de aplicações, como aquelas destinadas para Web, por exemplo, a definição

dos requisitos de interface e conseqüentemente, o protótipo navegacional, são aspectos cruciais que devem ser bem definidos durante o fluxo de requisitos de um projeto. Assim, julgamos sua importância tão alta que consideramos este um subfluxo a mais no diagrama de atividades. Quanto a *Validar Requisitos*, já verificamos que há uma lacuna, no RUP, para esta atividade, sendo esta uma fase com grandes desafios para a equipe do projeto, pois é necessário averiguar se a especificação dos requisitos do sistema está correta, para que possíveis problemas sejam sanados de imediato.

Os subfluxos Gerenciar o Escopo do Sistema, Gerenciar Mudanças de Requisitos e Refinar a Definição do Sistema foram mantidos com os mesmos nomes.

Nas subseções seguintes, apresentaremos os subfluxos do *WRE-Process*. Analisaremos os pontos críticos encontrados nos subfluxos correspondentes, do RUP, e daremos sugestões de melhorias baseadas na literatura de Engenharia de Requisitos, bem como em resultados alcançados na experiência prática do autor. Seguindo o modelo adotado no RUP, também usamos diagramas para representar os subfluxos. Estes diagramas são compostos pelas atividades do subfluxo, os artefatos de entrada e saída e os responsáveis pela execução das atividades. As atividades contornadas com um quadrado cinza, estão sendo propostas nesta dissertação, os artefatos grifados correspondem àqueles que foram estendidos ou são novos.

#### **4.2.1. SUBFLUXO: PLANEJAR ATIVIDADES DA ENGENHARIA DE REQUISITOS**

A Figura 4.2 apresenta o detalhamento do subfluxo *Planejar Atividades da Engenharia de Requisitos* através da identificação de atividades, responsáveis, artefatos de entrada e artefatos produzidos.

Ao *Planejar Atividades da Engenharia de Requisitos*, o problema é identificado e as principais atividades executadas são: *Definir a Visão do Sistema*, onde é gerado um documento de visão que apresenta, em alto nível, o sistema a ser construído; *Planejar a Elicitação de Requisitos*, *Planejar a Gerência de Requisitos* e *Planejar a Validação de Requisitos*, onde são planejadas as execuções das principais fases da Engenharia de Requisitos (Elicitação, Validação e Gerenciamento de Requisitos).

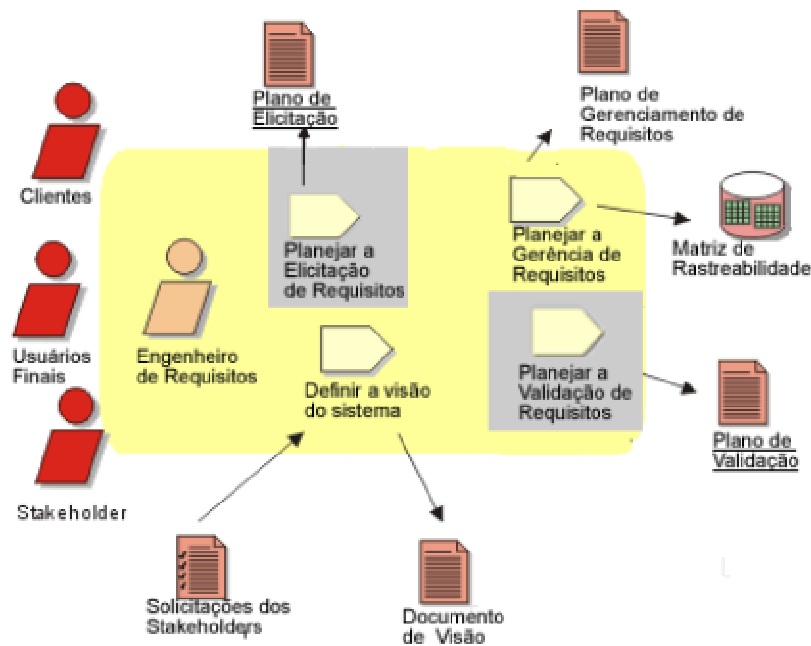


Figura 4.2 - Subfluxo Planejar a Engenharia de Requisitos

São objetivos deste fluxo:

- entender o problema a ser resolvido,
- identificar os stakeholders envolvidos,
- definir o domínio do sistema,
- identificar as restrições impostas ao sistema,
- planejar a execução das atividades da engenharia de requisitos.

A partir de uma reunião inicial com o cliente é obtido o documento de visão preliminar da aplicação. Para aplicações web, destinadas ao grande público, esta conversa introdutória acontece com quem está contratando a aplicação ou com pessoas que conhecem bem o domínio do problema. Esta conversa inicial deve fornecer uma base para determinar os principais domínios da aplicação, bem como as principais restrições. Uma vez tendo capturado o conhecimento preliminar sobre o domínio do problema, sugerimos partir, então, para a escolha de técnicas de elicitação de requisitos. A identificação dos stakeholders é também um fator importante a ser considerado, pois devem ser identificadas as pessoas que contribuirão com o desenvolvimento da aplicação. Outro fator importante que deve ser considerado é o planejamento da validação dos requisitos, haja vista ser necessário definir as técnicas de validação a serem aplicadas.

O subfluxo *Analisar o Problema*, do RUP, contempla a atividade Encontrar Atores e Casos de Uso, com base no modelo de negócios, gerado no subfluxo Modelagem de Negócios.

Como nosso foco é no fluxo de Engenharia de Requisitos e como acreditamos que é mais prudente conhecer melhor a aplicação para então descobrir os atores e casos de uso, deixamos esta atividade para o subfluxo *Elicitar Requisitos*, Seção 4.2.2.4.

Nas subseções a seguir comentaremos as atividades deste subfluxo e os artefatos produzidos nas mesmas: *Planejar a Elicitação de Requisitos*, *Planejar a Gerência de Requisitos*, *Planejar a Validação de Requisitos* e *Definir a Visão do Sistema*.

#### **4.2.1.1. Atividade: Planejar a Elicitação de Requisitos**

Propomos a inclusão desta atividade no processo por acreditarmos ser necessário, antes de iniciar a elicitação dos requisitos propriamente dita, planejar como a elicitação deverá ser conduzida e que técnicas serão utilizadas no projeto.

O primeiro passo é definir os grupos de clientes, usuários e *stakeholders* que devem ser contactados. Estes grupos devem conter pessoas que interagirão, de formas diferentes, com a aplicação, em termos de suas características pessoais e profissionais, interesses, necessidades e habilidades. Em [Kuj00], Kujala sugere contactar também uma classe de “usuários guias”, que são aquelas pessoas que podem revelar mais necessidades, haja vista trabalharem com produtos inadequados e precisarem inventar soluções para resolver seus problemas.

O segundo passo é identificar as técnicas de elicitação que serão aplicadas e planejar como serão realizadas. A escolha da técnica apropriada para elicitar requisitos depende do tempo e dos recursos disponíveis, assim como do tipo de informação necessária (as técnicas de elicitação de requisitos foram vistas no Capítulo 2, seção 2.5.1). Por exemplo, para as técnicas tradicionais como entrevistas e reuniões, devem ser definidas as pautas dos assuntos a serem tratados e os questionários que devem ser aplicados aos stakeholders. Para as técnicas de elicitação em grupo, como *workshops* e *brainstorming*, os materiais a serem apresentados devem ser definidos e preparados.

Em geral, na grande maioria das aplicações e, principalmente para aquelas direcionadas à Web, é fundamental a análise dos produtos existentes<sup>3</sup> e das documentações<sup>4</sup> referentes ao

---

<sup>3</sup> Dentro da organização, deve-se analisar os sistemas legados. Para o desenvolvimento de web sites ou pacotes, por exemplo, deve-se examinar produtos concorrentes e/ou similares.

<sup>4</sup> Corresponde a documentos relacionados com a aplicação, como documentos internos da organização ou artigos referentes ao assunto tratado, por exemplo.

domínio da aplicação. Esta análise facilita o entendimento do problema, por parte dos engenheiros de requisitos. Em aplicações Web, é muito comum a análise de aplicações similares e ou concorrentes como uma fonte de conhecimento do domínio da aplicação. Também são identificados os pontos positivos e negativos destas aplicações, servindo como uma base inicial de referência para o novo produto.

Dessa forma, durante o planejamento da Engenharia de Requisitos, é fundamental definir que produtos ou documentos serão vistos e que tipos de relatórios conclusivos devem ser produzidos.

O plano de elicitação de requisitos é um artefato que encapsula um conjunto de documentos: a relação de usuários candidatos a serem contactados, a agenda de reuniões e entrevistas, as pautas de reuniões e apresentações, questionários, relação de produtos e documentos a serem analisados, inclusive os sistemas legados, se existirem. Apresentamos os templates desenvolvidos para a agenda de reuniões (vide Tabela 4.1) e para a pauta de reuniões (vide Tabela 4.2). Não definiremos aqui os templates dos demais documentos que compõem o plano de elicitação de requisitos por eles serem bastante simples e seguirem o mesmo padrão dos que foram definidos.

*Tabela 4.1- Template para Agenda de Reuniões*

<b>Agenda de Reuniões</b>
<b>1. Objetivos</b> <i>Apresenta os objetivos a serem alcançados com a realização das reuniões.</i>
<b>2. Agenda</b> <i>Nessa seção é apresentada a relação de reuniões agendadas, onde são identificados: data/hora, local, área envolvidas, participantes e objetivo da reunião.</i>
<b>3. Requisitos de Logística para Reuniões</b> <i>Nessa seção são identificados os requisitos de logística necessários, como sala, micro-computador, canhão, etc.</i>
<b>4. Observações</b> <i>Nessa seção são relacionadas observações relevantes.</i>

*Tabela 4.2- Template para Pauta de Reuniões*

<b>Pauta de Reunião</b>
<b>1. Objetivos</b> <i>Apresenta os objetivos a serem alcançados com a realização da reunião.</i>
<b>2. Participantes</b> <i>Relação das pessoas convidadas para a reunião.</i>
<b>3. Tópicos a serem discutidos</b> <i>Nessa seção são identificados os tópicos que devem ser discutidos na reunião.</i>
<b>4. Observações</b> <i>Nessa seção são relacionadas observações relevantes.</i>

O RUP não contempla a atividade *Planejar a Elicitação de Requisitos*, mas sugere a produção do documento “Necessidades dos Usuários”, que serve como um guia para a realização



de entrevistas genéricas com os *stakeholders*. O template proposto no RUP explora aspectos relacionados à funcionalidade, usabilidade, performance e confiabilidade, entre outros. Para sistemas de médio e grande porte, acreditamos que um documento para a realização de entrevistas genéricas não seja suficiente e consideramos necessária a existência de um planejamento para a elicitación de requisitos conforme foi exposto.

Na próxima subseção é apresentada a atividade *Planejar a Gerência de Requisitos*, do Subfluxo *Planejar a Engenharia de Requisitos*.

#### **4.2.1.2. Atividade: Planejar a Gerência de Requisitos**

O gerenciamento apropriado dos requisitos ajuda a avaliar o impacto no projeto quando ocorrerem mudanças nos requisitos, bem como a analisar falhas ocorridas nos testes dos requisitos. Também facilita o gerenciamento do escopo do projeto e a verificar se todos os requisitos são tratados na implementação. O objetivo desta atividade é gerar um plano para gerenciar as mudanças de requisitos.

O RUP sugere a criação de um plano para gerenciar estas mudanças, especificando os mecanismos de controle que devem ser aplicados. Antes de serem iniciadas as descrições dos requisitos, deve-se decidir como documentá-los e organizá-los, que técnica de rastreamento será aplicada e se alguma ferramenta de gerência de requisitos será usada.

O RUP identifica os atributos que caracterizam os requisitos, como por exemplo, nome, tipo, descrição, prioridade, benefícios (da perspectiva dos *stakeholders*), esforço para implementação, riscos, estabilidade (probabilidade de ser alterado) e impacto sobre a arquitetura.

O *Plano de Gerência de Requisitos* define como os requisitos são mantidos e rastreados (entre si e entre os demais artefatos). O analista responsável deve periodicamente gerar relatórios de rastreamento para garantir que o plano de gerência de requisitos está sendo seguido. A Tabela 4.3 apresenta o Template do *Plano de Gerência de Requisitos* sugerido nesta proposta.

*Tabela 4.3- Template do Plano de Gerência de Requisitos*

<p>1. Introdução <i>A introdução apresenta uma visão geral do documento, incluindo seu propósito, o escopo do sistema, as definições/termos e abreviações que são usadas no documento.</i></p> <p>2. Gerenciamento de Requisitos <i>Esta seção descreve quem são os responsáveis pela gerência de requisitos, o ambiente computacional e as ferramentas usadas para gerenciar os requisitos.</i></p> <p>3. Identificação dos Requisitos <i>Descreve como os requisitos são identificados e descritos</i></p> <p>4. Rastreabilidade <i>Identifica a técnica de rastreamento usada, as expressões e convenções usadas nas matrizes de rastreamento, além de definir uma tabela para organizar as classes candidatas ao rastreamento.</i></p>
--

#### 5. Gerenciamento das Mudanças dos Requisitos

*Define os procedimentos necessários para as solicitações de mudanças nos requisitos. Descreve como devem ser submetidas as solicitações de mudanças, avaliadas e executadas. Além disto, são definidas as baselines de alteração dos requisitos*

#### 6. Treinamentos e Recursos

*Define os treinamentos e recursos necessários para uma gerência de requisitos efetiva.*

Em [Tor02-A], Toranzo cita algumas técnicas para gerenciar requisitos propostas na literatura, como por exemplo Gotel (1996), Pohl (1996), Pinheiro (1996), Ramesh (2001) e Toranzo (2002). Ao planejar a gerência de requisitos, alguma técnica deve ser selecionada de acordo com a realidade do projeto.

Nesta proposta adotamos a técnica de Toranzo [Tor02-A], apresentada no Capítulo 2, seção 2.5.5. Para a utilização desta abordagem, devem ser informadas as expressões e convenções usadas nas matrizes de rastreamento e definida uma tabela para organizar as classes candidatas ao rastreamento. Ainda no planejamento, as convenções e estruturas das matrizes de rastreamento são definidas. A Tabela 4.4 apresenta um exemplo de conteúdo para a seção de rastreabilidade (Vide Tabela 4.3), utilizando a técnica de Toranzo; e a Tabela 4.5 apresenta uma sugestão de tabela para organizar as classes candidatas ao rastreamento.

*Tabela 4.4- Exemplo de Rastreabilidade no Plano de Gerência de Requisitos*

1. O nome e tipo da associação representada na matriz são colocados na parte superior esquerda da matriz. O tipo da associação é expresso por um rótulo, por exemplo, o rotulo "<rec>" representada a uma associação de tipo de recurso;  
2. 3. Para a primeira coluna e primeira fila das matrizes de rastreamento usaremos expressões para identificar as instâncias de uma classe. Algumas das expressões são:  
[OBO] = Objetivos organizacionais      [PRO] = Programa do sistema  
[OBS] = Objetivos do sistema          [DGM] = Diagramas  
[REQ] = Requisitos do sistema          [SUB] = Subsistema do sistema

*Tabela 4.5- Sugestão de tabela para organizar as classes candidatas*

Níveis de informação	Análise	Projeto	Implementação	Teste
Ambiental				
Organizacional				
Gerencial				
Desenvolvimento				

Vimos a atividade *Planejar a Gerência de Requisitos*, na próxima subseção detalharemos a atividade *Planejar a Validação de Requisitos*, ainda no Subfluxo *Planejar a Engenharia de Requisitos*.

#### **4.2.1.3. Atividade: Planejar a Validação de Requisitos**

O objetivo desta atividade é gerar o plano de Validação de Requisitos. Este artefato define que técnicas de validação serão aplicadas para que os requisitos possam ser avaliados em relação

à consistência e completude. Algumas destas técnicas foram apresentadas no Capítulo 2, Seção 2.5.4.

É importante o desenvolvimento deste plano porque, com base na técnica escolhida, ou conjunto de técnicas, o gerente de projeto tem informações para planejar o cronograma da atividade de validação, além do fato de poder identificar, de antemão, a necessidade de gerar artefatos específicos para atender a validação.

Caso a técnica de validação selecionada seja *Revisão*, por exemplo, neste plano deve ser identificada a equipe de pessoas responsáveis pela revisão dos requisitos. Idealmente, segundo Sommerville [Som98], os requisitos devem ser revisados por um time multidisciplinar, formado por pessoas com diferentes tipos de experiência, como usuários finais das áreas envolvidas e representantes do cliente. Para as aplicações web, destinadas ao público geral, é mais difícil contar com usuários finais para revisarem os requisitos, então pessoas especialistas no domínio da aplicação devem ser envolvidas.

Embora o plano de validação dependa da técnica selecionada, apresentamos um template para este artefato na Tabela 4.6.

*Tabela 4.6 - Template do Plano de Validação*

1. Introdução

*A introdução apresenta uma visão geral do documento, incluindo seu propósito, o escopo do sistema, as definições/termos e abreviações que são usadas no documento.*

2. Validação dos Requisitos

*Esta seção identifica a técnica de validação usada, ou conjunto de técnicas.*

3. Agenda da Validação

*Esta seção descreve quem são os envolvidos na validação dos requisitos, as datas das reuniões de validação e o escopo de cada reunião, caso necessário, dependendo da técnica escolhida.*

4. Procedimentos Operacionais

*Define os procedimentos necessários para registrar as solicitações de mudanças nos requisitos, decorrentes de especificação não-conforme com as necessidades dos stakeholders. Descreve como devem ser submetidas as solicitações de mudanças, avaliadas e executadas. Além disto, são definidas as baselines de alteração dos requisitos.*

Vimos a atividade *Planejar a Validação de Requisitos*, na próxima subseção detalharemos a atividade *Definir a Visão do Sistema*, ainda no Subfluxo *Planejar a Engenharia de Requisitos*.

#### **4.2.1.4. Atividade: Definir a Visão do Sistema**

Conhecendo as regras de negócios e as principais necessidades dos usuários, é possível gerar o *Documento de Visão* do sistema, cujos objetivos incluem: obter a concordância sobre que problemas devem ser resolvidos, definir os limites do sistema e descrever as características primárias da aplicação. No *Documento de Visão* são definidos os fundamentos conceituais do sistema, e este passa a ser uma referência para a tomada de decisões sobre prioridades e

mudanças. O documento final deve incluir o escopo do projeto e suas limitações, bem como as principais características do software a ser desenvolvido.

A Tabela 4.7 apresenta o template proposto para o documento de visão, com base no modelo do RUP. Este documento tem características similares a uma proposta técnica para o desenvolvimento da aplicação, podendo desempenhar o papel de um contrato preliminar entre a equipe de desenvolvimento e o cliente.

Tabela 4.7 - Template do documento de visão

<p><b>1.Introdução</b> <i>A introdução apresenta uma visão geral do documento, incluindo seu propósito, o escopo do sistema e as definições/termos e abreviações que são usadas no documento.</i></p> <p><b>2. Posicionamento</b> <i>Nessa seção são apresentadas as oportunidades de negócio que devem surgir com o projeto, um sumário do problema a ser resolvido e os impactos por ele causados, bem como os benefícios que a nova solução trará e um posicionamento sobre o novo produto diante do mercado.</i></p> <p><b>3.Descrição dos Stakeholders e Usuários</b> <i>Nessa seção são apresentados os stakeholders e possíveis tipos de usuários da aplicação, bem como o ambiente e possíveis perfis de acesso à aplicação.</i></p> <p><b>4.Visão Geral do Produto</b> <i>Nessa seção é apresentada uma visão geral da aplicação, com suas características gerais. Aqui é descrito um sumário com as principais funcionalidades que devem ser contempladas na aplicação. Essa seção não descreve os requisitos em detalhes, mas provê uma base para seu entendimento.</i></p> <p><b>5. Restrições</b> <i>Nessa seção são descritas as principais restrições sobre as quais o sistema deverá ser desenvolvido e executado.</i></p>
--

Concluimos todas as atividades do Subfluxo *Planejar a Engenharia de Requisitos*, a próxima seção detalha o Subfluxo *Elicitar Requisitos*.

#### **4.2.2. SUBFLUXO: ELICITAR OS REQUISITOS**

A Figura 4.3 apresenta o detalhamento do subfluxo *Elicitar os Requisitos*. Comparando-o com subfluxo *Entender as Necessidades dos Stakeholders*, do RUP 2002, verifica-se que foram adicionadas as atividades *Descobrir Requisitos de Interface* e *Encontrar Requisitos Não-Funcionais (RNF's)*. Além disto, os seguintes artefatos são apresentados neste subfluxo do WRE-Process: *Glossário Léxico, Atas de Reuniões, Análise de Produtos e Documentos, Grafo de RNF*.

A atividade *Descobrir Requisitos de Interface* é fundamental no desenvolvimento de aplicações onde a interface homem-máquina deve ser bem definida, como é o caso de aplicações Web, por exemplo. Esta atividade é realizada pelo Engenheiro de Requisitos em conjunto com o projetista de interface, com o objetivo de capturar as informações necessárias para o projeto de interface, abrangendo questões como usabilidade, navegabilidade e projeto gráfico, além de serem identificadas as necessidades tecnológicas de interface. A atividade *Encontrar Requisitos*

*Não-Funcionais* é sugerida devido à importância que existe na identificação dos Requisitos Não-Funcionais para diversos tipos de aplicações, principalmente àquelas direcionadas à Web.

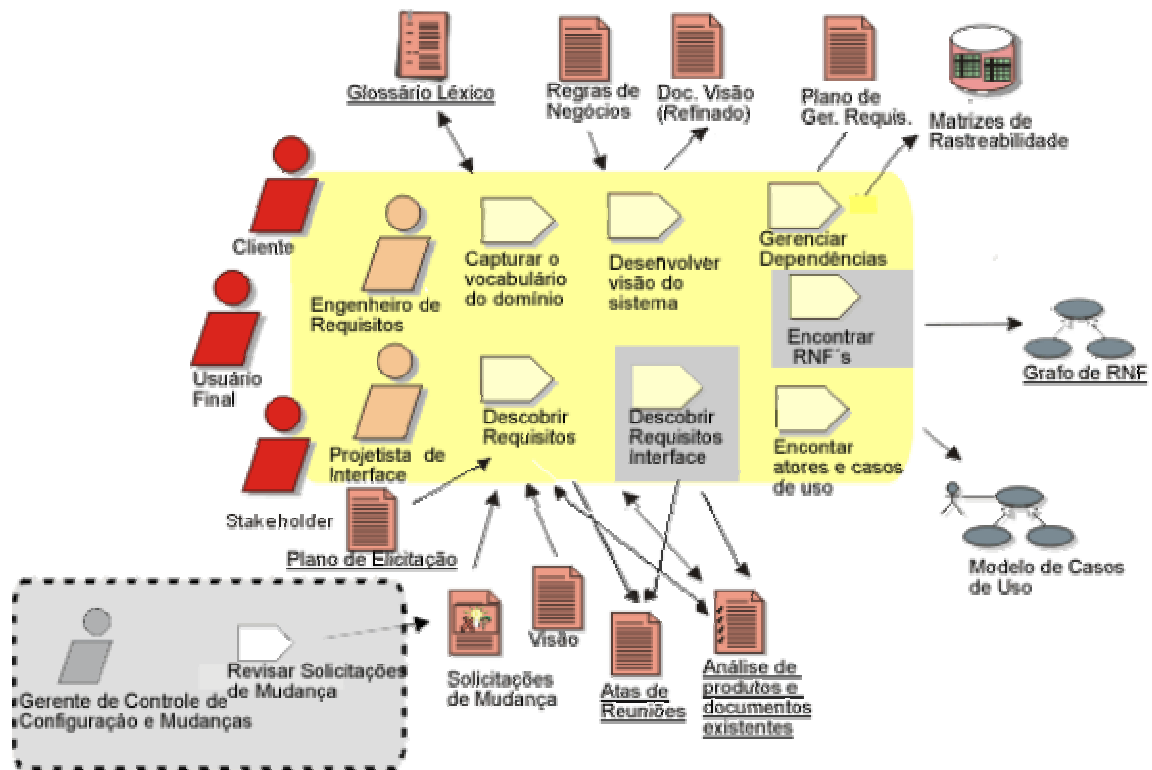


Figura 4.3- Subfluxo Elicitar Requisitos

Devemos salientar a integração entre este subfluxo e o processo de Gerência de Configuração e Mudança que deve ser aplicado no projeto, no que diz respeito ao gerenciamento e controle das solicitações de mudanças, por este motivo há uma área pontilhada na Figura 4.3 contornando a atividade *Revisar Solicitações de Mudança* e o executor *Gerente de Configuração e Mudança*.

Como o processo é iterativo e incremental, algumas atividades são executadas em vários subfluxos. Por exemplo, a atividade *Desenvolver Visão do Sistema* inicia-se no subfluxo *Planejar Engenharia de Requisitos* e também é realizada neste com o objetivo de aprimorar, cada vez mais, o documento de visão, que servirá como base de entendimento do domínio da aplicação. Sendo assim, como esta atividade já foi comentada na seção 4.2.1.4 e é executada novamente devido à característica incremental do processo, não mais a detalharemos e consideraremos que sua execução apenas refina o artefato gerado. Nas subseções a seguir apresentaremos as seguintes atividades deste subfluxo e os artefatos produzidos nas mesmas: *Descobrir Requisitos*, *Descobrir Requisitos de Interface*, *Capturar o Vocabulário do Domínio*,

*Encontrar Atores e Casos de Uso, Encontrar Requisitos Não-Funcionais e Gerenciar Dependências.*

#### **4.2.2.1. Atividade: Descobrir Requisitos**

O objetivo desta atividade é coletar os requisitos do sistema, seguindo o planejamento gerado na atividade *Planejar a Elicitação de Requisitos*, do subfluxo *Planejar Atividades da Engenharia de Requisitos*, seção 4.2.1.1.

Tendo já definido os grupos de pessoas a serem contactadas, os documentos e produtos similares/concorrentes a serem analisados e as técnicas de elicitação que devem ser aplicadas, a elicitação dos requisitos passa a ser realizada com mais agilidade. É claro que, durante a condução da elicitação, podem ser necessários efetuar novos contatos, ou analisar novos produtos e documentos, ou algumas técnicas sejam alteradas. Entretanto, ter o planejamento da elicitação pronto, com cronogramas, agendas e pautas de reuniões e a relação de produtos e documentos estabelecidos, torna a gerência do projeto mais efetiva, permitindo minimizar os riscos.

Desta forma, o planejamento realizado no subfluxo *Planejar Atividades da Engenharia de Requisitos*, deve ser seguido durante a elicitação de requisitos e documentadas as necessidades identificadas, bem como restrições e impactos apresentados.

Esta atividade tem como artefatos de entrada o *Plano de Elicitação de Requisitos*, o *Documento de Visão e as Solicitações de Mudança* (do processo de gerência de configuração e mudança). Durante a elicitação dos requisitos são produzidas *Atas de Reuniões* e *Análise de Produtos e Documentos Existentes*.

Em reuniões com *stakeholders*, por exemplo, devem ser redigidas atas que contemplem todos os assuntos abordados e indicados data, hora, local e participantes das reuniões. A Tabela 4.8 apresenta um template de ata de reunião. Na análise de produtos ou documentos já existentes, como sistemas similares, sistemas legados, contratos, legislações, etc., os pontos relevantes identificados devem ser registrados em Relatórios de Análise destes produtos, sendo fundamental informar as pessoas que participaram da análise, data e hora da realização.

*Tabela 4.8- Template de Ata de Reunião*

**1.Objetivo**

*Apresenta o objetivo da reunião, data, hora e local da realização da mesma.*

**2.Participantes**

*Apresenta os participantes da reunião.*

**3. Tópicos discutidos/definições**

*Apresenta os tópicos discutidos na reunião e os conceitos definidos.*

**4.Próximas Ações**

*Apresenta as próximas ações a serem tomadas, quem são os responsáveis e as datas previstas para conclusão.*

**5.Próxima Reunião**

*Data, hora e local da próxima reunião, caso seja agendada.*

Dependendo da técnica de elicitação usada, estes artefatos podem variar, mas é necessário que sejam documentados para que se possa manter um histórico da origem dos requisitos, permitindo rastreá-los no futuro.

Descrevemos a atividade *Descobrir Requisitos*, na próxima subseção detalharemos a atividade *Descobrir Requisitos de Interface*, ainda no Subfluxo *Elicitar Requisitos*.

#### 4.2.2.2. Atividade: Descobrir Requisitos de Interface

Esta atividade é sugerida devido à grande importância que os requisitos de interface exercem em determinados tipos de aplicações, como àquelas direcionadas à Web, por exemplo. Para a realização desta atividade é necessária a presença do Projetista (*Designer*) de Interface, que, junto com o Engenheiro de Requisitos, coletará as necessidades do cliente e usuários para produzir o padrão de interface e o modelo navegacional do software.

Um dos problemas relacionados à aplicação do RUP a projetos Web, reside no fato que ele é focado no objetivo da aplicação, não considerando artefatos e atividades direcionadas para a captura e especificação de requisitos característicos da Web, como interação, navegação e hipermídia. A descoberta de requisitos de interface deve procurar subsídios que auxiliem a:

- I. Definir as estruturas navegacionais necessitadas pelos diferentes usuários da aplicação, abrangendo usabilidade, navegabilidade e design gráfico;
- II. Identificar as necessidades tecnológicas de interface, visando facilitar o desenvolvimento do projeto, a realização dos testes e a homologação da interface por parte do cliente.

Segundo Neves, em [Nev99], começando com observações empíricas realizadas por profissionais das áreas de design e ciência da computação, o processo de análise dos requisitos deve identificar os objetivos e conteúdo do projeto, para que sejam identificadas as melhores alternativas e selecionada aquela que melhor satisfaça as expectativas dos usuários e clientes.

Da perspectiva de projetos de interface orientados a usuários, Medeiros cita, em [Med00] três tipos de modelos existentes no processo de comunicação homem-máquina usados na execução de uma tarefa:

1. **O modelo mental que o usuário tem de sua tarefa e do sistema usado:** corresponde a uma representação interna da tarefa e do sistema. Este modelo desenvolve-se naturalmente com a interação usuário-sistema.

2. **A imagem do sistema, ou sua “interface de uso”:** corresponde aos elementos do sistema com os quais o usuário tem contato e que permitem ao mesmo terem conhecimento sobre o sistema.
3. **O modelo conceitual do sistema:** corresponde a uma representação precisa e consistente do sistema. Este modelo é fornecido pelos projetistas de interface (*designers*), engenheiros, etc., para prover uma representação apropriada.

Em sistemas bem projetados, o modelo conceitual do projeto forma uma base para a “interface de uso” do sistema, que por sua vez é uma base para o modelo mental dos usuários. Um bom projeto inicia com o modelo conceitual e termina com a imagem do sistema, materiais de treinamento e modelos mentais consistentes com os modelos conceituais. A base para um bom modelo conceitual é a análise de potenciais usuários, as tarefas e sistemas por eles executados.

Com base nos conceitos expostos e em nossa experiência prática, sugerimos a realização de entrevistas com os stakeholders e a aplicação de um questionário para Descobrir os Requisitos de Interface com usuários [Ces03-A], [Ces03-B]. O questionário deve levantar questões relacionadas a:

- Objetivos do sistema: objetivos gerais e a longo prazo
- Usabilidade desejada: público alvo, perfis de usuários, características de comportamento do público alvo, etc.
- Identidade visual: associação entre o sistema e a marca da empresa, logotipo próprio, personalização do sistema por tipo de aplicação, etc.
- Expectativas do cliente: a interface do novo sistema e relação de softwares ou sites que agradam ou desagradam em relação à usabilidade e identidade visual.

Da mesma forma como tratamos a atividade *Descobrir Requisitos*, durante a atividade *Descobrir Requisitos de Interface* também devem ser registradas atas de reuniões e relatórios de avaliações dos produtos existentes (similares e/ou concorrentes), para que seja mantido um histórico dos requisitos. Esses relatórios de avaliação de produtos existentes correspondem a uma análise comparativa entre as interfaces de aplicações similares ou concorrentes, levando em consideração questões como *layout*, navegabilidade, tecnologia, personalização, etc.

A Tabela 4.9 apresenta um exemplo de Questionário de Requisitos de Interface (chamado de Questionário de Briefing de Design, em [Ces03-A] e [Ces03-B]).



Tabela 4.9 - Exemplo de Questionário de Requisitos de Interface

<p><b>Objetivos</b></p> <ol style="list-style-type: none"><li>1. Qual o objetivo geral do sistema?</li><li>2. Quais os objetivos específicos a curto e longo prazo?</li></ol> <p><b>Usabilidade</b></p> <ol style="list-style-type: none"><li>3. Qual o público alvo do sistema?</li><li>4. Dentro do público alvo, existe mais de um perfil de acesso? Quais são?</li><li>5. Quais as características de comportamento que esse público possui?</li><li>6. Com que assiduidade será usado o sistema?</li></ol> <p><b>Identidade Visual</b></p> <ol style="list-style-type: none"><li>7. Numa escala de 0 a 5, qual a necessidade de associação entre o sistema e a marca da empresa?</li><li>8. Numa escala de 0 a 5 qual o grau de autonomia visual do site em relação à comunicação visual da empresa?</li><li>9. O sistema possuirá um logotipo próprio?</li><li>10. Considerando-se os desdobramentos do sistema nas próximas iterações, e que existe um interesse de uma unidade visual do Sistema, de 0 a 5 qual o nível de personalização desejada para cada futuro setor?</li></ol> <p><b>Interface</b></p> <ol style="list-style-type: none"><li>11. Em relação à interface, quais os pontos positivos do sistema que está hoje no ar?</li><li>12. Quais os pontos negativos?</li><li>13. Indique uma lista de até 3 (três) softwares que você considera de fácil uso. O que chama a atenção nesses softwares?</li><li>14. Indique ao menos um software que você considera de uso inconveniente, por que?</li><li>15. Indique uma lista de 3 (três) serviços online que você considera de boa qualidade visual e fácil uso (Ex.: Internet banking, loja online etc)</li><li>16. Do ponto de vista exclusivamente visual, indique uma lista de até 3 (três) sites corporativos que lhe agradam. Por que?</li><li>17. Do ponto de vista exclusivamente visual, indique uma lista de até 3 (três), sites corporativos que lhe desagradam. Por que?</li><li>18. Em poucas palavras, quais as expectativas para a interface do novo sistema?</li></ol>
---

Descrevemos a atividade *Descobrir Requisitos de Interface*, na próxima subseção detalharemos a atividade *Capturar o Vocabulário do Domínio*, ainda no Subfluxo *Elicitar Requisitos*.

#### 4.2.2.3. Atividade: Capturar o Vocabulário do Domínio

O primeiro passo a ser executado ao analisar o problema é garantir que todos os envolvidos concordam com o que é o problema e o que estão tentando resolver e, para evitar mal entendidos é importante que seja definido um glossário com os termos usados no universo de informações da aplicação. Neste trabalho, propomos uma extensão do glossário de termos do RUP para que o mesmo passe a contemplar algumas características do Léxico Ampliado da Linguagem (LAL) [Cys01-A], [Kap00], [Cys01-C] visto no Capítulo 2, seção 2.4.1.1. O nosso objetivo ao sugerir esta extensão é evitar ambigüidades e inconsistências na definição do glossário e, conseqüentemente, nos demais artefatos que o usam como base.

O princípio básico deste glossário estendido, que chamaremos de *Glossário Léxico* é a definição de um vocabulário bem definido do Universo de Informações (UdI), construído através

de símbolos, definições e impactos, onde o vocabulário do domínio da aplicação seja capturado, bem como sua semântica, de forma que seja utilizado um vocabulário mínimo<sup>6</sup>, seguindo o princípio da circularidade<sup>7</sup>, provendo uma maior compreensão do Universo de Informações onde a aplicação estará inserida.

O LAL é utilizado nesta proposta devido a: (i) o papel que ele desempenha na “*baseline de requisitos*” (conforme apresentado no capítulo 2), atuando como uma ferramenta para apoiar o conhecimento da linguagem utilizada no domínio; (ii) possibilidade de funcionar como âncora para a produção dos modelos funcionais e não-funcionais e favorecer a integração destes dois modelos e (iii) por ajudar na elicitación de requisitos.

É importante salientar que, como o LAL, o *Glossário Léxico*, não é restrito apenas às informações relacionadas aos requisitos funcionais [Cys01-B]. A idéia, neste trabalho, é registrar todo o vocabulário do Universo de Informações e isto pode incluir os aspectos não-funcionais do sistema. Para atingirmos este objetivo, podemos utilizar a extensão do LAL proposta por Cysneiros em [Cys01-A], onde é permitido explicitar que um determinado símbolo possui um ou mais RNF’s a ele associados, ou seja, um determinado símbolo precisa satisfazer determinados RNF’s para atender às expectativas do cliente, e o que é preciso para satisfazer este RNF.

A nossa proposta é que o *Glossário Léxico* seja uma ferramenta que tanto clarifique o domínio da aplicação, para a equipe de desenvolvimento, como permita o entendimento, por parte do cliente, de alguns termos técnicos essenciais que deverão aparecer nos demais artefatos, por exemplo, a definição de protocolo http, html, etc. Contudo, nem todos os símbolos identificados possuem impactos associados.

Um exemplo de parte de um *Glossário Léxico* é apresentado na Tabela 4.10, para uma operadora de Cartão de Crédito, em ambiente Web. Vale salientar que os termos são definidos na coluna símbolos e os mesmos são classificados em sujeito, objeto ou ação. Na coluna Definições são apresentadas as noções que descrevem cada símbolo, na coluna impactos são determinadas as respostas comportamentais de cada símbolo e, finalmente, na coluna RNF’s Associados, são identificados os requisitos não-funcionais que devem ser satisfeitos. Ao utilizarmos o princípio da circularidade, os símbolos referenciados na descrição das noções ou impactos de outro símbolo

---

<sup>6</sup> O princípio do vocabulário mínimo dita que a descrição de um símbolo deve utilizar um vocabulário restrito da linguagem natural, minimizando o uso de símbolos que são externos ao léxico [Lei00],[Cys98].

<sup>7</sup> O princípio da circularidade dita que na descrição de noções (definições) e impactos sejam empregados os próprios símbolos da linguagem, maximizando o uso de símbolos na descrição de outros símbolos. [Lei00],[Cys98].

devem estar sublinhados para ficar claro ao leitor que há uma definição para aquele termo utilizado (noção de hipertexto).

Tabela 4.10- Exemplo de Glossário Léxico (Extensão proposta ao glossário do RUP)

Símbolo	Tipo	Definições	Impactos	RNF's Associados
Cartão de Crédito	Objeto	1. Cartão individual de conferência de assinatura e dados cadastrais, confeccionado em plástico, administrado pela operadora e por ela concedido, em caráter pessoal e intransferível, ao <u>cliente Titular</u> e ao <u>Cliente Dependente</u> , para aquisição de bens e/ou serviços nos <u>estabelecimentos credenciados</u> .	<ol style="list-style-type: none"> <li>1. Cada cartão possui prazo de validade certo, determinado e renovável</li> <li>2. Os cartões são aceitos para compras apenas nos <u>estabelecimentos credenciados</u></li> </ol>	SEGURANÇA: Para este tipo de aplicação, a segurança das informações é um requisito primordial.
Cliente	Sujeito	1. Pessoa usuária do Sistema de Cartão de Crédito, detentora do cartão de crédito	1. Devem ter acesso ao site através de senha.	CONFIDENCIALIDADE
Cliente Titular	Sujeito	1. Cliente, detentor do cartão de crédito e responsável pela conta onde são lançados os débitos e os créditos relativos à concessão, manutenção e utilização do cartão.	<ol style="list-style-type: none"> <li>1. Para ser cliente, seu cadastro deve ser, antes, aprovado pela Operadora.</li> <li>2. É necessário comprovar renda e ser maior de idade.</li> </ol>	CONFIDENCIALIDADE
Cliente Dependente	Sujeito	1. Pessoa física indicada pelo <u>Cliente Titular</u> para ser usuária do <u>Sistema de Cartão de Crédito</u> e portadora de <u>cartão de crédito</u> .	1. Gastos e despesas serão da exclusiva responsabilidade do <u>Cliente Titular</u> .	CONFIDENCIALIDADE
Consultar Extrato	Ação	1. Consulta que disponibiliza os <u>extratos aos clientes</u>	<ol style="list-style-type: none"> <li>1. Os <u>extratos</u> podem ser consultados por cartão e por período.</li> <li>2. O sistema deve validar se o usuário tem acesso à funcionalidade e ao extrato que está sendo solicitado.</li> </ol>	SEGURANÇA USABILIDADE
Efetuar Login	Ação	1. Ação executada pelo usuário ao identificar-se, através de seu login e senha, para iniciar uma sessão de trabalho.	<ol style="list-style-type: none"> <li>1. O login deve ser único para cada usuário da aplicação.</li> <li>2. Ao ser efetuado o login o sistema deve validar o login e senha estão corretos.</li> <li>3. Caso a senha seja digitada errada, por mais de três vezes consecutivas, o sistema deve bloquear o login do usuário.</li> <li>4. É desejável que esta ação tenha uma boa performance, embora diversas validações sejam necessárias.</li> </ol>	SEGURANÇA PERFORMANCE Como o sistema estará disponível no ambiente web, os usuários deverão possuir login (número do cartão) e senha para executar determinadas atividades. Para avaliar o login e senha do usuário, diversas validações podem ser necessárias, mas mesmo assim, o requisito performance deve ser atendido.
Extrato	Objeto	1. Documento que apresenta as movimentações realizadas pelos <u>clientes</u> do <u>cartão de crédito</u> .	1. Os extratos são documentos mensais e suas informações devem ser íntegras e corretas.	INTEGRIDADE, CONFIDENCIALIDADE

Futuro Cliente	Sujeito	1. Qualquer pessoa candidata a ser usuária do <u>Cartão de Crédito</u> .	1. Os futuros clientes podem <u>Realizar proposta</u> de adesão ao Cartão de Crédito	
Realizar proposta	Ação	1. Solicitar uma proposta de adesão ao <u>Cartão de Crédito</u> , através de informações dos dados pessoais, profissionais e referências do solicitante.	1. Qualquer pessoa poderá Realizar uma proposta. 2. O Captador de Clientes é responsável por analisar e aprovar ou reprovar a proposta.	
Solicitar Segunda via do cartão	Ação	1. Solicitação para emissão de segunda via de um determinado <u>cartão de crédito</u> .	1. Apenas o <u>cliente titular</u> deve ter acesso a esta funcionalidade.	SEGURANÇA
Protocolo http	Objeto	1. Hypertext Transfer Protocol 2. Protocolo que assegura a transmissão dos documentos <u>HTML</u> .		DESENV. WEB
Html	Objeto	1. Hypertext Markup Language 2. Bloco de construção fundamental da World Wide Web, 3. Linguagem de programação usada para criar páginas Web.		

A produção do *Glossário Léxico* deve seguir o enfoque iterativo e incremental do RUP, ou seja, o mesmo é aprimorado sempre que um novo símbolo é encontrado. Os benefícios a serem atingidos com a produção deste artefato são: permitir que a equipe envolvida no projeto entenda a linguagem do usuário, facilitar a produção do documento de requisitos e a identificação de atores e casos de uso do sistema, bem como as restrições de negócios envolvidas.

Descrevemos a atividade *Capturar o Vocabulário do Domínio*, na próxima subseção detalharemos a atividade *Encontrar Atores e Casos de Uso*, ainda no Subfluxo *Elicitar Requisitos*.

#### 4.2.2.4. Atividade: Encontrar Atores e Casos de Uso

Para encontrar atores e casos de uso, são utilizados, como provedores de informação, os artefatos gerados nas atividades previamente realizadas (*Desenvolver Visão do Sistema*, *Descobrir Requisitos*, *Capturar o Vocabulário do Domínio*, entre outras), como *Documento de Visão*, *Atas de Reuniões*, *Documentos de Análise de Produtos Existentes* e o *Glossário Léxico*. Como vimos na seção anterior, os termos do *Glossário Léxico* são classificados em sujeito, ação e objeto. Com base nestas informações, torna-se direto o mapeamento de sujeitos em atores e ações em possíveis casos de uso [Cys01-C]. Sendo assim, um modelo de casos de uso preliminar já pode ser gerado nesta etapa do projeto.

Os artefatos gerados no Subfluxo de Modelagem de Negócios também auxiliam na descoberta de Atores e Casos de Uso. Caso a modelagem de negócios tenha sido realizada através da técnica *i\**, por exemplo, também é possível identificar atores e casos de uso, utilizando a

técnica de Santander, vide [San02]. Neste trabalho, não iremos considerar tais artefatos, visto que não fazem parte do escopo desta dissertação.

O artefato gerado com a execução desta atividade é o modelo de casos de uso, seguindo a notação UML [Bit03]. A Figura 4.4 apresenta um *Modelo de Casos de Uso* preliminar, onde são representadas as associações entre os Atores e Casos de Uso identificados para um site de uma operadora de cartão de crédito. Com base no *Glossário Léxico* apresentado na Tabela 4.10 e no *Documento de Visão* (não apresentado para este exemplo), são identificados, inicialmente, os atores *cliente* e *futuro cliente*. Alguns casos de uso podem ser executados tanto pelos clientes como pelos futuros clientes, são aquelas funcionalidades destinadas ao público em geral como consultar promoções e pesquisar os estabelecimentos conveniados com a operadora. Um cliente, para acessar determinados serviços do site, deverá efetuar login e, em seguida, poderá utilizar os seguintes serviços: alterar senha, consultar extratos, consultar saldo, solicitar segunda via do cartão, solicitar alterações cadastrais, solicitar inclusão de dependente. O futuro cliente, que representa àquelas pessoas que ainda não possuem o cartão de crédito da operadora, mas que navegam pelo site, pode, além, daquelas funcionalidades públicas já mencionadas, realizar uma proposta para adquirir um cartão de crédito e consultar a proposta submetida.

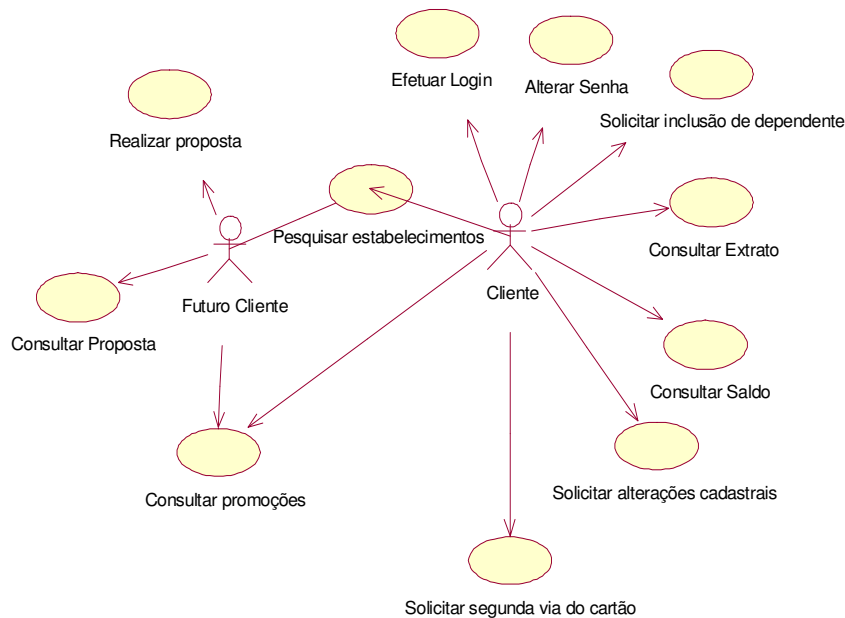


Figura 4.4 - Modelo de Casos de Uso Preliminar

Descrevemos a atividade *Encontrar Atores e Casos de Uso*, na próxima subseção detalharemos a atividade *Encontrar Requisitos Não-Funcionais*, ainda no Subfluxo *Elicitar Requisitos*.

