

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**UM CONJUNTO DE ABORDAGENS PARA A
GERAÇÃO DA MATRIZ DE RASTREABILIDADE DE
REQUISITOS COM SUORTE DE TÉCNICAS DE
INTELIGÊNCIA COMPUTACIONAL**

ANDRÉ DITOMASO

São Carlos - SP
Novembro/2014

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**UM CONJUNTO DE ABORDAGENS PARA A
GERAÇÃO DA MATRIZ DE RASTREABILIDADE DE
REQUISITOS COM SUPORTE DE TÉCNICAS DE
INTELIGÊNCIA COMPUTACIONAL**

ANDRÉ DITOMASO

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Doutor em Ciência da Computação, área de concentração: Engenharia de Software.

Orientadora: Dra. Sandra Carmargo Pinto Ferraz Fabbri.

São Carlos - SP
Novembro/2014

Ditomaso, Andre

Um Conjunto de Abordagens para a Geração da Matriz de Rastreabilidade de Requisitos com Suporte de Técnicas de Inteligência Computacional / Andre Ditomaso. -- 2015.

150 f. : 30 cm.

Tese (doutorado)-Universidade Federal de São Carlos, campus São Carlos, São Carlos

Orientador: Sandra Fabbri

Banca examinadora: Daniel Lucrédio; Heloisa de Arruda Camargo; Eliane Martins; Fernanda Maria Ribeiro de Alencar

Bibliografia

1. rastreabilidade de requisitos. 2. matriz de rastreabilidade de requisitos. 3. engenharia de software experimental. I. Orientador. II. Universidade Federal de São Carlos. III. Título.

Ficha catalográfica elaborada pelo Programa de Geração Automática da Secretaria Geral de Informática (SIn).

DADOS FORNECIDOS PELO(A) AUTOR(A)

Bibliotecário(a) Responsável: Ronildo Santos Prado – CRB/8 7325



UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a Defesa de Tese de Doutorado do candidato Andre Di Thommazo, realizada em 05/12/2014:

Profa. Dra. Sandra Camargo Pinto Ferraz Fabbri
UFSCar

Prof. Dr. Daniel Lucrédio
UFSCar

Profa. Dra. Heloisa de Artuda Camargo
UFSCar

Profa. Dra. Eliane Martins
UNICAMP

Profa. Dra. Fernanda Mária Ribeiro de Alencar
UFPE

AGRADECIMENTOS

Primeiramente, agradeço a Deus por permitir a realização deste trabalho.

A minha orientadora, Prof^ª. Dr^ª. Sandra Fabbri, pela confiança, paciência e dedicação. Muito obrigado pela constante ajuda na orientação e pela amizade.

Aos meus pais Zezo e Lena, que sempre apoiaram-me e incentivaram-me nessa caminhada. A minha irmã Luciana, pelo incentivo para a vida acadêmica e ajuda estatística. Ao meu irmão Renato, pela motivação.

À minha filha Letícia - minha princesa Bela - e ao meu filho Gabriel - o meu Homem-Aranha. Vocês conseguem transformar minha vida a cada momento. À minha esposa Aisla, pelo amor, carinho, paciência e muitos outros predicados com os quais me presenteia a cada dia. Não tenho dúvidas que Deus me deu muito mais do que mereço e minha alegria é estar com vocês! A todos de minha família, pela compreensão nos inúmeros momentos de ausência que este trabalho me exigiu.

Aos colegas do LaPES: Anderson Belgamo, Elis Hernandez, Kamilla Camargo, Juciara Nepomuceno, Fábio Octaviano, Cleiton Silva, Augusto Zamboni, Arlindo Neto, Erik Antonio, Rafael Gastaldi, Odair Souza, André Abade, Ana Eliza Pedroso e Thiago Lacerda. Um agradecimento especial aos amigos com os quais mais interagi na construção deste trabalho: Elis, Kamilla, Fábio e Anderson. Tenham certeza que a ajuda de vocês foi fundamental para a finalização desta tese. Aos alunos de iniciação científica que estiveram diretamente envolvidos com este projeto. Tenham a certeza que o resultado deste trabalho se deve ao esforço de cada um de vocês. Muito obrigado Thiago Ribeiro, Guilherme Olivatto, Gabriel Malimpensa, Rafael Rovina, Gislaine Gonçalves, Cleiton Silva, Denis Muniz, Emanuel Teodoro e Maiko Trindade. Desejo a vocês todo sucesso do mundo e que encontrem em seus caminhos pessoas e profissionais tão bons como vocês foram comigo.

Aos amigos Marcos Danilo e Pedro Nassif, pela longa amizade. Ao amigo e pesquisador Lúcio Jorge, pelo grande exemplo como pesquisador e por ter me aberto os olhos para as diversas aplicações práticas da inteligência computacional no dia-dia. Às minhas primas Thaís e Natália, pela ajuda com o texto, as formatações e as orações.

Aos professores e funcionários do Departamento de Computação da UFSCar e aos colegas do Instituto Federal de São Paulo – campus São Carlos.

Enfim, a todos que fizeram e fazem parte da minha vida e que, de uma forma ou de outra ajudaram na construção deste trabalho.

Resumo

Contexto: Problemas no gerenciamento de requisitos são apontados como uma das principais causas do insucesso dos sistemas. Uma das atividades que ajudam a garantir gerenciamento de requisitos de forma eficaz é a rastreabilidade de requisitos. Um dos artefatos mais importantes para a determinar e acompanhar a rastreabilidade dos requisitos é a Matriz de Rastreabilidade de Requisitos. Apesar disso, estabelecer e manter a Matriz de rastreabilidade de requisitos é tarefa extremamente trabalhosa e sujeita a erros. **Objetivo:** O objetivo desta tese é apresentar um conjunto de abordagens que possibilitam gerar a Matriz de Rastreabilidade de Requisitos de forma automatizada, sendo que dentre elas utilizam-se técnicas de inteligência computacional que propiciam a geração de links mais acurados. **Metodologia:** As abordagens propostas exploram os dados dos requisitos funcionais e soluções de processamento de linguagem natural para determinação da matriz de rastreabilidade. A partir de experimentos essas abordagens foram refinadas e combinadas com técnicas de inteligência computacional para aumentar a acurácia nos links de rastreabilidade. **Resultados:** Quatro abordagens foram propostas. Uma delas explora o fato de que a dependência dos requisitos está associada aos dados manipulados por eles. Outra explora o uso de processamento de linguagem natural. A terceira combina as duas abordagens anteriores com sistemas fuzzy. Por fim a quarta abordagem utiliza redes neurais artificiais tendo como entrada os dados das duas primeiras abordagens propostas. Os estudos experimentais conduzidos produziram resultados satisfatórios relativos aos links de rastreabilidade determinados pelas abordagens. Também foram realizados estudos de caso para avaliar o uso das abordagens na indústria. **Conclusões:** os resultados alcançados corroboram a tese defendida nesta pesquisa, comprovando que o uso de técnicas de inteligência computacional melhora a acurácia nos links de rastreabilidade.

Palavras-chave: rastreabilidade de requisitos, matriz de rastreabilidade de requisitos, engenharia de software experimental.

Abstract

Context: Problems in requirements management is considered one of the causes for system failures. One of the activities that helps the requirements management, making it more effective, is the requirement traceability. One of the most important artefacts to determine and monitor the traceability of requirements is the Requirements Traceability Matrix. Nonetheless, establish and maintain the Requirements Traceability Matrix is an extremely laborious and error prone task.

Objective: The objective of this thesis is to present a set of approaches that allow to generate the Requirements Traceability Matrix in an automated way, using computational intelligence techniques that allows the generation of more accurate links.

Methodology: The proposed approaches explore functional requirements data and natural language processing solutions for determining the traceability matrix. From some experimental studies, the approaches were refined and combined with computational intelligence techniques to increase the accuracy of the traceability links.

Results: Four approaches were proposed. The first approach considers that the dependence of the requirements is related to the data manipulated by them. The second approach explores the use of natural language processing. The third approach combines the previous two approaches with fuzzy systems. Finally, the fourth approach uses artificial neural networks taking the data from the first two proposed approaches as input data. The experimental studies produced satisfactory results for the traceability links determined by the proposed approaches. Case studies were also conducted to evaluate the use of the approaches in the industry.

Conclusions: The obtained results support the thesis defended in this research, demonstrating that the use of computational intelligence techniques improves accuracy in traceability links.

links.

Keywords: requirements traceability, requirements traceability matrix, experimental software engineering.

Lista de Figuras

Figura 2-1. Exemplos de curva de fuzzyficação (adaptado de Artero (2009)).....	34
Figura 2-2. Curva de fuzzyficação para a variável linguística “temperatura” (adaptado de Artero (2009)).	36
Figura 2-3. Exemplos de função de ativação (adaptado de Coppin (2010)).	38
Figura 2-4. Esquema de um neurônio artificial	39
Figura 2-5. Exemplo de uma rede neural	39
Figura 3-1. GQM de planejamento da revisão bibliográfica.	48
Figura 3-2. Técnicas de Rastreabilidade entre Requisitos.	50
Figura 3-3. Quantidade de artigos por período.....	52
Figura 4-1. Definição dos intervalos de dependência com uso de árvore de decisão	64

SUMÁRIO

CAPÍTULO 1 - INTRODUÇÃO.....	12
1.1 Contexto	12
1.2 Motivação e objetivos	15
1.3 Metodologia de pesquisa.....	17
1.4 Organização	21
CAPÍTULO 2 - FUNDAMENTAÇÃO TEÓRICA.....	22
2.1 Considerações iniciais	22
2.2 Engenharia de Requisitos.....	23
2.3 Rastreabilidade de requisitos	25
2.3.1 Matriz de Rastreabilidade de Requisitos (RTM)	28
2.3.2 Detecção automática de rastreabilidade	29
2.4 . Técnicas de Inteligência Computacional.....	32
2.4.1 Sistema de Inferência Fuzzy	33
2.4.2 Redes Neurais.....	37
2.4.3 Algoritmos Genéticos	41
2.4.4 Árvores de Decisão	44
2.5 Considerações finais.....	45
CAPÍTULO 3 - REVISÃO BIBLIOGRÁFICA	46
3.1 Considerações Iniciais	46
3.2 Planejamento	47
3.2.1 Questão 1: Quais são as técnicas para determinar a rastreabilidade entre os requisitos?	49
3.2.2 Questão 2: Como é feita a validação das propostas para determinar a rastreabilidade entre requisitos?	52
3.2.3 Questão 3: Quais são as técnicas para determinar a rastreabilidade entre os requisitos e outros artefatos?	54
3.3 Considerações finais.....	58
CAPÍTULO 4 - ABORDAGENS PARA GERAÇÃO DA RTM.....	60

4.1 Considerações Iniciais	60
4.2 Abordagem RTM-E: geração da Matriz de Rastreabilidade de Requisitos baseada nos dados de entrada	62
4.3 Abordagem RTM-NLP: geração da Matriz de Rastreabilidade de Requisitos baseada em Processamento de Linguagem Natural	64
4.4 Abordagem RTM-Fuzzy: geração da Matriz de Rastreabilidade de Requisitos baseada em Sistemas Fuzzy.....	68
4.5 Abordagem RTM-N: geração da Matriz de Rastreabilidade de Requisitos baseada em Redes Neurais Artificiais.....	74
4.6 Considerações Finais	77
CAPÍTULO 5 - ESTUDOS EXPERIMENTAIS	78
5.1 Considerações Iniciais	78
5.2 Estudo de Caso 1.....	80
5.3 Estudos Experimentais Controlados	87
5.3.1 Experimento 1	87
5.3.2 Experimento 2	91
5.3.3 Experimento 3	96
5.4 Estudo de Caso 2.....	103
5.5 Considerações Finais	106
CAPÍTULO 6 - BENEFÍCIOS DA RTM COM NÍVEIS DE DEPENDÊNCIA NO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE.....	108
6.1 Considerações iniciais	108
6.2 Priorização de Casos de Teste e Teste de Regressão.....	109
6.3 Abordagem Proposta.....	110
6.4 Estudo de Caso.....	115
6.5 Considerações Finais	124
CAPÍTULO 7 - CONCLUSÃO	126
7.1 Conclusões.....	126
7.2 Contribuições da tese	127
7.3 Limitações do trabalho.....	128
7.4 Publicações	129
7.5 Orientações de Iniciação Científica	133

7.6 Participação em outros projetos do grupo de pesquisa	135
7.7 Oportunidades Futuras	136
REFERÊNCIAS.....	137
APÊNDICE A.....	147
Introdução.....	147
Aspectos de implementação.....	147
Funcionalidades	148

Capítulo 1

INTRODUÇÃO

Este capítulo contextualiza o problema tratado nesta tese, a motivação para propor uma solução, a metodologia de trabalho utilizada para o desenvolvimento dessa solução e a organização deste texto.

1.1 Contexto

O uso de software no cotidiano das pessoas aumenta a cada dia mais, e pode-se dizer que já faz parte de, praticamente, qualquer atividade humana. A tendência de utilização da informática nos mais diversos setores produtivos é muito grande, principalmente, levando-se em consideração o surgimento de novas tecnologias, como a computação móvel e as redes sem fio. Dessa forma, o uso da informática está atingindo todos os públicos, gerando uma demanda por software excessivamente grande. Por outro lado, a indústria do software enfrenta desafios de gerar produtos que atinjam às expectativas dos clientes, obedecendo a prazos, custos e critérios de qualidade.

Pesquisas do instituto Standish Group (2014) mostram que a quantidade de projetos que foi finalizada com sucesso, ou seja, respeitando os prazos estabelecidos, o orçamento e, sobretudo, atendendo às expectativas dos clientes em todas as funcionalidades e características do software, é de apenas 39%. Apesar de ser um percentual baixo, observando-se o histórico publicado pelo mesmo instituto, vê-se que esse número apresenta uma melhora com relação aos anos anteriores. A Tabela 1, com dados dessa pesquisa, ilustra essa realidade.

Para a situação de o projeto ser classificado como “Falha”, é necessário que tenha sido abortado antes de concluído, ou que tenha sido entregue ao usuário, mas

nunca utilizado por ele. No caso do projeto ser definido como Problemático, é preciso que ele não tenha respeitado prazo, custo, e/ou que tenha sido entregue sem alguma característica solicitada pelo cliente.

Tabela 1 - Histórico sobre a situação de projetos de software [Standish Group, 2013]

Ano	Situação dos projetos de softwares		
	Sucesso	Falha	Problemático
2012	39 %	18 %	43 %
2010	37 %	21 %	42 %
2008	32 %	24 %	44 %
2006	35 %	19 %	46 %
2004	29 %	18 %	53 %
1998	26 %	28 %	46 %
1996	27 %	40 %	33 %
1994	16 %	31 %	53 %

A pesquisa realizada pelo mesmo instituto em 2014 (STANDISH GROUP, 2014) foi mais específica e buscava determinar quais fatores eram os mais importantes para se definir o sucesso ou não de um projeto de software. Dessa forma, ela procurou identificar o motivo pelo qual um projeto extrapola custo, prazo ou é entregue aos clientes com alguma de suas funcionalidades inconsistentes. Os dados da pesquisa estão resumidos na

Tabela 2.

Tabela 2 – Principais fatores de sucesso de um projeto de software [STANDISH GROUP, 2014]

Fatores que tornam um Projeto Crítico	% de Respostas
1. Falta de Especificação do Usuário	12.8%
2. Requisitos Incompletos	12.3%
3. Mudança nos Requisitos	11.8%
4. Falta de Apoio Executivo	7.5%
5. Tecnologia Imatura	7.0%
6. Falta de Recursos	6.4%
7. Expectativas previstas são irreais	5.9%
8. Objetivos obscuros	5.3%
9. Tempo irreal	4.3%
10. Tecnologia nova	3.7%
11. Outros	23.0%

Analisando-se a tabela, conclui-se que um fator crítico de sucesso para os projetos de software é a habilidade do processo em lidar com usuários e entender melhor as suas necessidades, além de estar apto e preparado para as mudanças nos

requisitos do usuário. Os três primeiros fatores estão associados diretamente com Requisitos e somam mais de um terço dos motivos que podem levar ao insucesso de um produto de software. Salem (2006) também afirma que a maioria dos erros de software é fruto de erros na fase de levantamento de requisitos e de deficiência durante o acompanhamento da evolução deles no decorrer do processo de desenvolvimento de software. Cleland-Huang e outros (2012) também destacam a importância da habilidade em lidar com os requisitos para o sucesso nos projetos de software.

Assim, considerando as dificuldades existentes durante a engenharia de requisitos, este grupo de pesquisa (LaPES – Laboratório de Pesquisa em Engenharia de Software) tem explorado alternativas para minimizar alguns dos problemas, tais como: i) a técnica TUCCA (*Technique for Use Case Construction and Construction-Based Requirements Analysis*), que tem por objetivo sistematizar a construção do Modelo de Casos de Uso sendo que, simultaneamente, é feita uma inspeção do Documento de Requisitos (BELGAMO e FABBRI., 2005); ii) a definição de um *template* para a descrição de requisitos, baseado em trabalhos propostos na literatura (KAWAI, 2005) e; iii) a ferramenta COCAR, que dá suporte a várias atividades do ciclo de desenvolvimento de software, baseadas nos requisitos e no modelo de casos de uso (DI THOMMAZO *et al.*, 2007), como por exemplo, o gerenciamento de requisitos, a geração dos pontos por casos de uso e a geração de casos de teste com base em casos de uso.

As atividades de gerenciamento de requisitos foram o foco do mestrado do autor desta tese. Nesta tese de doutorado foi explorado um importante aspecto do gerenciamento de requisitos, a rastreabilidade de requisitos. A partir do estabelecimento da rastreabilidade de requisitos no processo de desenvolvimento de software é possível ter uma base para evolução dos requisitos (GUO *et al.*, 2009). Um dos principais artefatos para o estabelecimento da rastreabilidade é a Matriz de Rastreabilidade de Requisitos (RTM), uma vez que registra o relacionamento existente entre estes dentro de um sistema. A relação de dependência entre cada par de requisitos pode ser entendida como um link de rastreabilidade, e a RTM reúne todos os links de rastreabilidade existentes entre os requisitos do sistema. Diversos autores destacam a importância da RTM para o processo de desenvolvimento de software e a dificuldade existente para a sua geração ([CLELAND-HUANG, *et al.* 2014], [TAVASSOLI, 2012], [SOMMERVILLE, 2011], [GOKNIL, 2011], [GUO *et al.*,

2009] [MUNSON, NGUYEN, 2005]). Com o estabelecimento da RTM, é possível prever o impacto de mudanças, bem como dar suporte à definição do conjunto de testes de regressão ([CUDDEBACK et al., 2010]).

Apesar dos benefícios citados anteriormente, a criação da RTM é uma tarefa trabalhosa e sujeita a erros (CLELAND-HUANG *et al.*, 2012). O número de requisitos funcionais cresce à medida que aumenta a complexidade do software e, conseqüentemente a quantidade de links que relacionam os requisitos na RTM. Além disso, não basta apenas que a RTM seja criada no início do desenvolvimento do software. Ao longo da construção do software os requisitos sofrerão alterações e se fará necessária a atualização dos links de rastreabilidade na RTM. Sendo assim, o contexto em que esta pesquisa se insere é a criação e manutenção da RTM, dando suporte a atividades de gerenciamento de requisitos e, conseqüentemente, de melhoria de qualidade do processo de desenvolvimento de software. A motivação e os objetivos deste trabalho são apresentados na próxima seção.

1.2 Motivação e objetivos

Dado o contexto apresentado, a importância da rastreabilidade de requisitos para a atividade de gerenciamento de requisitos e os projetos desenvolvidos pelo LaPES, tem-se a motivação para a tese aqui proposta, que pode ser redigida da seguinte forma:

O uso de técnicas de inteligência computacional melhora a acurácia de links de rastreabilidade de requisitos

Segundo Cuddeback et al. (2010), apesar da RTM oferecer suporte à previsão de impacto de mudanças e à definição de casos de testes de regressão, a determinação dos links de rastreabilidade para a criação da RTM é uma tarefa trabalhosa e sujeita a erros. Conseguir estabelecer a rastreabilidade de forma automática e com acurácia pode significar redução no tempo de criação da matriz e melhora na sua qualidade. Pensar na acurácia dos links de rastreabilidade significa minimizar a quantidade de falsos positivos e falsos negativos que ocorrem na detecção da rastreabilidade para criação da RTM.

Entende-se como um falso positivo a indicação de uma relação de dependência entre dois requisitos quando, na verdade, essa relação não existe. O impacto de um falso positivo pode ser que, a partir da modificação em algum requisito, sejam selecionados casos de testes que não deveriam estar no conjunto de testes de regressão ou a determinação de algum impacto que, na realidade, não existe. Sendo assim, um falso positivo pode gerar um esforço extra durante a gestão da mudança do requisito que, de fato, não deveria ocorrer.

Outro problema que afeta a acurácia nos links de rastreabilidade são os falsos negativos. Entende-se como um falso negativo a não indicação de uma dependência entre requisitos que, na realidade, existe. Se um falso positivo gera “apenas” um esforço extra no processo de desenvolvimento de software, o falso negativo pode acarretar problemas mais sérios, por exemplo, um caso de teste que deveria fazer parte dos testes de regressão pode não ser selecionado. Também pode ocorrer de o impacto de uma mudança não ser identificado.

Neste trabalho de doutorado, dado o cenário de i) importância da RTM para o processo de desenvolvimento; ii) dificuldade em se estabelecer a RTM de forma automática e confiável e iii) importância da acurácia dos links de rastreabilidade na RTM, buscou-se contribuir com a área de Engenharia de Software, propondo-se, abordagens para a criação da RTM de forma automática e com acurácia refinada por meio de técnicas de inteligência computacional, minimizando assim os falsos positivos e os falsos negativos na RTM.

Desta forma, estabeleceu-se como objetivos primários deste trabalho:

- Proposição de abordagens para geração da RTM;
- Refinamento das abordagens com técnicas de inteligência computacional;
- Medição da acurácia dos links de rastreabilidade gerados pelas abordagens.

Tem-se como objetivos secundários deste trabalho:

- Implementação das abordagens em uma ferramenta computacional com o intuito de se avaliar a acurácia das abordagens
- Investigar o uso das abordagens no processo de desenvolvimento de software, em particular na fase de teste.

A próxima seção mostra a metodologia seguida para atingir os objetivos propostos anteriormente, tendo como motivação a melhora na acurácia dos links de rastreabilidade com uso de técnicas de inteligência computacional.

1.3 Metodologia de pesquisa

Para se desenvolver essa pesquisa, fez-se um planejamento que considerasse:

1. O estudo das abordagens de rastreabilidade existentes na literatura para levantamento do estado da arte do tema e de possíveis alternativas que corroborassem a proposta da tese;
2. O estudo de técnicas de inteligência computacional que pudessem ajudar na melhoria da acurácia dos links de rastreabilidade;
3. O levantamento de experimentos realizados na área de rastreabilidade para se buscar a forma de avaliação das propostas criadas na literatura;
4. A proposição de novas abordagens para determinar a RTM de forma iterativa e interativa, seguindo-se uma sequência de proposição da abordagem e posterior avaliação por meio de um estudo experimental;
5. O refinamento das abordagens propostas com o uso de técnicas de inteligência computacional, buscando-se soluções que corroborassem a proposta da tese, com posterior avaliação em estudos experimentais;
6. A realização de estudo de caso real para análise da viabilidade de geração da RTM na indústria;
7. Um estudo de caso para mostrar os benefícios existentes em se definirem links de rastreabilidade com níveis de dependência diferenciados, mostrando e como isso pode contribuir para outras atividades de Engenharia de Software, como por exemplo testes de regressão, por exemplo.

Antes de se iniciar a proposição das abordagens, foi conduzido um Mapeamento Sistemático para se identificar as abordagens existentes na literatura para a criação e a manutenção da rastreabilidade. Para esse levantamento

bibliográfico, foi feito o planejamento da pesquisa utilizando o paradigma GQM (*Goal, Question, Metric*) (BASILI, CALDIEIRA, ROMBACH, 1994), como é detalhado no Capítulo 3.

Como a proposta da tese previu o uso de técnicas de inteligência computacional, esse tema também foi fonte de estudo para o autor deste trabalho. Diferentemente do levantamento bibliográfico sobre a rastreabilidade de requisitos, feito por meio do Mapeamento Sistemático, o estudo das técnicas de inteligência computacional – sobretudo sistemas fuzzy e redes neurais – passou pela compreensão de seus fundamentos e investigação de suas aplicações no tema desta tese. Da mesma forma que se aborda a lógica fuzzy e as redes neurais, também se apresenta um breve estudo sobre algoritmos genéticos e árvores de decisão, técnicas utilizadas para melhorar as abordagens propostas. Um resumo desse estudo está descrito na Seção 2.4. Também foi conduzido um mapeamento sistemático para identificar os experimentos realizados pela comunidade científica para avaliação de propostas de geração da RTM, com o intuito de ajudar a definir as estratégias para validação das abordagens que seriam propostas.

Feito isso, foram feitas as proposições de abordagens para construção da RTM. Propôs-se então uma abordagem baseada em processamento de linguagem natural e outra baseada nas informações dos dados de entrada dos requisitos funcionais. Essas duas abordagens foram inicialmente avaliadas em um estudo de caso real de uma empresa de aviação privada e posterior avaliação por meio de um estudo experimental controlado. Tais abordagens foram publicadas no XXVI Simpósio Brasileiro de Engenharia de Software (DI THOMMAZO *et al.*, 2012) e são descritas resumidamente na sequência:

1) Matriz de Rastreabilidade de Requisitos baseada nos dados de entrada dos requisitos funcionais

Nesta abordagem, a relação de dependência entre os Requisitos Funcionais é determinada pelo percentual de dados de entrada coincidentes entre cada par de requisitos funcionais. Não foi encontrada na literatura proposta similar. Esta abordagem foi chamada de RTM-E e já foi avaliada por meio de estudos experimentais. A abordagem é detalhada na Seção 4.2.

2) Matriz de Rastreabilidade de Requisitos baseada em Processamento de Linguagem Natural

Para determinar a relação de dependência entre os Requisitos Funcionais, esta abordagem utiliza o método do Vetor de Frequência e a Similaridade do Cosseno (SALTON, ALLAN, 1994). Esse método retorna o percentual de similaridade entre dois trechos de texto. Já existem na literatura iniciativas para a geração da rastreabilidade com uso em algoritmos que explorem o método do Vetor de Frequência e a Similaridade do Cosseno, e as diferenças dessas abordagens para a proposta deste trabalho são detalhadas na Seção 4.3. Esta abordagem foi chamada de RTM-NLP e também já foi avaliada por meio de estudos experimentais.

A partir dessas duas abordagens, foram feitas as proposições de outras duas. Estas, por sua vez, combinam as primeiras com o uso de técnicas de inteligência computacional. Elas estão descritas na sequência:

3) Matriz de Rastreabilidade de Requisitos baseada em Sistemas Fuzzy

Nesta abordagem foi utilizado um Sistema Fuzzy para determinar os links de rastreabilidade entre os requisitos funcionais. Como entradas do sistema fuzzy, foram utilizadas a RTM-E e a RTM-NLP. Procurou-se, dessa forma, combinar as melhores características de cada uma das propostas anteriores. Esta abordagem foi chamada de RTM-Fuzzy e também foi avaliada por meio de estudos experimentais cujos resultados foram apresentados no XXVII Simpósio Brasileiro de Engenharia de Software (DI THOMMAZO *et al.*, 2013a). As funções de pertinência do sistema fuzzy foram ajustadas com uso de algoritmos genéticos, conforme apresentado em Di Thommazo e outros (2014). A abordagem é detalhada na Seção 4.4.

4) Matriz de Rastreabilidade de Requisitos baseada em Redes Neurais Artificiais

Nesta abordagem foi utilizado uma Rede Neural Artificial para determinar os links de rastreabilidade entre os requisitos funcionais. Para o treinamento da rede neural artificial foram utilizados dados de experimentos que envolviam a RTM-E e a RTM-NLP. Depois que a rede neural foi treinada, mostrou-se capaz de identificar a dependência entre os requisitos funcionais. Esta abordagem foi chamada de RTM-N e também foi avaliada por meio de estudos experimentais cujos resultados foram apresentados no XXV International Conference on Software Engineering and Knowledge Engineering (DI THOMMAZO *et al.*, 2013b) e é detalhada na Seção 4.5.

Ressalta-se que por meio desses estudos experimentais mencionados pode-se então corroborar a tese proposta de que os links de rastreabilidade podem ser melhorados com o uso de técnicas de inteligência computacional. Além disso é importante destacar que o resultado dos estudos experimentais permitiu refinar ainda mais as abordagens, proporcionando uma melhora gradativa da acurácia dos links de rastreabilidade fazendo-se uso de outras técnicas de inteligência computacional (algoritmos genéticos e árvores de decisão). Essa melhora foi apresentada em Di Thommazo e outros (2014) e é detalhado no Capítulo 5.

Uma característica comum a todas as abordagens é que elas indicam não somente se existe ou não dependência entre os Requisitos Funcionais, mas também o nível da dependência (forte ou fraca). As vantagens dessa diferenciação em níveis são detalhadas no Capítulo 6 e adianta-se que, ao se considerarem os níveis de dependência, é possível priorizar casos de testes de regressão, bem como caracterizar o impacto de mudanças. Esses aspectos foram explorados em um estudo de caso que acompanhou o desenvolvimento de um software real.

Destaca-se que o foco deste trabalho está na rastreabilidade de requisitos funcionais do software, sendo que não foi explorada a rastreabilidade de requisitos não-funcionais.

Para que fosse possível a avaliação das abordagens propostas, simultaneamente à sua definição, elas foram implementadas na ferramenta COCAR que foi iniciada durante o mestrado do autor deste trabalho. Para a realização desta tese de doutorado, a ferramenta foi migrada de ambiente WEB para desktop com o intuito de facilitar a realização de experimentos controlados e para permitir o uso de algumas bibliotecas. Esse desenvolvimento contou com a contribuição de projetos de Iniciação Científica orientados pelo autor desta tese. Inicialmente foram implementadas funcionalidades de cadastro, atualização e exclusão de requisitos funcionais de um sistema, além de relatórios. Terminada essa etapa, as abordagens para criação da RTM foram incorporadas na ferramenta. A ferramenta é detalhada no Apêndice A.

Depois de realizados os estudos teóricos (mapeamentos sistemáticos e estudo das técnicas de inteligência computacional), as proposições e os refinamentos das abordagens para geração da RTM bem como suas avaliações por meio de estudos experimentais controlados no ambiente acadêmico, foram conduzidos estudos de

casos reais. O primeiro desses estudos de caso foi realizado em um software desenvolvido em uma empresa de aviação e que tratava do gerenciamento de estoque da empresa. No segundo desses estudos exploraram-se as abordagens propostas com base em requisitos de um software livre. Em um terceiro estudo de caso investigou-se os benefícios da definição de diferentes níveis de dependência associados aos links de rastreabilidade. Esse estudo de caso é detalhado no Capítulo 6 e adianta-se que, a partir da RTM criada, foi possível a definição de um conjunto de testes de regressão, bem como a caracterização do impacto das mudanças no software. Esses estudos de caso buscaram mostrar que as propostas desta tese de doutorado são viáveis de serem utilizadas na indústria.

1.4 Organização

Organizou-se a apresentação deste trabalho em sete capítulos e um apêndice. Este presente capítulo apresentou o contexto no qual a tese está inserida, bem como a motivação, os objetivos e a metodologia de pesquisa que foi seguida.

No Capítulo 2 é apresentada a Fundamentação Teórica dos temas que estão relacionados à pesquisa. Esses temas são: Engenharia de Requisitos, Rastreabilidade de Requisitos e Técnicas de Inteligência Computacional.

No Capítulo 3 é apresentada a Revisão Bibliográfica que foi planejada por meio do GQM e que explora temas essenciais para caracterizar a relevância e a originalidade desta tese. Com base no modelo GQM, foi elaborado o protocolo do mapeamento sistemático que foi conduzido para fazer tal caracterização.

No Capítulo 4 são apresentadas as abordagens propostas para a criação da RTM, a saber, RTM-E, RTM-NLP, RTM-Fuzzy e RTM-N,

No Capítulo 5 são descritos os estudos experimentais conduzidos para a avaliação das abordagens propostas.

No Capítulo 6 é apresentado um estudo de caso real para mostrar como a definição da RTM com níveis de dependência pode contribuir para o processo de desenvolvimento de software. Essa contribuição foi explorada no contexto da seleção

de casos de testes de regressão com priorização e a caracterização do impacto de uma mudança nos requisitos funcionais.

No Capítulo 7 apresenta-se a conclusão da tese, as contribuições da pesquisa e os trabalhos futuros.

Finalmente o Apêndice A descreve detalhes da ferramenta COCAR.

Capítulo 2

FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta uma visão geral dos principais temas relacionados com a tese que se propõe. Inicialmente são descritos aspectos da engenharia de requisitos e da rastreabilidade de requisitos. Também são detalhadas técnicas de inteligência computacional que foram utilizadas para a proposição das abordagens de rastreabilidade desta tese.

2.1 Considerações iniciais

Como apresentado no Capítulo 1, o objetivo desta tese é propor abordagens para a geração da RTM e combiná-las com técnicas de inteligência computacional. Sendo assim, apresenta-se, neste capítulo a fundamentação teórica para o trabalho desenvolvido. Considerando-se que a rastreabilidade de requisitos está no contexto da engenharia de requisitos, ambos são tratados neste capítulo. Como a tese fez uso da inteligência computacional, são apresentadas as técnicas utilizadas e também uma breve síntese sobre sistemas fuzzy, redes neurais, algoritmos genéticos e árvore de decisão.

O capítulo está organizado da seguinte forma: na Seção 2.2 são apresentados conceitos sobre engenharia de requisitos; na Seção 2.3 comenta-se sobre a rastreabilidade de requisitos e na Seção 2.4, técnicas de inteligência computacional, sobretudo lógica fuzzy, redes neurais, algoritmos genéticos e árvore de decisão. Por fim, a Seção 2.5 apresenta as considerações finais deste capítulo.

2.2 Engenharia de Requisitos

Requisito de software compreende tudo aquilo que o software deve fazer. De acordo com Abbott (1986), um requisito pode ser uma função, uma restrição ou outra característica que deve ser fornecida ou atendida para satisfazer as necessidades do usuário do sistema que será desenvolvido. Segundo o IEEE (1991), o requisito de software pode ser: i) uma condição ou capacidade que um usuário necessite para solucionar um problema ou atingir um objetivo; ii) uma condição que um sistema ou componente do sistema precise atender para satisfazer um contrato, padrão, especificação ou outro documento formalmente estabelecido; iii) uma representação documentada de uma condição ou capacidade, como em i) e ii).

Existem diversas maneiras de classificar os requisitos. Em seguida, mostra-se a classificação proposta por Sommerville (2011):

- **Requisitos Funcionais:** ligados diretamente à funcionalidade do software e definem como o sistema deve reagir às entradas específicas e como deve comportar-se em determinadas situações. Podem declarar explicitamente o que o software não deve fazer. Os trabalhos para a determinação da RTM nesta tese estão focados nos requisitos funcionais.
- **Requisitos Não Funcionais:** expressam as qualidades específicas que o software deve ter. São aqueles que não dizem respeito diretamente às funções específicas fornecidas pelo sistema (SOMMERVILLE, 2011). Enquanto um requisito funcional expressa o que uma parte do sistema deve fazer, o requisito não funcional pode estar associado a todo o sistema. Em alguns casos, se um desses requisitos não funcionais não for atendido, todo o software pode ficar

comprometido. Por exemplo, caso um software de aviação não atenda o critério de confiabilidade, ninguém irá utilizá-lo.

- **Requisitos de Domínio:** tem origem no domínio da aplicação e podem mostrar as características desse domínio. Podem ser tanto requisitos funcionais como não funcionais. Quando se trata de um software que calcula impostos, por exemplo, as regras e taxas que determinam como essa cobrança deve ser feita estão atreladas ao domínio da aplicação.

Segundo Sommerville (2011), o processo de descobrir, analisar, documentar e verificar essas funções e restrições é chamado de Engenharia de Requisitos. Sommerville (2011) divide a Engenharia de Requisitos em quatro fases: Levantamento dos Requisitos, Especificação dos Requisitos, Validação dos Requisitos e Gerenciamento de Requisitos. Essas quatro etapas serão explicadas a seguir:

- **Levantamento dos Requisitos:** trata-se da etapa em que os requisitos são extraídos dos clientes e usuários, determinando-se “o quê” o sistema deverá fazer. Trata-se de uma tarefa complexa, uma vez que envolve diversos aspectos como relações humanas, contexto social e econômico do ambiente em que o software será implantado. Sommerville (2011) lista algumas dessas dificuldades:
 - os *stakeholders* desconhecem os detalhes que desejam do sistema computacional e expressam naturalmente os requisitos em seus próprios termos, com conhecimentos implícitos de sua área de atuação;
 - os diferentes *stakeholders* têm em mente os mesmos requisitos mas podem expressá-los de maneiras distintas, causando conflitos;
 - os fatores políticos podem influenciar os requisitos do sistema;
 - o ambiente econômico e de negócios no qual o requisito está inserido é dinâmico. Inevitavelmente, ele se modifica durante o processo de análise. Novos requisitos podem surgir por parte de novos *stakeholders*, que não haviam sido consultados anteriormente.
- **Especificação de Requisitos:** é a etapa em que os requisitos levantados na fase anterior são documentados no Documento de Requisitos para uma posterior validação. Esse documento é a declaração oficial do que o sistema deve fazer. Power e Moyniahn (2003) definem o Documento de Requisitos

como o “contrato” entre cliente e aqueles que irão desenvolver o sistema, enfatizando também que, dada a sua importância, ele deva ser assinado pelas partes envolvidas.

- **Validação dos Requisitos:** é a etapa em que as possíveis inconsistências do Documento de Requisitos são eliminadas. A validação busca eliminar problemas de ambiguidade e inconsistência. A preocupação desta fase é com a qualidade do Documento de Requisitos produzido. Essa etapa é muito importante porque a ocorrência de erros nesse tipo de documento ocasiona grandes custos em todo decorrer do desenvolvimento do sistema. Diversos custos extras podem ser evitados com um processo de validação de requisitos bem feito. O objetivo dessa fase é liberar o Documento de Requisitos contendo de forma clara e consistente o que deve ser implementado.
- **Gerenciamento de Requisitos:** é uma atividade que deve ocorrer ao longo de todo desenvolvimento e que tem como principais objetivos organizar e armazenar os requisitos, bem como gerenciar suas mudanças ([SOMMERVILLE, 2011], [ZISMAN, SPANOUDAKIS, 2004], [CLELAND-HUANG *et al.*,2012]). Essas mudanças ocorrem tanto pela constatação de novas funcionalidades, quanto pela necessidade de se corrigir problemas nos requisitos registrados até o momento. Dentro deste tópico é que estão inseridos os aspectos de rastreabilidade de requisitos, detalhados na sequência deste capítulo.

2.3 Rastreabilidade de requisitos

Conforme mencionado anteriormente, enquanto os requisitos são levantados, documentados e validados, faz-se necessário o seu gerenciamento. Sempre que os requisitos levantados com os clientes forem alterados, os planos de software, os artefatos e as atividades afetadas devem sofrer ajustes para continuarem consistentes. O fato é que os requisitos são ativos e estão em uso durante todo o ciclo de vida, sendo a base para a modelagem do sistema.

Ao longo do processo de desenvolvimento de software, os requisitos sofrerão alterações. Existem várias razões que justificam a mudança, mas a principal delas é que, durante o desenvolvimento do software, cliente e usuários passam a entender melhor suas necessidades e propõem alterações no sistema inicialmente solicitado. Algumas dessas necessidades são percebidas somente após parte do sistema ter sido implantado e utilizado pelos clientes. É comum o cliente receber a primeira versão do software e questionar aqueles que o desenvolveram sobre a possibilidade de novas funcionalidades ou de alterar as existentes. Essa situação gera problemas para os gerentes de projetos, que sempre se questionam sobre o impacto dessas alterações propostas.

