

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

FELIPE GOMES DIAS

Elaboração de Requisitos de Software:
uma Abordagem baseada em
Fragmentos de Casos de Uso

RIO DE JANEIRO
2008

Felipe Gomes Dias

**Elaboração de Requisitos de Software: uma Abordagem
baseada em Fragmentos de Casos de Uso**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática, Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Informática.

Orientador: Eber Assis Schmitz

Co-orientadora: Maria Luiza Machado Campos

Rio de Janeiro
2008

Dias, Felipe Gomes.

Elaboração de Requisitos de Software: uma Abordagem baseada em Fragmentos de Casos de Uso. Rio de Janeiro: UFRJ, 2008.

Dissertação (Mestrado em Informática) – Universidade Federal do Rio de Janeiro – UFRJ, 2008.

Orientador: Eber Assis Schmitz

Co-orientador: Maria Luiza Machado Campos

1. Casos de Uso. 2. Requisitos de Software. 3. Fragmentos de Casos de Uso. I. Eber Assis Schmitz e Maria Luiza Machado Campos (Orient.). II. Universidade Federal do Rio de Janeiro, Instituto de Matemática, Programa de Pós-Graduação em Informática. III. Título.


Felipe Gomes Dias
**Elaboração de Requisitos de Software: uma Abordagem
baseada em Fragmentos de Casos de Uso**

Dissertação submetida como parte dos requisitos necessários para obtenção do grau de Mestre em Informática.

Apresentado por:


Felipe Gomes Dias

Aprovado por:


Professor Eber Assis Schmitz, Ph.D.
(Presidente)


Professora Maria Luiza Machado Campos, Ph.D.


Professor Alexandre Luis Correa, D.Sc.


Professor Julio Cesar Sampaio do Prado Leite, Ph.D.

RIO DE JANEIRO, RJ - BRASIL

Outubro de 2008

RESUMO

DIAS, Felipe Gomes. **Elaboração de Requisitos de Software: uma Abordagem baseada em Fragmentos de Casos de Uso**. Dissertação (Mestrado em Informática) – Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2008.

Este trabalho apresenta uma abordagem para a elaboração de requisitos de software de alta qualidade baseada em Fragmentos de Caso de Uso. Um Fragmento de Caso de Uso é uma seqüência de interações entre o ator e o sistema que pode ser reutilizado em diferentes partes dos documentos de requisitos. O maior ganho de qualidade proveniente do uso dessa abordagem na escrita de Casos de Uso é obtido por alunos e profissionais iniciantes que não possuem a experiência exigida para a documentação de requisitos com a alta qualidade necessária. A abordagem foi analisada através de dois experimentos realizados com alunos de graduação para avaliar a qualidade dos documentos de requisitos elaborados pelos mesmos. Os resultados dessa avaliação permitem concluir que os alunos que utilizaram a abordagem de Fragmentos produziram documentos com menor quantidade de defeitos de especificação de requisitos, em comparação com o desempenho dos alunos que não utilizaram nenhuma abordagem em especial. Este trabalho conclui que o uso de Fragmentos pode auxiliar a produção de documentos com melhor qualidade para aqueles que não possuem experiência em escrever Casos de Uso.

Palavras Chaves: Casos de Uso, Requisitos de Software, Fragmentos de Casos de Uso.

ABSTRACT

*DIAS, Felipe Gomes. **Software Requirements Elaboration: an Approach based on Use Case Fragments.** Thesis (Master's Degree in Informatics) – Federal University of Rio de Janeiro, Rio de Janeiro, 2008.*

This dissertation presents an approach, based on Use Case Fragments, aimed at assisting the preparation of high quality software requirement documents. A Use Case Fragment is a set of use cases excerpts that can be reused in different parts of requirement documents. The usage of Fragments is directed towards students and novice professionals, who do not have the experience required for elaborating the requirements documentation with the quality needed. Two experiments were conducted with undergraduate students to assess the quality of the requirement documents produced by them. In conclusion of this assessment, students who used the Fragments approach produced documents with fewer specification defects when compared to the students who did not use any approach in particular. This study concludes that using Fragments may help inexperienced people to produce high quality use case documents.