#### **4.2.2.5. Atividade: Encontrar Requisitos Não-Funcionais**

Esta atividade é sugerida nesta dissertação devido à importância que os Requisitos Não-Funcionais (RNF's) exercem em determinados tipos de aplicações, como àquelas direcionadas à Web, por exemplo.

Os Requisitos Não-Funcionais vêm sendo citados em vários processos de desenvolvimento de software, como restrições ou condições de contorno, mas sempre de modo informal, do ponto de vista da elicitação de requisitos. Este tipo de tratamento causa, aos desenvolvedores, dificuldades para tratar os RNF's adequadamente, gerando, muitas vezes, insatisfação do cliente.

Poucos trabalhos propõem um tratamento explícito para os RNF's, sob a ótica do desenvolvimento de software. O RUP, por exemplo, aborda os Requisitos Não-Funcionais através do documento de especificações suplementares, mas verificamos que há uma lacuna na integração entre os Requisitos Funcionais e Não-Funcionais. Portanto, isto é um dos grandes desafios da Engenharia de Requisitos, principalmente para aplicações Web. Frequentemente, quando um RNF é adicionado a uma especificação de requisitos, são forçadas tomadas de decisões que podem afetar positiva ou negativamente outros requisitos. A integração entre os Requisitos Funcionais e Não-Funcionais deve permitir a análise das interdependências entre estes tipos de requisitos, diminuindo a possibilidade de futuros conflitos entre as visões não-funcional e funcional do sistema. Um RNF pode influenciar positivamente outro RNF contribuindo para sua satisfação, bem como negativamente, contribuindo para que um dos RNF's não seja satisfeito ou seja satisfeito parcialmente.

A estratégia de elicitação dos RNF's inicia com a montagem do *Glossário Léxico*, pois alguns RNF's aparecem naturalmente durante o levantamento dos símbolos e suas definições, devido ao fato de terem grande destaque em determinados domínios de aplicações, como, por exemplo, o RNF *Segurança* no segmento bancário.

Como o processo é iterativo e incremental, podemos dizer que as primeiras versões do *Glossário Léxico* só captarão os RNF's que se destaquem demasiadamente. A inclusão destes RNF's no *Glossário Léxico* e sua representação através de mecanismos que permitam aos

desenvolvedores terem os requisitos não-funcionais sempre em mente são os objetivos desta atividade.

O trabalho proposto por Cysneiros em [Cys01-A], [Kap00] propõe uma estratégia para tratar RNF's e integrá-los aos Requisitos Funcionais. Para representar os RNF's é utilizado o NFR Framework, através dos grafos de RNF's, propostos por Chung em [Chu00]. Segundo Cysneiros, o primeiro passo é definir o Léxico Ampliado da Linguagem (LAL) do Universo de Informações (UdI). Uma vez obtido o LAL, parte-se para a construção dos modelos funcionais e não funcionais. Os dois são construídos separadamente, não havendo, a princípio, uma preocupação em relacioná-los. O tratamento de interdependências é feito utilizando-se o grafo de RNFs. Para construir o modelo de RNF's deve-se buscar, em todas as entradas do LAL, as noções que expressam a necessidade de um RNF. Para cada RNF deve-se criar um grafo de RNF's expressando todas as operacionalizações necessárias para satisfazê-lo. No final do processo, tem-se um conjunto de grafos de RNF's que representam os aspectos não-funcionais do sistema [Cys01-A], [Cys01-B], [Cys01-C].

Aplicando a proposta de Cysneiros [Cys01-A], [Cys01-B], [Cys01-C] no WRE-Process, podemos considerar que estando os RNF's captados no *Glossário Léxico*, partimos para identificá-los e representá-los através dos *Grafos de RNF's* (artefato produzido nessa atividade). Passamos então a realizar as operacionalizações necessárias para encontrar as interdependências, positivas ou negativas, entre eles. A utilização de catálogos ajuda a encontrar as operacionalizações, pois estes armazenam conhecimento acumulado em experiências prévias e são fontes de informações sobre requisitos não-funcionais [Chu00]. Numa iteração inicial, são identificados apenas os RNF's primários, mas como o processo é incremental, os grafos de RNF's são refinados nos demais subfluxos.

Cysneiros, em [Cys01-A], [Cys01-B], [Cys01-C], propõe algumas heurísticas, que podem nos ajudar, para sistematizar a análise dos grafos de RNF. São elas:

1. Comparar todos os grafos de um mesmo tipo, procurando pelas possíveis interdependências.
2. Comparar todos os grafos de tipos possivelmente conflitantes. Por exemplo, grafos que envolvam Segurança e Performance, mesmo quando aplicados a diferentes tópicos.
3. Comparar os grafos por pares de forma a comparar um dado grafo com todos os outros, sem contudo, repetir comparações que já tenham sido feitas anteriormente.

Um ponto importante a ser considerado nesta dissertação é que, como estamos focando na Engenharia de Requisitos para aplicações Web, podemos concentrar nossos esforços em RNF's

primários que representam propriedades básicas para este tipo de aplicação: Segurança, Performance, Distribuição, Usabilidade e Consistência.

Um exemplo de grafo de RNF para uma aplicação voltada a cartões de crédito é apresentado na Figura 4.5, seguindo a notação descrita no Capítulo 2, Seção 2.4.2.1. Podemos observar que, inicialmente são identificados os RNF's primários da aplicação (*Performance*, *Segurança* e *Usabilidade*). Estes RNF's, também chamados de “softgoals”, por Chung, em [Chu00], são então refinados e relacionados. No caso de *Segurança*, por exemplo, o refinamento levou à *Confidencialidade*, *Integridade* e *Disponibilidade*. O passo seguinte é identificar os RNF's prioritários, neste caso a *Confidencialidade* foi selecionada. A partir de então, passam a ser descobertas as possíveis operacionalizações, por exemplo: *Efetuar Login*. Durante o processo de construção deste grafo, são identificadas as interdependências (positivas e negativas) entre os softgoals, podemos observar que *Registrar Logs* garante a *recuperabilidade*, entretanto interfere negativamente na *performance* e a operacionalização *Validar Acesso do Usuário* aumenta a *precisão*, mas interfere negativamente na *usabilidade* e no *número de acessos*.

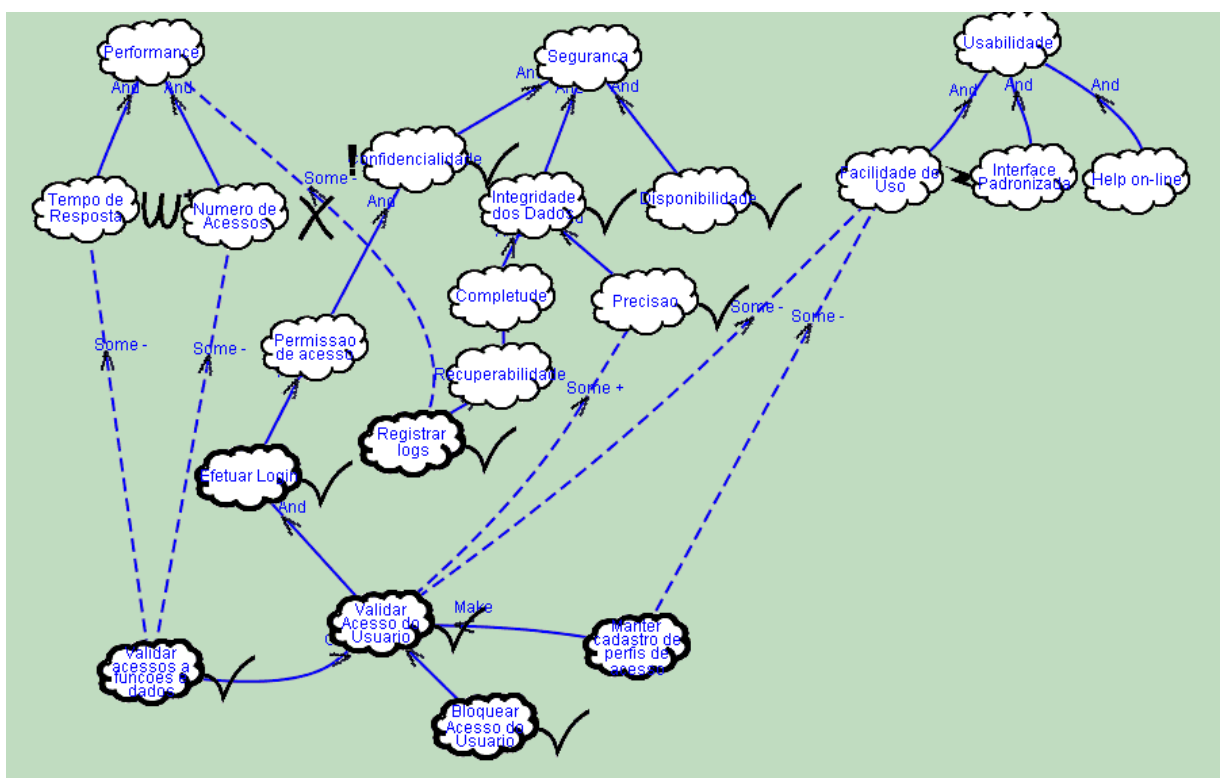


Figura 4.5 - Grafo de RNF's com operacionalizações

Descrevemos a atividade *Encontrar Requisitos Não-Funcionais*, na próxima subseção detalharemos a atividade *Gerenciar Dependências*, ainda no Subfluxo *Elicitar Requisitos*.



#### 4.2.2.6. Atividade: Gerenciar Dependências

Esta atividade tem o objetivo de ajudar a gerenciar o escopo do projeto e as alterações dos requisitos, através do rastreamento dos mesmos. Como o processo é iterativo e incremental, e mudanças nos requisitos sempre ocorrem, esta atividade é sempre realizada, estando presente em vários subfluxos do Fluxo de Requisitos. Neste subfluxo será apresentada uma visão geral desta atividade, entretanto mais detalhes serão apresentados no subfluxo *Gerenciar Mudanças de Requisitos*, seção 4.2.8. As alterações necessárias nos requisitos são registradas através do formulário de *Solicitação de Mudanças*. Na realidade, este artefato pertence ao processo de Gerência de Configuração e Mudança, e é usado para solicitar qualquer tipo de alteração sobre o projeto, não apenas requisitos.

Seguindo o *Plano de Gerência de Requisitos* e a técnica de gerência de requisitos escolhida, as matrizes de rastreabilidade começam a ser desenvolvidas com base nos requisitos identificados e artefatos produzidos. Alguns exemplos de rastreamento que podem ser identificados, com base nas atividades anteriores são: entre requisitos, entre requisitos e casos de uso, entre os requisitos e demais artefatos (por exemplo: Atas de Reuniões, Documentos de Testes, Código de Implementação, etc) .

No *Plano de Gerência de Requisitos* é definido como os requisitos são rastreados com outros artefatos. O Engenheiro de Requisitos deve, então, estabelecer a rastreabilidade desejada e, periodicamente, consultar relatórios para averiguar se a rastreabilidade está sendo mantida conforme a especificado no plano.

De acordo com o RUP, o documento de Atributos de Requisitos, que corresponde a um repositório com os requisitos, seus atributos e as matrizes de rastreabilidade entre os requisitos, deve estar sempre atualizado. Este artefato é composto de matrizes de atributos de requisitos (para cada tipo de requisito, são definidos, em um eixo, os atributos e no outro os requisitos), matrizes de rastreabilidade entre requisitos e árvore de rastreabilidade (permite uma visão gráfica do rastreamento).

Em [Spe03], Spence diz que embora existam diversas estratégias de rastreabilidade que podem ser usadas para facilitar o processo de gerenciamento de requisitos, é possível utilizar qualquer uma delas ao trabalhar com um framework como o RUP. Ou seja, a equipe de desenvolvimento pode decidir que técnica de rastreamento é mais adequada para o projeto. Neste trabalho, sugerimos a utilização da proposta de Toranzo, citada em [Tor02-A] e [Tor02-B], que

tem o objetivo de melhorar o rastreamento de requisitos através de um conjunto de diretrizes a serem seguidas. Com a utilização desta técnica, é possível rastrear qualquer tipo de artefato gerado no desenvolvimento da aplicação. No Capítulo 2, seção 2.5.5, são apresentados detalhes da proposta de Toranzo.

Observando algumas técnicas de rastreamento de requisitos, verificamos que, na maioria delas são geradas matrizes de rastreabilidade, por este motivo, sugerimos a substituição do artefato Atributos de Requisitos, do RUP, pela matriz referenciada. Dependendo da técnica escolhida e do que se deseja rastrear, o reflexo da alteração deste artefato pode ser apenas sintático, ou seja, na sua nomenclatura, ou semântico, caso o rastreamento não envolva apenas os requisitos, mas também os demais artefatos que são gerados ao longo do desenvolvimento, como, por exemplo, é sugerido por Toranzo.

Descrevemos as atividades executadas no Subfluxo *Elicitar Requisitos*, na próxima seção apresentaremos o Subfluxo *Analisar e Documentar Requisitos*.

#### **4.2.3. SUBFLUXO: ANALISAR E DOCUMENTAR REQUISITOS**

A Figura 4.6 apresenta o detalhamento do subfluxo *Analisar e Documentar Requisitos*. Comparando-o com o Subfluxo Definir o Sistema, no RUP 2002, verificamos que no WRE-Process, ao invés de ser executada a atividade *Desenvolver Visão do Sistema*, é realizada a atividade *Documentar Requisitos* devido à sua relevância no processo genérico de gerência de requisitos (conforme apresentado no Capítulo 2), e por ser responsável por gerar o *Documento de Requisitos* e a *Especificação de Casos de Uso*. Além disto, novas atividades foram incluídas neste subfluxo: *Encontrar RNF's* e *Integrar Visão Funcional e Não-Funcional*, que são relevantes para aplicações Web.

A documentação de requisitos de software é a declaração oficial do que é exigido dos desenvolvedores do sistema. Ela deve incluir os requisitos de usuário e a especificação detalhada dos requisitos de sistema [Som03]. Em alguns casos, os requisitos de usuário e de sistema são incluídos em uma única descrição (O RUP, por exemplo, as considera internas ao modelo de Casos de Uso). Em outros casos, os requisitos de usuários são definidos em um documento que serve de introdução à especificação dos requisitos de sistema. No WRE-Process, estas descrições são tratadas em dois documentos separados: Documento de Requisitos e Documento de Especificação de Casos de Uso.

A atividade *Integrar Visão Funcional e Não-Funcional* foi adicionada ao **Subfluxo Analisar e Documentar Requisitos** com o objetivo de incorporar as necessidades da visão não-funcional à visão funcional da aplicação. No RUP há uma lacuna na integração entre as visões funcional e não-funcional, o que pode resultar na falta de tratamento de requisitos não-funcionais, gerando insatisfação do cliente.

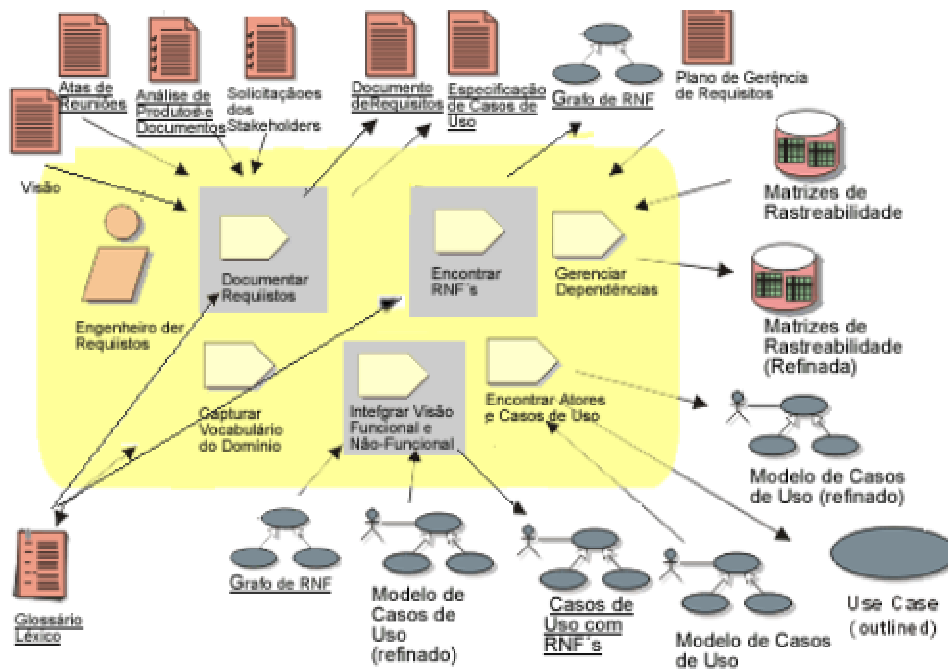


Figura 4.6 - Subfluxo Analisar e Documentar Requisitos

Devido à característica iterativa e incremental do processo, as atividades *Capturar o Vocabulário do Domínio*, *Encontrar Atores e Casos de Uso*, *Encontrar RNF's* e *Gerenciar Dependências*, que foram originadas no Subfluxo *Elicitar Requisitos*, seção 4.2.2, são re-executadas nesse subfluxo com o objetivo de refinar os artefatos produzidos, caso haja necessidade. Por esta razão, nas subseções a seguir apresentaremos apenas as seguintes atividades deste subfluxo e os artefatos produzidos nas mesmas: *Encontrar Atores e Casos de Uso (Refinamento)*, *Documentar Requisitos* e *Integrar Visão-Funcional e Não-Funcional*.

#### 4.2.3.1. Atividade: Encontrar Atores e Casos de Uso

No Subfluxo *Elicitar Requisitos*, seção 4.2.2.4, ao ser executada a atividade *Encontrar Atores e Casos de Uso*, são identificados os atores e casos de uso, baseados nas informações captadas. Com o desenvolvimento iterativo e incremental, o modelo de Casos de Uso pode ser refinado com novas informações, que não tenham sido identificadas no levantamento inicial de

requisitos. Desta forma, caso o levantamento inicial tenha sido preciso o suficiente, a re-execução desta atividade torna-se desnecessária.

A Figura 4.7 apresenta o refinamento do Modelo de Casos de Uso para um site de operadora de cartão de crédito. Comparando o modelo refinado com o original, vemos que novos atores foram encontrados, como o *Sistema Integrado de Cartão de Crédito* e o *Captador de Clientes*. O *Sistema Integrado de Cartão de Crédito* é responsável por fornecer, processar e armazenar as informações solicitadas pelo cliente no site da operadora. O *Captador de Clientes* é a pessoa responsável por analisar, aprovar e rejeitar as propostas submetidas pelos  *futuros clientes*.

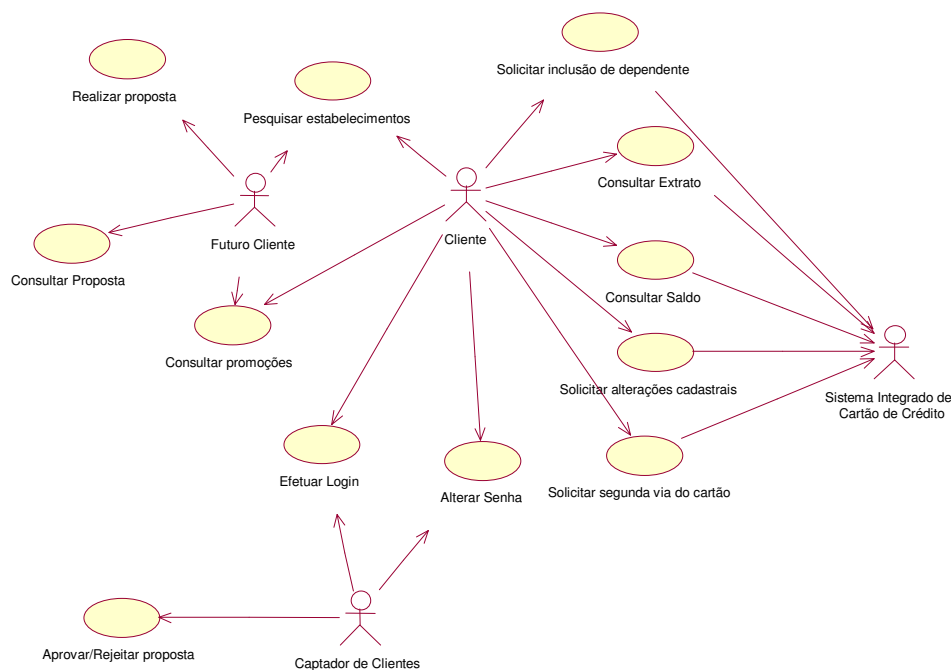


Figura 4.7- Diagrama de Casos de Uso

Descrevemos a atividade *Encontrar Atores e Casos de Uso*, na próxima subseção detalharemos a atividade *Documentar Requisitos*, ainda no Subfluxo *Analisar e Documentar Requisitos*.

#### **4.2.3.2. Atividade: Documentar Requisitos**

Com base no *Diagrama de Casos de Uso*, *Glossário Léxico*, *Atas de Reuniões e Avaliação de Produtos e Documentos Existentes* tem-se como desenvolver a visão funcional da aplicação, documentando os requisitos encontrados.

O documento de requisitos tem um conjunto diversificado de usuários, abrangendo desde a alta gerência da organização, que está pagando pelo sistema, até os usuários finais e engenheiros de software. Sendo assim, diferentes níveis de especificação de sistema são úteis porque comunicam informações sobre o sistema para diversos tipos de leitores [Som03]. Ao serem documentados requisitos no WRE-Process, são gerados dois artefatos distintos:

- *Documento de Requisitos*: são declarações sobre as funções que o sistema deve fornecer e as restrições sobre as quais deve operar, sem aprofundar detalhes. O público alvo deste artefato é composto pelos gerentes do cliente, gerentes do fornecedor, usuários finais do sistema, engenheiros e arquitetos de software, e engenheiros de testes.
- *Especificação de Casos de Uso*: descreve, detalhadamente, as funções do sistema, ou seja, os casos de uso da aplicação, através de seqüências que apresentam a interação dos atores com o sistema. Este documento é, a princípio, direcionado a usuários finais do sistema, engenheiros e arquitetos de software e engenheiros de testes.

Estes documentos são escritos em linguagem natural e representam o “contrato” entre os clientes e desenvolvedores. O *Documento de Requisitos* apresenta uma visão geral dos requisitos identificados na aplicação e, cada requisito, pode ser mapeado em um ou mais casos de uso. A *Especificação de Casos de Uso* fornece uma visão detalhada dos casos de uso da aplicação,

Nas subseções a seguir apresentamos detalhes do *Documento de Requisitos* e da *Especificação de Casos de Uso*.

##### **4.2.3.2.1. Artefato: Documento de Requisitos**

Os requisitos devem possuir um nível de detalhamento suficiente para permitir que os *stakeholders*, desenvolvedores e testadores possam entender a aplicação. Os requisitos devem ter identificações únicas, para garantir a rastreabilidade. Por experiência prática, para sistemas com muitos requisitos, achamos interessante agrupar os requisitos similares em “pacotes”.

Observando, no Capítulo 2 (seção 2.3), o padrão IEEE [Som98], [Som03], o template sugerido no RUP [Rup02], as considerações de Sommerville [Som98], [Som03] e a proposta de Amador Duran [Dur00], propusemos um modelo para o *Documento de Requisitos* do WRE-Process, conforme apresentado na Tabela 4.11. Observe que no template há seções para *Requisitos Funcionais*, *Requisitos Não-Funcionais* e *Requisitos Organizacionais*, cujas definições foram vistas nas seções 2.4.1, 2.4.2 e 2.4.3 respectivamente.

Para cada requisito deve ser indicada a sua prioridade (*essencial, importante ou desejável*), conforme Vasconcelos et al apresenta em [Vas01], pois isto ajuda ao gerente do projeto planejar cada iteração:

- *Essencial* é o requisito sem o qual o sistema não entra em funcionamento. Requisitos essenciais são requisitos imprescindíveis, que têm que ser implementados impreterivelmente.
- *Importante* é o requisito sem o qual o sistema entra em funcionamento, mas de forma não satisfatória. Requisitos importantes devem ser implementados, mas, se não forem, o sistema poderá ser implantado e usado mesmo assim.
- *Desejável* é o requisito que não compromete as funcionalidades básicas do sistema, isto é, o sistema pode funcionar de forma satisfatória sem ele. Requisitos desejáveis são requisitos que podem ser deixados para versões posteriores do sistema, caso não haja tempo hábil para implementá-los na versão que está sendo especificada.

Na seção reservada para os requisitos não-funcionais da aplicação, são apresentados aqueles requisitos gerais, que devem ser atendidos pela aplicação, seguindo a classificação de Sommerville [Som98]: Requisitos de Processo, Requisitos Externos e Requisitos do Produto.

Um exemplo de documento de requisitos pode ser visto no Apêndice A.

Tabela 4.11- Template do Documento de Requisitos

<p><b>1.Introdução</b> <i>A introdução apresenta uma visão geral do documento, incluindo seu propósito, o escopo do sistema e as definições/termos e abreviações que são usadas no documento.</i></p> <p><b>2.Descrição dos Usuários</b> <i>Nessa seção são apresentados os tipos de usuários da aplicação. Sugerimos esta descrição para que fique claro o entendimento dos engenheiros de requisitos, projetistas, testadores e clientes em relação aos usuários do sistema.</i></p> <p><b>3.Visão Geral do Produto</b> <i>Nessa seção é apresentada uma visão geral da aplicação, com suas características gerais. Aqui podem ser descritos os objetivos de cada módulo (ou pacote) da aplicação.Essa seção não descreve e requisitos, mas provê uma base para seu entendimento.</i></p> <p><b>4. Restrições</b> <i>Nessa seção são descritas as principais restrições sobre as quais o sistema deverá ser desenvolvido e executado. Também deve ser especificado o escopo negativo da aplicação, ou seja, o que o sistema não contemplará.</i></p> <p><b>5. Requisitos Organizacionais</b> <i>Nessa seção são descritos os Requisitos de Negócio da Aplicação, com identificação única.</i></p> <p>    <b>5.1 [RN001]</b> <i>O requisito é descrito e indicada sua prioridade: essencial, importante, desejável</i></p> <p>    <b>5.2 [RN002]</b></p> <p>    <b>5.3 [RN00...]</b></p> <p><b>6. Requisitos Funcionais</b> <i>Nessa seção são descritos os Requisitos Funcionais da Aplicação, com identificação única. Os requisitos podem ser agrupados em pacotes (ou módulos). Para cada requisito deve ser identificada sua prioridade: essencial, importante ou desejável</i></p> <p>    <b>6.1 [RF001] Pacote 1</b></p> <p>        <b>6.1.1 [RF001.001] Requisito 1.1</b></p> <p>        <b>6.1.2 [RF001.002] Requisito 1.2</b></p> <p>    <b>6.2 [RF002] Pacote 2</b></p> <p><b>7. Requisitos Não-Funcionais</b> <i>Nessa seção são descritos os Requisitos de Não-Funcionais gerais da aplicação, de modo ainda informal.</i></p> <p>    <b>7.1 Requisitos de Processo</b>         <b>[NF001] Sistema para ambiente Web</b></p> <p>    <b>7.2 Requisitos Externos</b>         <b>[NF002] Baixo custo</b></p> <p>    <b>7.3 Requisitos de Produto</b>         <b>[NF003] Segurança</b>         <b>[NF004] Performance</b></p> <p><b>8. Referências</b> <i>Essa seção apresenta as referências usadas no documento e que podem ser consultadas.</i></p>
--

#### **4.2.3.2.2. Artefato: Documento de Especificação de Casos de Uso**

Analisando o *Glossário Léxico*, as *Atas de Reuniões*, o *Documento de Visão*, a *Análise de Produtos e Documentos Existentes* e o *Documento de Requisitos*, temos uma base de informações para especificar os casos de uso da aplicação. Para iniciar a especificação dos casos de uso, devemos ter em mente o objetivo de cada caso de uso identificado, as pré e pós-condições para sua execução e o contexto onde o mesmo será inserido.

No Capítulo 2, seções 2.4.1.2 e 2.4.1.3 foram apresentados conceitos e modelos de cenários e casos de uso. É interessante observar que, na proposta de cenários, de Leite [Cys01-A], [Lei00], o contexto onde o caso de uso está inserido é considerado. Nós acreditamos que este item é bastante útil, pois auxilia na identificação do escopo que o caso de uso deve abranger. Outro ponto citado nos cenários e que não são tratados nos demais templates são os recursos necessários (seção Contexto), ou seja, caso existam elementos físicos ou informações relevantes, que precisem estar disponíveis para que o caso de uso possa ser executado, estes devem ser descritos. O template proposto neste trabalho, é descrito na Tabela 4.12 e incorpora elementos das propostas de Leite [Cys01-A], [Lei00], Duran [Dur00], Sommerville [Som98], [Som03] e RUP [Rup02] (Todas descritas no capítulo 2, seção 2.4.1.3).

Vale salientar que as especificações de casos de uso podem ser representadas através de um documento para cada caso de uso, ou um documento com um conjunto de casos de uso pertencentes a um pacote de casos de uso, ou ainda um documento único contendo todos os casos de uso da aplicação (para sistemas de pequeno porte).

O Apêndice B apresenta um documento de Especificação de Casos de Uso.



Tabela 4.12- Template para a Especificação dos Casos de Uso

<p><b>1. Identificação do Caso de Uso</b> <i>A identificação de cada caso de uso deve ser única para permitir a rastreabilidade e referências entre os documentos. O nome do caso de uso deve indicar o objetivo do mesmo.</i></p> <p><b>2. Breve Descrição</b> <i>Nessa seção é apresentada uma breve descrição do objetivo do caso de uso.</i></p> <p><b>3. Ator</b> <i>Nessa seção é apresentado o ator responsável pela execução do caso de uso.</i></p> <p><b>4. Contexto</b> <i>Nessa seção é descrito o contexto sobre o qual o caso de uso deve ser executado. Este contexto pode ser uma determinada localização física ou uma localização temporal, por exemplo. Além disso, devem ser descritos no contexto os elementos físicos ou informações que precisem estar disponíveis (recursos necessários).</i></p> <p><b>5. Requisitos Relacionados</b> <i>Essa seção apresenta os requisitos atendidos pelo caso de uso.</i></p> <p><b>6. Pré-Condições</b> <i>A pré-condição corresponde ao estado que o sistema deve estar antes do caso de uso ser executado.</i></p> <p><b>7. Pós-Condições</b> <i>As pós-condições correspondem ao estado que o sistema passará a ter imediatamente após a execução do caso de uso.</i></p> <p><b>8. Fluxo de Eventos Principal</b> <i>O Fluxo de Eventos Principal corresponde ao conjunto de passos que formam o fluxo básico do caso de uso, ou seja aquele que é executado, sem desvios, caso nenhuma situação de exceção ou alternativa ocorra.</i></p> <p><b>9. Fluxos de Eventos Alternativos e de Exceção</b> <i>O Fluxo de Eventos Alternativos e de Exceção corresponde aos caminhos alternativos para onde cada passo do Fluxo de Eventos Principal pode ser desviado.</i></p> <p><b>10. Requisitos não-funcionais</b> <i>Nessa seção são descritos os Requisitos de Não-Funcionais que devem ser tratados no Caso de Uso.</i></p> <p><b>11. Observações</b> <i>Nessa seção são apresentados comentários e observações extras que precisem ser descritos, como por exemplo: frequência esperada de execução por unidade de tempo, prioridade do caso de uso e outros comentários.</i></p> <p><b>12. Referências</b> <i>Essa seção apresenta as referências usadas no documento e que podem ser consultadas.</i></p>
---

Descrevemos a atividade *Documentar Requisitos*, na próxima subseção detalharemos a atividade *Integrar Visão Funcional e Não-Funcional*, ainda no Subfluxo *Analisar e Documentar Requisitos*.

#### 4.2.3.3. Atividade: Integrar Visão Funcional e Não-Funcional

Uma vez que as visões funcionais e não-funcionais existem, é necessário encontrar os pontos de convergência onde a integração entre ambas seja realizada. Ou seja, incorporar na visão funcional as necessidades da visão não-funcional. Cysneiros propõe, em [Cys01-B] e [Cys01-C], um processo de integração que pode ser usado nas fases iniciais do projeto, onde os RNF's são integrados ao modelo de casos de uso (ou de cenários).

Desta forma, integrando os Requisitos Não-Funcionais (RNF's) ao modelo e especificação de casos de uso, todos os modelos conceituais derivados destes artefatos também refletirão os RNF's.

Um ponto importante a ser considerado nesta proposta é que podemos concentrar nossos esforços, inicialmente, em alguns RNF's primários que representam propriedades básicas para o tipo da aplicação. Em sistemas para web, por exemplo, podemos considerar como RNF's primários: Segurança, Performance, Distribuição, Usabilidade e Consistência.

A estratégia de integração é a seguinte: para cada Diagrama de Casos de Uso, deve-se verificar quais símbolos do *Glossário Léxico* aparecem no modelo e em seus casos de uso. Para cada símbolo encontrado, deve-se procurar o conjunto de Requisitos Não-Funcionais (RNF's) associados a eles e analisar os *Grafos de RNF's*, produzidos na atividade *Encontrar Requisitos Não-Funcionais*, seção 4.2.2.5. Uma vez encontrados os grafos, precisa-se verificar se no diagrama de casos de uso estão sendo refletidas as operacionalizações realizadas nos grafos de RNF's.

Considerando as alterações supracitadas no fluxo de requisitos, ou seja, criação do Glossário Léxico, desenvolvimento dos grafos de RNF's e sua integração com os casos de uso, claramente notamos uma sobrecarga no trabalho dos Engenheiros de Requisitos, entretanto o resultado deste esforço produzirá um conjunto de requisitos mais completo, diminuindo as manutenções causadas por negligenciar os RNF's e os custos nas correções de erros.

Ao analisarmos o Grafo de RNF's da Figura 4.5, do exemplo site de uma operadora de cartão de crédito, verificamos que, para atender o requisito de *Confidencialidade e Permissão de Acesso*, é necessário *Efetuar Login* (que também é operacionalizado em *Validar acesso do usuário*, *Validar acesso a funções e dados*, *Bloquear acesso do usuário* e *Manter cadastro de perfil*) e para atender o RNF *Integridade dos dados*, *Completeness e Recuperabilidade* é necessário *Registrar Logs*. Comparando este grafo com o Diagrama de Casos de Uso, apresentado na Figura 4.7, vemos que a operacionalização *Efetuar Login* já é satisfeita através de um caso de uso definido, enquanto que *Registrar Logs* ainda não estava refletida neste diagrama. A Figura 4.8, apresenta o artefato produzido nesta atividade: o Modelo de Casos de Uso refletindo algumas das operacionalizações realizadas no grafo de RNF's. Observando este modelo podemos identificar os novos Casos de Uso: *Registrar Logs*, *Validar Acesso do Usuário*, *Validar Acesso Funções e Dados* e *Bloquear Acesso do Usuário*. Para os casos de uso que não surgiram de nenhuma operacionalização, mas têm RNF's associados, devem ser criadas notas

explicativas para que, durante as fases de análise, projeto e implementação os desenvolvedores tratem estes RNF's.

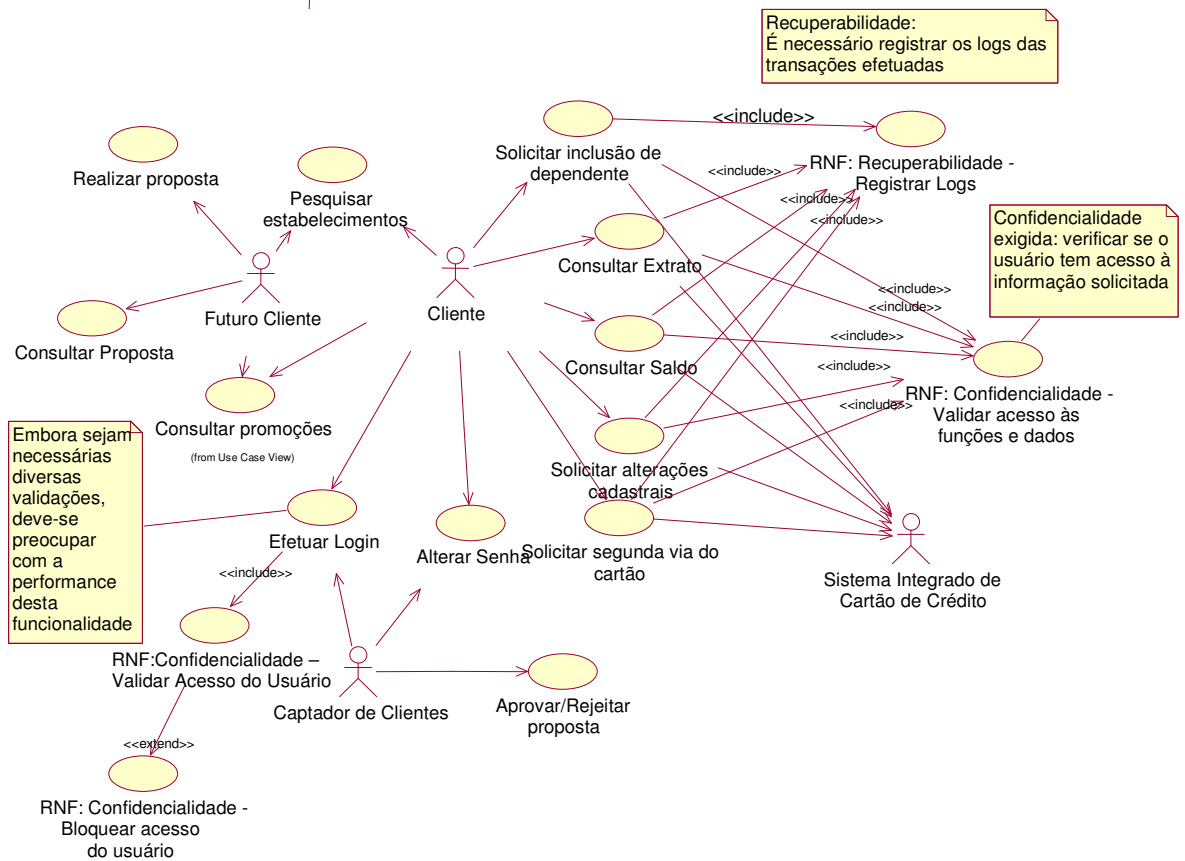


Figura 4.8- Modelo de Casos de Uso Refletindo RNF's

No documento de Especificação de Casos de Uso, os novos casos de uso que foram descobertos através desta atividade devem ser especificados. Ao citar o objetivo do caso de uso, é importante salientar qual RNF é satisfeito com a execução do mesmo.

Descrevemos as atividades executadas no Subfluxo *Analisar e Documentar Requisitos*, com exceção de *Capturar o Vocabulário do Domínio*, *Encontrar RNF's* e *Gerenciar Dependências* pelo fato delas já terem sido detalhadas no subfluxo *Elicitar Requisitos*. Na próxima seção apresentaremos o Subfluxo *Definir a Interface*.

#### 4.2.4. SUBFLUXO: DEFINIR A INTERFACE

A interface com o usuário é um componente peculiar da arquitetura de uma aplicação. Jones, em [Jon03], afirma que esta é a única parte do software totalmente subjetiva, pois o fato de

uma tela ser “usável” ou não depende da opinião dos seus usuários. Esta natureza subjetiva da interface vem causando uma mudança na responsabilidade do projeto da mesma.

O principal objetivo a ser alcançado ao desenvolver um sistema de informações, principalmente as aplicações Web, é atender às expectativas dos usuários finais e permitir que eles executem suas atividades de forma eficiente e eficaz. Para alcançar este objetivo é necessário incorporar novas técnicas que auxiliem a capturar as necessidades dos usuários, considerando critérios de usabilidade. Somente desta forma, será possível desenvolver interfaces intuitivas e fáceis de usar, que permitam aos usuários extraírem o máximo de benefícios das aplicações. Dado à importância que as interfaces com usuários detêm sobre aplicações web, foi adicionado, ao WRE-Process, o subfluxo *Definir Interface*.

Alguns leitores podem indagar “Por que tratar os requisitos de interface aqui se eles são requisitos não-funcionais?” É verdade que quando tratamos os RNF’s e realizamos suas operacionalizações, encontramos o RNF *Usabilidade* (conforme apresentado no Capítulo 2, junto com alguns aspectos para a definição de interface com usuário). Os RNF’s encontrados com a operacionalização da *Usabilidade* ajudam a projetar o sistema e a interface. Mas também é preciso definir um modelo que descreva as características de interface dos diferentes tipos de usuários da aplicação, com o conjunto de tarefas que devem ser executadas. Assim, o subfluxo proposto deve definir a estrutura do *layout*, os cenários de utilização da interface e um protótipo da mesma, com base nos requisitos funcionais, não-funcionais e de interface elicitados.

A Análise de Requisitos de Interface deve destacar as estruturas navegacionais necessitadas pelos diferentes usuários da aplicação e, por este motivo, sugerimos a produção de um Modelo Navegacional<sup>8</sup>, baseado nos aspectos expostos no Capítulo 2, Seção 2.6.2.2. A Figura 4.9 apresenta o subfluxo *Definir Interface*.

Como já foi dito na descrição da atividade *Descobrir Requisitos de Interface*, Seção 4.2.2.2 vimos que o RUP é limitado no que diz respeito à captura e especificação de requisitos de interface, principalmente para aplicações Web. O RUP 2002 trata a definição de interface no Subfluxo *Refinar a Definição do Sistema* (Vide Seção 3.3.5), através das atividades *Modelar a Interface do Usuário* e *Prototipar a Interface*. É introduzido um modelo de casos de uso

---

<sup>8</sup> representado pelos artefatos: *Modelo de Casos de Uso de Interface*, *Documento de Requisitos de Interface*, *Especificação de Casos de Uso de Interface*, *Protótipo da Interface*, *Guia de desenvolvimento da Interface e Da descrição do Lay-out*.

adicional chamado “*casos de uso storyboards*”, que descreve os aspectos referentes à interface com o usuário. Para cada caso de uso funcional, existe um *caso de uso storyboard*.

No subfluxo *Definir a Interface* do WRE-Process, são executadas quatro novas atividades: *Especificar Requisitos de Interface*, *Projetar Visibilidade da Aplicação*, *Definir Protótipo de Interface* e *Projetar Conteúdo*. Detalharemos estas atividades e os artefatos produzidos nas próximas subseções.

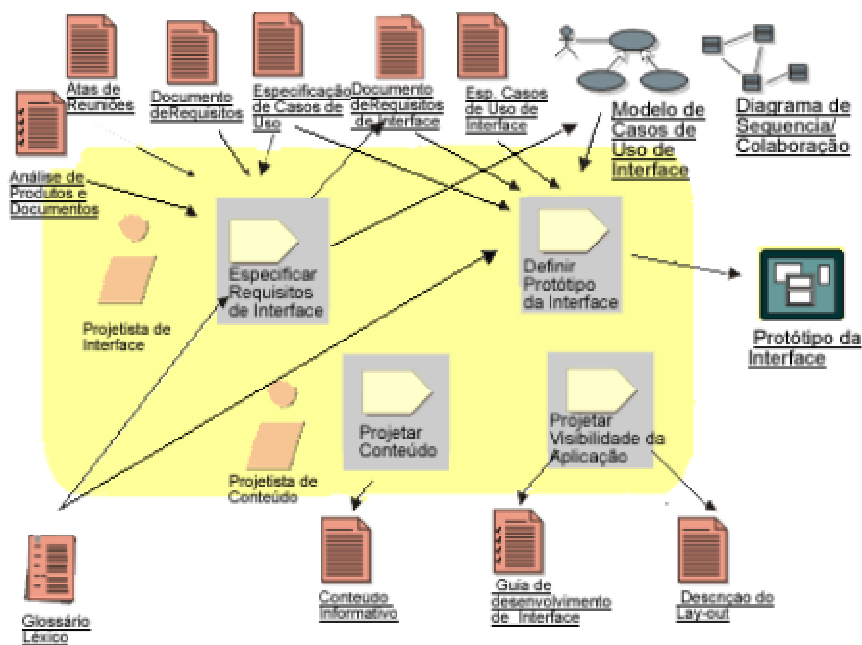


Figura 4.9 - Subfluxo *Definir a Interface*

#### 4.2.4.1. Atividade: Especificar Requisitos de Interface

O objetivo desta atividade é analisar e especificar os requisitos da interface do sistema que foram elicitados. Tendo como base os artefatos *Atas de Reuniões*, *Análise de Produtos e Documentos*, *Documento de Requisitos*, *Especificação de Casos de Uso* e *Glossário Léxico*, o projetista de interface (*designer*) tem informações para especificar estes requisitos e, a partir daí, começar a projetar a interface. Vale salientar que, ao serem analisados os produtos e documentos existentes, também nos referimos às aplicações similares ou concorrentes. Desta forma, para um projeto de um web site, por exemplo, é interessante que seja produzido uma análise comparativa entre serviços, *layout* e tecnologia nos sites correlatos/concorrentes, fornecendo bases para novas implementações e recursos.

Segundo Nielsen [Nie95], a melhor forma de obter os critérios de usabilidade de uma aplicação é gastar o máximo de esforço possível antes de projetar a *layout* da interface. Com base neste conceito e na Análise de Tarefas (vide Capítulo 2, seção 2.6.2.2), vemos que não é apenas necessário conhecer os usuários e suas habilidades, mas também conhecer as tarefas que eles precisarão executar. Os documentos de Requisitos e Especificação de Casos de Uso são fontes de informações pois apresentam todas as funcionalidades a serem executadas pela aplicação.

Na Análise de Requisitos de Interface são destacadas as estruturas navegacionais necessitadas pelos diferentes tipos de usuários da aplicação. Da mesma forma como é realizado no Framework W2000 [Bar01], o WRE-Proces produz um diagrama com notação UML, chamado *Modelo de Casos de Uso de Interface*. Baseado no *Modelo de Casos de Uso Funcionais* são definidas as capacidades de navegação associadas a cada ator. Os usuários genéricos de um website, por exemplo, são tratados neste artefato.

Como já foi citado anteriormente, o RUP propõe o Modelo de Casos de Uso Storyboard para especificação da interface. Utilizamos este modelo como base para produzir o *Documento os Casos de Uso de Interface*, já que ele permite, na fase de projeto, modelar a interface através de objetos de fronteira (UML) que dão suporte à interação associada ao caso de uso, como diagramas de seqüência (ou colaboração) e diagramas de classes.

Caso o engenheiro de requisitos e o gerente de projeto considerem necessário especificar o aspecto comportamental da interface da aplicação, podem ser produzidos Diagramas de Seqüência e/ou Colaboração para os casos de uso de interface identificados. Com a produção deste artefato, junto com os diagramas de casos de uso, é possível utilizar a técnica proposta por Sánchez, em [San02], para a geração automática do protótipo de interface gráfica. Embora não tenhamos adotado esta técnica, deixamos claro que o processo é flexível, o suficiente, para permitir a geração de interfaces gráficas, automaticamente, através da aplicação de atividades do VRU<sup>9</sup> [San02].

Ao executar a atividade *Especificar Requisitos de Interface*, são produzidos os seguintes artefatos: *Modelo de Casos de Uso de Interface*, *Documento de Requisitos de Interface*, *Especificação de Casos de Uso de Interface* e, opcionalmente, o *Diagrama de Seqüência e/ou Colaboração* para os casos de uso de Interface.

---

<sup>9</sup> Validación de Requisitos de Usuário.