Seria uma utopia congelar os requisitos após a etapa de validação (ROCHA et al., 2001) uma vez que, como dito anteriormente, as regras de negócio e a dinâmica das organizações não são estáveis. Assim como os requisitos adaptam-se às mudanças, os sistemas também têm que se adaptar. Essa habilidade de adaptação é mérito, em grande parte, do processo de gerenciamento de requisitos. A capacidade de adaptação do processo de desenvolvimento às alterações nos requisitos pode ser um diferencial estratégico das empresas de software. As empresas que detêm essa vantagem competitiva podem oferecer produtos com melhor qualidade e ainda prever, com mais precisão, o impacto de uma mudança no software proposta pelo usuário, sendo possível avaliar custos e prazos de forma mais eficiente. Dessa forma, a agilidade no processo de tratamento dessas mudanças – que certamente acontecerão – é fundamental. Um fator crítico de sucesso para essa agilidade é a possibilidade de rastreabilidade dos requisitos.

A rastreabilidade de requisitos refere-se à habilidade de descrever e acompanhar a vida de um requisito em ambas as direções [(GOTTEL, FINKELSTEIN, 1997), (GUO et al., 2009)]. Esse controle deve abranger toda a sua existência, desde a fonte de origem - quando esse foi levantado, especificado e validado - passando pela fase de projeto, implementação e terminando na fase de testes do produto, sendo, portanto, parte do processo de desenvolvimento do software (DI THOMMAZO et al. 2012). É uma técnica que permite a visibilidade do relacionamento de dependência entre os requisitos, identificando relacionamentos existentes. Além

dessa dependência, deve ser possível identificar a dependência entre os requisitos e os artefatos que eles originaram e se relacionam.

De acordo com Guo e outros (2009), a rastreabilidade de requisitos é uma importante atividade no gerenciamento de requisitos, uma vez que ela pode fornecer base para a evolução das mudanças nos requisitos, além de atuar diretamente na garantia da qualidade do processo de software. Castor e outros (2004) classificam a rastreabilidade como um fator significativo na eficiência do gerenciamento de projeto de software e da qualidade do software desenvolvido, sendo que a falha na rastreabilidade despende tempo e gera altos custos nas correções e adaptações no software.

Segundo Cleland-Huang e outros (2012), dentre as vantagens do uso da rastreabilidade, destacam-se:

- análise de impacto das mudanças;
- suporte à geração de testes de regressão;
- identificação de candidatos ao reuso;
- acompanhamento do progresso do projeto;
- diminuição de retrabalho.

De acordo com Zisman e Spanadousk (2004) e Cleland-Huang e outros (2012), existem dois tipos de rastreabilidade:

- Horizontal: quando o relacionamento entre os requisitos acontece entre diferentes artefatos. Nesse caso, acompanha-se como o requisito foi definido no Documento de Requisitos, como ele foi abstraído em na modelagem, depois para o código fonte, até chegar na atividade de testes.
- Vertical: quando os requisitos são analisados em um mesmo artefato, como, por exemplo, o Documento de Requisitos. Analisando-se os requisitos funcionais desse artefato, é possível identificar o relacionamento entre eles e, a partir desta informação, gerar a RTM. O foco desta tese está inserido no contexto da rastreabilidade vertical.

2.3.1 Matriz de Rastreabilidade de Requisitos (RTM)

De acordo com Goknil e outros (2011), apesar de existirem várias pesquisas que tratam da rastreabilidade entre os requisitos e outros artefatos, uma menor atenção é dada para o relacionamento dos requisitos entre si, isto é, para a rastreabilidade vertical. Os autores também afirmam que esse relacionamento influencia várias atividades durante o desenvolvimento de software, tais como a verificação de consistência dos requisitos e gerenciamento de mudanças. Sommerville (2011) afirma que a forma para mapear essa dependência entre os requisitos é a criação da RTM. Cuddeback e outros (2010) relatam que a RTM dá suporte a muitas atividades de engenharia de software e atividades de validação e verificação, tais como análise de impacto de mudanças, engenharia reversa, reuso e testes de regressão. Além disso, eles também dizem que a criação da RTM é trabalhosa e está sujeita a erros, motivos pelos quais, frequentemente, ela não seja gerada ou atualizada.

Conforme mencionado anteriormente, a RTM indica o relacionamento existente entre os requisitos do software da seguinte forma: cada requisito funcional é representado em uma linha e uma coluna da matriz, e a dependência entre os requisitos é registrada na célula correspondente à intersecção linha/coluna. Conforme já mencionado neste trabalho, vários autores ([TAVASSOLI, 2012], [SOMMERVILLE, 2011], [GOKNIL, 2011], [GUO et al., 2009], [MUNSON, NGUYEN, 2005]) comentam a importância e a necessidade da RTM para o processo de desenvolvimento de software, uma vez que essa matriz permite a previsão do impacto de uma mudança ou da inserção de um novo requisito no sistema. Sommerville (2011), além de enfatizar a dificuldade de se obter e manter esse tipo de matriz, propõe que a indicação da dependência entre os requisitos determine, além da existência ou não de dependência, uma forma de registrar o nível dessa dependência.

Na Tabela 3, exemplifica-se uma RTM. Neste exemplo, quando existe dependência entre dois requisitos funcionais isso fica representado por um "X" na matriz, por exemplo: o RF1 possui relação de dependência com o RF2 e o RF5. A matriz gerada é simétrica, ou seja, $M(i,j)=M(j,i)$.

Tabela 3 - Exemplo de Matriz de Rastreabilidade de Requisitos

Requisitos	RF1	RF2	RF3	RF4	RF5	RF6	RF7	RF8
RF1	-	X			X			
RF2	X	-			X			
RF3			-				X	
RF4				-				
RF5	X	X			-		X	
RF6						-		
RF7			X		X		-	X
RF8							X	-

2.3.2 Detecção automática de rastreabilidade

De acordo com Cleland-Huang e outros (2012), apesar dos primeiros estudos sobre rastreabilidade – ainda no início dos anos 90 – destacarem a importância do estabelecimento da rastreabilidade para o processo de desenvolvimento de software, atualmente o foco está em estudos que propõe formas para o suporte automatizado na determinação desta rastreabilidade. Isso ocorre uma vez que a tarefa de criar e manter a rastreabilidade é extremamente trabalhosa e sujeita a erros ([PORT *et al.*, 2011] [CUDDERBACK *et al.*, 2010], [OLSEN *et al.*, 2007], [CLELAND-HUANG, 2006]). O foco desta tese está na determinação automática da rastreabilidade, conforme mencionado no Capítulo 1, e que será detalhado no decorrer deste trabalho.

De acordo com Wang e outros (2009), dentre as principais técnicas utilizadas para o estabelecimento automático da rastreabilidade, destacam-se os modelos baseados em espaço vetorial, a modelagem probabilística e a Latent Semantic Indexing (LSI). Essas três abordagens fazem uso de mineração de texto para determinação da rastreabilidade. Segue uma breve descrição delas e os principais trabalhos feitos com elas:

- Modelos baseados em Espaço Vetorial: pelo Modelo do Espaço Vetorial, todo o texto dos requisitos é representado em um vetor. Cada posição do vetor corresponde a uma palavra da sentença com a frequência de sua ocorrência no texto. A medida do cosseno entre os vetores (calculada pelo produto vetorial desses vetores) calcula a distância entre

eles. Dessa forma, quanto mais próximos os dois vetores, mais similares são as sentenças a que correspondem. De acordo com essa similaridade, são determinados os links de rastreabilidade. Dentre as iniciativas que fazem uso do Modelo do Espaço Vetorial para rastreabilidade entre requisitos, pode-se destacar [DE LUCIA *et al.*, 2006], [HAYES *et al.*, 2003, 2006]), entre requisitos e código fonte ([ABADI *et al.*, 2008], [ANTONIOL *et al.*, 2000, 2000], [DE LUCIA *et al.*, 2006], [MARCUS, MALETIC, 2003], [MARCUS *et al.*, 2005] e entre diagramas de UML e código fonte [DE LUCIA *et al.*, 2006], [SETTIMI *et al.*, 2004].

- Modelagem Probabilística: no contexto da recuperação de informação, a modelagem probabilística é utilizada para classificar documentos em ordem decrescente de probabilidade de relevância de acordo com a consulta (CRESTANI, 1998). Quando se pensa na determinação de links de rastreabilidade, considera-se que, caso os termos presentes na “origem” (artefato ou trecho de um artefato) tenham alta probabilidade de estar presentes no “destino” (também representado por um artefato ou trecho de um artefato), maior é a chance de existir um link de rastreabilidade entre eles. Vários modelos probabilísticos têm sido propostos para recuperar as ligações entre artefatos de software ([ABADI *et al.*, 2008], [ANTONIOL *et al.*, 1999], [CLELAND-HUANG *et al.*, 2005]). O modelo probabilístico tem sido utilizado para recuperar as ligações entre os requisitos ([GIBIEC *et al.*, 2010], [CLELAND-HUANG *et al.*, 2010]), os requisitos e diagramas UML (CLELAND-HUANG *et al.*, 2005), os requisitos e código fonte ([ABADI *et al.*, 2008], [ANTONIOL *et al.*, 2000, 2000, 2002], [PENTA *et al.*, 2002]), e páginas de manual e código fonte (ANTONIOL *et al.*, 1999, 2000)
- Latent Semantic Indexing (LSI): de acordo com a LSI, não se considera somente a frequência das palavras nos textos (DEERWESTER *et al.*, 1990). Busca-se também o contexto em suas construções, as dependências entre termos e documentos, além das associações entre termos e documentos. Essa técnica assume que existe uma "estrutura latente" no uso das palavras, algo que pode ser parcialmente ponderado

não somente pela ocorrência de determinada palavra num texto. A partir deste princípio são utilizadas técnicas estatísticas para estimar esta estrutura latente. Por exemplo, os termos "aluno" e "estudante" provavelmente irão aparecer em diferentes documentos com termos relacionados, tais como "disciplina" e "matrícula". Partindo da determinação dessa "estrutura latente" é feita a determinação dos links de rastreabilidade. Hayes e outros (2006) utilizaram a LSI para identificar de forma automática a rastreabilidade.

Conforme dito anteriormente, essas abordagens fazem uso de técnicas de mineração de texto. Segundo Feldman e Sanger (2006), mineração de texto pode ser definida como um processo de conhecimento intensivo em que o usuário interage com uma coleção de documentos por meio de um conjunto de ferramentas de análise.

Em geral, para que sejam aplicados os algoritmos de mineração de texto é necessário que seja aplicado um pré-processamento no texto. O objetivo do pré-processamento é extrair, de textos escritos em linguagem natural, uma representação estruturada, concisa e manipulável por algoritmos de agrupamento de textos. Segundo Rezende e outros (2011), para atingir esse objetivo, são executadas atividades de tratamento e padronização na coleção de textos, seleção dos termos (palavras) mais significativos e, por fim, representação da coleção textual em um formato estruturado que preserve as principais características do texto.

Segundo Cleland-Huang e outros (2012), os principais passos de pré-processamento aplicados quando o objetivo é a determinação da rastreabilidade são:

- Normalização do texto: promove a retirada de espaços em branco, operadores, símbolos lógicos e números.
- Divisão de termos: divide palavras compostas em duas palavras com maior sentido. Alguns autores (ENSLEN *et al.*, 2009) salientam que esse passo deve ser executado com precisão pois, do contrário, podem piorar a eficácia da detecção. Esse passo é sensível ao idioma em que o texto foi escrito.
- Eliminação de *stopwords*: retira do texto termos que não possuem sentido semântico para os documentos, como artigos, pronomes, advérbios e conjunções. Esses termos são chamados *stopwords*. Uma

característica dessa técnica é que ela também é sensível ao idioma e exige uma lista de *stopwords* previamente definida.

- *Stemming*: reduz as palavras nos seus radicais. Esse processamento é relativamente mais complexo que os anteriores, uma vez que exige a análise morfológica de cada um dos termos. O objetivo dessa etapa é que, reduzindo as palavras a seus radicais, elas tenham o mesmo peso nos algoritmos de processamento que serão aplicados. Um exemplo do benefício do uso do *stemming* é que palavras como “professores”, “professora” e “professor” seriam todos reduzidos ao radical e teriam o mesmo peso nos algoritmos de similaridade utilizados.

Após a representação da coleção de documentos em um formato estruturado, é possível aplicar técnicas de extração de conhecimento utilizando sistemas de aprendizado. Nessa etapa, o objetivo é descobrir padrões úteis e desconhecidos presentes nos documentos. Caso os documentos estejam representados no formato de uma tabela de atributo-valor, como exemplificado anteriormente, métodos de Recuperação de Informação e Aprendizado de Máquina são comumente aplicados, assim como ocorre no processo de mineração de dados.

Na Seção 4.1.3, é detalhada uma abordagem proposta pelo autor deste trabalho para determinação da rastreabilidade com uso de técnicas de mineração de texto.

2.4 . Técnicas de Inteligência Computacional

Nesta seção são apresentadas duas técnicas de inteligência computacional que foram utilizadas no contexto deste trabalho. A proposta foi combinar as abordagens RTM-E e RTM-NLP com o intuito de criar uma nova RTM que considere aspectos tratados pela inteligência computacional. As técnicas exploradas foram: sistemas fuzzy e redes neurais.

Utilizou-se sistemas fuzzy, pois eles permitem tratar a incerteza existente na definição dos links de rastreabilidade. Com relação às redes neurais, foi possível treinar uma rede neural capaz de identificar o nível de rastreabilidade entre dois

requisitos funcionais, fornecendo-se como entrada os valores numéricos de dependência encontrados pelas abordagens RTM-E e RTM-NLP. Foram utilizadas as redes neurais *Multilayer-Perceptron*. Para melhorar as abordagens propostas, utilizaram-se também algoritmos genéticos para aprimorar as funções de pertinência do sistema fuzzy e árvores de decisão para melhorar os intervalos que definiam se um link de rastreabilidade era nulo, forte ou fraco.

Na sequência, são detalhadas as técnicas comentadas anteriormente, assim como a relação delas com esta tese de doutorado.

2.4.1 Sistema de Inferência Fuzzy

Um sistema de inferência fuzzy é baseado na lógica fuzzy, também conhecida como lógica nebulosa, e foi desenvolvida por Zadeh (1965). Ela propõe que, ao invés de usar valores de falso ou verdadeiro, seja utilizada uma variação entre a completa falsidade e a verdade absoluta. Em geral, ela é utilizada no tratamento de problemas em que se lidam com incertezas, dados vagos ou valores aproximados. Nesta seção diversos conceitos relacionados com um sistema de inferência fuzzy serão detalhados. Artero (2009) utiliza um exemplo simples para explicar os conceitos envolvidos: a análise da classificação da temperatura de um líquido. Neste exemplo não se pode dizer simplesmente se o líquido está quente ou frio. Existe também um estado intermediário: morno.

2.4.1.1 Relação de pertinência na teoria dos conjuntos nebulosos

Na teoria clássica dos conjuntos, existem apenas duas possibilidades de pertinência de um elemento em relação a um dado conjunto: ou o elemento pertence ou ele não pertence ao conjunto (ARTERO, 2009). Sendo assim, o grau de pertinência do valor x em um determinado conjunto A ($\mu_A(x)$) é dado por:

$$\mu_A(x): U \rightarrow \{0,1\}$$
$$\mu_A(x) = \begin{cases} 1 & \text{se } x \in A \\ 0 & \text{se } x \notin A \end{cases}$$

Na Teoria de Conjuntos Fuzzy, a relação de pertinência é dada por uma função cujos valores pertencem ao intervalo fechado real entre 0 e 1. Nessa forma de

representação, ao invés de um elemento pertencer ou não a um conjunto, ele tem um grau de pertinência ao conjunto.

$$\mu_A(x): U \rightarrow [0,1]$$

Nesse caso, não existe mudança brusca para dizer se um elemento está em um conjunto ou em outro. Considerando o exemplo dado anteriormente da temperatura de um líquido, se dissermos que o grau de pertinência para classificarmos o líquido como “quente” for de 0.8, ou seja, $\mu_{QUENTE}(x) = 0.8$, significa que x pertence ao conjunto “quente” com 80% de confiança.

2.4.1.2 Fuzzyficação

De acordo com Artero (2009), para que os valores absolutos possam ser processados em um sistema de inferência fuzzy, é preciso que eles sejam convertidos em valores nebulosos. O processo de converter um número real na sua representação fuzzy, com o intuito de expressá-lo como uma medida de imprecisão, é chamado de “fuzzyficação”. Para representar os valores exatos em valores nebulosos são utilizadas funções matemáticas chamadas de função de pertinência. Nesta função, é representado como a pertinência irá variar. Considerando o mesmo exemplo anterior da temperatura de um líquido, a Figura 2.1, mostra a curva de “fuzzyficação” para a temperatura fria.

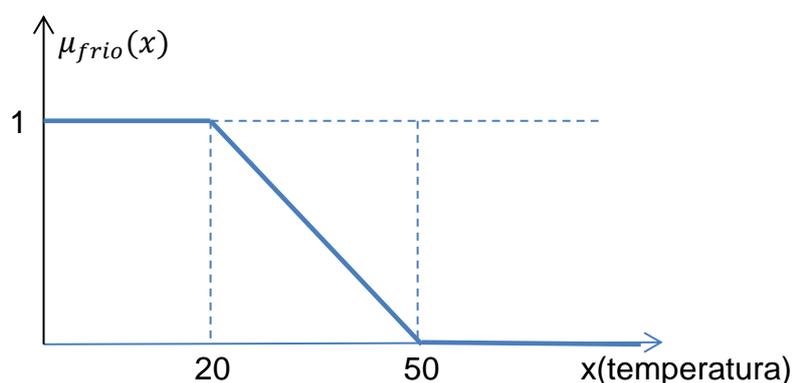


Figura 2-1. Exemplos de curva de fuzzyficação (adaptado de Artero (2009)).

2.4.1.3 Variáveis linguísticas

Variáveis linguísticas são variáveis cujos valores são palavras ou sentenças em linguagem natural em vez de números (ZIMMERMANN, 1991). Segundo Castro (2004), grande parte da experiência e do conhecimento dos seres humanos está

armazenada na forma linguística, mais geral e imprecisa, dificultando seu aproveitamento por computadores, os quais exigem informações precisas. Com o conceito de variáveis linguísticas, problemas naturalmente imprecisos e complexos passam a ser manipuláveis por computadores. Segundo o autor, a tradução da informação linguística (imprecisa) para a informação numérica (precisa) utilizada por computadores pode ser realizada usando os conjuntos fuzzy.

De acordo com Artero (2009), uma variável linguística é uma variável cujos valores são nomes de conjuntos nebulosos. No exemplo que estamos tratando a temperatura de um líquido, podemos pensar que a variável linguística “temperatura” pode assumir os valores (conjuntos nebulosos) “fria”, “morna” e “quente”.

A Figura 2.2 ilustra como variável linguística “temperatura” pode ser representada pelos possíveis valores.

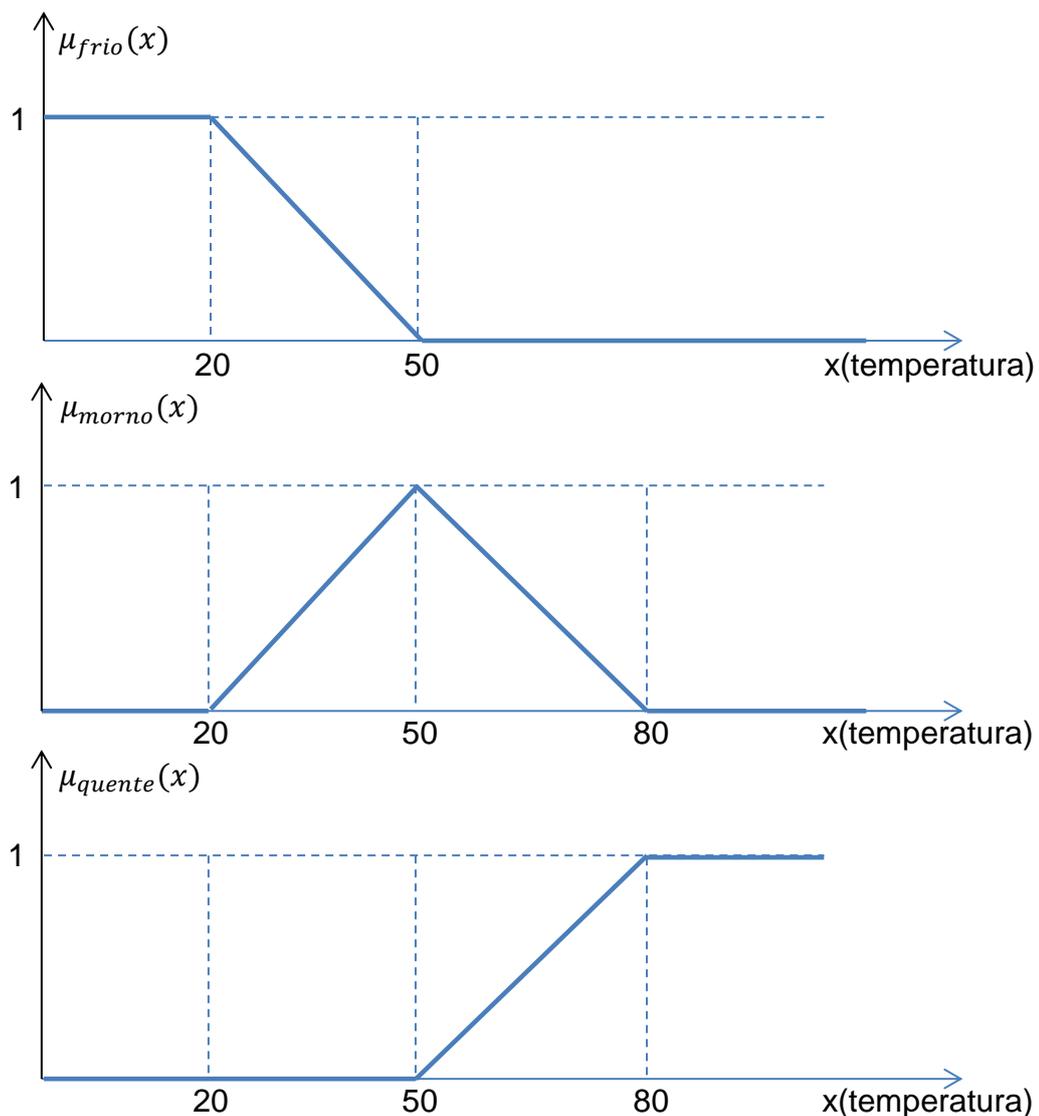


Figura 2-2. Curva de fuzzyficação para a variável linguística “temperatura” (adaptado de Artero (2009)).

2.4.1.4 Inferência Fuzzy

A inferência é realizada a partir de regras de produção nas quais os termos (tanto o antecedente como o conseqüente) são conjuntos nebulosos. A ideia é que especialistas na área do problema representem o conhecimento necessário para as tomadas de decisão por meio de regras.

Considerando-se que temos duas variáveis linguísticas denominadas peso e estatura, pode-se pensar em uma regra de inferência que define que, para uma pessoa ter aptidão para ser jôquei, é recomendável que seja leve e de baixa estatura. Sendo assim, deve ser criada uma regra para essa inferência:

Se (PESO_{leve}) E (ESTATURA_{baixa}) ENTÃO JOQUEI

O uso da inferência fuzzy permite que o conhecimento contido nas funções de pertinência possa ser combinado agregando conhecimento no sistema fuzzy.

2.4.1.5 Defuzzyficação

Após o processo de inferência utilizando as regras fuzzy que foram criadas, pode ser interessante que o sistema fuzzy produza uma saída numérica e não em termos linguísticos. Este processo que mapeia saídas fuzzy em valores numéricos é denominado defuzzificação (KLIR, YUAN, 1995). Três dos principais métodos para o processo de defuzzificação são o do máximo critério (MAX), o da média dos máximos (MOM) e o do centro de área (CoA).

O valor numérico da saída obtido pelo método do “máximo critério” corresponde ao maior grau de pertinência dentre os conjuntos fornecidos. No método “média dos máximos”, o valor de saída é gerado calculando-se a média entre os dois elementos de maior grau de pertinência das variáveis linguísticas de saída. Já o método de “centro de massa” encontra o centro de gravidade da distribuição de saída fuzzy para determinar o valor numérico. Para isso, é calculada a área da curva da variável linguística de saída e encontrado o índice correspondente que divide esta área na metade. Trata-se do método mais utilizado para realizar a defuzzificação.

2.4.1.1 Sistemas Fuzzy no contexto deste trabalho

Sistemas Fuzzy tem sido usada em várias áreas da Engenharia de Software: na área de Engenharia de Requisitos pode-se destacar o trabalho de Ramzan e outros (2009), que faz a priorização dos requisitos com base em um sistema fuzzy, e também o trabalho de Yen e outros (1996), em que um sistema fuzzy auxilia na análise da precisão dos requisitos coletados; na área de métricas, MacDonell e outros (1999) também fizeram uso de sistemas fuzzy para propor métricas para o processo de desenvolvimento de software; na área de reuso, Sandhu e Singh (2006) utilizaram um sistema fuzzy para analisar a qualidade dos componentes criados.

No contexto deste trabalho, utilizou-se lógica fuzzy, uma vez que ela permite tratar a incerteza inerente ao processo de determinação dos links de rastreabilidade. Ao invés de definir o valor fixo para cada faixa de intervalo, o sistema fuzzy permitiu que fosse utilizada uma função de pertinência para cada categoria (“sem dependência”, “dependência fraca” ou “dependência forte”), onde o grau de pertinência de cada categoria varia de acordo com o percentual de dependência entre os requisitos. Cada uma das abordagens utilizadas como entrada foi modelada com suas funções de pertinência. Elas foram combinadas através das regras fuzzy, no processo de inferência, tendo o resultado determinado com base na curva de defuzzificação. Essa abordagem foi denominada RTM-Fuzzy e é detalhada na Seção 4.4.

2.4.2 Redes Neurais

As primeiras propostas para a criação de redes neurais artificiais surgiram na década de 40 com o neurofisiologista McCulloch e com o matemático Pitts. Na proposta desses autores, deveriam ser utilizados elementos de hardware como resistores variáveis conectados a amplificadores para se comportarem como os neurônios. Porém, em 1969, Minsk e Papert demonstraram uma série de limitações dessa teoria, ficando adormecida até a década de 80 quando voltaram a atrair atenção

da comunidade científica com o surgimento de novos modelos mais precisos (ARTERO, 2009).

2.4.2.1 Neurônios Artificiais

As redes de neurônios artificiais são modeladas em relação ao cérebro humano e são compostas por vários neurônios artificiais. Os neurônios artificiais tendem a ter menos conexões que os neurônios biológicos. Os neurônios artificiais foram criados por McCulloch e Pitts (1943). Cada neurônio (ou um nó) em uma rede neural recebe uma série de entradas. Uma função, chamada de função de ativação, é aplicada a esses valores de entrada e, como resultado da função, é gerado o nível de ativação do neurônio, sendo este o valor de saída do neurônio. A função de ativação pode variar. As principais delas estão representadas na Figura 2.3. O eixo x do gráfico representa o valor de entrada para cada neurônio, e o eixo y representa a saída (ou o nível de ativação) do neurônio.

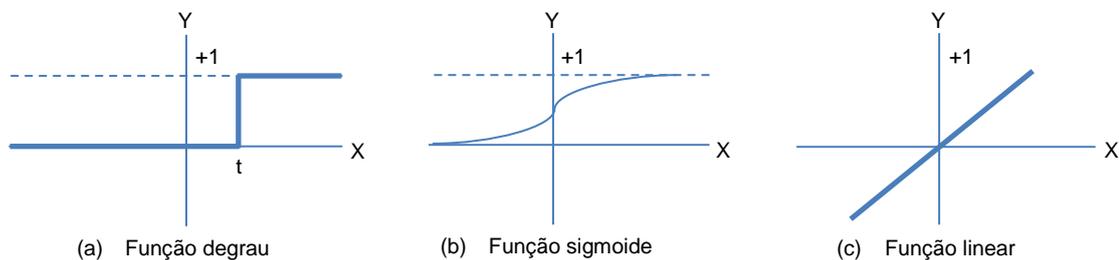


Figura 2-3. Exemplos de função de ativação (adaptado de Coppin (2010)).

A figura 2.4 mostra a representação de um neurônio recebendo diversas entradas (x_i) com seus pesos (w_i). O valor de saída do neurônio depende das entradas, dos pesos e da função de ativação utilizada. O comportamento do neurônio pode ser definido pela Equação 1:

$$X = \sum_{i=1}^n w_i x_i \quad (1)$$

De acordo com o valor de X calculado, o valor de Y é encontrado na função de ativação, gerando a saída do neurônio.

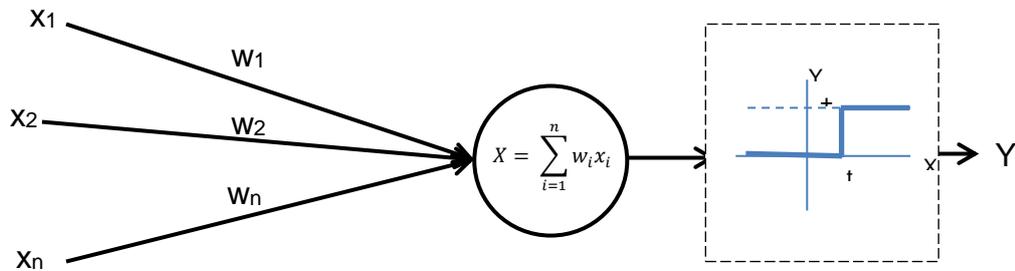


Figura 2-4. Esquema de um neurônio artificial

2.4.2.2 Classificação das redes neurais

Uma rede neural consiste em um conjunto de neurônios artificiais conectados. A forma como eles se conectam determina a topologia da rede. Na Figura 2.5 é apresentado um exemplo de rede neural.

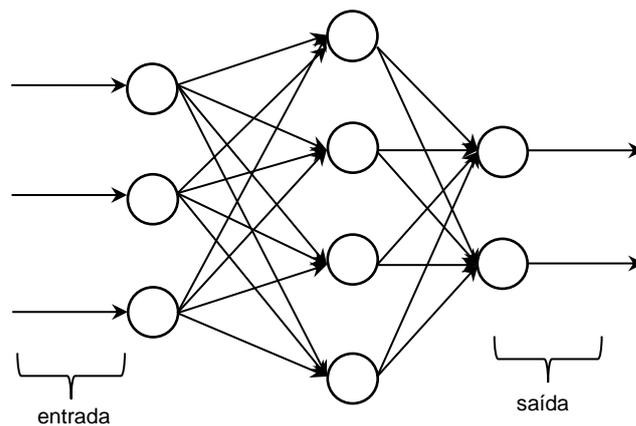


Figura 2-5. Exemplo de uma rede neural

A rede neural pode ter várias classificações, que podem depender da quantidade de camadas ou do tipo de conectividade (parcialmente conectada ou totalmente conectada). Outra possibilidade é pensar no fluxo dos sinais processados;

nesse caso temos as redes de alimentação adiante (*feed-forward*) ou com realimentação (*feed-backward*).

Outra possibilidade de classificação é relacionada com o modo como é feito o treinamento. O treinamento consiste em apresentar padrões à entrada da rede para que ela possa ir modificando seus pesos, de modo que as suas saídas apresentem uma resposta adequada quando as entradas receberem valores parecidos, mas não necessariamente iguais, com os usados no treinamento (ARTERO, 2009). Sendo assim, o treinamento da rede pode ser supervisionado (quando é apresentado para a rede neural a entrada e a saída desejada) e não supervisionado (quando é apresentado somente a entrada da rede neural e ela se encarrega de definir os valores de saída).

2.4.2.3 Redes Neurais Multilayer-Perceptron(MLP)

A rede neural MLP é composta por nós-fonte que formam a camada de entrada da rede, uma ou mais camadas intermediárias e uma camada de saída. Com exceção da camada de entrada, as outras são compostas por neurônios. Além disso, uma rede MLP tem conectividade *feed-forward*, ou seja, a saída de cada neurônio se conecta unicamente com todos os neurônios da próxima camada, sem a presença de laços de realimentação. Dessa forma, o sinal se propaga na rede de forma progressiva. Em geral, as redes MLP são completamente conectadas.

Com relação ao treinamento, praticamente todas as MLPs são treinadas de forma supervisionada com o algoritmo de retropropagação do erro, chamado *Backpropagation*. Esse algoritmo tem basicamente dois passos: o primeiro utiliza um valor de entrada e o processa com os valores dos pesos existentes nas ligações entre os neurônios até a última camada (saída) – inicialmente os valores dos pesos são gerados de forma randômica; o segundo passo faz o caminho contrário, do final para o início da rede, recalculando o valor dos pesos. Nesse cálculo do peso, busca-se um resultado de saída mais próximo que o esperado, ou seja, com um erro menor que o calculado anteriormente. A condição de parada pode ser um determinado número de iterações (idas e vindas) ou quando o erro acumulado da rede estiver abaixo de algum valor determinado.

2.4.2.4 Redes neurais no contexto deste trabalho

As redes neurais têm a capacidade de adquirir conhecimento para reconhecimento de padrões (COPPIN, 2010). Utilizou-se uma rede neural MLP tendo como entradas as abordagens RTM-E e RTM-NLP. As saídas foram três: “sem dependência”, “dependência fraca” e “dependência forte”. Para o treinamento da rede foram utilizados dados de experimentos coletados (DI THOMMAZO *et al.*, 2012). O conhecimento adquirido nesse experimento foi modelado na rede neural para reconhecer o tipo de dependência existente. Essa abordagem foi denominada RTM-N e é detalhada na Seção 4.5.

2.4.3 Algoritmos Genéticos

Os algoritmos genéticos fazem parte da computação evolutiva e se baseiam em mecanismos evolutivos encontrados na natureza. Foram inspirados na Genética de Mendel e na Teoria da Evolução de Darwin. Fazem uso de diversos conceitos dessas teorias, tais como reprodução, mutação e avaliação da capacidade que os indivíduos de uma determinada população têm para sobreviver. Assim como na Teoria da Evolução de Darwin, espera-se que as características dos melhores indivíduos sejam transferidas para a próxima geração.

Segundo Artero (2009), utiliza-se de algoritmos genéticos para resolver problemas para os quais não existe um algoritmo conhecido. Para isso, gera-se um conjunto de valores aleatórios iniciais – chamados cromossomos ou indivíduos – que são utilizados para avaliar se algum deles corresponde à solução do problema a ser resolvido. Caso a solução não seja encontrada entre os indivíduos dessa geração inicial, alguns indivíduos são descartados e outros combinados através de cruzamentos para a determinação da próxima geração, que também será avaliada. Na sequência são apresentados conceitos dos algoritmos genéticos para posterior detalhamento do algoritmo em si.

- Genes: são a representação de algum parâmetro do problema estudado, podendo ser valores inteiros, reais ou cadeia de caracteres.

- Cromossomos: correspondem a uma cadeia de genes e representam cada indivíduo da população.
- Indivíduos: correspondem a um cromossomo e representam uma possível solução para o problema
- População: formada por um conjunto de indivíduos que competem pela sobrevivência e pela reprodução, com o objetivo de manter suas características nas gerações.
- Função de aptidão: é utilizada para medir a habilidade de cada indivíduo para sobreviver.

Na Figura 2-6 são representados os passos do algoritmo genético.

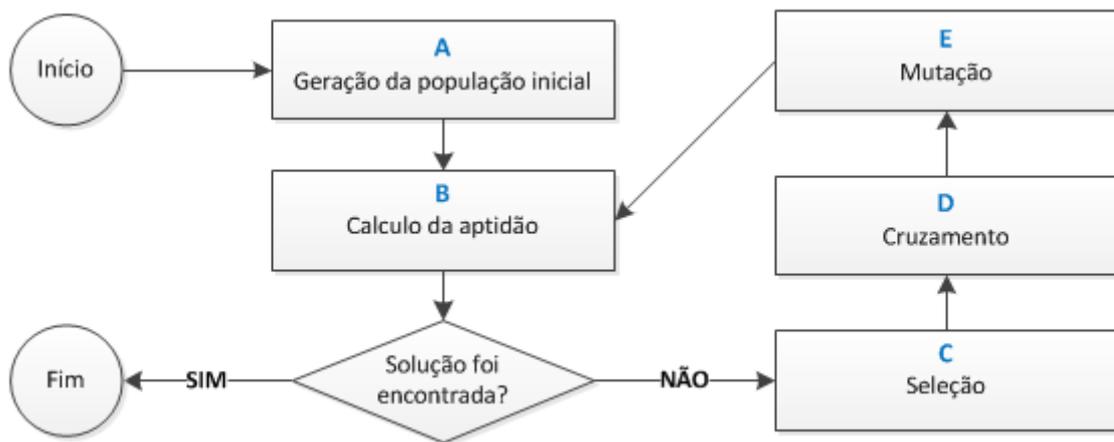


Figura 2-6. Algoritmo Genético

Conforme dito anteriormente, a primeira etapa do algoritmo genético é a geração da população inicial, composta por diversos indivíduos (cromossomos). Em geral, os indivíduos são gerados aleatoriamente (Figura 2-6-A). O próximo passo é a avaliação da população buscando-se algum indivíduo que corresponda à solução do problema (Figura 2-6-B). Essa busca é feita com o cálculo da aptidão de cada indivíduo em resolver o problema em questão. Caso seja encontrado algum indivíduo que resolva o problema, então a condição de parada é satisfeita e o algoritmo é finalizado, apresentando-se esse indivíduo como a solução do problema. Caso contrário, entra-se na fase de Seleção (Figura 2-6-C). Nesta etapa, devem ser escolhidos os indivíduos que permanecerão na próxima geração e que darão origem aos novos indivíduos através de mutações ou cruzamentos (*crossover*). As principais técnicas utilizadas para a seleção são:

- seleção aleatória: alguns dos indivíduos da população são escolhidos aleatoriamente para gerar a nova geração;
- seleção por torneio: a cada dois indivíduos escolhidos aleatoriamente, o melhor deles, segundo a função de aptidão, permanece para gerar a nova geração;
- seleção usando a roleta: a aptidão de cada indivíduo é calculada e os valores obtidos corresponderão proporcionalmente aos setores de uma roleta. Quanto maior a aptidão de um indivíduo na resolução do problema, maior a probabilidade de ele ser “sorteado” como um indivíduo que deva permanecer na próxima geração.

Definidos os indivíduos que sobreviveram à etapa de Seleção, são realizados cruzamentos (Figura 2-6-D) e mutações (Figura 2-6-E) para se definir os novos indivíduos dessa nova geração. Quando ocorre o cruzamento entre dois indivíduos, o material genético deles é trocado de forma aleatória. Em uma mutação, o cromossomo é modificado em algumas de suas partes, dando assim novas características aos indivíduos. Segundo Artero (2009), durante a criação de uma nova geração, em geral, a taxa de mutação é baixa (em torno de 1%), enquanto a taxa de cruzamento é bem maior (dentre 60% e 90%).

Após o surgimento dos novos indivíduos da população e, conseqüentemente, a criação da nova geração, os indivíduos são novamente submetidos à avaliação com o cálculo de aptidão (Figura 2-6-B), para verificar se o algoritmo encontrou alguma solução viável. Esse ciclo ocorre até que seja encontrada uma solução ou caso seja fixado um número máximo de gerações.