Keywords: *Use Cases, Software Requirements, Use Case Fragments.*

AGRADECIMENTOS

Primeiramente, agradeço a **Deus** por estar sempre me dando as forças que preciso para conquistar o que desejo. Agradeço a Deus também pelo dom da vida, tornando possível a minha existência e a de todos os seres humanos, agradecendo especialmente o dom da vida das que serão citadas em seguida.

Agradeço a toda minha família e, principalmente, aos meus pais, **Paulo e Vera**, e a minha irmã, **Simone**, por todo o apoio, amor e carinho que recebo ao longo dos anos. Fico feliz em poder ser um filho e irmão bem amado e motivo de felicidade para minha família.

Agradeço a minha linda e querida noiva, **Liliane**, por todo amor, dedicação, respeito e compreensão que tenho recebido. Obrigado por me ajudar nos momentos difíceis da faculdade e da vida, contribuindo para que eu pudesse concluir meus estudos e trabalhos. Sou muito feliz por ter a honra de ser seu noivo e sou muito feliz por nosso amor.

Agradeço aos professores **Eber e Maria Luiza** pela orientação acadêmica e por me dar a oportunidade de aprender com seus ensinamentos e experiências de vida. Agradeço por confiar no meu trabalho e por estarem sempre dispostos a me ajudar.

Agradeço aos professores **Alexandre e Júlio** por participarem da avaliação do meu trabalho e também pelas muitas vezes que pudemos juntos resolver problemas que contribuíram para meu aprendizado e para a elaboração deste trabalho.

Agradeço aos meus amigos que estão sempre comigo. Todos foram fundamentais para que meu trabalho pudesse ser conduzido da forma mais agradável possível.

Por fim, agradeço à **Universidade Federal do Rio de Janeiro** pelo mestrado que estou concluindo, a todos os **professores, secretárias e funcionários** do departamento e à **CAPES** pela bolsa de pesquisa que recebi ao longo do curso.

LISTA DE FIGURAS

Figura 1	Um diagrama de Casos de Uso contendo um ator e um Caso de Uso.	14
Figura 2	Visões relacionadas a Casos de Uso (MARTINS, 2007)	14
Figura 3	Relacionamentos entre Casos de Uso e outros modelos, traduzidos do trabalho de Bittner e Spence (2002)	16
Figura 4	Propriedades da descrição de um Caso de Uso, traduzidas da visão de Bittner e Spence (2002)	18
Figura 5	Fluxo básico principal com seus fluxos alternativos (BITTNER, SPENCE, 2002)	19
Figura 6	Fases do processo de engenharia de requisitos, traduzidas da visão de Windle e Abreo (2002)	20
Figura 7	Lista dos dez principais erros encontrados em descrições de Casos de Uso, traduzida do trabalho de	23
Figura 8	Exemplo de conteúdo de um Caso de Uso Essencial “Retirar Dinheiro”	26
Figura 9	Tradução de um exemplo de um padrão de cenário	28
Figura 10	Modelo de explicação de um Fragmento e de suas partes	34
Figura 11	Fragmento “Obter Confirmação de Registro”	37
Figura 12	Macro-processo para o uso dos Fragmentos	40
Figura 13	Associar cada sub-objetivo ao Fragmento	42
Figura 14	Personalizando os Fragmentos com objetos do domínio do negócio	43
Figura 15	Compondo a descrição do Caso de Uso	45
Figura 16	Visualização das partes da descrição do Caso de Uso, composta a partir dos trechos de Fluxos e detalhes de Estrutura de Dados e Regras	46
Figura 17	Diagrama de classes que representa o modelo conceitual dos termos de negócio	47
Figura 18	Diagrama de Casos de Uso do exemplo de Aluguel de Carros	47
Figura 19	Personalização do Fragmento “Obter Confirmação de Registro” para o sub-objetivo “Obter Confirmação de Registro do Aluguel”	51
Figura 20	Fluxo Básico resultante da composição dos trechos de Fluxos Básicos	53
Figura 21	Fluxos Alternativos associados aos passos do Fluxo Básico	54
Figura 22	Detalhes das Estruturas de Dados de todos os termos de negócio participantes do Caso de Uso	55
Figura 23	Detalhes das Regras de Negócio que restringem os termos de negócio	55