O *Modelo de Casos de Uso de Interface* é um artefato gerado pelo Projetista de Interface que identifica os caminhos navegacionais que podem ser percorridos pelos usuários. A estes “caminhos navegacionais” chamamos de Casos de Uso de Interface. A diferença entre estes e os Casos de Uso Funcionais é que nos Casos de Uso de Interface são representados os possíveis caminhos a serem navegados pelos papéis de usuários identificados, não as funcionalidades por eles executadas. Usuários genéricos, por exemplo, que não executam nenhuma ação no sistema, mas que navegam nas páginas da aplicação, são representados no Modelo de Casos de Uso de Interface, mas não aparecem nos Casos de Uso Funcionais (A subseção a seguir apresentará um exemplo de Modelo de Casos de Uso de Interface). É evidente que os casos de uso funcionais podem vir a ser casos de uso de interface também, mas o objetivo principal deste modelo é exatamente descobrir os casos de uso de interface e descrever aqueles que não são representados por casos de uso funcionais.

O Documento de Requisitos de Interface também é desenvolvido pelo projetista de interface e, diferentemente do Documento de Requisitos Funcionais, abrange princípios e objetivos, relativos à apresentação do sistema, a serem alcançados de forma padronizada, para toda a aplicação. Por exemplo: padrões de navegabilidade, usabilidade, *design* gráfico, etc.

Para evitar sobrecarga de trabalho e redundância de informações, a Especificação de Casos de Uso de Interface deve considerar aqueles casos de uso de interface que não são representados por casos de uso funcionais; ou que têm interface crítica para a aplicação; ou aqueles que devem servir de modelo para as demais interfaces do mesmo tipo. Aqueles casos de uso de interface que têm um caso de funcional correspondente, não precisam ser especificados se no caso de uso funcional estiver descrito o comportamento da interface com o usuário. Vale salientar que este artefato tem como objetivo ajudar o projetista de interface a desenvolver as telas do protótipo da interface com o usuário, assim uma vez a interface esteja definida e desenhada, não há motivo para manter este documento. Algumas características que podem ser definidas na especificação de casos de uso de interface, por exemplo, são: identificar onde devem ser exibidos balões de ajuda ao usuário, a média de registros ou volume de informações de um determinado objeto, a média de uso de determinadas ações, etc.

Nas seções seguintes, detalharemos os artefatos *Modelo de Casos de Uso de Interface*, *Documento de Requisitos de Interface* e *Especificação de Casos de Uso de Interface*.

#### **4.2.4.1.1. Artefato: Modelo de Casos de Uso de Interface**

A Figura 4.10 apresenta um exemplo de Modelo de Casos de Uso de Interface para o site da operadora de Cartão de Crédito. Este diagrama define a navegabilidade de cada ator identificado. Analisando este modelo e comparando-o com o Modelo de Casos de Uso apresentado na Figura 4.8, vemos que um ator genérico *Usuário Web* foi adicionado pois, para o efeito de navegabilidade, há necessidade de trabalharmos com um usuário genérico que acessará a aplicação. Este usuário representa uma generalização dos atores *Cliente* e *Futuro Cliente*. O *Usuário Web* navegará pela página principal e poderá realizar os casos de uso de interface *Navegar Promoções*, *Navegar Notícias*, *Pesquisar Estabelecimentos* e *Consultar Promoções*. deve-se notar que nenhum destes correspondem a casos de uso funcionais, com exceção de *Pesquisar Estabelecimentos*. Dependendo do papel desempenhado pelo usuário, ele poderá acessar as funcionalidades específicas de um cliente, desde que efetue o login. Caso ele não seja cliente poderá *Realizar Proposta* e *Consultar Proposta*.

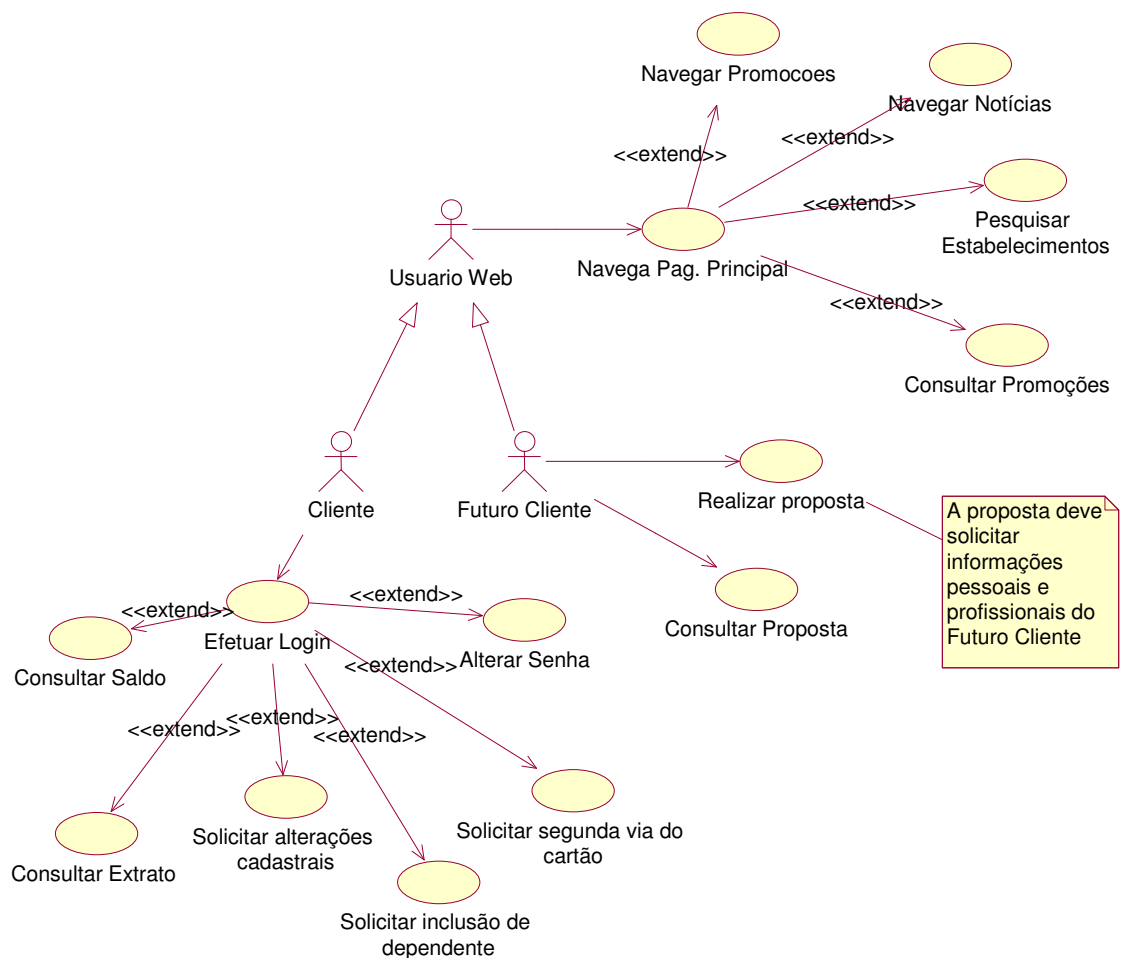




Figura 4.10 - Modelo de Casos de Uso de Interface

#### 4.2.4.1.2. Artefato: Documento de Requisitos de Interface

Evoluindo com a documentação, é interessante que seja formulado o Documento de Requisitos de Interface, abrangendo questões como navegabilidade, *design* gráfico, usabilidade e necessidades tecnológicas de interface. Este documento é desenvolvido pelo Projetista de Interface. Devem ser identificados os objetivos da interface, o público alvo da aplicação e descritos os requisitos de interface elicitados. A Tabela 4.13 apresenta um template para a criação deste documento e o Apêndice C apresenta um Documento de Requisitos de Interface.

Tabela 4.13 - Template do Documento Requisitos de Interface

<p><b>1. Introdução</b> <i>A introdução apresenta uma visão geral do documento de Requisitos de Interface, incluindo seu propósito, o escopo do sistema, as definições/termos e abreviações que são usadas no documento.</i></p> <p><b>2. Princípios Técnicos</b> <i>Esta seção descreve os princípios técnicos que deverão ser adotados como padrões para o desenvolvimento do projeto de interface da aplicação, como por exemplo, Web Browser padrão, padronização de botões, utilização de frames, arquitetura navegacional usada, etc.</i></p> <p><b>3. Visão Geral da Interface do Sistema</b> <i>Esta seção descreve os objetivos gerais e objetivos específicos da interface do sistema, bem como a identificação do público alvo do sistema.</i></p> <p><b>4. Requisitos de Interface</b> <i>Nessa seção são descritos os Requisitos de Interface da Aplicação, com identificação única, abordando usabilidade, identidade visual, etc.</i></p> <p><b>5. Referências</b> <i>Essa seção apresenta as referências usadas no documento e que podem ser consultadas.</i></p>
---

#### 4.2.4.1.3. Artefato: Especificação de Casos de Uso de Interface

O documento de Especificação de Casos de Uso de Interface proposto para o WRE-Process, estende o Modelo de Casos de Uso Storyboard pois especifica os cenários de uso da interface do sistema, visando possibilitar o conhecimento dos conteúdos visuais e textuais das telas a serem elaboradas no projeto de interface. Estes cenários apresentam possíveis interações entre o usuário e a aplicação web. Este artefato é produzido pelo projetista de interface.

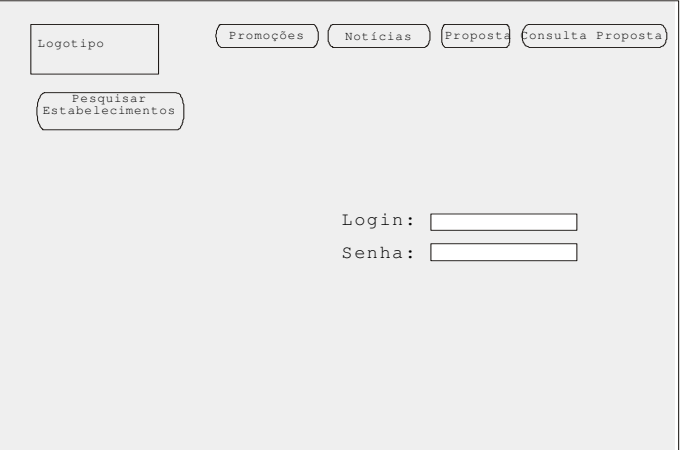

Este artefato auxilia o projetista a elaborar o protótipo de interface e aos desenvolvedores, no momento da implementação.

É opcional desenvolver diagramas de seqüência e/ou colaboração ao especificar os casos de uso de interface. A confecção destes diagramas gera sobrecarga de trabalho, mas em compensação, permite descrever as interações entre os componentes do caso de uso de interface. Com o desenvolvimento destes diagramas e as especificação de casos de uso, é possível, através

da técnica e da ferramenta proposta por Sanchez [San02], gerar os protótipos de interface automaticamente.

A Tabela 4.14 apresenta um template da Especificação dos Casos de Uso de Interface e o Apêndice D apresenta um documento de Especificação de Casos de Uso de Interface.

Tabela 4.14 - Template Especificação de Casos de Uso de Interface

<p><b>1.Introdução</b>  <i>A introdução apresenta uma visão geral do documento de Casos de Uso de Interface, incluindo seu propósito, o escopo do sistema e as definições/termos e abreviações que são usadas no documento.</i></p> <p><b>2.Caso de Uso</b>  <i>Descrição textual de alto nível apresentando como será a interação entre o usuário e o sistema.</i></p> <p><b>3.Cenários</b>  <i>Esta seção descreve os cenários de uso exclusivos de cada tipo de usuário (ator do modelo de casos de uso de interface).</i>  <i>Exemplo:</i></p> <ul style="list-style-type: none"> <li>- Abre a tela do sistema</li> <li>- Clica na opção de promoções</li> </ul>	
<ul style="list-style-type: none"> <li>- Abre-se uma janela flutuante com as promoções.</li> <li>- Depois de ler as promoções fecha a janela flutuante.</li> </ul>	
<p><b>4. Usabilidade</b>  <i>Essa seção apresenta uma descrição textual dos requisitos específicos de usabilidade que precisam ser considerados ao produzir o protótipo e implementar a interface com o usuário</i></p> <p><b>5. Diagramas</b>  <i>Essa seção apresenta os diagramas de seqüência e colaboração, especificando como o caso de uso é realizado em termos de objetos de fronteira e atores.</i></p>	

*Também são apresentados os diagramas de classe, descrevendo as classes de fronteira e os relacionamentos presentes na realização do caso de uso.*

#### **6. Referências**

*Essa seção apresenta as referências usadas no documento e que podem ser consultadas.*

Nesta seção descrevemos a atividade *Especificar Requisitos de Interface* e os artefatos produzidos na mesma. A definição da interface com o usuário é uma atividade crítica no desenvolvimento de software; e para aplicações Web a navegabilidade é um fator que torna esta atividade ainda mais importante. Sendo assim, definimos aqui um conjunto de artefatos que definem a estrutura navegacional da aplicação. O projeto navegacional evoluirá naturalmente a partir do *Modelo de Casos de Uso de Interface*, pois serão identificadas as classes que representam as páginas de informação e os caminhos navegacionais, gerando, então o modelo navegacional. A atividade não define o passo a passo para a criação do modelo navegacional pois este não é o escopo deste trabalho, haja vista estarmos definindo um processo de engenharia de requisitos, não de análise e projeto de sistemas. Diversos trabalhos têm sido realizados relativos à definição de navegação de aplicações Web, apresentando métodos adequados para a derivação do modelo navegacional, vide [Sou02], [Sch98], [Tro97] e [Koc00]. Na próxima subseção detalharemos a atividade *Projetar Visibilidade da Aplicação*, ainda no Subfluxo *Definir a Interface*.

#### **4.2.4.2. Atividade: Projetar Visibilidade da Aplicação**

Os desenvolvedores, em geral, têm uma perspectiva diferente da aplicação, com relação ao conteúdo e funcionalidades. O objetivo desta atividade é a especificação do padrão de interface para as funcionalidades, estruturas de informação e caminhos navegacionais. Sendo assim, os projetistas de interface devem desenvolver guias para auxiliar a construção das interfaces e um documento descritivo do *layout* adotado. Esta atividade deve ser realizada, principalmente, para aplicações de comportamento complexo.

Os guias para construção de interface são documentos que ensinam como as interfaces devem ser criadas, os padrões de arquivos usados, os tipos e tamanho de fontes, etc. O conteúdo dos guias depende da tecnologia adotada para a construção da aplicação. Por exemplo, as interfaces web de uma aplicação podem ser arquivos .HTML, já de outra aplicação, podem ser .JSP. A formatação dos guias de construção da interface é particular para cada tipo de arquitetura

e ferramentas usadas no projeto. Por este motivo, o WRE-Process não define template para este artefato.

O documento de descrição do *layout* descreve, de acordo com os requisitos de interface levantados, o que é necessário para criar a interface de acordo com os padrões de usabilidade, navegabilidade e design gráfico. A Tabela 4.15 apresenta o template deste documento.

*Tabela 4.15- Template do documento de descrição do layout*

<p><b>1. Introdução</b> <i>Apresenta a finalidade e padrões adotados pelo presente documento.</i></p> <p><b>2. Definição conceitual da proposta:</b> <i>Apresenta e defende os conceitos empregados no desenvolvimento da interface no que diz respeito ao design gráfico e usabilidade, dentro do escopo de interatividade com o usuário da apresentação da informação. São estabelecidos os padrões de desenvolvimento da interface</i></p> <p><b>Cores</b> <i>Apresenta o padrão de cores utilizado na interface</i></p> <p><b>Imagens</b> <i>Apresenta o padrão de apresentação de imagens</i></p> <p><b>Menus</b> <i>Apresenta o padrão de apresentação dos menus de acesso</i></p> <p><b>3. Referências</b> <i>Contém uma lista de referências para outros documentos e sites relacionados</i></p>
--

Descrevemos a atividade *Projetar Visibilidade da Aplicação*, na próxima subseção detalharemos a atividade *Definir Protótipo da Interface* no Subfluxo *Definir a Interface*.

#### **4.2.4.3. Atividade: Definir Protótipo da Interface**

O objetivo desta atividade é criar um protótipo da interface do usuário, com base nos requisitos e casos de uso de interface. O RUP prevê esta atividade no Subfluxo Refinar a Definição do Sistema, entretanto, no WRE-Process o protótipo é definido no Subfluxo Definir a Interface.

A utilização de protótipos como técnica tanto de validação como de elicitação está cada vez mais difundida entre as práticas de engenharia de requisitos [Dur00]. Deve-se salientar que o protótipo tem como principal objetivo permitir que os clientes, usuários e desenvolvedores tenham uma idéia mais clara sobre o produto a ser desenvolvido. Como o software é um produto subjetivo, o protótipo permite um grau de comunicabilidade mais alto que os documentos textuais. Além disto, permite também que a interface seja validada pelo cliente retornando um *feedback* aos Engenheiros de Requisitos e Projetistas de Interface.

As atuais ferramentas de *Desenvolvimento Rápido de Aplicações* [Dur00] (*RAD, Rapid Application Development*) permitem a construção rápida de interfaces de usuários. Como o WRE-

Process é um processo genérico de Engenharia de Requisitos, que pode ser associado às diversas técnicas, é possível aplicar a técnica de Sanchez [San02] para gerar o protótipo automaticamente. Entretanto, como o escopo deste trabalho foca na definição do processo, não validaremos a aplicação de técnicas para a geração de protótipos.

Descrevemos a atividade *Definir Protótipo da Interface*, na próxima subseção apresentaremos a atividade *Projetar Conteúdo* no Subfluxo *Definir a Interface*.

#### **4.2.4.4. Atividade: Projetar Conteúdo**

As aplicações Web além de encapsularem componentes de software responsáveis pela execução de funções para a realização de algum negócio, também possuem um caráter de disseminação e ou apresentação de informações a respeito do negócio por elas suportado. E, uma vez definida, essa informação precisa ser levantada e projetada de acordo com sua especificação [Ara01]. Por este motivo foi inserida a atividade *Projetar Conteúdo* no Subfluxo *Definir a Interface*.

O projetista de conteúdo, a partir das páginas de informação identificadas no *Modelo de Casos de Uso de Interface*, pesquisa e edita as informações necessárias. Como esta atividade é fundamentalmente de cunho editorial, seu responsável deve ter um perfil que envolva habilidades como as de jornalistas, escritores, etc. Os detalhes de execução desta atividade não foram definidos nesta dissertação por não representar o foco deste trabalho.

Descrevemos as atividades executadas no Subfluxo *Definir a Interface*. Na próxima seção apresentaremos o Subfluxo *Gerenciar o Escopo do Sistema*.

#### **4.2.5. SUBFLUXO: GERENCIAR O ESCOPO DO SISTEMA**

Após identificar e descrever os requisitos, atores, casos de uso e integrar as visões funcional e não-funcional da aplicação, o Engenheiro de Requisitos deve determinar o escopo da iteração corrente, ou seja o que deve estar “pronto” ao término da iteração, dar condições, aos arquitetos de software, de identificarem os casos de uso mais relevantes para esta iteração. Normalmente, na iteração inicial do projeto são priorizados os casos de uso que ajudam a definir e validar a arquitetura do sistema.

Os objetivos deste subfluxo são:

- priorizar e refinar os requisitos que devem ser inseridos na iteração corrente,
- definir o conjunto de casos de uso que têm relevância significativa para esta iteração.

- definir como manter a rastreabilidade dos requisitos

O escopo de um projeto é definido pelo conjunto de requisitos alocados a ele. Gerenciar o escopo é uma atividade contínua, que requer o desenvolvimento iterativo e incremental, quebrando o escopo geral do projeto em partes pequenas e mais fáceis de gerenciar.

A Figura 4.11 apresenta o Subfluxo Gerenciar Escopo e salientamos a integração entre este subfluxo e o processo de Gerência de Configuração e Mudança que deve ser aplicado no projeto, no que diz respeito ao gerenciamento e controle das solicitações de mudanças, por este motivo há uma área pontilhada na Figura 4.11 contornando a atividade *Revisar Solicitações de Mudança* e o executor *Gerente de Configuração e Mudanças*.

As subseções a seguir apresentam as atividades: *Priorizar Casos de Uso* e *Gerenciar Dependências*.

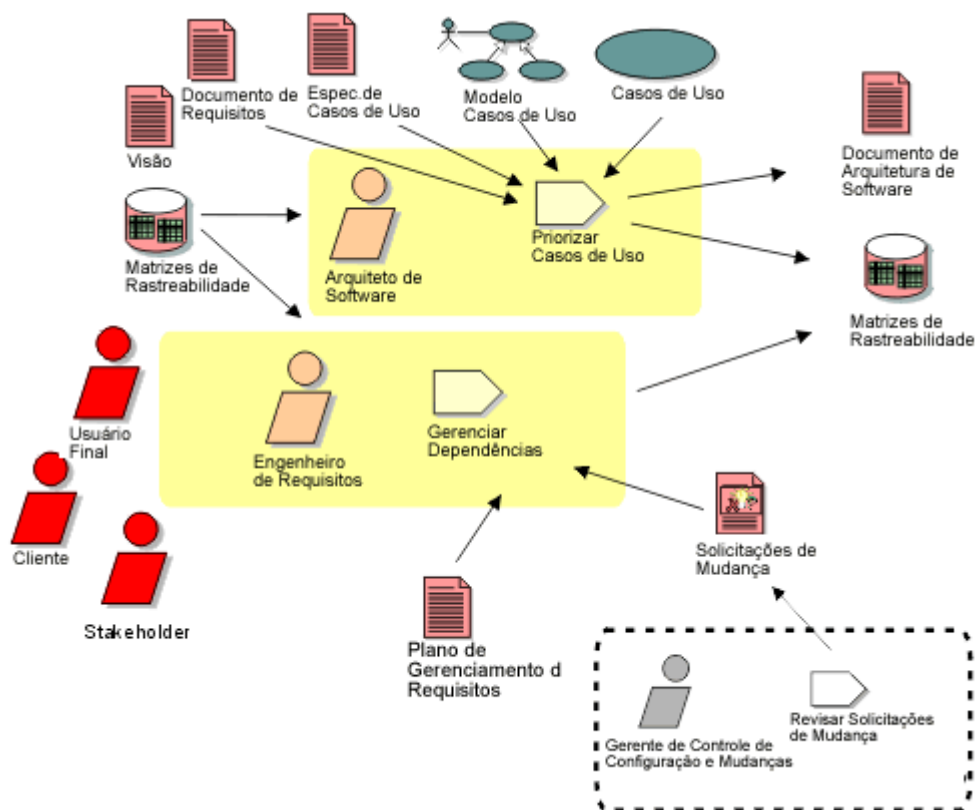


Figura 4.11- Subfluxo Gerenciar Escopo

#### 4.2.5.1. Atividade: Priorizar Casos de Uso

O objetivo desta atividade é selecionar um conjunto de casos de uso para serem trabalhados na iteração corrente. Sendo assim, o arquiteto de software prioriza os casos de uso de acordo com a importância que eles têm para a validação da arquitetura ou pelos riscos a eles

associados ou ainda, aqueles que representam funcionalidades centrais da aplicação. Com os casos de uso priorizados, os arquitetos refinam a iteração e modelam as visões de casos de uso (fase de análise).

Esta atividade auxilia no desenvolvimento do plano da iteração, que é criado pelo gerente de projeto.

#### **4.2.5.2. Atividade: Gerenciar Dependências**

Como já foi apresentado no *Subfluxo Elicitar Requisitos*, seção 4.2.2.6, esta atividade tem o objetivo de ajudar a gerenciar o escopo do projeto e as alterações dos requisitos, através das solicitações de mudança e do rastreamento dos requisitos. Devido à característica iterativa e incremental do processo, a realização desta atividade é constante, permitindo refinamentos e otimizações das matrizes de rastreabilidade utilizadas no projeto.

Desta forma, ao serem priorizados os casos de uso, as matrizes de rastreabilidade precisam ser analisadas e atualizadas. Dependendo da técnica de rastreabilidade adotada para o projeto, pode ser necessário inserir, nestas matrizes, rastreamento entre o documento de arquitetura de software e casos de uso. É fundamental que o Engenheiro de Requisitos mantenha a rastreabilidade apropriada para os artefatos gerados na fase de requisitos.

Descrevemos as atividades executadas no Subfluxo *Gerenciar o Escopo do Sistema*. Na próxima seção apresentaremos o Subfluxo *Refinar a Definição do Sistema*.

#### **4.2.6. SUBFLUXO: REFINAR A DEFINIÇÃO DO SISTEMA**

Este subfluxo tem o objetivo de descrever, com detalhes, os casos de uso priorizados para a iteração corrente. A Figura 4.12 apresenta o Subfluxo Refinar a Definição do Sistema.

O RUP sugere a construção do protótipo de interface nesse subfluxo, entretanto, no WRE-Process essa atividade é realizada no *Subfluxo Definir Interface*, conforme já foi apresentado e justificada sua relevância para aplicações web. As próximas subseções apresentam as atividades *Detalhar Casos de Uso* e *Detalhar Requisitos de Software*.

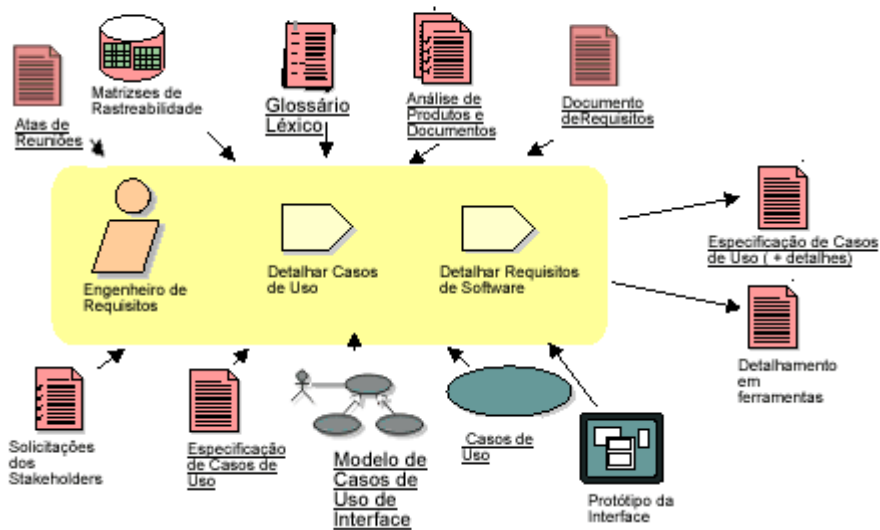


Figura 4.12 - Subfluxo Refinar a Definição do Sistema

#### 4.2.6.1. Atividade: Detalhar Casos de Uso

O objetivo desta atividade é acrescentar detalhes à descrição dos fluxos de eventos dos casos de uso priorizados para a iteração corrente. Como o documento de especificação de casos de uso é uma base para os projetistas e implementadores, detalhar a especificação auxilia o trabalho nas fases de análise e projeto, implementação e testes destes casos de uso. Além disso, o cliente e usuários finais contam com uma documentação mais rica em informações.

Na atividade Documentar Requisitos, do Subfluxo Analisar e Documentar Requisitos, os casos de uso foram descritos, mas não necessariamente contém todas as informações necessárias para o desenvolvedor. Ao detalhar os casos de uso as seções do documento, apresentado no template da Tabela 4.12, são analisadas, podendo serem enriquecidas com mais informações. O Apêndice B apresenta alguns casos de uso detalhados.

#### 4.2.6.2. Atividade: Detalhar Requisitos de Software

Esta atividade tem o objetivo de organizar o conjunto de artefatos gerados, acrescentar detalhes ao documento de requisitos e garantir que, com as alterações, a rastreabilidade esteja



sendo mantida. Caso estejam sendo usadas ferramentas para gerência de requisitos, por exemplo, as informações devem ser detalhadas nas ferramentas.

Descrevemos as atividades executadas no Subfluxo *Refinar a Definição do Sistema*. Na próxima seção apresentaremos o Subfluxo *Validar os Requisitos*.

#### **4.2.7. SUBFLUXO: VALIDAR OS REQUISITOS**

A validação dos requisitos é o estágio final do processo de Engenharia de Requisitos. O objetivo deste subfluxo é garantir que os requisitos do sistema atendem às expectativas dos usuários e clientes [Som98]. O processo de validação envolve os *stakeholders*, engenheiros de requisitos, projetistas de interface e projetistas de sistema, onde todos devem analisar os requisitos e identificar se existem problemas, ambigüidades ou omissões. Em alguns casos, a validação pode envolver, também, pessoas conhecedoras do domínio da aplicação e pessoal de marketing, por exemplo.

A validação é um processo delicado, pois envolve diversas pessoas. Essas pessoas devem ler e analisar os documentos que especificam os requisitos e verificar se estes realmente atendem às expectativas dos *stakeholders* e clientes, bem como os padrões organizacionais.

O RUP contempla, no subfluxo *Gerenciar Mudanças dos Requisitos*, a atividade *Revisar Requisitos*, que considera a condução da revisão através de uma reunião onde é verificada a qualidade dos artefatos revisados.

Como a validação é uma etapa de grande importância para a Gerência de Requisitos, o WRE-Process contém um subfluxo específico para realização dessa fase. A validação dos requisitos do WRE-Process é embasada na metodologia de Amador Duran [Dur00], apresentada no Capítulo 2, seção 2.5.4. São combinadas as técnicas de *walkthrough* dos casos de uso e a utilização de protótipos de interface de usuários, para ajudar aos clientes e usuários a revisarem a documentação de requisitos.

A Figura 4.13 apresenta o Subfluxo Validar Requisitos proposto no WRE-Process.

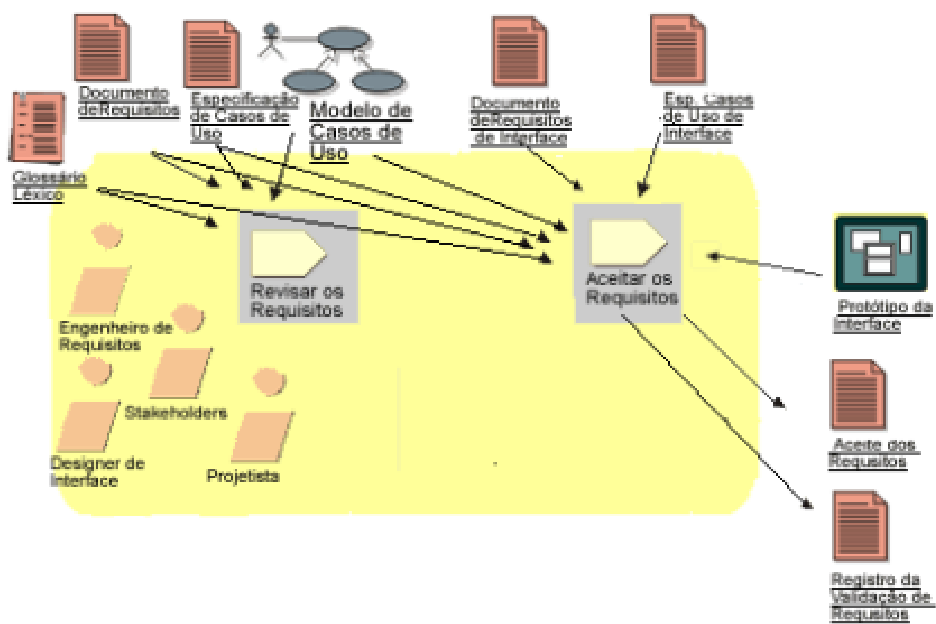


Figura 4.13- Subfluxo Validar Requisitos

#### 4.2.7.1. Atividade: Revisar os Requisitos

O objetivo desta atividade é permitir que um grupo de pessoas representando os *stakeholders*, usuários finais e cliente, leia e analise os artefatos que descrevem os requisitos do sistema (Por exemplo: *Glossário Léxico*, *Documento de Requisitos*, *Especificação de Casos de Uso*, *Requisitos de Interface* e *Especificação de Casos de Uso de Interface*) procurando por possíveis problemas (anomalias e/ou omissões). Esta atividade é contemplada através da técnica de *walkthrough* [Dur00], onde os participantes da validação recebem, com antecedência, o material a ser validado para que possam estudá-lo e preparar as questões a serem discutidas.

Caso seja um sistema de médio à grande porte, cada revisor deve analisar os requisitos e casos de uso por pacote, de acordo com sua área de trabalho. No *Plano de Validação de Requisitos*, produzido no Subfluxo *Identificar Requisitos*, vide seção 4.2.1.3, é relacionada uma equipe multidisciplinar de revisores responsáveis pela execução desta atividade.

Os tipos de problemas mais comuns que podem ser encontrados são:

- falta de clareza na especificação: Os requisitos e/ou casos de uso podem estar mal expressos ou com omissão de informações. O autor, então, terá que melhorar a documentação,

- conflito de informações: Caso haja conflitos entre os requisitos, é necessário que haja uma negociação posterior com os stakeholders,
- falta de informações: Caso o revisor sinta falta de determinadas informações, é tarefa do Engenheiro de Requisitos pesquisá-las junto aos stakeholders ou outras fontes,
- requisitos irreais: Caso o revisor verifique que os requisitos e/ou casos de uso não são implementáveis na tecnologia disponível.

Em uma revisão de requisitos, a equipe de revisão deve verificar cada requisito em termos de sua consistência e completude. Além disto, os revisores devem averiguar [Som03]:

- *facilidade de verificação*: o requisito é realmente passível de ser testado?
- *facilidade de compreensão*: O requisito pode ser adequadamente compreendido pelos clientes e usuários finais?
- *facilidade de rastreamento*: A origem do requisito é claramente definida?
- *adaptabilidade*: O requisito é adaptável? Ele pode ser modificado sem que isso provoque efeitos colaterais em outros requisitos do sistema?

Conflitos, contradições, erros e omissões nos requisitos devem ser destacados durante a revisão e formalmente registrados. Desta forma realizamos a Revisão dos Requisitos, na próxima seção apresentamos a atividade *Aceitar os Requisitos*, onde os requisitos são efetivamente validados.

#### **4.2.7.2. Atividade: Aceitar os Requisitos**

Após a revisão dos documentos, deve haver uma reunião formal com todas as pessoas envolvidas na Validação de Requisitos. Como apresentado no Capítulo 2, seção 2.5.4, na técnica de *walkthrough*, [Dur00], podem ser identificados os seguintes papéis para os participantes das sessões de validação: *Autor, Apresentador, Moderador, Registrador e demais Revisores*.

Durante a reunião, o *Apresentador* lê cada requisito, com detalhes, para o grupo de pessoas da sessão, com o objetivo de gerar discussões sobre os mesmos. Com isto, problemas podem ser identificados.

Quanto aos participantes da reunião de validação dos requisitos, deve-se pesar entre incluir todas as pessoas desejadas e manter a reunião gerenciável e produtiva. É necessário ter prudência e convidar apenas aqueles indivíduos que têm a contribuir para que o objetivo da reunião seja atingido. Algumas vezes é mais produtivo ter várias reuniões de validação com um menor número de participantes que apenas uma envolvendo muitas pessoas.

A proposta metodológica para validação de requisitos apresentada por Duran [Dur00], valida a documentação de requisitos mediante *walkthroughs* assistidos por protótipos de interface de usuário. Esta técnica permite uma validação conjunta, de forma que os casos de uso guiam a sessão de validação com a apresentação do protótipo referente a cada funcionalidade da aplicação.

Desta forma, sugerimos, inicialmente validar o *Glossário Léxico*, o *Documento de Requisitos*, o *Documento de Requisitos de Interface*. Após o entendimento geral destes documentos, é a vez de validar a *Especificação de Casos de Uso* junto com o *Protótipo da Interface*. Essa estratégia facilita a validação, pois algumas pessoas sentem dificuldades em visualizar como sentenças escritas nos documentos podem ser transformadas em software executável.

Sommerville sugere, em [Som98], a utilização de questões de *checklist* gerais para serem aplicadas na validação de cada requisito. Estas questões estariam relacionadas à qualidade dos requisitos, por exemplo: fácil entendimento, redundância, completude, ambigüidade, consistência, etc. A cada requisito ou caso de uso validado, as questões são aplicadas para averiguar se realmente são atendidas.

Segundo Pinheiro, em [Pin03], para aplicações Web, direcionadas ao público em geral, a validação também deve considerar: (i) a disponibilização de protótipos para avaliação do sistema e do comportamento do usuário ao usá-lo [Rup02]; (ii) a coleta de possíveis sugestões dos usuários; (iii) a escolha de um grupo de usuários piloto para avaliação da aplicação antes de torná-la disponível publicamente, a exemplo dos softwares de prateleira

Para registrar os resultados da validação de requisitos, o registrador deve gerar o artefato *Registro da Validação de Requisitos*, que é um documento formal criado para auxiliar a gerenciar as revisões necessárias em cada artefato, a Tabela 4.16 apresenta um template deste documento. Para cada requisito e caso de uso validados, por exemplo, são enumeradas as alterações necessárias.

Tabela 4.16- Template do Registro da Validação dos Requisitos

<p><b>1. Introdução</b> <i>Apresenta uma visão geral do documento, identificando o sistema, a data/hora e local onde foi realizada a reunião de validação e os artefatos validados.</i></p> <p><b>2. Participantes da Validação</b> <i>Lista dos indivíduos que participaram da revisão/validação dos requisitos e seus papéis.</i></p> <p><b>3. Problemas identificados e recomendações</b> <i>Esta seção lista os problemas encontrados durante a validação. Os tipos de problemas mais comuns são:</i></p> <ul style="list-style-type: none"><li>• <i>Problemas nos artefatos: defeitos que requerem correção. Neste caso, são listados, para cada artefato, os problemas encontrados. (É fundamental que sejam identificados os requisitos e casos de uso que necessitam de correção, não apenas o artefato!)</i></li><li>• <i>Problemas no projeto, cujos sintomas só foram identificados na revisão dos artefatos. Neste caso, são indicadas as diretrizes para solucionar cada problema.</i></li><li>• <i>Problemas no produto, cujos sintomas só foram identificados na revisão dos artefato. Neste caso, são indicadas as diretrizes para solucionar cada problema.</i></li></ul> <p><b>4. Lista de ações</b> <i>Para os problemas identificados no item anterior, são listadas as ações a serem executadas, os responsáveis pela execução e o prazo de conclusão.</i></p> <p><b>5. Previsão de Liberação</b> <i>Determina os prazos para a correção e especifica se haverá uma nova reunião de validação e a data/hora prevista</i></p> <p><b>6. Referências</b> <i>Essa seção apresenta as referências usadas no documento e que podem ser consultadas.</i></p>
--

Após os requisitos serem validados, deve ser gerado um termo de aceite dos requisitos, onde o cliente assina concordando com a especificação de requisitos apresentada.

Descrevemos as atividades executadas no Subfluxo *Validar os Requisitos*. Na próxima seção apresentaremos o Subfluxo *Gerenciar Mudanças de Requisitos*.

#### **4.2.8. SUBFLUXO: GERENCIAR MUDANÇAS DE REQUISITOS**

Inevitavelmente ocorrem mudanças nos requisitos de uma aplicação. Por esta razão, as atividades do *Subfluxo Gerenciar Mudanças de Requisitos* são aplicadas, continuamente, durante todo o ciclo de vida do projeto. Os objetivos deste subfluxo são:

- Avaliar as solicitações de mudança submetidas e determinar seu impacto sobre o conjunto de requisitos existente;
- Rastrear os requisitos através das matrizes de rastreabilidade.

O gerenciamento de requisitos é o processo de compreender e controlar as mudanças nos requisitos de sistemas. As atividades do subfluxo de gerência de requisitos são realizadas em conjunto com os demais fluxos e subfluxos do processo de desenvolvimento [Som03]. Mudanças nos requisitos, naturalmente, afetam todas as fases do ciclo de vida de desenvolvimento do software, por exemplo: o modelo do sistema, que é desenvolvido na Fase de Análise e Projeto, os

documentos e projetos de testes, que são criados na Fase de Testes (baseados nos requisitos definidos), entre outros artefatos. Os relacionamentos de rastreabilidade identificados ao Gerenciar Dependências (vide seções 4.2.2.6,4.2.5.2) são os pontos chave para podermos entender os impactos. Por este motivo, consideramos como artefatos de entrada para a gerência de requisitos os diversos documentos e diagramas gerados em todas as etapas do processo de construção do software.

A Figura 4.14 apresenta o Subfluxo *Gerenciar Mudanças nos Requisitos* proposto para o WRE-Process. Comparando-o com o mesmo subfluxo do RUP 2002, verifica-se que, no RUP, a atividade *Revisar Requisitos* é executada nesse subfluxo. Entretanto, como o WRE-Process contempla o Subfluxo *Validar Requisitos*, vide seção 4.2.7, a revisão dos requisitos é realizada durante a validação dos requisitos, como foi tratado na seção anterior. Outra atividade sugerida pelo RUP, para este subfluxo, mas que não é contemplada no WRE-Process, é “*Estruturar o Modelo de Casos de Uso*”. Segundo nossa proposta, à medida que o modelo é desenvolvido e refinado, sua estrutura também é definida, ou seja, os casos de uso que incluem ou estendem a chamada a outro, são identificados sempre que o modelo for refinado. Desta forma, no subfluxo *Gerenciar Mudanças nos Requisitos* do WRE-Process contempla apenas a atividade *Gerenciar Dependências*, que também é executada em diversos subfluxos do processo com o objetivo de auxiliar a gerenciar o escopo do projeto e as mudanças nos requisitos, a próxima subseção apresenta esta atividade.

Observe a integração entre este subfluxo e o processo de Gerência de Configuração e Mudança que deve ser aplicado no projeto, no que diz respeito ao gerenciamento e controle das solicitações de mudanças, por este motivo há uma área pontilhada na Figura 4.14 contornando a atividade *Revisar Solicitações de Mudança* e o executor *Gerente de Configuração e Mudança*.

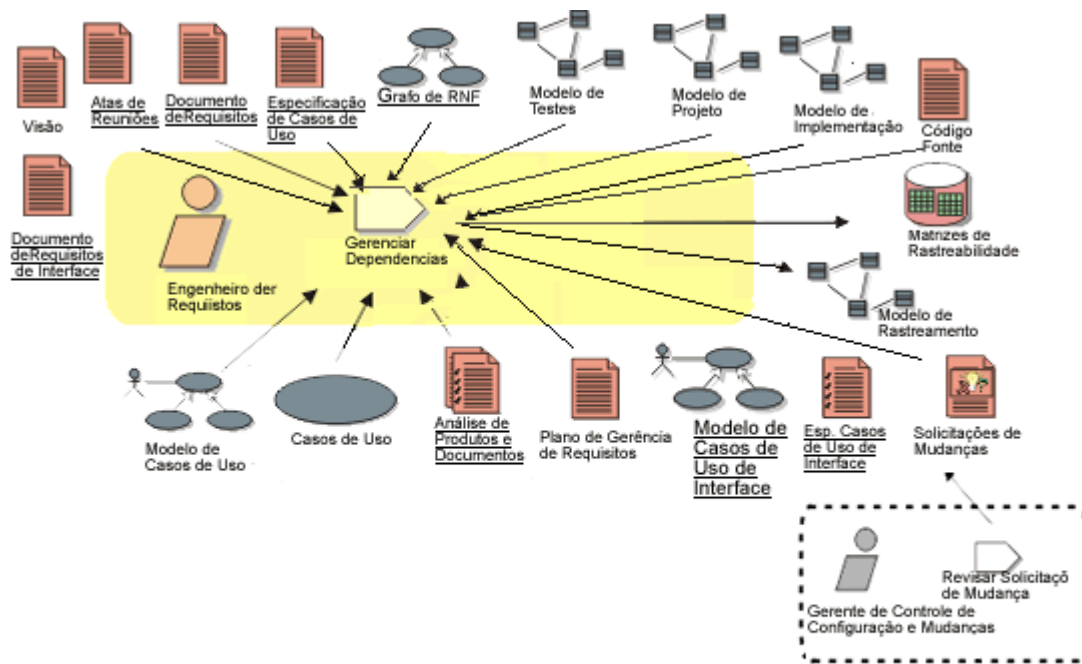


Figura 4.14 - Subfluxo Gerenciar Mudanças nos Requisitos

#### 4.2.8.1. Gerenciar Dependências

Como já foi vista no *Subfluxo Elicitar Requisitos*, seção 4.2.2.6, o objetivo desta atividade é usar os atributos e a rastreabilidade dos requisitos para auxiliar a gerenciar o escopo do projeto e as mudanças nos requisitos.

No Plano de Gerência de Requisitos são definidos os atributos para cada tipo de requisito. Na atividade Gerenciar Dependências, são identificados e analisados os conteúdos para cada atributo dos requisitos. Assim, requisitos instáveis, com alto risco, que exigem muito esforço e que trazem benefícios à aplicação, devem ser analisados com mais rigor. Já aqueles requisitos que proporcionam baixo benefício, com alto risco e exigem muito esforço, são candidatos a serem removidos.

Ainda no Plano de Gerência de Requisitos, é definido como os requisitos são rastreados com outros artefatos. O Engenheiro de Requisitos deve, então, estabelecer a rastreabilidade desejada e, periodicamente, consultar relatórios para averiguar se a rastreabilidade está sendo mantida conforme a especificação do plano.

Em [Spe03], Spence cita que embora existam diversas estratégias de rastreabilidade que podem ser usadas para facilitar o processo de gerenciamento de requisitos, é possível utilizar qualquer uma delas ao trabalhar com um framework como o RUP.

Com base no último parágrafo, vemos que a equipe de desenvolvimento pode decidir que técnica de rastreamento é mais adequada para o projeto. Nessa dissertação, sugerimos a utilização da proposta de Toranzo, citada em [Tor02-A] e [Tor02-B] e apresentada no Capítulo 2, seção 2.5.5, devido aos benefícios desta técnica e por termos utilizado-a para validar a proposta. Entretanto, o WRE-Process é flexível para que outra técnica possa ser utilizada.

Ao aplicar a técnica de Toranzo [Tor02-B], devem ser seguidas suas recomendações para construir um modelo de rastreamento, bem como as diretrizes relacionadas com cada recomendação. Com a utilização dessa técnica, durante o planejamento devem ser informadas as expressões e convenções usadas nas matrizes de rastreamento. Depois disto, são identificadas as classes de informações que podem afetar o sistema, nos níveis ambiental, organizacional, gerencial e de desenvolvimento. Após a identificação de todas as classes (cada classe é representada em uma planilha que deve ser preenchida com os dados referentes à mesma), devem ser organizadas e estruturadas as classes candidatas, para então, estabelecer os relacionamentos entre elas. Com isto, são identificadas as propriedades de cada classe e definidas/preenchidas as matrizes de rastreabilidade para cada relacionamento encontrado no modelo. Estas matrizes são, então validadas, através de uma análise de impacto dos dados envolvidos.

A principal vantagem na utilização desta técnica é a possibilidade de rastrear qualquer tipo de artefato gerado no desenvolvimento da aplicação (por exemplo: Requisitos x Casos de Uso, Requisitos x Código, Requisitos x Casos de Testes, etc.). Além disto, ela permite rastrear os requisitos analisando seus relacionamentos com diversos tipos de informações e artefatos e fornece um processo simples para identificar as classes e relacionamentos que podem compor o modelo de rastreamento de um projeto.

Tradicionalmente, o documento de requisitos é baseado na linguagem natural. A ênfase desse documento é conter requisitos claros, consistentes e não ambíguos. Mas existem várias dificuldades atribuídas à gerência manual dos documentos de requisitos:

- é difícil manter o documento atualizado e sincronizado com outros documentos. Já é complicado manter um documento com uns poucos requisitos, então a situação pode ser mais difícil se devemos gerenciar vários documentos com centenas de requisitos;



- a identificação manual dos artefatos afetados pela introdução de uma mudança de requisitos no sistema pode contribuir para elevar os custos de um projeto;
- a dificuldade para comunicar as mudanças aos membros de um projeto;
- a gerência manual das propriedades dos requisitos é trabalhosa;
- é difícil definir e registrar os relacionamentos entre artefatos porque, em geral, não existe uma cultura computacional sobre a gerência de requisitos.