2.4.3.1 Algoritmos Genéticos no contexto deste trabalho

Conforme detalhado, os algoritmos genéticos permitem resolver problemas para os quais não existe um algoritmo conhecido. Também foi visto anteriormente que uma das abordagens para a criação da RTM foi o uso de sistemas fuzzy. Durante a construção do sistema fuzzy utilizado na RTM-Fuzzy, foi necessário criar as funções de pertinência para representar os valores das variáveis linguísticas. Esse processo de criação das funções de pertinência foi feito inicialmente baseado na experiência do

autor desta tese. Para encontrar as melhores funções de pertinência foram utilizados algoritmos genéticos. Todo esse processo é detalhado na Seção 4.4 e foi apresentado por Di Thommazo e outros (2014).

2.4.4 Árvores de Decisão

A árvore de decisão é uma representação gráfica das alternativas disponíveis e oferece apoio à tomada de decisão. Segundo Artero (2009), é um dos métodos de aprendizado mais utilizados na prática. O aprendizado ocorre à medida que são feitas observações em um conjunto de dados para que sejam criadas regras que são representadas graficamente na árvore de decisão.

As regras criadas fazem uso de um conjunto de instruções *if-then-else*. A decisão da árvore de decisão é obtida após a sequência de testes feitas ao longo dos nós. Os nós-folhas indicam o valor de retorno, caso a folha seja atingida após a execução dos testes. Vale destacar que os atributos de entrada podem ser discretos ou contínuos.

2.4.4.1 Árvores de Decisão no contexto deste trabalho

Conforme visto anteriormente, as árvores de decisão criam regras a partir de um conjunto de dados. Também foi comentado que as abordagens criadas classificam o nível de dependência em “sem dependência”, “dependência forte” ou “dependência fraca”. As abordagens RTM-E e RTM-NLP fazem essa classificação a partir de regras *if-else* nos percentuais determinados nas abordagens. Inicialmente essas regras foram criadas de forma empírica. Num segundo momento, a partir de dados de estudos experimentais, utilizaram-se árvores de decisão para determinar os melhores pontos de corte para definir o nível de dependência entre os requisitos funcionais. Para o processamento dos dados foi utilizada a ferramenta WEKA (<http://www.cs.waikato.ac.nz/ml/weka/>). Essa ferramenta disponibiliza diversos algoritmos de inteligência computacional. Todo esse processo é detalhado na Seção 4.2 e foi apresentado por Di Thommazo e outros (2014).

2.5 Considerações finais

Neste capítulo foram apresentados os temas envolvidos diretamente com a proposta da tese: Engenharia de Requisitos, Rastreabilidade de Requisitos e técnicas de Inteligência Computacional.

Sobre a Engenharia de Requisitos foram mostradas suas fases, sua importância e foi dado enfoque para o gerenciamento de requisitos. No contexto da rastreabilidade, foram apresentados os seus conceitos e a sua importância para o processo de desenvolvimento de software, bem como a definição da RTM e as formas para detecção da rastreabilidade. Esse tema é retomado no Capítulo 3.

Também foi apresentado um breve estudo sobre algumas técnicas de inteligência computacional: sistemas de inferência fuzzy, redes neurais, algoritmos genéticos e árvores de decisão. Essas técnicas foram utilizadas para criar e refinar abordagens para a geração da RTM.

Todos os temas apresentados neste capítulo são passíveis de discussões mais extensas e completas. Contudo, optou-se por apresentar apenas os principais conceitos e descrever de maneira sucinta cada um deles, contextualizando-os para as aplicações desta tese.

Capítulo 3

REVISÃO BIBLIOGRÁFICA

Este capítulo apresenta o planejamento da pesquisa e a revisão bibliográfica desta tese de doutorado. O planejamento foi feito com base na estratégia GQM (Goal-Question-Metric) e a bibliográfica por meio de um mapeamento sistemático. Os dados obtidos com o mapeamento sistemático deram embasamento teórico para as propostas desta tese.

3.1 Considerações Iniciais

Como apresentado no Capítulo 1, esta tese de doutorado propõe o uso de técnicas de inteligência computacional na definição e no refinamento de abordagens para a determinação da RTM com o objetivo de melhorar a acurácia na detecção de links de rastreabilidade. Para dar embasamento teórico à proposta, este capítulo apresenta a revisão bibliográfica que foi conduzida. Foram investigadas na literatura as abordagens propostas para a rastreabilidade, bem como os artefatos envolvidos na rastreabilidade. Também foi feita uma investigação buscando, nos trabalhos que tratassem da rastreabilidade entre requisitos, como essas abordagens propostas foram validadas.

Toda a estratégia utilizada é detalhada neste capítulo, bem como os resultados obtidos. Na Seção 3.2 é apresentado o planejamento da revisão bibliográfica, com base no GQM, e na Seção de 3.3 o Mapeamento Sistemático conduzido é sumarizado. Por fim, a Seção 3.4 apresenta considerações finais deste capítulo.

3.2 Planejamento

Como apresentado no Capítulo 1, Seção 1.3, o planejamento dessa pesquisa foi feito de acordo com o paradigma GQM (BASILI, CALDIERA, ROMBACH, 1996).

O planejamento inicial teve o enfoque na revisão da literatura, a fim de fundamentar a tese desenvolvida. Sendo assim, objetivos, questões e métricas foram definidas com o intuito de direcionar esta etapa, como é ilustrado na Figura 3.1.

O GQM foi dividido em duas partes. A primeira parte do GQM é referente à busca por abordagens que tratassem exclusivamente da rastreabilidade entre os requisitos, ou seja, a rastreabilidade vertical. Nessa primeira parte foram definidas duas questões de pesquisas. A primeira delas, tratada na Seções 3.2.1, buscou identificar quais as técnicas utilizadas para determinar a rastreabilidade, além das ferramentas utilizadas para determinar essa rastreabilidade. A segunda questão de pesquisa, tratada na Seção 3.2.2, buscou identificar as formas de validação das abordagens por meio de experimentação. A segunda parte do GQM é referente à rastreabilidade horizontal, considerando a rastreabilidade entre os artefatos produzidos ao longo do processo de desenvolvimento de software (documentos de requisitos, modelos, código-fonte, casos de teste, documentação, etc). Buscou-se também determinar quais as técnicas que eram utilizadas para determinar a rastreabilidade, assim como as ferramentas envolvidas nessa tarefa. Identificou-se também quais os artefatos eram rastreados ao longo do processo de desenvolvimento de software.

A revisão bibliográfica foi realizada por meio de Mapeamentos Sistemáticos, seguindo Petersen e outros (2008), sendo que cada questão do GQM resulta em uma questão de pesquisa de um mapeamento sistemático, e as métricas são atributos do formulário de extração de dados. A ferramenta StArt ([FABBRI et al., 2012], [ZAMBONI et al., 2010], [HERNANDES et al., 2010]) foi utilizada como apoio computacional para essa atividade.

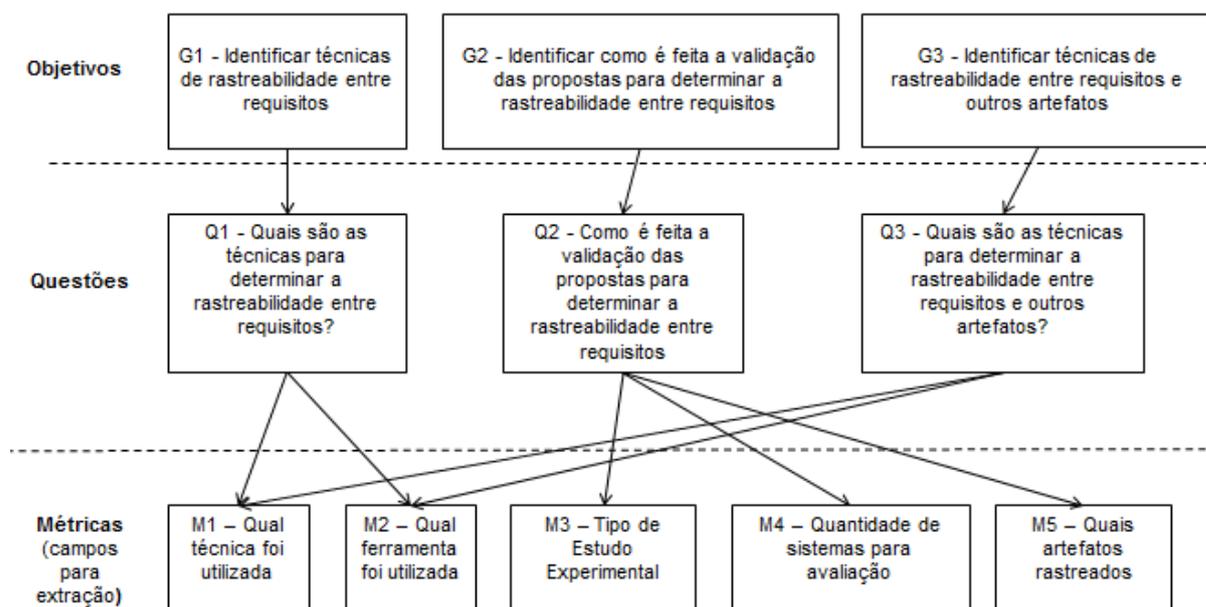


Figura 3-1. GQM de planejamento da revisão bibliográfica.

Para definir as *strings* de busca do mapeamento sistemático, ensaios foram feitos na máquina de busca SciVerse Scopus¹ para encontrar a *string* apropriada e permitir que esta fosse aplicada em cada uma das máquinas de busca selecionadas.

Inicialmente esperava-se construir *strings* de busca para cada uma das questões do mapeamento. Entretanto, durante o processo iterativo e interativo de ajuste da *string* de busca, percebeu-se que seria muito difícil conseguir recuperar estudos que tratassem exclusivamente de cada questão. Isso ocorreu, pois, em muitos casos, os *abstracts* dos artigos não definiam qual ou quais artefatos eram tratados por eles. Sendo assim, adotou-se a estratégia de construir uma *string* de busca mais ampla para a recuperação dos estudos relevantes. Conseqüentemente, o trabalho de marcação de quais artefatos seriam estudados em cada artigo se daria durante o preenchimento do formulário de extração.

A *string* de busca definida foi:

$(\text{"requirements traceability"} \vee ((\text{"tracing"} \vee \text{"trace"}) \wedge \text{"requirements"})) \wedge (\text{"software"})$

Dessa forma, foram retornados artigos que tratavam de rastreabilidade de requisitos. O segundo termo ("software") foi colocado para restringir os resultados a estudos da área da computação. Inicialmente, diversos trabalhos de áreas que não são de interesse deste trabalho eram recuperados pelas máquinas de busca.

¹ www.scopus.com

Com a *string* definida, foi criado um protocolo de pesquisa. A mesma *string* foi aplicada nas máquinas de busca IEEE, Web of Science e ACM. Destaca-se ainda que o Mapeamento Sistemático foi conduzido inicialmente em outubro de 2012 e foi atualizado em julho de 2013.

Como resultados gerais do mapeamento sistemático, foram aceitos 360 artigos que tinham como tema principal a rastreabilidade de requisitos. O primeiro filtro aplicado nesses artigos foi identificar se eles tratavam de abordagens para determinação da rastreabilidade. Ao todo, 239 artigos foram recuperados com essa característica. O restante (121 artigos) tratava de outros aspectos, tais como o processo de rastreabilidade, as boas práticas, o custo-benefício, o retorno sobre investimento, etc.

Nas seções a seguir, um resumo do mapeamento sistemático conduzido é apresentado, assim como observações acerca dos estudos. Ressalta-se que os dados apresentados neste capítulo são referentes apenas aos estudos aceitos no mapeamento.

3.2.1 Questão 1: Quais são as técnicas para determinar a rastreabilidade entre os requisitos?

Por meio do mapeamento sistemático conduzido, foi possível identificar as principais abordagens utilizadas para a determinação da rastreabilidade entre os requisitos.

Dos 239 artigos que tratavam de abordagens para determinar rastreabilidade de requisitos, 58 tratavam exclusivamente da rastreabilidade entre os requisitos. Desses 58 artigos analisados, em 11 deles não foi possível identificar qual abordagem foi utilizada, pois apenas mencionavam o uso de rastreabilidade, sem apresentar detalhes da abordagem. Dos 47 restantes, foram encontradas as seguintes categorias: Modelo do Espaço Vetorial, *Latent Semantic Indexing*, Modelo Probabilístico, Uso de Ontologia, Uso de Algoritmo de *Swarm Intelligence*, Modelos Formais e Aprendizado de Máquina, conforme o gráfico exibido na Figura 3.2.

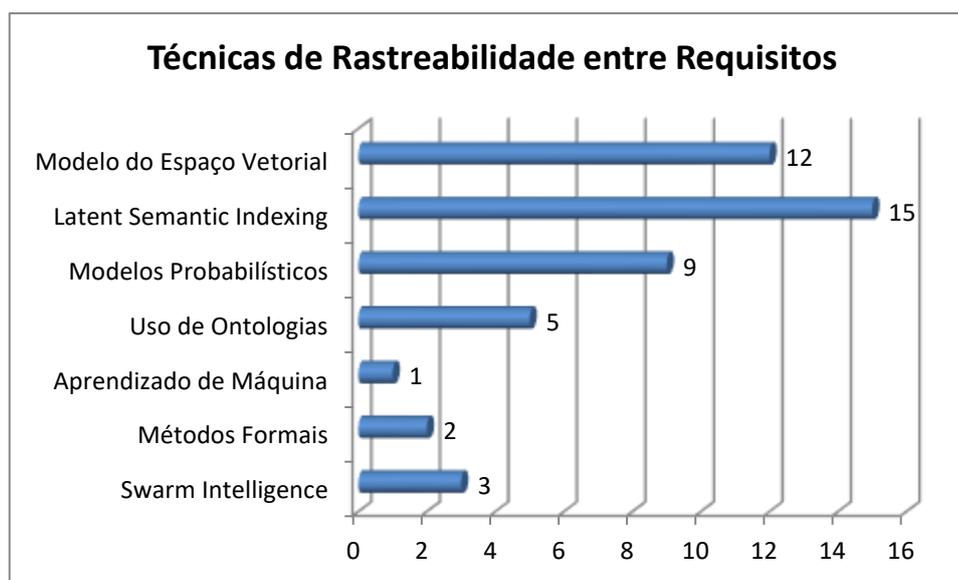


Figura 3-2. Técnicas de Rastreabilidade entre Requisitos.

No momento de classificação dos artigos, houve dificuldade em verificar como as abordagens tratavam o problema da rastreabilidade para que pudessem ser categorizadas de forma correta. Verificou-se que as três principais propostas coincidem com as que foram apontadas na Seção 2.2: Modelo do Espaço Vetorial, *Latent Semantic Indexing* e Modelo Probabilístico.

Dentre os trabalhos recuperados, destaca-se o trabalho de Wei-Keat e Hayes (2011) para a melhora da detecção de rastreabilidade com uso do Modelo do Espaço Vetorial. Neste artigo, os autores propõem que seja considerada a posição dos termos no Vetor Espacial. Dessa forma, é atribuído um peso maior quando duas palavras estão próximas em ambos os textos, gerando uma maior dependência nesses casos. Outra iniciativa com o uso do Modelo de Espaço Vetorial foi a de Aponte e Marcus (2011), em que se propõe a sumarização dos Requisitos Funcionais antes da aplicação do Modelo de Vetor Espacial. Dessa forma, os autores relatam que somente os termos mais importantes representarão cada requisito funcional, permitindo que o cálculo de similaridade do algoritmo tenha melhores resultados.

A equipe de Port (2011) utilizou a *Latent Semantic Indexing* para a determinação da rastreabilidade entre requisitos funcionais e requisitos nãofuncionais em projetos da NASA. O autor destaca a importância do estabelecimento desse tipo de rastreabilidade e aponta a *Latent Semantic Indexing* como alternativa para obtê-la. Outras alternativas com uso da *Latent Semantic Indexing* também podem ser

destacadas, como os trabalhos de Stone e Sawyer (2006) em que buscou-se extrair conhecimento tácito dos requisitos para a determinação dos links de rastreabilidade.

O Modelo Probabilístico foi a abordagem utilizada por Cleland-Huang e outros (2005) para encontrar o relacionamento entre os requisitos. Neste trabalho, o modelo probabilístico utiliza uma rede de relacionamento entre os termos encontrados nos requisitos para propor os links de rastreabilidade.

O uso de Ontologias também tem sido utilizado para a determinação da rastreabilidade. Os trabalhos de Assawamekin (2011) e Pavkovic e outros (2011) utilizaram essa abordagem. Nesses trabalhos, os autores destacam que o principal benefício do uso das ontologias é a capacidade de recuperação de links de rastreabilidade com eficácia, uma vez que os termos dos requisitos são mapeados na ontologia. Por outro lado, destacam que a rastreabilidade é dependente da criação e uso da ontologia, tarefa não tão simples segundo os próprios autores.

Outra iniciativa explorada por Sultanov e outros ([SULTANOV *et al.*, 2011],[SULTANOV, HAYES, 2010]) é o uso de *swarm intelligence*. Os autores fazem uso dos algoritmos baseados em “inteligência coletiva” para determinar a rastreabilidade.

Outra informação extraída dos artigos foi a forma como as abordagens eram implementadas: em ferramentas específicas ou em outros aplicativos de propósitos gerais. Dos 58 artigos, 27 faziam uso de alguma ferramenta de gerenciamento de requisitos ou a apresentavam. As ferramentas mais relacionadas foram a RETRO: *REquirements TRacing On target*, com seis ocorrências, e a MUPRET, com três ocorrências. Com relação aos outros 31 trabalhos que tratam da rastreabilidade entre os requisitos, as abordagens eram desenvolvidas em outros software, como MATLAB, protótipos adaptados ou bibliotecas de processamento de texto.

Também foi feito um estudo sobre o ano das publicações. Essas informações estão representadas no gráfico da Figura 3.3. Mesmo considerando o aumento no número de artigos indexados nos últimos anos, é possível observar que a quantidade de publicações na área cresceu significativamente a partir de 2011, podendo isso representar um aumento no interesse pelo tema. Esses dados foram levantados em julho de 2013. Sendo assim, acredita-se que no ano de 2013 ainda possam existir alguns estudos que não tenha sido recuperado pelas máquinas de busca.

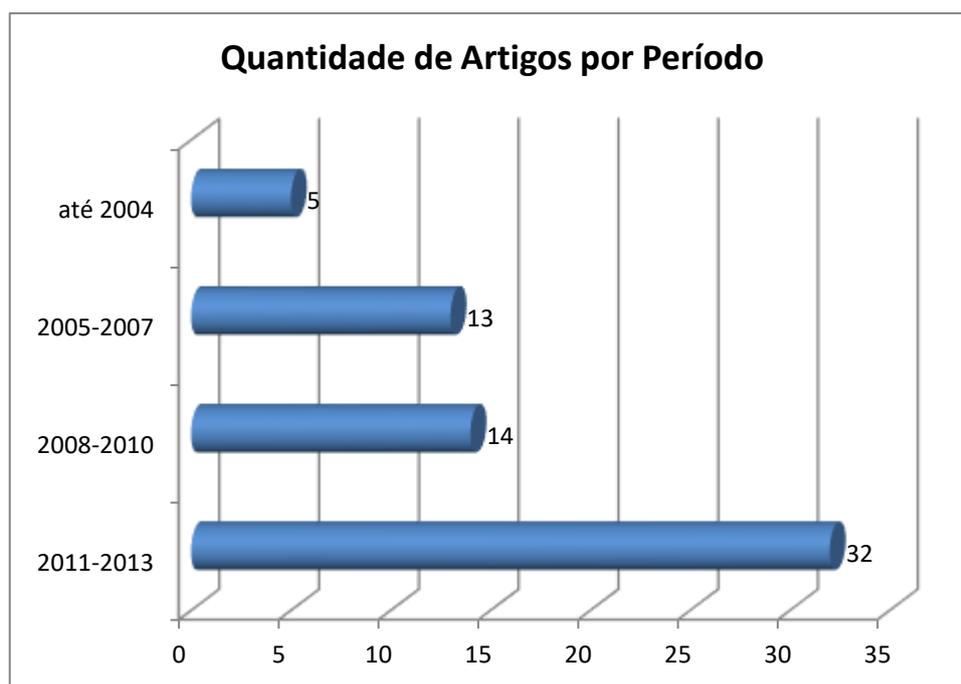


Figura 3-3. Quantidade de artigos por período.

As informações obtidas com esse mapeamento sistemático embasam o desenvolvimento da proposta desta tese, uma vez que, levantadas e estudadas as diversas abordagens para a detecção da rastreabilidade entre os requisitos, comprovou-se a originalidade do trabalho bem como o crescente interesse pela área.

3.2.2 Questão 2: Como é feita a validação das propostas para determinar a rastreabilidade entre requisitos?

A partir dos estudos identificados como propostas para a rastreabilidade entre requisitos, foi feito um levantamento para identificar como as abordagens propostas na literatura eram validadas. Foram identificadas quatro categorias para validar essas abordagens: Estudo de Caso de sistema real, Estudo de Caso de sistema não-real, Estudo Experimental Controlado e Exemplo de Uso. Nos estudos também foi identificada a quantidade de sistemas utilizados para validar cada abordagem proposta.

A Figura 3-4 mostra os resultados sobre a forma como as abordagens foram avaliadas. É possível observar a pequena quantidade de experimentos controlados que foram conduzidos para avaliar as abordagens propostas (apenas nove). Em geral,

são utilizados Exemplos de Uso ou Estudos de Caso Reais que mostram como a abordagem é utilizada para determinar a rastreabilidade.

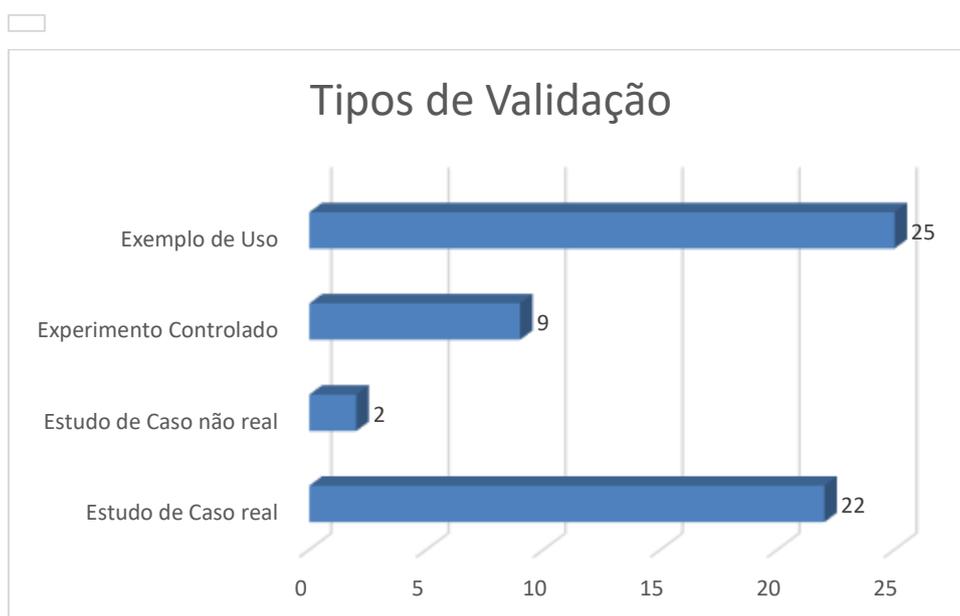


Figura 3-4. Quantidade de Artigos por tipo de validação

A Figura 3-5 mostra os resultados sobre a quantidade de sistemas utilizados para validar cada abordagem proposta. É possível observar que a grande maioria das abordagens foi validada em apenas um sistema. Isso ocorreu 37 vezes. Sete estudos utilizaram dois sistemas, enquanto que em cinco estudos foram utilizados apenas três sistemas para validar a abordagem proposta. Essas três categorias totalizam 49 estudos com propostas que sempre foram validadas através de Exemplo de Uso e Estudo de Caso Real ou Não Real. Os nove trabalhos que envolvem dez ou mais sistemas para a validação foram feitos por meio de experimentos controlados. Desses nove trabalhos, três são do autor desta tese de doutorado.

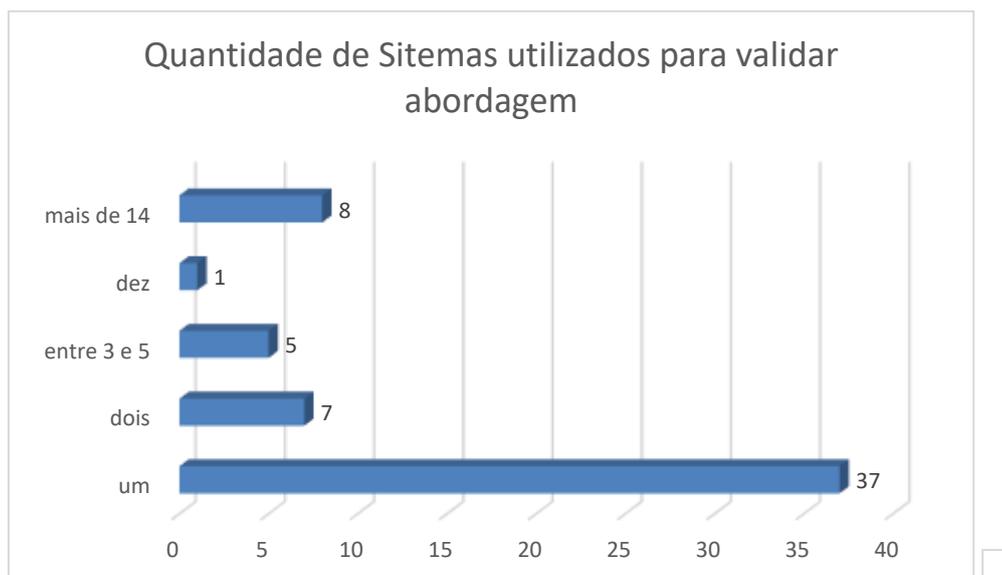


Figura 3-5. Quantidade de Sistemas utilizados para validar a abordagem

As informações obtidas com esse mapeamento sistemático mostram uma pequena quantidade de estudos experimentais controlados assim como ele evidencia a importância da realização de estudos de caso reais. Nesta tese de doutorado realizou-se um total de três experimentos controlados e dois estudos de caso reais com as abordagens propostas. Também foi conduzido um terceiro estudo de caso para mostrar os benefícios de determinar a RTM com níveis de dependência para o processo de desenvolvimento de software.

3.2.3 Questão 3: Quais são as técnicas para determinar a rastreabilidade entre os requisitos e outros artefatos?

Nessa segunda parte do mapeamento sistemático conduzido foram identificadas as principais abordagens utilizadas para a determinação da rastreabilidade entre os artefatos criados ao longo do processo de desenvolvimento de software.

Dos 181 [239 (artigos recuperados) – 58 (artigos que tratavam da rastreabilidade vertical)] artigos aceitos e analisados, em 22 deles não foi possível identificar qual abordagem foi utilizada, pois apenas mencionam o uso de rastreabilidade, sem apresentar detalhes da abordagem. Dos 159 restantes, foram encontradas as seguintes abordagens: Modelo do Espaço Vetorial, *Latent Semantic*

Indexing, Modelo Probabilístico, Uso de Ontologia, Redes Bayseanas, Modelos Formais, Aprendizado de Máquina, Uso de Regras, Rastreabilidade a partir da geração de artefatos criados com uso de ferramentas CASE e soluções mistas, conforme o gráfico exibido na Figura 3.6.

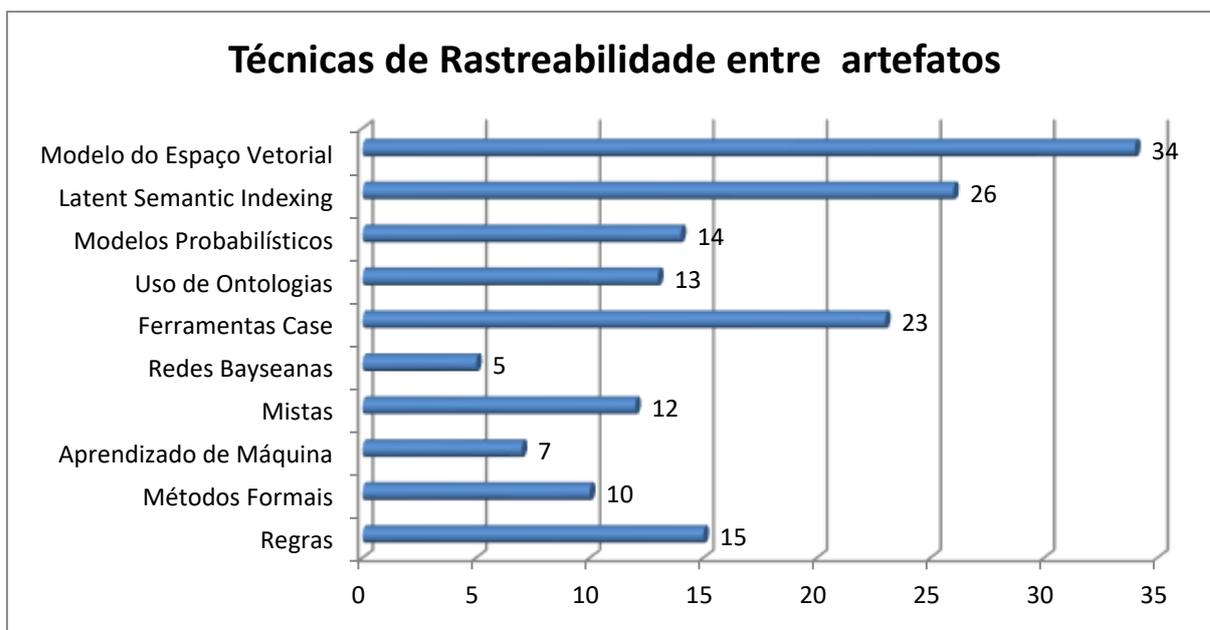


Figura 3-6. Abordagens de Rastreabilidade entre artefatos

Assim como ocorreu durante o mapeamento dos artigos que tratavam da rastreabilidade entre os requisitos, também houve dificuldade em se detectar as abordagens que tratavam da rastreabilidade entre os demais artefatos. As principais abordagens encontradas foram o Modelo do Espaço Vetorial, a *Latent Semantic Indexing* e a geração da rastreabilidade a partir de ferramentas CASE.

Muitos dos trabalhos propõem melhorias em algumas das abordagens clássicas. Um exemplo é o trabalho de Ali e outros (2011), em que é proposta uma modificação do Modelo do Espaço Vetorial para detecção de links de rastreabilidade entre os Requisitos e o Código-fonte, com base em informações dos *logs* do repositório do código-fonte. Udagawa (2011) também propôs uma melhoria no Modelo do Espaço Vetorial por meio da definição de um conjunto de termos que poderiam ser utilizados para o cálculo de similaridade.

Outra abordagem também bastante explorada é o uso da *Latent Semantic Indexing*. A equipe de Jiang (2008) propôs uma melhoria na abordagem, criando a *Incremental Latent Semantic Indexing*, que utiliza como entrada não somente os

artefatos envolvidos no desenvolvimento de software, mas também as modificações que são feitas neles.

Outra abordagem que foi muito utilizada na determinação da rastreabilidade foi o uso dos links de rastreabilidade criados com a geração automática de código a partir de modelos ou em ferramentas que, a partir dos casos de uso, criavam os casos de testes. Nesses casos, a determinação dos links é feita conforme o artefato é criado. Apesar dessa abordagem não apresentar nenhum processamento de texto, ela é muito eficaz na determinação dos links, uma vez que o próprio suporte da ferramenta já armazena as transformações, por exemplo, de classes modeladas em código-fonte.

Outra informação extraída dos artigos está relacionada com os artefatos cobertos pela rastreabilidade. Consideraram-se quatro tipos de artefatos: Documento de Requisitos, Modelos, Código-Fonte e artefatos relacionados com a etapa de testes. Sendo assim, para cada artigo lido foram identificados os artefatos cobertos pela proposta de rastreabilidade. Apesar do foco desta tese de doutorado estar na determinação da rastreabilidade entre os requisitos, foi feita a investigação das abordagens em outros artefatos para buscar possíveis alternativas que, combinadas com as técnicas já descritas na Seção 3.2.1, pudessem colaborar para a detecção da rastreabilidade de forma mais efetiva. Essas informações estão registradas na Figura 3.7.

Pelo gráfico, é possível observar que existe proposta para rastreabilidade com praticamente todos os conjuntos de artefatos que são criados ao longo do processo de desenvolvimento de software. Algumas soluções são específicas para o tipo de artefatos, especialmente as que tratam da Modelagem do software. Neste caso, durante a construção dos modelos é feita a geração automática de outros modelos ou de código-fonte. Durante esse processo são gerados os links de rastreabilidade entre os artefatos.

Também foi feito um estudo sobre o ano da publicação dos estudos que tratam da rastreabilidade entre os artefatos. Os resultados desse estudo estão no gráfico da Figura 3.8. Mesmo considerando o aumento no número de artigos indexados, é possível observar que a quantidade de publicações na área cresce significativamente em cada período de estudo. Apesar do aumento geral no número de publicações que ocorreu no tempo, isso pode representar um aumento no interesse pelo tema. Conforma já mencionado, esses dados foram levantados em julho de 2013. Sendo

assim, acredita-se que no ano de 2013 ainda possam existir estudos que não tenham sido recuperados pelas máquinas de busca.

□

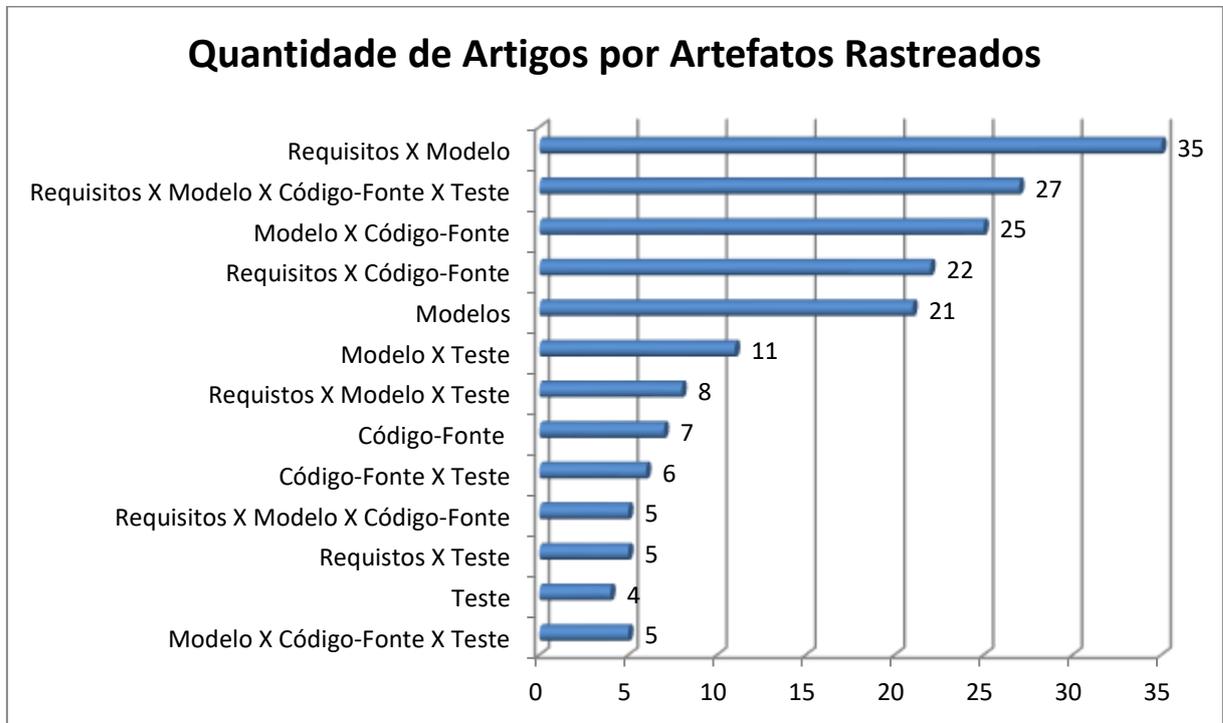


Figura 3-7. Quantidade de Artigos por Artefatos Rastreados

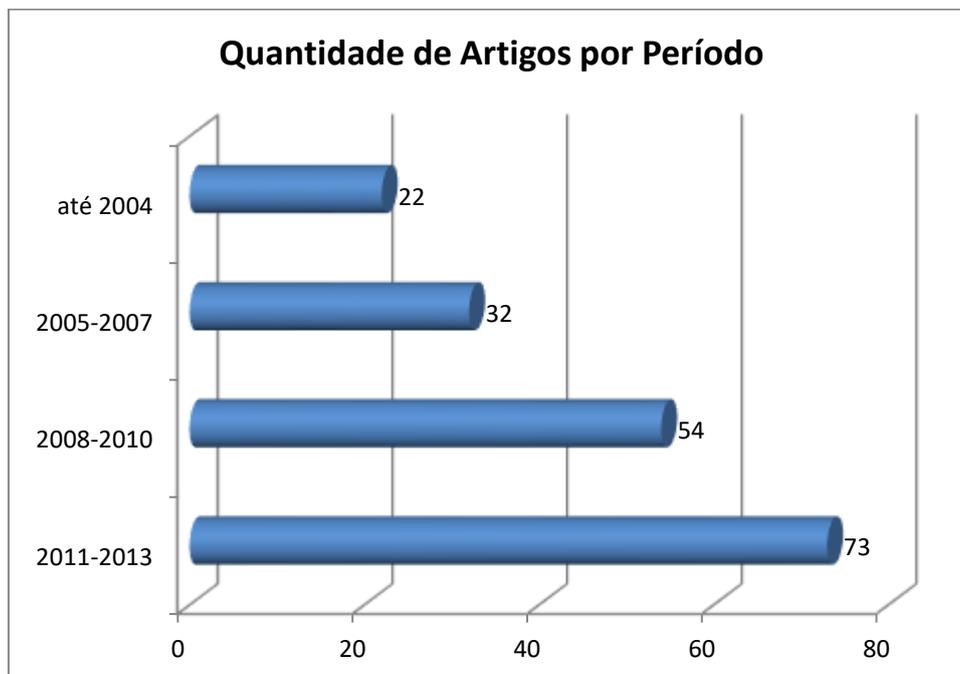


Figura 3-8. Quantidade de Artigos por período

As informações obtidas com esse mapeamento sistemático ajudaram a embasar a proposta de Tese que se apresenta, uma vez que também demonstram diversas possibilidades para a detecção da rastreabilidade de requisitos e um crescente interesse pela área.

3.3 Considerações finais

Este capítulo apresentou o mapeamento sistemático que foi planejado de acordo com o paradigma GQM. Esse mapeamento sistemático forneceu bom embasamento da literatura para permitir o desenvolvimento da proposta de tese apresentada.

A primeira parte do mapeamento (Seção 3.2.1) identificou os estudos que tratam da rastreabilidade entre os requisitos, foco desta tese. Além de serem identificados os estudos, foi feita a investigação sobre as abordagens por eles utilizadas. O levantamento permitiu conhecer todas as abordagens propostas. As mais utilizadas foram estudadas. Uma delas (Modelo do Espaço Vetorial) foi implementada e utilizada nesta tese de doutorado, conforme detalhado no próximo capítulo. Este mapeamento também identificou as ferramentas utilizadas para dar suporte ao processo. Diversos autores destacam a importância do processo de rastreabilidade ter suporte automatizado de ferramentas ([CLELAND-HUANG *et al.*, 2012] [SOMMMERVILLE, 2011], [SUNDARAM *et al.*, 2010], [ZISMAN, SPANOUDAKIS, 2004]). Também foi identificado o aumento no número de publicações na área nos últimos anos.

A segunda questão do mapeamento investigou a forma como as abordagens propostas na literatura para a determinação da rastreabilidade entre os requisitos foram avaliadas. A minoria dos trabalhos (cerca de 15%) é validada por meio de estudos experimentais controlados. A maioria desses trabalhos são apresentados com apenas um estudo de caso ou exemplo de uso, em apenas um ou dois documentos de requisitos.