LISTA DE QUADROS

Quadro 1	Associação entre sub-objetivos do Caso de Uso e Fragmentos do Catálogo	49
----------	--	----

LISTA DE TABELAS

Tabela 1	Resultados do Experimento 1 obtidos a partir da aplicação do teste U de Mann-Whitney	64
Tabela 2	Resultados do Experimento 2 obtidos a partir da aplicação do teste U de Mann-Whitney	65
Tabela 3	Situação de cada hipótese nula do Experimento 1	67
Tabela 4	Situação de cada hipótese nula do Experimento 2	68
Tabela 5	Dados obtidos do questionário e do exercício	89
Tabela 6	Defeitos encontrados a partir da aplicação do <i>Checklist</i> no Experimento 1	90
Tabela 7	Dados obtidos do questionário e do exercício no Experimento 2	91
Tabela 8	Defeitos encontrados a partir da aplicação do <i>Checklist</i> no Experimento 2	92

SUMÁRIO

1 INTRODUÇÃO	3
1.1 MOTIVAÇÃO	3
1.2 OBJETIVOS	5
1.3 DEFINIÇÃO DO ESCOPO DO TRABALHO.....	5
1.4 DELIMITAÇÃO	6
1.5 CONTRIBUIÇÕES.....	7
1.6 ORGANIZAÇÃO DA MONOGRAFIA	7
2 CASOS DE USO	9
2.1 INTRODUÇÃO	9
2.2 ENGENHARIA DE REQUISITOS	9
2.3 TÉCNICA DE CASOS DE USO.....	13
2.4 ETAPAS DE ESPECIFICAÇÃO DE CASOS DE USO.....	19
2.5 QUALIDADE DE CASOS DE USO: PROPOSTAS DE BOAS PRÁTICAS E USO DE PADRÕES.....	22
3 FRAGMENTOS DE CASOS DE USO	30
3.1 INTRODUÇÃO	30
3.2 DESENVOLVIMENTO DA PESQUISA	30
3.3 DEFINIÇÃO DE FRAGMENTOS DE CASOS DE USO.....	32
3.4 UM FRAGMENTO DE CASO DE USO.....	36
3.5 COMO USAR OS FRAGMENTOS DE CASOS DE USO	40
3.5.1 <i>Obter Sub-Objetivos</i>	41
3.5.2 <i>Associar a Fragmentos</i>	41
3.5.3 <i>Personalizar Fragmentos</i>	42
3.5.4 <i>Compor Descrição</i>	44
3.5.5 <i>Completar Descrição</i>	46
3.6 EXEMPLO: CASO DE USO ESCRITO COM FRAGMENTOS	47
3.6.1 <i>Obter Sub-Objetivos</i>	48
3.6.2 <i>Associar a Fragmentos</i>	49
3.6.3 <i>Personalizar Fragmentos</i>	50
3.6.4 <i>Compor Descrição</i>	51
3.6.5 <i>Completar Descrição</i>	52
4 AVALIAÇÃO	56
4.1 MATERIAIS E MÉTODOS.....	56
4.1.1 <i>Descrições Gerais</i>	56
4.1.2 <i>Participantes</i>	56
4.1.3 <i>Contexto</i>	56
4.1.4 <i>Instrumentos</i>	57
4.1.5 <i>Planejamento</i>	57
4.1.6 <i>Hipóteses</i>	59
4.1.7 <i>Variáveis</i>	60
4.1.8 <i>Validação</i>	61
4.2 OS EXPERIMENTOS.....	62
4.2.1 <i>Experimento 1</i>	62
4.2.2 <i>Experimento 2</i>	62
4.3 RESULTADOS	63