Uma solução alternativa para alguns desses problemas é introduzir uma ferramenta de gerência de requisitos no processo de desenvolvimento. Algumas das funcionalidades das ferramentas de gerência de requisitos são: armazenar requisitos; identificar de forma única todos os requisitos; facilitar a seleção manual ou automática dos requisitos de um texto; definir e manipular as propriedades dos requisitos; e gerar filtros (visões) e relatórios sobre os requisitos. As ferramentas RequisitePro, Caliber, QSSrequireit, DOORS e RTM são as mais usadas, atualmente, na indústria. Infelizmente, para a técnica de Toranzo, utilizada nessa dissertação, ainda não existe uma ferramenta que automatize o processo de rastreamento de requisitos.

Concluimos a apresentação dos subfluxos e atividades propostas no WRE-Process. A próxima seção apresenta a conclusão deste capítulo.

### **4.3. Considerações Finais**

Neste capítulo, foi apresentada uma proposta de um processo para a engenharia e gerência de requisitos em um projeto de software, baseado no processo genérico de ER, descrito por Sommerville em [Som98] e no fluxo de Requisitos do RUP 2002 [Sav01]. Nosso objetivo ao propormos este processo é, além de preencher algumas lacunas existentes no fluxo de requisitos do RUP, atender mais apropriadamente o desenvolvimento de aplicações Web. Isto não significa que o processo seja restrito a aplicações WEB, na verdade ele é um processo genérico, que pode ser adaptado a qualquer tipo de aplicação, mas considera aspectos fundamentais para aplicações web, que não são contemplado em outros processos.

Comparando o WRE-Process com o RUP 2002 e o processo genérico, podemos observar que as principais características desta proposta foram: a inclusão de um subfluxo para Definir a Interface de uma aplicação, já que isto é fundamental em aplicações web; a criação do subfluxo Validar Requisitos, que é uma atividade presente no processo genérico e de grande importância; e a criação de atividades e artefatos para integrar as visões funcional e não funcional do software, onde os aspectos não-funcionais são incorporados aos funcionais, permitindo que sejam

detectadas as interferências entre os requisitos. Além disso, a nomenclatura de alguns subfluxos foi alterada para que ficassem mais próximas do processo genérico. Atividades e artefatos foram criados nos diversos subfluxos, baseados nas propostas acadêmicas e na experiência prática.

Outro ponto importante no processo de Engenharia e Gerência de Requisitos é que para uma organização atingir o nível de maturidade 2 do CMM, uma das premissas é que seu processo de Engenharia e Gerência de Requisitos seja efetivo. Apesar do CMM estabelecer a ACP Gerência de Requisitos, ele não descreve um processo de definição de requisitos. As organizações que desejarem realizar uma gerência efetiva de requisitos devem definir e documentar seu próprio processo de definição de requisitos para auxiliar na elicitação, documentação, validação e gerência dos mesmos. O WRE-Process é um processo que cobre os requisitos da ACP de gerência de requisitos do CMM

O próximo capítulo apresenta a validação da nossa proposta através da aplicação da mesma em um sistema de gestão de acessos para aplicações em ambiente web, desenvolvido pelo CESAR para uma Operadora de Cartões de Crédito.

## Capítulo 5 RELATO DE EXPERIÊNCIA: SISTEMA DE GESTÃO DE ACESSO EM APLICAÇÕES WEB

---

### 5.1. Visão Geral

Este capítulo compreende a aplicação das práticas de engenharia e gerência de requisitos propostas neste trabalho sobre o desenvolvimento de uma solução para a gestão de acesso de usuários em um ambiente Web. Esta aplicação, chamada Sistema de Gestão de Acesso, foi contratada por uma operadora de cartão de crédito e desenvolvida pelo CESAR (Centro de Estudos e Sistemas Avançados do Recife) no período entre agosto/2001 e janeiro/2003. O processo de software padrão utilizado neste desenvolvimento foi o PROSCES, que é uma customização do RUP, de propriedade do CESAR. Durante o desenvolvimento desta aplicação, grande parte das atividades e artefatos gerados para o cliente foram produtos do PROSCES, mas outros surgiram da necessidade específica do projeto e não estavam inicialmente previstos. O produto de software desenvolvido é de propriedade da operadora de cartões de crédito, que possui todos os direitos reservados sobre o código fonte do sistema e artefatos entregues no decorrer do projeto. Por esta razão, o conteúdo dos artefatos produzidos que estiverem relacionados com regras de negócio do software desenvolvido não serão publicados neste trabalho.

Podemos dizer que no método de pesquisa adotado, o autor desempenhou um papel ativo no levantamento do problema, haja vista ter sido o engenheiro de requisitos responsável pelo Sistema de Gestão de Acesso, além da avaliação e resolução do mesmo. Além disto, o trabalho também foi fundamentado em uma revisão bibliográfica e na experiência prática do autor. Chamamos este capítulo de “*Relato de Experiência*” porque o fruto deste trabalho corresponde à aplicação do WRE-Process para solucionar problemas encontrados no desenvolvimento do projeto real.

Na fase de Requisitos do Projeto de Gestão de Acesso, embora tenhamos seguido as diretrizes e técnicas propostas pelo RUP, encontramos algumas dificuldades:

- *entender o domínio da aplicação*: tanto o domínio, como as regras de negócio da aplicação eram bastante complexas e os artefatos produzidos não davam o suporte necessário para o perfeito entendimento;

- *encontrar atores e casos de uso*: devido à complexidade da aplicação, não foi trivial encontrar os atores e os casos de uso, e os artefatos produzidos antes da realização desta atividade não auxiliaram;
- *tratar requisitos não-funcionais*: como não havia um método bem definido para tratar as interdependências entre os requisitos não-funcionais e funcionais, surgiram dificuldades nas etapas posteriores, como por exemplo: requisitos não-funcionais não foram considerados durante o projeto e houve necessidade de alteração de código, depois de implementado, para satisfazer estes requisitos;
- *tratar requisitos de interface*: no projeto real foram criados cenários de uso da interface, entretanto sentimos falta de uma definição precisa do modelo navegacional da aplicação, bem como do projeto de conteúdo informativo, que deveria ser fornecido pelo próprio cliente.

Sendo assim, com o aprendizado obtido no desenvolvimento do sistema de gestão de acesso, conduzimos este “*Relato de Experiência*” através da execução de cada subfluxo definido no WRE-Process, buscando sempre soluções para as dificuldades encontradas, no projeto real, e mostrando os benefícios das atividades e artefatos propostos, de forma a garantir a integridade do processo e validar sua utilização.

Este capítulo é estruturado de forma que, inicialmente é apresentado o Sistema Gestão de Acesso e a descrição dos seus usuários, depois é detalhada a aplicação do WRE-Process sobre este sistema e, finalmente, a conclusão do relato de experiência.

## **5.2. Descrição do Sistema de Gestão de Acessos**

O sistema de gestão de acesso foi desenvolvido no intuito de centralizar e permitir a definição, tanto de uma política de acesso quanto de uma base de validação de acessos únicas para todos os sistemas desenvolvidos para a plataforma web.

O ambiente computacional proposto foi projetado levando em consideração os seguintes requisitos:

- o sistema deverá estar disponível 7 dias por semana, 24 horas por dia, excetuando-se horários previamente negociados com o cliente para a manutenção preventiva, *backups*, *upgrades*, manutenção do software, equipamentos, servidor ou alterações que tornem imperativa a paralisação dos serviços oferecidos pelo menor tempo possível;

- o sistema deverá suportar, inicialmente, uma média de 400 usuários/dia, sendo que destes, 200 usuários, pertencentes a Central de Atendimento, estarão acessando o sistema simultaneamente;

As seguintes políticas de identificação e acesso foram determinadas pelo cliente para serem atendidas.

- todo login de acesso deve estar associado a um conjunto de perfis pré-definidos pelos gestores dos sistemas, de acordo com as atribuições de cada usuário;
- o sistema deve gerar automaticamente, e de forma randômica, as primeiras senhas de acesso;
- o sistema deve contemplar um mecanismo para definição de: período de expiração de senhas, quantidade de acessos errados até o bloqueio e controle de senhas repetidas;
- deve ser possível manter uma trilha de auditoria, para todos os acessos e transações realizadas pelos usuários;
- as senhas de acesso devem ser armazenadas criptografadas;
- o acesso dos usuários ao sistema deve ser realizado através de um controle de perfil, a fim de permitir o acesso apenas às funções determinadas. O controle de perfil deve também validar, além do acesso a funções, o acesso aos dados da aplicação. Assim, se um usuário tiver acesso à funcionalidade alterar dados cadastrais de clientes, mas não possuir acesso ao elemento de dados cliente, ele não poderá realizar a alteração;
- cada usuário cadastrado pode ter um conjunto de perfis associados. O perfil de acesso de um usuário corresponde ao somatório de todos os perfis atribuídos a ele;
- o sistema permitirá a manutenção das informações dos usuários habilitados a utilizá-lo. O usuário responsável pela gestão de acesso realizará operações como cadastrar um novo usuário, alterar dados cadastrais, aprovar o cadastro do usuário, gerar senha descartável, bloquear, desbloquear, associar perfis de acesso e desativar usuários.
- o menu de acesso de usuários deve ser montado dinamicamente, de acordo com as permissões de acesso associadas aos perfis do usuário. Assim, só devem ser apresentados os sistemas, subsistemas e funções acessíveis ao usuário.
- todos os sistemas, subsistemas e funções devem ser cadastrados na aplicação, bem como os elementos de dados manipulados e os atributos destes elementos de dados. As

associações entre estes objetos são cadastradas no sistema e validadas de acordo com este cadastro, ou seja a aplicação deve ser configurável através de parâmetros.

### 5.3. Descrição dos Usuários da Aplicação

Os usuários da aplicação são todos os funcionários da organização, que têm acesso aos sistemas de informações, em um primeiro momento. Ao ser implantado o sistema de gestão de acessos, foi também implantado, sobre a plataforma definida pelo sistema de gestão, um sistema de autorizações de compras, cujos usuários são os operadores das centrais de atendimento localizadas em diferentes cidades. À medida que novas aplicações são desenvolvidas e implantadas, aumenta-se o leque de usuários acessando a aplicação, já que qualquer usuário passa, primeiramente, pelo sistema de gestão de acessos e todo e qualquer software para o ambiente web deve ser integrado com o mesmo. Durante a redação deste trabalho, estavam sendo testados os sistemas de serviços financeiros e contabilidade da operadora, e em concepção os sistemas de lojistas e crédito.

A expectativa é que os clientes, lojistas e as empresas conveniadas também acessem a aplicação assim que os sistemas direcionados a estes tipos de usuários sejam implantados no novo ambiente.

Para o sistema de gestão de acesso, os seguintes papéis são bem definidos:

- **gestor de Acesso:** Usuário responsável pela administração de acesso de cada sistema da aplicação, sendo responsável pelo cadastro de perfis, associação de perfis, liberação e bloqueio de acesso aos usuários;
- **administrador de segurança:** Usuário responsável pela segurança geral do sistema e definição dos parâmetros da política de segurança adotada. A cada novo sistema a ser implantado, o administrador de segurança deve avaliar as permissões de acesso liberadas e especificar as funcionalidades e atributos de elementos de dados que devem ser logados;
- **configurador de aplicações:** Estes usuários correspondem aos engenheiros de software e analistas de sistemas responsáveis pela implantação de um novo sistema;
- **usuário final:** Corresponde à grande maioria dos usuários: funcionários, estagiários, clientes, lojistas, conveniados e parceiros.

## 5.4. O Processo de Engenharia de Requisitos

No projeto real, a equipe de Engenharia de Requisitos foi composta por funcionários do CESAR, sendo um gerente de projetos e um analista de negócios, ambos experientes e com habilidades técnicas em engenharia de requisitos. Na fase de modelagem de negócios do Sistema de Gestão de Acessos foram elaboradas uma proposta técnica de desenvolvimento da aplicação, com uma “visão geral” do sistema a ser desenvolvido e os requisitos técnicos que puderam ser preliminarmente identificados, e uma proposta comercial para o desenvolvimento e implantação da solução. Desta forma, a proposta técnica identificava as solicitações dos stakeholders que haviam sido coletadas.

Nesta seção apresentamos a aplicação dos subfluxos do WRE-Process no sistema Gestão de Acesso e a execução das atividades propostas em cada subfluxo. Nosso principal objetivo é validar o processo proposto no capítulo 4, onde são utilizados os mais recentes resultados de pesquisas realizadas na área de Engenharia de Requisitos, em um projeto do mundo real.

As subseções seguintes descrevem a utilização de cada subfluxo. O primeiro subfluxo a ser tratado é o *Planejar a Engenharia de Requisitos*.

### 5.4.1. SUBFLUXO: PLANEJAR ENGENHARIA DE REQUISITOS

Para planejar a engenharia de requisitos do sistema de Gestão de Acessos, executamos as seguintes atividades: *Planejar Elicitação de Requisitos*, *Planejar Gerência de Requisitos*, *Planejar Validação de Requisitos* e *Definir a Visão do Sistema*.

As subseções seguintes detalham a execução de cada uma destas atividades.

#### 5.4.1.1. Atividade: Planejar a Elicitação de Requisitos

Durante a execução deste subfluxo foram identificados, junto à equipe da Diretoria de Tecnologia da Operadora de Cartões de Crédito os principais *stakeholders*, ou seja, as pessoas “chave”, que deveriam ser envolvidos na elicitação dos requisitos. Com isto, foi executada a atividade *Planejar a Elicitação de Requisitos*, onde foram definidas as agendas e pautas de reuniões e distribuídos, via e-mail, os convites aos participantes. A Tabela 5.1 apresenta o documento com a agenda de reuniões planejada para a elicitação de requisitos do projeto. Para mantermos em sigilo os nomes dos participantes reais, resolvemos identificá-los por *pessoa1*, *pessoa2*,..., *pessoaN*.

Tabela 5.1- Agenda de Reuniões

AGENDA DE REUNIÕES					
Levantamento de Requisitos					
Sistema de Gestão de Acesso					
Responsável: Ana Cláudia Beltrão			Início: 10/07/01		
Local: CESAR			Término: 17/07/01		
<b>1.Objetivos</b>					
1.1.Definir termos específicos do domínio da aplicação;					
1.2.Delinear as funcionalidades do módulo de Gestão de Acesso;					
1.3.Definir o que será tratado pelo módulo e o que será tratado fora do módulo;					
1.4.Definir quem e o quê irá interagir com o módulo (usuários e sistemas legados)					
1.5.Conhecer o ambiente atual do módulo (infra-estrutura e modelos de dados)					
1.6.Obter informações necessárias para realizar o desenvolvimento da interface gráfica					
<b>2.Agenda</b>					
Data	Horário	Áreas Envolvidas	Local	Participantes Operadora	Objetivo
10/07	15:00 – 18:00	Diretoria de Tecnologia, Segurança	Sala de Reunião 2	Pessoa 1 Pessoa 2 Pessoa 3	Levantamento dos requisitos funcionais, de integração e de segurança com a área técnica
12/07	08:00 – 09:00	Diretoria de Tecnologia, Segurança	Sala de Reunião 2	Pessoa 1 Pessoa 2 Pessoa 3	Validação da reunião do dia 10/07.
12/07	09:00 – 12:00	Diretoria de Tecnologia, Usuários Finais	Sala de Reunião 2	Pessoa 1 Pessoa 3 Pessoa 4	Levantamento de requisitos funcionais com usuários
12/07	16:00 – 18:00	Diretoria de Tecnologia, Designer	Sala de Reunião 2	Pessoa 1 Pessoa 5 Pessoa 6 Pessoa 7	Levantamento dos requisitos de interface.
13/07	15:00 – 18:00	Diretoria de Tecnologia, Segurança Auditoria	CESAR – Sala 1	Pessoa 1 Pessoa 8 Pessoa 9	Levantamento dos requisitos funcionais e de segurança com auditoria.
17/07	09:00 – 12:00	Diretoria de Tecnologia, Segurança	Sala de Reunião 2	Pessoa 1 Pessoa 2 Pessoa 3	Dúvidas e Pendências.
17/07	14:00 – 15:00	Diretoria de Tecnologia, Segurança Auditoria	Sala de Reunião 2	Pessoa 1 Pessoa 2 Pessoa 3 Pessoa 8 Pessoa 9	Validação da reunião com auditoria, realizada dia 13/07.
17/07	15:00 – 16:00	Diretoria de Tecnologia, Usuários Finais	Sala de Reunião 3	Pessoa 1 Pessoa 3 Pessoa 4	Validação da reunião com usuários, realizada dia 12/07.
17/07	16:00 – 18:00	Diretoria de Tecnologia, Interface	Sala de Reunião 4	Pessoa 1 Pessoa 5 Pessoa 6 Pessoa 7	Validação da reunião de interface, realizada dia 12/07.
<b>3. Requisitos de Logística para as Reuniões</b>					
3.1.Sala de Reunião;					
3.2.Documentações Técnicas e Funcionais relacionadas ao objetivo de cada reunião.					
<b>4.Observações</b>					
É muito importante que as pessoas que irão participar dos levantamentos dos requisitos:					
4.1.Tenham conhecimento do negócio, capacidade de comunicação e disponibilidade;					
4.2.Participem das próximas reuniões de validações definidas no projeto.					



A Tabela 5.2 apresenta um exemplo de pauta/convite de reunião elaborada para este projeto e enviada por e-mail a todos os possíveis participantes (Para mantermos em sigilo a identificação do cliente, demos nomes fictícios aos convidados).

O benefício alcançado com a utilização deste artefato foi uma melhoria no planejamento das atividades dos participantes e alcance de maior objetividade nas reuniões.

Tabela 5.2- Exemplo de Pauta de Reunião

<b>Pauta/convite de Reunião</b>													
<b>Levantamento de requisitos</b>													
Sistema Gestão de Acesso													
Responsável: Ana Cláudia Beltrão	Data: 12/07												
Local: CESAR	Horário: 09:00 às 12:00												
<p><b>I. Objetivo</b> Levantamento e validação de requisitos do Módulo Gestão de Acesso com usuários</p> <p><b>II.Participantes</b></p> <table style="width: 100%; border: none;"> <tr> <td style="width: 30%;"><b>Cliente</b></td> <td style="width: 30%;">Fernanda</td> <td style="width: 40%;">Analista de Sistemas</td> </tr> <tr> <td></td> <td>Clara</td> <td>Usuário visão gerencial</td> </tr> <tr> <td></td> <td>André</td> <td>Usuário visão operacional</td> </tr> <tr> <td><b>CESAR</b></td> <td>Ana Cláudia Beltrão</td> <td>Analista de Negócios</td> </tr> </table> <p><b>III.Tópicos a Serem Discutidos</b> Levantar requisitos funcionais</p> <ol style="list-style-type: none"> <li>1. Controle de Login: Características do Login, controle de expiração e criptografia de senha</li> <li>2. Controle de Perfis: Cadastro de perfis, sistemas, funções</li> <li>3. Cadastro de Usuários: Inclusão, exclusão e alteração de usuários, alteração de senha</li> <li>4. Consultas Gerenciais: Consultas e relatórios</li> </ol> <p><b>IV.Observação</b> É importante que os participantes dessa reunião levem todos os documentos existentes relativos aos tópicos acima (ex. modelos de relatórios, telas do sistema) e qualquer outra documentação relacionada.</p> <p><b>V.Próxima Reunião</b> Dia: 17/07 Horário: 15:00 às 16:00 hs Local: Cliente Objetivo: Leitura e validação da ata referente aos pontos discutidos nesta reunião</p>		<b>Cliente</b>	Fernanda	Analista de Sistemas		Clara	Usuário visão gerencial		André	Usuário visão operacional	<b>CESAR</b>	Ana Cláudia Beltrão	Analista de Negócios
<b>Cliente</b>	Fernanda	Analista de Sistemas											
	Clara	Usuário visão gerencial											
	André	Usuário visão operacional											
<b>CESAR</b>	Ana Cláudia Beltrão	Analista de Negócios											

#### 5.4.1.2. Atividade: Planejar Gerência de Requisitos

De acordo com a nossa proposta, ao planejar a Gerência de Requisitos, é necessário selecionar uma técnica para gerenciar os requisitos de forma a definir as estruturas das matrizes de rastreabilidade que devem ser utilizadas no projeto, para garantir um gerenciamento efetivo dos requisitos. Sendo assim, selecionamos a técnica proposta por Toranzo [Tor02-A] para ser aplicada no sistema Gestão de Acesso.

No planejamento da Gerência, devem ser definidas: (i) as expressões e convenções usadas nas matrizes de rastreamento, conforme apresentado na Tabela 5.3 e (ii) a estrutura da tabela para organizar as classes candidatas ao rastreamento, conforme mostra a Tabela 5.4.

Tabela 5.3- Expressões usadas nas matrizes de rastreamento

Expressões usadas nas matrizes de rastreamento	
[OBO] = Objetivo organizacional	[PRO] = Programa do sistema
[OBS] = Objetivo do sistema	[DGM] = Diagrama
[REQ] = Requisito do sistema	[PES] = Pessoa
[SUB] = Subsistema do sistema	[FMT] = Formatação bibliográfica

Tabela 5.4- Tabela para organizar as classes candidatas ao rastreamento

Níveis de informação	Análise	Projeto	Implementação	Teste
Ambiental				
Organizacional				
Gerencial				
Desenvolvimento				

A Tabela 5.5 apresenta o Plano de Gerência de Requisitos proposto para o Sistema.

Tabela 5.5 - Plano de Gerência de Requisitos

<p><b>1. Introdução</b> Este documento apresenta as diretrizes usadas para Gerenciar os Requisitos do Sistema Gestão de Acessos.</p> <p><b>2. Gerenciamento de Requisitos</b> A responsabilidade pela gerência de requisitos está distribuída entre o Gerente de Projetos e os Engenheiros de Requisitos do projeto. As mudanças nos requisitos devem ser realizadas mediante a submissão e análise de Solicitações de Mudanças. O rastreamento de requisitos será realizado mediante a utilização da técnica proposta por Toranzo [Tor02-A].</p> <p><b>3. Identificação dos Requisitos</b> Os requisitos têm identificação única, onde é inicialmente identificado o pacote a que pertence, seguido do número do requisito e seu nome. Os casos de usp também têm identificação única, seguindo o mesmo padrão: identificação do pacote e número do caso de uso.</p> <p><b>4. Rastreabilidade</b> O rastreamento de requisitos será realizado mediante a utilização da técnica proposta por Toranzo [Tor02-A]. Expressões Usadas nas Matrizes de Rastreamento:            [OBO] = Objetivo organizacional                    [PRO] = Programa do sistema            [OBS] = Objetivo do sistema                    [DGM] = Diagrama            [REQ] = Requisito do sistema                    [PES] = Pessoa            [SUB] = Subsistema do sistema                    [UC] = Caso de Uso</p> <p><b>5. Gerenciamento das Mudanças dos Requisitos</b> As solicitações de mudanças devem ser realizadas mediante a utilização de formulários padronizados para a Solicitação de Mudança (SM), onde devem ser apresentados os documentos, requisitos e/ou casos de uso que precisam da alteração. Os responsáveis pela gerência de requisitos analisam a solicitação e, estimam a mudança de acordo com suas necessidades e impactos. Após a aprovação e validação da mudança, a SM é executada.</p> <p><b>6. Treinamentos e Recursos</b> É necessário treinamento na utilização da técnica de rastreamento de Toranzo.</p>
--

### 5.4.1.3. Atividade: Planejar Validação de Requisitos

Ao planejar a validação de Requisitos, é identificada a técnica de validação que deve ser empregada no projeto e planejadas as reuniões de validação. No exemplo em análise, foram elaboradas agendas de reuniões, as quais foram enviadas, junto com pautas convites, aos

possíveis participantes. O planejamento das reuniões de validação foi realizado de forma que, em cada reunião, um ou mais pacotes de casos de uso correlacionados deveriam ser validados.

Em relação aos convidados para as reuniões de validação dos requisitos, foram selecionados aqueles indivíduos que poderiam contribuir para que o objetivo da reunião fosse atingido. Desta forma, para a validação dos casos de uso mais críticos da aplicação, que envolviam questões relacionadas à política de acesso e segurança da aplicação, foram convidados um analista de sistemas da organização, líder de projeto e responsável pelo legado; um analista de segurança de sistemas; um auditor de sistemas; e um membro do comitê executivo, conhecedor do domínio da aplicação e da área de segurança de informações. Para a validação dos casos de uso relacionados às operações a serem realizadas nos cadastros de usuários, o convite foi direcionado ao analista de sistemas da organização, líder de projeto e responsável pelo legado; ao analista de segurança de sistemas e a um gestor de sistemas. Quanto ao pacote de casos de uso referentes à trilha de auditoria, foram convidados o analista de sistemas da organização, líder de projeto e responsável pelo legado, e um auditor de sistemas.

De acordo com o planejamento inicial, as reuniões deveriam ser realizadas em dois dias, mas na prática foram necessários três dias para a conclusão da atividade.

Como as atas e pautas seguiram os mesmos modelos apresentados na atividade *Planejar Elicitação de Requisitos*, subseção 5.4.1.1 não repetiremos aqui, tabelas similares àquelas já apresentadas.

#### 5.4.1.4. Atividade: Definir a Visão do Sistema

O objetivo desta atividade é produzir o documento de visão da aplicação. Na Tabela 5.6 apresentamos o Documento de Visão proposto para o projeto, conforme o template apresentado no capítulo 5.

*Tabela 5.6 - Documento de Visão*

<p><b>1.Introdução</b></p> <p>Este documento apresenta uma proposta de prestação de serviços para a Operadora de Cartão, e deverá servir como o instrumento base para o acompanhamento do serviço prestado.</p> <p>O serviço aqui especificado compreende o desenvolvimento de um Sistema de Gestão de Acessos, único e centralizado que poderá ser utilizado por qualquer aplicação disponível em ambiente Web da Operadora de Cartão de Crédito.</p> <p><b>2. Posicionamento</b></p> <p>O Sistema de Gestão de Acessos será desenvolvido com o objetivo de centralizar e permitir a definição de uma política única de acesso para os módulos computacionais que serão desenvolvidos sobre o ambiente Web.</p> <p>Para que as aplicações da Operadora possam utilizar o componente de controle de acesso, este irá publicar a sua interface, onde serão definidas todas as funcionalidades que o mesmo poderá prover.</p> <p>Com a utilização deste módulo, nenhuma aplicação precisará desenvolver mecanismos de segurança de acesso, havendo economia de custo e prazo. Além disso, será utilizado um cadastro de único de usuários e estabelecidas permissões de acesso a funções e</p>
--

dados, garantindo maior segurança, inclusive contra hackers no ambiente web.

### **3.Descrição dos Stakeholders e Usuários**

Os usuários da aplicação são todos os usuários dos sistemas sobre plataforma web. Assim, os funcionários da organização, os terceiros e prestadores de serviços, lojistas, parceiros e clientes são usuários candidatos. À medida que novas aplicações são desenvolvidas e implantadas, aumenta o leque de usuários acessando a aplicação, já que qualquer usuário, passa, primeiramente, pelo sistema de gestão de acessos e todo e qualquer software para ambiente web deve ser integrado com o mesmo.

Para o sistema de gestão de acesso, os seguintes papéis são definidos:

a)Gestor de Acesso: Usuário responsável pela administração de acesso de cada sistema da aplicação, sendo responsável pelo cadastro de perfis, associação de perfis, liberação e bloqueio de acesso aos usuários.

b)Administrador de Segurança: Usuário responsável pela segurança geral do sistema, definindo os parâmetros da política de segurança adotada. A cada novo sistema a ser implantado, o administrador de segurança deve avaliar as permissões de acesso liberadas e especificar as funcionalidades e atributos de elementos de dados que devem ser logados.

c)Configurador de Aplicações: Estes usuários correspondem aos engenheiros de software e analistas de negócios responsáveis pela implantação de um novo sistema.

d)Usuário final: Estes são a grande maioria dos usuários, podendo ser diversos funcionários, clientes, lojistas, conveniados e parceiros.

### **4.Visão Geral do Produto**

Existem cinco módulos para o Sistema de Gestão de Acessos:

4.1. Controle de Acesso: o sistema deverá prover uma base de validação de acesso única para todos os sistemas a serem desenvolvidos sobre o ambiente Web.

As seguintes políticas de identificação e acesso dos usuários serão atendidas:

a)Todo login de acesso deverá estar associado a um perfil pré-definido pelos gestores dos sistemas, de acordo com as atribuições de cada usuário;

b)O sistema terá um mecanismo para definição de: período de expiração de senhas, quantidade de acessos errados até o bloqueio e controle de senhas repetidas;

c)As senhas de acesso serão armazenadas criptografadas

4.2.Controle de Perfis: o acesso dos usuários ao sistema deverá ser realizado através de um controle de perfil, a fim de permitir o acesso apenas às funções determinadas. O controle de perfil deverá também permitir a definição de quais aplicações e dados o usuário poderá acessar.

O controle de perfil possuirá as seguintes características:

a)Cada usuário cadastrado terá um conjunto de perfis associados.

b)A um perfil de acesso poderão ser associadas diversas funcionalidades de cada sistema;

4.3.Manutenção de Usuários: permitirá a manutenção das informações dos usuários habilitados a utilizar o sistema, ou seja, operações de inclusão, alteração, exclusão de usuários e alteração de senhas.

4.4.Consultas Gerenciais: O sistema disponibilizará consultas gerenciais, como: relação de usuários ordenados por nome; relação de usuários ordenados por perfil; relação de alterações de perfis de usuários; e outras a serem definidas

4.5.Trilha de Auditoria: para permitir a auditoria dos registros de acessos

### **5. Restrições**

5.1. O ambiente computacional deve considerar as seguintes premissas: (i)o sistema deverá estar disponível 7 dias por semana, 24 horas por dia; (ii)O sistema deverá suportar, inicialmente, uma média de 300 usuários/dia acessando o sistema simultaneamente;

5.2. A seguir apresentamos o que não será contemplado nesta proposta, conforme acordado com a equipe técnica da Operadora: (i) Não é escopo do projeto desenvolver sistemas multilíngue e multi-moeda; (ii) Os módulos a serem desenvolvidos não contemplam a implementação de rotinas para efetuar backup e restore de arquivos-texto armazenados em disco. Estas são responsabilidade da equipe de produção do cliente; (iii) Este projeto não contempla o escalonamento e ativação das rotinas batch. Estas são responsabilidade da equipe de produção do cliente;

Desta forma, concluímos o subfluxo *Planejar a Engenharia de Requisitos*. A Próxima seção apresenta a aplicação do Subfluxo *Elicitar Requisitos* e a execução de suas atividades.

## **5.4.2. SUBFLUXO: ELICITAR REQUISITOS**

Ao elicitar os requisitos do sistema Gestão de Acessos, executamos as seguintes atividades: *Descobrir Requisitos*, *Descobrir Requisitos de Interface*, *Capturar o Vocabulário do Domínio*, *Encontrar Atores e Casos de Uso*, *Encontrar Requisitos Não-Funcionais* e *Gerenciar Dependências*.

As subseções seguintes detalham a execução de cada uma destas atividades.

### **5.4.2.1. Atividade: Descobrir Requisitos**

Como a principal técnica de elicitação utilizada no projeto foi através da realização de reuniões e entrevistas com clientes, usuários e stakeholders, a execução desta atividade seguiu as agendas e pautas de reuniões programadas no planejamento da elicitação de requisitos. Todas as reuniões foram documentadas através da produção de atas de reuniões. Após a produção das atas, as mesmas eram enviadas, por e-mail, a todos os participantes para que as atas fossem validadas. Para todas as reuniões, após a redação das atas e distribuição das mesmas entre os participantes e pessoas interessadas, havia uma reunião de aproximadamente uma hora, para a validação da ata redigida, com as mesmas pessoas que haviam participado da discussão anterior.

Outra técnica também utilizada foi a análise do módulo de controle de acesso do Sistema Legado, onde foi produzida uma documentação com as telas do sistema e a descrição dos campos de cada tela.

Um exemplo de ata de reunião redigida no projeto é apresentada na Tabela 5.7. Neste trabalho, foi mantido em sigilo o nome do cliente e foram dados nomes fictícios aos participantes da reunião.

Tabela 5.7 - Exemplo de ata de registro de reunião

ATA DE REUNIÕES	
Levantamento de Requisitos	
Projeto Gestão de Acesso	
<b>REDATOR: ANA CLÁUDIA BELTRÃO</b>	<b>INÍCIO: 10/07</b>
<b>LOCAL: CESAR</b>	<b>TÉRMINO: 17/07</b>
<b>I.Objetivo</b>	
O objetivo da reunião foi efetuar o levantamento de requisitos das funcionalidades do Módulo de Gestão de Acesso.	
<b>II.Participantes</b>	
<b>Cliente</b>	Fernanda Júlia Clara
<b>CESAR</b>	Ana Cláudia Ana Sofia
	Analista de Sistemas Engenheiro de Segurança Gestor do Projeto  Analista de Negócios Gerente de Projeto
<b>III.Tópicos discutidos/definições</b>	
<b>Funcionalidades do sistema legado</b>	
<ul style="list-style-type: none"> <li>• O controle de acesso é individual, através de login e senha, sendo utilizado por funcionários e terceiros.</li> <li>• Quando um usuário é criado, o mesmo é associado a um perfil. Cada perfil libera o acesso a um conjunto de funções associadas ao sistema, de acordo com as atribuições de cada usuário. O usuário só terá acesso aos recursos disponibilizados para o seu perfil, na forma de consulta, edição, gravação, atualização e exclusão de dados. Uma pessoa só possui um perfil.</li> <li>• Existem acessos especiais a funções que são determinados por perfil. Por exemplo, um perfil que tem acesso especial pode cadastrar uma taxa de administração diferenciada.</li> <li>• Os perfis são criados pelo Administrador. Somente o administrador pode criar / manter perfil, associar perfil a um usuário.</li> </ul>	
<b>IV. Expectativas para o Sistema Gestão de Acesso</b>	
<b>Controle de Login / Segurança de acesso</b>	
<ul style="list-style-type: none"> <li>• O login deve ser um conjunto de caracteres. O usuário deve informar seu nome completo e o login será gerado automaticamente obedecendo uma regra de formação.</li> <li>• O sistema gerará automaticamente as primeiras senhas de acesso de forma randômica;</li> <li>• A senha de acesso deve obedecer a uma regra de formação.</li> <li>• As senhas de acesso serão armazenadas criptografadas.</li> <li>• Será disponibilizada uma funcionalidade que permita a troca de senhas para qualquer usuário cadastrado.</li> <li>• As senhas devem ser substituídas, no máximo, a cada 40 dias de utilização..</li> <li>• Na terceira tentativa errada de digitação de uma senha, o login deve ser bloqueado.</li> <li>• Se o usuário passar 40 dias sem acessar o sistema, o mesmo terá o seu acesso ao sistema bloqueado automaticamente.</li> <li>• O desbloqueio só poderá ser feito pelo gestor do sistema e será atribuída uma nova senha descartável ao usuário.</li> <li>• O controle de acesso não deve permitir acesso simultâneo de um mesmo usuário em mais de uma estação de trabalho.</li> </ul>	
<b>Controle de Perfis</b>	
<ul style="list-style-type: none"> <li>• Deverá ser mantido cadastro de sistemas, funções de sistemas e operações (ou eventos), bem como cadastros de usuários.</li> <li>• O cadastro de perfis de acesso define um grupo de usuários com características de acesso semelhantes. A segurança de acesso será implementada através do relacionamento entre os usuários e os perfis.</li> <li>• Existe uma hierarquia de perfil, semelhante a uma estrutura em árvore. Um perfil filho só pode acessar um subconjunto de funcionalidades do perfil pai.</li> <li>• Cada sistema terá um gestor de acesso. Um usuário poderá ter um ou mais perfil, sendo um por cada sistema.</li> </ul>	
<b>Logs de Acesso</b>	
<ul style="list-style-type: none"> <li>• Todo acesso aos sistemas deve ser registrado e guardado por um período de 6 meses.</li> </ul>	
<b>V.Próximas Ações</b>	
<ul style="list-style-type: none"> <li>• Análise da apresentação do Menu</li> </ul>	
Responsável: Equipe do CESAR	
Data de entrega: 30/07 (reunião de validação de requisitos)	
<b>VI.Próxima Reunião</b>	
Dia: 12/07 às 8:00H	
Local: sala 1 CESAR	

#### **5.4.2.2. Atividade: Descobrir Requisitos de Interface**

Como a aplicação é direcionada para o ambiente Web e sendo o público alvo, a longo prazo, bastante heterogêneo, os requisitos de interface têm importância elevada.

A elicitação para os requisitos de interface foi realizada através da aplicação de questionários aos *stakeholders* e alguns usuários finais; da realização de reuniões, conforme planejado nas agendas e pautas de reuniões; da análise da interface do legado e de algumas aplicações similares.

*Foi aplicado um Questionário de Briefing de Design, conforme apresentado na*

Tabela 4.9 da seção 4.2.2.2. Após reunião com stakeholders e usuários finais, foi redigida a ata de reunião apresentada na Tabela 5.8 (Como o cliente é mantido em sigilo, utilizamos nomes fictícios para os participantes da reunião).

Tabela 5.8-Ata de Reunião de Elicitação de Requisitos de Interface

ATA DE REUNIÕES		
Levantamento de Requisitos de Interface		
Projeto: Gestão de Acesso		
<b>REDATOR: ANA CLÁUDIA BELTRÃO</b>	<b>DATA: 12/07</b>	
<b>LOCAL: CESAR</b>	<b>HORÁRIO: 16:15 AS 18:10</b>	
<b>I.Objetivo</b>		
Levantamento de requisitos para construção de Interface		
<b>II.Participantes</b>		
Cliente	João Clara Pedro	Comitê Executivo Gestor do Projeto Suporte
CESAR	Eliana Ana Cláudia	Designer Engenheira de Requisitos
<b>III.Tópicos discutidos/definições</b>		
<ul style="list-style-type: none"> <li>• Requisitos para a Camada de Apresentação <ul style="list-style-type: none"> <li>▪ Foi colocado pelo cliente como requisito desejável que a camada de apresentação suportasse a geração de conteúdo dinâmico baseado em perfis do usuário/empresa;</li> <li>▪ Estes perfis de usuário/empresa devem ser independentes dos perfis de usuário do Sistema de Gestão de Acesso. Associado a cada usuário deve existir um perfil de interface específico, por exemplo, a apresentação do sistema para um lojista difere da apresentação para um funcionário interno e para um cliente. Isto significa que a camada de apresentação, ao reconhecer o usuário logado, irá apresentar a interface própria definida e associada ao usuário.</li> </ul> </li> <li>• Definição dos requisitos para Interface do usuário: <ul style="list-style-type: none"> <li><b>Objetivos do Sistema</b> <ul style="list-style-type: none"> <li>▪ Objetivo geral: Suportar todas as operações da operadora, considerando o máximo de flexibilidade na identidade visual do sistema;</li> <li>▪ Objetivos Específicos: em curto prazo: atingir os diversos setores da Operadora (Contabilidade, Serviços, Lojistas e Crédito). A médio e longo prazo: estender aos outros setores e usuários externos, incluindo lojistas e clientes.</li> </ul> </li> <li><b>Padrões Adotados</b> <ul style="list-style-type: none"> <li>▪ Navegador Internet Explorer em plataforma PC/OS Windows. Melhor Visualizado em 800x600 pixels, mínimo de 256 cores;</li> <li>▪ Provável uso de Adobe Acrobat;</li> <li>▪ Padronização de botões e na Arquitetura Navegacional;</li> <li>▪ Desconsiderar o uso de frames;</li> <li>▪ Uso de templates de páginas;</li> </ul> </li> <li><b>Usabilidade</b> <ul style="list-style-type: none"> <li>▪ Público do Sistema: Em um primeiro momento apenas funcionários. Posteriormente deverá ser aberto a usuários externos tendo acesso apenas a opções de acordo com o perfil específico.</li> <li>▪ Características de comportamento do público alvo: funcionários de todos os níveis gerenciais e operacionais, indo desde a vice-presidência até funcionários do Callcenter.</li> <li>▪ Assiduidade de uso do sistema: na grande maioria serão telas de trabalho de oito horas de uso diárias, o que implica em uma maior atenção na facilidade de uso.</li> </ul> </li> <li><b>Identidade Visual</b> <ul style="list-style-type: none"> <li>▪ As cores devem seguir o padrão da logomarca da Operadora;</li> <li>▪ A identificação visual deverá seguir o padrão da Operadora;</li> <li>▪ A necessidade de associação entre o sistema e a logomarca, foi considerado importante;</li> <li>▪ A autonomia visual do Site em relação à comunicação visual da empresa deverá ser considerada;</li> <li>▪ O Sistema não possuirá logotipo próprio;</li> <li>▪ Não deverá ser usado padrão de cores para diferenciar os diversos módulos do sistema;</li> <li>▪ Considerar as limitações de cores para daltônicos.</li> </ul> </li> <li><b>Interface</b> <ul style="list-style-type: none"> <li>▪ Serviços on-line de boa qualidade visual: <a href="http://www.bowstreet.com">www.bowstreet.com</a>, <a href="http://www.bb.com.br">www.bb.com.br</a>, <a href="http://www.peapod.com">www.peapod.com</a></li> </ul> </li> </ul> </li> <li>• Expectativas: leveza (visual clean), padronização do sistema, facilidade de uso e rapidez do sistema, na primeira página deverá haver espaço para promoções e notícias.</li> </ul>		
<b>IV.Próximas Ações</b>		
1.Apresentação da interface do sistema atual		
2.Envio do draft de interface da Operadora		
<b>V.Próxima Reunião</b>		
17/07 às 16:00 no CESAR.		



### 5.4.2.3. Atividade: Capturar o Vocabulário do Domínio

Para clarificar o domínio da aplicação de forma satisfatória e auxiliar à produção dos artefatos ligados aos modelos funcionais e não-funcionais da aplicação, desenvolvemos o *Glossário Léxico*, artefato sugerido em nossa proposta e baseado no LAL (Léxico Ampliado da Linguagem, de Leite e Cysneiros [Kap00] [Cys01-C]). Este glossário foi produzido de acordo com a base de informações coletadas nas reuniões de elicitação de requisitos, descritas nas seções 5.4.2.1 e 5.4.2.2.

A Tabela 5.9 apresenta o glossário léxico do sistema Gestão de Acesso.

Tabela 5.9- Glossário Léxico do Sistema Gestão de Acesso

Símbolo	Tipo	Definições	Impactos	RNF's Associados
Administrador de segurança	Sujeito	Pessoa responsável pela administração e segurança de todos os sistemas da organização.	1.Os administradores de segurança cadastram novos <u>sistemas</u> , definem as <u>políticas de segurança</u> da organização e cadastram os <u>perfis</u> dos <u>gestores de acesso</u>	
Aprovar Cadastro de Usuário	Ação	Após ser cadastrado no sistema, o <u>usuário</u> deve ter seu cadastro aprovado pelo <u>gestor de acesso</u> .	1. O <u>usuário</u> só pode ter seu acesso desbloqueado se o mesmo tiver sido aprovado pelo <u>gestor de acesso</u> .	SEGURANÇA
Associado	Sujeito	Funcionário ou estagiário de alguma empresa do grupo.	1.Correspondem à grande maioria de <u>usuários</u> finais da aplicação.	
Bloquear Usuário	Ação	Impedir que um <u>usuário</u> acesse o sistema.	1. Sob determinadas condições, alguns <u>usuários</u> devem ser bloqueados, por exemplo: após um determinado número de tentativas de acesso inválidas, determinado período sem acessar o sistema, etc. 2. Todo <u>usuário</u> , ao ser cadastrado, deve ser inserido "Bloqueado"	SEGURANÇA
Criptografia	Ação	Escrita secreta por meio de abreviaturas ou de sinais convencionados de modo a preservar a confidencialidade da informação. Usada, por exemplo, na armazenagem de senhas de acesso e na transmissão de dados entre plataformas distintas.	1.As senhas de acesso devem ser armazenadas criptografadas e a transmissão de dados com os sistemas legados também deve ser criptografada.	SEGURANÇA
Configurar Política de Segurança	Ação	Configurar os critérios definidos na <u>Política de Segurança</u>	1. Apenas os <u>Administradores de Segurança</u> , devem configurar os parâmetros da <u>Política de Segurança</u> .	SEGURANÇA USABILIDADE (O sistema deve ser facilmente configurável)
Desativar Usuário	Ação	Escluir, logicamente, um usuário do sistema.	1. Após ser desativado, o usuário não poderá mais acessar o sistema. 2. O usuário não é excluído fisicamente da base de dados para não perder informações históricas.	SEGURANÇA RECUPERABILIDADE

Desbloquear Usuário	Ação	Liberar um determinado <u>usuário</u> para acessar o sistema.	1. Apenas o gestor de acesso deve poder desbloquear um <u>usuário</u> . 2. Ao desbloquear um <u>usuário</u> , deve ser gerada uma <u>senha descartável</u> para o <u>usuário</u> .	
Elemento de dado	Objeto	Corresponde a um dado do sistema normalmente representado por um objeto e seus atributos.	1.O sistema deve manter um cadastro dos elementos de dados manipulados pelas <u>funções</u> de cada <u>sistema</u>	SEGURANÇA
Funções	Objeto	Corresponde às funcionalidades que podem ser executadas pelos sistemas da organização.	1.O sistema deve manter um cadastro das funções executáveis por cada sistema. 2.No cadastro das funções deve indicar se a função é logável ou não, para que sua execução seja registrada nos <u>logs de transações</u> . 3.As funções podem ser on-line ou batch.	SEGURANÇA
Gestor de acesso	Sujeito	Pessoa, ou equipe de pessoas, responsáveis pela gestão de determinado sistema da organização.	1.Os gestores de acesso são responsáveis pela criação de <u>perfis de acesso</u> e associação destes aos usuários da aplicação. 2. A liberação do acesso de um <u>usuário</u> , também é de responsabilidade do gestor. 3.Um gestor de acesso, cuja posição hierárquica seja superior aos demais, poderá associar um <u>perfil</u> a si próprio, os demais não.	
Log	Objeto	Armazenamento de informações relativas a manipulação da aplicação pelos <u>usuários</u> , visando garantir o rastreamento de informações, quando necessário.	1.As ações executadas durante o acesso dos <u>usuários</u> aos sistemas devem ser registradas. 2. O sistema deve prover mecanismos de consultas aos logs registrados. 3. A limpeza dos logs deve ser realizada automaticamente, de acordo com as configurações definidas para o sistema.	SEGURANÇA
Logar-se	Ação	Ação executada pelo <u>usuário</u> ao identificar-se, através de seu <u>login</u> , para iniciar uma sessão de trabalho.	1.Sempre que algum <u>usuário</u> tentar efetuar o <u>login</u> , o sistema deverá validar seu acesso, caso o <u>usuário</u> erre a senha numa quantidade maior que o máximo de erros permitidos, o mesmo é bloqueado. 2.Caso o <u>usuário</u> esteja <u>logado</u> em outra estação, o sistema também não permite o login. 3.Embora muitas validações sejam necessárias, é fundamental que o tempo de resposta desta funcionalidade seja satisfatório.	SEGURANÇA USABILIDADE PERFORMANCE
Login	Objeto	Identificação do <u>usuário</u> para acessar os sistemas da organização. Ao acessar o sistema, o usuário abre uma sessão de trabalho.	1.Cada <u>usuário</u> deve ter um login único de acesso.	
Logout	Ação	Encerramento da sessão de trabalho do usuário.	1.Sempre que a sessão do usuário for encerrada, o sistema deverá registrar a data e hora de encerramento no <u>log de acesso</u>	SEGURANÇA

			gravado. 2. Caso a sessão do usuário permaneça ociosa por um determinado período de tempo, a mesma é encerrada através de um logout automático.	
Montar Menu	Ação	Apresentar os <u>sistemas, subsistema e funções</u> de acordo com as permissões de acesso do <u>usuário</u> logado.	1. O sistema deve montar dinamicamente o menu do usuário, onde só deve ser apresentado o que o <u>usuário</u> tenha permissão de acesso.	USABILIDADE
Perfil de acesso	Objeto	Características gerais de acesso de um grupo de <u>usuários</u> às funcionalidades de cada sistema da organização.	1. Um <u>usuário</u> poderá ter um conjunto de perfis de acesso. 2. Cada perfil pode ter um conjunto de usuários associados a ele. 3. Cada sistema poderá ter diversos perfis. 4. Cada perfil estará associado a apenas um sistema. 5. É fundamental que a manutenção do cadastro de perfis tenha interface amigável e seja fácil de usar. 6. Apenas os <u>gestores de acesso e administradores de segurança</u> deverão manipular os perfis de acesso.	SEGURANÇA USABILIDADE
Política de Segurança	Objeto	Conjunto de critérios e procedimentos que devem ser adotados por todos os usuários dos sistemas	1. Os <u>administradores de segurança</u> definem os parâmetros da política de segurança que devem ser aplicados a todos os <u>usuários</u> .	SEGURANÇA
Restrição de acesso	Objeto	Conjunto de permissões de acesso atribuídas a um perfil para um <u>elemento de dados</u>	1. A permissão de acesso configurada nos <u>perfis</u> dos <u>usuários</u> deve estabelecer permissões de leitura, inclusão, alteração e exclusão dos elementos de dados manipulados pelas funcionalidades associadas aos <u>perfis</u> . 2. Para os atributos dos <u>elementos de dados</u> , também devem ser configuradas as permissões para leitura e alteração das informações de cada atributo.	
Senha	Objeto	Conjunto de caracteres secretos que autorizam o acesso de um determinado <u>usuário</u> ao sistema.	1. As senhas devem ser criptografadas. 2. As senhas de acesso devem ser trocadas periodicamente pelos usuários.	SEGURANÇA
Senha Descartável	Objeto	<u>Senha</u> de acesso gerada randomicamente pelo sistema, para um determinado <u>usuário</u> , que deve ser trocada no primeiro acesso do mesmo.	1. A senha descartável deve ser enviada ao usuário de forma sigilosa e segura. 2. É fundamental que a senha descartável seja trocada pelo usuário no primeiro acesso ao sistema.	
Sistema	Objeto	Sistemas computacionais da organização.	1. Deve ser mantido um cadastro de sistemas. 2. Cada sistema deverá ter um	

			conjunto de <u>funções</u> ou <u>subsistemas</u> associados. 3.Cada sistema deverá ter um ou mais <u>gestores de acesso</u> .	
Subsistema	Objeto	Módulos dos Sistemas computacionais da organização.	1. Deve ser mantido um cadastro de subsistemas. 2. Cada subsistema deverá ter um conjunto de <u>funções</u> associadas.	
Trilha de auditoria	Objeto	Consultas extraídas a partir dos registros de <u>log de acesso</u> e/ou <u>log de transações</u> que permitem identificar como e quando determinadas ações ou situações ocorreram nos sistemas. Garantem a rastreabilidade de informações.	As consultas às informações logadas devem ser fáceis de manipular	SEGURANÇA USABILIDADE
Usuários	Sujeito	Pessoas que operam os sistemas da organização.	1. O sistema deve prover funcionalidades para a administração dos usuários cadastrados. 2. O sistema deve manter um cadastro dos usuários.	
Usuário Logado	Sujeito	<u>Usuário</u> com sessão de trabalho aberta no sistema.	Um <u>usuário</u> não poderá <u>logar-se</u> mais de uma vez, ao mesmo tempo, em estações de trabalho diferentes.	