Na segunda parte do mapeamento sistemático (Seção 3.2.3), foram investigadas as abordagens para determinar a rastreabilidade entre os artefatos (documento de requisitos, modelos, código-fonte e artefatos da fase de testes).

Apesar de não ser o foco deste trabalho, essas abordagens foram estudadas para que se buscassem novas alternativas para determinar a rastreabilidade entre os requisitos. Nessa parte do mapeamento também foi identificado um aumento no número de publicações na área.

Nos relatos dos artigos também fica evidenciada a necessidade de ferramentas que deem suporte à rastreabilidade, dada a dificuldade em determinar os links de rastreabilidade e de mantê-los ao longo do processo de desenvolvimento de software ([CLELAND-HUANG *et al.*, 2012] [SOMMMERVILLE, 2011], [SUNDARAM *et al.*, 2010], [ZISMAN, SPANOUDAKIS, 2004]). O fato da revisão da literatura ter sido conduzida por meio de um processo sistemático – o de mapeamento sistemático (PETERSEN *et al.*, 2008) – facilitou a atualização bibliográfica referente à pesquisa. Sendo assim, os dados do mapeamento que foram conduzidos inicialmente em outubro de 2012 foram atualizados em julho de 2013 com a replicação das *strings* de busca registradas no protocolo e uso da ferramenta StArt para atualização da pesquisa.

Capítulo 4

ABORDAGENS PARA GERAÇÃO DA RTM

Este capítulo apresenta as abordagens criadas para a geração da RTM. Inicialmente são apresentadas a RTM-E e RTM-NLP. Depois são detalhadas a RTM-Fuzzy e a RTM-N mostrando como estas combinaram as anteriores com o uso de técnicas de inteligência computacional.

4.1 Considerações Iniciais

Conforme apresentado no Capítulo 1, a proposta desta tese é:

“O uso de técnicas de inteligência computacional melhora a acurácia dos links de rastreabilidade entre requisitos”.

Sendo assim, o objetivo desta tese foi a proposição de novas abordagens para a geração da RTM, utilizando-se de técnicas de inteligência computacional para buscar melhoria nos links de rastreabilidade. A melhora na detecção deveria ser em relação à utilização isolada de abordagens, comprovando assim que a combinação de abordagens com técnicas de inteligência computacional pode melhorar a acurácia dos links de rastreabilidade. Vale destacar que, das duas abordagens iniciais, uma era baseada em processamento de linguagem natural, ou seja, com iniciativas já encontradas na literatura (RTM-NLP) e outra era totalmente original, baseada nos dados de entrada dos requisitos funcionais (RTM-E).

Assim, com base nesse objetivo, identifica-se a hipótese a ser investigada neste trabalho:

H₁: Utilizar técnicas de inteligência computacional melhora a acurácia dos links de rastreabilidade.

Com base no objetivo e na hipótese de trabalho, no Capítulo 2 apresentou-se a fundamentação teórica abrangendo engenharia de requisitos, rastreabilidade de requisitos e técnicas de inteligência computacional.

Além disso, utilizou-se o GQM para se planejar uma revisão bibliográfica conduzida por meio de mapeamento sistemático, a qual teve o intuito de fundamentar a tese proposta e avaliar pesquisas relacionadas ao trabalho que se propõe. O resultado deste mapeamento está no Capítulo 3. A partir dele, ficou caracterizada a importância da geração da RTM, bem como das abordagens utilizadas na literatura. Constatou-se também que não foram encontrados em outros trabalhos, iniciativas como as que são descritas nesta tese de doutorado.

Sendo assim, neste capítulo são descritas as quatro abordagens, frutos desta tese de doutorado. Conforme visto no Capítulo 1, as abordagens são:

- RTM-E, baseada nos dados de entrada dos requisitos funcionais;
- RTM-NLP, baseada no processamento de linguagem natural;
- RMT-Fuzzy, baseada em Sistema Fuzzy e que utiliza a RTM-E e RTM-NLP como entrada;
- RTM-N, baseada em Redes Neurais Artificiais e que utiliza a RTM-E e RTM-NLP como entrada;

Uma vez que essas abordagens foram propostas e combinadas, foi necessária a execução de estudos experimentais para confirmar a melhora na detecção de links de rastreabilidade, para poder confirmar a hipótese apresentada. Todo o trabalho de experimentação é detalhado no Capítulo 5 desta tese.

A seguir, são detalhadas as quatro abordagens. Destaca-se ainda que elas foram implementadas na ferramenta COCAR, desenvolvida para dar suporte a várias atividades do ciclo de desenvolvimento de software, baseadas nos requisitos e no modelo de casos de uso (DI THOMMAZO et al., 2007). Sua primeira versão foi desenvolvida para ambiente WEB. No contexto deste trabalho de doutorado, ela foi migrada para ambiente Desktop para permitir que fossem agregadas algumas funcionalidades de visualização e para facilitar a realização de experimentos. A ferramenta é detalhada no Apêndice A.

Nas próximas seções, são apresentadas cada uma das abordagens criadas no contexto desta tese.

4.2 Abordagem RTM-E: geração da Matriz de Rastreabilidade de Requisitos baseada nos dados de entrada

Na abordagem RTM-E, a relação de dependência entre os requisitos funcionais é determinada pelo percentual de dados de entrada coincidentes entre cada par de requisitos funcionais. A dependência é mapeada com uso do Índice de Jaccard (1996). Esse índice é utilizado para comparação de similaridade e diversidade entre conjuntos de dados e sua apresentação está na Equação 2.

$$J(A,B) = \frac{n(A \cap B)}{n(A \cup B)} \quad (2)$$

O numerador é dado pela quantidade de dados da intersecção entre os dois conjuntos (A e B), enquanto o denominador é constituído pela quantidade de dados da união entre esses dois conjuntos.

A abordagem RTM-E define a dependência entre dois requisitos funcionais da seguinte forma: considerando-se RFa o conjunto de dados de entrada de um requisito funcional A e RFb o conjunto de dados de entrada do requisito funcional B, a dependência entre eles é dada pela Equação 3:

$$J(RFa,RFb) = \frac{n(RFa \cap RFb)}{n(RFa \cup RFb)} \quad (3)$$

Assim, de acordo com a abordagem RTM-E, cada posição (i,j) da matriz de rastreabilidade RTM(i,j) corresponde ao valor da Equação 4:

$$RTM(i,j) = J(RF-i, RF-j) \quad (4)$$

As posições da diagonal principal da RTM não são calculadas, uma vez que indicam a dependência entre os RFs com eles mesmos. Além disso, a matriz gerada é simétrica, ou seja, RTM(i,j) é igual a RTM(j,i).

A implementação dessa abordagem na COCAR foi possível porque a forma de armazenamento dos requisitos na ferramenta separa atômicamente os dados de entrada de cada requisito, seguindo o *template* definido por Kawai (2005). Sendo assim, cada vez que um dado de entrada é inserido em um requisito funcional na ferramenta, ele fica disponível para ser vinculado em outros requisitos funcionais, evitando possível ambiguidade dos dados. Vale destacar que, na literatura, não foram encontradas iniciativas que fizessem uso de dados de entrada de requisitos funcionais para determinação automática da RTM. A RTM gerada a partir desta abordagem foi denominada RTMe.

Quando a dependência entre dois requisitos funcionais é calculada com a Equação 3, o resultado é um valor entre 0 e 100%. A partir deste valor foi preciso determinar se existia ou não relação de dependência entre os requisitos funcionais e qual o seu nível, caso ela existisse. Inicialmente foi feito um piloto com três sistemas, e determinou-se que, se o valor fosse maior ou igual a 50%, a dependência seria forte. Se o valor estivesse entre zero (maior que zero) e menor que 50%, a dependência seria fraca. Caso o valor fosse 0%, ou seja, caso não existisse nenhum dado de entrada comum aos requisitos funcionais, a relação de dependência não existiria. A partir desses níveis de dependência, foram realizados experimentos, como detalhado em DI THOMMAZO (2012). Com o intuito de melhorar a acurácia nos links de rastreabilidade, foi feito um estudo para ajustar a definição desses intervalos. Para isso, os dados dos experimentos que avaliaram a eficácia da abordagem foram submetidos a uma árvore de decisão, com o intuito de encontrar os níveis de dependência que melhor representassem a rastreabilidade. Esse processo é representado na Figura 4-1 na qual é possível observar todos os valores gerados pela abordagem RTM-E com os respectivos oráculos, ou seja, o resultado que os especialistas determinaram como “correto” para cada relação entre requisitos funcionais. Esses valores foram apresentados como entrada para uma árvore de decisão que extraiu as melhores regras que representassem a classificação da rastreabilidade. Utilizou-se a árvore de decisão disponível na ferramenta WEKA (<http://www.cs.waikato.ac.nz/ml/weka/>) para esse processamento. Sendo assim, os novos intervalos definidos foram:

- Sem dependência: 0% até 18%;

- Dependência Fraca: 18% até 57%;
- Dependência Forte: 57% até 100%.

Esse processo também foi detalhado e apresentado em Di Thommazo (2014) e o impacto desta modificação é detalhado na Seção 5.3.3.

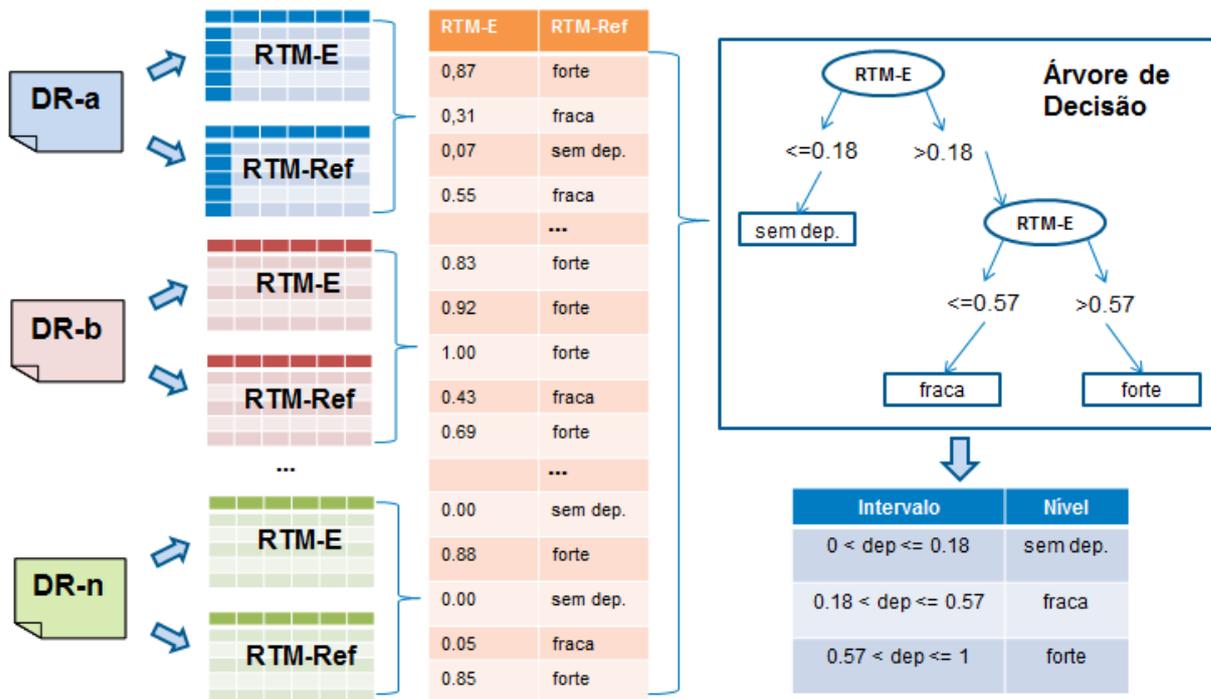


Figura 4-1. Definição dos intervalos de dependência com uso de árvore de decisão

4.3 Abordagem RTM-NLP: geração da Matriz de Rastreabilidade de Requisitos baseada em Processamento de Linguagem Natural

Muito embora existam iniciativas que fazem uso de processamento de linguagem natural para determinação da rastreabilidade no processo de desenvolvimento de software, a minoria dos trabalhos trata da rastreabilidade dentro de um mesmo artefato (GOKNIL *et al.*, 2011). Isso também foi mostrado nos resultados do Mapeamento Sistemático conduzido. Apesar de a minoria dos trabalhos

explorar esse tipo de rastreabilidade, os benefícios do estabelecimento desta rastreabilidade já foram destacados na motivação desta tese de doutorado.

De acordo com Deeptimahanti e Sanyal (2011), o uso de processamento de linguagem natural na engenharia de requisitos não tem como objetivo a compreensão do texto em si, mas tem o intuito de extrair conceitos contidos do Documento de Requisitos. Sendo assim, essa abordagem proposta utiliza os conceitos extraídos dos requisitos funcionais por meio de processamento de linguagem natural para se determinar a rastreabilidade entre os requisitos funcionais.

Inicialmente, o processamento de linguagem natural foi aplicado somente no campo que descreve o processamento (ações) do requisito funcional. Depois da avaliação do estudo experimental (DI THOMMAZO et al., 2012.), percebeu-se que muitas vezes, como existe no *template* um campo para os dados de entrada (como já destacado na proposta RTM-E), os analistas não registravam novamente esses dados de entrada no campo de processamento, reduzindo o valor da similaridade. Desta forma, essa abordagem foi melhorada, processando-se todos os campos de texto contidos no *template*. Então, o *template* faz com que o engenheiro de requisitos levante todos os dados solicitados e todas essas informações são utilizadas no algoritmo que faz o cálculo de similaridade. Conforme estudos preliminares, essa modificação impactou em melhoria na eficácia da abordagem.

Para determinar o nível de dependência entre os conteúdos dos campos de processamento de dois requisitos funcionais, utiliza-se o método do Vetor de Frequência e a Similaridade do Cosseno (SALTON, ALLAN, 1994). Esse método retorna o percentual de similaridade entre dois trechos de texto.

Antes da aplicação do Vetor de Frequência e Similaridade do Cosseno, é feito um pré-processamento do texto, com o intuito de aumentar a eficiência do método. A importância do uso de pré-processamento já havia sido destacada na Seção 2.3.2. Sendo assim, inicialmente é retirado do texto o conjunto de palavras que podem ser consideradas irrelevantes, como, por exemplo, artigos, pronomes, preposições e conjunções, denominadas *stopwords* (Figura 4-2-A). Depois, é aplicado o processo de *stemming* (Figura 4-2-B), que reduz as palavras a seus radicais, fazendo com que elas tenham o mesmo peso para determinação de similaridade entre os textos. Após essas duas etapas, o método determina a similaridade entre os textos de dois

requisitos funcionais (Figura 4-2-C), identificando aqueles que, segundo a técnica, são similares. Os passos para a determinação da similaridade foram descritos em (Di Thommazo et al, 2012) e são reproduzidos na sequência.

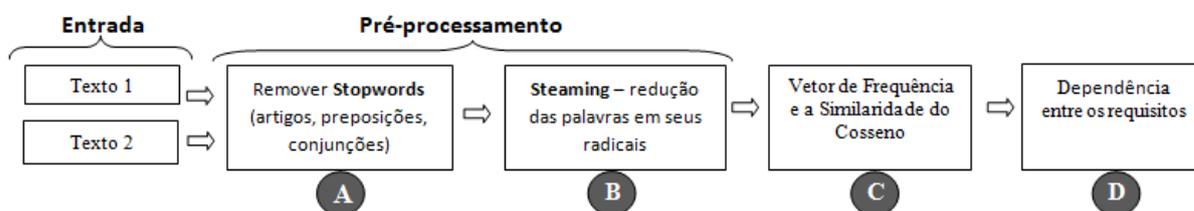


Figura 4-2 Passos da abordagem RTM-NLP

O primeiro passo para a aplicação do Vetor de Frequência e Similaridade do Cosseno é a representação de cada sentença em um vetor. Nesse vetor, cada posição é preenchida por uma palavra da sentença. A medida do cosseno entre eles irá representar a similaridade. Considerando dois requisitos funcionais RF1 e RF2, descritos, respectivamente, pelas sentenças S1 e S2. O cálculo da similaridade é feito da seguinte forma:

- 1- Representação de S1 no vetor x e S2 no vetor y. Cada palavra irá ocupar uma posição em cada vetor. Se S1 tiver p palavras, o vetor x também possuirá p posições inicialmente. Da mesma forma, se S2 possuir q palavras, o vetor y também possuirá q posições.
- 2- No vetor não deve ocorrer repetição de palavras. Sendo assim, as ocorrências de cada palavra devem ser contadas para determinar a sua frequência no vetor. Feito isso, as repetições são retiradas, deixando-se no vetor apenas uma ocorrência por palavra. Além da palavra, também é colocada, em cada posição do vetor, a frequência com que ela apareceu em cada sentença.
- 3- Reordenação dos vetores em ordem alfabética.
- 4- Cruzamento de cada termo dos vetores, buscando correspondentes no outro vetor. Quando não houver a palavra no outro vetor, deve ser inserida uma posição equivalente no vetor que não a possui, com frequência igual a zero. Sendo assim, cada palavra que exista em S2 e que não exista em S1 dará origem a uma posição vazia no vetor x, com frequência zero. Dessa forma, após a aplicação desse passo, surgirão novas posições em ambos os vetores,

representando os valores que não tem correspondente com o outro vetor. Ao final dessa etapa, ambos os vetores possuirão o mesmo número de posições.

- 5- Com os vetores ajustados, deve ser aplicada a Equação 5 de similaridade entre os vetores x e y – $\text{sim}(x,y)$ – considerando n o número de posições dos vetores.

$$\text{sim}(x,y) = \frac{\sum_{i=1}^n (x_i \cdot y_i)}{\sqrt{\sum_{i=1}^n (x_i^2) \cdot \sum_{i=1}^n (y_i^2)}} \quad (5)$$

Conforme mencionado anteriormente, embora existam, na literatura, iniciativas que fazem uso de Modelo do Espaço Vetorial, mais especificamente do Vetor de Frequência e Similaridade do Cosseno, diferentemente desses trabalhos, nesta tese de doutorado, existe a determinação do nível de dependência (fraco ou forte). A importância dessa separação em níveis é explorada no Capítulo 6. A RTM gerada a partir desta abordagem foi denominada RTMnlp. Assim como no caso da RTM-E, inicialmente os intervalos que classificariam o nível de dependência foram determinados com base em um piloto com 3 sistemas, conforme detalhado em Di Thommazo e outros (2012). Os valores eram:

- Sem dependência: 0% até 40%;
- Dependência Fraca: 40% até 70%;
- Dependência Forte: 70% até 100%.

Assim como para a RTM-E, os dados dos experimentos (DI THOMMAZO et al., 2012) foram processados por uma árvore de decisão para encontrar os melhores intervalos para RTM-NLP. Esse processo foi apresentado em Di Thommazo e outros (2014). Depois de processados os dados pela árvore de decisão, os novos valores de intervalo foram:

- Sem dependência: 0% até 37%;
- Dependência Fraca: 37% até 63%;
- Dependência Forte: 63% até 100%.

O impacto desta modificação é detalhado na Seção 5.3.3.

4.4 Abordagem RTM-Fuzzy: geração da Matriz de Rastreabilidade de Requisitos baseada Sistemas Fuzzy

O propósito dessa abordagem é combinar as duas anteriores através de um Sistema de Inferência Fuzzy, considerando os dois aspectos explorados anteriormente – relacionamento entre os dados manipulados pelos requisitos funcionais (RTM-E) e o texto que compõe esses requisitos funcionais (RTM-NLP) – para criar a RTM.

Nas abordagens apresentadas anteriormente, a classificação da dependência entre dois requisitos funcionais como “sem dependência”, “dependência fraca” ou “dependência forte” é determinada de acordo com o valor gerado pela abordagem em relação às faixas de intervalos criadas. Um dos problemas em analisar os dados dessa maneira é que a diferença entre a classificação em uma faixa ou outra pode ser muito sutil. Se considerarmos que o limite entre um nível e outro seja de 37% na abordagem RTM-NLP e a abordagem gerar um valor de 36.5% para a dependência entre dois requisitos funcionais, isso não indicaria nenhuma dependência na RTM, ou seja, não existiria um link de rastreabilidade entre os requisitos funcionais. Se o valor gerado fosse 37.5%, então já haveria dependência fraca. Utilizando lógica fuzzy, esse problema é minimizado, pois é possível trabalhar com uma faixa nebulosa nesses intervalos, através da função de pertinência.

Conforme visto anteriormente, esse processo de conversão dos valores absolutos em sua representação fuzzy é chamado Fuzzificação, dando origem às funções de pertinência. Nas funções de pertinência, o eixo X representa o percentual de dependência entre os requisitos funcionais (de 0 a 100%) e o eixo Y, representa, por sua vez, o nível de pertinência, isto é, a probabilidade de pertencer ao conjunto fuzzy (“sem dependência”, “dependência fraca” ou “dependência forte”), podendo variar entre 0 e 1.

A Figura 4-3 representa o sistema fuzzy adotado, o qual possui como entrada as matrizes RTMe e RTMnlp. As Figuras 4-4 e 4-5 representam, respectivamente, a função de pertinência das abordagens RTM-E e RTM-NLP. Nessas figuras, o eixo X indica o percentual de dependência calculado por cada abordagem. O eixo Y indica o

grau de pertinência e varia entre 0 e 1. Quanto maior o valor de pertinência, maior a chance de a dependência estar em um dos possíveis conjuntos (“sem dependência”, “dependência fraca” ou “dependência forte”). Existem faixas de valores onde a pertinência pode ser alta para um conjunto e baixa para outro (por exemplo, a faixa com percentual de dependência entre 35% e 55% na Figura 4-4).

A Figura 4-6 indica as regras criadas para o Sistema Fuzzy. Essas regras são utilizadas para o cálculo do valor de saída, ou seja, para a determinação da RTMfuzzy.

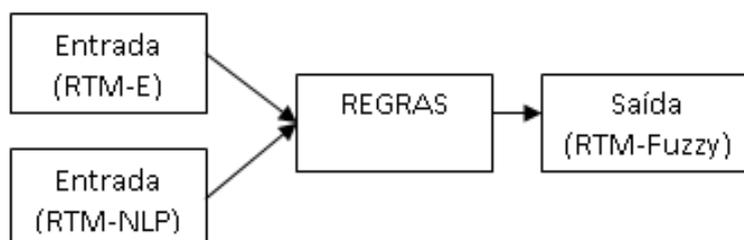


Figura 4-3. Sistema Fuzzy

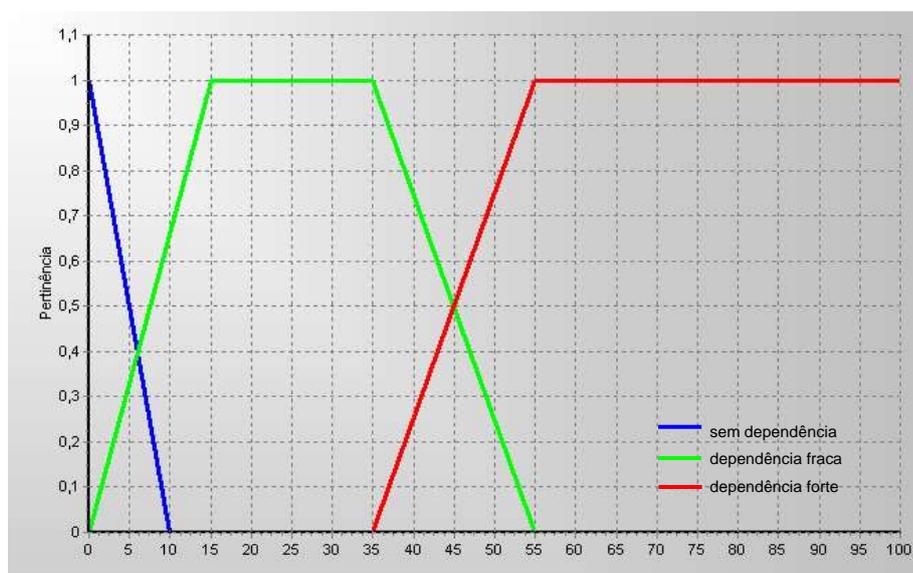


Figura 4-4. Função de Pertinência RTM-E

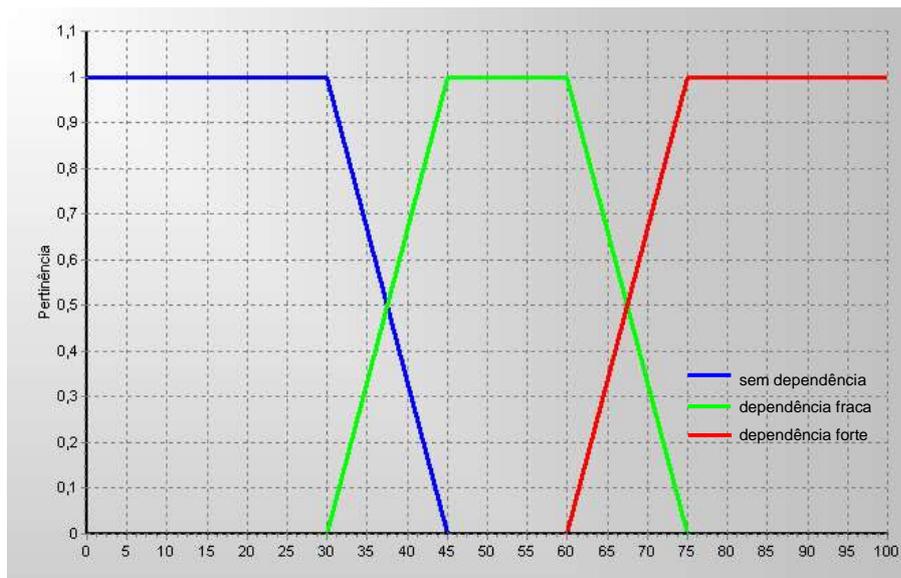


Figura 4-5. Função de Pertinência RTM-NLP

	Antecedente		Consequente
Se	RTM-E = "sem dependência" E RTM-NLP= "sem dependência"	então	"sem dependência"
Se	RTM-E = "dependência fraca" E RTM-NLP= "dependência fraca"	então	"dependência fraca"
Se	RTM-E = "sem dependência" E RTM-NLP= "dependência forte"	então	"dependência fraca"
Se	RTM-E = "dependência forte" E RTM-NLP= "dependência forte"	então	"dependência forte"
Se	RTM-E = "sem dependência" E RTM-NLP= "dependência fraca"	então	"sem dependência"
Se	RTM-E = "dependência fraca" E RTM-NLP= "sem dependência"	então	"dependência fraca"
Se	RTM-E = "sem dependência" E RTM-NLP= "dependência forte"	então	"dependência fraca"
Se	RTM-E = "dependência forte" E RTM-NLP= "dependência fraca"	então	"dependência forte"
Se	RTM-E = "dependência forte" E RTM-NLP= "sem dependência"	então	"dependência fraca"

Figura 4-6 Regras utilizadas no Sistema Fuzzy

A Figura 4-7 mostra a função de pertinência de saída. Da mesma forma que as Figuras 4-4 e 4-5, o eixo X indica o percentual de dependência da RTMfuzzy. O eixo Y indica o grau de pertinência e varia de 0 a 1.

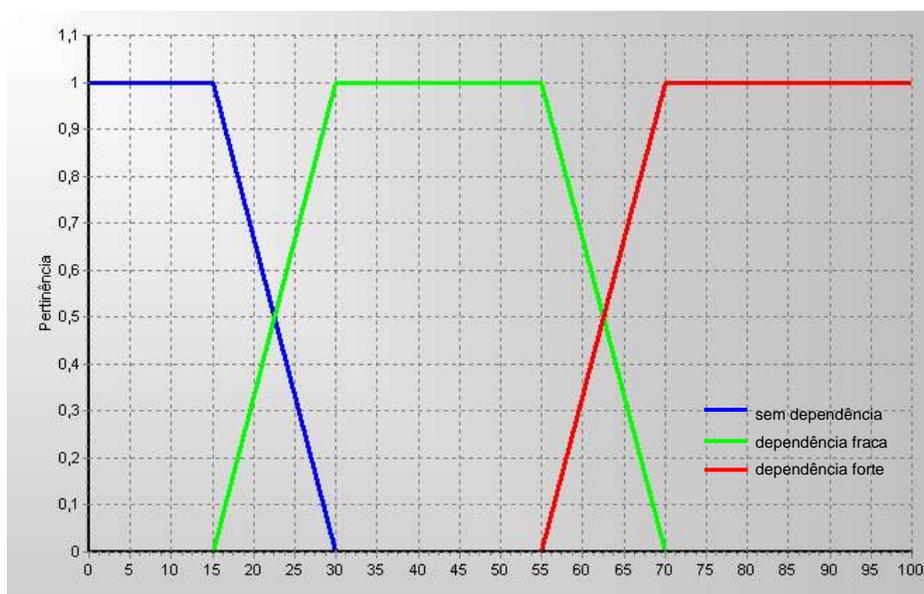


Figura 4-7. Função de Pertinência RTM-Fuzzy para a saída do Sistema Fuzzy

Essa abordagem foi avaliada por meio de estudos experimentais que são detalhados na Seção 5.3.3 e que foram apresentados em Di Thommazo e outros (2013a). Inicialmente, todas as funções de pertinência foram definidas pelo autor desta tese a partir da análise dos intervalos de cada uma das abordagens. No entanto, para melhorar a acurácia dos links de rastreabilidade foram utilizados algoritmos genéticos para redefinir as funções de pertinência. Conforme detalhado no capítulo anterior, os algoritmos genéticos são inspirados na evolução natural e utilizam conceitos como seleção, mutação e cruzamentos (*crossover*). Outro conceito importante são os cromossomos. Eles correspondem ao nosso “problema”. O primeiro passo da aplicação do algoritmo genético foi representar as características do problema, ou seja, de cada função de pertinência, como um cromossomo.

A Figura 4-8 representa a modelagem do problema. Cada função de pertinência da RTM-E, da RTM-NLP e da Função de Pertinência de Saída foi representada no cromossomo. Os limites de cada trecho são representados no cromossomo. Como exemplo, na Figura 4-8 foram destacados os limites da função de “dependência fraca”

da RTM-E dando origem a trechos do cromossomo destacados em verde. Após a criação deste cromossomo, o próximo passo é a geração da primeira população, composta por outros 99 cromossomos com trechos gerados randomicamente. Esses passos são representados na Figura 4-9. A próxima etapa é a fase de Seleção, onde os indivíduos que permanecerão na próxima geração vão ser selecionados. Foi utilizado o algoritmo da Roleta, selecionando 40% da população original, baseado na função de ajuste (*fitness function*). Esta função avalia a eficácia de cada cromossomo em resolver a detecção da rastreabilidade. Para isso, foram utilizados dados de um estudo experimental com 18 sistemas (DI THOMMAZO *et al.*, 2012) em que valores da RTMe, RTMnlp e a RTM de referência eram conhecidos. A função de ajuste avaliava o quão próximo cada cromossomo estava da “melhor solução”. Após selecionar os indivíduos (cromossomos) que deveriam permanecer vivos na próxima geração, eram aplicados os operadores de mutação e de cruzamento. Quando um operador de mutação era aplicado, uma posição randômica de um cromossomo era modificada com um novo valor, também gerado randomicamente.

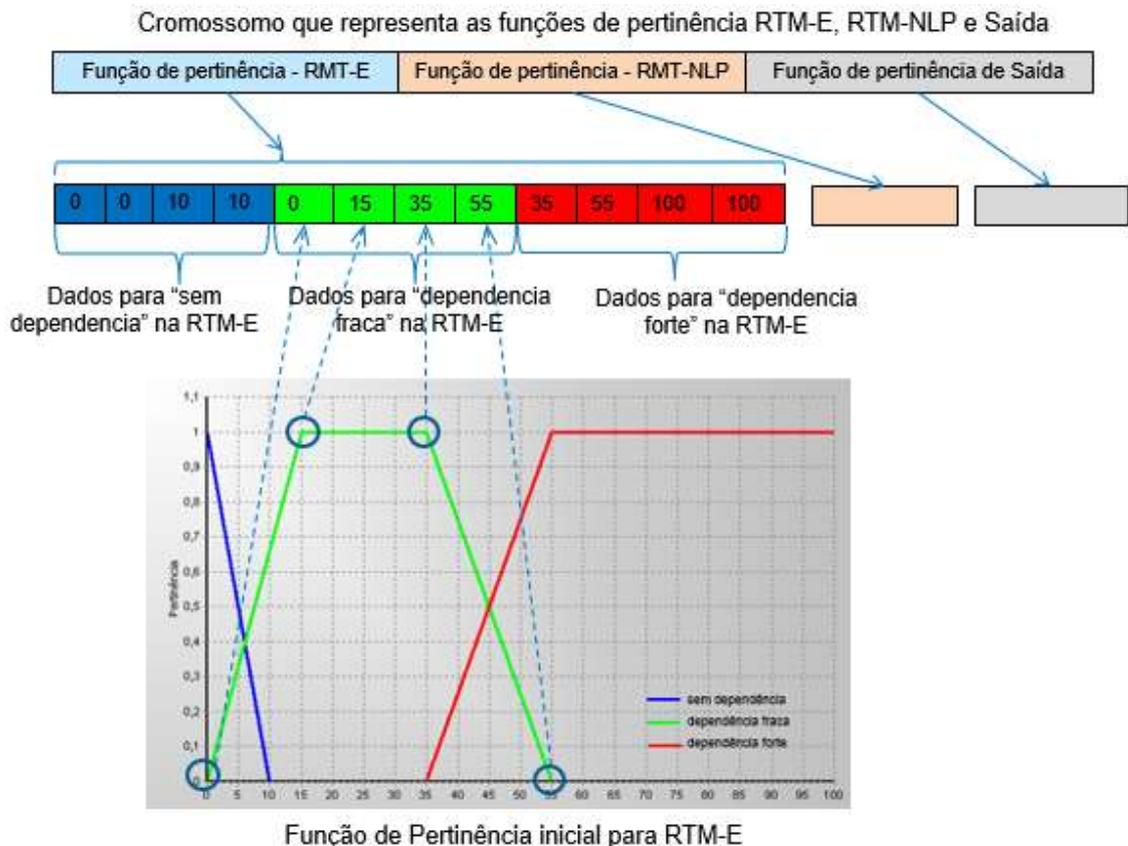


Figura 4-8. Criação do cromossomo utilizado no algoritmo genético

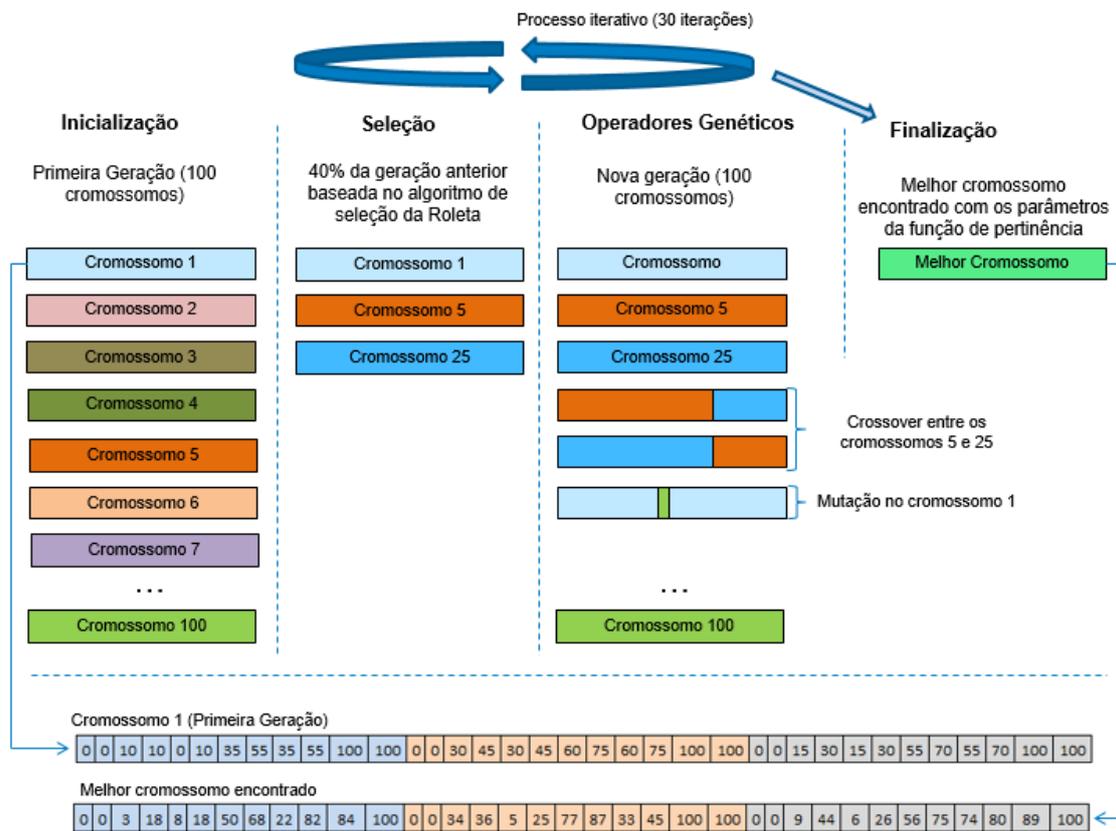


Figura 4-9. Passos do algoritmo genético

Na nova população 5% dos indivíduos eram gerados a partir do operador de mutação. Os outros indivíduos eram gerados utilizando-se o operador de cruzamento (*crossover*). Aplicar o operador de cruzamento implicava em selecionar aleatoriamente dois cromossomos para serem cruzados, gerando novos indivíduos, conforme mostra a Figura 4-9. Foram feitos cruzamentos até que se completassem os 100 indivíduos desta geração. Quando a geração estava completa, então novamente era aplicado a função de ajuste, em um processo iterativo. Foi estabelecido como critério de parada a determinação de um indivíduo (cromossomo) que encontrasse com precisão todos os links de rastreabilidade ou o máximo de 30 iterações. Ao final das 30 iterações foi escolhido o melhor indivíduo. A Figura 4-10 exemplifica a mudança das funções de pertinência para a RTM-E, comparando como era no início e depois de utilizados os dados determinados pelo algoritmo genético.