Com a utilização do *Glossário Léxico* observamos que com a definição dos símbolos e dos impactos associados, novos símbolos e impactos passaram a serem encontrados, tornando mais fácil à descoberta e o conhecimento do domínio da aplicação. Desta forma, passamos a ter subsídeos que permitem um maior conhecimento do domínio da aplicação e a identificação de algumas regras de negócio associadas com cada símbolo, além disso, ele também permite a identificação de alguns requisitos não-funcionais associados aos símbolos identificados.

#### 5.4.2.4. Atividade: Encontrar Atores e Casos de Uso

Para executar esta atividade foram utilizadas as informações registradas no *Glossário Léxico* e nas atas de reuniões. Segundo a nossa proposta, a utilização do *Glossário Léxico* auxilia a identificar os atores e casos de uso, pois os símbolos classificados como “sujeitos” são candidatos a serem atores da aplicação e as “ações”, casos de uso. Como exemplo prático: observando na Tabela 5.9 o símbolo “Logar-se”, classificado com ação, podemos identificar o caso de uso “Efetuar Login”. Já os símbolos “Administrador de Segurança” e “Gestor de Acesso”, ajudam a identificar os atores de mesmo nome.

Além disto, observando as definições e os impactos de cada símbolo, podemos identificar funcionalidades que devem ser providas pelo sistema. Por exemplo, para o símbolo “Administrador de Segurança” (Vide Tabela 5.9), a coluna impacto diz que “Os administradores

de segurança cadastram novos sistemas, definem políticas de segurança da organização e cadastram os perfis dos gestores de acesso”, analisando esta frase identificamos 3 possíveis casos de uso: *Cadastrar Sistemas, Configurar Política de Segurança e Cadastrar Perfis*. Outro exemplo é a identificação dos casos de uso *Cadastrar Sistemas, Cadastrar Subsistemas e Cadastrar Funções* através da análise dos símbolos *Sistemas, Subsistemas e Funções*.

Como o projeto é muito grande, os diagramas de casos de uso foram organizados em pacotes, de acordo com as áreas de negócios que estavam associados: *Configuração do Sistema Gestão de Acesso, Gestão de Usuários, Controle de Acesso e Trilha de Auditoria*. Nesta seção apresentamos o modelo preliminar de casos de uso subdivididos entre os pacotes citados.

- **Configuração do Sistema Gestão de Acesso:** Os casos de uso deste pacote representam as funcionalidades para a configuração de políticas de segurança; cadastramento de sistemas, subsistemas e funções; e configuração de perfis de acesso que serão aplicadas na gestão de acessos. A Figura 5.1 apresenta o modelo preliminar deste pacote, onde foram identificados os atores *Gestor de Acesso* e *Usuário Geral*, que executam os seguintes casos de uso: *Configurar Política de Segurança, Cadastrar Sistemas, Cadastrar Subsistemas, Cadastrar Perfil de Acesso e Cadastrar Elementos de Dados*.

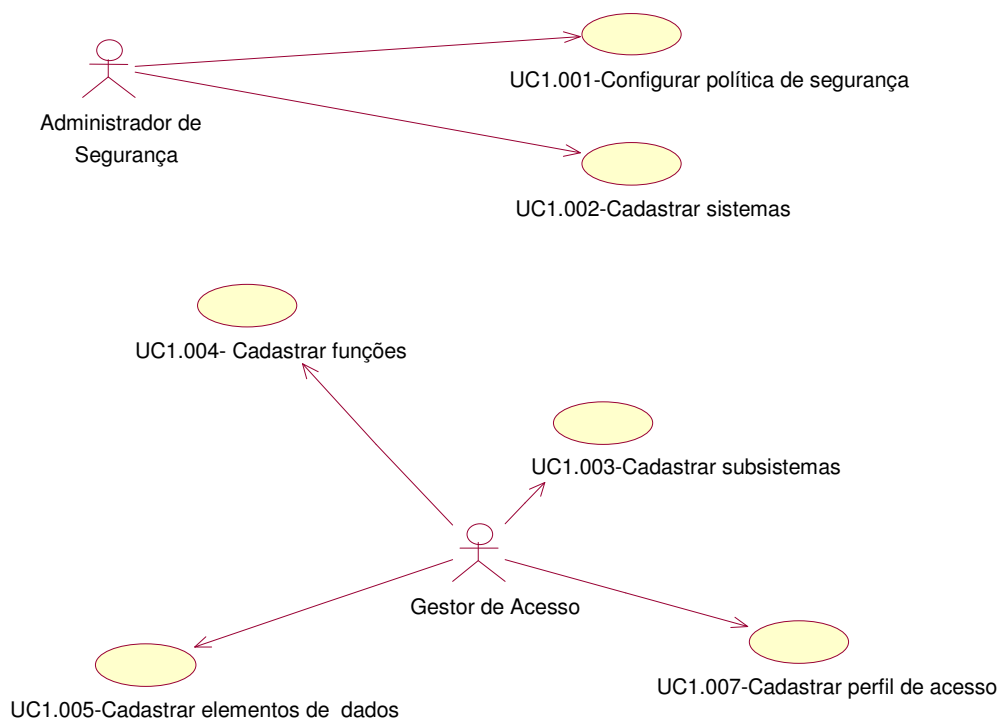


Figura 5.1 -Diagrama de Casos de Uso Preliminar - Pacote Configuração do Sistema Gestão de Acesso

- **Gestão de Usuários:** Os casos de uso deste pacote representam as funcionalidades para cadastramento, administração e gerência de usuários. A Figura 5.2 apresenta o diagrama de casos de uso inicial para este pacote.

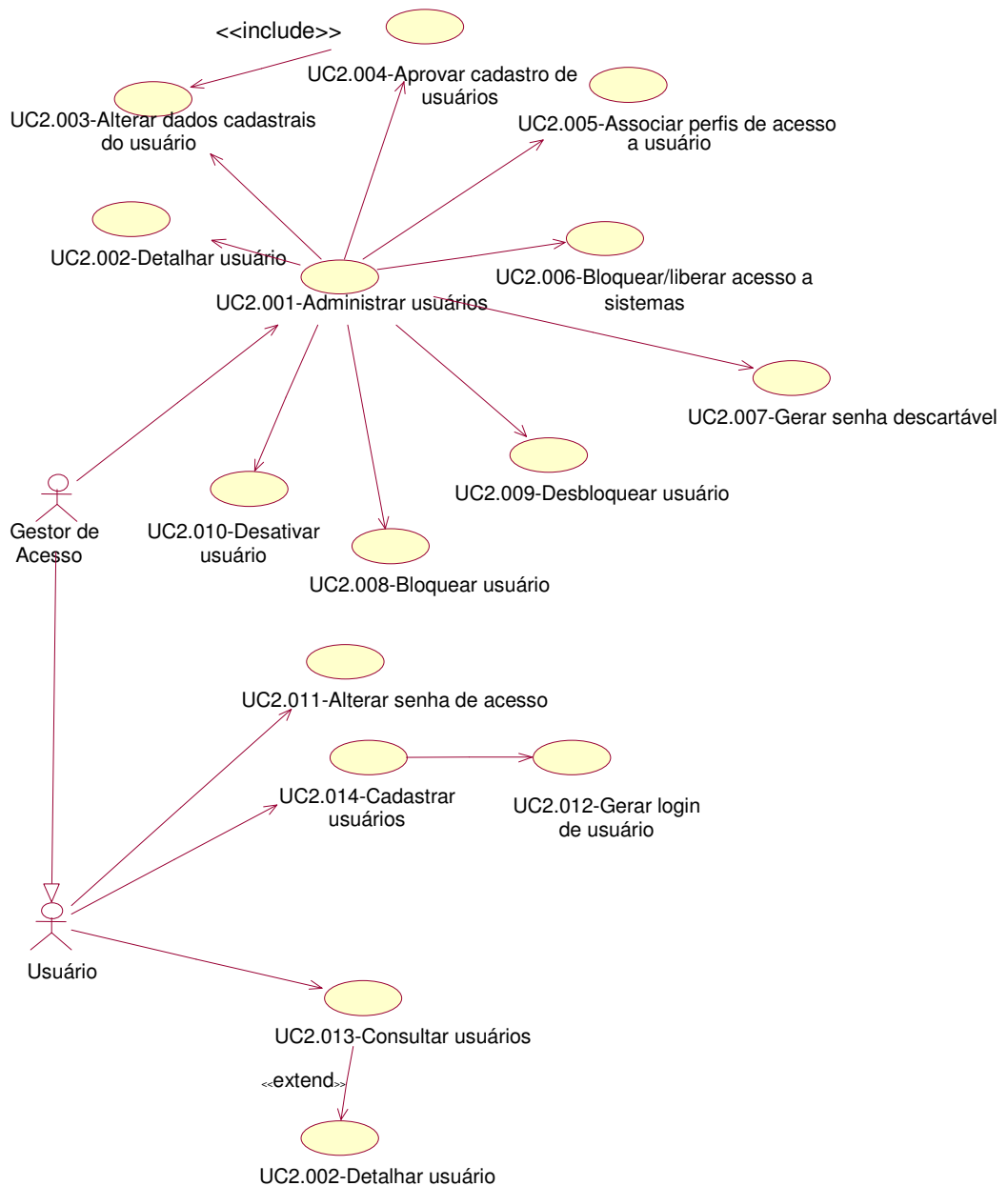


Figura 5.2- Diagrama de Casos de Uso Preliminar - Pacote Gestão de usuários

Observando a Figura 5.2 verificamos que foram identificados os atores *Gestor de Acesso* e *Usuário Geral*. Estes atores executam os casos de uso: *Administrar Usuários*, *Detalhar Usuário*, *Alterar Dados Cadastrais do Usuário*, *Aprovar Cadastro de Usuários*, *Associar*

*Perfis de Acesso a Usuários, Bloquear/Liberar Acesso a Sistemas, Gerar Senha Descartável, Bloquear Usuário, Desbloquear Usuário, Desativar Usuário, Alterar Senha, Gerar Login de Usuário, Consultar Usuários.*

- **Controle de Acesso:** Os casos de uso deste pacote representam as funcionalidades que implementarão o controle de acesso ao sistema, tais como login/logout e montar telas de menus dinâmicos. A Figura 5.3 apresenta o diagrama de casos de uso inicial para este pacote, onde foram identificados os atores *Usuário Geral* e *Funções de Sistemas*. O usuário geral executa os casos de uso *Efetuar Login*, *Encerrar Sessão* e *Selecionar Opção do Menu*. Os casos de uso *Montar Menu do Sistema* e *Registrar Logs de Acesso* são executados após o login ser efetuado. Ao ser executado o caso de uso *Selecionar Opção do Menu*, é chamada a função correspondente no sistema.

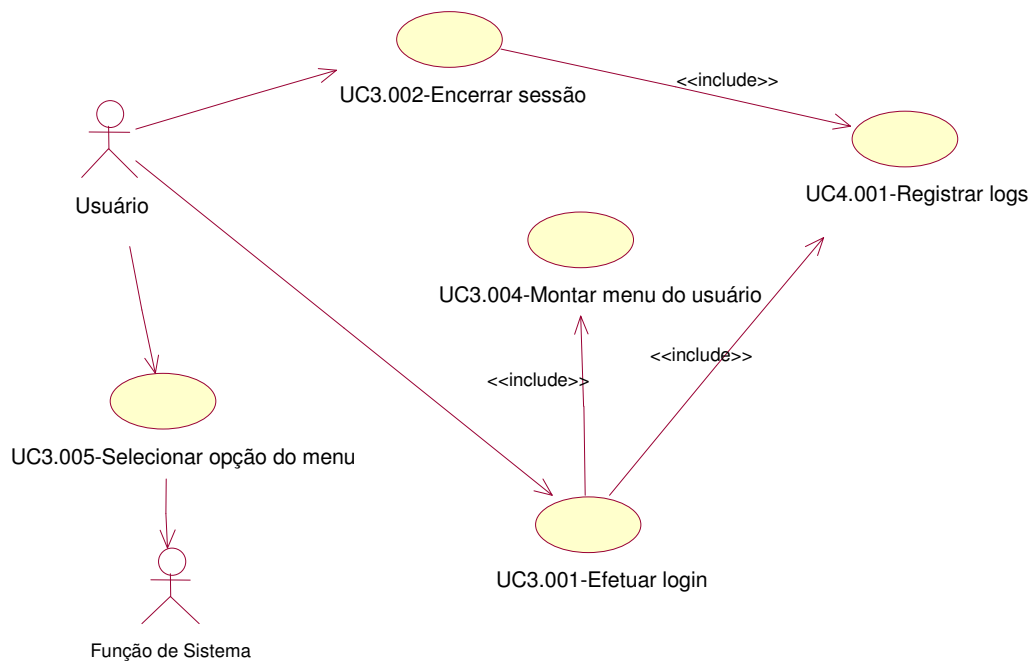


Figura 5.3- Diagrama de Casos de Uso Preliminar - Pacote Controle de Acesso

- **Trilha de Auditoria:** Os casos de uso deste pacote representam as funcionalidades que implementarão a trilha de auditoria da aplicação. A Figura 5.4 apresenta o diagrama de casos de uso inicial, onde foram identificados os atores *Funções de Sistemas*, *Usuário Geral* e *Clock*, que executam, respectivamente, os casos de uso: *Registrar Logs*, *Consultar Logs* e *Limpar Logs*.

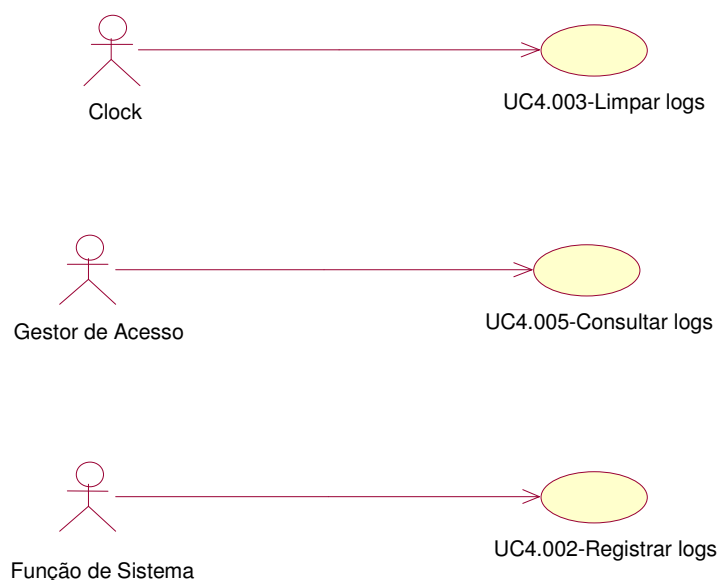


Figura 5.4 - Diagrama de Casos de Uso Preliminar - Pacote Trilha de Auditoria

A ferramenta utilizada para desenvolver os Diagramas de Casos de Uso foi o Rational Rose [Ros93]. A vantagem em usar esta ferramenta é que ela já está difundida e solidificada no mercado de software. A desvantagem é que é uma ferramenta relativamente cara, onde nem todos têm condições de adquiri-la.

#### 5.4.2.5. Atividade: Encontrar Requisitos Não-Funcionais (RNF's)

Segundo a nossa proposta, com a utilização do *Glossário Léxico* alguns requisitos não-funcionais aparecem naturalmente durante o levantamento de símbolos e suas definições. Como o processo é iterativo e incremental, sabemos que o *Glossário Léxico* está em evolução constante, e novos símbolos e requisitos não-funcionais são identificados continuamente.

Analisando o *Glossário Léxico* apresentado na Tabela 5.9, podemos observar que os Requisitos *Segurança*, *Performance* e *Usabilidade* aparecem em diversos símbolos. Seguindo a proposta de Cysneiros, em [Cys01-A] e [Kap00], constatamos que para cada RNF identificado



deveremos criar um grafo de RNF's e identificar as operacionalizações<sup>10</sup> necessárias para satisfazê-los.

Os grafos de RNF's foram desenvolvidos na ferramenta OME [Ome03], produzida na Universidade de Toronto, que não é comercializada na indústria, sendo seu uso limitado à academia.

A Figura 5.5 apresenta o Grafo de RNF's gerado para o sistema de Gestão de Acessos. Com base no glossário léxico e nos catálogos apresentados em [Chu00] foram identificados os RNF's primários: *Segurança*, *Performance* e *Usabilidade*. Estes RNF's foram então refinados e relacionados: o RNF *Segurança*, foi refinado em *Confidencialidade*, *Integridade dos Dados* e *Disponibilidade*. O RNF *Integridade dos Dados*, por sua vez, foi refinado em *completude* e *precisão*. Para garantir a *Completude*, foram identificadas as seguintes operacionalizações: *Registrar Logs de Acesso*, *Registrar Logs de Transações* e *Registrar Logs de Dados*. Estas três operacionalizações interferem positivamente em relação à *Precisão* e negativamente em relação ao RNF *Performance*. Para atender o requisito *Confidencialidade*, foi identificada a operacionalização *Permissão de Acesso*, que gerou as operacionalizações *Validar Acesso a Funções*, *Validar Acesso a Dados* e *Efetuar Login*. Estas operacionalizações interferem positivamente para a operacionalização *Configurar Acesso* e negativamente no RNF *Performance*. A operacionalização *Configurar Acesso*, que interfere negativamente na *Facilidade de Uso*, é realizada através das operacionalizações *Manter Cadastro de Perfil* e *Associar Perfil a Usuário*.

Como o RNF prioritário, para este projeto, é *Confidencialidade*, verificamos que os *softgoals* (definido na seção 2.4.2.1) ligados a este RNF ou que interferem positivamente com o mesmo devem ser atendidos. Já aqueles *softgoals* que sofrem interdependência negativa devido à *Confidencialidade*, não são atendidos ou são fracamente atendidos, como é o caso de *Tempo de Resposta* e *Número de Acessos*. Da mesma forma, podemos observar que, no caso de garantir a *Configuração do Acesso*, o RNF *Facilidade de Uso* será fracamente atendido.

Realizando esta atividade, constatamos que, através das operacionalizações podemos descobrir novos requisitos funcionais, candidatos a casos de uso, como foi o caso das operacionalizações *Registrar Logs de Acesso*, *Registrar Logs de Transações*, *Registrar Logs de Dados*, *Validar Acesso do Usuário*, *Validar Acesso às Funções* e *Validar Acesso aos Dados*; ou

---

<sup>10</sup> Estas operacionalizações foram encontradas também com a ajuda de catálogos apresentados por Chung, em [Chu00], que armazenam conhecimento acumulado em experiências prévias e são fontes de informações sobre requisitos não-funcionais. Vide seção 2.4.2.1.

certificar a real necessidade de determinados casos de uso, que ainda não estavam muito claros para o Engenheiro de Requisitos, como é o caso das operacionalizações *Manter Cadastro de Perfil e Associar Perfil de Acesso a Usuário*.

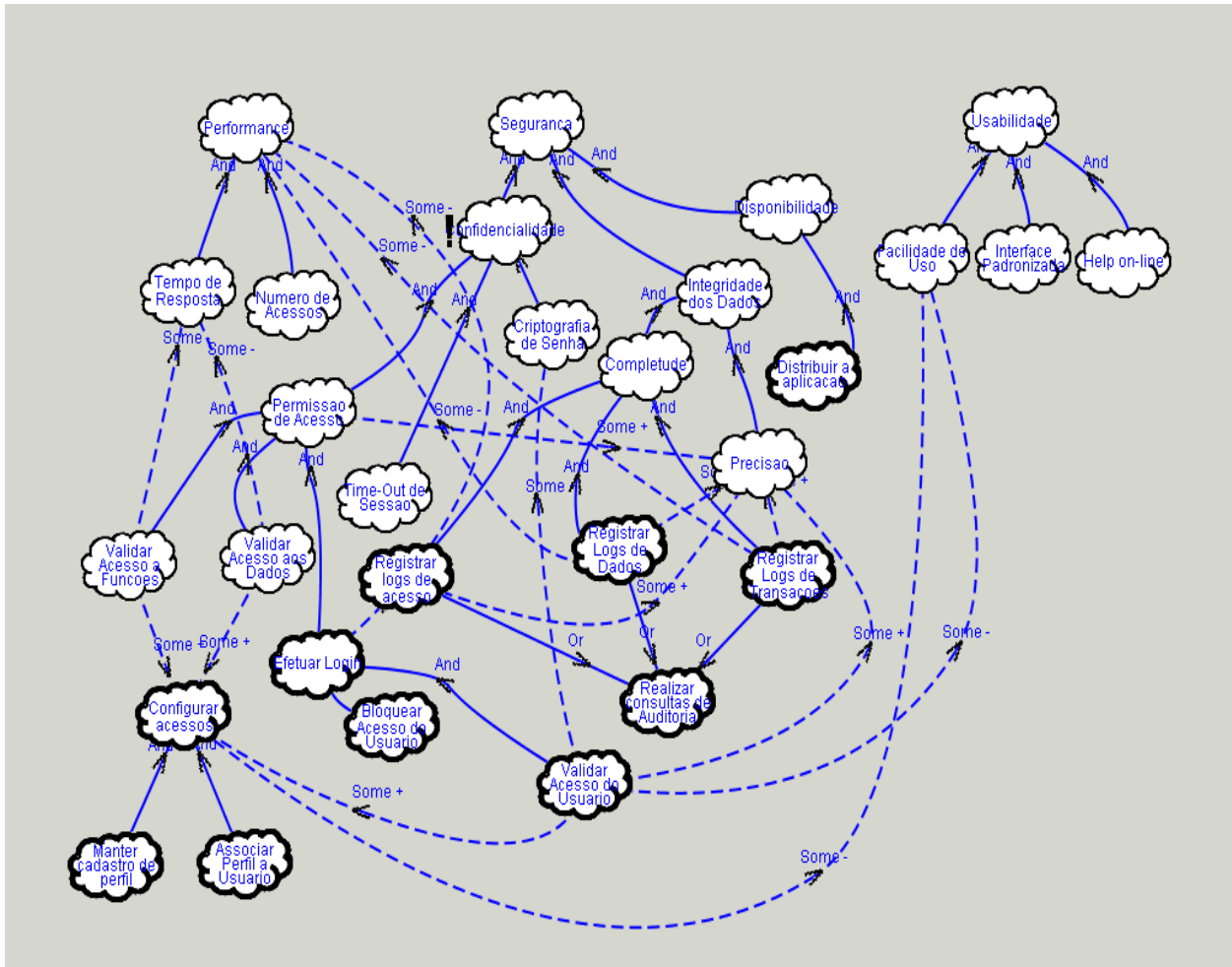


Figura 5.5- Grafo de RNF's com operacionalizações

Para simplificar o grafo de RNF's, apenas utilizamos os relacionamentos que apresentam as interdependências positivas ou negativas e as relações de contribuição "E" (*and*) e "OU" (*or*).

A dificuldade encontrada na execução desta atividade reside no fato da ferramenta OME3 ainda ser experimental e sua usabilidade ser limitada. Além disto, vemos que há uma sobrecarga de trabalho quando tentamos produzir este tipo de grafo, entretanto, a vantagem observada foi a identificação de novos Requisitos Funcionais, que neste caso, podem vir a ser casos de uso.

#### 5.4.2.6. Atividade: Gerenciar Dependências

Para o rastreamento de requisitos, aplicaremos as recomendações da técnica de rastreamento proposta por Toranzo, em [Tor02-A] e [Tor02-B].

As expressões usadas nas matrizes de rastreamento são apresentadas na Tabela 5.10.

*Tabela 5.10 - Expressões usadas nas matrizes de rastreamento*

Expressões usadas nas matrizes de rastreamento	
[OBO] = Objetivo organizacional	[REQ] = Requisito do sistema
[OBS] = Objetivo do sistema	[DGM] = Diagrama
[ATA] = Ata de Reunião	[ECU] = Especificação de Caso de Uso
[GLO] = Glossário Léxico	[PRO] = Programa do sistema

Como o processo proposto é iterativo e incremental, esta atividade é realizada repetidamente em mais de uma iteração. Novas matrizes serão criadas e as demais podem ser refinadas. Na execução desta atividade, no Subfluxo *Elicitar Requisitos*, seção 5.4.2.6, apenas as quatro primeiras diretrizes, propostas por Toranzo em [Tor02-A] e [Tor02-B], foram seguidas, pois ainda não temos informações suficientes para seguir adiante com as demais diretrizes:

1. **Identificar as informações que podem afetar o sistema:** O objetivo da diretriz é identificar classes<sup>11</sup>, no nível de informação ambiental, que podem afetar o sistema. A Tabela 5.11 apresenta estas informações do ambiente externo.

*Tabela 5.11- Matriz de Informações do Ambiente Externo*

Informações do Ambiente externo	
[RN01]	O sistema deve estar disponível 24 horas por dia, 7 dias por semana
[RN02]	Deve obedecer a uma política única, de controle de acesso

2. **Identificar os objetivos, estratégias ou regras de negócio que serão rastreadas:** O objetivo dessa diretriz é identificar os objetivos, estratégias ou regras registradas que podem afetar o sistema desejado. A Tabela 5.12 apresenta a matriz com os objetivos organizacionais identificados.

*Tabela 5.12 - Matriz com os Objetivos Organizacionais*

Objetivos Organizacionais	
[OBO01]	Deve obedecer a uma política única, de controle de acesso
[OBO02]	Gerenciar e centralizar o acesso de todos os usuários aos sistemas web da organização
[OBO03]	Disponibilizar consultas para a realização de trilhas de auditoria
[OBO04]	Controlar o acesso dos usuários a funções e dados da aplicação
[OBO05]	Cada usuário só poderá enxergar as funcionalidades acessíveis ao seu perfil

3. **Incluir classes de informação da gerência de projeto no modelo de rastreamento:** O objetivo dessa diretriz é recomendar a inclusão das tarefas do cronograma do projeto e os objetivos do sistema nos modelos de rastreamento dos projetos para reduzir o *gap* da

<sup>11</sup> Na técnica de Toranzo, ele organiza as informações em classes.

gerência de projeto com a gerência de requisitos. A Tabela 5.13 apresenta a matriz com os objetivos do sistema que foram identificados.

*Tabela 5.13 - Matriz com os Objetivos do Sistema*

Objetivos do Sistema	
[OBS01]	Validar acesso dos usuários
[OBS02]	Manter o cadastro dos usuários
[OBS03]	Gerar login para os usuários
[OBS04]	Manter o cadastro de perfis de acesso
[OBS05]	Associar perfis a usuários
[OBS06]	Configurar a política de segurança
[OBS07]	Manter o cadastro de sistema, subsistemas e funções
[OBS08]	Montar o menu do usuário dinamicamente, de acordo com seu perfil de acesso
[OBS09]	Gravar logs
[OBS10]	Consultar logs

4. **Identificar subsistemas:** O objetivo dessa diretriz é identificar os subsistemas que compõem o sistema desenvolvido. A Tabela 5.14 apresenta a matriz com os subsistemas identificados.

*Tabela 5.14 - Matriz com os Subsistemas*

Subsistemas	
[SUB01]	Gestão de Usuários
[SUB02]	Configuração da Gestão de Acessos
[SUB03]	Controle de Acesso
[SUB04]	Trilha de Auditoria

Desta forma, concluímos todas as atividades do subfluxo *Elicitar Requisitos*. A Próxima seção apresenta a aplicação do Subfluxo *Analisar e Documentar Requisitos* e a execução de suas atividades.

### **5.4.3. SUBFLUXO: ANALISAR E DOCUMENTAR REQUISITOS**

O objetivo deste subfluxo é documentar os requisitos. Como os artefatos são incrementados, à medida que o domínio do problema vai sendo cada vez mais conhecido (através de interações com os *stakeholders*, por exemplo), então o *Glossário Léxico*, os *Diagramas de Casos de Uso* e os *Grafos de RNF's*, definidos na seção anterior são refinados. Com base nesses documentos, nas *Atas de Reuniões* e na *Análise de Produtos e Documentos Existentes*, são produzidos os artefatos *Documento de Requisitos* e *Especificação de Casos de Uso*.

Neste subfluxo são detalhadas as execuções das seguintes atividades: *Capturar o Vocabulário do Domínio, Encontrar Atores e Casos de Uso, Integrar Visão Funcional e Não-Funcional e Documentar Requisitos.*

#### 5.4.3.1. Atividade: Capturar o Vocabulário do Domínio

Com o aprofundamento do entendimento do domínio da aplicação e a análise do *Grafo de RNF's* produzido, verificamos que, para atender as necessidades dos auditores, os registros de logs devem ser realizados de três formas distintas: registrar log de acesso, registrar logs de transações e registrar logs de dados. Desta forma, o *Glossário Léxico*, apresentado na Tabela 5.9, foi incrementado com as seguintes linhas:

Tabela 5.15- Incremento para o Glossário Léxico da Aplicação

Símbolo	Tipo	Definições	Impactos	RNF's Associados
Auditor	Sujeito	Pessoa responsável pela realização de auditorias nos softwares da organização.		
Log de acesso	Objeto	Armazenamento de informações relativas aos acessos de <u>usuários</u> aos sistemas, visando garantir o rastreamento de informações, quando necessário.	1.Os acessos de todos os <u>usuários</u> aos sistemas devem ser registrados.	SEGURANÇA, COMPLETEUDE
Log de dados	Objeto	Armazenamento de informações relativas à manipulação de dados, visando garantir o rastreamento de informações, quando necessário.	1.Sempre que houver manipulação de um <u>elemento de dados</u> e o sistema estiver configurado para registrar os logs de dados do <u>elemento de dados</u> em questão, esta informação deve ser registrada, armazenando os valores anteriores e posteriores à transação realizada.	SEGURANÇA, COMPLETEUDE
Log de transação	Objeto	Armazenamento de informações relativas à execução de determinadas transações nos sistemas, visando garantir o rastreamento de informações, quando necessário.	1.Sempre que uma <u>função</u> for executada, cuja configuração do sistema indique que cada execução da mesma deva ser registrada no log, o sistema registra o log de transação	SEGURANÇA, COMPLETEUDE

### 5.4.3.2. Atividade: Integrar Visão Funcional e Não-Funcional

Após a identificação dos Requisitos Não-Funcionais e suas operacionalizações, através do *Grafo de RNF's*, apresentado na Figura 5.5, sentimos a necessidade de integrar as visões funcional e não-funcional da aplicação. Para integrar estas visões, usamos a estratégia proposta por Cysneiros, em [Cys01-B] e [Cys01-C], onde as operacionalizações encontradas no *Grafo de RNF's* passam a ser refletidas nos *Diagramas de Casos de Uso*.

Analisando o Grafo de RNF's produzido na Figura 5.5, verificamos que, as seguintes operacionalizações encontradas, a partir do requisito Segurança, são candidatas a serem casos de uso da aplicação: *Registrar Logs de Acesso*, *Registrar Logs de Transações*, *Registrar Logs de Dados*, *Validar Acesso do Usuário*, *Validar Acesso à Funções* e *Validar Acesso aos Dados*.

Desta forma, os diagramas de casos de uso são refinados para refletirem as operacionalizações encontradas. A seguir são apresentados os modelos de casos de uso resultantes da integração entre as visões funcional e não-funcional dos pacotes que foram modelados na seção anterior:

- **Configuração do Sistema Gestão de Acesso:** Ao integrar o modelo de casos de uso proposto para este pacote, mostrado na Figura 5.1, com o Grafo de RNF's da Figura 5.5, obtemos o modelo da Figura 5.6, onde novos casos de uso são incorporados. Verifica-se a adição dos casos de uso *Gravar Restrições de Acesso* e *Obter Restrições de Acesso*, sendo que este último deve ser chamado por todos os casos de uso de qualquer aplicação gerenciada pelo sistema, pois é responsável por garantir a segurança das informações que são apresentadas ou manipuladas pelo usuário logado.

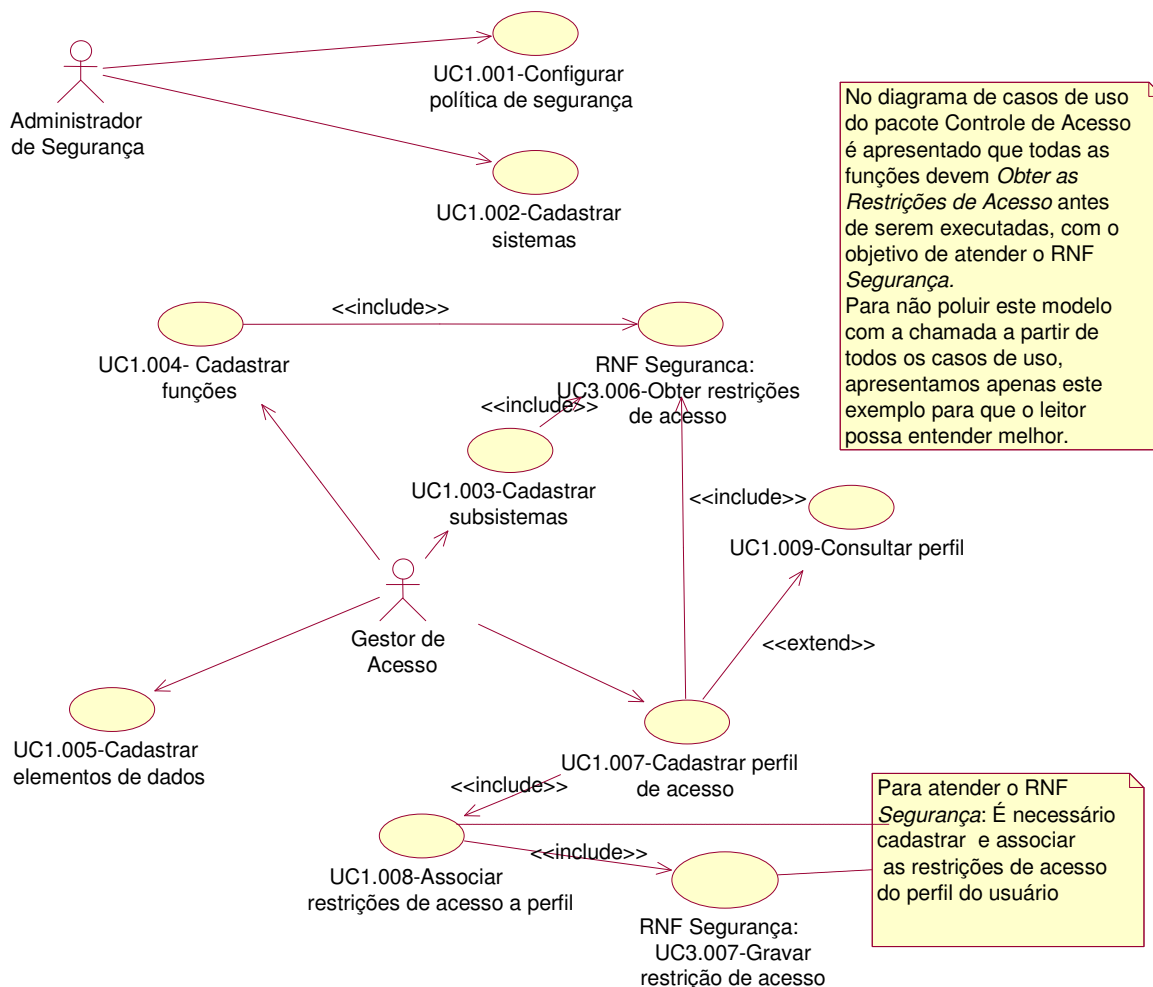


Figura 5.6-Diagrama de casos de uso refletindo RNF's - Pacote Configuração do Sistema Gestão de Acesso

- **Gestão de Usuários:** A Figura 5.2 apresenta o diagrama de casos de uso preliminar, analisando-a, junto com o Grafo de RNF's da Figura 5.5, podemos refinar o modelo e obter o modelo apresentado na Figura 5.7.

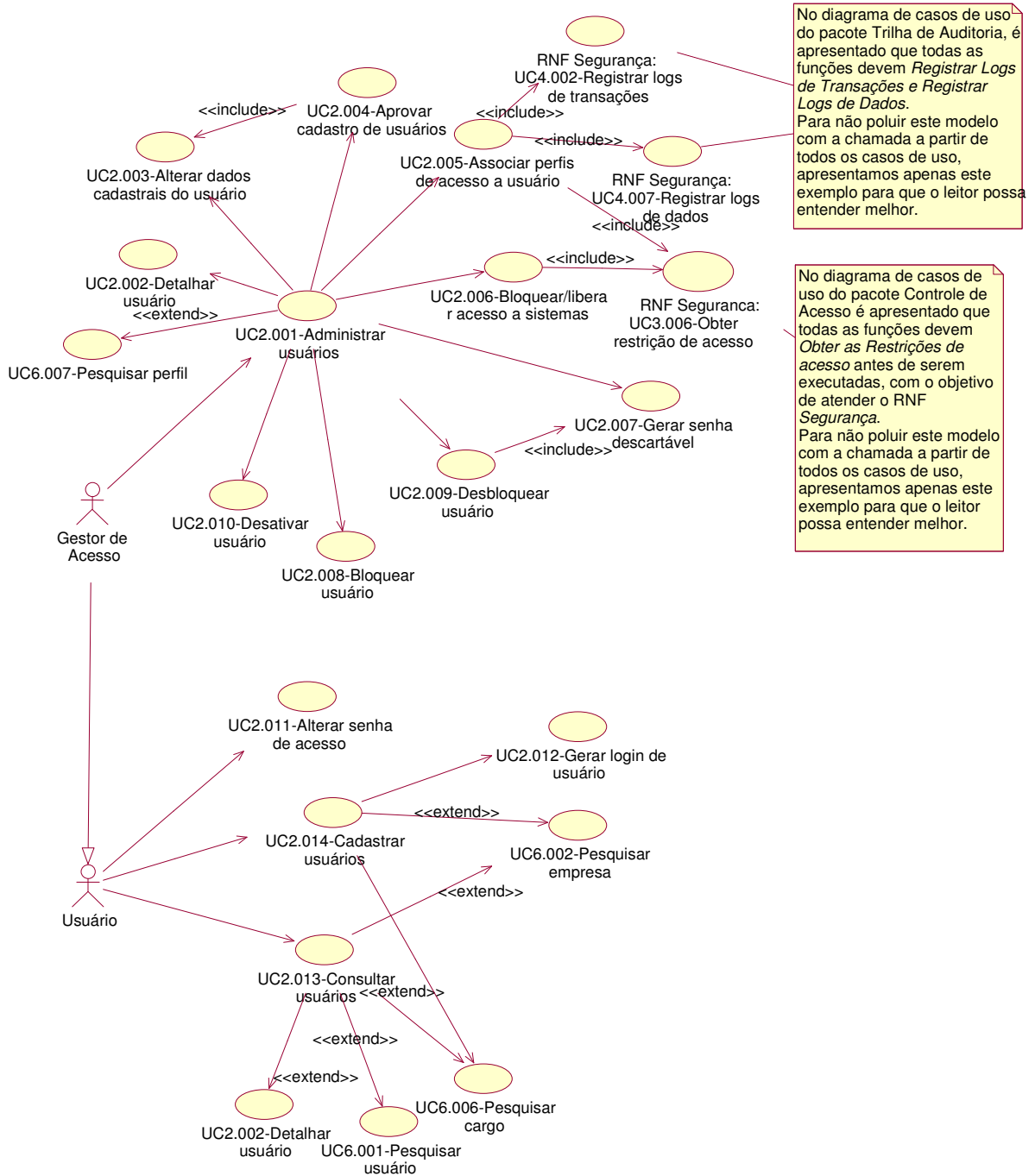


Figura 5.7- Diagrama de casos de uso refletindo RNF's - Pacote Gestão de Usuários

- Controle de Acesso:** A Figura 5.3 apresenta o diagrama de casos de uso inicial para este pacote. Integrando-a com o Grafo de RNF's da Figura 5.5, podemos refinar o modelo e obter a Figura 5.8. Verificamos que no modelo de casos de uso que reflete as operacionalizações, é executado o caso de uso *Registrar Logs de Acesso*, sempre que um



usuário tentar efetuar o login ou quando encerrar a sessão (para registrar a data de hora do início e fim do acesso). Também podemos observar através desse modelo que ao serem executadas as funcionalidades da aplicação, devem ser obtidas as restrições de acesso do usuário logado.

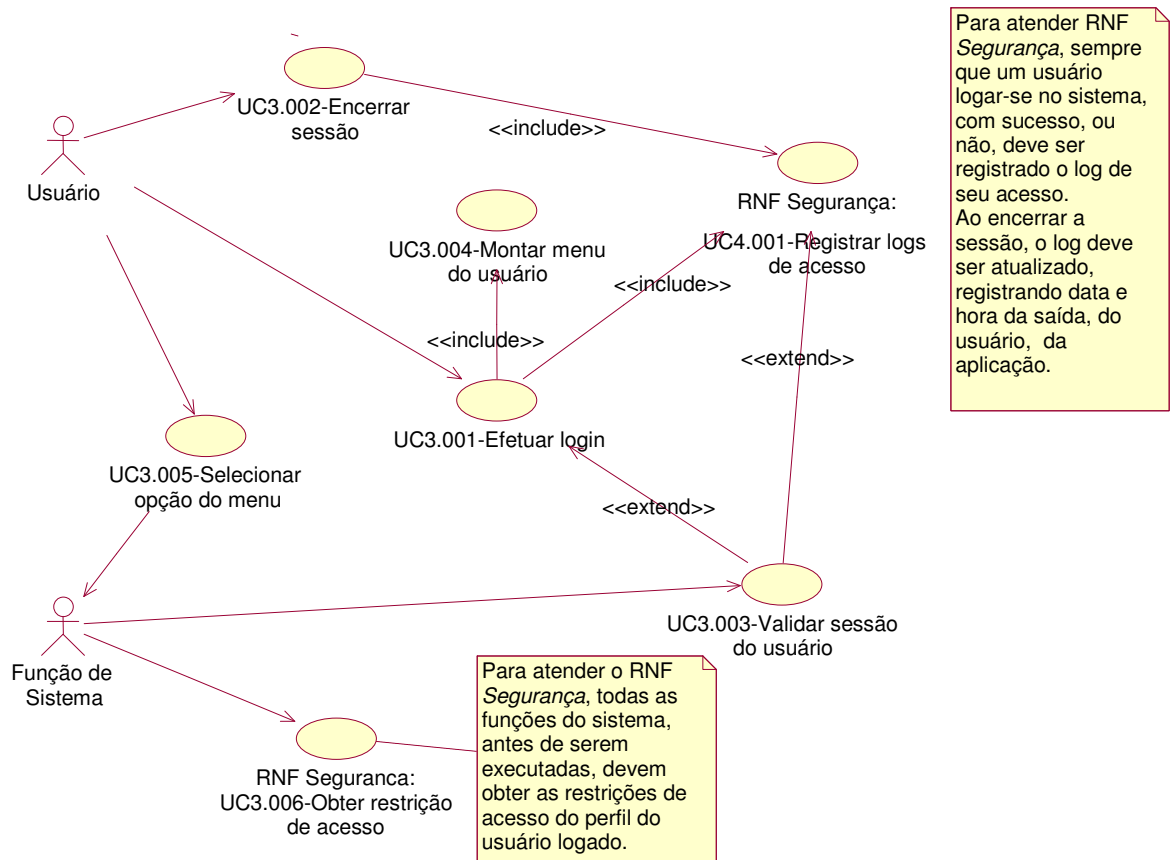


Figura 5.8- Diagrama de casos de uso refletindo RNF's - Pacote Controle de Acesso

- **Trilha de Auditoria:** A Figura 5.4 apresenta o diagrama de casos de uso preliminar para este pacote. Integrando-a, com o Grafo de RNF's da Figura 5.5, podemos refinar o modelo e chegar naquele apresentado na Figura 5.9. Nesse modelo verificamos que as funções das aplicações gerenciadas pelo sistema devem registrar os logs de transação, indicando que a função está sendo executada, e os logs de dados, gravando as informações manipuladas pelos usuários.