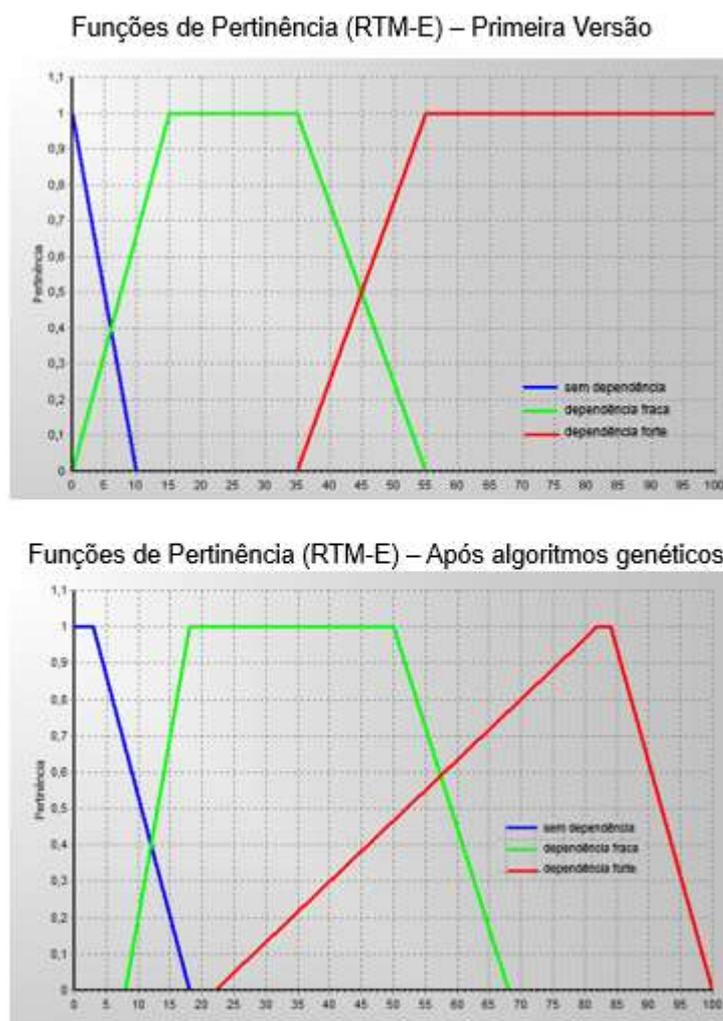


Figura 4-10. Comparação das Funções de Pertinência antes e depois do uso dos algoritmos genéticos

4.5 Abordagem RTM-N: geração da Matriz de Rastreabilidade de Requisitos baseada em Redes Neurais Artificiais

O objetivo desta abordagem é detectar a rastreabilidade combinando a RTM-E e a RTM-NLP por meio de redes neurais artificiais. Uma rede neural *Multilayer Perceptron* (MLP) foi utilizada no desenvolvimento desta abordagem. A MLP tem conectividade *feed-forward*, ou seja, a saída de cada neurônio se conecta com todos os neurônios da próxima camada. O sinal se propaga na rede de forma progressiva.

As entradas da rede neural foram os valores de dependência determinados pelas abordagens RTM-E e RTM-NLP (valores entre 0 e 1). As saídas da rede neural foram “sem dependência”, “dependência fraca” e “dependência forte”, conforme detalhado na topologia da rede neural da Figura 4-11.

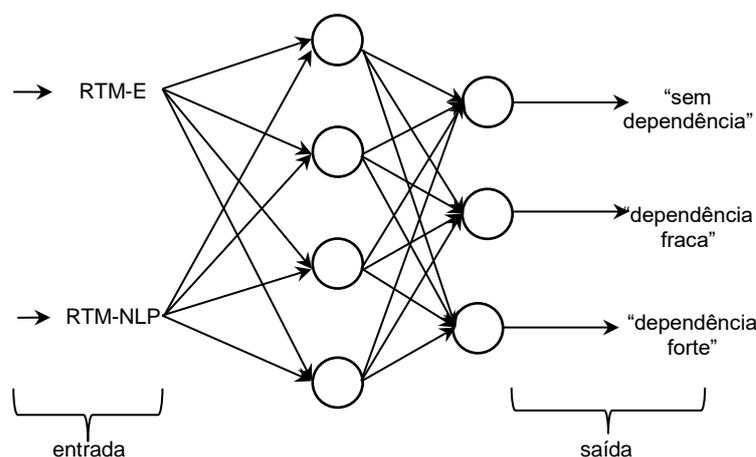


Figura 4-11 Topologia da rede neural utilizada na RTM-N

Para treinar uma rede neural, é necessário “ensinar” a rede, e isso se faz fornecendo padrões corretos a ela. Sendo assim, foram utilizados dados do estudo experimental conduzido com as abordagens RTM-E e RTM-NLP (DI THOMMAZO et al., 2012). Esse estudo experimental é detalhado na Seção 5.3.3, porém adianta-se que ele foi conduzido a partir da análise de 18 sistemas. Foram construídas as RTMs de referência para cada um desses sistemas baseando-se na análise detalhada de cada par de requisitos funcionais. Sendo assim, para treinamento da rede neural, utilizou-se como entrada os valores gerados por cada abordagem (RTM-E e RTM-NLP) e como saída o valor da RTM de referência construída para cada sistema. A Figura 4-12 detalha como foram processados os dados. Por meio das abordagens RTM-E e RTM-NLP foram geradas respectivamente a RTMe e a RTMnlp de cada Documento de Requisitos (DR). Também foi criada a Matriz de Referência de cada sistema (RTMref). Os dados de cada par de requisitos foram colocados em uma tabela para serem apresentados na rede neural. Esses dados foram os padrões para o treinamento da rede. Cada linha da tabela foi utilizada como um dado para treinamento. A rede neural foi ajustando os pesos de ligação entre os neurônios para que se tornasse apta a reconhecer os padrões que seriam identificados (“sem dependência”, “dependência fraca”, “dependência forte”).

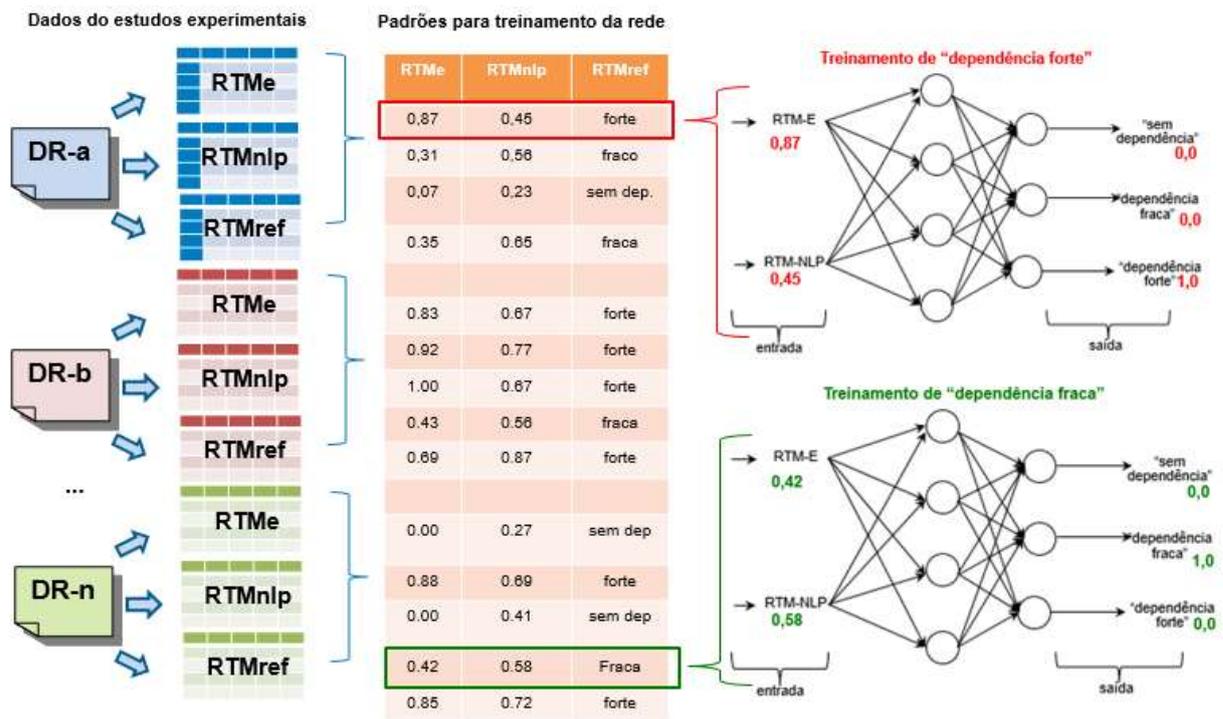


Figura 4-12. Processo de treinamento da rede neural para a abordagem RTM-N

Uma vez que a rede neural foi treinada, tornou-se apta a classificar a rastreabilidade em novas relações entre requisitos funcionais em novos sistemas. A classificação que a rede neural faz está de acordo com o treinamento que recebeu, ou seja, ela classifica a relação entre dois requisitos funcionais como “sem dependência”, “dependência fraca” ou “dependência forte”. A RTM gerada pela abordagem RTM-N foi denominada RTMn. A abordagem foi avaliada por meio de um estudo experimental que será detalhado na Seção 5.3.3 e que foi apresentado em Di Thommazo e outros (2013b). Destaca-se ainda que para avaliar essa proposta inicialmente os dados foram processados na ferramenta WEKA. Uma vez validada a abordagem a implementação foi incorporada na ferramenta COCAR.

4.6 Considerações Finais

Neste capítulo, foram apresentadas as quatro abordagens criadas pelo autor para gerar a RTM. Foi apresentada a RTM-E que identifica a dependência entre os requisitos funcionais baseada na quantidade de dados de entrada comum entre os requisitos funcionais. Além de apresentar esta abordagem, também foi mostrado como os intervalos que definem os níveis de dependência entre os requisitos funcionais foram refinados, utilizando-se de uma árvore de decisão com os dados de um estudo experimental publicado. Também foi apresentada a RTM-NLP que identifica a dependência entre os requisitos funcionais a partir de processamento de linguagem natural, mais especificamente com o uso do Modelo do Espaço Vetorial.

Com o intuito de se mostrar que essas abordagens combinadas com inteligência computacional podem contribuir para a melhoria da acurácia na detecção dos links de rastreabilidade, foi apresentada também a abordagem RTM-Fuzzy que combina as anteriores com um sistema fuzzy. Também foi mostrado como as funções de pertinência do sistema fuzzy foram refinadas com o uso de algoritmos genéticos. Além dessa abordagem, também foi apresentada a abordagem RTM-N, que faz uso de redes neurais para determinar a RTM. Vale destacar que todas as abordagens estão implementadas na ferramenta COCAR.

Destaca-se ainda que todas as abordagens foram avaliadas por meio de estudos experimentais, conforme comentado no decorrer deste capítulo e detalhado no Capítulo 5.

Capítulo 5

ESTUDOS EXPERIMENTAIS

Este capítulo apresenta os estudos experimentais que foram conduzidos para avaliar as abordagens propostas. Os estudos estão apresentados na ordem cronológica, conforme foram executados. Dois estudos foram feitos com sistemas reais de indústrias enquanto que três estudos são experimentos controlados no contexto acadêmico. Os resultados dos estudos corroboram a tese proposta.

5.1 Considerações Iniciais

Conforme descrito no Capítulo 1 e detalhado no Capítulo 4, esta tese tem como proposta melhorar a acurácia dos links de rastreabilidade com o uso de técnicas de inteligência computacional. Para que pudesse ser possível afirmar que houve “melhora” na detecção foi fundamental medir a acurácia das abordagens inicialmente e, depois de utilizar técnicas de inteligência computacional, fazer uma nova medição e avaliar os resultados. Como já mencionado nos capítulos anteriores, o processo para avaliar as abordagens propostas nesta tese foi conduzido por meio de um processo de experimentação, cujos resultados foram apresentados em eventos relevantes da área.

Travassos, Gurov e Amaral (2002) afirmam que os experimentos são apropriados para confirmar as teorias e o conhecimento convencional, explorar os relacionamentos, avaliar a predição dos modelos ou validar medidas. Para Basili e outros (1996), novos métodos, técnicas, linguagens e ferramentas para engenharia de software não devem ser apresentados sem antes passar por um processo de experimentação. Sendo assim, as novas propostas e sugestões precisam ser

comparadas com as existentes a fim de evidenciar suas reais contribuições para a área.

Neste capítulo, são apresentados os estudos experimentais que foram realizados com as abordagens propostas. As comparações que são feitas nos permitem afirmar que as abordagens criadas com técnicas de inteligência computacional (RTM-Fuzzy e RTM-N) apresentam melhor acurácia que as duas primeiras abordagens (RTM-E e RTM-NLP). Outra análise também foi feita: comparou-se a efetividade das abordagens propostas em relação à efetividade das mesmas abordagens que foram refinadas com uso de técnicas de inteligência computacional. Nesse caso, as quatro abordagens apresentaram melhora na acurácia dos links de rastreabilidade, também sugerindo que a tese pudesse ser corroborada.

Dentre os diferentes tipos de estudos experimentais destacados por Wohlin e outros (2000), fez-se uso de experimentos controlados e de estudos de caso.

O experimento controlado oferece o maior nível de controle. Um experimento controlado é um evento planejado e fortemente controlado pelo pesquisador, tanto em relação ao processo quanto às variáveis envolvidas. Assim, uma ou algumas variáveis são manipuladas, enquanto as outras são mantidas fixas e, ao final, o resultado é medido (WOHLIN et al, 2000).

O estudo de caso, por sua vez, é utilizado para monitorar projetos, atividades e atribuições (Wohlin et al, 2000). De acordo com Travassos, Gurov e Amaral (2002), o objetivo desse tipo de estudo é observar um atributo específico (um caso particular) e, posteriormente, estabelecer relacionamentos entre atributos diferentes.

O processo de proposição das abordagens desta tese seguiu a metodologia apresentada no Capítulo 1 na qual, depois de feita a proposição da abordagem, esta era submetida a um experimento para avaliação e, posteriormente, um estudo de possíveis melhorias. Os experimentos e estudos de caso são apresentados neste capítulo de forma cronológica.

São apresentados três experimentos controlados (Experimento 1, Experimento 2 e Experimento 3) e dois estudos de caso de sistemas reais (Estudo de Caso 1 e Estudo de Caso 2). O primeiro experimento controlado foi conduzido para avaliar as abordagens RTM-E e RTM-NLP. O segundo experimento avaliou as abordagens RTM-Fuzzy e RTM-N, além de reavaliar as abordagens RTM-E e RTM-NLP. Com

base nessas avaliações, foram feitas melhorias nas abordagens e conduzido o terceiro experimento controlado.

Os dois estudos de casos reais foram conduzidos com o intuito de verificar como as abordagens se comportavam na detecção da rastreabilidade na indústria. Kitamura (2013) destaca que faltam estudos da rastreabilidade em sistemas reais. O primeiro estudo de caso foi aplicado no contexto de uma empresa de aviação, no sistema que gerencia o estoque de produtos que abastecem as aeronaves. Esse estudo de caso também foi utilizado para exemplificar como as abordagens processam os dados dos requisitos funcionais para determinação da rastreabilidade. Essa exemplificação está contida nos artigos publicados e é colocada neste capítulo para facilitar a compreensão de cada abordagem. O segundo estudo de caso foi aplicado em um sistema de software livre construído pela fundação Telefônica e faz a gestão de informações referentes a projetos sociais, convênio médico, dados escolares e condições de moradia de crianças e adolescentes.

A estrutura deste capítulo foi definida da seguinte forma: para auxiliar na compreensão das abordagens propostas nesta tese de doutorado, inicialmente é apresentado o primeiro estudo de caso (Seção 5.2). Nele, foi detalhada como as abordagens “processaram” os requisitos funcionais para determinar o nível de dependência entre cada par de requisitos, criando assim a RTM. Após a apresentação desse Estudo de Caso são descritos os três estudos experimentais controlados que foram conduzidos para avaliar as abordagens (Seção 5.3). Por fim, apresenta-se o segundo Estudo de Caso (Seção 5.4). As considerações finais estão na Seção 5.5.

5.2 Estudo de Caso 1

Conforme mencionado anteriormente, neste estudo de caso é mostrado um exemplo de como algumas dependências nas relações de requisitos funcionais foram determinadas por cada abordagem proposta. Espera-se que esta sequência de exemplos facilite a compreensão das abordagens. Essa descrição foi apresentada em Di Thommazo e outros (2012) e Di Thommazo e outros (2013). Na ocasião as

abordagens não haviam passado pelas melhorias descritas no Experimento 3 descrito na Seção 5.2.3.

Contexto: Esse estudo de caso foi conduzido em uma empresa de aviação que possui diversas aeronaves e realiza voos entre as principais cidades da Europa e Ásia. O software objeto de estudo informatiza toda a parte de estoque da empresa: como ela possui bases em várias cidades, o sistema gerencia vários estoques e os produtos que são inseridos, retirados ou realocados em seus diversos estoques. Apesar da empresa ter sede em Londres, a companhia possui uma equipe de análise e desenvolvimento no Brasil, local onde foi conduzido o estudo de caso. Destaca-se ainda que, quando o estudo de caso foi conduzido (2012) somente as abordagens RTM-E e RTM-NLP haviam sido propostas. Apesar disso, depois que as abordagens RTM-Fuzzy e RTM-N foram propostas, gerou-se a rastreabilidade a partir delas para observar como elas se comportavam em um sistema real.

Objetivo: Verificar como as abordagens propostas se comportavam na detecção de links de rastreabilidade na indústria.

Participantes: gerente do projeto, analisa, dois *stakeholders* e um membro da equipe de desenvolvimento

Artefatos:

- Documento de Requisitos:

- elaborado pelos analistas da empresa e validado pelos *stakeholders*;
- possuía 14 Requisitos Funcionais;
- inserido na ferramenta COCAR.

- RTM-Ref:

- criada a partir do documento de requisitos registrado na ferramenta COCAR.
- elaborada pelo autor desta tese com base na leitura e análise detalhada de cada par de requisitos funcionais e ajuda direta do gerente do projeto e um dos *stakeholders*.

Métrica: a métrica utilizada foi a eficácia das abordagens no que diz respeito à coincidência das dependências encontradas por cada abordagem em relação à RTM-Ref. A eficácia é dada pela relação entre a quantidade de relações encontradas corretamente em cada abordagem pela quantidade total de possíveis relacionamentos existentes entre os requisitos funcionais. Considerando um sistema com n requisitos funcionais, a quantidade total de possíveis relacionamentos (T) é dada pela Equação 6:

$$T = \frac{n(1 - n)}{2} \quad (6)$$

Sendo assim, a eficácia é dada pela Equação 7:

$$\text{Eficácia} = \frac{\text{Quantidade de relacionamentos encontrados corretamente}}{T} \quad (7)$$

Conforme dito anteriormente, com o intuito de facilitar a compreensão das abordagens propostas, são apresentados na sequência alguns exemplos de como cada abordagem determinou alguns links de rastreabilidade.

RTM-E

Conforme detalhado no capítulo anterior, a RTM-E é baseada nos dados de entrada dos requisitos funcionais. Para exemplificar como é feito o cálculo da dependência, detalha-se o estabelecimento da dependência entre dois requisitos funcionais. Na Figura 5.1, a célula da matriz destacada pelo retângulo sinaliza a dependência entre o requisito funcional 3 (RF3) e o requisito funcional 5 (RF5). No caso, o RF3 está relacionado com a entrada ou saída de produtos de um determinado estoque (armazém) da empresa, e o RF5 refere-se à transferência de um item de um estoque para outro. Como esses dois requisitos funcionais tratam de itens de estoque, é natural que exista relacionamento entre eles.

O RF3 possuía os seguintes dados de entrada: Contato, Data da Transação, Armazém, Quantidade, Preço unitário e Usuário. O RF5 por sua vez tinha como dados de entrada: Contato, Data da Transação, Armazém, Quantidade, Preço unitário,

Usuário, Armazém de origem, Armazém destino e Status. Como a quantidade de elementos do conjunto interseção de RF3 e RF5 ($RF3 \cap RF5$) era 6 e a quantidade de elementos do conjunto união dos mesmos requisitos funcionais ($RF3 \cup RF5$) era 9, aplicando-se o indicador proposto para a RTM-E para estabelecer a dependência entre RF3 e RF5, tem-se na Equação 8:

$$J(RF3,RF5) = \frac{n(RF3 \cap RF5)}{n(RF3 \cup RF5)} = \frac{6}{9} = 66.67\% \quad (8)$$

Essa dependência de 66.67% está destacada na Figura 5.1, que corresponde à RTMe construída com a abordagem. Vale destacar que as cores indicam o nível de dependência mapeada da seguinte forma: verde para dependência fraca e vermelho para dependência forte. Quando não existe relacionamento, a célula não é colorida. A Figura 5.2 ilustra os conjuntos interseção e união dos dados de entrada quando a abordagem RTM-E é aplicada nos requisitos RF3 e RF5 utilizados anteriormente. Vale ressaltar que, para minimizar o erro de cadastro de requisitos, a ferramenta COCAR apresenta uma lista das entradas já cadastradas para que os mesmos dados de entrada não sejam cadastrados com nomes diferentes.

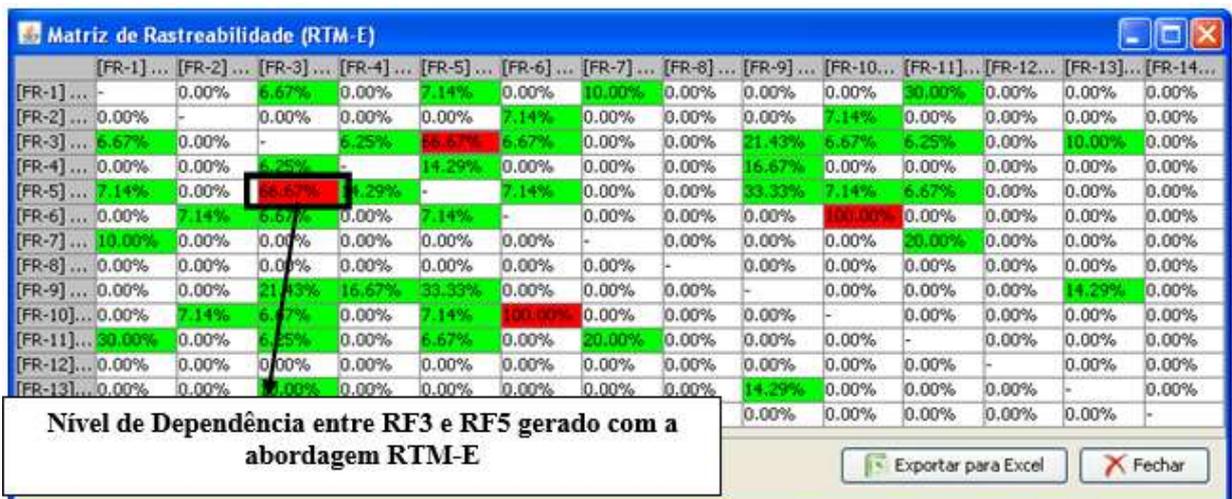


Figura 5.1 RTM gerada com o uso da abordagem RTM-E



Figura 5.2 Exemplo de União e Intersecção dos dados de entrada

RTM-NLP

Para exemplificar a abordagem RTM-NLP, é utilizado o mesmo par de requisitos funcionais RF3 e RF5 utilizado para ilustrar a abordagem RTM-E anteriormente. Sendo assim, todo conteúdo de texto de cada um desses requisitos funcionais passou pelo pré-processamento, retirando-se os *stopwords* e aplicando-se o *stemming*. Feito isso, foi utilizado o Vetor de Frequência e Similaridade do Cosseno. A Figura 5.3 mostra a RTM gerada a partir da abordagem RTM-NLP para o sistema do estudo de caso. Faz sentido o valor alto (88,63%) no relacionamento entre RF3 e RF5, uma vez que o texto dos dois requisitos tem grande similaridade.

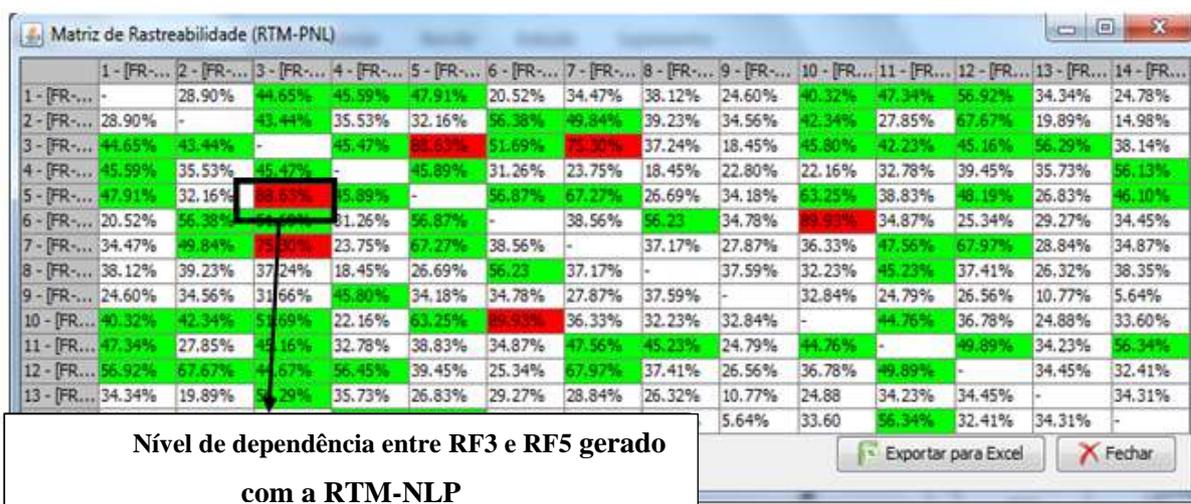


Figura 5.3 RTM gerada com o uso da abordagem RTM-NLP

RTM-Fuzzy

Para exemplificar a abordagem RTM-Fuzzy, foi escolhido o par de requisitos composto pelo RF3 – já utilizado anteriormente – e o requisito funcional 7 (RF7), relacionado com os relatórios sobre o estoque. Tal combinação foi utilizada pelo fato de não existirem dados de entrada comum entre eles, ou seja, a abordagem RTM-E não detecta nenhuma dependência entre os dois requisitos funcionais. Por outro lado, a abordagem RTM-NLP indica dependência forte entre esses requisitos (75,3%). Isso ocorre, pois ambos lidam com um escopo semelhante no texto que os descreve. Na terceira regra utilizada no Sistema Fuzzy (disponível na Figura 4-6), caso tenhamos RTM-E com “dependência nula” e RTM-NLP com “dependência forte”, então a dependência resultante era a “dependência fraca”. É importante observar que essa regra não determina diretamente que a dependência será fraca. Antes disso, é aplicada a Inferência Mandami para calcular o valor na função de pertinência de saída. Neste caso, o valor calculado foi de 42,5. Como pode ser observado na Figura 5.4, esse valor indica “dependência fraca” com pertinência 1.

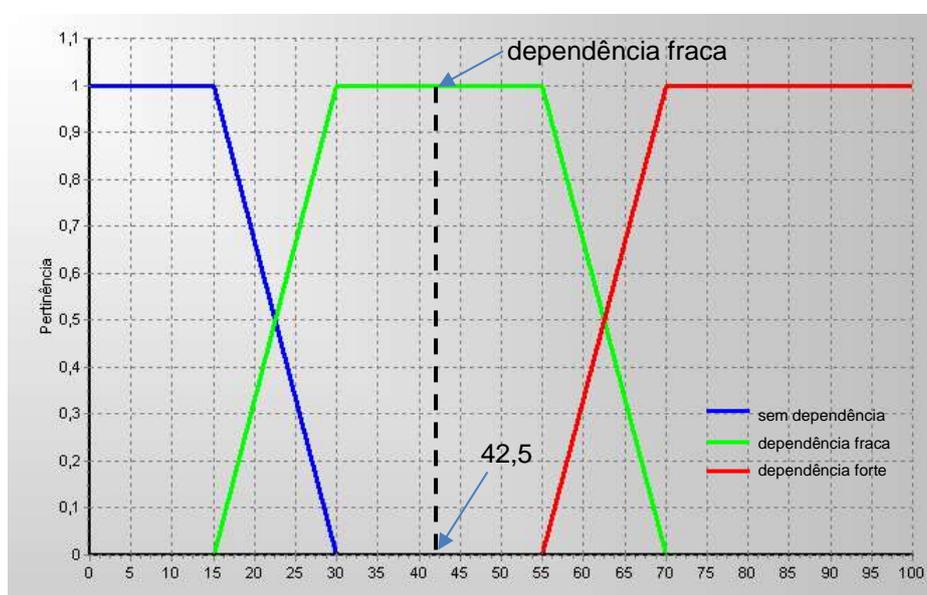


Figura 5.4 Cálculo da dependência entre RF3 e RF7

RTM-N

A RTM-N também será exemplificada com os requisitos RF3 e RF5. Conforme descrito na Seção 4.5, a RTM-N utiliza a RTM-E e RTM-NLP como entrada para determinar o nível de dependência. Na Figura 5.5, é mostrada como os valores da RTM-E e RTM-NLP são utilizados como entrada para rede neural e como a rede neural já treinada termina o valor de saída. Neste caso, RTM-N também identifica probabilidade de 99% de “dependência forte” entre RF3 e RF5.

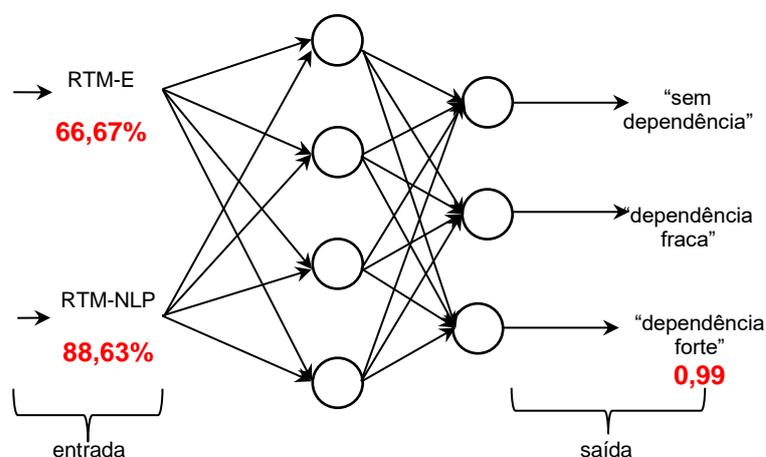


Figura 5.5 Cálculo da dependência entre RF3 e RF7 com a RTM-N

Resultados: Os resultados são apresentados na sequência:

RTM-E: efetividade de 78,1%

RTM-NLP: efetividade de 71,7%

RTM-Fuzzy: efetividade de 81,7%

RTM-N: efetividade de 82,3%

Análise dos Resultados: Os resultados do estudo de caso mostram que as abordagens podem ser utilizadas na indústria. O grande questionamento que surgiu por parte da empresa foi: “o que fazer com a RTM depois de estabelecida?”. Mesmo não sendo o foco desta pesquisa, apresentou-se para a empresa uma possibilidade de aplicação da RTM para a seleção de um conjunto de casos de testes de regressão e caracterização de impacto de mudança. Esse estudo está sendo conduzido na empresa com base na experiência adquirida durante a condução do estudo de caso apresentado no Capítulo 6.

Ameaças à validade: Como ameaça à validade do estudo de caso, pode-se destacar a construção da RTM de referência. Nesse caso, o contato entre o autor desta tese e os envolvidos no projeto foi bastante frequente, realizado por meio de reuniões presenciais para refinar a RTM de referência e esclarecer algumas dúvidas nos requisitos funcionais.

5.3 Estudos Experimentais Controlados

Nesta seção, serão apresentados os três experimentos mencionados anteriormente. Eles seguiram a estrutura proposta por Wohlin e outros (2000) e destacam: contexto, objetivo, participantes, artefatos utilizados, métrica, resultado, análise dos resultados e ameaças à validade.

A Tabela 4 mostra um panorama geral dos experimentos destacando as variações que ocorreram entre eles.

5.3.1 Experimento 1

Este estudo experimental foi apresentado no XXVI Simpósio Brasileiro de Engenharia de Software (DI THOMMAZO *et al.*, 2012) e avaliou as abordagens RTM-E e RTM-NLP. Toda a descrição na sequência foi extraída do artigo publicado.

Contexto: O estudo experimental foi realizado com alunos da disciplina de Linguagem de Programação II do curso de graduação em Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP) no campus de São Carlos. Como atividade da disciplina, cada aluno deveria desenvolver uma aplicação que envolvesse um *stakeholder* real. Além disso, o Documento de Requisitos (DR) deveria ser criado na ferramenta COCAR.

Tabela 4: Comparação entre os experimentos

	Experimento 1	Experimento 2	Experimento 3
Evento em que foi publicado	SBES 2012	SEKE 2013 e SBES 2013	ICEIS 2014
Artefatos	18 documentos de requisitos feitos por alunos de graduação	14 documentos de requisitos feitos em duplas por alunos de graduação	16 documentos de requisitos feitos em duplas por alunos de graduação
Abordagens avaliadas	RTM-E e RTM-NLP	RTM-E, RTM-NLP, RTM-Fuzzy E RTM-N	RTM-E, RTM-NLP, RTM-Fuzzy E RTM-N
Modificações para reduzir ameaças a validade		Construção do documento de requisitos por dupla de aluno	Construção do documento de requisitos por dupla de aluno
		Inspeção no documento de requisitos	Inspeção no documento de requisitos
Modificações para aumentar a eficácia das abordagens		Modificação no cálculo da RTM-NLP, considerando todos os campos de cada requisito funcional contido no template de Kawai (2005)	
			Novos valores para definir nível de dependência na RTM-E e RTM-NLP (uso de árvore de decisão)
			Uso de algoritmos genéticos para definir funções de pertinência na RTM-Fuzzy
			Novo treinamento da Rede Neural para a RTM-N

Objetivo: avaliar a eficácia das abordagens RTM-E e RTM-NLP em comparação com a RTM de referência (chamada RTM-Ref) construída a partir da análise criteriosa do DR. A criação da RTM-Ref será detalhada na sequência.

Participantes: 18 alunos de graduação do curso Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de São Paulo, campus de São Carlos.

Artefatos utilizados:

- Documento de Requisitos:
 - elaborado pelos próprios alunos;
 - referente a uma aplicação de utilidade prática real;

- relativo a aplicações de sistemas de informação com as funcionalidades básicas de cadastro, recuperação, atualização e exclusão de dados.
- relativo às necessidades de um *stakeholder* real, com amplo conhecimento do sistema que seria desenvolvido.
- inserido na ferramenta COCAR.

- RTM-Ref:

- criada a partir do documento de requisitos registrado na ferramenta COCAR.
- elaborada com base na leitura e análise detalhada de cada par de requisitos funcionais e determinando a dependência entre esses requisitos funcionais como nula, fraca ou forte.
- registrada em uma planilha para que a matriz RTM-Ref criada pudesse ser comparada com a RTMe e RTMnlp de cada sistema.
- construída pelo autor desta tese com colaboração dos autores do artigo que publicou esse experimento. Houve consulta ao aluno autor do documento de requisitos sempre que houvesse alguma dúvida. Em alguns casos, foi necessário que o aluno fizesse contato com o *stakeholder* para avaliar a dependência entre os requisitos funcionais, melhorando a especificação do documento de requisitos e indicando o relacionamento de forma correta.

Métrica: a métrica utilizada foi a eficácia das abordagens RTM-E e RTM-NLP no que diz respeito à coincidência das dependências encontradas por cada abordagem em relação à RTM-Ref, conforme já definido na Equação 7.

Resultados: Os resultados da comparação entre os dados da RTM-Ref com a RTM-E e RTM-NLP estão apresentados na Tabela 5. A primeira coluna tem o nome do sistema que foi especificado. Na segunda coluna está a quantidade de requisitos funcionais. A terceira coluna é composta pelo total de possíveis dependências (forte, fraca e nula) que podem existir entre os requisitos funcionais, ou seja, é composta pelo total de células que ficam abaixo da diagonal principal da matriz e dada pela Equação 6. A quarta coluna tem o total de relacionamentos coincidentes entre as matrizes RTM-E e RTM-Ref; por exemplo: se na RTM-Ref foi determinada uma

dependência “forte” em uma célula e a abordagem RTM-E também registrou a dependência como “forte” na mesma posição, então contabiliza-se um relacionamento correto. A eficácia da abordagem RTM-E é dada pela relação da quantidade de dependências encontradas corretamente por essa abordagem (quarta coluna) pelo total de dependências que poderiam ser encontradas (terceira coluna). Os valores da eficácia da abordagem RTM-E estão disponíveis na quinta coluna. A sexta coluna mostra o total de relacionamentos coincidentes entre as matrizes RTM-NLP e RTM-Ref. A sétima coluna mostra a eficácia da abordagem RTM-NLP, calculada pela divisão entre a quantidade de dependências encontradas corretamente pela abordagem RTM-NLP (sexta coluna) pelo total de dependências que poderiam ser encontradas (terceira coluna).

Tabela 5 – Resultados do estudo experimental

Sistema	Quantidade de Requisitos	Possibilidade de dependências (T)	RTM-E Corretos	RTM-E Eficácia (%)	RTM-NLP Corretos	RTM-NLP Eficácia(%)
Sistema Natação	30	435	376	86	235	54
Controle de Estoque	17	136	135	99	78	57
Sistema de Compras	15	105	75	71	58	55
Venda Livros	19	171	112	65	61	36
Sistema para Hotel	16	120	97	81	72	60
Vendas Cosméticos	15	105	93	89	67	64
Vendas Eletrônicos	15	105	95	90	45	43
Gestão de Moradia	19	171	139	81	87	51
Consultório Médico	16	120	89	74	67	56
Manutenção/Hardware	14	91	72	79	56	62
Loja de Roupas	20	190	156	82	72	38
Grupo de Jovens	19	171	156	91	98	57
Gestão CDs e DVDs	18	153	125	82	72	47
Gamer	18	154	123	80	69	45
Sist. Personal Trainer	15	105	93	89	63	60
Gestão de Projetos	16	120	92	77	64	53
Gestão de Tarefas	9	36	28	78	19	53
Marmoraria	16	120	101	84	71	59

Análise dos resultados: A partir dos dados da eficácia das abordagens propostas é possível observar que a RTM-E mostrou-se mais eficaz que a RTM-NLP. Foi utilizado o software SigmaPlot (<http://www.sigmaplot.com/>) para a análise dos dados. Foi aplicado o teste Shapiro-Wilk para confirmar a distribuição normal dos dados. Comprovada a normalidade dos dados (para RTM-E, p-value = 0.1147; e para RTM-NLP, p-value = 0.1679), foi realizado o teste T para avaliar a confiabilidade. Para a RTM-E tem-se a média de 82,2% com desvio padrão de 7,4% e intervalo de confiança (95%) de 78,2% até 86,1%. Para a RTM-NLP, tem-se média de 52,8% com desvio padrão de 8,1%. Nesse caso, o intervalo de confiança é de 48,3% até 56,5%.

Como a eficácia da abordagem RTM-NLP foi consideravelmente menor que a da RTM-E, fez-se um estudo para investigar os motivos que levaram a esse resultado. Notou-se que houve muitos casos de falsos positivos, ou seja, dependências foram encontradas quando não havia relação entre os requisitos funcionais. De acordo com Sundaram e outros (2010), a ocorrência de falsos positivos é uma característica do uso de PLN, embora esse tipo de processamento recupere com facilidade os relacionamentos entre os requisitos funcionais. Por outro lado, a RTM-E mostra-se muito eficaz ao minimizar os falsos-positivos.

Ameaças à Validade: O estudo possui alguns riscos à validade, podendo-se citar, sua realização em ambiente acadêmico. Por esse motivo, um possível problema é a inexperiências dos alunos para fazer o levantamento de requisitos com os *stakeholders*. No entanto, por se tratarem de sistemas de informação de domínios conhecidos, considera-se que esse fato minimize a questão da inexperiência. Para minimizar essa ameaça também foi feito treinamento para os alunos sobre os conceitos envolvidos no experimento. Outro risco é o fato da RTM-Ref ter sido construída por pessoas que não tiveram contato direto com o *stakeholder* e, portanto, essa matriz pode retratar os eventuais problemas que tenham ocorrido na elaboração dos próprios documentos de requisitos. Para minimizar esse risco, sempre que houve dúvidas na determinação de um relacionamento, a ajuda do aluno foi solicitada. Em alguns casos, foi necessária uma compreensão melhor dos requisitos junto ao *stakeholder*, o que certamente minimizou a ocorrência de erros na criação da RTM-Ref.

Também faz sentido considerar que os resultados poderiam ser diferentes com um conjunto diferente de participantes, com um conjunto diferente de sistemas e requisitos funcionais.

5.3.2 Experimento 2

Os resultados desse estudo experimental foram apresentados em dois eventos: no XXVII Simpósio Brasileiro de Engenharia de Software (DI THOMMAZO *et al.*, 2013a) foi apresentada a RTM-Fuzzy e no XXV International Conference on Software

Engineering and Knowledge Engineering (DI THOMMAZO *et al.*, 2013b) foi apresentada a RTM-N.

Embora o objetivo principal desse estudo tenha sido avaliar as duas abordagens mencionadas anteriormente, aproveitou-se a oportunidade para reavaliar as abordagens RTM-E e RTM-NLP. A RTM-NLP que havia apresentado uma efetividade baixa (52,8%) no experimento anterior foi ajustada: anteriormente para o cálculo da similaridade era considerado apenas o texto contido no campo processamento do *template* de cadastro de cada requisito funcional. Com a mudança, neste experimento foram considerados os textos de todos os campos de cadastro de cada requisito funcional. Outro ajuste feito na condução do experimento foi que, para mitigar uma das ameaças à validade, os documentos de requisitos foram produzidos por dois alunos e não mais individualmente. Também inseriu-se uma etapa de inspeção no Documento de Requisitos para aumentar a qualidade do artefato produzido. As descrições do experimento foram extraídas dos dois artigos publicados.

Contexto: O estudo experimental foi realizado com alunos da disciplina de Engenharia de Software do curso de graduação em Bacharelado em Ciência de Computação da Universidade Federal de São Carlos (UFSCar). Como atividade da disciplina, cada par de alunos deveria desenvolver uma aplicação que envolvesse um *stakeholder* real. Além disso, o documento de requisitos deveria ser criado na ferramenta COCAR.

Objetivo: avaliar a eficácia das abordagens RTM-E, RTM-NLP, RTM-Fuzzy E RTM-N em comparação com a RTM de referência (chamada RTM-Ref) construída a partir da análise criteriosa do documento de requisitos.

Participantes: 28 alunos de graduação do curso de Bacharelado em Ciência de Computação da Universidade Federal de São Carlos (UFSCar).

Artefatos utilizados:

- Documento de Requisitos, com as seguintes características:
 - elaborado por dois alunos;

- referente a uma aplicação de utilidade prática real;
- relativos a aplicações de sistemas de informação com as funcionalidades básicas de cadastro, recuperação, atualização e exclusão de dados;
- relativo às necessidades de um *stakeholder* real, com amplo conhecimento do sistema que seria desenvolvido.
- inserido na ferramenta COCAR.

- RTM-Ref:

- criada a partir do documento de requisitos registrado na ferramenta COCAR;
- elaborada com base na leitura e análise detalhada de cada par de requisito funcional e determinando a dependência entre esses requisitos funcionais como nula, fraca ou forte;
- registrada em uma planilha para que a matriz RTM-Ref criada pudesse ser comparada com a RTMe, RTMnlp, RTMfuzzy e RTMn de cada sistema;
- construída pelo autor desta tese em colaboração com os autores do artigo que publicou o experimento com consulta à dupla de alunos autores do documento de requisitos, sempre que houvesse alguma dúvida. Assim como no experimento anterior, em alguns casos foi necessário que o aluno fizesse contato com o *stakeholder* para avaliar a dependência entre os requisitos funcionais, melhorando a especificação do documento de requisitos e indicando o relacionamento de forma correta.

Métrica: foi utilizada a mesma métrica do experimento anterior: a eficácia foi determinada pela relação entre a quantidade de relações encontradas corretamente em cada abordagem pela quantidade total de possíveis relacionamentos existentes entre os requisitos funcionais (Equações 6 e 7).

Resultados: Os resultados da comparação entre os dados da RTM-Ref com a RTM-E, RTM-NLP, RTM-Fuzzy e RTM-N estão apresentados na Tabela 6. A primeira coluna tem o nome do sistema que foi especificado. Na segunda coluna está a

quantidade de requisitos funcionais. A terceira coluna é composta pelo total de possíveis dependências (forte, fraca e nula) que podem existir entre os requisitos funcionais, ou seja, é composta pelo total de células que ficam abaixo da diagonal principal da matriz e é dada pela Equação 6. A quarta coluna tem o total de relacionamentos coincidentes entre as matrizes RTM-E e RTM-Ref. A eficácia da abordagem RTM-E é dada pela relação da quantidade de dependências encontradas corretamente por essa abordagem (quarta coluna) pelo total de dependências que poderiam ser encontradas (terceira coluna). Os valores da eficácia da abordagem RTM-E estão disponíveis na quinta coluna. A sexta coluna mostra o total de relacionamentos coincidentes entre as matrizes RTM-NLP e RTM-Ref. A sétima coluna mostra a eficácia da abordagem RTM-NLP, calculada pela divisão entre a quantidade de dependências encontradas corretamente pela abordagem RTM-NLP (sexta coluna) pelo total de dependências que poderiam ser encontradas (terceira coluna). Os valores da eficácia da abordagem RTM-Fuzzy estão disponíveis na nona coluna, enquanto que a eficácia da RTM-N é apresentada na décima primeira coluna.

Tabela 6 – Resultados do segundo estudo experimental

Sistema	Qtdd Req	Possib. dep.	RTM-E		RTM-NLP		RTM-Fuzzy		RTM-N	
			Correto	Eficácia	Correto	Eficácia	Correto	Eficácia	Correto	Eficácia
Zoológico	19	171	131	77%	138	81%	143	84%	149	87%
Moradia	24	276	233	84%	205	74%	241	87%	235	85%
República	28	378	295	78%	325	86%	342	90%	337	89%
Táxi	15	105	82	78%	77	73%	85	81%	91	87%
Roupas	27	351	295	84%	253	72%	296	84%	319	91%
Frete	16	120	98	82%	85	71%	102	85%	102	85%
Contábil	24	276	204	74%	181	66%	212	77%	218	79%
Financeiro	17	136	94	69%	101	74%	101	74%	107	79%
Admin.	19	171	134	78%	129	75%	137	80%	148	87%
Loja Livros	19	171	129	75%	145	85%	147	86%	149	87%
Ingressos	15	105	88	84%	91	87%	94	90%	95	90%
Filmes	16	120	88	73%	82	68%	97	81%	91	76%
Ônibus	15	105	72	69%	78	74%	81	77%	81	77%
Escola	15	105	82	78%	77	73%	84	80%	91	87%

Análise dos resultados: A análise estatística foi conduzida utilizando o software SigmaPlot. Aplicando-se o teste de Shapiro-Wilk foi constatado que os dados seguem uma distribuição normal. Para comparar a efetividade entre as abordagens foi utilizada a análise de variância para medidas repetidas (ANOVA) com *post hoc* de Holm-Sidak. A RTM-N obteve a maior efetividade com média de 84.7% e desvio padrão de 4.9. A segunda melhor efetividade foi com a RTM-Fuzzy com média de 82.6 e desvio padrão de 4.9. A RTM-E obteve efetividade média de 77,4 com desvio padrão de 5,1. A abordagem RTM-NLP, por sua vez, obteve efetividade média de 75,7% com desvio padrão de 6,6.

A partir desses resultados, observa-se que a abordagem RTM-E obteve resultados semelhantes aos resultados do primeiro estudo experimental. Observa-se também que a modificação feita na abordagem RTM-NLP (que passou a considerar todos os campos cadastrados em cada requisito funcional e não somente o texto do campo processamento) surtiu um efeito muito positivo, uma vez que a efetividade subiu de 52,8% para 75,7%.

Além desse resultado, pode ser observado que as abordagens RTM-Fuzzy e RTM-N produziram resultados melhores que os das abordagens RTM-E e RTM-NLP isoladamente. Esse resultado corrobora a proposta da tese em utilizar inteligência computacional para melhorar a acurácia nos links de rastreabilidade.

Ameaças à Validade: Assim como no estudo experimental apresentado anteriormente, este também possui alguns riscos à validade. Uma das ameaças é o ambiente acadêmico onde o experimento foi conduzido. Uma das ameaças que havia no experimento anterior relacionado com a qualidade do documento de requisitos produzido pelos alunos foi mitigado trabalhando com dois alunos em cada sistema e com uma etapa de inspeção de cada DR para posterior retirada de defeitos dos artefatos produzidos. Para minimizar a ameaça da pouca experiência dos alunos com levantamento e análise de requisitos também foi feito treinamento sobre essas atividades. Outro risco que persistiu foi o fato da RTM-Ref também ter sido construída por pessoas que não tiveram contato direto com o *stakeholder* e, portanto, essa matriz pode retratar os eventuais problemas que tenham ocorrido na elaboração dos próprios

DRs. Da mesma forma como no experimento anterior, sempre que houve dúvidas na determinação de um relacionamento, a ajuda do aluno foi solicitada para minimização desse risco. Novamente faz sentido considerar que os resultados poderiam ser diferentes com um conjunto diferente de participantes, com um conjunto diferente de sistemas e requisitos funcionais.

5.3.3 Experimento 3

Os resultados desse terceiro experimento foram apresentados no XXVI International Conference on Enterprise Information Systems (DI THOMMAZO *et al.*, 2014). Para a realização desse estudo experimental as quatro abordagens propostas neste trabalho foram aprimoradas com uso de técnicas de inteligência computacional para serem novamente avaliadas. As modificações pelas quais cada abordagem passou são listadas na sequência:

- RTM-E: os intervalos que identificam se a dependência entre dois requisitos funcionais é nula, fraca ou forte foram redefinidos com ajuda de uma árvore de decisão conforme detalhado na Seção 4.2;
- RTM-NLP: os intervalos que identificam se a dependência entre dois requisitos funcionais é nula, fraca ou forte também foram redefinidos com ajuda de uma árvore de decisão conforme detalhado na Seção 4.3;
- RTM-Fuzzy: as funções de pertinência foram redefinidas com o uso de algoritmos genéticos, conforme detalhado na Seção 4.4
- RTM-N: os dados do segundo estudo experimental foram utilizados para treinar novamente a rede neural que faz a detecção do nível de dependência entre os requisitos funcionais.

Como esta tese busca constatar que o uso de técnicas de inteligência computacional melhora a acurácia dos links de rastreabilidade, a proposta deste estudo experimental foi verificar se as abordagens RTM-Fuzzy e RTM-N possuíam maior acurácia na detecção dos links de rastreabilidade em relação às abordagens RTM-E e RTM-NLP. Este estudo também fez uma comparação entre os resultados da

eficácia com as abordagens refinadas e com as abordagens não refinadas. Essa comparação foi feita com o intuito de mostrar que o refinamento das abordagens com as técnicas de inteligência computacional melhora a acurácia dos links de rastreabilidade, corroborando assim a tese proposta. As descrições do experimento foram extraídas dos artigos publicados.

Contexto: O estudo experimental foi realizado com alunos de graduação em Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP) no campus de São Carlos. Como atividade da disciplina, cada par de alunos deveria desenvolver uma aplicação que envolvesse um *stakeholder* real. Além disso, o DR deveria ser criado na ferramenta COCAR.

Objetivo: o estudo teve dois objetivos principais:

1. avaliar a eficácia das abordagens RTM-E, RTM-NLP, RTM-Fuzzy E RTM-N que foram refinadas com as técnicas de inteligência computacional em comparação com a RTM de referência (chamada RTM-Ref), construída a partir da análise criteriosa dos DRs. (referente à análise de resultados Parte 1)
2. comparar os resultados da avaliação da eficácia das quatro abordagens refinadas com as técnicas de inteligência computacional em relação à eficácia das quatro abordagens sem os refinamentos citados. Em outras palavras, os documentos de requisitos utilizados neste experimento também foram processados com as abordagens na sua versão anterior, isto é, na versão utilizada no Experimento 2. (referente à análise de resultados Parte 2)

Participantes: 32 alunos de graduação do curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP), campus de São Carlos.

Artefatos utilizados:

- Documento de Requisitos, com as seguintes características:
 - elaborado por dois alunos;

- referente a uma aplicação de utilidade prática real;
- relativos a aplicações de sistemas de informação com as funcionalidades básicas de cadastro, recuperação, atualização e exclusão de dados;
- relativo às necessidades de um *stakeholder* real, com amplo conhecimento do sistema que seria desenvolvido.
- inserido na ferramenta COCAR.

- RTM-Ref:

- criada a partir do documento de requisitos registrado na ferramenta COCAR.
- elaborada com base na leitura e análise detalhada de cada par de requisito funcional e determinando a dependência entre esses requisitos funcionais como nula, fraca ou forte.
- registrada em uma planilha para que a matriz RTM-Ref criada pudesse ser comparada com a RTMe, RTMnlp, RTMfuzzy e RTMn de cada sistema.
- construída pelo autor desta tese em colaboração com outros autores do artigo que publicou esse experimento. Foi criada com consulta à dupla de alunos autores do documento de requisitos, sempre que houvesse alguma dúvida. Assim como no experimento anterior, em alguns casos foi necessário que o aluno fizesse contato com o *stakeholder* para avaliar a dependência entre os requisitos funcionais, melhorando a especificação do documento de requisitos e indicando o relacionamento de forma correta.

Métrica: foi utilizada a mesma métrica do primeiro estudo experimental: a eficácia foi determinada pela relação entre a quantidade de relações encontradas corretamente em cada abordagem pela quantidade total de possíveis relacionamentos existentes entre os requisitos funcionais (Equações 6 e 7).

Resultados e Análises: Os resultados e as análises são apresentados em duas partes: na Parte 1 são mostrados os resultados e as análises da comparação entre as quatro abordagens refinadas com as técnicas de inteligência computacional em

relação aos dados das RTM-Ref de cada sistema. Na Parte 2 mostram-se os resultados e as análises da comparação entre a eficácia de cada abordagem refinada em relação às mesmas abordagens não refinadas.

Parte 1:

Resultados: Os resultados da comparação entre os dados da RTM-Ref com a RTM-E, RTM-NLP, RTM-Fuzzy e RTM-N estão apresentados na Tabela 7. A primeira coluna tem o nome do sistema que foi especificado. Na segunda coluna está a quantidade de requisitos funcionais. A terceira coluna é composta pelo total de possíveis dependências (forte, fraca e nula) que podem existir entre os requisitos funcionais, ou seja, é composta pelo total de células que ficam abaixo da diagonal principal da matriz e dada pela Equação 6. A quarta coluna tem o total de relacionamentos coincidentes entre as matrizes RTM-E e RTM-Ref. A eficácia da abordagem RTM-E é dada pela relação da quantidade de dependências encontradas corretamente por essa abordagem (quarta coluna) pelo total de dependências que poderiam ser encontradas (terceira coluna). Os valores da eficácia da abordagem RTM-E estão disponíveis na quinta coluna. A sexta coluna mostra o total de relacionamentos coincidentes entre as matrizes RTM-NLP e RTM-Ref. A sétima coluna mostra a eficácia da abordagem RTM-NLP, calculada pela divisão entre a quantidade de dependências encontradas corretamente pela abordagem RTM-NLP (sétima coluna) pelo total de dependências que poderiam ser encontradas (terceira coluna). Os valores da eficácia da abordagem RTM-Fuzzy estão disponíveis na nona coluna, enquanto a eficácia da RTM-N é apresentada na décima primeira coluna.

Análise dos resultados: A análise estatística foi conduzida utilizando o software SigmaPlot. Aplicando o teste de Shapiro-Wilk foi constatado que os dados seguem uma distribuição normal. Para comparar a efetividade entre as abordagens, foi utilizada a análise de variância para medidas repetidas (ANOVA) com *post hoc* de Holm-Sidak. A RTM-N obteve a maior efetividade com média de 87,3% e desvio padrão de 2,2. A segunda melhor efetividade foi com a RTM-Fuzzy com média de 86,6 e desvio padrão de 3,1. A RTM-E obteve efetividade média de 79,6 com desvio padrão de 3,5; enquanto a abordagem RTM-NLP obteve efetividade média de 75,7% com desvio padrão de 3,7.

Tabela 7 – Resultados do terceiro estudo experimental

Sistema	Qtdd Req	Possib. dep.	RTM-E		RTM-NLP		RTM-Fuzzy		RTM-N	
			Correto	Eficácia	Correto	Eficácia	Correto	Eficácia	Correto	Eficácia
Médico	17	136	115	85%	106	78%	124	91%	122	90%
Aluguel carros	23	253	204	81%	190	75%	215	85%	217	86%
Vendas	18	153	122	80%	121	79%	130	85%	133	87%
Roupas	17	136	103	76%	105	77%	117	86%	119	88%
Venda carros	16	120	97	81%	93	78%	107	89%	107	89%
Moradia	16	120	103	86%	93	78%	110	92%	110	92%
Livraria	21	210	167	80%	157	75%	177	84%	180	86%
Pizzaria	16	120	95	79%	86	72%	104	87%	106	88%
Vendas	22	231	195	84%	186	81%	213	92%	206	89%
Administração	17	136	103	76%	101	74%	113	83%	117	86%
Filmes	17	136	103	76%	93	68%	112	82%	114	84%
Jogos	18	153	119	78%	118	77%	130	85%	131	86%
Refeição	19	171	134	78%	132	77%	144	84%	145	85%
Estudante	17	136	102	75%	94	69%	114	84%	117	86%
Computadores	15	105	82	78%	78	74%	91	87%	91	87%
Ingressos	20	190	159	84%	155	82%	169	89%	171	90%

A partir desses dados, observa-se que abordagem RTM-E obteve resultados semelhantes com os dos dois estudos experimentais anteriores. O mesmo pode ser constatado com relação à abordagem RTM-NLP e ao estudo experimental anterior.

Assim como no estudo anterior, as abordagens RTM-Fuzzy e RTM-N produziram resultados melhores que as abordagens RTM-E e RTM-NLP isoladamente. Esse resultado corrobora a proposta da tese: utilizar inteligência computacional melhorar a acurácia nos links de rastreabilidade. É possível observar também que a abordagem RTM-Fuzzy produziu melhores resultados se comparados com o estudo experimental anterior, assim como a RTM-N.

Parte 2

Resultados: Os resultados da comparação entre os dados da efetividade das quatro abordagens refinadas com técnicas de inteligência computacional em relação aos dados da efetividade das quatro abordagens sem o refinamento estão dispostos na

Tabela 8. A primeira coluna tem o nome do sistema que foi especificado. Na segunda coluna, está a eficácia da abordagem RTM-E com os refinamentos, enquanto a terceira coluna contém a eficácia da abordagem RTM-E sem o refinamento das técnicas de inteligência computacional. Na quarta coluna, está a eficácia da abordagem RTM-NLP com os refinamentos, enquanto na quinta está a eficácia da abordagem RTM-NLP sem o refinamento das técnicas de inteligência computacional. De forma semelhante, na sexta coluna está a eficácia da abordagem RTM-Fuzzy com os refinamentos, enquanto na sétima registra-se a eficácia da abordagem RTM-Fuzzy sem o refinamento das técnicas de inteligência computacional. Na oitava coluna está, por sua vez, a eficácia da abordagem RTM-N com os refinamentos, enquanto a nona indica a eficácia da abordagem RTM-N sem o refinamento das técnicas de inteligência computacional.

Tabela 8 – Resultados do estudo experimental comparando as abordagens com as melhorias

Sistema	RTM-E (refinada)	RTM-E	RTM-NLP (refinada)	RTM-NLP	RTM-Fuzzy (refinada)	RTM-Fuzzy	RTM-N (refinada)	RTM-N
Médico	85%	77%	78%	84%	91%	87%	90%	85%
Aluguel carros	81%	75%	75%	83%	85%	85%	86%	81%
Vendas	80%	79%	79%	81%	85%	84%	87%	80%
Roupas	76%	75%	77%	83%	86%	85%	88%	76%
Venda carros	81%	75%	78%	85%	89%	88%	89%	81%
Moradia	86%	72%	78%	87%	92%	90%	92%	86%
Livraria	80%	75%	75%	79%	84%	82%	86%	80%
Pizzaria	79%	72%	72%	85%	87%	88%	88%	79%
Vendas	84%	81%	81%	86%	92%	87%	89%	84%
Administração	76%	73%	74%	81%	83%	85%	86%	76%
Filmes	76%	68%	68%	80%	82%	83%	84%	76%
Jogos	78%	76%	77%	83%	85%	85%	86%	78%
Refeição	78%	77%	77%	83%	84%	84%	85%	78%
Student	75%	65%	69%	82%	84%	85%	86%	75%
Computadores	78%	72%	74%	86%	87%	85%	87%	78%
Ingressos	84%	81%	82%	83%	89%	87%	90%	84%

Análise dos resultados: A análise estatística foi conduzida utilizando o software SPSS, Version 17.0 (SPSS, Chicago, IL). Aplicando-se o teste Shapiro-Wilk foi constatado que os dados seguem uma distribuição normal. O teste paramétrico

aplicado para verificar a possível diferença entre as médias de cada abordagem e sua respectiva versão melhorada foi utilizado o teste t de *Student* pareado. A magnitude do tamanho do efeito da aplicação das técnicas de inteligência computacional para melhorar o desempenho de cada abordagem (a diferença entre a primeira versão de cada abordagem e sua respectiva versão refinada) foi calculada e a escala de classificação da magnitude do tamanho do efeito proposta por Field (Field, 2009) foi utilizada. Os resultados são apresentados em média \pm DP. O nível de significância adotado foi $p < 0,05$. Os valores da média e desvio padrão das abordagens são destacados na Tabela 9.

Tabela 9 – Comparação entre a primeira versão de cada abordagem e sua respectiva versão refinada

Sistema	RTM-E	RTM-NLP	RTM-Fuzzy	RTM-N
Primeira versão (referente ao Experimento 2)	78,89 \pm 3,49	74,56 \pm 4,27	83,33 \pm 2,37	85,62 \pm 2,06
Versão melhorada (referente ao Experimento 3)	79,81 \pm 3,46*	75,87 \pm 3,86*	86,56 \pm 3,18*	87,43 \pm 2,12*

Os resultados expressam que houve melhora após a aplicação das técnicas de inteligência computacional ($p < 0,05$). Isso dá indicativos que a tese proposta neste trabalho de doutorado pode ser corroborada. Seguindo Field (2009), além de encontrar significância nos dados é importante determinar o tamanho do efeito, ou seja, definir a magnitude do efeito da mudança causada por algum tratamento. No caso, o tratamento realizado foi a modificação nas abordagens para determinar os links de rastreabilidade. A magnitude significa dizer se a melhora produzida por essa modificação foi pequena, moderada ou grande. Field (2009) mostra a escala do efeito baseada no cálculo do fator estatístico r , valor dado pelo software SPSS quando realizado o processamento do Teste-T pareado. Se r for maior que 0,5 significa dizer que a melhora foi “grande”. A tabela 10 resume os resultados.

Tabela 10 – Magnitude do tamanho do efeito da aplicação das melhorias em cada abordagem.

Comparação	r	Tamanho do efeito
RTM-E x RTM-E (refinada)	0,77	Grande
RTM-NLP x RTM-NLP (refinada)	0,61	Grande
RTM-Fuzzy x RTM-Fuzzy (refinada)	0,88	Grande
RTM-N x RTM-N (refinada)	0,86	Grande

Ameaças à Validade: Assim como nos estudos experimentais apresentados anteriormente, este possui os mesmos riscos à validade: qualidade do documento de requisitos produzido pelos alunos (mesmo tendo o risco mitigado com a etapa de inspeção de cada documento de requisitos) e o fato da RTM-Ref também ter sido construída por pessoas que não tiveram contato direto com o *stakeholder*. Assim como nos experimentos anteriores, também faz sentido considerar que os resultados poderiam ser diferentes com um conjunto diferente de participantes, sistemas e requisitos funcionais.

5.4 Estudo de Caso 2

Contexto: Esse estudo de caso foi conduzido com o Sistema REDECA (www.redeca.org.br). O REDECA - Sistema de informação das redes pelo ECA - é um projeto de software livre que foi desenvolvido de forma colaborativa pela Fundação Telefônica com mais oito municípios paulistas. Este software armazena informações referentes a projetos sociais, convênio médico, dados escolares e condições de moradia de crianças e adolescentes em um banco de dados. Existe uma ampla documentação sobre o sistema.

Objetivo: Verificar como as abordagens propostas se comportavam na detecção de links de rastreabilidade na indústria.

Participantes: o autor desta tese e uma analista de requisitos

Artefatos:

- Documento de Requisitos: documentação com os requisitos do sistema contendo 37 requisitos funcionais. Todos foram inseridos na ferramenta COCAR para o posterior processamento.
- RTM-Ref: A construção da RTM de referência (gabarito) do sistema foi construída com base na leitura detalhada de toda a documentação do sistema. Essa etapa foi realizada pelo autor desta tese e outra analista.

Métrica: Como métrica, além da eficácia definida pela Equação 7, também foi calculado o valor de duas outras métricas disponíveis na literatura: Precisão e Retorno (BAEZA-YATES E RIBEIRO-NETO, 1999). Essas métricas estão definidas nas Equações 8 e 9 e foram utilizadas nesse estudo de caso para que pudesse ser feita uma comparação entre os resultados das abordagens propostas com os resultados de trabalhos encontrados na literatura.

$$\text{Retorno} = \frac{\text{links relevantes} \cap \text{links recuperados}}{\text{links relevantes}} \quad (8)$$

$$\text{Precisão} = \frac{\text{links relevantes} \cap \text{links recuperados}}{\text{links recuperados}} \quad (9)$$

Nas Equações 8 e 9, os “links relevantes” indicam uma dependência que realmente exista entre dois requisitos funcionais, enquanto um “link recuperado” indica uma relação de dependência que a abordagem encontrou, podendo ela estar certa ou errada. Quanto mais próximo do valor 1 estiverem as métricas, melhor a acurácia na detecção da rastreabilidade. Além dessas métricas, também foi feita a determinação da eficácia, segundo a Equação 7, que já foi utilizada nos estudos anteriores.

Resultados: Os resultados da eficácia são apresentados na Tabela 11. Os valores medidos em cada uma das abordagens apresentaram valores semelhantes aos obtidos nos experimentos descritos anteriormente.

Tabela 11. Resultados do estudo de caso

	RTM-E	RTM-NLP	RTM-Fuzzy	RTM-N
Eficácia	77,9	69,1	81,7	82,5

Com relação à Precisão e ao Retorno, é possível observar que, como destacado na literatura, quanto maior o Retorno, menor tende a ser a Precisão (CLELAND-HUANG et al.,2012). Também foi feita uma comparação com dois trabalhos da literatura sendo o primeiro conduzido por Abadi e outros (2008) e o segundo, por Oliveto e outros (2010). Ambos utilizaram o Modelo do Espaço Vetorial e a *Latent Semantic Indexing (LSI)*, detalhados na Seção 3.2.2. Essa comparação está disponível na Tabela 12.

Tabela 12. Comparação da Precisão e Retorno com trabalhos da Literatura

	RTM-E	RTM-NLP	RTM-Fuzzy	RTM-N	Abadi et al.,2008	Oliveto et al.,2010	Abadi et al.,2008	Oliveto et al.,2010
					Espaço Vetorial		LSI	
Precisão	0,67	0,42	0,69	0,69	0,43	0,68	0,26	0,78
Retorno	0,79	0,74	0,79	0,80	0,76	0,47	0,64	0,46

Análise dos Resultados: Vale destacar que, apesar de terem sido utilizadas métricas da literatura (Precisão e Retorno), a utilização delas com as abordagens propostas não se adequa exatamente na forma como utilizada nas abordagens na literatura. Isso ocorre porque as abordagens da literatura definem apenas se “existe dependência” ou “não existe dependência”. As abordagens propostas nesta tese definem o nível de dependência (“sem dependência”, “dependência fraca”, “dependência forte”). Sendo assim, caso alguma das abordagens propostas neste trabalho identifique algum link como “dependência fraca” e o gabarito (RTM-Ref) seria “dependência forte”, fica

registrado um “erro de detecção”, ao passo que as outras abordagens consideram tudo como “existe dependência”, independentemente do nível. Comparando os resultados é possível observar que as abordagens RTM-Fuzzy e RTM-N continuam com resultados melhores que as abordagens RTM-E e RTM-NLP. Comparando os resultados das abordagens propostas com os resultados da literatura existem situações com resultados são melhores nas abordagens propostas nesta tese e outras situações nas quais os resultados são melhores nas abordagens da literatura.

Ameaças à validade: Como ameaças à validade, pode-se destacar a construção da RTM de referência, uma vez que, dada a complexidade do sistema, em alguns momentos houve dúvidas se a dependência seria forte ou fraca.

5.5 Considerações Finais

Neste capítulo, foram apresentados os estudos experimentais realizados para avaliar as quatro abordagens propostas. Foram realizados três experimentos controlados. No primeiro deles, foram avaliadas as abordagens RTM-E e RTM-NLP. Os resultados obtidos foram utilizados para propor melhorias na abordagem RTM-NLP e motivaram a combinação das abordagens com uso de técnicas de inteligência computacional. A partir dessa combinação foram criadas as abordagens RTM-Fuzzy e RTM-N. Estas duas novas abordagens e as duas primeiras foram avaliadas por meio de um novo estudo experimental controlado. Esse estudo corroborou a proposta da tese, indicando que a combinação das abordagens com técnicas de inteligência computacional melhora a detecção dos links de rastreabilidade. Finalizado esse segundo estudo experimental, foi feita uma avaliação para proposição de melhoria nas quatro abordagens. Novamente foram utilizadas técnicas de inteligência computacional para refinar as abordagens, as quais foram avaliadas por um novo estudo experimental controlado. Os resultados desse estudo mais uma vez corroboraram a proposta da tese: as técnicas refinadas com técnicas de inteligência computacional tiveram melhor acurácia na detecção dos links de rastreabilidade.

Também foram apresentados dois estudos de caso real, conduzidos para avaliar a utilização das abordagens na indústria. O primeiro estudo de caso também foi utilizado para detalhar a aplicação de cada uma das abordagens. Ambos os estudos de casos produziram resultados semelhantes aos dos estudos experimentais controlados. Vale destacar que os três experimentos controlados e o primeiro estudo de caso já foram publicados na literatura, compondo quatro artigos científicos deste autor.

O próximo capítulo mostra um estudo de caso real com o intuito de mostrar os benefícios de ter, no processo de desenvolvimento de software, a RTM determinada com os níveis de dependência (“sem dependência”, “dependência fraca”, “dependência forte”). Dentre os benefícios, destacam-se a seleção de um conjunto de testes de regressão com priorização e a caracterização do impacto de uma mudança.

Capítulo 6

BENEFÍCIOS DA RTM COM NÍVEIS DE DEPENDÊNCIA NO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

Este capítulo apresenta um estudo de caso para mostrar os benefícios de se estabelecerem níveis de dependência entre os requisitos funcionais na determinação da RTM. Esse estudo de caso teve como objetivo mostrar que utilizando-se os níveis de dependência entre os requisitos funcionais é possível selecionar um conjunto de testes de regressão com priorização, assim como caracterizar o impacto de uma mudança no software.

6.1 Considerações iniciais

O objetivo desta tese foi mostrar que o uso de técnicas de inteligência computacional melhora a acurácia dos links de rastreabilidade. No Capítulo 4 foram apresentadas as quatro abordagens aqui propostas e como elas foram definidas e refinadas com o uso de inteligência computacional. No Capítulo 5 foram apresentados os experimentos que avaliaram as abordagens. Também foram apresentados os estudos de casos reais que foram conduzidos para mostrar que as abordagens também podem ser utilizadas na indústria. Os resultados detalhados no Capítulo 5 permitiram observar que combinar as abordagens de geração da RTM com técnicas de inteligência computacional melhora a detecção dos links de rastreabilidade, conforme tese definida neste trabalho.

Uma das características das abordagens propostas é que elas definem não somente se existe ou não dependência entre os requisitos funcionais, mas determinam também o nível dessa dependência. A motivação deste capítulo é mostrar que a contribuição deste trabalho não passa somente pela determinação da RTM, mas que o fato de se estabelecer a RTM com os níveis de dependência (“sem dependência”, “dependência fraca” e “dependência forte”) traz alguns benefícios para o processo de desenvolvimento de software. Um exemplo direto de se estabelecerem os níveis de dependência entre os requisitos funcionais é a determinação de um conjunto de testes de regressão com priorização dos casos de testes e a possibilidade de caracterização do impacto de mudanças nos requisitos funcionais.

Para mostrar esses benefícios para o processo de desenvolvimento de software, faz-se uma breve revisão sobre testes de regressão e priorização de casos de testes. Na sequência é detalhada uma abordagem proposta para se considerar os níveis de dependência da RTM na definição do conjunto de testes de regressão com priorização e para a caracterização do impacto de uma mudança. Um estudo de caso que foi conduzido com base nesta abordagem também é apresentado. Vale destacar que, para poder conduzir o estudo de caso, também foi utilizado o módulo de gestão de Casos de Testes (CTs) disponibilizado na ferramenta COCAR.

6.2 Priorização de Casos de Teste e Teste de Regressão

O processo para se verificar as modificações de um software durante a fase de manutenção é chamado Teste de Regressão (MAHESWARI e JEYAMALA, 2013). O teste de regressão deve ser executado após o software sofrer alguma modificação, seja pela inserção de um novo requisito, ou seja pela modificação de requisito existente, a fim de validar se essas alterações não inseriram novos erros e prejudicaram a qualidade do produto. À medida que o software é desenvolvido, o conjunto de teste de regressão aumenta; porém, muitas vezes a viabilidade do cronograma não permite a execução exaustiva de todos os CTs (KUKREJA *et al.*, 2013). O teste de regressão pode ser uma das etapas mais caras da atividade de manutenção, já que uma modificação no software pode exigir que todo o conjunto de

testes seja executado novamente. Se esse conjunto de testes for muito grande, o esforço a ser dedicado nessa execução também será grande.

O teste de regressão também deve ser planejado para identificar como, quando e por quem deverá ser executado (MYERS, 2004). O planejamento permite que essa atividade seja executada com maior eficiência. Pesquisadores têm buscado técnicas para a redução dos custos do teste de regressão, incluindo seleção de casos de teste e técnicas de minimização do conjunto de testes de regressão (ROTHERMEL *et al.*, 2001). A priorização de CTs é uma técnica que pode auxiliar na identificação de um subconjunto de teste que consumirá menos tempo e recursos durante o teste de regressão. A priorização de CTs auxilia a equipe na execução do teste de regressão a fim de se alcançar mais rapidamente os objetivos do teste no processo de teste (MAHESWARI e JEYAMALA, 2013). Essa técnica permite que a equipe ordene os casos de teste de acordo com sua prioridade, seguindo algum critério de seleção. Assim, os testes com maior prioridade serão executados antes que os de baixa prioridade (ROTHERMEL *et al.*, 2001). Isso pode aumentar a probabilidade de os defeitos serem revelados mais brevemente durante o processo de teste. Revelar antes os defeitos permite que o software possa ser corrigido mais rapidamente, diminuindo assim o tempo de entrega do produto (MYERS, 2004). A priorização dos CTs deve ser feita seguindo algum critério que permita identificar por que um caso de teste terá mais prioridade que outro. Esse critério deve estar alinhado com o objetivo que se espera alcançar ao diminuir o conjunto de testes de regressão: redução de custos, aumento da taxa de detecção de defeitos, redução de tempo de entrega do produto, entre outros.

A seguir apresenta-se a abordagem proposta neste trabalho, que utiliza o nível de dependência entre os requisitos funcionais como um critério para a priorização de casos de teste e que faz a seleção dos CTs de regressão.

6.3 Abordagem Proposta

Nesta seção é apresentada a abordagem proposta para tratar as modificações nos requisitos funcionais durante o processo de desenvolvimento de software. É

importante destacar que, quando esta tese de doutorado menciona “modificação nos requisitos” refere-se tanto à modificação de alguma funcionalidade existente e já identificada, quanto à inserção de um novo requisito funcional. O objetivo da proposta é que, a partir de qualquer modificação em um requisito funcional, ou em um conjunto de requisitos funcionais, seja possível selecionar um conjunto de CTs de regressão, por meio de priorização, além de se caracterizar a complexidade de mudanças nos requisitos funcionais. Para a definição da abordagem estabeleceu-se como base os níveis de dependência entre os requisitos funcionais definidos durante a geração da RTM, pelas abordagens propostas nesta tese.

Assim, uma vez que os requisitos funcionais do sistema estejam especificados e relacionados com seus respectivos CTs, os testes de regressão decorrentes de toda modificação/inserção de um requisito funcional devem ser selecionados e priorizados. Conforme mencionado anteriormente, para dar suporte à abordagem proposta, foi utilizada a ferramenta COCAR que, além das funcionalidades relacionadas com os requisitos e com a rastreabilidade, permite também o cadastro de CTs para cada requisito funcional.

Para facilitar o entendimento da abordagem, considere um software que foi desenvolvido e que já tem seus requisitos funcionais e CTs cadastrados na ferramenta COCAR. Quando for solicitada qualquer modificação no software será necessário registrar as modificações nos requisitos funcionais, implementar as modificações e testá-las. A Figura 6-1 apresenta o fluxograma da abordagem aqui proposta. Atenta-se para o fato de que o resultado da abordagem é um conjunto de testes de regressão com as prioridades de execução estabelecidas. Isso é gerado de acordo com o nível de dependência entre os requisitos funcionais na RTM. Quanto mais alta a prioridade do CT, mais chances ele terá de encontrar defeitos e, por isso, deverá ser executado primeiro (ROTHERMEL *et al.*, 2001).

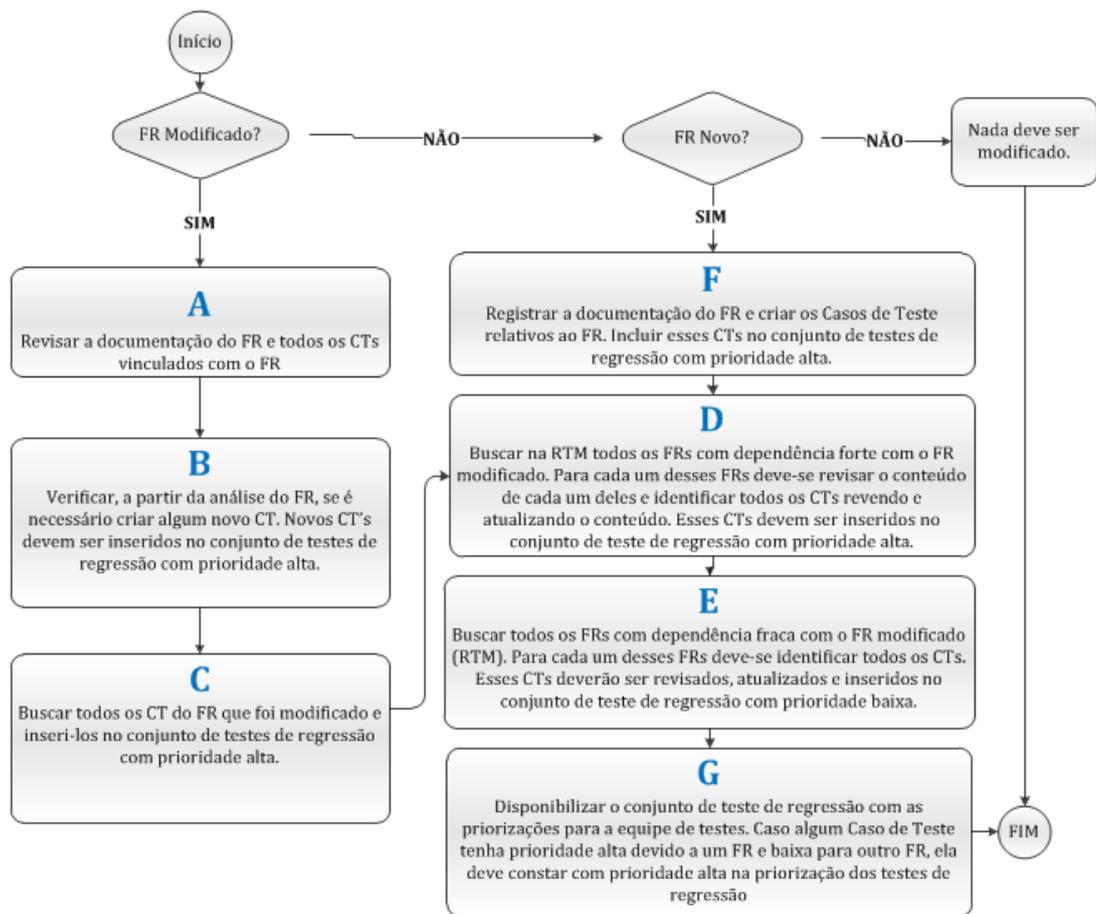


Figura 6-1. Etapas da abordagem para seleção de CTs priorizados com base na dependência dos FRs

Além de definir os CTs prioritários, essa abordagem permite caracterizar o impacto das mudanças no software. Sendo assim, antes da implementação das modificações por parte da equipe de desenvolvimento, para a posterior aplicação dos testes de regressão, pode ser interessante para os envolvidos no projeto conhecer e dimensionar a complexidade relacionada em cada uma das mudanças. Com base nessa informação podem ser tomadas ações para a gestão de risco, como a criação de planos de contingência para mudanças de grande complexidade, ou mesmo para a decisão do melhor momento em que mudanças mais complexas devam ser implementadas. Também pode ser possível alocar membros mais experientes da equipe para lidar com os problemas mais complexos.