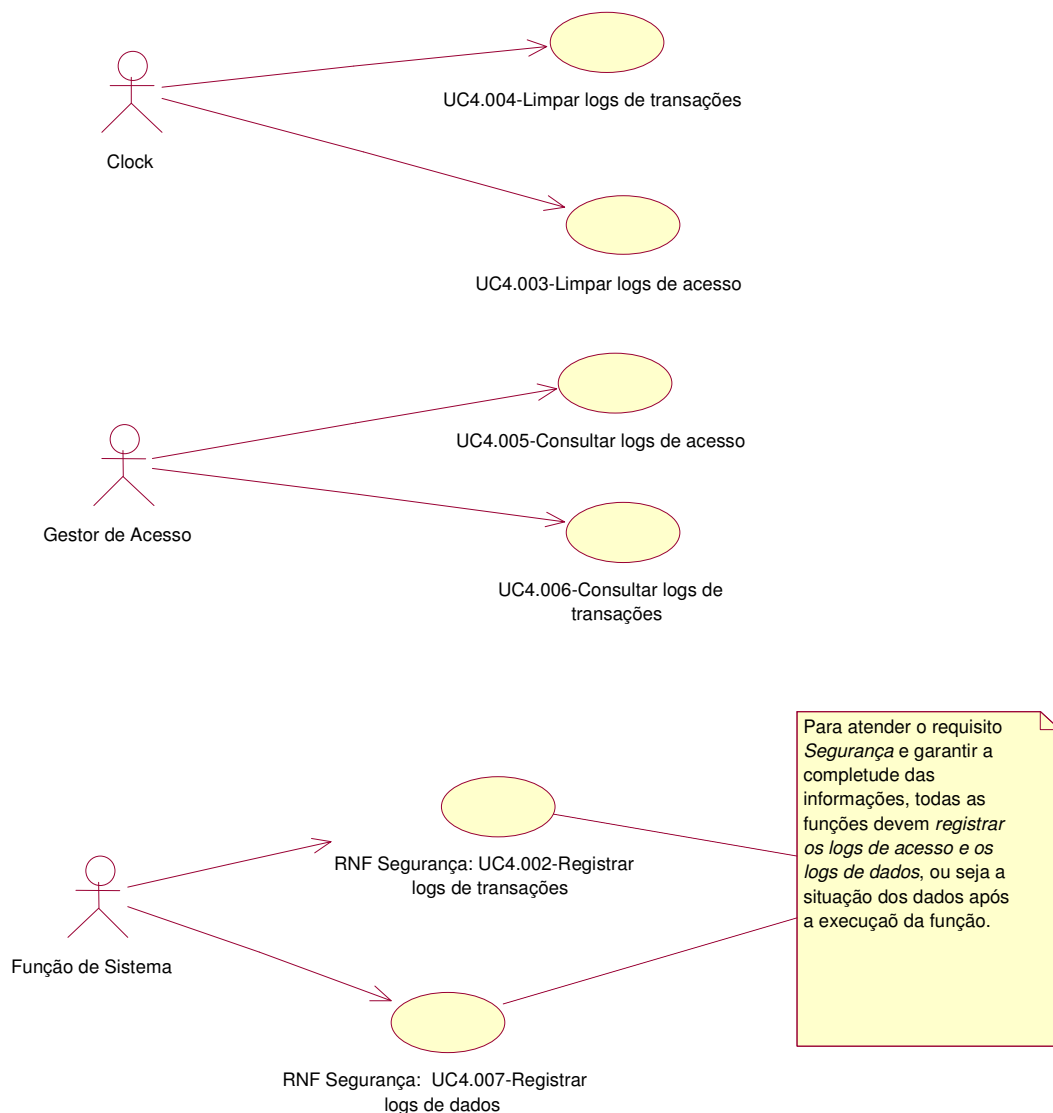


Figura 5.9 - Diagrama de casos de uso refletindo RNF's - Pacote Trilha de Auditoria

### 5.4.3.3. Atividade: Documentar Requisitos

A realização desta atividade produz os artefatos Documento de Requisitos e Especificação de Casos de Uso. Estes documentos são escritos em linguagem natural, segundo os templates definidos no capítulo 4, seção 4.2.3.2.

Primeiramente foi produzido o documento de requisitos, onde os requisitos foram divididos em pacotes (de acordo com a regra de negócio a qual pertenciam), identificados com

unicidade e descritos em linguagem natural. O documento de requisitos produzido é apresentado no Apêndice A, a Tabela 5.16 apresenta a descrição de um dos requisitos do sistema.

Tabela 5.16- Exemplo de Requisito

<b>[RF1.001] Manter Cadastro de Sistemas e Subsistemas</b>			
<i>Este requisito permite consultar e manter o cadastro de sistemas sobre o ambiente Web e cadastro de seus subsistemas. Os itens do menu de acesso serão montados e agrupados a partir da hierarquia de sistemas e subsistemas cadastrados.</i>			
<b>Prioridade:</b>	<input checked="" type="checkbox"/> Essencial	<input type="checkbox"/> Importante	<input type="checkbox"/> Desejável

Com base no Documento de Requisitos e no Modelo de Casos de Uso, foi produzido o documento de Especificação de Casos de Uso, onde são descritos, com poucos detalhes, os casos de uso identificados. Estes casos de uso irão sendo detalhados de acordo com o planejamento de cada iteração. O subfluxo *Refinar a Definição do Sistema*, prevê a execução da atividade *Detalhar Casos de Uso*, onde apenas os casos de uso planejados para a iteração devem ser detalhados. O documento de casos de uso preliminar é apresentado no Apêndice B<sup>12</sup>. A Tabela 5.17 apresenta a descrição preliminar de um caso de uso do sistema.

Tabela 5.17- Exemplo de Caso de Uso

<b>[UC1.002] Cadastrar sistemas</b>		
Objetivo: <b>Manter o cadastro dos sistemas gerenciados pelo sistema Gestão de Acesso</b>		
Ator Responsável: <b>Administrador de Segurança</b>		
Requisitos Atendidos: <b>RF1.001</b>		
<b>Contexto</b>		
<b>Eventos que Disparam o Caso de Uso</b>	<b>Entradas</b>	<b>Saídas</b>
Selecionar função Cadastrar sistemas		
<b>Pré condições:</b> o usuário logado deve ter perfil de administrador de segurança.		
<b>Pós condições:</b> se o caso de uso for realizado com sucesso os sistemas serão incluídos, alterados ou excluídos no controle de acesso.		
<b>Fluxo de eventos principal</b>		
1. O sistema apresenta a lista dos sistemas cadastrados com as seguintes informações: <ul style="list-style-type: none"> <li>• Sigla do sistema</li> <li>• Nome do sistema</li> <li>• Nome abreviado do sistema</li> <li>• Ordem no menu</li> </ul>		
2. O sistema solicita que o usuário selecione um sistema da lista e informe a ação que pretende executar. Conforme a resposta recebida desvia o fluxo para um dos subfluxos abaixo: <ul style="list-style-type: none"> <li>• Incluir sistema ([SB001] Incluir sistema)</li> <li>• Alterar sistema ([SB002] Alterar sistema)</li> <li>• Excluir sistema ([SB003] Excluir sistema)</li> </ul>		

<sup>12</sup> Como o volume de informações é muito grande, apenas alguns casos de uso são detalhados, no Apêndice B, com o objetivo de validar o uso do template proposto.

**[SB001] Incluir sistema**

1. O sistema exibe formulário solicitando informações do sistema.
2. O usuário informa os dados e solicita a inclusão do sistema
3. O sistema inclui os dados do sistema e emite mensagem de operação realizada com sucesso.
4. O sistema executa o caso de uso [UC4.002] Registrar log de transação.
5. O sistema retorna ao passo 1 deste subfluxo para permitir a inclusão de um novo sistema.

**[SB002] Alterar sistema**

1. O sistema exibe um formulário com as informações do sistema a ser alterado.
2. O usuário altera os dados e solicita efetivação
3. O sistema altera os dados do sistema e emite mensagem de operação realizada com sucesso.
4. O sistema executa o caso de uso [UC4.002] Registrar log de transação.
5. O sistema retorna ao fluxo principal.

**[SB003] Excluir sistema**

1. O sistema exibe um formulário de consulta apresnetando as informações do sistema selecionado.
2. O sistema solicita que o usuário confirme a exclusão do sistema;
3. O usuário confirma a exclusão do sistema;
4. O sistema efetua a exclusão
5. O sistema executa o caso de uso [UC4.002] Registrar log de transação.
6. O sistema retorna ao fluxo principal.

**Fluxos secundários (alternativos e de exceção)**

**Requisitos Não-Funcionais Associados**

**Segurança:**

**Observações:**

Desta forma, concluímos todas as atividades do subfluxo *Analisar e Documentar Requisitos*. A Próxima seção apresenta a aplicação do Subfluxo *Definir a Interface* e a execução de suas atividades.

#### **5.4.4. SUBFLUXO: DEFINIR A INTERFACE**

Através deste fluxo de trabalho são especificados os requisitos de interface e gerado o protótipo de interface gráfica. Com base no *Glossário Léxico*, nos *Diagramas de Casos de Uso*, *Documento de Requisitos* e *Especificação de Casos de Uso*, bem como nas *Atas de Reuniões*, são produzidos o *Modelo de Casos de Uso de Interface*, *Documento de Requisitos de Interface* e *Especificação de Casos de Uso de Interface*.

Neste subfluxo são detalhadas as execuções das seguintes atividades: *Especificar Requisitos de Interface*, *Projetar Visibilidade da Aplicação* e *Definir Protótipo de Interface*. A atividade *Projetar Conetúdo*, por ser fundamentalmente editorial, não estará sendo validada neste relato de experiência.

#### 5.4.4.1. Atividade: Especificar Requisitos de Interface

O objetivo desta atividade é analisar e especificar os requisitos de interface. A Figura 5.10 apresenta o *Modelo de Casos de Uso de Interface*, onde é apresentada a navegabilidade a partir da página inicial da aplicação, até o momento que o usuário seleciona uma opção de menu.

No projeto prático, constatamos que o *Modelo de Casos de Uso de Interface* deve ser produzido nos casos onde a interação entre o usuário e a aplicação precise ser bem definida para que possa ser entendida por todas as pessoas envolvidas no projeto, desde os *stakeholders* até os desenvolvedores. Verificamos que há uma sobrecarga de trabalho caso sejam produzidos modelos de casos de uso de interface para as interações entre todos os casos de uso definidos e os usuários. Além disto, em muitos casos, a própria especificação dos casos de uso funcionais já deixa claro como deve ser a interação entre o usuário e a aplicação, tornando fácil a “visualização mental” da interface.

Sendo assim, concentramos nossos esforços na definição da interface onde a especificação de casos de uso não é suficientemente clara a respeito da interação entre o usuário e a aplicação. Um exemplo disto ocorre no caso onde o usuário acessa a página inicial: é possível, para qualquer usuário web, navegar pelas promoções, navegar pelas notícias, navegar pelo mapa do site e pela ajuda. Esta estrutura navegacional não é definida na especificação de casos de uso, pois não são funcionalidades da aplicação. Apenas os usuários com acesso à aplicação podem efetuar o login e, a partir daí, o sistema monta, automaticamente, o menu do usuário, de acordo com suas permissões de acesso; o usuário, então, seleciona uma das opções do menu e executa as funcionalidades especificadas através dos casos de uso. Desta forma, observamos que com a definição do “modelo navegacional” é possível ter uma visibilidade clara da forma como a aplicação está estruturada visualmente e do comportamento do usuário.

Para complementar o modelo desenvolvemos documentos para definição dos requisitos de interface, conforme apresentado no Apêndice C, e os Casos de Uso de Interface, no Apêndice D. No primeiro documento, são descritos os requisitos de interface que o sistema deve contemplar, A Tabela 5.18 apresenta a descrição de um destes requisitos. No segundo documento são descritos os casos de uso de interface, que foram identificados no modelo de casos de uso de interface e apresentados alguns cenários de uso da interface, a Tabela 5.19 apresenta a descrição de um destes casos de uso.

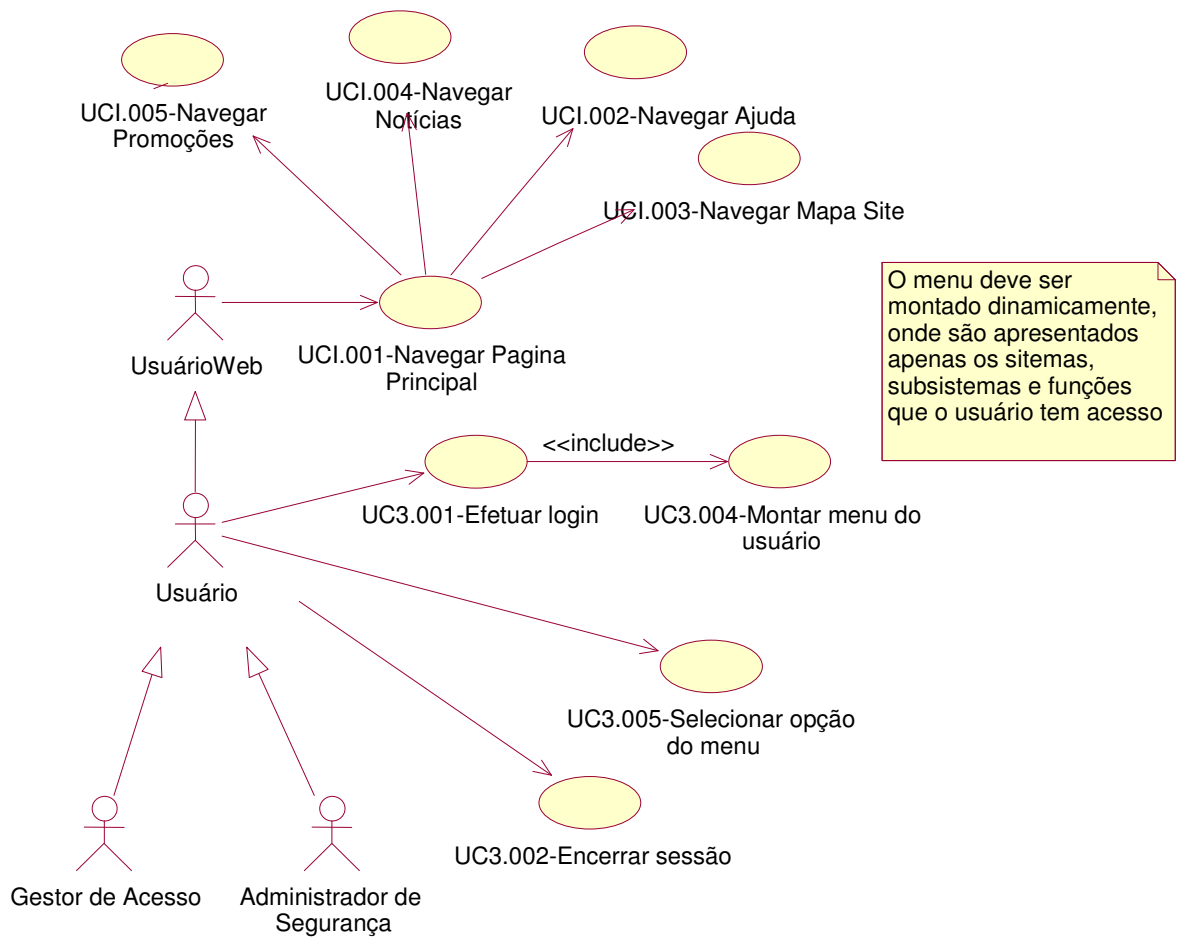


Figura 5.10- Modelo de Casos de Uso de Interface

Tabela 5.18 - Exemplo de Requisito de Interface

<b>[RI001] Interface Simplificada</b>
<i>O sistema deve prover uma interface simples, para facilitar ao máximo o uso do mesmo por parte dos usuários uma vez que a grande maioria das telas do sistema serão telas de trabalho de oito horas/dia. A navegação deverá ser intuitiva, fornecendo ao usuário opções possíveis, evitando desta forma a entrada de dados que possam apresentar conflitos.</i>
<b>Prioridade:</b> <input checked="" type="checkbox"/> Essencial <input type="checkbox"/> Importante <input type="checkbox"/> Desejável
<b>Forma de comprovação:</b> através da aprovação do protótipo navegacional, protótipo de interface e utilização do sistema por parte dos usuários.

Tabela 5.19 - Exemplo de Caso de Uso de Interface

[UCI.001] Navegar Página Principal
<ol style="list-style-type: none"><li>1. Ao entrar na página inicial da aplicação, o usuário visualiza links para:<ul style="list-style-type: none"><li>• Navegar pelas promoções</li><li>• Navegar pelas notícias</li><li>• Navegar pela ajuda ([UCI.002] Navegar Ajuda)</li><li>• Navegar pelo mapa do site</li></ul></li><li>2. O sistema também apresenta um formulário para solicitar:<ul style="list-style-type: none"><li>• Login</li><li>• Senha</li></ul></li><li>3. O usuário seleciona um dos links ou informa login e senha para tentar acessar o sistema.</li></ol>

Neste projeto não realizamos a geração automática do protótipo de interface<sup>13</sup>, logo não foram elaborados diagramas de classe e de seqüência. Além disso, como esses artefatos são gerados no fluxo de *Análise e Projeto*, através da realização dos casos de uso, deixamos a definição formal do aspecto comportamental da aplicação para o projetista de software.

#### 5.4.4.2. Atividade: Projetar Visibilidade da Aplicação

Com a realização desta atividade são produzidos guias para orientar a definição das interfaces e um documento descrevendo o layout da aplicação.

A Tabela 5.20 apresenta um trecho de um documento definido para guiar a produção das interfaces da aplicação.

Tabela 5.20 - Exemplo do Documento de Descrição da Interface

Sistema Gestão de Acessos Memorial Descritivo da Interface
<b>1.Introdução</b> Este documento define e justifica, de acordo com os requisitos de interface levantados do sistema Gestão de Acessos, as ações tomadas pela equipe de design com o objetivo de criar a interface dentro dos padrões de usabilidade, navegabilidade e design gráfico. Durante o desenvolvimento da aplicação só serão tratados os requisitos de interface para construção do layout que será usado para os módulos deste sistema, e que servirá como padrão para as novas aplicações desenvolvidas para a operadora de cartão de crédito.
<b>2.Definição conceitual da proposta:</b> Levando-se em consideração os requisitos, procuramos fazer uma analogia do software com a mesa de trabalho de um usuário. É o ambiente onde ele passa uma média de 8 horas/dia. Para que o trabalho seja mais produtivo, a interface se preocupou em organizar as informações sem muita interferência visual, se limitando ao estritamente necessário.
<b>2.1. Cores</b> A escolha das cores foi baseada na paleta de cores do site institucional da Operadora. São elas: vermelho, laranja e cinza para detalhes e texto, e o uso intensivo do branco para o fundo da tela.
<b>2.2. Imagens</b> Para humanizar o software sugerimos que a aplicação utilize imagens que representem o usuário, para

<sup>13</sup> Através da aplicação da técnica de Sanchez [San02] é possível gerar, automaticamente, o protótipo de interface, com base nos casos de uso e diagramas de seqüência. Entretanto, não validamos a aplicação desta técnica nesta dissertação devido às limitações de tempo para o aprendizado, configuração e instalação de ferramentas.

que o mesmo se identifique com o sistema. Por exemplo, uma das imagens escolhidas mostra as mãos de uma pessoa executando uma ação no computador.

### **2.3.Menu**

As opções de menu estão divididas em três grupos:

- O primeiro menu localizado na área superior direita que apresenta um grupo de conteúdo não relacionado ao sistema (promoções e notícias) acompanhado de funções de suporte ao sistema (ajuda e mapa do site);
- O segundo apresentando, num primeiro nível, os sistemas disponíveis para o usuário, desdobrando-se nas opções de cada sistema. Localizando-se na área superior da tela logo abaixo do cabeçalho de imagem;
- O terceiro menu é apresentado com as opções do quarto nível de navegação, sua criação tem o intuito de facilitar o acesso às opções do sistema, sem sobrecarregar a tela com vários desdobramentos do menu superior.

### **5.4.4.3. Atividade: Definir Protótipo da Interface**

Analisando os *Documentos de Requisitos, Requisitos de Interface, Especificação dos Casos de Uso, Especificação de Casos de Uso de Interface e os Guias Descritivos para o desenvolvimento de interfaces gráficas* produzimos o protótipo da interface gráfica com o usuário.

A utilização do *Glossário Léxico*, também auxilia a nomear os campos de telas, pois são considerados os termos usados no domínio da aplicação, que têm sido identificados e definidos pelo usuário.

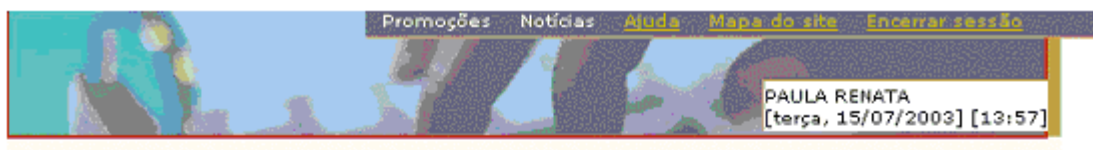
A definição do *lay-out* do protótipo, as telas de menu e algumas funcionalidades básicas, foram desenvolvidas pelos projetistas de interface (*designers*). Já as telas de funcionalidades representadas nos documentos de casos de uso, foram implementadas pelos Engenheiros de Requisitos, haja vista eles terem um conhecimento mais profundo da aplicação.

Neste projeto, as interfaces foram produzidas utilizando a ferramenta DreamWeaver, onde páginas HTML são geradas, de acordo com o padrão de interfaces definido. Não foi possível validar a utilização da técnica de Sánchez [San02], onde o protótipo é gerado automaticamente com base nos casos de uso e diagramas de seqüência, devido às limitações de tempo para o aprendizado, configuração e instalação das ferramentas necessárias. A Figura 5.11 e a Figura 5.12 apresentam dois exemplos de telas do protótipo de interface.





Figura 5.11- Exemplo da tela de login, do protótipo de interface



Olá PAULA RENATA,

Bem-vindo(a) aos Sistemas

O seu último acesso foi em 14/07/2003 às 18:59.

#### Gestão de Acesso

Sistema gerenciador de informações para controlar o acesso a todas as aplicações.

[clique aqui](#)

#### Controle Financeiro

Sistema Financeiro de Operadora de Cartão de Crédito.

[clique aqui](#)

#### Central de Atendimento

Funcionalidades operacionais e consultas gerenciais para realização das atividades da Central de Atendimento.

[clique aqui](#)

#### Ferramentas

Ferramentas gerais de auxílio ao usuário.

[clique aqui](#)

Figura 5.12- Exemplo da tela de menu, no protótipo de interface

Desta forma, concluímos todas as atividades do subfluxo *Definir a Interface*. A Próxima seção apresenta a aplicação do Subfluxo *Gerenciar Escopo* e a execução de suas atividades.

### 5.4.5. SUBFLUXO: GERENCIAR ESCOPO

Este subfluxo é executado paralelamente às atividades do fluxo de Gerência de Projetos, pois seu objetivo é determinar o escopo da iteração atual. Neste subfluxo são apresentadas as execuções das atividades *Priorizar Casos de Uso* e *Gerenciar Dependências*.

#### 5.4.5.1. Atividade: Priorizar Casos de Uso

O critério adotado de seleção dos casos de uso para serem implementados na primeira iteração foi escolher aqueles que permitissem validar a arquitetura da aplicação, juntamente com aqueles que poderiam formar um ciclo de negócio, tornando-se um produto a ser entregue ao usuário. Sendo assim, para validar a arquitetura proposta, foram selecionados os casos de uso *Cadastrar Sistemas*, *Cadastrar Perfil de Acesso*, *Consultar Perfis*, *Configurar Política de Segurança*, *Montar Menu do Usuário*, *Efetuar Login*, *Administrar Usuários* e *Associar Perfil de Acesso a Usuários*.

Com a seleção destes casos de uso, os engenheiros de requisitos e projetistas de software passam a trabalhar focando na solução dos mesmos. Na seção 5.4.6.1 é descrita a atividade onde estes casos de uso são detalhados.

#### 5.4.5.2. Atividade: Gerenciar Dependências

Devido ao fato do processo ser iterativo e incremental, essa atividade deve ser executada repetidamente. Aplicando as recomendações da técnica de rastreamento proposta por Toranzo, em [Tor02-A] e [Tor02-B], podemos gerenciar as dependências através do refinamento das matrizes definidas na atividade *Gerenciar Dependências* do Subfluxo *Elicitar Requisitos*.

Com o conhecimento adquirido neste estágio do projeto, é possível aplicar as seguintes diretrizes da técnica de Toranzo:

- Diretriz 5: Definir uma matriz para a identificação dos requisitos,
- Diretriz 6: Identificar os tipos de diagramas nos quais os requisitos são representados.

Adequando o escopo definido por Toranzo à nossa proposta, podem ser definidas matrizes para identificação dos casos de uso, requisitos de interface e casos de uso de interface.

A Tabela 5.21 apresenta a matriz dos requisitos do pacote *Configuração do Sistema Gestão de Acessos* e a Tabela 5.22 apresenta a matriz dos casos de uso do mesmo pacote. A Tabela 5.23 apresenta os diagramas desenvolvidos até agora, que devem ser rastreados.

Não elaboraremos aqui todas as matrizes devido ao grande volume de informações, trabalharemos apenas com um subconjunto dos requisitos para validarmos a aplicabilidade da técnica de Toranzo ao modelo proposto.

*Tabela 5.21- Matriz dos Requisitos do Pacote Configuração*

Requisitos do Pacote Configuração
[RF1.001] Manter Cadastro de Sistemas e Subsistemas
[RF1.002] Manter Cadastro de Funções
[RF1.003] Definir Restrições de Acesso a Dados
[RF1.004] Definir Perfis de Acesso
[RF1.005] Consultar Perfis de Acesso
[RF1.006] Configurar Política de Segurança

*Tabela 5.22-Matriz de Casos de Uso do Pacote Configuração*

Casos de Uso do Pacote Configuração
[UC1.001] Configurar política de segurança
[UC1.002] Cadastrar sistemas
[UC1.003] Cadastrar subsistemas
[UC1.004] Cadastrar funções
[UC1.005] Cadastrar elementos de dados
[UC1.006] Cadastrar restrições
[UC1.007] Cadastrar perfil de acesso
[UC1.008] Configurar restrições por perfil

*Tabela 5.23 - Matriz com os diagramas do sistema Gestão de Usuários*

Diagramas
[DIAG01] Diagrama de Casos de Uso do Pacote Configuração
[DIAG02] Diagrama de Casos de Uso do Pacote Gestão de Usuários
[DIAG03] Diagrama de Casos de Uso do Pacote Controle de Acesso
[DIAG04] Diagrama de Casos de Uso do Pacote Trilha de Auditoria
[DIAG05] Diagrama de Casos de Uso de Interface

Apresentamos as atividades do subfluxo *Gerenciar Escopo*. A próxima seção apresenta a aplicação do Subfluxo *Refinar a Definição do Sistema* e a execução da atividade *Detalhar Casos de Uso*.

#### **5.4.6. SUBFLUXO: REFINAR A DEFINIÇÃO DO SISTEMA**

Este subfluxo é executado a cada iteração, com o objetivo de detalhar os requisitos e casos de uso planejados para a iteração atual. Apenas a atividade *Detalhar Casos de Uso* é executada neste subfluxo. Esta atividade é descrita na subseção a seguir.

### 5.4.6.1. Atividade: Detalhar Casos de Uso

Como descrito na seção 5.4.5.1, para validar a arquitetura proposta, foram selecionados os casos de uso *Cadastrar Sistemas*, *Cadastrar Perfil de Acesso*, *Consultar Perfis*, *Configurar Política de Segurança*, *Montar Menu do Usuário*, *Efetuar Login*, *Administrar Usuários* e *Associar Perfil de Acesso a Usuários*. Na primeira iteração do sistema, estes casos de uso são detalhados na *Especificação de Casos de Uso* e validados. Os demais casos de uso, que haviam sido identificados nos modelos de casos de uso, são descritos com detalhes nas iterações posteriores. Sugerimos que, caso o engenheiro de requisitos tenha informações e tempo para descrever, mesmo que brevemente, os casos de uso não selecionados para a iteração atual, ele deve fazê-lo, pois facilita a visão integrada do projeto. O Apêndice B apresenta o detalhamento de alguns dos casos de uso do pacote *Configuração do Sistema Gestão de Acessos*, que foram priorizados para a primeira iteração do desenvolvimento.

O acréscimo de detalhes à especificação dos casos de uso permite um maior entendimento das funcionalidades, além de fornecer mais subsídios aos projetistas e engenheiros de software para as etapas de análise de projeto e implementação. A Tabela 5.24 apresenta o detalhamento do caso de uso *Cadastrar Sistemas* (Vide página 165).

Tabela 5.24- Exemplo de Caso de Uso Detalhado

[UC1.002] Cadastrar sistemas		
Objetivo: <b>Manter o cadastro dos sistemas gerenciados pelo sistema Gestão de Acesso</b>		
Ator Responsável: <b>Administrador de Segurança</b>		
<b>Contexto</b>		
<b>Eventos que Disparam o Caso de Uso</b>	<b>Entradas</b>	<b>Saídas</b>
Selecionar função Cadastrar sistemas		
<p><b>Pré condições:</b> o usuário logado deve ter perfil de administrador de segurança.</p> <p><b>Pós condições:</b> se o caso de uso for realizado com sucesso os sistemas serão incluídos, alterados ou excluídos no controle de acesso.</p> <p><b>Fluxo de eventos principal</b></p> <ol style="list-style-type: none"> <li>O sistema apresenta a lista dos sistemas cadastrados com as seguintes informações: <ul style="list-style-type: none"> <li>Sigla do sistema</li> <li>Nome do sistema</li> <li>Nome abreviado do sistema</li> <li>Ordem no menu</li> </ul> </li> <li>O sistema solicita que o usuário selecione um sistema da lista e informe a ação que pretende executar. Conforme a resposta recebida desvia o fluxo para um dos subfluxos abaixo: <ul style="list-style-type: none"> <li>Incluir sistema ([SB001] Incluir sistema)</li> <li>Alterar sistema ([SB002] Alterar sistema)</li> <li>Excluir sistema ([SB003] Excluir sistema),([FS007]-Sistema Gestão de Acesso não pode ser excluído)</li> </ul> </li> </ol> <p><b>[SB001] Incluir sistema</b></p> <ol style="list-style-type: none"> <li>O sistema exibe um formulário com as seguintes informações: <ul style="list-style-type: none"> <li>Sigla do sistema *</li> </ul> </li> </ol>		

- Nome do sistema\*
- Período para expiração de log de transação\* (em dias)
- Indicador de bloqueio de acesso\* (S/N)
- Menu
- Ordem de exibição \*
- Nome abreviado \*
- Descrição \*
- Figura associada \*

Obs.: Os campos assinalados com \* são obrigatórios

2. O usuário informa os dados e solicita a inclusão do sistema (*[FS001] Campos obrigatórios não preenchidos , [FS002] Sistema já cadastrado, [FS004] Sistema Gestão de Acesso não pode ser bloqueado, [FS005] Ordem de exibição do menu já utilizada* ).
3. O sistema inclui os dados do sistema e emite mensagem de operação realizada com sucesso.
4. O sistema executa o caso de uso [UC4.002] Registrar log de transação.
5. O sistema retorna ao passo 1 deste subfluxo para permitir a inclusão de um novo sistema.

#### **[SB002] Alterar sistema**

1. O sistema exibe um formulário com as seguintes informações referentes ao sistema selecionado:

- Sigla do sistema #
- Nome do sistema\*
- Período para expiração de log de transação (em dias)\*
- Indicador de bloqueio de acesso\* (S/N)
- Menu
- Ordem de exibição \*
- Nome abreviado \*
- Descrição \*
- Figura associada \*

Obs.: Os campos assinalados com \* são obrigatórios

Os campos assinalados com # não podem ser alterados

2. O usuário altera os dados e solicita efetivação (*[FS001] Campos obrigatórios não preenchidos, [FS002] Sistema já cadastrado, [FS004] Sistema Gestão de Acesso não pode ser bloqueado, [FS005] Ordem de exibição do menu já utilizada, [FS006] Sistema não foi selecionado* ).
3. O sistema altera os dados do sistema e emite mensagem de operação realizada com sucesso.
4. O sistema executa o caso de uso [UC4.002] Registrar log de transação.
5. O sistema retorna ao fluxo principal.

#### **[SB003] Excluir sistema**

1. O sistema exibe um formulário de consulta com as seguintes informações referentes ao sistema selecionado:

- Sigla do sistema
- Nome do sistema
- Período para expiração de log de transação (em dias)
- Indicador de bloqueio de acesso
- Menu
- Ordem de exibição
- Nome abreviado
- Descrição
- Figura associada

2. O sistema solicita que o usuário confirme a exclusão do sistema;
3. O usuário confirma a exclusão do sistema;
4. O sistema efetua a exclusão (*[FS003] Sistema em uso, [FS006] Sistema não foi selecionado*).
5. O sistema executa o caso de uso [UC4.002] Registrar log de transação.
6. O sistema retorna ao fluxo principal.

#### **Fluxos secundários (alternativos e de exceção)**

##### **[FS001] Campos obrigatórios não preenchidos**

Se o usuário não preencher algum dos campos obrigatórios do formulário, o sistema exibe mensagem solicitando seu preenchimento.

##### **[FS002] Sistema já cadastrado**

O sistema pesquisa se já existe sistema ou subsistema cadastrado com a mesma sigla ou o mesmo nome informado. Caso exista, exibe a mensagem: "Sistema já cadastrado".

**[FS003] Sistema em uso**

O sistema verifica se há algum subsistema, função ou perfil cadastrado para o sistema selecionado. Caso exista, o sistema exibe mensagem de erro.

**[FS004] Sistema gestão de acesso não pode ser bloqueado**

Caso o usuário tente bloquear o sistema gestão de acesso, o sistema exibe mensagem informando que este sistema não pode ser bloqueado.

**[FS005] Ordem de exibição do menu já utilizada**

Caso a ordem de exibição informada já esteja sendo utilizada por algum outro sistema, então o sistema exibe a mensagem "Ordem de exibição no menu já utilizada".

**[FS006] Sistema não foi selecionado**

Caso nenhum sistema tenha sido selecionado então é exibida a mensagem "Não foi selecionado nenhum sistema".

**[FS007] Sistema Gestão de Acesso não pode ser excluído**

Caso o usuário tente excluir o Sistema Gestão de Acesso, o sistema exibe mensagem informando que não é permitida a exclusão deste sistema

**Requisitos Não-Funcionais Associados**

**Segurança:** Obter as restrições de acesso do usuário através da execução do caso de uso UC3.006 – Obter Restrições de Acesso

**Observações**

Este caso de uso será executado eventualmente, apenas quando houver necessidade de cadastrar ou alterar os sistemas existentes.

Na próxima seção apresentamos o Subfluxo *Validar os Requisitos* e a execução de suas atividades.

### **5.4.7. SUBFLUXO: VALIDAR OS REQUISITOS**

O objetivo deste subfluxo é revisar os requisitos obtidos nos subfluxos anteriores para confirmar se eles realmente refletem as necessidades dos usuários e definem a aplicação. As próximas subseções apresentam as atividades executadas neste subfluxo: *Revisar os Requisitos* e *Legalizar os Requisitos*.

#### **5.4.7.1. Atividade: Revisar os Requisitos**

Como já foi visto no capítulo anterior, seção 4.2.7, esta atividade assemelha-se à fase de preparação da técnica *walkthrough*, onde os participantes da validação de requisitos estudam, antecipadamente à reunião de validação, os documentos *Glossário Léxico*, *Documento de Requisitos*, *Especificação de Casos de Uso*, *Documento de Requisitos de Interface* e *Especificação de Casos de Uso de Interface* para que possam preparar as questões que devem ser discutidas na reunião de validação. Estes documentos são enviados por *e-mail* aos *stakeholders* envolvidos no processo de validação com, no mínimo, dois dias de antecedência para que eles

possam revisar, cuidadosamente, os documentos e anotar os comentários a serem expostos na reunião de validação.

#### **5.4.7.2. Atividade: Aceitar os Requisitos**

No projeto prático a validação foi realizada em três reuniões, conforme havia sido planejado na atividade Planejar Validação de Requisitos.

Em todas as reuniões, são identificados alguns papéis entre os participantes, como um apresentador, responsável por conduzir o evento; um moderador, responsável por reconduzir a reunião caso a mesma desviasse do foco principal; e o registrador, responsável por redigir a ata de reunião, registrando os conflitos identificados pelos revisores e as ações sugeridas.

A condução das reuniões segue a técnica proposta no capítulo anterior, ou seja uma combinação entre as técnicas *walkthrough* e a utilização do protótipo de interfaces do usuário para ajudar os clientes e usuários a visualizarem o funcionamento da aplicação.

Nas reuniões de validação comparecem os engenheiros de requisitos, responsáveis pela elicitação e documentação, o gerente de projetos, o arquiteto de software, usuários finais e representantes dos stakeholders. As reuniões de validação são realizadas por pacotes de casos de uso e, em cada reunião, participavam os usuários finais especialistas no pacote tratado. O apresentador, inicialmente identifica o(s) pacote(s) de caso de uso a ser(em) validado(s) e apresenta uma visão geral dos respectivos requisitos. A partir daí, inicia-se a leitura do documento de requisitos (seções gerais e pacote referente à validação), seguida do documento de casos de uso. A cada caso de uso validado é apresentado o protótipo da funcionalidade correspondente, onde os participantes debatem os conflitos, as ações a serem tomadas e chegam, até, a discutir alterações para a interface, como nome e posição de campos nas telas, usabilidade, etc. Como exemplo podemos citar o caso de uso Cadastrar Sistemas, onde a validação segue a descrição do caso de uso, vide Tabela 5.24, e as telas apresentadas nas Figura 5.13 e 5.14.

Após as reuniões de validação, os registradores produzem o documento de Registro de Validação dos Requisitos, relacionando os problemas identificados e a lista de ações a serem executadas. A Tabela 5.25 apresenta parte de um Registro de Validação de Requisitos.

Com a atualização dos documentos de requisitos, é gerado um termo de aceite dos requisitos, onde os *stakeholders* responsáveis pelo projeto assinam concordando com a especificação de requisitos apresentada.

Promoções Notícias **Ajuda** Mapa do site Encerrar sessão

PAULA RENATA  
[terça, 15/07/2003] [17:15]

### Cadastrar Sistemas

Selecione um sistema na lista abaixo e indique a ação que deseja executar. Para incluir um novo sistema, selecione apenas a ação **Incluir Sistema**.

	Sigla	Nome	Nome abreviado	Ordem no Menu
<input type="radio"/>	ACS	Controle de Acesso	Controle de Acesso	1
<input type="radio"/>	TLS	Ferramentas	Ferramentas	4
<input type="radio"/>	SRV	Serviços Financeiros	Serviços Financeiros	2

próxima página >>

Selecionar a ação **Incluir sistema**

Figura 5.13- Tela Cadastrar Sistema - Protótipo de Interface

Promoções Notícias **Ajuda** Mapa do site Encerrar sessão

PAULA RENATA  
[terça, 15/07/2003] [17:24]

### Incluir Sistema

Informe os dados do sistema e clique no botão **Salvar**

Sigla do sistema*	<input type="text"/>
Nome do sistema*	<input type="text"/>
Período para expiração de log de transação*	<input type="text"/> em dias
Indicador de bloqueio de acesso*	<input type="radio"/> sim <input checked="" type="radio"/> não

#### Menu

Ordem de exibição*	<input type="text"/>	<input type="button" value="Pesquisar sistemas existentes"/>
Nome abreviado*	<input type="text"/>	
Descrição*	<input type="text"/>	
Figura associada*	<input type="text"/>	<input type="button" value="Browse..."/>

Os campos marcados com \* são obrigatórios.

Figura 5.14 - Tela Inserir Sistema - Protótipo de Interface



Tabela 5.25 - Registro de Validação de Requisitos

<b>1. Introdução</b>		
Este documento apresenta o resultado da reunião de validação dos requisitos do Sistema Gestão de Acesso para a operadora de Cartão de Crédito.		
<b>2. Participantes da Validação</b>		
Cliente	João	Comitê Executivo
	Clara	Gestor do Projeto
	Pedro	Suporte
CESAR	Ana Sofia	Gerente de Projeto
	Ana Cláudia	Engenheira de Requisitos
<b>3. Problemas identificados e recomendações</b>		
Esta seção lista os problemas encontrados durante a validação dos requisitos.		
<ul style="list-style-type: none"> <li>• Problema nos Artefatos: Na especificação do caso de uso Cadastrar Sistemas, ao serem apresentadas as relações de campos de telas, deveria haver uma explicação para cada campo. Por exemplo: Período para expiração de log de transação deve ser melhor explicado, pois corresponde ao período em que os logs do sistema são armazenados e a partir deste, deve ser executada a limpeza automática dos logs.</li> <li>• Problema no Projeto: Como o sistema de Gestão de Acesso é um gerenciador das aplicações corporativas, é necessário que, para a sua implantação, algumas informações já estejam cadastradas na sua base de dados, como por exemplo: o próprio sistema de gestão de acesso, os subsistemas e funções do sistema Gestão de Acesso, um perfil com acesso a todas as funções e um usuário com acesso a este perfil. Isto deve estar especificado como um requisito não-funcional.</li> <li>• Problema no produto: Não foram identificados.</li> </ul>		
<b>4. Lista de ações e Previsão de Correção</b>		
Os documentos de requisitos e casos de uso serão corrigidos em um prazo de 10 dias úteis, e serão enviados aos participantes desta reunião de validação. Estes, por sua vez, têm 5 dias úteis para revisar as alterações e enviar sugestões e críticas		
<b>7. Referências</b>		
Documento de Requisitos, Especificação de Casos de Uso, Documento de Requisitos de Interface, Especificação de Casos de Uso de Interface, Glossário Léxico, Atas de Reuniões.		

Concluimos a apresentação do Subfluxo *Validar Requisitos* e na próxima seção apresentamos o Subfluxo *Gerenciar Mudanças nos Requisitos* e a execução de suas atividades.

#### 5.4.8. SUBFLUXO: GERENCIAR MUDANÇAS NOS REQUISITOS

O gerenciamento de requisitos deve ser aplicado continuamente no projeto. Por este motivo, a atividade *Gerenciar Dependências* (original do RUP) é executada nos subfluxos: *Elicitar Requisitos*, *Gerenciar Escopo* e *Gerenciar Mudanças de Requisitos*, além de ser planejada no subfluxo *Planejar Engenharia de Requisitos*.

Para rastrear os requisitos do Sistema de Gestão de Acesso aplicamos a técnica de Toranzo, descrita em [Tor02-B], nos requisitos referentes ao pacote de *Configuração do Sistema de Gestão de Acessos*. Devemos salientar que a técnica de Toranzo realiza o rastreamento desde os requisitos até códigos de programas e casos de testes. Como esta dissertação foca apenas no

processo de Engenharia de Requisitos, não consideramos, no nosso experimento, a identificação de programas e casos de testes, que são tratados nas diretrizes 7 e 8, em [Tor02-B].

Outro ponto importante que deve ser considerado neste relato de experiência é o fato que validamos a primeira iteração do projeto, onde ainda não havia nenhuma solicitação de mudança registrada. Entretanto, em etapas posteriores do desenvolvimento, as solicitações de mudanças devem ser analisadas e executadas.

Na subseção seguintes apresentamos a execução da atividade *Gerenciar Dependências*.

#### **5.4.8.1. Atividade: Gerenciar Dependências**

No subfluxo Elicitar Requisitos, seção 5.4.2.6, ao executar a atividade *Gerenciar Dependências*, foram aplicadas as quatro primeiras diretrizes propostas por Toranzo<sup>14</sup> [Tor02-B] e produzidas matrizes com as expressões usadas no rastreamento: Informações do ambiente externo, Objetivos Organizacionais, Objetivos do Sistema e Subsistemas. No subfluxo Gerenciar Escopo, seção 5.4.5.2, na execução da atividade *Gerenciar Dependências*, foram aplicadas as diretrizes 5 e 6 e produzidas matrizes de requisitos, casos de uso e diagramas para o pacote *Configuração do Sistema de Gestão de Acesso*.

O próximo passo, então, é verificar se foram consideradas todas as classes a serem rastreadas. A Tabela 5.26 apresenta as classes candidatas para o Sistema de Gestão de Acessos:

---

<sup>14</sup> Oito primeiras diretrizes da Técnica de Toranzo:

Diretriz 1: Identificar as informações que podem afetar o sistema,

Diretriz 2: Identificar os objetivos, estratégias ou regras de negócios que serão rastreadas,

Diretriz 3: Incluir classes de informação da gerência de projeto no modelo de rastreamento,

Diretriz 4: Identificar subsistemas,

Diretriz 5: Identificar requisitos,

Diretriz 6: Identificar diagramas,

Diretriz 7: Identificar programas,

Diretriz 8: Identificar teste.

Tabela 5.26 - Classes candidatas ao rastreamento

Níveis de informação	Análise	Projeto	Implementação	Teste
Ambiental	InformacoesExternas			
Organizacional	ObjetivoOrganizacional			
Gerencial	ObjetivoSistema (Visão)			
Desenvolvimento	Requisitos Casos de Uso Requisitos de Interface Casos de Uso de Interface	Diagrama Subsistema		

Aplicamos, então, as seguintes diretrizes:

- Diretriz 9: Remover as classes de informação irrelevantes;
- Diretriz 10: Integrar as classes com o mesmo significado;
- Diretriz 11: Integrar novas classes;

Após aplicar essas diretrizes, observamos que não houve alterações sobre o número e significado das classes identificadas. Entretanto, como estamos considerando apenas a fase de requisitos, nos concentraremos nas classes ObjetivoSistema, Requisitos, Casos de Uso, Requisitos de Interface e Casos de Uso de Interface (ver Tabela 5.26). Uma dúvida que surgiu foi se deveríamos ou não acrescentar o *Glossário Léxico* e as *Atas de Reuniões* como uma das classes de rastreamento. Chegamos à conclusão que esta decisão depende do quão detalhado seja necessário o rastreamento. Neste caso, não adicionamos estes documentos devido ao fato de não estarmos utilizando nenhuma ferramenta que auxilie o rastreamento de requisitos.

Para estabelecer os relacionamentos entre as classes candidatas, aplicamos as seguintes diretrizes:

- Diretriz 12: Organizar as classes;
- Diretriz 13: Estabelecer Relacionamentos.

Desta forma foram descobertos os relacionamentos entre as classes candidatas. A Figura 5.15 apresenta o modelo de rastreamento para o Sistema Gestão de Acessos. Analisando o modelo, verificamos que cada relacionamento pode ser representado através de matrizes de rastreamento.

Para desenvolver as matrizes de rastreabilidade, selecionamos os relacionamentos a serem rastreados e aplicamos a seguinte diretriz:

- Diretriz 15: Definir uma matriz para cada relacionamento do modelo.