Destaca-se que não é objetivo da abordagem definir uma métrica para mensurar o impacto das mudanças relacionada com esforço em homens-hora ou pontos por função. O intuito é apresentar um comparativo entre as mudanças, para que se possa caracterizar dentre todas elas, quais representam maior risco

justamente por possuírem maior dependência com outros requisitos funcionais. Conforme visto anteriormente, na abordagem proposta a definição dos testes de regressão está baseada na dependência entre os requisitos funcionais e entre esses com os CTs. Sendo assim, utilizaram-se as mesmas dependências para caracterizar a complexidade das mudanças, assumindo-se que, quanto maior a dependência entre os requisitos funcionais, mais difícil deve ser sua manutenção.

Se um requisito funcional tem dependência com vários outros, modificar esse requisito funcional implica em ajustar modelos que tenham dependência forte com outros modelos e trechos de código-fonte que também tenham dependência forte com outras funcionalidades. Isso implica, muitas vezes, em ser necessário ter o entendimento de outros trechos de código para que se possa fazer a mudança no requisito funcional em questão. Em geral, o esforço em compreender código-fonte de terceiros envolve muito tempo e atenção por parte dos desenvolvedores. Como essa é uma possível fonte de defeitos, o esforço na etapa de testes e correção poderá ser maior. Por outro lado, modificar ou inserir um requisito que tenha dependência fraca (ou nenhuma dependência) com outros requisitos funcionais já implementados, não apresenta a dificuldade de ajustar a implementação com outras funcionalidades já criadas. Para caracterizar o impacto da mudança, utilizou-se a quantidade de CTs que devem ser executados para cada modificação, considerando suas respectivas prioridades. Considerou-se que, se um requisito funcional possui muitos CTs, isso pode indicar que aquele único requisito funcional possui uma regra de negócio complexa. Por outro lado, caso um requisito funcional tenha apenas um CT relacionado, isso pode indicar que sua regra de negócio seja menos complexa.

Assim, para caracterizar o impacto da mudança, levou-se em consideração a prioridade dos CTs estabelecendo-se peso 3 para prioridade alta e peso 1 para prioridade baixa. Observa-se que, ao considerar a prioridade do CT, indiretamente está se considerando o nível de dependência dos RFs. A Equação 10 mostra como é feito o cálculo dessa complexidade. O valor de peso 3 para ponderar a dependência forte em relação ao peso 1 para dependência fraca foi assumido por considerar-se que implementar uma mudança complexa pode ser pelo menos três vezes mais difícil que uma mudança simples. Entende-se que esse valor possa ser ajustado com base em outros dados que estejam sendo coletados pelo gerente do projeto. Ressalta-se

que, no estudo real utilizado para definir esses valores, eles foram consenso na equipe de desenvolvimento.

$$C_{RF9} = (\text{qtdPrAlta} * 3) + (\text{qtdPrBaixa}) \quad (10)$$

Nessa equação, qtdPrAlta é a quantidade de CTs com prioridade alta identificados pela abordagem para a mudança no Requisito Funcional X (RFx), e qtdPRBaixa indica a quantidade de CTs com prioridade baixa identificados pela abordagem para a mudança no RFx. A Figura 6-2 resume a abordagem para a caracterização da complexidade.

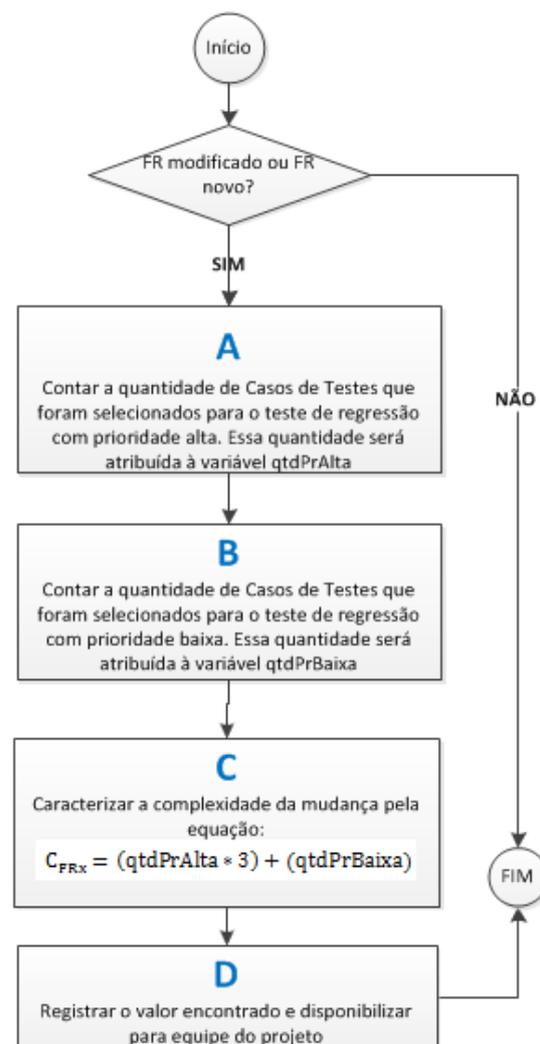


Figura 6-2. Abordagem para caracterização da complexidade das mudanças nos requisitos funcionais.

Vale destacar que tanto a seleção dos casos de testes com priorização quanto caracterização do impacto das mudanças como proposto só foram possíveis porque

a RTM gerada pelas abordagens propostas define níveis para a dependência (“sem dependência”, “dependência fraca”, “dependência forte”). Na próxima seção é apresentado o estudo de caso realizado com um sistema real, cuja evolução e consequente alteração nos requisitos funcionais seguiram a abordagem proposta.

6.4 Estudo de Caso

Este estudo de caso foi conduzido para avaliar a seleção do conjunto de testes de regressão com priorização baseado no nível de dependência entre os requisitos funcionais e na rastreabilidade entre requisito funcional e CT. O estudo de caso também caracteriza a complexidade para a modificação de um requisito funcional, utilizando-se também do nível de dependência entre os requisitos funcionais (definidos na RTM) e os testes de regressão que foram determinados para testar a mudança requerida.

O software objeto de estudo foi desenvolvido para avaliar atividades físicas de pacientes submetidos a esforços em teste de degrau e de esteira. O software fez parte de um projeto do Departamento de Fisioterapia da UFSCar e está vinculado com a patente de um novo produto. Durante a evolução desse produto, diversas modificações têm sido necessárias, tanto no hardware que faz a coleta dos dados, quanto no software objeto deste estudo. Nesse cenário iterativo e interativo de mudanças, a habilidade em lidar com as modificações dos requisitos auxiliou o processo de desenvolvimento do produto. Quando se fala em seleção de CTs com priorização, refere-se à capacidade de definir a prioridade de cada CT que permite identificar mais defeitos e de forma mais rápida, além de cobrir as mudanças que ocorreram no software.

Os passos para a realização do estudo de caso foram:

1. Registro dos requisitos funcionais do software na ferramenta COCAR:

Foi feito o levantamento de requisitos com as pessoas que seriam usuários do sistema. Todos os requisitos foram cadastrados na ferramenta COCAR,

registrando-se descrição, processamento, dados de entrada, restrições, saídas e stakeholders envolvidos. Cadastrou-se um total de 30 requisitos funcionais.

2. Geração da RTM de acordo com a abordagem RTM-Fuzzy

A Figura 6.4.1 ilustra a RTM gerada. Em verde são indicadas as dependências fracas entre os requisitos funcionais e, em vermelho, as dependências fortes entre os requisitos funcionais. A abordagem RTM-Fuzzy foi utilizada.



Figura 6-3. RTM gerada a partir da RTM-Fuzzy

3. Inserção dos CTs na Ferramenta COCAR

A partir da análise de cada requisito funcional foram criados CTs funcionais para os requisitos. Como os CTs foram criados a partir dos requisitos funcionais, a rastreabilidade entre requisito funcional e CTs foi estabelecida automaticamente na ferramenta. Essa rastreabilidade é necessária para a identificação da prioridade dos casos testes. Sendo assim, segundo a abordagem proposta, se um requisito funcional for modificado então é necessário que todos os CTs vinculados a esse RF sejam executados primeiramente na fase de testes de regressão, conforme é detalhado na sequência. Foram criados um total de 106 CTs.

4. Desenvolvimento do Software objeto do estudo

O desenvolvimento envolveu dois programadores e foi feito com linguagem C# para ambiente desktop.

5. Aplicação dos CTs para encontrar defeitos no software desenvolvido.

Depois que o software foi implementado todos os CTs produzidos na etapa 3 foram executados. Os defeitos encontrados foram corrigidos e os CTs que encontraram esses defeitos foram executados novamente para que a primeira versão do software fosse considerada estável. Na execução dos 106 CTs foram gastos 160 minutos aproximadamente.

Depois dessas etapas o software foi entregue para o Departamento de Fisioterapia para que fosse realizado o teste de aceitação. O software foi utilizado no dia-a-dia para avaliação de pacientes submetidos a testes de esforço físico. Esperava-se que, após a primeira execução do teste de aceitação seriam feitas propostas de melhorias no software: tanto no que diz respeito à mudança nos requisitos funcionais existentes quanto na inserção de novos requisitos funcionais. Além disso, esperava-se também que pudessem ser encontrados defeitos que não foram detectados na primeira etapa de testes.

6. Tratamento de mudanças

Após duas semanas de uso do software, a equipe se reuniu com os usuários para a avaliação do software. Foram feitas as seguintes solicitações de modificação:

- Modificação no RF9, relacionado com o cadastro de paciente, inserindo um novo campo.
- Inserção de um novo requisito funcional, relacionado com a inclusão do preenchimento de um questionário pelos pacientes (será denominado RF31);
- Inserção de um novo requisito funcional, relacionado com a criação de backups do sistema (será denominado RF32).

As etapas utilizadas para determinar a implementação das modificações estão descritas na Figura 6-1. A seguir apresenta-se uma descrição detalhada de cada etapa realizada neste estudo de caso.

Sendo assim, para tratar a modificação no RF9 foi necessário revisar sua documentação e todos os CTs relacionados a esse requisito (passo A da Figura 6-1). Quando um RF é modificado, os CTs relacionados a ele devem ser revisados

e atualizados para que continuem de acordo com o requisito funcional. Depois foi feita uma análise para verificar se seria necessário criar algum novo CT, o que não ocorreu (passo B da Figura 6-1).

Na sequência todos os CTs relacionados com o RF9 foram identificados, sendo selecionados os CTs CT34, CT35, CT36 e CT37 (Figura 6-4-A). Esses quatro CTs foram inseridos no universo de testes de regressão com prioridade alta, completando assim o passo C da Figura 6-1.

O passo seguinte foi identificar todos os requisitos funcionais que possuem dependência forte (segundo a abordagem RTM-Fuzzy) com o RF9. O único RF que possuía relacionamento forte era o RF10. Este requisito está relacionado com a atualização de um paciente no sistema. Esse relacionamento é mostrado pela seta em vermelho na Figura 6-4. Todos os CTs do RF10 foram inseridos no universo de testes de regressão com prioridade alta (CT38, CT39 e CT40). Foi necessário criar um novo CT, denominado CT107, para que o novo campo inserido pudesse ser validado. Este CT também foi inserido no universo de testes de regressão com prioridade alta. Completou-se assim o passo 4 da Figura 6-1, e representados pela parte B da Figura 6-4.

Por fim, foram recuperados os requisitos funcionais que têm dependência fraca com o RF9 (RF5, RF6, RF7 e RF8) marcados com a seta em verde na Figura 6-4 e seus respectivos CTs (CT19 até CT33), conforme descrito no passo E da Figura 6-1. Esses CTs foram colocados no universo de testes de regressão com prioridade baixa, conforme Figura 6-4-C.

Sendo assim, o último passo da abordagem proposta seria disponibilizar para a equipe de testes os CTs de CT34 a CT40 com prioridade alta e os CTs de CT23 a CT33 com prioridade baixa (passo F da Figura 6-1).

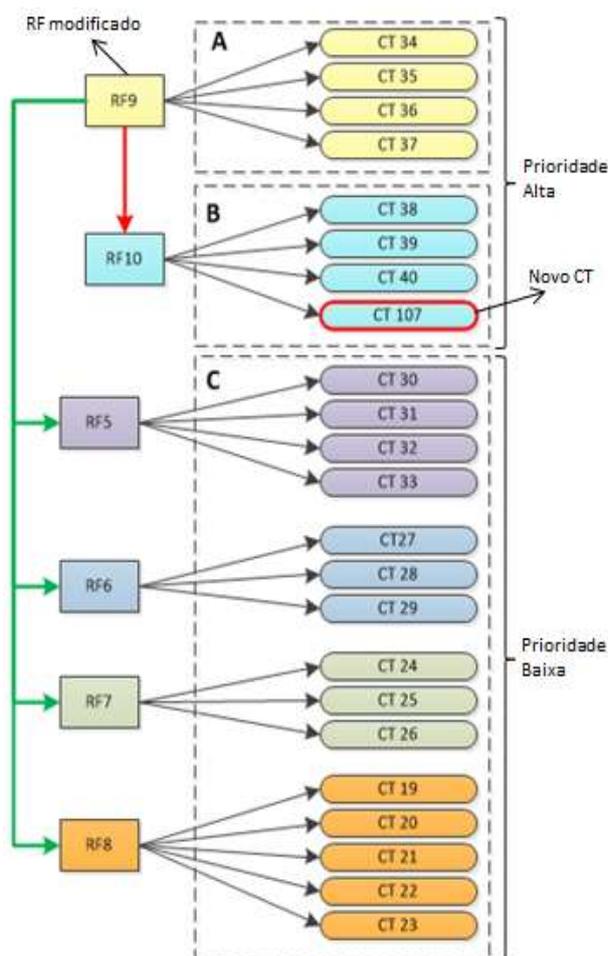


Figura 6-4. Modificação no FR9

Outra alteração no conjunto de requisitos foi a inserção do RF31, relacionado com o preenchimento de um questionário pelos pacientes que realizaram a avaliação. Após a inserção desse requisito foram criados os CTs para essa funcionalidade (CT108, CT109 e CT110). Eles foram inseridos no universo dos testes de regressão com prioridade alta. (Passo F da Figura 6- 1 representado pela Figura 6-5-A).

Seguindo a abordagem proposta, buscou-se na RTM gerada pela abordagem RTM-Fuzzy todos os requisitos funcionais com dependência forte com relação ao RF31. Nesse caso, não houve nenhum RF nessa situação (Passo D da Figura 6-1).

O próximo passo (Passo E da Figura 6-1) foi identificar quais requisitos funcionais possuem dependência fraca com o RF que foi inserido (RF31). No caso, foram recuperados os requisitos RF9, RF10, RF11, RF12, RF25 e RF26. Essas

dependências fracas são marcadas na Figura 6-5 pelas setas verdes. Os CTs relacionados com esses seis requisitos funcionais foram recuperados e inseridos no conjunto de CTs de regressão com prioridade baixa. Esses CTs recuperados a partir da inserção do RF31 foram disponibilizados para a equipe de testes (Passo G da Figura 6-1).

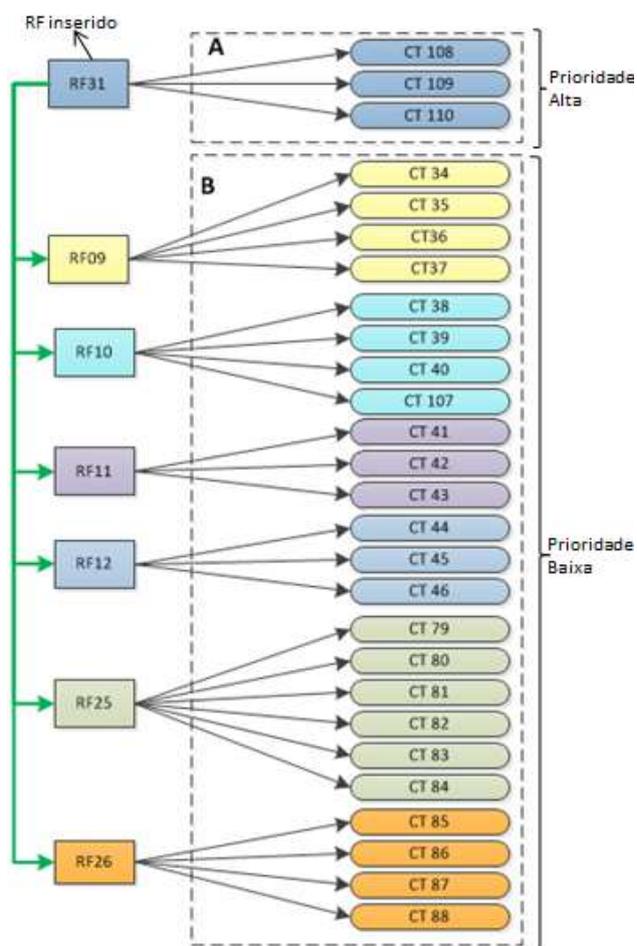


Figura 6-5. Inserção do FR31

Por fim, detalha-se a inserção do RF32, relacionado com a criação e importação de backups do sistema. Como se trata de um novo RF, a primeira ação é a criação dos CTs para esse RF (passo F da Figura 6-1 representado na Figura 6-6). O próximo passo foi buscar os requisitos funcionais com dependência forte para então selecionar os CTs vinculados a estes (passo D da Figura 6-1). Não foram encontrados requisitos funcionais com dependência forte nem com dependência fraca (passo E da Figura 6-1). Sendo assim, somente os CTs criados

especificamente para tratar esse requisito (RF32) foram inseridos no conjunto de CTs de regressão.

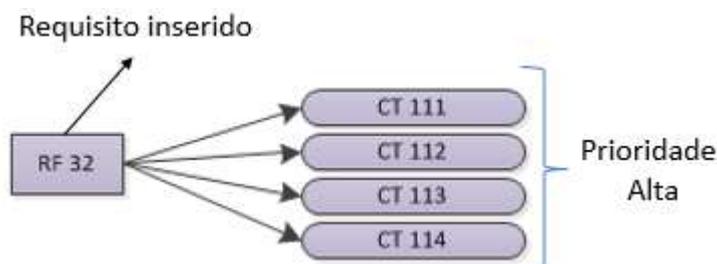


Figura 6-6. Inserção do RF32

O resultado da execução dos passos descritos na Figura 6-1 foi um conjunto de testes de regressão. Vale destacar que os CTs CT34, CT35, CT36 e CT37 foram indicados com prioridade alta quando o RF9 foi modificado e prioridade baixa quando o RF31 foi inserido. Nessas situações, sempre deve ser considerada a maior prioridade, conforme detalhado no passo G da Figura 6-1.

7. Caracterização do impacto das mudanças

Para exemplificar a caracterização da complexidade definida na abordagem, construiu-se a Tabela 13 com os dados do estudo de caso. Utilizou-se a Equação 10 para caracterizar a complexidade de cada alteração nos requisitos funcionais. Na sequência (Equação 11) um exemplo de como foi feito o cálculo para a modificação no RF9:

$$C_{RF9} = 8 * 3 + 15 = 39 \quad (11)$$

O mesmo procedimento foi utilizado para caracterizar a complexidade dos requisitos RF31 e RF32. Dessa forma, é possível perceber que a mudança nos requisitos RF9 e RF31 tem impacto maior que a do RF32. Essa informação pode ajudar o gerente do projeto a alocar as pessoas mais experientes para as tarefas de maior complexidade ou, caso existam várias mudanças com grande complexidade, ele poderá estimar qual será o tempo necessário para que um novo *release* do software seja feito. Com o tempo e conhecimento da empresa, sob seu próprio

processo, é possível traçar a relação do valor absoluto da complexidade com esforços em homem-hora ou pontos por função.

Tabela 13. Complexidade das modificações

RF	CTs do RF	CTs que devem ser executados com prioridade alta	CTs que devem ser executados com prioridade baixa	Quantidade de CTs com prioridade alta (qtdPrAlta)	Quantidade de CTs com prioridade baixa (qtdPrBaixa)	Caracterização da complexidade para modificar o RF
RF9	CTCT34, CT35, CT36, CT37	CT34, CT35, CT36, CT37, CT38, CT39, CT40, CT107	CT19, CT20, CT21, CT22, CT23, CT24, CT25, CT26, CT27, CT28, CT29, CT30	8	15	39
F31	CT108, CT109, CT110	CT108, CT109, CT110	CT34, CT35, CT36, CT37, CT38, CT39, CT40, CT107, CT41, CT42, CT43, CT44, CT45, CT46, CT79, CT80, CT81, CT82, CT83, CT84, CT85, CT86, CT87, CT88	3	25	34
F32	CT111, CT112, CT113	CT111, CT112, CT113, CT114	-	4	0	12

8. Implementação das modificações no software

A modificação no RF9 e a inserção do RF31 e do RF32 foram implementadas pela equipe de desenvolvimento.

9. Aplicação dos testes de regressão para detecção de defeitos no software

Nesta etapa, o conjunto de testes de regressão, com os CTs priorizados, foi aplicado no software implementado no passo anterior (passo 8). Esse conjunto de CTs possuía duas categorias:

- 15 CTs com prioridade alta: CT34, CT35, CT36, CT37, CT38, CT39, CT40, CT107, CT108, CT109, CT110, CT111, CT112, CT113 e CT114;

- 28 CTs com prioridade baixa: CT19, CT20, CT21, CT22, CT23, CT24, CT25, CT26, CT27, CT28, CT29, CT30, CT41, CT42, CT43, CT44, CT45, CT46, CT79, CT80, CT81, CT82, CT83, CT84, CT85, CT86, CT87 e CT88.

A execução dos testes de Prioridade Alta teve os seguintes resultados: 6 casos falharam e 9 passaram. Os CTs que falharam foram: CT37, CT40, CT107, CT112, CT113 e CT114. Gastaram-se aproximadamente 20 minutos para a execução desses 15 testes. Após a execução dos testes de Prioridade Baixa, os seguintes CTs falharam: CT42, CT44, CT84. Para a execução desses 28 testes, foram necessários aproximadamente 35 minutos.

Com o objetivo de medir a diferença entre a execução de todos os 114 CTs sem as priorizações repetiu-se o processo de teste com a mesma versão do software construído no Passo 8. Os CTs foram executados na ordem crescente (do CT01 até o CT114). Gastou-se para isso aproximadamente 160 minutos e exatamente os mesmos 9 testes que falharam na etapa anterior falharam nessa execução. A Tabela 14 resume os dados sobre as execuções de teste

Tabela 14. Dados do estudo de caso

		Quantidade de CTs		Tempo (min)		Defeitos encontrados	
Teste de Regressão	Prioridade Alta	15	40	20	55	6	9
	Prioridade Baixa	25		35		3	
Teste de todos os CTs		114		160		9	

10. Avaliar a efetividade dos testes de regressão

Com base nos dados da Tabela 14, pode-se verificar que:

- O tempo para realizar os testes de regressão foi, aproximadamente, três vezes menor que o tempo utilizado para realizar todos os testes (50 minutos contra 160);
- Todos os defeitos encontrados durante a execução de todo o conjunto de testes também foram encontrados no conjunto de testes de regressão com prioridade;
- Os CTs de prioridade alta foram os que mais defeitos encontraram quando comparado aos CTs de prioridade baixa (6 defeitos contra 3).

Os defeitos mais graves foram encontrados primeiro, o que comprova que a priorização de testes de fato aumenta a taxa de detecção de defeitos. Isso permite que os defeitos mais graves, identificados primeiro, possam ser corrigidos prioritariamente. Em um projeto de teste, em que o teste de regressão possua um tempo ou orçamento limitados, essa priorização ajuda a garantir a qualidade do produto após uma modificação, mesmo que não haja tempo suficiente para a execução de todo o conjunto de testes durante o teste de regressão.

Como ameaças à validade do estudo de caso pode ser destacado o fato de que a equipe de testes não possuía grande experiência prática em testes de software. A equipe de testes recebeu um breve treinamento conceitual antes do início das atividades. Destaca-se ainda o fato de que, apesar de gerar um produto para o mercado, haviam poucos envolvidos no desenvolvimento do software: dois analistas, dois desenvolvedores e um testador.

6.5 Considerações Finais

Neste capítulo exemplificou-se benefícios do estabelecimento da RTM com níveis de dependência para o processo de desenvolvimento de software. Foi apresentada uma abordagem para tratar as modificações nos requisitos funcionais, que é capaz de selecionar um conjunto de testes com priorização, além da caracterização do impacto de uma mudança nos requisitos funcionais. Apesar de não ser o foco principal desta tese, pretendeu-se mostrar a aplicabilidade das abordagens no processo de desenvolvimento de software, evidenciando que benefícios podem ser alcançados pelo fato de se estabelecer a RTM, principalmente diferenciando os níveis de dependências existentes entre os requisitos. Vale destacar ainda que está sendo conduzido um novo estudo de caso com a empresa de aviação na qual foi conduzido o estudo de caso detalhado na Seção 5.2.

Ressalta-se também que existem duas revisões sistemáticas que tratam da seleção de casos de testes (CTs) para testes de regressão. Nelas [(SINGH *et al.*, 2012), (ENGSTRÖM *et al.*, 2010)] não foram encontradas iniciativas que gerem o

conjunto de Testes de Regressão com priorização a partir da matriz de rastreabilidade de requisitos funcionais e dos níveis de dependência, como proposto neste trabalho. Nesses dois trabalhos encontrados na literatura são listadas as técnicas para geração de Teste de Regressão com priorização. Uma das técnicas cita a rastreabilidade (FILHO *et al.*, 2010); porém, essa rastreabilidade é entre modelos, código-fonte e CTs. Outra técnica citada (SRIKANTH *et al.*, 2005) tem como foco os requisitos, mas a definição e priorização dos casos de testes não são baseadas na rastreabilidade. As outras técnicas citadas nos trabalhos não têm foco em requisitos.

Destacados os benefícios da geração da RTM com níveis de dependência e, conseqüentemente, desta proposta, no próximo capítulo sumarizam-se as principais contribuições e limitações deste trabalho, bem como as possibilidades de novas investigações.

Capítulo 7

CONCLUSÃO

Este capítulo apresenta as conclusões, contribuições e limitações da tese apresentada, bem como lições aprendidas, oportunidades de pesquisa identificadas e as publicações obtidas até o momento.

7.1 Conclusões

Esta pesquisa apresentou **um conjunto de abordagens para a geração da matriz de rastreabilidade de requisitos com suporte de técnicas de inteligência computacional** e os estudos realizados para corroborar a tese de que **o uso de técnicas de inteligência computacional melhora a acurácia dos links de rastreabilidade entre requisitos**.

Inicialmente, propôs-se uma técnica baseada em trabalhos da literatura que envolve o processamento de linguagem natural (RTM-NLP) e outra baseada em proposta original com uso dos dados de entrada dos requisitos (RTM-E). Após a avaliação das duas abordagens, foram feitas duas outras propostas que utilizaram as anteriores como entradas: uma delas é baseada em sistemas fuzzy (RTM-Fuzzy), e a outra, em redes neurais (RTM-N). Após a execução de experimentos controlados (detalhados no Capítulo 5), constatou-se que essas duas últimas apresentaram melhor acurácia na detecção da rastreabilidade se comparadas com as duas primeiras, obtendo então resultados positivos sobre a premissa deste trabalho. Depois de se definirem as abordagens, foram feitas propostas de melhorias em todas elas, também com o uso de técnicas de inteligência computacional (árvores de decisão e

algoritmos genéticos). Um novo experimento conduzido mostrou melhora na acurácia dos links de rastreabilidade.

Também foram conduzidos dois estudos de casos reais para mostrar a possibilidade de aplicação das abordagens na indústria. O primeiro estudo de caso foi conduzido em um sistema de gestão de estoque de uma empresa de aviação. O segundo foi conduzido em um projeto de software livre da Fundação Telefônica que informatiza os dados de crianças e jovens em programas sociais. Ambos os estudos mostraram resultados próximos aos encontrados nos estudos experimentais controlados.

Considerando que as abordagens propostas definem não somente se existe ou não link de rastreabilidade entre dois requisitos funcionais, mas também o nível dessa dependência (“sem dependência”, “dependência fraca” e “dependência forte”), foi conduzido um estudo de caso para mostrar os benefícios da geração da RTM com esses níveis de dependência. Esse estudo foi detalhado no Capítulo 6 e complementa a contribuição desta pesquisa para outras etapas do desenvolvimento de software, mostrando que é possível fazer a seleção de casos de teste de regressão com priorização, bem como a caracterização do impacto de mudanças nos requisitos, uma vez que os níveis de dependência na RTM sejam distinguidos.

7.2 Contribuições da tese

As contribuições da tese são detalhadas na sequência:

- Definição de quatro abordagens diferentes para estabelecer a dependência entre requisitos funcionais na RTM. As abordagens foram apresentadas em eventos relevantes da área.
- Refinamento das abordagens por meio de processo de experimentação. Os resultados obtidos nos experimentos motivaram melhorias em cada uma das abordagens.
- Uso de inteligência computacional para tornar a detecção de links de rastreabilidade menos dependente de limites rígidos. O uso de sistemas

fuzzy e redes neurais minimizam a utilização de valores fixos em escalas pré-definidas.

- Utilização de níveis de dependência nos links de rastreabilidade. Ao invés de determinar se existe ou não dependência, as abordagens propostas classificam a dependência como forte ou fraca.
- Seleção de Teste de regressão com priorização: Foi feita a proposta de uma abordagem que, a partir de uma modificação nos requisitos funcionais, possibilita a seleção de um conjunto de casos de testes com priorização. Essa proposta só foi possível pelo fato das abordagens definirem o nível de dependência e não somente se existe ou não dependência.
- Caracterização do impacto de uma mudança: foi definida uma proposta que, a partir de uma modificação nos requisitos funcionais, possibilita caracterização do impacto da mudança ao longo do processo de desenvolvimento de software. Essa proposta também utilizou o nível de dependência entre os requisitos que é estabelecido na RTM pelas abordagens propostas.
- Desenvolvimento da Ferramenta COCAR que possibilita a aplicação das abordagens propostas no trabalho. Esta ferramenta possui módulos para:
 - Gestão de Requisitos;
 - Geração da RTM de acordo com as quatro abordagens propostas nesta tese;
 - Visualização de Dependência entre requisitos funcionais;
 - Gestão de Casos de Testes;
 - Módulo para realização de estudos experimentais.

7.3 Limitações do trabalho

Apesar das contribuições destacadas anteriormente, algumas limitações são inerentes e são identificadas na sequência:

- 1) Os experimentos foram conduzidos em ambiente acadêmico, e isso implica em uma ameaça à validade dos resultados, como mencionado no momento de sua apresentação;
- 2) Não foram explorados requisitos funcionais com características diferentes dos sistemas de informação com cadastros, atualizações e relatórios.
- 3) Os resultados obtidos nos experimentos não podem ser generalizados para outros contextos.
- 4) As regras de inferência do sistema fuzzy não foram refinadas.
- 5) A ferramenta COCAR, que tornou viável a utilização das abordagens propostas e a realização dos experimentos, ainda necessita de melhorias para se tornar mais estável;
- 6) Nas abordagens propostas, não são considerados os requisitos não funcionais.

Outros resultados desta pesquisa como publicações, orientações de iniciação científica e participação em outros projetos deste grupo de pesquisa são apresentados a seguir.

7.4 Publicações

Nessa seção são apresentadas as publicações do autor desta tese durante o período de doutorado. As publicações estão organizadas em três subseções: artigos completos publicados em anais de congressos (*full papers*), artigos completos publicados em periódicos e capítulos de livros.

7.4.1.1 Trabalhos completos publicados em anais de congressos

- 1) **DI THOMMAZO, A.** ; **ROVINA, R;** **RIBEIRO, T.** ; **OLIVATO, G.** ; **HERNADES, E.** ; **WERNECK, V. M. B.** ; **FABBRI, S. C. P. F.** . *Using artificial*

intelligence techniques to enhance traceability links. In: International Conference on Enterprise Information Systems, 2014, Lisboa. International Conference on Enterprise Information Systems, 2014. v. 1. p. 1-10

Esse artigo apresentou as melhorias feitas nas abordagens com uso de técnicas de inteligência computacional. Também mostrou os Experimento 3 (Seção 5.3.3).

- 2) **DI THOMMAZO, A. ; RIBEIRO, T. ; OLIVATO, G. ; ROVINA, R. ; Werneck, V. M. B. ; FABBRI, S. C. P. F. .** *Detecting traceability links through neural networks. In: International Conference on Software Engineering and Knowledge Engineering, 2013, Boston. Proceedings of the Twenty-Fifth International Conference on Software Engineering & Knowledge Engineering, 2013. v. 1. p. 36-41.*

Esse artigo apresentou a abordagem RTM-N e parte do Experimento 2 (Seção 5.3.2).

- 3) **DI THOMMAZO, A. ; RIBEIRO, T. ; OLIVATO, G. ; Werneck, V. M. B. ; FABBRI, S. C. P. F. .** *An automatic approach to detect traceability links using fuzzy logic. In: Simpósio Brasileiro de Engenharia de Software, 2013, Brasília. Anais do XXVII Simpósio Brasileiro de Engenharia de Software, 2013. v. 1. p. 36-45.*

Esse artigo apresentou a abordagem RTM-Fuzzy e parte do Experimento 2 (Seção 5.3.2).

- 4) **DI THOMMAZO, A. ; MALIMPENSA, G. ; OLIVATTO, G. ; FABBRI, S. C. P. F. .** *Geração Automática da Matriz de Rastreabilidade de Requisitos com suporte de Visualização. In: Simpósio Brasileiro de Engenharia de Software, 2012, Natal. Anais do XXVI Simpósio Brasileiro de Engenharia de Software, 2012. v. 1. p. 1-12.*

Esse artigo apresentou as abordagens RTM-E e RTM-NLP, o Experimento 1 (Seção 5.3.1) e o Estudo de Caso 1 (Seção 5.2).

- 5) OCTAVIANO, F. R. ; **DI THOMMAZO, A.** ; CAMARGO, K. G. ; FABBRI, S. C. P. F. . *Test Scenarios Generation Based on Use Cases*. In: *Ibero-American Conference on Software Engineering, 2012, Buenos Aires. Proceedings of XV Ibero-American Conference on Software Engineering, 2012. v. 1. p. 1-12.*

Esse artigo apresentou uma proposta para cenários de testes baseados em casos de uso. Foi escrito em parceria com colegas de laboratório e fruto da evolução do trabalho de mestrado do autor desta tese.

- 6) HERNANDES, E. C. M.; TEODORO, E.; **DI THOMMAZO, A.**, FABBRI, S. C. P. F. *Using visualization and text mining to improve qualitative*. In: *16th International Conference on Enterprise Information Systems, 2014, Lisboa. Proceedings of the 16th International Conference on Enterprise Information Systems. v. 1. p. 66-79.*

Esse artigo foi escrito em parceria com colegas de laboratório no contexto do desenvolvimento da ferramenta Insight da qual o autor desta tese participa. As técnicas de mineração de texto utilizadas nesta tese foram adaptadas para o contexto de análise qualitativa.

- 7) FABBRI, S. C. P. F. ; HERNANDES, E. C. M. ; **DI THOMMAZO, A.** ; BELGAMO, ANDERSON ; ZAMBONI, A. ; SILVA, CLEITON . *Managing literature reviews information through visualization*. In: *International Conference on Enterprise Information Systems, 2012, Wroclaw. International Conference on Enterprise Information Systems, 2012. v. 1. p. 1-10.*

Essa publicação foi fruto do envolvimento do autor desta tese com o projeto StArt. Nessa publicação são apresentadas as funcionalidades de Visualização da ferramenta StArt no processo de revisão da literatura.

O trabalho recebeu Premiação de melhor artigo (*Best Paper*)

- 8) ZAMBONI, A.; **DI THOMMAZO, A.**; HERNANDES, E.; FABBRI, S. C. P. F.. StArt: uma Ferramenta Computacional de Apoio à Revisão Sistemática. In: *Simpósio Brasileiro de Engenharia de Software, 2010, Slavador. CBSOft - Brazilian Conference on Software: Theory and Practice. Salvador, 2010. v. 1.*

Essa publicação também foi fruto do envolvimento do autor desta tese com o projeto StArt. Nessa publicação são apresentadas as funcionalidades de ferramenta StArt.

- 9) HERNANDES, E. C. M.; ZAMBONI, A. ; **DI THOMMAZO, A.** ; FABBRI, S. C. P. F. . Avaliação da ferramenta StArt utilizando o modelo TAM e o paradigma GQM. In: *Experimental Software Engineering Latin American Workshop, 2010, Goiânia. Experimental Software Engineering Latin American Workshop, 2010. v. 1.*

Essa publicação também foi fruto do envolvimento do autor desta tese com o projeto StArt. Nessa publicação a ferramenta StArt foi avaliada para proposta de melhorias em suas funcionalidades.

7.4.1.2 Artigos completos publicados em periódicos

- 1) HERNANDES, E. C. M.; ZAMBONI, A.; FABBRI, S. C. P. F.; **DI THOMMAZO, A.** Using GQM and TAM to evaluate StArt a tool that supports Systematic Review. *CLEI Electronic Journal*, v. 15, p. 13-25, 2012.

Essa publicação, que é uma extensão de uma publicação em congresso, foi fruto do envolvimento do autor desta tese com o projeto StArt.

- 2) HERNANDES, E. C. M.; TEODORO, E.; **DI THOMMAZO, A.**; FABBRI, S. C. P. F.. Uso de visualização e mineração de texto no processo de análise qualitativa: um estudo de viabilidade. *Revista de Sistemas e Computação - RSC*, v. 3, p. 132-145, 2013.

Essa publicação foi fruto do envolvimento do autor desta tese com o projeto Insigth.

7.4.1.3 Capítulos de livros publicados

- 1) **FABBRI, S. C. P. F.; HERNANDES, E. C. M.; DI THOMMAZO, A.; BELGAMO, A.; ZAMBONI, A.; SILVA, C.** Using information visualization and text mining to facilitate the conduction of systematic literature reviews. In: José Cordeiro, Leszek A. Maciaszek, Joaquim Filipe. (Org.). *Lecture Notes in Business Information Processing*,. 1ed. Berlin: Springer, 2013, v. 141, p. 1-12.

Essa publicação, que é uma extensão de uma publicação premiada como melhor artigo em um congresso, foi fruto do envolvimento do autor desta pesquisa com o projeto StArt

7.5 Orientações de Iniciação Científica

Durante a tese de doutorado, o autor deste trabalho orientou um total de nove projetos de iniciação científica focados no desenvolvimento e validação das ferramentas relatadas na Seção 7.2.5.