Neste trabalho preencheremos a representação matricial de alguns relacionamentos do modelo de rastreamento. Após a Figura 5.15 apresentaremos as matrizes dos relacionamentos selecionados, para o pacote Configuração do Controle de Acesso:

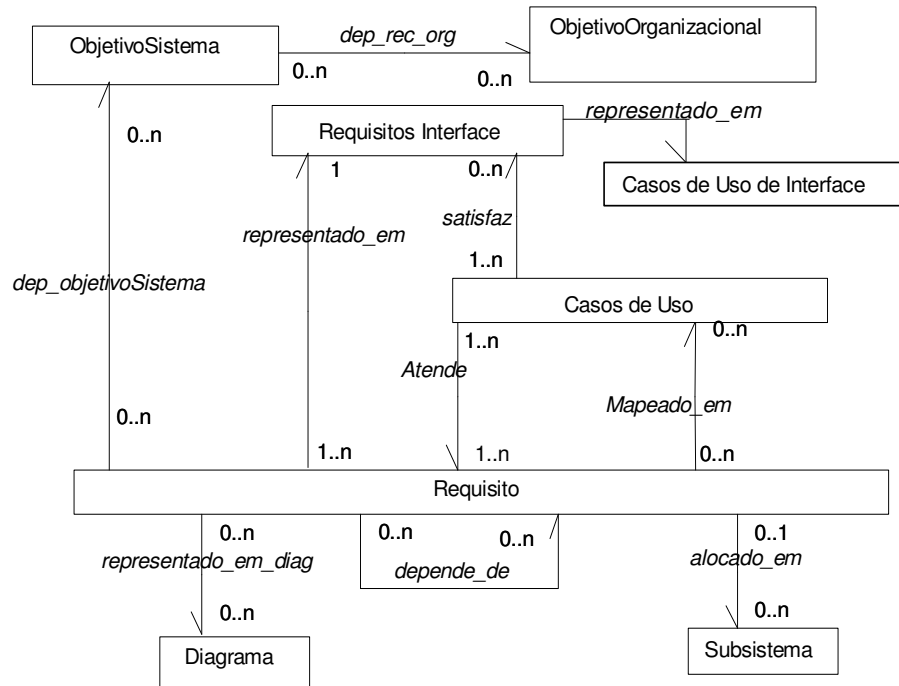


Figura 5.15 - Modelo de Rastreamento do Sistema Gestão de Acessos

1. *dep\_rec\_org*, entre as classes *ObjetivoSistema* e *ObjetivoOrganizacional*: apresentada na Tabela 5.27, mostra que objetivos de sistema (Tabela 5.13) devem ser atendidos para satisfazer os objetivos organizacionais (Tabela 5.12). Por exemplo, para satisfazer o [OBO01] *Deve obedecer uma política única de controle de acesso*, deve ser atendido o [OBS06] *Configurar a política de segurança*.
2. *dep\_objetivoSistema*, entre as classes *ObjetivoSistema* e *Requisito*: apresentada na Tabela 5.28, mostra que requisitos (Tabela 5.21) devem ser atendidos para satisfazer os objetivos do sistema (Tabela 5.13). Por exemplo, para atender o [OBS01] *Validar acesso dos usuários*, o requisito [RF1003] *Definir Restrições de Acesso a Dados* deve ser satisfeito.

3. *Mapeado\_em*, entre *Requisito e Casos de Uso*: apresentado na Tabela 5.29, mostra que casos de uso (Tabela 5.22) atendem um determinado requisito (Tabela 5.21). Por exemplo, os casos de uso [UC1.002] *Cadastrar Sistemas* e [UC1.003] *Cadastrar Subsistemas* atendem o requisito [RF1.001] *Manter Cadastro de Sistemas e Subsistemas*.
4. *Representado\_em\_diag* entre *Requisito e Diagrama*: apresentado na Tabela 5.30, mostra em que diagramas (Tabela 5.23) os requisitos (Tabela 5.21) estão sendo representados. Por exemplo no [DIAG01] *Diagrama de Casos deUso do Pacote de Configuração* estão sendo representados os requisitos [RF1.001], [RF1.002], [RF1.003], [RF1.004], [RF1.005] e [RF1.006].

*Tabela 5.27- Representação Matricial do Relacionamento dep\_reg\_org*

<i>dep_rec_org</i>	[OBS01]	[OBS02]	[OBS03]	[OBS04]	[OBS05]	[OBS06]	[OBS07]	[OBS08]	[OBS09]	[OBS10]
[OBO01]						X				
[OBO02]	X	X	X	X	X		X			
[OBO03]									X	X
[OBO04]	X			X	X		X	X		
[OBO05]	X			X						

*Tabela 5.28- Representação Matricial do Relacionamento dep\_objetivoSistema*

<i>dep_objetivoSistema</i>	[OBS01]	[OBS02]	[OBS03]	[OBS04]	[OBS05]	[OBS06]	[OBS07]	[OBS08]	[OBS09]	[OBS10]
[RF1.001]							X			
[RF1.002]							X			
[RF1.003]	X			X						
[RF1.004]				X	X					
[RF1.005]				X						
[RF1.006]						X				

*Tabela 5.29- Representação Matricial do Relacionamento Mapeado\_em*

<i>Mapeado_em</i>	[UC1.001]	[UC1.002]	[UC1.003]	[UC1.004]	[UC1.005]	[UC1.006]	[UC1.007]	[UC1.008]
[RF1.001]		X	X					
[RF1.002]				X				
[RF1.003]					X	X		
[RF1.004]							X	X
[RF1.005]							X	X
[RF1.006]	X							

Tabela 5.30- Representação Matricial do Relacionamento Representado\_em\_diag

Representado_em_Diag	[DIAG01]	[DIAG02]	[DIAG03]	[DIAG04]	[DIAG05]
[RF1.001]	X				
[RF1.002]	X				
[RF1.003]	X				
[RF1.004]	X				
[RF1.005]	X				
[RF1.006]	X				

A aplicação do rastreamento de requisitos, no sistema de Gestão de Acessos, é positiva porque, como os relacionamentos entre as informações do sistema são gradualmente transcritos para as matrizes, o entendimento do sistema torna-se cada vez maior. Além disso, tomando por base a matriz de requisitos, sempre que houver alguma mudança em requisitos, é fácil observar os impactos das mudanças sobre os diversos artefatos da aplicação.

A dificuldade que encontramos reside na falta de ferramenta, no local de trabalho, para a execução desta atividade, tornando o rastreamento bastante trabalhoso.

## 5.5. Considerações Finais

Este capítulo apresentou um relato de experiência com a aplicação do processo WRE-Process no desenvolvimento de um sistema de gestão de acessos de usuários em aplicações web, de uma operadora de cartões de crédito.

A realização deste experimento foi fundamental para verificação da efetividade do processo proposto, bem como para refinar alguns pontos que, inicialmente haviam sido definidos, mas que com o experimento, sofreram modificações. Um exemplo deste fato está no *Subfluxo Analisar e Documentar Requisitos*, onde, na proposta, refinamos o *Diagrama de Casos de Uso*, mas no nosso projeto prático não houve necessidade de realizar este refinamento, pois o diagrama já estava bem definido.

Um ponto considerado positivo para esta proposta, reside no fato da autora ter vivenciado a experiência prática de desenvolver o projeto real, onde com a análise crítica dos pontos que precisavam de melhorias e com o estudo do estado da arte no processo de engenharia de requisitos, foi derivado um processo factível de ser utilizado na indústria de software.

Uma dificuldade encontrada na validação da proposta foi na utilização da técnica de Toranzo [Tor02-B] para rastrear requisitos, haja visto ainda não existir ferramenta que automatize o processo desta técnica. Não pôde ser validado de forma satisfatória o gerenciamento de

requisitos, pois o relato de experiência tratou apenas do desenvolvimento da primeira iteração do projeto, onde não haviam sido registradas solicitações de mudanças dos requisitos.

Outro ponto importante que deve ser citado para a validação da proposta é a aplicação do WRE-Process em outros projetos reais, onde o cliente participe efetivamente, retornando críticas e sugestões sobre as atividades realizadas e artefatos produzidos.

No próximo capítulo apresentamos as considerações finais desta dissertação bem como os trabalhos relacionados e futuros.

## Capítulo 6 CONCLUSÃO E TRABALHOS FUTUROS

---

### 6.1. Principais Contribuições

O trabalho descrito nesta dissertação focou na melhoria da qualidade do processo de Engenharia de Requisitos, através da definição do WRE-Process. Este processo resultou da integração entre o RUP (Rational Unified Process) e os resultados de pesquisas acadêmicas realizadas através do Projeto West (Web Oriented Software Technology), que tem o objetivo desenvolver e gerar a tecnologia necessária para obter métodos de desenvolvimento de software para aplicações de negócios em ambientes Web. A principal motivação para o WRE-Process foi a necessidade de um processo sistemático para elicitar, analisar, documentar e gerenciar os requisitos em aplicações Web.

Com o desenvolvimento da proposta, procuramos criar um processo que fosse genérico o suficiente para atender diversos domínios e tipos de aplicações, mas que contemplasse necessidades das aplicações Web, como: integração entre requisitos funcionais e não-funcionais, elicitação e desenvolvimento de requisitos de interface, entre outros.

Podemos citar como contribuições deste trabalho:

- *levantamento do estado da arte da Engenharia de Requisitos e Engenharia de Requisitos para Web*: o segundo capítulo deste trabalho apresenta os conceitos e práticas destas áreas, relacionando trabalhos de diversos pesquisadores, também é descrito o processo genérico de Engenharia de Requisitos e detalhadas suas fases;
- *proposta de um processo para engenharia de requisitos*: Este trabalho propõe o **WRE-Process**, um processo genérico para realizar a Engenharia de Requisitos em diversos tipos de aplicações, que utiliza uma abordagem sistemática para elicitar, documentar, validar e gerenciar os requisitos de um sistema. Além disto, este modelo considera algumas características do desenvolvimento de aplicações Web, as quais requerem maior atenção em relação ao desenvolvimento de aplicações tradicionais, como questões sobre elicitação de requisitos, interface do usuário e maior ênfase nos requisitos não-funcionais. Outra característica desse processo é o fato dele estar alinhado com os objetivos da ACP (Área Chave do Processo) *Gerência de Requisitos* do CMM Nível 2, pois ele prevê a execução do gerenciamento e rastreamento dos requisitos, garantindo consistência dos artefatos



gerados. Como os requisitos são inicialmente documentados através dos documentos de Requisitos, Especificação de Casos de Uso e Modelo de Casos de Uso, esses artefatos são utilizados como base para o desenvolvimento dos modelos posteriores, tanto do fluxo de Requisitos, como de fluxos executados em etapas futuras;

- *relato de experiência da aplicação do WRE-Process no desenvolvimento de uma aplicação web*: o relato de experiência apresentado compreendeu a aplicação das práticas de engenharia e gerenciamento de requisitos propostas neste trabalho, sobre o desenvolvimento de uma solução para a gestão de acesso de usuários em um ambiente Web. A abordagem do desenvolvimento iterativo e incremental do WRE-Process, com base em aprendizados adquiridos no decorrer da aplicação do processo, permitiu a detecção de inconsistências e a realização de correções, antes da produção da versão final do processo. Desta forma, podemos dizer que o processo foi sendo refinado à medida que o aplicamos sobre um projeto real.

## 6.2. Trabalhos Relacionados

Existem vários trabalhos relacionados a essa área de pesquisa por esta envolver uma necessidade real das grandes organizações no cenário atual, onde os sistemas estão cada vez maiores, mais complexos e envolvendo uma quantidade maior de pessoas.

Nos capítulos 4 e 5 desta dissertação foram aplicados trabalhos da área de Engenharia de Requisitos na definição e validação do **WRE-Process**, como: a metodologia para elicitação de requisitos de sistemas de software, definida por Amador Duran em [Dur02], [Dur00]; a integração entre requisitos funcionais e não-funcionais de Cysneiros e Júlio Leite [Cys98], [Cys01-A], [Cys01-B]; a proposta para validação de requisitos de usuário mediante prototipagem, de Sanchez [San02]; a proposta para melhorar o rastreamento de requisitos, de Toranzo [Tor02-A], [Tor02-B]; a extensão de UML para modelagem de aplicações web (Framework W2000) de Baresi [Bar01]; entre outros.

Além dos trabalhos supracitados, que estão contextualizados na área de Engenharia de Requisitos, podemos também identificar diversos métodos para o desenvolvimento de aplicações Web que têm sido propostos:

- *Framework de Análise e Projeto Baseado no RUP para o Desenvolvimento de Aplicações Web [Ara01]* – compreende o estudo do RUP e a análise da sua adequação às necessidades de de um processo de desenvolvimento de software Web, propondo

uma adaptação do seu fluxo de Análise e Projeto para o desenvolvimento de aplicações Web.

- *Uma Extensão do Fluxo de Análise e Projeto do RUP para o Desenvolvimento de Aplicações Web [Sou02]*: consiste da criação de novas atividades e modificações de atividades já existentes no fluxo de Análise e Projeto do RUP, e visa tratar características específicas de aplicações Web, com ênfase nos aspectos de navegação e de apresentação.
- *OOHDM (Object-Oriented Hypermedia Design Method) [Ros96],[Sch98]*: compreende as atividades de modelagem conceitual, projeto navegacional, projeto de interface abstrata e implementação. Estas atividades são realizadas em um misto de estilos de desenvolvimento incremental, iterativo e baseado em protótipo.
- *WSDM (Web Site Design Method) [Tro97]*: é uma abordagem centrada no usuário que define os objetos de informação com base nos requisitos de informação dos usuários de uma aplicação web. Consiste de três fases principais: modelagem de usuário, projeto conceitual e projeto de implementação.
- *UPHD (Unified Process-based Hypermedia Systems Development) [Koc00]*: metodologia para o desenvolvimento de sistemas hipermídia, baseada no Processo Unificado.

### **6.3. Perspectiva de Trabalhos Futuros**

O processo de Engenharia de Requisitos proposto neste trabalho consistiu da criação de novos subfluxos, atividades e artefatos, em relação ao fluxo de Requisitos do RUP. Um dos subfluxos propostos, Definir Interface, contempla as atividades Projetar Conteúdo, Especificar Requisitos de Interface, Definir Protótipo de Interface e Projetar a Visibilidade da Aplicação. E esta última atividade tem como propósito a criação de esboços das interfaces gráficas para interação do usuário com a aplicação, com base nas necessidades navegacionais. Entretanto, esta atividade pode ser refinada para tratar questões relativas à organização e aparência da interface, utilizando, para tanto, técnicas específicas para interação entre homem e máquina. Outro ponto que pode ser aperfeiçoado é a criação de uma atividade para Projetar a Interface, definindo classes navegacionais e identificando e criando os caminhos navegacionais.

Este trabalho propôs a utilização da técnica de Toranzo [Tor02-B] para rastrear requisitos, entretanto, até o momento, ainda não há nenhum sistema automático que gerencie e rastreie os

requisitos. Um trabalho futuro seria a implementação de um sistema de gerência de requisitos que incluía a questão do rastreamento automático de requisitos.

Também consideramos a possibilidade de utilizar a técnica de Sanchez [San02] para a geração automática do protótipo de interface, entretanto não validamos sua execução. Uma perspectiva de trabalho seria, com o uso do WRE-Process, aplicar esta técnica, junto com as ferramentas determinadas por ela, para gerar o protótipo.

Um aspecto importante e que deve ser dado continuidade com este trabalho, é a aplicação do WRE-Process em outros experimentos, de forma que o Processo de Engenharia de Requisitos proposto possa ser melhor validado. Propomos utilizar o processo em aplicações reais, de diferentes domínios pois, desta forma, este trabalho pode amadurecer, com a identificação de possíveis falhas. É fundamental captar o retorno dos clientes, usuários e *stakeholders* a respeito do processo aplicado, colhendo críticas e sugestões.

Além do exposto acima, seria interessante observar as contribuições dessa proposta no tratamento de riscos do projeto, por exemplo, o *Glossário Léxico* poderia ser estendido para identificar e tratar os riscos.

Este trabalho adaptou apenas o fluxo de Requisitos do RUP para contemplar características das aplicações Web, entretanto, os demais fluxos de desenvolvimento do RUP também necessitam ser estendidos para torná-los mais adequados ao desenvolvimento deste tipo de software, como o fluxo de Modelagem de Negócios, Implementação, Testes e Distribuição. Além disso, seria interessante um estudo sobre as necessidades de adequação dos fluxos de suporte (Gerência de Projeto, Gerência de Configuração e Mudanças, e Ambiente).

#### **6.4. Considerações Finais**

Com base na experiência obtida a partir do desenvolvimento do Sistema de Gestão de Acesso, utilizado como estudo de caso deste trabalho, pode-se concluir que a utilização de práticas da engenharia e gerência de requisitos é de grande relevância para a obtenção de sucesso em projetos de software. Embora algumas dificuldades sejam encontradas no decorrer dos projetos, como a sobrecarga de trabalho devido à geração e manutenção dos artefatos, a equipe de desenvolvimento tende a alcançar os objetivos globais estabelecidos com a sistematização de um processo e a utilização de uma abordagem iterativa para o desenvolvimento do software.

Porém, a adoção de práticas de engenharia e gerência de requisitos não é suficiente, sendo necessário o apoio da gerência de projetos, da gerência sênior e da gerência de garantia da qualidade, que devem estar comprometidas com a disponibilização dos recursos necessários para

a realização das atividades e revisões dos artefatos produzidos no decorrer do projeto. Além disso, a alta gerência deve respaldar a obrigatoriedade da produção dos artefatos descritos nas metodologias adotadas, bem como a conformidade destes com os padrões adotados e a integridade dos mesmos durante todo o ciclo de vida do processo de software.

## REFERÊNCIAS

---

- [Ara01] Araújo, A.C.M., “Framework de Análise e Projeto Baseado no RUP para o Desenvolvimento de Aplicações Web”, Dissertação de Mestrado, UFPE, Centro de Informática, 2001.
- [Bar01] Baresi, L., Garzotto, F., Paolini, P. “Extending UML for Modeling Web Applications”, 34 th Hawaii International Conference on System Sciences, USA, 2001.
- [Bit03] Bittner, K., Spence, I., “Use Case Modeling”, Addison Wesley Pearson Education, ISBN 0-201-70913-9, 2003.
- [Bre00] Breitman, K.; “*Evolução de Cenários*”; Phd Tese, Departamento de Informática da Pontifícia Universidade Católica do Rio de Janeiro, Maio, 2000.
- [Bub93] Bubenko, J. A., “Extending The Scope of Information Moldeling”; Proc. 4th Int. Workshop on the Deductive Approach to Information Systems and DataBases, Lloret-Costa Brava, Catalonia, 20-22, pp. 73-98, September, 1993.
- [Cam02] Campêlo, G.M.C., “A utilização de métricas na Gerência de Projetos de Software: Uma abordagem focada no CMM Nível 2”, Dissertação de Mestrado, UFPE, Centro de Informática, 2002.
- [Cas99] Castro, J.; Toranzo, M., “*A Comprehensive Traceability Model to Support the Design of Interactive Systemss*”, International Workshop on Interactive System Development and Object Models - WISDOM 99; Lisboa, Portugal, 1999.
- [Cas00-A] Castro, J.; Mylopoulos, J.; Kolp, M. “Developing Agent-Oriented Information Systems for the Enterprise”, Proceedings Second International Conference on Enterprise Information Systems, Stafford, UK, July 4-7, 2000
- [Cas00-B] Castro, J; Alencar, F.; Cysneiros, G. “*Closing the GAP Between Organizational Requirements and Object Oriented Modeling*”, XIV Simpósio Brasileiro de Engenharia de Software, Joao Pessoa, Brasil, Outubro, 2000.
- [Ces03-A] Processo de Software do Centro de Estudos e Sistemas Avançados do Recife

(CESAR), “ProsCes”, propriedade e direitos reservados do CESAR, 2003.

- [Ces03-B] Documentação do Sistema de Gestão de Acessos do FiS, do CESAR, propriedade e direitos reservados do CESAR, 2003.
- [Char03] Charette, R., “The decision is in: Agile versus Heavy Methodologies”, Vol.2, No.19, 2002, Cutter Consortium, disponível em: <http://www.cutter.com/freestuff/epmu0119.html>; último acesso em 20/07/2003.
- [Chu00] Chung, L., Nixon, B., Yu, E., Mylopoulos, J., “Non-Functional Requirements in Software Engineering”, Kluwer Academic Publishers, London, 2000.
- [Cmm03] “Capability Maturity Model for Software (SW-CMM)”, SEI - Software Engineering Institute, <http://www.sei.cmu.edu/cmm/cmm.html>, Último acesso em 02/06/2003.
- [Coc00] Cockburn, A., “Writing Effective Use Cases”, Addison-Wesley Pub Co; 1st ed., ISBN: 0201702258, 2000.
- [Cod98] Coda, F., et al., “Towards a Software Engineering Approach to Web Site Development”, 9th. International Workshop on Software Specification and Design, 1998.
- [Cys98] Cysneiros, L.M., Leite, J.C.S., “Utilizando Requisitos Não-Funcionais para Análise de Modelos Orientados a Dados”, 1o. Workshop de Engenharia de Requisitos, Maringá, Outubro, 1998.
- [Cys01-A] Cysneiros, L.M., “Requisitos Não Funcionais: Da Elicitação ao Modelo Conceitual”. Phd Tese, Departamento de Informática da Pontifícia Universidade Católica do Rio de Janeiro, Fevereiro, 2001.
- [Cys01-B] Cysneiros, L.M., Leite, J.C.S., “Driving Non-Functional Requirements to Use Cases and Scenarios”, Proc. Of The XV Brazilian Symposium on Software Engineering, Rio de Janeiro, Brasil, October, 2001.
- [Cys01-C] Cysneiros, L.M., Leite, J.C.S., “Using UML to Reflect Non-Functional Requirements”, Proc of the CASCON 2001; Toronto, November, 2001.
- [Dar93] Dardene, A.; Lamsweerde, V.; Fikas, S.; “Goal-Directed Requirements Acquisition”; Science of Computer Programming, 20, pp. 3-50, 1993.

- [Dur00] Durán, A.; “*Metodología para la Elicitación de Requisitos de Sistemas Software*”; Phd Tese, Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla, Valencia, Novembro, 2000.
- [Dur02] Durán, A.; Bernardéz, B.; “*Metodología para la Elicitación de Requisitos de Sistemas Software*”; Versión 2.3. Informe Técnico LSI-2000-10; Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla, Valencia, abril, 2002.
- [Gil00] Gil, G.D., Figueroa, D.A, Oliveros, A., “*Producción del LEL en un Dominio Técnico. Informe de un caso*”, 3th. Workshop de Engenharia de Requisitos, Rio de Janeiro, Brasil, 2000.
- [Gin01] Ginige, A., Murugesan, S., “Web Engineering: An Introduction”, IEEE Multimedia, Jan/Mar, 2001
- [Had99] Hadad, G.D.S, Doorn, J.H., Kaplan, G.N., Leite, J.C.S.P., “*Enfoque Middle-Out em la Construcción e Integración de Escenarios*”, Workshop de Engenharia de Requisitos, Buenos Aires, Argentina, 1999.
- [Hol01] Holck, J., Clemmensen, T., “What makes web-development different?”, 24th. Information Systems Research Seminar in Scandinavia, 2001.
- [Jac95] Jacobson, I., *Object Oriented Software Engineering: A Use Case Driven Approach*, Addison-Wesley, 1995.
- [Jac99] Jacobson, I; J. Rumbaugh; Booch, G., “The Unified Modeling Language User Guide”, Addison Wesley Object Technology, ISBN 0-201-57168-4, 1999.
- [Joh03] Johnson, J., H., “*A Recipe for Success*”, 2000, The Standish Group International, Inc.<http://www.standishgroup.com>, Último acesso em 06/05/2003.
- [Jon03] Jones, W.L., “*Distinguish Between Application and Interface: An Alternative Approach to User Interface Development*”; 2002; Artigo disponível em The Rational Edge: <http://www.therationaledge.com>, último acesso em 10/01/2003.
- [Jul01] Juling, W., Gaedke, M., “Web Engineering”, Curso de Inverno da Universidade de Karlsruhe, 2001.
- [Kap00] Kaplan, G.N., Hadad,, G.D.S, Doorn, J.H., Leite, J.C.S.P, “*Inspección del Lexico Extendido del Language*”, 3th. Workshop de Engenharia de Requisitos, Rio de

Janeiro, Brasil, 2000.

- [Koc00] Koch, N.; “Hypermedia Systems Development based on the Unified Process”; Technical Report 003, Ludwig-Maximilians-University Munich, 2000.
- [Kru98] Kruchten, P., “The Rational Unified Process: An Introduction”, Addison Wesley Longman, Inc. , ISBN: 0-201-60459-0, 1998.
- [Kuj00] Kujala, S., Mäntylä, M, “*Studying Users for Developing Usable and Useful Products*”; In Proceedings of 1st Nordic Conference on Computer-Human Interaction, pp.1-11; Stockholm, Sweden, October, 2000.
- [Lam93] Lamsweerde, A. et al., “Goal-directed Requirements Acquisition”, Science of Computer Programming, 1993.
- [Lan95] Landauer, T.K. “*The trouble with Computers: Usefulness, Usability and Priductivity*”, MIT Press, 1995.
- [Lei97] Leite, J., Rossi, G., Maiorana, V., Balaguer, F., Kaplan, G., Hadad, G., Oliveros, A.; “*Enhancing a Requirements Baseline with Scenarios*”, Third IEEE International Symposium on Requirements Engineering - IEEE Computer Society Press. Los Alamitos, Ca, USA, pp. 44 -53, 1997.
- [Lei00] Leite, J.C.S.P, Hadad, G.D.S., Doorn, J.H., Kaplan, G.N., “*A Scenario Construction Process*”, Requirements Engineering Journal, (2000) 5:38-61, 2000.
- [Ler03] Web site do Grupo LER (Laboratório de Engenharia de Requisitos) do Centro de Informática da UFPE, disponível em : <http://www.cin.ufpe.br/~ler/west>; último acesso em 20/06/2003.
- [Med00] Medeiros, J.H., Kafure, I.M., Lula, B.Jr., “*TAOS: a Task-and-Action Oriented Framework for User’s Task Analysis in the Context of Human-Computer Interfaces Design*”; IEEE, 0-7695-0810-3/00, 2000.
- [Nev99] Neves, A.M.M, Vasconcelos, A.M.L, “*Model for Alternative Analysis*”, 8<sup>th</sup> International Conference on Human-Computer Interaction, Munich, 1999.
- [Nie95] Nielsen J.; “*Scenarios in Discount Usability Engineering. Scenario-Based Design: Envisioning Work and Technology in System Development*”; John Wiley & Sons; 1995.



- [Ome03] OME (Organization Modelling Environment), Version 3.1; disponível para download em <http://www.cs.toronto.edu/km/ome/index.html>. Último acesso em [21/07/2003](#)
- [Pad01] Pádua, W. Engenharia de Software – Fundamentos, Métodos e Padrões. Livros Técnicos e Científicos Editora ,ISBN: 85-216-1260-5, 2001.
- [Pao02] Paolini, P., Bolchini, D., “*Capturing Web Application Requirements through Goal-Oriented Analysis*”, V Workshop on Requirements Engineering, Valencia, Spain, 2002.
- [Pin03] Pinheiro, F.A.C., “Técnicas de Análise e Elicitação de Requisitos para a Web”, disponível em <http://www.cic.unb.br/~facp/facphome.html>, último acesso em 07/06/2003.
- [Pmi03] The Project Management Institute Inc., “A Guide to the Project Management Body of Knowledge – PMBOK Guide 2000 Edition”, disponível em <http://www.pmi.org/publictm/download/2000welcome.htm>, último acesso em 06/05/2003.
- [Rat02] Web site da Rational Corporation, disponível em : <http://www.rational.com>; último acesso em 20 de outubro de 2002.
- [Ros03] Rational Rose Enterprise Edition, Version 2003; disponível para download em <http://www.rational.com/tryit/evals/roseeval.jsp>. Último acesso em [21/07/2003](#).
- [Ros96] Rossi, G., “OHDM: Object-Oriented Hypermedia Design Method”, PhD Thesis, PUC-Rio, Brasil, 1996.
- [Rup02] “The Rational Unified Process”, Rational Software Corporation; versão 2002.05.00, disponível nos laboratórios do Centro de Informática da UFPE, Propriedade e direitos reservados da Rational Software Corporation.
- [San02] Santander, V.F.A.; “*Integrando Modelagem Organizacional com Use Cases na Engenharia de Requisitos*”, Phd Tese, Centro de Informática da Universidade Federal de Pernambuco, 2002.
- [San02] Sánchez, J., “*Validación de requisitos de usuario mediante técnicas de transformación de modelos y prototipación automática de interfaces de usuario*”,

Phd Tese, Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla, Valencia, Novembro, 2002.

- [Sav01] Savigne, A., Finkelstein, A., “A Framework for Requirements Engineering for Context-aware Services”, First International Workshop from Software Requirements to Architectures, 2001.
- [Sch98] Schwabe, D.; Rossi, G.; “Developing Hypermedia Applications using OOHDM”; Proceedings of Workshop on Hypermedia Development Process, Methods and Models, Hypertext, 1998.
- [Sha99] Sharp, H. Finkelstein, A., Galal, G. “*Stakeholder Identification in the Requirements Engineering Process*”, Workshop on Requirements Engineering Process, Florença, Itália, 1999.
- [Shi03] Shiple, J. “*Information Architecture Tutorial*”; Artigo disponível em [http://hotwired.lycos.com/webmonkey/design/site\\_building/tutorials/tutorial1.html](http://hotwired.lycos.com/webmonkey/design/site_building/tutorials/tutorial1.html), último acesso em 20/05/2003.
- [Som98] Sommerville, I., Kotonya, G., “Requirements Engineering: Processes and Techniques”, John Wiley & Sons Ltda, ISBN: 0-471-97208-8, 1998.
- [Som03] Sommerville, I; “Engenharia de Software”; tradução André Ribeiro; 6<sup>a</sup>. Edição, Addison Wesley, São Paulo, ISBN 85-88639-07-6, 2003.
- [Sou02] Souza, R.A.C., “Uma Extensão do Fluxo de Análise e Projeto do RUP para o Desenvolvimento de Aplicações Web”, Dissertação de Mestrado, UFPE, Centro de Informática, 2002.
- [Spe03] Spence, I.; Probasco, L.; “*Traceability Strategies for Managing Requirements with Use Cases*”; Rational Software White Paper, TP166, 2000. Disponível em <http://www.rational.com>, último acesso em 12/02/03.
- [Tor02-A] Toranzo, M., Castro, J.F.B., Mello, E., “*Uma Proposta para Melhorar o Rastreamento de Requisitos*”, Proceedings of the V Workshop on Requirements Engineering, Valencia, Spain, 2002.
- [Tor02-B] Toranzo, M.; “*Uma Proposta para Melhorar o Rastreamento de Requisitos*”; Phd Tese, Centro de Informática da Universidade Federal de Pernambuco, 2002.

- [Tro97] Troyer, O; Leune, C.; “WSDM: A User-Centered Design Method for Web Sites”; Proceedings of the 7th International World Wide Web Conference, 1997.
- [Ubi01] Projeto Ubiquitous Web Applications. “General Definition of the UWA Framework”, Relatório Técnico, 2001
- [Vas01] Vasconcelos, A.; Alcântara, A.; Valença, A., Neves, D.; “Especificação de Requisitos e Análise e Projeto Orientados a Objetos”; Versão 2.0, Parte II do Manual de Treinamento; realizado por Qualiti Software e Process e CESAR (Centro de Estudos e Sistemas avançados do Recife); Abril, 2001.
- [Yu95] Yu, E., “*Modelling Strategic Relationships for Process Reengineering*”, Phd Thesis, University of Toronto, 1995.

## Apêndice A DOCUMENTO DE REQUISITOS

---

### Apresentação

Este apêndice apresenta uma versão sintetizada do documento de requisitos do Sistema de Gestão de Acessos. Como este sistema é bastante complexo, sua documentação é volumosa; e como o objetivo desta dissertação é validar o processo de Engenharia de Requisitos WRE-Process, procuramos simplificar o documento de modo que permita ao leitor ter um entendimento geral do problema e averiguar a aplicação do template proposto em um projeto real, de modo não exaustivo. Desta forma, são descritos apenas os requisitos funcionais dos pacotes (i) *Configuração do Controle de Acesso* e (ii) *Controle de Acesso*, e alguns requisitos não-funcionais. Os pacotes *Gestão de Usuários e Trilha de Auditoria*, devido ao seu volume, não serão abordados aqui.

# **Sistema Gestão de Acessos**

## **Documento de Requisitos**

Versão 01.02  
Responsáveis:  
Ana Cláudia Beltrão (Analista de Negócios)

## Informação do Documento de Requisitos

<b>Título do documento</b>	Documento de Requisitos do Sistema Gestão de Acesso		
<b>Autor</b>	Ana Cláudia V. Beltrão Didier Analista de Negócios CESAR		
<b>Nome do arquivo</b>	Documento de Requisitos		
<b>HISTÓRICO DE REVISÕES</b>			
<b>Revisão</b>	<b>Data</b>	<b>Descrição</b>	<b>Autor(es)</b>
00	31/07/02	Versão Inicial	Ana Cláudia
01	13/08/02	Ajustes realizados após validação dos requisitos com os usuários	Ana Cláudia

## 1. Introdução

Este documento especifica o Sistema de Gestão de Acesso, fornecendo aos clientes e desenvolvedores as informações necessárias para a aceitação dos requisitos, realização do projeto, implementação, testes e homologação do sistema.

### Visão geral deste documento

Esta introdução fornece as informações necessárias para fazer um bom uso deste documento, explicitando seus objetivos e as convenções que foram adotadas no texto. As demais seções apresentam a especificação do Sistema de Gestão de Acesso e estão organizadas como descrito abaixo.

- **Seção 2 – Descrição dos Usuários:** apresenta uma visão geral dos usuários do sistema.
- **Seção 3 – Visão geral do sistema:** apresenta uma visão geral do sistema, caracterizando qual é o seu escopo e descrevendo seus usuários.
- **Seção 4 – Requisitos funcionais:** lista os requisitos funcionais do sistema, especificando seus objetivos e prioridades.
- **Seção 5 – Requisitos não funcionais:** especifica todos os requisitos não funcionais do sistema.
- **Seção 6 – Referências:** contém uma lista de referências para outros documentos relacionados.

### Convenções, termos e abreviações

#### Identificação dos Requisitos

Por convenção, a referência a requisitos funcionais é feita através do identificador do pacote onde eles estão descritos, seguidos do identificador do requisito, de acordo com o esquema abaixo:

*<nome da seção>.[RF(identificador do pacote).(identificador do requisito)]*

Por exemplo, o requisito Configuração do Controle de Acesso.[RF1.001] está descrito na seção Configuração do Controle de Acesso, em um bloco identificado pelo número [RF1.001].

Já a referência aos requisitos não funcionais é feita através do identificador do requisito, de acordo com o esquema abaixo:

*<nome da seção>.[NF(identificador do requisito)]*

Por exemplo, o requisito Segurança.[NF008] está descrito na seção de requisitos não funcionais, na seção Segurança em um bloco identificado por [NF008].

### **Prioridades dos Requisitos**

Para estabelecer a prioridade dos requisitos foram adotadas as denominações “essencial”, “importante” e “desejável”.

- **Essencial** é o requisito sem o qual o sistema não entra em funcionamento. Requisitos essenciais são requisitos imprescindíveis, que têm que ser implementados impreterivelmente.
- **Importante** é o requisito sem o qual o sistema entra em funcionamento, mas de forma não satisfatória. Requisitos importantes devem ser implementados, mas, se não forem, o sistema poderá ser implantado e usado mesmo assim por um tempo determinado e acordado com o cliente.
- **Desejável** é o requisito que não compromete as funcionalidades básicas do sistema, isto é, o sistema pode funcionar de forma satisfatória sem ele. Requisitos desejáveis são requisitos que podem ser deixados para versões posteriores do sistema caso não haja tempo hábil para implementá-los na versão que está sendo especificada. Entretanto deve-se observar estes requisitos para que a arquitetura desenvolvida seja extensível para da implementação deste requisito.

## **2. Descrição dos usuários**

### **Administrador de Segurança**

Representa a figura do responsável pela segurança dos dados da organização. O administrador de segurança é responsável pela configuração da segurança de acesso.

### **Gestor de Acesso**

Representa o gestor de acesso às funcionalidades cadastradas nos sistemas da organização. É responsável pela gestão de usuários além da criação e manutenção de perfis de acesso.

### **Usuários da área de Tecnologia**

Representam os usuários da área de Tecnologia da Operadora. Responsáveis por administrar e garantir a funcionalidade técnica do sistema; prover as manutenções necessárias nos sistemas; efetuar procedimentos operacionais do sistema; manter cadastros de características técnicas, como: cadastros de sistemas, subsistemas, funções, elementos de dados e restrições de negócio a dados.

### **Usuários Gerais**

Representam os diversos usuários com acesso aos sistemas da organização. Estes usuários podem ser representados por funcionários, estagiários, terceiros, lojistas, clientes, etc.

## 3. Visão geral do sistema

### Abrangência do Sistema

#### Escopo do Sistema

O sistema implementa o módulo de Gestão de Acesso aos sistemas corporativos da Operadora de Cartão de Crédito, sob o ambiente Web, tendo como escopo principal:

- Configurar o acesso a múltiplos sistemas corporativos da empresa, permitindo o cadastramento de sistemas, subsistemas e funções de acesso;
- Permitir a configuração dos parâmetros associados à política de segurança única para acesso aos sistemas corporativos;
- Efetuar o login único de acesso dos usuários aos sistemas corporativos sob o ambiente Web;
- Permitir a construção de menu dinâmico, de acordo com as permissões de acesso do usuário logado;
- Permitir o cadastramento de perfis de usuários, associando a cada perfil as autorizações de acesso a funcionalidades do sistema, bem como restrições de acesso a dados que devem ser aplicadas ao perfil.
- Gerenciar e administrar o acesso de usuários (associados, estagiários e terceiros) aos sistemas corporativos, permitindo efetuar o bloqueio e liberação de acesso aos sistemas, geração de senhas descartáveis, alteração de senha, cadastro e consulta de usuários bem como a desativação de acesso de usuários;
- Permitir a associação de múltiplos perfis de acesso por usuário;
- Permitir o registro de logs de acesso e de transações possibilitando a realização de trilhas de auditoria.

#### Escopo Negativo

Funcionalidades e requisitos que não estão contemplados no escopo do sistema:

- Restrição de acesso à estação física de trabalho específica;
- Restrição de acesso por horário de trabalho;
- Mecanismos para compressão de log's;
- Rotinas para monitoração de log's expirados;
- Rotinas para realização de backup's físicos;
- Rotinas para recuperação de backup's físicos.

## 4. Requisitos funcionais

### Configuração do Controle de Acesso

Nesta seção são descritas as funcionalidades de cadastramento de sistemas/subsistemas e funções, configuração de perfis de acesso e regras de acesso além de parametrização de políticas de segurança que serão aplicadas ao controle de acesso.



#### [RF1.001] Manter Cadastro de Sistemas e Subsistemas

*Este requisito permite consultar e manter o cadastro de sistemas sobre o ambiente Web e cadastro de seus subsistemas. Os itens do menu de acesso serão montados e agrupados a partir da hierarquia de sistemas e subsistemas cadastrados.*

**Prioridade:**         Essencial                       Importante                       Desejável

#### [RF1.002] Manter Cadastro de Funções

*Este requisito permite consultar e manter o cadastro das funcionalidades dos sistemas que serão acessadas pelos usuários. Uma função pode ser filha de um sistema, subsistema ou outra função.*

**Prioridade:**         Essencial                       Importante                       Desejável

#### [RF1.003] Definir Restrições de Acesso a Dados

*Este requisito permite consultar e manter o cadastro dos elementos de dados manipulados pelos sistemas corporativos além do cadastramento de suas respectivas restrições de acesso.*

**Prioridade:**         Essencial                       Importante                       Desejável

#### [RF1.004] Definir Perfis de Acesso

*Este requisito permite consultar e manter o cadastro de perfis de acesso a funcionalidades do sistema, bem como configurar as restrições de acesso a dados aplicadas ao perfil. O cadastro de perfis de acesso define um grupo de usuários com características de acesso semelhantes. Existe uma hierarquia de perfis, semelhante a uma estrutura em árvore, onde um perfil filho só pode acessar um subconjunto de funcionalidades do perfil pai.*

**Prioridade:**         Essencial                       Importante                       Desejável

#### [RF1.005] Consultar Perfis de Acesso

*Este requisito permite realizar consulta aos perfis de acesso cadastrados, sob critérios a serem escolhidos pelo usuário, possibilitando visualizar detalhes de cada perfil, a lista das funções com acesso liberado bem como a lista das restrições de acesso dos elementos de dados.*

**Prioridade:**         Essencial                       Importante                       Desejável

#### [RF1.006] Configurar Política de Segurança

*Este requisito permite efetuar a configuração dos parâmetros associadas à política de segurança única aplicada aos sistemas corporativos, tais como: tamanho mínimo e máximo de senha, período para expiração de senha, tempo máximo com sessão ociosa, número máximo de tentativas de acesso erradas para bloquear login, quantidade de dias sem acesso aos sistemas.*

**Prioridade:**         Essencial                       Importante                       Desejável

## Controle de Acesso

Nesta seção são descritas as funcionalidades do pacote “Controle de Acesso”, que implementarão o controle de acesso ao sistema, tais como login/logout, obter/gravar restrições de acesso e montar telas de menus dinâmicos.

### [RF3.001] Efetuar Login

*Este requisito valida a identificação do usuário através de seu login e senha, verifica sua situação junto aos parâmetros de segurança e monta o menu de opções de acordo com as funcionalidades permitidas pelos seus perfis de acesso. Ao efetuar o login é registrado a abertura do log de acesso do usuário ao sistema.*

**Prioridade:**         Essencial                       Importante                       Desejável

### [RF3.002] Encerrar Sessão do Usuário

*Este requisito permite que o usuário efetue logout ao sistema e encerre a sua sessão. Ao efetuar o logout, o sistema registra o encerramento do log de acesso e retorna para a tela de login.*

**Prioridade:**         Essencial                       Importante                       Desejável

### [RF3.003] Construir Menu Dinâmico

*Este requisito possibilita que o menu de acesso seja construído dinamicamente, de acordo com os perfis de acesso do usuário. Só aparecem no menu, os sistemas/subsistemas e funções que o usuário tem permissão de acesso.*

**Prioridade:**         Essencial                       Importante                       Desejável

### [RF3.004] Obter Restrições de Acesso

*Este requisito disponibiliza às funções dos sistemas corporativos um serviço para obtenção das restrições de acesso cadastradas para os elementos de dados manipulados pelas funções e relativos aos perfis de acesso do usuário logado.*

**Prioridade:**         Essencial                       Importante                       Desejável

### [RF3.005] Validar Restrições de Acesso

*Este requisito disponibiliza às funções dos sistemas corporativos um serviço para efetuar as validações de acesso a dados cadastradas para os elementos de dados manipulados pela função e relativos aos perfis de acesso do usuário logado.*

**Prioridade:**         Essencial                       Importante                       Desejável

### [RF3.006] Gravar Restrições de Acesso

*Este requisito disponibiliza às funções dos sistemas corporativos um serviço para efetuar a persistência das propriedades associadas às restrições de acesso a dados.*

**Prioridade:**         Essencial                       Importante                       Desejável

## Trilha de auditoria

Nesta seção são descritas as funcionalidades relacionadas com os registros, consultas e limpeza de logs de acesso e transação de sistemas.

### [RF4.001] Registrar Logs de Acesso

*Este requisito permite registrar os logs de acessos (login, logout, bloqueio e desbloqueio) dos usuários aos sistemas. Os seguintes dados devem ser armazenados nos logs de acesso: usuário, data/hora (timestamp), ip da máquina de onde o cliente está conectado e operação realizada.*

**Prioridade:**       Essencial                       Importante                       Desejável

### [RF4.002] Registrar Logs de Transações

*Este requisito disponibiliza às funções dos sistemas corporativos um serviço para registro de acesso dos usuários às transações dos sistemas.*

**Prioridade:**       Essencial                       Importante                       Desejável

### [RF4.003] Registrar Logs de Dados

*Este requisito efetua o registro das alterações de informações realizadas nas transações dos sistema, sendo guardados seus conteúdos anteriores e respectivos valores alterados.*

**Prioridade:**       Essencial                       Importante                       Desejável

### [RF4.004] Consultar Logs

*Este requisito permite a consulta aos registros de logs de acesso, transações e dados sob critérios a serem escolhidos pelo usuário.*

**Prioridade:**       Essencial                       Importante                       Desejável

## 5. Requisitos não Funcionais

### Interface

Os requisitos não funcionais associados à interface com o usuário são descritos no [Documento de Requisitos de Interface](#).

### Segurança

Esta seção descreve os requisitos não funcionais associados à segurança, confidencialidade das informações, severidade de falhas do sistema e habilidade de recuperação das mesmas, bem como à corretude do sistema.

### [NF001] Confidencialidade

*O sistema deverá garantir o sigilo das informações, criando mecanismos que impeçam pessoas não autorizadas a terem acesso aos dados.*

**Prioridade:**         Essencial                       Importante                       Desejável

**Requisito(s) associado(s):** Todos os requisitos descritos na seção 4 deste documento.

#### **[NF002] Disponibilidade**

*O sistema deve estar disponível 7 dias por semana, 24 horas por dia, excetuando-se horários previamente negociados para manutenção preventiva, backups, upgrades, manutenção do software, hardware ou alterações que tornem imperativa a paralisação dos serviços oferecidos pelo menor tempo possível.*

**Prioridade:**         Essencial                       Importante                       Desejável

**Requisito(s) associado(s):** Todos os requisitos descritos na seção 4 deste documento.

#### **[NF003] Integridade dos Dados**

*O sistema deve garantir integridade, completude e precisão das informações por ele armazenadas.*

**Prioridade:**         Essencial                       Importante                       Desejável

**Requisito(s) associado(s):** Todos os requisitos descritos na seção 4 deste documento.

#### **[NF004] Criptografia de senha**

*As senhas de acesso dos usuários devem ser criptografadas, garantindo assim o sigilo das informações trafegadas.*

**Prioridade:**         Essencial                       Importante                       Desejável

**Requisito(s) associado(s):** [RF2.003], [RF2.011], [RF3.001]

#### **[NF005] Timeout de sessão**

*O sistema deverá contemplar o tratamento de sessão expirada por timeout, isto é, a sessão do usuário estar ociosa a um período maior ou igual ao especificado nos parâmetros de segurança.*

**Prioridade:**         Essencial                       Importante                       Desejável

**Requisito(s) associado(s):** Todos os requisitos descritos na seção 4 deste documento.

### **Performance**

Esta seção descreve os requisitos não funcionais associados à eficiência, uso de recursos e tempo de resposta do sistema.

### [NF006] Tempo de Resposta

*O sistema deverá prover respostas rápidas aos usuários, entretanto, é importante considerar o novo ambiente computacional sobre o qual será instalado o sistema, além da infra-estrutura de acesso dos usuários. Devem ser avaliados os limite mínimos de performance para que o sistema seja efetivamente implantado.*

**Prioridade:**         Essencial                       Importante                       Desejável

**Requisito(s) associado(s):** Todos os requisitos descritos na seção 4 deste documento.

### [NF007] Número de Acessos

*O sistema deverá suportar elevado número de acessos simultâneos, sem que haja perda de desempenho da aplicação.*

**Prioridade:**         Essencial                       Importante                       Desejável

**Requisito(s) associado(s):** Todos os requisitos descritos na seção 4 deste documento.

## Processo

Esta seção descreve os requisitos não funcionais associados ao processo de desenvolvimento do sistema.

### [NF011] WRE-Process

*O sistema deverá seguir o processo WRE-Process para a fase de requisitos.*

**Prioridade:**         Essencial                       Importante                       Desejável

### [NF012] RUP

*O processo a ser seguido para o desenvolvimento de software deve ser o RUP, exceto durante a fase de Engenharia de Requisitos, onde deve ser seguido o WRE-Process.*

**Prioridade:**         Essencial                       Importante                       Desejável

## 6. Referências

Esta seção lista os documentos relacionados a este documento, com links para os mesmos:

1. [Atas de reuniões de levantamento de requisitos;](#)
2. [Documento de Requisitos de Interface;](#)
3. [Especificação dos Casos de Uso;](#)
4. [Glossário Léxico;](#)
5. [Diagrama dos Casos de Uso;](#)

## Apêndice B ESPECIFICAÇÃO DE CASOS DE USO

---

### Apresentação

Este apêndice apresenta uma versão sintetizada da Especificação de Casos de Uso do Sistema de Gestão de Acessos. Como este sistema é bastante complexo, sua documentação é volumosa; e como o objetivo desta dissertação é validar o processo de Engenharia de Requisitos WRE-Process, procuramos simplificar o documento de modo que permita ao leitor ter um entendimento geral do documento e avaliar a aplicação do template proposto em um projeto real, de modo não exaustivo. Desta forma, resolvemos trabalhar apenas com o pacote de casos de uso *Configuração do Controle de Acesso*, onde todos os casos de uso são enumerados, mas apenas alguns são detalhados (consideramos apenas os casos de uso tratados na primeira iteração do projeto). Os casos de uso dos pacotes *Controle de Acesso*, *Gestão de Usuários* e *Trilha de Auditoria*, devido ao seu volume, não serão abordados aqui.