- 1) Gabriel Gioannini Malimpensa. Criação do módulo de persistência de dados e geração de relatórios no software StArt - State of the Art through Systematic Review. 2010. Iniciação Científica. (Graduando em Tecnologia em Análise e Desenvolvimento de Sistema) - Instituto Federal de Educação, Ciência e Tecnologia de São Paulo.
- 2) Cleiton Rodrigo Queiroz Silva. Desenvolvimento de Ferramenta CASE para Suporte a Rastreabilidade no Processo de Desenvolvimento de Software. 2011. (Graduando em Tecnologia em Análise e

- Desenvolvimento de Sistema) - Instituto Federal de Educação, Ciência e Tecnologia de São Paulo
- 3) Maiko Charles Trindade. Desenvolvimento de Ferramenta CASE para suporte ao desenvolvimento de software com uso de Data Mining, Text Mining e Visualização. Início: 2012. (Graduando em Tecnologia em Análise e Desenvolvimento de Sistema) - Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior.
 - 4) Denis Bittencourt Muniz. Aprimoramento da ferramenta iCode com técnicas de visualização e mineração de texto. Início: 2012. (Graduando em Tecnologia em Análise e Desenvolvimento de Sistema) - Instituto Federal de Educação, Ciência e Tecnologia de São Paulo.
 - 5) Emanuel Teodoro. Desenvolvimento da ferramenta iCode com novas técnicas de mineração de texto e visualização conceitual e estratégica. Início: 2013. (Graduando em Tecnologia em Análise e Desenvolvimento de Sistema) - Instituto Federal de Educação, Ciência e Tecnologia de São Paulo.
 - 6) Thiago Ribeiro Oliveira. Desenvolvimento de Ferramenta com detecção automática de dependência entre artefatos de software com uso de Data Mining e Visualização. Início: 2012 - Instituto Federal de Educação, Ciência e Tecnologia de São Paulo.
 - 7) Guilherme Olivatto. Uso de Lógica Fuzzy para determinação de links de rastreabilidade de requisitos. Início: 2013 (Graduando em Tecnologia em Análise e Desenvolvimento de Sistema) - Instituto Federal de Educação, Ciência e Tecnologia de São Paulo.

- 8) Thiago Ribeiro Oliveira. Uso de Inteligência Artificial para determinação de links de rastreabilidade de requisitos. Início: 2013 (Graduando em Tecnologia em Análise e Desenvolvimento de Sistema) - Instituto Federal de Educação, Ciência e Tecnologia de São Paulo.
- 9) Gislaine Gonçalves. Realização de estudos experimentais para avaliação de abordagens de determinação automática da rastreabilidade de requisitos. Início: 2013 (Graduanda em Tecnologia em Análise e Desenvolvimento de Sistema) - Instituto Federal de Educação, Ciência e Tecnologia de São Paulo.

7.6 Participação em outros projetos do grupo de pesquisa

Durante o desenvolvimento desta pesquisa de doutorado o autor desta tese se envolveu com outros projetos desenvolvidos no laboratório, tendo contribuído tanto com ideias e propostas bem como com a parte de implementação de ferramentas. Destaca-se ainda que o framework JAVA criado para dar suporte a algoritmos de processamento de linguagem natural, utilizados inicialmente na ferramenta COCAR, foi incorporado nas ferramentas StArt e Insigth.

- StArt - State of the Art Through Systematic Review: a ferramenta StArt dá suporte ao processo de revisão sistemática. Possui funcionalidades de processamento de linguagem natural e de visualização. A ferramenta está em processo de registro.
- Insigth: a ferramenta Insight dá suporte à Análise Qualitativa. Também possui funcionalidades de processamento de linguagem natural e de visualização. A ferramenta também está em processo de registro.

7.7 Oportunidades Futuras

Apresentam-se algumas ideias que deverão ser desenvolvidas no grupo de pesquisa como continuidade desta tese, a saber:

- Evoluir a ferramenta COCAR para dar suporte à seleção dos casos de testes de regressão com priorização e caracterização do impacto de mudança, conforme apresentado no Capítulo 6.
- Adaptar as abordagens propostas para determinar de forma automática a rastreabilidade entre requisitos funcionais e casos de testes.
- Integrar as abordagens propostas com ferramenta de gestão de requisitos para permitir a realização de novos experimentos. Dessa forma será possível importar na ferramenta COCAR requisitos de sistemas que já estão cadastrados em outras ferramentas, para posterior geração da RTM.

Explorar como que o suporte a visualização da rastreabilidade entre os requisitos pode contribuir para o processo de gestão de requisitos e eventualmente para facilitar outras atividades do processo de desenvolvimento de software. Observa-se que a visualização está implementada na ferramenta COCAR, mas no decorrer deste trabalho esse aspecto não foi explorado. Focou-se essencialmente nos links de rastreabilidade no formato de matriz. Exemplos da aplicação da visualização e uma breve revisão bibliográfica sobre o tema estão contidos em (DI THOMMAZO, 2012). O Apêndice A também exemplifica o uso de visualização para representar a rastreabilidade.

REFERÊNCIAS

ABADI, A.; NISENSEN, M.; SIMIONOVICI, Y. A Traceability Technique for Specifications. In: THE 16TH IEEE INTERNATIONAL CONFERENCE ON PROGRAM COMPREHENSION, **Proceedings...** 2008.

ABBOTT, R. J. **An integrated approach to software development**. Wiley, 1986.

ALI, N.; AND GUEHENEUC, Y.-G.; AND ANTONIOL, G.; Trust-Based Requirements Traceability, 2011 IEEE 19th International Conference on Program Comprehension (ICPC), **Proceedings...** 2011

ANTONIOL, G. et al. Recovering code to documentation links in OO systems. In: SIXTH WORKING CONFERENCE ON REVERSE ENGINEERING. **Proceedings...** 1999.

ANTONIOL, G. et al. Information retrieval models for recovering traceability links between code and documentation. In: INTERNATIONAL CONFERENCE ON SOFTWARE MAINTENANCE, **Proceedings...** 2000.

ANTONIOL, G. et al. Tracing object-oriented code into functional requirements. In: 8TH INTERNATIONAL WORKSHOP ON PROGRAM COMPREHENSION, 2000. **Proceedings...** IWPC 2000.

ANTONIOL, G. et al. Recovering traceability links between code and documentation. **IEEE Transactions on Software Engineering**, v. 28, n. 10, p. 970 – 983, out. 2002.

APONTE, J.; MARCUS, A. Improving traceability link recovery methods through software artifact summarization. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, **Proceedings** 2011.

ARTERO, A. O. **Inteligencia Artificial - Teorica e Pratica** LIVRARIA DA FISICA, 2009.

ASSAWAMEKIN, N. Resolving semantic heterogeneity in multiperspective requirements traceability using ontology matching, **Journal of Convergence Information Technology**, 2011

BAEZA-YATES, R.; RIBEIRO-NETO, B. **Modern Information Retrieval**. 1st. ed. [s.l.] Addison Wesley, 1999.

BASILI G., et al., M. **Packaging researcher experience to assist replication of experiments**. ISERN Meeting, Sydney, pp. 3-6. 1996.

BASILI, V. R.; CALDIERA, G.; ROMBACH, H. D. The Goal Question Metric Approach. In: **Encyclopedia of Software Engineering**. Wiley, 1994. .

BELGAMO, A., FABBRI, S.C.P.F. GUCCRA: Técnica de Leitura para apoiar a Construção de Modelos de Caso de Uso e a Análise de Documentos de Requisitos. In: SBES – 19º Simpósio Brasileiro de Engenharia de Software, **Proceedings...** Uberlândia, MG, 3-7 out., 2005

CASTOR, A. et al. Towards Requirement Traceability in TROPOS. In: WORKSHOP EM ENGENHARIA DE REQUISITOS. **Proceedings...**, 2004.

CASTRO, P.A.D. **Um Paradigma Baseado em Algoritmos Genéticos para o Aprendizado de Regras Fuzzy**. Tese Mestrado em Ciência da Computação) – Departamento de Computação, Universidade Federal de São Carlos, São Carlos, 2004.

CLELAND-HUANG, J. et al. Utilizing supporting evidence to improve dynamic requirements traceability In: IEEE International Conference on Requirements Engineering. **Proceedings...** 2005

CLELAND-HUANG, J. Just enough requirements traceability. In: INTERNATIONAL COMPUTER SOFTWARE AND APPLICATIONS CONFERENCE. **Proceedings...** 2006

CLELAND-HUANG, J. et al. A machine learning approach for tracing regulatory codes to product specific requirements 32nd ACM/IEEE International Conference on Software Engineering - Volume 1. **Proceedings...**: ICSE'10. New York, NY, USA: ACM.2005

CLELAND-HUANG, J.; GOTEL, O.; ZISMAN, A. **Software and Systems Traceability**. 2012. ed. 1 Springer, 2012.

CLELAND-HUANG, J. et al. , Software Traceability: Trends and Future Directions, In: FOSE 2014 Proceedings of the on Future of Software Engineering, **Proceedings...** 2014

COPPIN, B. **Inteligência Artificial**. [s.l.] LTC, 2010.

CRESTANI, F. et al. "Is this document relevant?...Probably": A survey of probabilistic models in information retrieval. **ACM Comput. Surv.**, v. 30, n. 4, p. 528–552, dez. 1998.

CUDDEBACK, D.; DEKHTYAR, A.; HAYES, J. Automated Requirements Traceability: The Study of Human Analysts. In: REQUIREMENTS ENGINEERING CONFERENCE (RE), 2010 18TH IEEE INTERNATIONAL. **Proceedings...** 2010

DEEPTIMAHANTI, D. K.; SANYAL, R. **Semi-automatic generation of UML models from natural language requirements** In:4th India Software Engineering Conference 2011, ISEC'11 **Proceedings...** 2011.

DE LUCIA, A., OLIVETO, R., SGUEGLIA, P.: Incremental approach and user feedbacks: A Silver Bullet for traceability recovery. In: Proceedings of 22nd IEEE International Conference on Software Maintenance, pp. 299–309. Sheraton Society Hill, Philadelphia, PA. **Proceedings...** IEEE CS Press (2006)

DEERWESTER, S. et al. Indexing by latent semantic analysis. **JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE**, v. 41, n. 6, p. 391–407, 1990.

DI THOMMAZO, A. **Gerenciamento de Requisitos no Ambiente COCAR** 2007. Dissertação (Mestrado em Ciências da Computação) – Departamento de Computação, Universidade Federal de São Carlos, São Carlos, 2007

DI THOMMAZO, A. Requirements Traceability Matrix: Automatic Generation and Visualization. In: 2012 26TH BRAZILIAN SYMPOSIUM ON SOFTWARE ENGINEERING. Los Alamitos, CA, USA: IEEE Computer Society, **Proceedings...** 2012.

DI THOMMAZO, A et al. (2013). Detecting traceability links through neural networks. In 25th International Conference on Software Engineering and Knowledge Engineering, SEKE, Boston, USA. July. Illinois: Knowledge Systems Institute, **Proceedings...** 2013. 36-41.

DI THOMMAZO, A. et al. An automatic approach to detect traceability links using fuzzy logic. In: 2013 27TH BRAZILIAN SYMPOSIUM ON SOFTWARE ENGINEERING. Brasilia, Brasil, **Proceedings...** 2013.

DI THOMMAZO, A. et al. Using Artificial Intelligence Techniques To Enhance Traceability Links. 2014 In: 16th International Conference on Enterprise Information Systems, 2014, Lisboa. **Proceedings...**2014

ENSLEN, E. et al. Mining source code to automatically split identifiers for software analysis In:2009 6th IEEE International Working Conference on Mining Software Repositories, MSR **Proceedings...** 2009.

Engström et al “A Systematic Review on Regression Test Selection Techniques” **Information and Software Technology**, v. 52, n. 1, p. 14–30, 2010

FABBRI, S. C. P. F. et al. Managing literature reviews information through visualization. In: INTERNATIONAL CONFERENCE ON ENTERPRISE INFORMATION SYSTEMS, 14th. Jun 2012, Wroclaw. **Proceedings...** Lisboa: INSTICC, 2012. p. 36-45

FELDMAN, R.; SANGER, J. **The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data**. Cambridge University Press, 2006.

FIELD, A. **DISCOVERING STATISTICS USING SpSS**, SAGE Publications, London, 2009.

Filho et al Supporting concern based regression testing and prioritization in a model driven environment In: 34th Annual Computer Software and Applications Conference Workshops (COMPSACW), **Proceedings...** Seul 2010

GIBIEC, M.; CZAUDERNA, A.; CLELAND-HUANG, J. Towards mining replacement queries for hard-to-retrieve traces IEEE/ACM international conference on Automated software engineering. **Proceedings...**: ASE'10. New York, NY, USA: ACM, 2010

GOKNIL, A. et al. Semantics of trace relations in requirements models for consistency checking and inferencing. **Software and Systems Modeling**, v. 10, n. 1, p. 31–54, 2011.

GOTEL, O.; FINKELSTEIN, A. Extended requirements traceability: results of an industrial case study In: Third IEEE International Symposium on Requirements Engineering, 1997. **Proceedings...** 1997

GUO, Y. et al. An ontology based improved software requirement traceability matrix. In: 2ND INTERNATIONAL SYMPOSIUM ON KNOWLEDGE ACQUISITION AND MODELING, **Proceedings...** 2009.

HAYES, J. H.; DEKHTYAR, A.; OSBORNE, J. Improving requirements tracing via information retrieval. In: REQUIREMENTS ENGINEERING CONFERENCE, 2003. **Proceedings...** 11th IEEE INTERNATIONAL. set. 2003

HAYES, J. H.; DEKHTYAR, A.; SUNDARAM, S. K. Advancing candidate link generation for requirements tracing: the study of methods. **IEEE Transactions on Software Engineering**, v. 32, n. 1, p. 4 – 19, jan. 2006.

HERNANDES, E. C. M. et al. Avaliação da ferramenta StArt utilizando o modelo TAM e o paradigma GQM. In: EXPERIMENTAL SOFTWARE ENGINEERING LATIN AMERICAN WORKSHOP, ESELAW, 10th. Nov., 2010, Goiânia. **Proceedings...** São Carlos: ICMC, 2010. pp. 10-22.

IEEE Standard Computer Dictionary. A Compilation of IEEE Standard Computer Glossaries. **IEEE Std 610**, p. 1, 1991.

JIANG, H. et al. Incremental Latent Semantic Indexing for Automatic Traceability Link Evolution Management, Proceedings of the 2008 23rd International Conference on Automated Software Engineering, **Proceedings...** 2008

KAWAI, K.K. **Guidelines for preparation of requirements document with emphasis on the Functional Requirements**. 2005. Dissertação (Mestrado em Ciências da Computação) – Departamento de Computação, Universidade Federal de São Carlos, São Carlos, 2005.

MAHESWARI, R.U.; JEYAMALA, D., "A novel approach for test case prioritization," Computational Intelligence and Computing Research (ICCIC), 2013 IEEE International Conference on , vol., no., pp.1,5, 26-28 Dec. 2013
doi: 10.1109/ICCIC.2013.6724209

KITAMURA, M. et al. A Representation Method to Simplify Traceability Links between Software Artifacts" In: 12th IEEE International Conference on Intelligent Software Methodologies, Tools and Techniques, Budapest, **Proceedings...** Hungary, 2013

KLIR, G. J.; YUAN, B. **Fuzzy Sets and Fuzzy Logic: Theory and Applications**. 1st. ed. Prentice Hall, 1995.

KUKREJA, N.; HALFOND, W.G.J.; TAMBE, M., Randomizing regression tests using game theory. In: Automated Software Engineering (ASE), 2013 **Proceedings...** 2013

LUCIA, A. DE; OLIVETO, R.; SGUEGLIA, P. Incremental Approach and User Feedbacks: a Silver Bullet for Traceability Recovery. In: 22ND IEEE INTERNATIONAL CONFERENCE ON SOFTWARE MAINTENANCE, 2006. ICSM '06. **Proceedings...** . 2006

MACDONELL, S. G.; GRAY, A. R.; CALVERT, J. M. FULSOME: a fuzzy logic modeling tool for software metricians. In: FUZZY INFORMATION PROCESSING SOCIETY, 1999. NAFIPS. 18TH INTERNATIONAL CONFERENCE OF THE NORTH AMERICAN. **Proceedings...** 1999

MARCUS, A.; MALETIC, J. I. Recovering documentation-to-source-code traceability links using latent semantic indexing. In: 25TH INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, **Proceedings...** 2003.

MARCUS, A.; MALETIC, J. I.; SERGEYEV, A. Recovery of traceability links between software documentation and source code. **International Journal of Software Engineering and Knowledge Engineering**, p. 811–836, 2005.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, v. 5, n. 4, p. 115–133, 1 dez. 1943.

MUNSON, E. V.; NGUYEN, T. N. Concordance, conformance, versions, and traceability. In: The 3rd ACM int. Workshop on traceability in emerging forms of software engineering 2005, held with the 20th IEEE/ACM int. Conf. Automated software engineering, **Proceedings...** 2005

MYERS, G. J. et al. **The art of software testing**. John Wiley and Sons, 2004. ISBN

OLIVETO, R. et al. On the equivalence of information retrieval methods for automated traceability link recovery. In: Proceedings of the 2010 IEEE 18th International Conference on Program Comprehension, pp. 68–71. IEEE Computer Society, Washington, DC, **Proceedings...** (2010)

OLSEN, G. K.; OLDEVIK, J. **Scenarios of traceability in model to text transformations**. v. 4530 LNCS

PAVKOVIC, N. et al. Case studies to explore indexing issues in product design traceability framework , ICED 11 - 18th International Conference on Engineering Design - Impacting Society Through Engineering Design **Proceedings...** 2011

PENTA, M. DI; GRADARA, S.; ANTONIOL, G. **Traceability recovery in RAD software systems**. In: 10TH INTERNATIONAL WORKSHOP ON PROGRAM COMPREHENSION, **Proceedings...** . 2002

PETERSEN, K. et al. **Systematic mapping studies in software engineering** In:12th international conference on Evaluation and Assessment in Software Engineering. **Proceedings...** ...: EASE'08.Swinton, UK, UK: British Computer Society, 2008.

PORT, D. et al. Text mining support for software requirements: Traceability assurance In:Annual Hawaii International Conference on System Sciences. **Proceedings...** 2011.

POWER, N.; MOYNIHAN, T. **A theory of requirements documentation situated in practice**Proc. of the 21st Annu. Int. Conf. on Doc. In: SIGDOC 2003: **Proceedings...** . 2003

RAMZAN, M. et al. Value based fuzzy requirement prioritization and its evaluation framework. In: 4TH International Conference On Innovative Computing, Information and Control, ICICIC **Proceedings...** 2009.

JACCARD,. The Probabilistic Basis of Jaccard's Index of Similarity. **Systematic Biology**, v. 45, n. 3, p. 380–385, 1 set. 1996.

REZENDE, S. O.; MARCACINI, R. M.; MOURA, M. F. O uso da mineração de textos para extração e organização não supervisionada de conhecimento. **Revista de Sistema de Informação da FSMA**, v. 1, n. n.7, p. 7–21, 11 jul. 2011.

ROCHA, A. R. C. DA; MALDONADO, J. C.; WEBER, K. C. **Qualidade de software: teoria e prática** Prentice Hall, 2001.

ROTHERMEL, G. et al. Prioritizing test cases for regression testing," Software Engineering, IEEE Transactions on , vol.27, no.10, pp.929,948, **Proceedings...** 2001

SALEM, A. M. Improving Software Quality through Requirements Traceability Models In: Computer Systems and Applications, 2006. IEEE International Conference on. **Anais...** In: COMPUTER SYSTEMS AND APPLICATIONS, 2006. **Proceedings...** , 2006.

SALTON, G.; ALLAN, J. Text Retrieval Using the Vector Processing Model. In: 3RD SYMPOSIUM ON DOCUMENT ANALYSIS AND INFORMATION RETRIEVAL. University of Nevada, **Proceedings...** Las Vegas: 1994.

SETTIMI, R. et al. Supporting software evolution through dynamically retrieving traces to UML Artifacts. In: Proceedings of 7th IEEE International Workshop on Principles of Software Evolution, pp. 49–54. IEEE CS Press, **Proceedings...** Kyoto, Japan 2004

SANDHU, P.; SINGH, H. A Fuzzy-Inference System Based Approach for the Prediction of Quality of Reusable Software Components. In: INTERNATIONAL CONFERENCE ON ADVANCED COMPUTING AND COMMUNICATIONS, 2006. **Proceedings...** 2006.

SOMMERVILLE, I. **Software Engineering**. 9. ed. Addison-Wesley, 2011.

SINGH et al Systematic Literature Review on Regression Test Prioritization Techniques. **Informatica** 2012

SRIKANTH, H.; WILLIAMS, L.; OSBORNE, J. System test case prioritization of new and regression test cases. In: **Empirical Software Engineering, 2005 International Symposium on**. IEEE, 2005. p. 10 pp.

STANDISH. **The CHAOS Reports 2013** - Standish Group. Disponível em: <http://www.versionone.com/assets/img/files/CHAOSManifesto2013.pdf> Acesso em: 07/04/2014.

STANDISH. **The CHAOS Reports 2014** - Standish Group. Disponível em: <http://www.projectsmart.co.uk/docs/chaos-report.pdf> Acesso em: 01/10/2014.

STONE, A.; SAWYER, P. Using pre-requirements tracing to investigate requirements based on tacit knowledge. In: ICSOFT 2006 - 1ST International Conference on Software and Data Technologies, **Proceedings...** 2006.

SUNDARAM, S. K. et al. Assessing traceability of software engineering artifacts. **Requirements Engineering**, v. 15, n. 3, p. 313–335, 2010.

SULTANOV, H., HAYES, J.H., Application of swarm techniques to requirements engineering: Requirements tracing, Proceedings of the 2010 18th IEEE International Requirements Engineering Conference, RE2010, **Proceedings...** 2010

SULTANOV, H.A.; HAYES, J. H.; KONG, W., **Application of swarm techniques to requirements tracing**, Requirements Engineering, 2011

TAVASSOLI, D. **10 steps to better requirements management** - IBM Software Requirements Definition and Management, jun.2011, Disponível em: <http://public.dhe.ibm.com/common/ssi/ecm/en/raw14059usen/RAW14059USEN.PDF> Acesso em: 05/12/2012.

TRAVASSOS, G. H., GUROV, D., AMARAL, E. A. G. D. **Introdução à Engenharia de Software Experimental**. COPPE / UFRJ, p.52. 2002. (RELATÓRIO TÉCNICO 590/02).

Udagawa, Y. An Augmented Vector Space Information Retrieval for Recovering Requirements Traceability, 11th International Conference on Data Mining Workshops (ICDMW), **Proceedings...** 2011

WANG, X.; LAI, G.; LIU, C. Recovering Relationships between Documentation and Source Code based on the Characteristics of Software Engineering. **Electronic Notes in Theoretical Computer Science**, v. 243, p. 121–137, 2009.

KONG, W.; HAYES, J.H. Proximity-based traceability: An empirical validation using ranked retrieval and set-based measures, First International Workshop on Empirical Requirements Engineering (EmpiRE), **Proceedings...** 2011

WOHLIN, C.; RUNESON, P.; HÖST, M. **Experimentation in software engineering: an introduction**. London: Springer, 2000. 250p.

YEN, J.; TIAO, W. A. A formal framework for the impacts of design strategies on requirements. In: FUZZY SYSTEMS SYMPOSIUM, 1996. Software Computing In Intelligent Systems And Information Processing, **Proceedings....** 1996

ZADEH, L. A. Fuzzy sets. **Information and Control**, v. 8, n. 3, p. 338–353, jun. 1965.

ZAMBONI, A. B. et al. StArt Uma ferramenta computacional de apoio à Revisão Sistemática. In: CONGRESSO BRASILEIRO DE SOFTWARE – SESSÃO DE FERRAMENTAS, CbSoft, 1th. Set., 2010, Salvador. **Proceedings...** Salvador: UFBA, 2010. p. 10-12

ZIMMERMANN, H. J. **Fuzzy Set Theory-And Its Applications**. Springer, 2001.

ZISMAN, A.; SPANOUDAKIS, G. **Software Traceability: Past, Present, and Future**. The Newsletter of the Requirements Engineering Specialist Group of the British Computer Society – Disponível em : <http://www.resg.org.uk/archive/rq33.pdf>. Acesso em: 05/03/2011

APÊNDICE A

Esse Apêndice contém detalhes da ferramenta COCAR que foi construída para dar suporte às abordagens propostas e para possibilitar a realização dos experimentos

Introdução

A primeira versão da ferramenta COCAR foi implementada durante o mestrado do autor desta tese. Na ocasião ela foi desenvolvida para ambiente WEB e as principais funcionalidades eram relacionadas com a inserção dos requisitos de um sistema, suporte ao gerenciamento de requisitos, elaboração de modelos de casos de uso e geração automática de pontos de casos de uso. No início da pesquisa desta tese de doutorado tomou-se a decisão de migrar a ferramenta para ambiente desktop com o intuito de facilitar a realização de experimentos e para a utilização de algumas bibliotecas de visualização e mineração de texto. No decorrer deste apêndice serão apresentados os aspectos de implementação da ferramenta e suas principais funcionalidades.

Aspectos de implementação

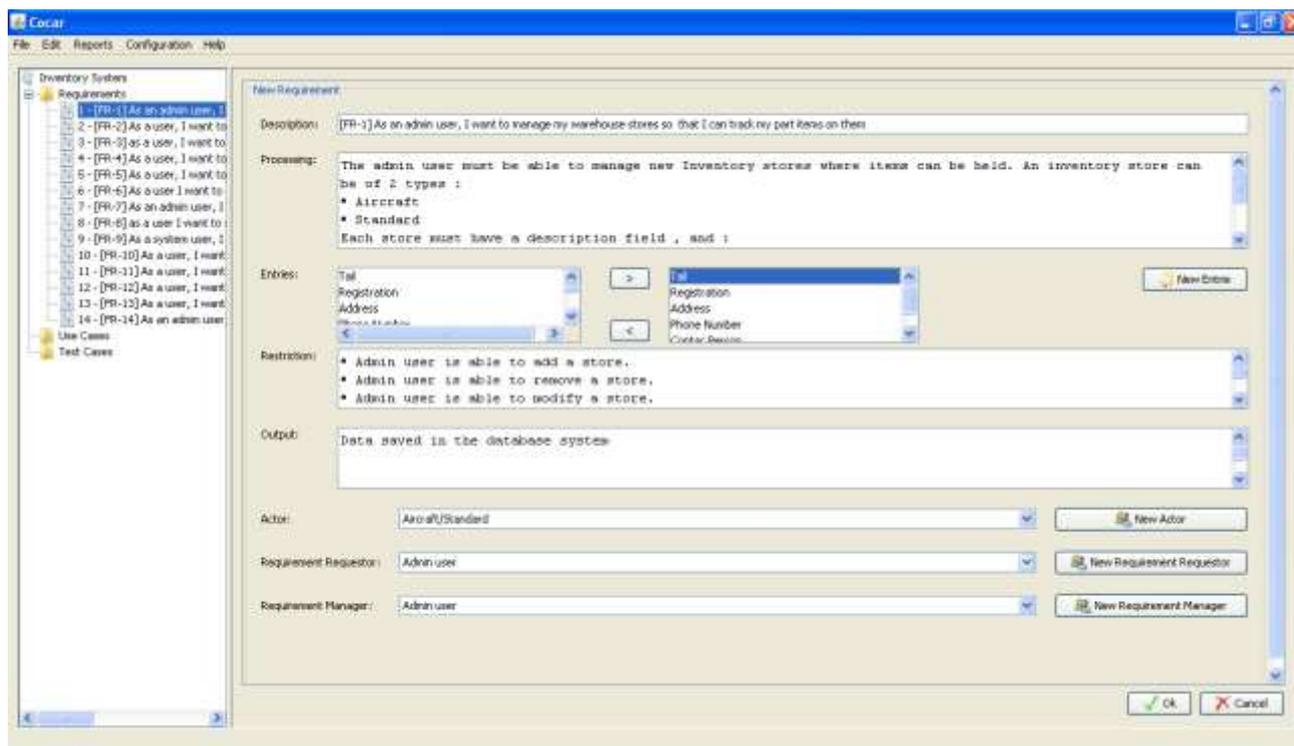
A ferramenta foi implementada em linguagem JAVA (JSE) na versão 7. A linguagem foi escolhida pelo seu suporte ao desenvolvimento de aplicações para desktop com a biblioteca SWING e por possuir um ambiente de bom desempenho e gratuito, o NetBeans.

Para dar agilidade ao desenvolvimento foi utilizado o framework Java Persistente API (JPA) para a persistência dos dados. O banco de dados utilizado foi o HSQLDB (<http://hsqldb.org/>), Optou-se por utilizar o HSQLDB pela facilidade de implementação e para facilitar a distribuição de sistemas cadastrados na ferramenta. Sendo assim, quando os requisitos de um sistema são salvos na ferramenta é construído um arquivo de banco de dados, em encapsulado em um arquivo “.cocar”. Essa prática facilitou a realização dos estudos experimentais.

Funcionalidades

Cadastro de Sistemas: A ferramenta dá suporte ao cadastramento de sistemas. Quando o sistema é inserido na ferramenta as demais funcionalidades da ferramenta são habilitadas.

Cadastro de requisitos: O cadastramento de requisitos segue o *template* proposto por Kawai (2005), A pesquisadora fez uma investigação em diversos *templates* disponíveis na literatura e, a partir desse estudo, propôs os campos que devem estar disponíveis em cada requisito funcional. A Figura A1 mostra a tela de cadastro de requisitos:



A1- Cadastro de requisitos na ferramenta COCAR

Descrição: explicação breve do requisito. O objetivo deste item é registrar “o quê” a funcionalidade deve fazer e não o “como”. Uma vez cadastrados diversos requisitos no sistema, uma listagem deste item “descrição” permite uma boa ideia de tudo o que o sistema deve realizar. Na interface do ambiente COCAR utilizou-se esse campo para apresentar um requisito quando fosse necessário.

Processamento: composto pelo fluxo de atividades ou ações que o sistema deve realizar para alcançar o que está registrado no item “Descrição”. O campo do processamento permite inserção de grandes textos. Trata-se de um dos campos mais importantes do cadastro de um requisito funcional

Entradas: nesse campo são indicados todos os dados que estão envolvidos na funcionalidade. É composto por uma lista de dados que são cadastráveis no ambiente, indicando-se o nome, a descrição e o tipo (inteiro, *float*, texto, data ou lógico) do dado de entrada.

Restrição: nesse campo são registradas restrições ou condições necessárias para que o requisito seja realizado. O nome original desse campo era Condição/Restrição e no contexto do ambiente foi renomeado simplesmente para Restrição.

Saída: a resposta do sistema ao requisito deve ser indicada nesse campo.

Ator Principal: É representado por uma lista de atores que podem ser cadastrados no sistema, indicando-se o nome e a descrição.

Solicitante do Requisito: Nesse caso, o usuário do sistema deve escolher os solicitantes do requisito dentre os disponíveis em uma lista. Os solicitantes são cadastrados no ambiente COCAR, fornecendo-se nome e descrição.

Gerente do Requisito: Esse atributo também tem foco na rastreabilidade. Nesse caso, o usuário do sistema deve escolher somente um gerente dentre os disponíveis em uma lista. Os gerentes são cadastrados fornecendo-se nome e descrição.

Geração da RTM:

A ferramenta dá suporte a geração da RTM com as quatro abordagens propostas e detalhadas nesta tese: RTM-E, RTM-NLP, RTM-Fuzzy E RTM-N. Na sequência são destacadas figuras com a RTM gerada pela ferramenta.

Matriz de Rastreabilidade (RTM-E)

	[FR-1] ...	[FR-2] ...	[FR-3] ...	[FR-4] ...	[FR-5] ...	[FR-6] ...	[FR-7] ...	[FR-8] ...	[FR-9] ...	[FR-10] ...	[FR-11] ...	[FR-12] ...	[FR-13] ...	[FR-14] ...
[FR-1] ...	-	0.00%	6.67%	0.00%	7.14%	0.00%	10.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
[FR-2] ...	0.00%	-	0.00%	0.00%	0.00%	7.14%	0.00%	0.00%	0.00%	7.14%	0.00%	0.00%	0.00%	0.00%
[FR-3] ...	6.67%	0.00%	-	6.25%	66.67%	6.67%	0.00%	0.00%	21.43%	6.67%	6.25%	0.00%	10.00%	0.00%
[FR-4] ...	0.00%	0.00%	6.25%	-	14.29%	0.00%	0.00%	0.00%	16.67%	0.00%	0.00%	0.00%	0.00%	0.00%
[FR-5] ...	7.14%	0.00%	66.67%	14.29%	-	7.14%	0.00%	0.00%	33.33%	7.14%	6.67%	0.00%	0.00%	0.00%
[FR-6] ...	0.00%	7.14%	6.67%	0.00%	7.14%	-	0.00%	0.00%	0.00%	100.00%	0.00%	0.00%	0.00%	0.00%
[FR-7] ...	10.00%	0.00%	0.00%	0.00%	0.00%	0.00%	-	0.00%	0.00%	0.00%	20.00%	0.00%	0.00%	0.00%
[FR-8] ...	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	-	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
[FR-9] ...	0.00%	0.00%	21.43%	16.67%	33.33%	0.00%	0.00%	0.00%	-	0.00%	0.00%	0.00%	14.29%	0.00%
[FR-10] ...	0.00%	7.14%	6.67%	0.00%	7.14%	100.00%	0.00%	0.00%	0.00%	-	0.00%	0.00%	0.00%	0.00%
[FR-11] ...	30.00%	0.00%	6.25%	0.00%	6.67%	0.00%	20.00%	0.00%	0.00%	0.00%	-	0.00%	0.00%	0.00%
[FR-12] ...	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	-	0.00%	0.00%
[FR-13] ...	0.00%	0.00%	10.00%	0.00%	0.00%	0.00%	0.00%	0.00%	14.29%	0.00%	0.00%	0.00%	-	0.00%
[FR-14] ...	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	-

Visualizar Grafo Todos Exportar para Excel Fechar

Cocar

	[FR-1] ...	[FR-2] ...	[FR-3] ...	[FR-4] ...	[FR-5] ...	[FR-6] ...	[FR-7] ...	[FR-8] ...	[FR-9] ...	[FR-10] ...	[FR-11] ...	[FR-12] ...	[FR-13] ...	[FR-14] ...
[FR-1] ...	-	12.71%	20.18%	32.32%	18.64%	19.47%	26.89%	18.34%	20.14%	18.05%	19.27%	4.60%	4.37%	0.00%
[FR-2] ...	12.71%	-	16.48%	16.30%	16.94%	17.32%	4.44%	7.58%	13.59%	16.06%	9.55%	11.70%	10.11%	0.00%
[FR-3] ...	20.18%	16.48%	-	8.95%	70.46%	30.75%	30.43%	7.08%	12.69%	28.50%	25.27%	11.92%	10.00%	0.00%
[FR-4] ...	32.32%	16.30%	8.95%	-	8.03%	18.07%	4.83%	35.58%	29.12%	16.75%	13.49%	4.85%	8.24%	0.00%
[FR-5] ...	18.64%	16.94%	70.46%	8.03%	-	36.29%	35.69%	3.18%	14.24%	33.64%	25.01%	14.48%	0.00%	0.00%
[FR-6] ...	19.47%	17.32%	30.75%	18.07%	36.29%	-	34.91%	17.86%	4.58%	92.72%	18.76%	18.79%	5.96%	5.83%
[FR-7] ...	26.89%	4.44%	30.43%	4.83%	35.69%	34.91%	-	10.69%	9.78%	0.00%	20.00%	0.00%	0.00%	0.00%
[FR-8] ...	18.34%	7.58%	7.08%	35.58%	3.18%	17.86%	10.69%	-	32.36%	16.56%	13.13%	4.39%	5.21%	0.00%
[FR-9] ...	20.14%	13.59%	12.69%	29.12%	14.24%	4.58%	9.78%	32.36%	-	4.24%	21.03%	1.40%	13.36%	0.00%
[FR-10] ...	18.05%	16.06%	28.50%	16.75%	33.64%	92.72%	32.36%	16.56%	4.24%	-	17.39%	17.43%	5.52%	0.00%
[FR-11] ...	19.27%	9.55%	25.27%	13.49%	25.01%	18.76%	20.85%	13.13%	21.03%	17.39%	-	20.73%	5.48%	5.36%
[FR-12] ...	4.60%	11.70%	11.92%	4.85%	14.48%	18.79%	15.00%	4.39%	1.40%	17.43%	20.73%	-	0.00%	0.00%
[FR-13] ...	4.37%	10.11%	0.00%	8.24%	0.00%	5.96%	2.55%	5.21%	13.36%	5.52%	5.48%	0.00%	-	0.00%
[FR-14] ...	0.00%	0.00%	0.00%	0.00%	0.00%	5.83%	4.99%	0.00%	0.00%	5.41%	5.36%	0.00%	0.00%	-

Visualizar Todos Exportar para Excel Cancelar

A2- Exemplos de RTM gerada na ferramenta COCAR

Visualização da Rastreabilidade entre os requisitos:

A partir da RTM estabelecida é possível visualizar essa dependência em forma de grafos. Nas Figura A3 e A4 é possível observar exemplos de visualização da rastreabilidade com grafos.

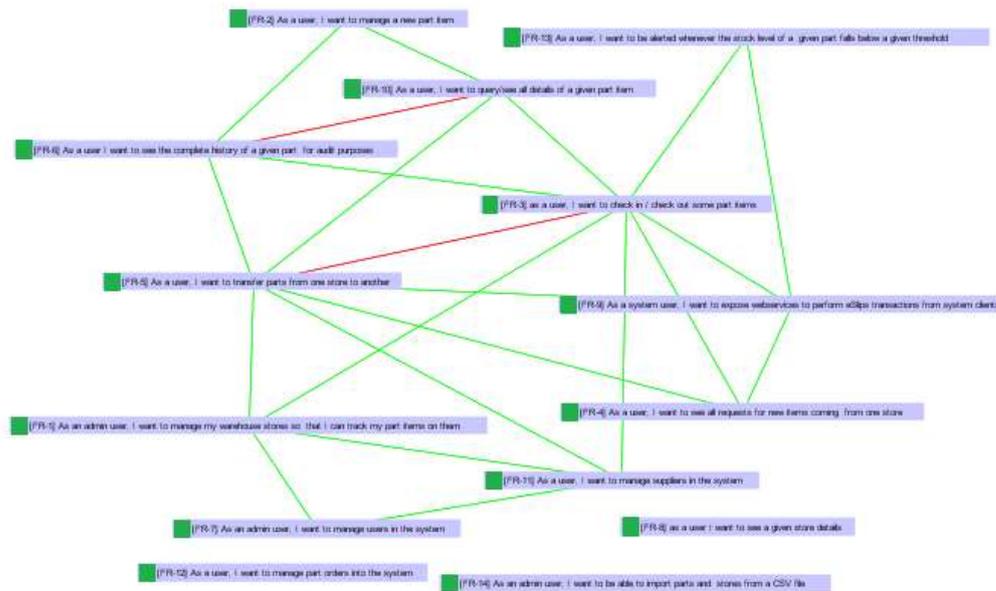


Figura A3. Visualização do relacionamento entre todos os requisitos funcionais



Figura A4. Visualização do relacionamento do requisito funcional RF3 com os demais

Módulo de Experimentação

Esse módulo facilita a realização de experimentos. Por meio dele é possível o pesquisador criar a RTM de referência. A ferramenta exibe os dois requisitos funcionais

envolvidos na definição de cada link de rastreabilidade e o pesquisador define se existe dependência e o seu nível (forte e fraco). Depois de definida a RTM de referência, o sistema calcula os valores da efetividade. Para facilitar as propostas de melhorias realizadas nesta tese, também eram destacados quais os links foram detectados de forma equivocada.

Relatórios

A ferramenta também dá suporte à geração de relatórios sobre os requisitos do sistema cadastrado, permitindo impressão dos resultados e geração de arquivos PDF.