# **Sistema Gestão de Acessos Especificação de Casos de Uso**

Versão 01.02  
Responsáveis:  
Ana Cláudia Beltrão (Analista de Negócios)  
01/02/03

## Informação do Documento de Especificação de Casos de Uso

<b>Título do documento</b>	Especificação de Casos de Uso do Sistema Gestão de Acesso		
<b>Autor</b>	Ana Cláudia Beltrão Analista de Negócios CESAR		
<b>Nome do arquivo</b>	Especificação_Casos_Uso		
<b>HISTÓRICO DE REVISÕES</b>			
<b>Revisão</b>	<b>Data</b>	<b>Descrição</b>	<b>Autor</b>
00	03/08/02	Versão Inicial	Ana Cláudia
01	13/08/02	Versão após validação dos requisitos com os usuários	Ana Cláudia

## 1. Introdução

Este documento especifica os casos de uso do Sistema Gestão de Acessos, descrevendo os fluxos de eventos, entradas e saídas de cada caso de uso a ser implementado.

### Visão geral deste documento

Esta introdução fornece as informações necessárias para fazer um bom uso deste documento, explicitando seus objetivos e as convenções que foram adotadas no texto. As demais seções apresentam a especificação do módulo de Controle de Acesso e estão organizadas como descrito abaixo.

**Seção 2 – Casos de Uso:** apresenta os casos de uso do sistema.

**Seção 3 – Referências:** contém uma lista de referências para outros documentos relacionados.

### Convenções, termos e abreviações

A correta interpretação deste documento exige o conhecimento de algumas convenções e termos específicos, que são descritos a seguir.

#### Identificação dos casos de uso

Por convenção, a referência a casos de uso é realizada de acordo com o esquema abaixo:

*([identificador do pacote.identificador do caso de uso] – nome do caso de uso)*

Por exemplo, o caso de uso ([UC1.001] – Configurar política de segurança) está descrito no primeiro pacote, em um bloco identificado pelo número [UC1.001] e seu nome é “Configurar política de segurança”.

#### Identificação de subfluxos

A descrição dos casos de uso inclui referência a subfluxos, que serão identificados como segue:

*([identificador do subfluxo] – nome do subfluxo)*

Por exemplo, quando for encontrada a referência [SB001 – Alterar usuário ] na descrição do caso de uso, isto implica um redirecionamento de fluxo para os passos descritos no subfluxo “Alterar usuário”.

#### Identificação de fluxos secundários

A descrição dos casos de uso inclui referência a fluxos secundários (alternativos e de exceção), que serão identificados como segue:

*([identificador do fluxo secundário] – nome do fluxo secundário)*

Por exemplo, quando for encontrada a referência [FS001 – Campos obrigatórios não informados] na descrição do caso de uso, isto indica que, dependendo das condições de execução do caso de uso, o fluxo secundário “Campos obrigatórios não informados” pode ser executado.



## 2. Casos de uso

### Configuração do Controle de Acesso

Os casos de uso desta seção descrevem as funcionalidades de cadastramento de sistemas/subsistemas e funções, configuração de perfis de acesso e regras de acesso além de parametrização de políticas de segurança que serão aplicadas ao controle de acesso.

#### [UC1.001] Configurar política de segurança

Objetivo: <b>O sistema deve apresentar uma tela com os parâmetros da política de segurança da organização e o usuário deverá mantê-los.</b>		
Ator Responsável: <b>Administrador de Segurança</b>		
Requisitos Atendidos: <b>RF1.006</b>		
<b>Contexto</b>		
<b>Eventos que Disparam o Caso de Uso</b>	<b>Entradas</b>	<b>Saídas</b>
Selecionar função Configurar política de segurança	Perfil do usuário logado	

**Pré condições:** o usuário logado deve ter um perfil de administrador de segurança.

**Pós condições:** se o caso de uso for realizado com sucesso as políticas de segurança serão atualizadas no sistema.

#### [UC1.002] Cadastrar sistemas

Objetivo: <b>Manter o cadastro dos sistemas gerenciados pelo sistema Gestão de Acesso</b>		
Ator Responsável: <b>Administrador de Segurança</b>		
Requisitos Atendidos: <b>RF1.001</b>		
<b>Contexto</b>		
<b>Eventos que Disparam o Caso de Uso</b>	<b>Entradas</b>	<b>Saídas</b>
Selecionar função Cadastrar sistemas		

**Pré condições:** o usuário logado deve ter perfil de administrador de segurança.

**Pós condições:** se o caso de uso for realizado com sucesso os sistemas serão incluídos, alterados ou excluídos no controle de acesso.

#### Fluxo de eventos principal

3. O sistema apresenta a lista dos sistemas cadastrados com as seguintes informações:
  - Sigla do sistema
  - Nome do sistema
  - Nome abreviado do sistema
  - Ordem no menu
4. O sistema solicita que o usuário selecione um sistema da lista e informe a ação que pretende executar. Conforme a resposta recebida desvia o fluxo para um dos subfluxos abaixo:
  - Incluir sistema ([SB001] Incluir sistema)
  - Alterar sistema ([SB002] Alterar sistema)
  - Excluir sistema ([SB003] Excluir sistema),([FS007]-Sistema Gestão de Acesso não pode ser excluído)

### **[SB001] Incluir sistema**

6. O sistema exibe um formulário com as seguintes informações:
  - Sigla do sistema \*
  - Nome do sistema\*
  - Período para expiração de log de transação\* (em dias)
  - Indicador de bloqueio de acesso\* (S/N)
  - Menu
  - Ordem de exibição \*
  - Nome abreviado \*
  - Descrição \*
  - Figura associada \*

Obs.: Os campos assinalados com \* são obrigatórios
7. O usuário informa os dados e solicita a inclusão do sistema (*[FS001] Campos obrigatórios não preenchidos*, *[FS002] Sistema já cadastrado*, *[FS004] Sistema Gestão de Acesso não pode ser bloqueado*, *[FS005] Ordem de exibição do menu já utilizada* ).
8. O sistema inclui os dados do sistema e emite mensagem de operação realizada com sucesso.
9. O sistema executa o caso de uso [UC4.002] Registrar log de transação.
10. O sistema retorna ao passo 1 deste subfluxo para permitir a inclusão de um novo sistema.

### **[SB002] Alterar sistema**

6. O sistema exibe um formulário com as seguintes informações referentes ao sistema selecionado:
  - Sigla do sistema #
  - Nome do sistema\*
  - Período para expiração de log de transação (em dias)\*
  - Indicador de bloqueio de acesso\* (S/N)
  - Menu
  - Ordem de exibição \*
  - Nome abreviado \*
  - Descrição \*
  - Figura associada \*

Obs.: Os campos assinalados com \* são obrigatórios  
Os campos assinalados com # não podem ser alterados
7. O usuário altera os dados e solicita efetivação (*[FS001] Campos obrigatórios não preenchidos*, *[FS002] Sistema já cadastrado*, *[FS004] Sistema Gestão de Acesso não pode ser bloqueado*, *[FS005] Ordem de exibição do menu já utilizada*, *[FS006] Sistema não foi selecionado* ).
8. O sistema altera os dados do sistema e emite mensagem de operação realizada com sucesso.
9. O sistema executa o caso de uso [UC4.002] Registrar log de transação.
10. O sistema retorna ao fluxo principal.

### **[SB003] Excluir sistema**

7. O sistema exibe um formulário de consulta com as seguintes informações referentes ao sistema selecionado:
  - Sigla do sistema
  - Nome do sistema
  - Período para expiração de log de transação (em dias)
  - Indicador de bloqueio de acesso
  - Menu
  - Ordem de exibição
  - Nome abreviado
  - Descrição
  - Figura associada
8. O sistema solicita que o usuário confirme a exclusão do sistema;
9. O usuário confirma a exclusão do sistema;
10. O sistema efetua a exclusão (*[FS003] Sistema em uso*, *[FS006] Sistema não foi selecionado*).

11. O sistema executa o caso de uso [UC4.002] Registrar log de transação.
12. O sistema retorna ao fluxo principal.

**Fluxos secundários (alternativos e de exceção)**

**[FS001] Campos obrigatórios não preenchidos**

Se o usuário não preencher algum dos campos obrigatórios do formulário, o sistema exibe mensagem solicitando seu preenchimento.

**[FS002] Sistema já cadastrado**

O sistema pesquisa se já existe sistema ou subsistema cadastrado com a mesma sigla ou o mesmo nome informado. Caso exista, exibe a mensagem: "Sistema já cadastrado".

**[FS003] Sistema em uso**

O sistema verifica se há algum subsistema, função ou perfil cadastrado para o sistema selecionado. Caso exista, o sistema exibe mensagem de erro.

**[FS004] Sistema gestão de acesso não pode ser bloqueado**

Caso o usuário tente bloquear o sistema gestão de acesso, o sistema exibe mensagem informando que este sistema não pode ser bloqueado.

**[FS005] Ordem de exibição do menu já utilizada**

Caso a ordem de exibição informada já esteja sendo utilizada por algum outro sistema, então o sistema exibe a mensagem "Ordem de exibição no menu já utilizada".

**[FS006] Sistema não foi selecionado**

Caso nenhum sistema tenha sido selecionado então é exibida a mensagem "Não foi selecionado nenhum sistema".

**[FS007] Sistema Gestão de Acesso não pode ser excluído**

Caso o usuário tente excluir o Sistema Gestão de Acesso, o sistema exibe mensagem informando que não é permitida a exclusão deste sistema

**Requisitos Não-Funcionais Associados**

**Segurança:** Obter as restrições de acesso do usuário através da execução do caso de uso UC3.006 – Obter Restrições de Acesso

**Observações**

Este caso de uso será executado eventualmente, apenas quando houver necessidade de cadastrar ou alterar os sistemas existentes.

<b>[UC1.003] Cadastrar subsistemas</b>
--

<b>Objetivo: Manter o cadastro dos subsistemas, associados aos sistemas já cadastrados.</b>		
<b>Ator Responsável: Gestor de Acesso</b>		
<b>Requisitos Atendidos: RF1.001</b>		
<b>Contexto</b>		
<b>Eventos que Disparam o Caso de Uso</b>	<b>Entradas</b>	<b>Saídas</b>
Selecionar função cadastrar subsistemas		

**Pré condições:** o usuário logado deve ter o perfil de um gestor de acesso e só devem ser apresentados os sistemas gerenciados pelo gestor de acesso.

**Pós condições:** se o caso de uso for realizado com sucesso os subsistemas serão incluídos, alterados ou excluídos no sistema.

**[UC1.004] Cadastrar funções**Objetivo: **Manter o cadastro das funcionalidades associadas a cada subsistema.**Ator Responsável: **Gestor de Acesso**Requisitos Atendidos: **RF1.002**

Contexto		
Eventos que Disparam o Caso de Uso	Entradas	Saídas
Selecionar função Cadastrar funções		

**Pré condições:** o usuário logado deve estar associado a um perfil que tenha permissão de gestor de acesso e só devem ser apresentados conjunto de sistemas que são gerenciados pelo gestor de acesso.

**Pós condições:** se o caso de uso for realizado com sucesso a função será incluída, alterada ou excluída no sistema.

**[UC1.005] Cadastrar elementos de dados**Objetivo: **Manter o cadastro dos elementos de dados manipulados pelas funções dos sistemas gerenciados pelo sistema Gestão de Acesso.**Ator Responsável: **Gestor de Acesso**Requisitos Atendidos: **RF3.005**

Contexto		
Eventos que Disparam o Caso de Uso	Entradas	Saídas
Selecionar função Cadastrar elementos de dados		

**Pré condições:** o usuário logado deve estar associado a um perfil de gestor de acesso que tenha permissão de acesso a esta funcionalidade.

**Pós condições:** se o caso de uso for realizado com sucesso os elementos de dados e suas restrições de acesso serão registradas no sistema.

**[UC1.006] Associar elemento de dados à função**Objetivo: **Permitir que sejam associados às funcionalidades cadastradas os elementos de dados por elas manipulados. Esta associação é a base para que possam ser validadas as restrições de acesso dos usuários no momento da execução de cada função do sistema.**Ator Responsável: **Gestor de Acesso**Requisitos Atendidos: **RF3.005**

Contexto		
Eventos que Disparam o Caso de Uso	Entradas	Saídas
Selecionar função Associar elemento de dados à função	Função a ser associado ao elemento	

**Pré condições:** o usuário logado deve estar associado a um perfil que tenha permissão de gestor de acesso e deve ter selecionado a função que deseja associar aos elementos de dados.

**Pós condições:** se o caso de uso for realizado com sucesso a função será associada aos elementos de dados selecionados.

**[UC1.007] Cadastrar perfil de acesso**Objetivo: **Manter o cadastro dos perfis de acesso de usuários.**Ator Responsável: **Gestor de Acesso**Requisitos Atendidos: **RF1.004**

Contexto		
Eventos que Disparam o Caso de Uso	Entradas	Saídas
Selecionar função Cadastrar perfil de acesso		

**Pré condições:** o usuário logado deve estar associado a um perfil de gestor de acesso ou de administrador de segurança. Só devem ser apresentados o conjunto de sistemas que são gerenciados pelo gestor de acesso.

**Pós condições:** se o caso de uso for realizado com sucesso o perfil de acesso será incluído, alterado ou excluído do sistema.

### **Fluxo de eventos principal**

1. O sistema apresenta tela com estrutura hierárquica em árvore, contendo os sistemas e seus respectivos perfis pais e perfis filhos cadastrados. Nesta árvore só devem ser exibidos os sistemas que o usuário logado é gestor (de acordo com as informações selecionadas pelo componente de validação). Um perfil está associado a um único sistema e poderá ter perfis filhos associados a ele. Os perfis filhos acessam um subconjunto das funcionalidades do seu perfil pai e herdam as restrições de acesso do perfil pai.
2. O usuário pode expandir ou comprimir a estrutura mostrada.
3. O sistema solicita que o usuário selecione um sistema (quando desejar incluir um perfil pai) ou um perfil exibido na árvore (para as demais opções) e informe a ação que pretende executar. Conforme a resposta o sistema desvia o fluxo para um dos subfluxos abaixo:
  - Incluir perfil (mestre) ([FS011] Sistema não selecionado, [SB001] Incluir perfil pai (mestre))
  - Incluir perfil filho ([FS010] Perfil pai não selecionado, [SB002] Incluir perfil filho)
  - Alterar perfil ([FS012] Perfil não selecionado, [SB003] Alterar perfil)
  - Excluir perfil ([FS012] Perfil não selecionado, [FS001] Existe usuário com perfil, [FS002] Existe perfil filho), ([SB004] Excluir perfil)
  - Detalhar perfil ([FS012] Perfil não selecionado, [SB005] Detalhar perfil)
  - Configurar restrições ([FS012] Perfil não selecionado, [SB006] Configurar restrições de acesso)

#### **[SB001] Incluir perfil pai (mestre)**

1. O sistema exibe uma janela com as seguintes informações e solicita que o usuário realize o cadastramento do perfil:
  - Sistema #
  - Tipo do Perfil \* (Administrador de Segurança (0) , Gestor de Acesso (1) , Usuário (2))
  - Descrição do perfil\*
  - Lista de Funções com acesso liberado (lista hierárquica com todas as funções do sistema pai permitindo que o usuário marque/desmarque o acesso a uma função).Os campos assinalados com \* são obrigatórios.
2. O usuário informa os dados do perfil e assinala as funções para as quais o perfil terá acesso e solicita a inclusão do perfil. ([FS003] Campos obrigatórios não preenchidos, [FS004] Nenhuma função assinalada, [FS007] Função de gestor não pode ser atribuída ao perfil, [FS008] Função de administrador não pode ser atribuída ao perfil).
3. O sistema inclui os dados do perfil (sistema, descrição, tipo do perfil, lista de funções com acesso) e emite mensagem de operação realizada com sucesso.
4. O sistema inclui as restrições de acesso aos elementos de dados manipulados pelas funções associadas ao perfil. Estas restrições são inclusas com acesso negado (flags com valor zero).
5. O sistema inclui as restrições de acesso a atributos dos elementos de dados manipulados pelas funções associadas ao perfil. Estas restrições são inclusas com acesso negado (flags com valor zero).
6. O sistema pesquisa as restrições de negócio associadas aos elementos de dados manipulados pelas funções associadas ao perfil e as adiciona para o novo perfil.
7. O sistema executa o caso de uso [UC4.002] Registrar log de transação.

#### **[SB002] Incluir perfil filho**

1. O sistema exibe uma janela com as seguintes informações e solicita que o usuário realize o cadastramento do perfil:
  - Sistema #
  - Perfil pai #

- Tipo do Perfil # ( herdado do perfil pai)
- Descrição do perfil\*
- Lista de Funções com acesso liberado (lista hierárquica com todas as funções do perfil pai permitindo que o usuário marque/desmarque o acesso a uma função).

Os campos assinalados com \* são obrigatórios.

Os campos assinalados com # não podem ser alterados.

2. O usuário informa os dados do perfil e assinala as funções para as quais o perfil terá acesso e solicita a inclusão. ([FS003] Campos obrigatórios não preenchidos, [FS002] Nenhuma função assinalada).
3. O sistema inclui os dados do perfil (sistema, perfil pai, descrição, tipo do perfil, lista de funções acessadas, restrições do perfil pai (acesso, negócio e atributos), e emite mensagem de operação realizada com sucesso.
4. O sistema executa o caso de uso [UC4.002] Registrar log de transação.

### **[SB003] Alterar perfil**

1. O sistema exibe uma janela com as seguintes informações e solicita que o usuário realize a alteração do perfil:

- Sistema #
- Perfil Pai #
- Descrição do Perfil\*
- Tipo do Perfil # (Administrador de Segurança (0), Gestor de Acesso (1), Usuário (2))
- Lista de Funções com acesso liberado: lista hierárquica com todas as funções cadastradas para o perfil pai (ou para o sistema, caso o perfil alterado seja um perfil pai/mestre), sendo que as funções já associadas ao perfil alterado devem estar marcadas. O sistema permite que o usuário marque/desmarque o acesso a uma função.

Os campos assinalados com \* são obrigatórios.

Os campos assinalados com # não podem ser alterados.

2. O usuário altera as informações do perfil solicita sua efetivação. ([FS003] Campos obrigatórios não preenchidos, [FS004] Nenhuma função assinalada, [FS005] Perfil sem permissão de alteração, [FS006] Perfil filho não pode alterar perfil pai, [FS007] Função de gestor não pode ser atribuída ao perfil, [FS008] Função de administrador de segurança não pode ser atribuída ao perfil).
3. O sistema grava as alterações realizadas no perfil. ([FS009] Propagar alteração nos perfis filhos)
4. O sistema emite mensagem de operação realizada com sucesso.
5. O sistema executa o caso de uso [UC4.002] Registrar log de transação.

### **[SB004] Excluir perfil**

1. O sistema exibe um formulário de consulta com as seguintes informações:

- Sistema #
- Perfil Pai #
- Descrição do Perfil\*
- Tipo do Perfil # (Administrador de Segurança (0), Gestor de Acesso (1), Usuário (2))
- Lista de Funções com acesso liberado (lista hierárquica com todas as funções cadastradas).

2. O sistema solicita que o usuário confirme a exclusão do perfil;
3. O usuário confirma a exclusão do perfil;
4. O sistema efetua a exclusão do perfil e das associações existentes com o mesmo. ([FS001] Existe usuário com perfil, [FS002] Existe perfil filho).
5. O sistema executa o caso de uso [UC4.002] Registrar log de transação.

### **[SB005] Detalhar perfil**

1. O sistema executa o caso de uso [UC1.009] Detalhar perfil.

### **[SB006] Configurar restrições de acesso**

1. O sistema executa o caso de uso [UC1.008] Configurar restrições por perfil.

### **Fluxos secundários (alternativos e de exceção)**

***[FS001] Existe usuário associado ao perfil***

Caso o usuário logado tente excluir perfil que esteja associado a algum usuário, o sistema exibe mensagem de erro e não permite a exclusão.

***[FS002] Existe perfil filho***

Caso o usuário logado tente excluir perfil que seja pai de algum outro, o sistema exibe mensagem de erro e aborta a operação.

***[FS003] Campos obrigatórios não preenchidos***

Se o usuário não preencher algum dos campos obrigatórios do formulário o sistema exibe mensagem solicitando seu preenchimento.

***[FS004] Nenhuma função assinalada***

O sistema verifica se o usuário não selecionou nenhuma função para o perfil exibindo a seguinte mensagem de erro “Nenhuma função assinalada para o perfil”.

***[FS005] Perfil sem permissão para alteração***

O sistema verifica se o perfil selecionado é igual ao do usuário logado e se este perfil possui um perfil pai associado. Em caso afirmativo, usuário não tem permissão para manter o seu próprio perfil e o sistema exibe a seguinte mensagem de erro “Perfil sem permissão de alteração”.

***[FS006] Perfil filho não pode alterar perfil pai***

O sistema verifica se o perfil selecionado é pai do perfil do usuário logado. Em caso afirmativo, usuário não tem privilégio para manter o seu perfil pai e o sistema exibe a seguinte mensagem de erro “Perfil filho não pode alterar perfil pai”.

***[FS007] Função de gestor não pode ser atribuída ao perfil***

O sistema verifica se alguma função exclusiva para perfil de gestor foi assinalada e o perfil que está sendo cadastrado não corresponde a um perfil de gestor (Tipo do perfil = 1). Em caso afirmativo o sistema exibe a seguinte mensagem de erro “Função de gestor não pode ser atribuída ao perfil”.

***[FS008] Função de administrador não pode ser atribuída ao perfil***

O sistema verifica se alguma função exclusiva para perfil de administrador de segurança foi assinalada e o perfil que está sendo cadastrado não corresponde a um perfil de administrador (Tipo do perfil = 0). Em caso afirmativo o sistema exibe a seguinte mensagem de erro “Função de administrador não pode ser atribuída ao perfil”.

***[FS009] Propagar alteração nos perfis filhos***

O sistema verifica se o perfil em alteração possui perfis filhos.

Para cada função desassociada ao perfil pai, o sistema desassocia em cada perfil filho (processo recursivo). Caso o perfil filho fique sem funções então o sistema efetua a exclusão do perfil filho e realiza as desassociações existentes com usuários.

Para as novas funções associadas, o sistema questiona se deverá propagar as alterações com seus perfis filhos. Em caso positivo, para cada nova função associada ao perfil pai, o sistema propaga recursivamente esta liberação para os perfis filhos.

***[FS010] Perfil pai não selecionado***

Caso o perfil pai não tenha sido selecionado, então será apresentada a seguinte mensagem de erro “Perfil filho não pode ser incluído sem um perfil pai”.

***[FS011] Sistema não selecionado***

Caso o sistema não tenha sido selecionado, então será apresentada a seguinte mensagem de erro “Perfil pai não pode ser incluído sem um sistema”.

***[FS012] Perfil não selecionado***

Caso o perfil não tenha sido selecionado, então será apresentada a seguinte mensagem de erro “O Perfil não foi selecionado”.

**Requisitos Não-Funcionais**

**Segurança:** Obter as restrições de acesso do usuário através da execução do caso de uso UC3.006 – Obter Restrições de Acesso

**Performance:** Como a estrutura apresentada é uma árvore, a performance deve ser considerada para que não exauste o usuário.

### Observações

Este caso de uso será executado eventualmente, apenas quando houver necessidade de cadastrar ou alterar perfis de usuários.

#### [UC1.008] Configurar restrições por perfil

Objetivo: <b>No cadastro de perfis, configurar as restrições de acesso associadas ao perfil selecionado</b>		
Ator Responsável: <b>Gestor de Acesso</b>		
Requisitos Atendidos: <b>RF1.005</b>		
Contexto		
Eventos que Disparam o Caso de Uso	Entradas	Saídas
Selecionar função Associar restrições por perfil	Perfil selecionado.	

**Pré condições:** o usuário logado deve estar associado a um perfil de gestor de acesso ou de administrador de segurança.

**Pós condições:** se o caso de uso for realizado com sucesso as restrições de acesso a dados do perfil serão registradas no sistema.

#### [UC1.009] Detalhar perfil

Objetivo: <b>No cadastro de perfis, permitir que sejam apresentadas informações detalhadas do perfil selecionado</b>		
Ator Responsável: <b>Gestor de Acesso</b>		
Requisitos Atendidos: <b>RF1.005</b>		
Contexto		
Eventos que Disparam o Caso de Uso	Entradas	Saídas
Selecionar função Detalhar perfil	Perfil selecionado	

**Pré condições:** o usuário logado deve estar associado a um perfil que tenha permissão de acesso a esta funcionalidade.

**Pós condições:** Não há.

#### [UC1.010] Consultar perfis

Objetivo: <b>Permitir a realização de consultas aos perfis de acesso. A seleção dos perfis a serem consultados deve ser realizada através de combinação de critérios de em filtros de pesquisas.</b>		
Ator Responsável: <b>Usuário do Sistema</b>		
Contexto		
Eventos que Disparam o Caso de Uso	Entradas	Saídas
Selecionar função Consultar perfil		

**Pré condições:** o usuário logado deve estar associado a um perfil que tenha permissão de acesso a esta funcionalidade.

**Pós condições:** Não há.

### Fluxo de eventos principal

- O sistema mostra tela com filtros que podem ser usados na pesquisa de perfis:
  - Sistema
  - Descrição do Perfil
  - Tipo do Perfil (Administrador de Segurança, Gestor de Acesso, Usuário)
  - Função com acesso liberado ([FS003] Pesquisar função).
- O usuário informa os filtros pelos quais deseja pesquisar e submete a pesquisa.



3. O sistema realiza a pesquisa e apresenta uma lista de perfis que atendem às condições previstas, mostrando os campos citados abaixo: ([FS001] Nenhum perfil encontrado)
- Sistema
  - Perfil Pai (se existir)
  - Descrição do Perfil ([FS002] Detalhar perfil)
  - Tipo do Perfil (Administrador de Segurança, Gestor de Acesso, Usuário)

#### **Fluxos secundários (alternativos e de exceção)**

##### **[FS001] Nenhum perfil encontrado**

Caso nenhum perfil seja encontrado para os filtros escolhidos, o sistema apresenta a mensagem “Não foi selecionado nenhum perfil”.

##### **[FS002] Detalhar perfil**

O sistema executa o caso de uso [UC1.009] Detalhar perfil.

##### **[FS003] Pesquisar função**

O sistema executa o caso de uso [UC6.003] Pesquisar função.

#### **Requisitos Não-Funcionais Associados**

**Segurança:** Obter as restrições de acesso do usuário através da execução do caso de uso UC3.006 – Obter Restrições de Acesso

**Performance:** A performance deve ser considerada para que não seja exaustivo para o usuário.

**Usabilidade:** Este caso de uso será executado por diversos tipos de usuários, logo sua interface deve ser de simples, clara e de fácil manipulação.

#### **Observações**

Este caso de uso será executado por diversos tipos de usuários, eventualmente.

## **Gestão de Usuários**

Nesta seção devem ser descritos os casos de uso do pacote Gestão de Usuários....

## **Controle de Acesso**

Nesta seção devem ser descritos os casos de uso do pacote Controle de Acesso....

## **Trilha de Auditoria**

Nesta seção devem ser descritos os casos de uso do pacote Trilha de Auditoria....

## **3. Referências**

Esta seção lista os documentos relacionados a este documento, com links para os mesmos :

1. [Ata de reunião de levantamento de requisitos](#); realizada no dia 10/07;
2. [Ata de reunião de levantamento de requisitos](#); realizada no dia 12/07;
3. [Ata de reunião de levantamento de requisitos](#); realizada no dia 13/07;
4. [Ata de reunião de levantamento de requisitos](#); realizada no dia 17/07;
5. [Documento de Requisitos](#);
6. [Documento de Requisitos de Interface](#);
7. [Glossário Léxico](#);
8. [Diagrama dos Casos de Uso](#).

## **Apêndice C REQUISITOS DE INTERFACE**

---

### **Apresentação**

Este apêndice apresenta uma versão do Documento de Requisitos de Interface do Sistema de Gestão de Acessos, seguindo o template proposto no WRE-Process. Este documento permite que o leitor tenha um entendimento dos requisitos de interface e avalie a aplicação do template proposto em um projeto real.

# **Sistema Gestão de Acessos**

## **Requisitos de Interface**

Responsáveis:  
Ana Cláudia Beltrão (Analista de Negócios)  
15/03/03

## Informação do Documento de Requisitos de Interface

<b>Título do documento</b>	Documento de Requisitos de Interface.		
<b>Autor</b>	Designer Gráfico CESAR		
<b>Comentários</b>	Este é o Documento de Requisitos de Interface do projeto Gestão de Acesso.		
<b>Nome do arquivo</b>	Requisitos_Interface		
<b>HISTÓRICO DE REVISÕES</b>			
<b>Revisão</b>	<b>Data</b>	<b>Descrição</b>	<b>Autor</b>
00	20/07	Primeira versão	Ana Claudia Didier

## 1. Introdução

Este documento especifica e consolida os requisitos de interface do sistema *Gestão de Acessos*, fornecendo aos clientes, desenvolvedores e designers as informações necessárias para o projeto da interface, além das necessidades tecnológicas de interface e sua implementação, visando facilitar a criação de layouts e o acompanhamento do projeto.

Durante o desenvolvimento do projeto só serão tratados os requisitos de interface para construção do layout que serão usados para os módulos deste sistema, e que servirão como padrão para as novas aplicações desenvolvidas sobre plataforma web.

### Visão geral deste documento

Este documento foi organizado em 6 seções, conforme descritas abaixo:

**Seção 2 – Princípios Técnicos:** apresenta os padrões adotados para o desenvolvimento do projeto de interface do Sistema de Gestão de Acesso.

**Seção 3 – Visão Geral da Interface do Sistema:** os objetivos gerais e objetivos específicos da interface do sistema.

**Seção 4 – Requisitos de Interface:** especifica todos os requisitos para a interface do sistema.

**Seção 5 – Referências:** contém uma lista de referências para outros documentos e sites relacionados.

### Convenções, termos e abreviações

A correta interpretação deste documento exige o conhecimento de algumas convenções e termos específicos que são descritos a seguir.

### Identificação dos Requisitos

Por convenção, a referência a requisitos é feita através do nome da subseção onde eles estão descritos, seguidos do identificador do requisito, de acordo com o esquema abaixo:

*[nome da subseção.identificador do requisito]*

Por exemplo, o requisito não funcional [Usabilidade.RI001] está descrito na seção de requisitos de interface Usabilidade, em um bloco identificado por [RI001].

### Prioridades dos Requisitos

Para estabelecer a prioridade dos requisitos foram adotadas as denominações “essencial”, “importante” e “desejável”, conforme descritas a seguir:

**Essencial** é o requisito sem o qual o sistema não entra em funcionamento. Requisitos essenciais são requisitos imprescindíveis, que têm que ser implementados impreterivelmente.

**Importante** é o requisito sem o qual o sistema entra em funcionamento, mas de forma não satisfatória. Requisitos importantes devem ser implementados, mas, se não forem, o sistema poderá ser implantado e usado mesmo assim por um tempo determinado e acordado com o cliente.

**Desejável** é o requisito que não compromete as funcionalidades básicas do sistema, isto é, o sistema pode funcionar de forma satisfatória sem ele. Requisitos desejáveis são requisitos que podem ser deixados para

versões posteriores do sistema, caso não haja tempo hábil para implementá-los na versão que está sendo especificada. Entretanto deve-se observar estes requisitos para que a arquitetura desenvolvida seja extensível para da implementação deste requisito.

## 2. Princípios Técnicos

Esta seção descreve os princípios técnicos que deverão ser adotados como padrões para o desenvolvimento do projeto de interface.

### **Navegador**

O navegador preferencial será o Internet Explorer em plataforma PC/OS Windows, melhor visualizado em 800x600 pixels, mínimo de 256 cores.

### **Add-on**

Considerar o provável uso de Adobe Acrobat para leitura de documentos.

### **Estrutura**

#### **A estrutura da interface deverá considerar:**

- Padronização de botões;
- Padronização de lay-out de telas e de acesso às funcionalidades;
- Padronização da Arquitetura Navegacional;
- Uso de templates de páginas;
- Uso de CSS (Cascade Style Sheet);
- Autonomia de importância mediana entre a Logomarca/Comunicação Visual da empresa e o sistema, isto é, a Logomarca/Comunicação Visual não deverá estar em grande evidência no layout da interface gráfica do sistema;
- Não utilização de logotipo próprio para o sistema.

## 3. Visão Geral da Interface do Sistema

Esta seção descreve os objetivos gerais e objetivos específicos da interface do sistema de *Gestão de Acessos* e de todas as aplicações que venham a ser gerenciadas por ele, bem como a identificação do público alvo do sistema.

### **Objetivo Geral da Interface**

Suportar todas as operações da operadora, considerando o máximo de flexibilidade na identidade visual do sistema.

### **Público Alvo**

Em um primeiro momento, funcionários de todos os níveis gerenciais e operacionais, indo desde a vice-presidência até funcionários do Callcenter. Posteriormente, a aplicação será aberto a clientes externos, lojistas, etc.

### **Requisitos de Interface**

### **Usabilidade**

Esta seção descreve os requisitos associados à facilidade de uso da interface com o usuário.

### [RI001] Interface Simplificada

*O sistema deve prover uma interface simples, para facilitar ao máximo o uso do mesmo por parte dos usuários uma vez que a grande maioria das telas do sistema serão telas de trabalho de oito horas/dia. A navegação deverá ser intuitiva, fornecendo ao usuário opções possíveis, evitando desta forma a entrada de dados que possam apresentar conflitos.*

**Prioridade:**  Essencial  Importante  Desejável

**Forma de comprovação:** através da aprovação do protótipo navegacional, protótipo de interface e utilização do sistema por parte dos usuários.

### [RI002] Interface Padronizada

*O sistema deve prover uma interface padronizada, para facilitar o uso do sistema pelo usuário independentemente do módulo ao qual ele está acessando. Alguém que seja treinado num módulo, poderá usar facilmente qualquer outra aplicação uma vez que elas possuem o mesmo comportamento de interface. O padrão de cores não deve ser determinado pelos módulos do sistema.*

**Prioridade:**  Essencial  Importante  Desejável

**Forma de comprovação:** através de aprovação da proposta do novo layout.

### [RI003] Help On-Line

*O sistema deve prover ajuda on-line aos usuários para a navegação e execução das funcionalidades providas pelo sistema.*

**Prioridade:**  Essencial  Importante  Desejável

**Forma de comprovação:** utilização do sistema pelos usuários.

### [RI004] Interface Limpa

*O sistema deve prover uma interface clean, sem muitos elementos gráficos desnecessários, prezando pela limpeza visual da tela e possibilitando a rapidez de acesso. Utilizar componentes gráficos leves que não retardem o carregamento da página pelo browser.*

**Prioridade:**  Essencial  Importante  Desejável

**Forma de comprovação:** através da apresentação e aprovação da proposta do novo layout/skin, além da utilização do sistema pelos usuários.

## Identidade Visual

Esta seção descreve os requisitos associados identidade visual que deverá ser desenvolvida para o layout/skin do FiS.

### [RI005] Padrão de cores da Operadora

*O layout do sistema deverá obedecer ao padrão de cores da Operadora.*

**Prioridade:**  Essencial  Importante  Desejável

**Forma de comprovação:** através da apresentação e aprovação da proposta do novo layout/skin.

### [RI006] Limitação de cores para daltônicos

*A identidade visual do sistema deverá considerar as limitações de cores para daltônicos.*

**Prioridade:**  Essencial  Importante  Desejável

**Forma de comprovação:** através da apresentação e aprovação da proposta do novo layout e utilização do sistema por parte dos usuários daltônicos.

## 4. Referências

Esta seção enumera os sites considerados de boa qualidade visual:

[www.bowstreet.com](http://www.bowstreet.com);

[www.bb.com.br](http://www.bb.com.br);

[www.bancoreal.com.br](http://www.bancoreal.com.br).

## **Apêndice D CASOS DE USO DE INTERFACE**

---

### **Apresentação**

Este apêndice apresenta uma versão sintetizada do Documento de Casos de Uso de Interface do Sistema de Gestão de Acessos. Como este sistema é bastante complexo, sua documentação é volumosa; e como o objetivo desta dissertação é validar o processo de Engenharia de Requisitos WRE-Process, procuramos simplificar o documento de modo que permitisse ao leitor ter um entendimento geral do problema e averiguar a aplicação do template proposto em um projeto real, de modo não exaustivo. Desta forma, são descritos apenas os casos de interface relativos à navegação pela página principal e na apresentação do menu dinâmico com as funcionalidades acessadas por um perfil fictício.



# **Sistema Gestão de Acessos**

## **Especificação de Casos de Uso de Interface**

Versão 01.02  
Responsáveis:  
Ana Cláudia Beltrão (Analista de Negócios)  
21/04/03

### Informação do Documento de Casos de Uso da Interface

<b>Título do documento</b>	Especificação de Casos de Uso de Interface.		
<b>Autor</b>	Ana Cláudia Beltrão Analista de Negócios CESAR		
<b>Comentários</b>	Este é o Documento de Casos de Uso da Interface do Sistema Gestão de Acesso.		
<b>Nome do arquivo</b>	Casos_Uso_Interface		
<b>HISTÓRICO DE REVISÕES</b>			
<b>Revisão</b>	<b>Data</b>	<b>Descrição</b>	<b>Autor</b>
1.0	30/03	Primeira versão	Ana Cláudia
1.1	17/03	Ajustes realizados após reunião de validação de requisitos de interface	Ana Cláudia

## 1. Introdução

Este documento especifica os cenários de uso da interface do Sistema Gestão de Acessos, visando possibilitar o conhecimento dos conteúdos visuais e textuais de cada tela a ser elaborada no projeto de interface.

É importante ressaltar que o aspecto visual (diagramação, cor, etc.) mostrado nas imagens a seguir não deve ser considerado como solução visual, bem como o da arquitetura navegacional e usabilidade do sistema simulados neste documento. Este documento tem caráter apenas de especificação de conteúdo previsto para cada tela de acordo com o perfil do usuário.

## 2. Casos de Uso de Interface

### [UCI.001] Navegar Página Principal

5. Ao entrar na página inicial da aplicação, o usuário visualiza links para:
    - Navegar pelas promoções
    - Navegar pelas notícias
    - Navegar pela ajuda ([UCI.002] Navegar Ajuda)
    - Navegar pelo mapa do site
  6. O sistema também apresenta um formulário para solicitar:
    - Login
    - Senha
  7. O usuário seleciona um dos links ou informa login e senha para tentar acessar o sistema.
- \* Ver Cenário 1

### [UCI.002] Navegar Ajuda

1. O sistema apresenta uma tela popup com a Ajuda inicial da aplicação, apresentando a que se destina a aplicação, como o usuário deve proceder para efetuar login ou navegar pelos links apresentados.

### [UCI.003] Navegar Mapa do Site

1. O sistema apresenta uma tela popup com o mapa do site da aplicação. Este mapa do site é apresentado em uma estrutura hierárquica (árvore), onde só são apresentadas as funcionalidades acessíveis ao usuário. Logo, a partir da página inicial, o mapa do site apenas apresenta os links que o usuário web pode acessar.

#### **[UCI.004] Navegar Notícias**

1. O sistema apresenta uma tela para acesso às notícias publicadas pela operadora. Estas notícias têm conteúdo dinâmico e são carregadas pela equipe de assessoria de imprensa da operadora.

#### **[UCI.005] Navegar Promoções**

1. O sistema apresenta uma tela para acesso às promoções publicadas pela operadora. Estas promoções têm conteúdo dinâmico e são carregadas pela equipe de assessoria de marketing e comercial da operadora.

#### **[UCI.006] Efetuar Login**

1. O sistema executa o caso de uso funcional UC3.001 – Efetuar Login, onde caso o login seja bem sucedido, o sistema monta o menu do usuário [UC3.004- Montar Menu do Usuário], apenas com os sistemas, subsistemas e funções acessíveis pelos perfis do usuário.
2. O usuário então, pode selecionar uma das opções do menu, através da execução do caso de uso funcional [UC3.005] Selecionar opção de menu. De acordo com a opção selecionada, o sistema executa o caso de uso correspondente do sistema ao qual a função pertence.
3. O usuário pode encerrar a sessão [UC3.002-Encerrar Sessão] clicando no link Encerrar Sessão.

### **3. Cenários**

Nesta seção são apresentados alguns cenários de uso da interface para que o leitor possa visualizar o comportamento do sistema para diferentes tipos de perfis de acesso. O cenário 2 apresenta um possível perfil de acesso de um usuário para um sistema de call center da operadora. Este perfil é apenas ilustrativo, para facilitar o entendimento da aplicação que é um sistema para gestão de acessos de aplicações web.

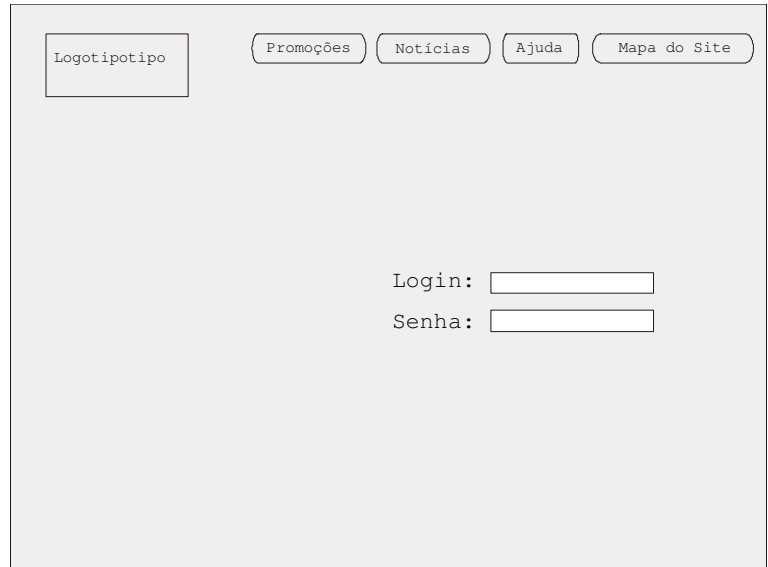
#### **Cenário 1**

O cenário 1 apresenta um possível perfil de acesso\* de um usuário Gestor de Acesso, que executa os seguintes casos de uso de interface:

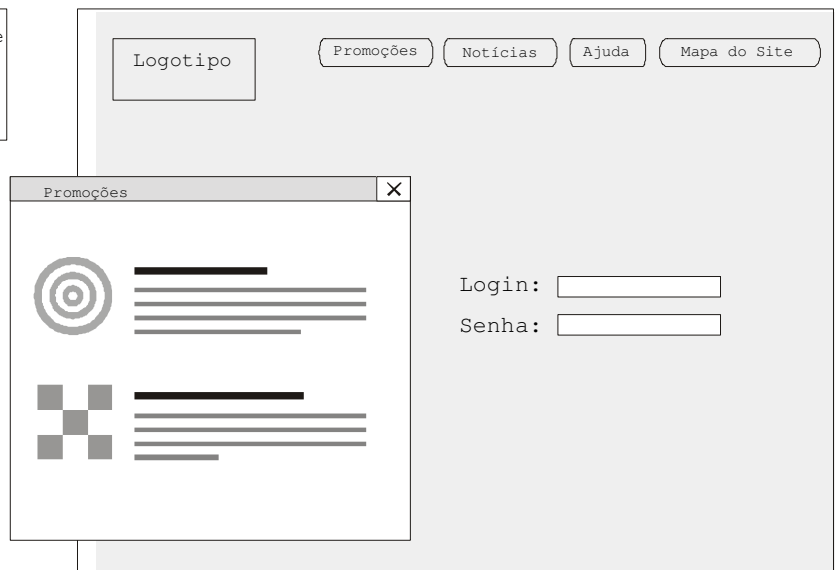
1. [UCI.001] Navegar Página Principal
2. [UCI.005] Navegar Promoções
3. [UCI.006] Efetuar Login
4. O sistema apresenta o menu com as opções de acesso permitidas ao usuário, neste caso, para o sistema de Gestão de Acesso.

\* Este perfil é apenas ilustrativo, para facilitar o entendimento da aplicação que é um sistema para gestão de acessos de aplicações web.

- Abre a tela do sistema
- Clica na opção de promoções



- Abre-se uma janela flutuante com as promoções.
- Depois de ler as promoções fecha a janela flutuante.



- Abre a tela do sistema Geral
- usuário 1 acessa o Sistema Geral através do Login e Senha.

Logotipo

Promoções Notícias Ajuda Mapa do Site

Usuário 1  
[ terça, 31.07.2003 ] [ 17:51 ]

Login:

Senha:

- Por ter apenas acesso a parte de "controle de acesso" nenhum outro sistema estará visível em sua tela.
- O Usuário 1 seleciona a opção 'Alterações nos perfis'

Logotipo

Promoções Notícias Ajuda Mapa do Site

Usuário 1  
[ terça, 31.07.2003 ] [ 17:51 ]

[home](#) >

- Usuários por nome
- Usuários por perfil
- Perfis de Usuários
- Autorização de um perfil
- Autorização de acesso
- Funcionalidades por sistema
- Alterações nos perfis
- Alter. na assoc.de perfis

Voltar Salvar

logout

## Cenário 2

O cenário 2 apresenta um possível perfil de acesso\* de um usuário do Call Center, que executa os seguintes casos de uso de interface:

1. [UCI.001] Navegar Página Principal
2. [UCI.006] Efetuar Login
3. O sistema apresenta o menu com as opções de acesso permitidas ao usuário, neste caso, para o sistema Call Center.

\* Este perfil é apenas ilustrativo, para facilitar o entendimento da aplicação.

- Abre a tela do sistema Geral  
- usuário 2 acessa o Sistema Geral através do Login e Senha.

The screenshot shows the login page of the system. At the top left is a box labeled 'Logotipo'. To its right are four buttons: 'Promoções', 'Notícias', 'Ajuda', and 'Mapa do Site'. Below these is a horizontal line, followed by the text 'Usuário 2' and the date and time '[ terça, 31.07.2003 ] [ 17:51 ]'. In the center, there are two input fields: 'Login:' with the value 'Usuário 2' and 'Senha:' with the value '\*\*\*\*\*'.

- Por ter apenas acesso ao Sistema Call Center nenhum outro sistema estará visível em sua tela.

-O Usuário 2 escolhe a opção 'Serviços de Atendimento'.

The screenshot shows the menu page of the system. At the top left is a box labeled 'Logotipo'. To its right are four buttons: 'Promoções', 'Notícias', 'Ajuda', and 'Mapa do Site'. Below these is a horizontal line, followed by the text 'Usuário 2' and the date and time '[ terça, 31.07.2001 ] [ 17:51 ]'. Below this is the text 'homeFBS > \_Call Center'. A list of menu items is displayed in a table-like structure:

Autorizar Compra
Suspender Cartão
Retirar Suspensão
Crédito Instantâneo
Serviços de Atendimento
Emissão de Cartas
Cancelar Autorização
Consultas e informações

A mouse cursor is pointing to the 'Serviços de Atendimento' option. At the bottom right, there are two buttons: 'Voltar' and 'Salvar'. At the bottom left, there is a 'logout' button and a progress indicator consisting of several small squares.