4.3.1	<i>Experimento 1</i>	63
4.3.2	<i>Experimento 2</i>	64
5	DISCUSSÃO	66
5.1	ANÁLISE ESTATÍSTICA	66
5.1.1	<i>Critério para rejeição de hipóteses</i>	66
5.1.2	<i>Experimento 1</i>	66
5.1.3	<i>Experimento 2</i>	68
5.2	DISCUSSÃO SOBRE OS EXPERIMENTOS	69
5.2.1	<i>Conclusões sobre os resultados</i>	69
5.2.2	<i>Comparação com outros Trabalhos</i>	71
6	CONCLUSÃO	73
6.1	CONSIDERAÇÕES FINAIS	73
6.2	TRABALHOS FUTUROS	74
	REFERÊNCIAS	75
7	APÊNDICE A – CATÁLOGO DE FRAGMENTOS DE CASOS DE USO	80
7.1	FRAGMENTO: SELEÇÃO ÚNICA DE UMA LISTA	80
7.2	FRAGMENTO: SELEÇÃO MÚLTIPLA DE UMA LISTA	81
7.3	FRAGMENTO: OBTER DETALHES DE UM OBJETO	82
7.4	FRAGMENTO: OBTER CONFIRMAÇÃO DE REGISTRO	83
7.5	FRAGMENTO: OBTER CONFIRMAÇÃO DE ENVIO DE ARQUIVO	84
7.6	FRAGMENTO: OBTER CONFIRMAÇÃO DE EXCLUSÃO DE OBJETO	85
7.7	FRAGMENTO: BUSCAR OBJETOS	86
	APÊNDICE B – ENUNCIADO DO EXPERIMENTO	87
	APÊNDICE C – DADOS COLETADOS NO EXPERIMENTO 1	89
	APÊNDICE D – DADOS COLETADOS NO EXPERIMENTO 2	91
	APÊNDICE E – QUESTIONÁRIO DO EXPERIMENTO	94
	ANEXO A – CHECKLIST EM PORTUGUÊS	95

1 INTRODUÇÃO

1.1 Motivação

A especificação de requisitos através de Casos de Uso tem se tornado uma das formas mais utilizadas dentre as diversas abordagens disponíveis para a especificação de requisitos de sistemas de informação (KULAK, GUINEY, 2003). De um modo sucinto, o objetivo da utilização dessa forma de especificação é auxiliar o Engenheiro de Software a descrever a maioria dos requisitos funcionais para um sistema em desenvolvimento. De acordo com Booch *et al* (2005), um Caso de Uso especifica o comportamento de um sistema, constituindo a descrição de um conjunto de seqüências de ações, incluindo variantes realizadas pelo sistema para produzir um resultado observável do valor de um ator.

Um modelo de Casos de Uso é composto por Diagramas de Casos de Uso e especificações textuais (que chamaremos neste trabalho simplesmente de Casos de Uso) que descrevem todas as formas de serviços fornecidos aos usuários (ÖVERGAARD, PALMKVIST, 2004; MERRICK, BARROW, 2005). A notação gráfica utilizada nos Diagramas de Casos de Uso possui uma sintaxe definida e mantida pelo Grupo de Gerenciamento de Objetos (*Object Management Group*, OMG)¹ e fornece uma representação visual, mostrando os Casos de Uso e os relacionamentos entre os mesmos e as entidades externas, chamadas de Atores.

Casos de Uso tornaram-se muito populares devido ao seu estilo de escrita informal e simples, que é compreensível tanto para os técnicos como para os interessados não-técnicos que participam da elaboração dos documentos de requisitos de software. Tal fator torna-se

¹ Página oficial do grupo OMG: <http://www.omg.org> .

importante devido à dificuldade de comunicação comumente encontrada entre profissionais de negócio e de sistemas, como explicado por Martínez *et al* (2004) e Weill e Broadbent (1999). Porém, como o desenvolvimento de Casos de Uso ainda é uma tarefa artesanal, a qualidade resultante da especificação é diretamente dependente da experiência do autor e da complexidade da solução (BITTNER, SPENCE, 2002; JAGIELSKA *et al*, 2006; BELGAMO, FABBRI, 2004).

A escrita de Casos de Uso, embora aparentemente simples, é uma tarefa difícil de ser completada com qualidade. Isso é demonstrado pelos inúmeros problemas encontrados, tais como: ambigüidades, redundâncias, inconsistências, conflitos com a terminologia do domínio, como mostra Fantechi *et al* (2003), e a inclusão de detalhes de implementação (Casos de Uso contaminados por jargões), o que os torna difíceis de serem mantidos e entendidos pelos interessados não-técnicos (DENNEY, 2005; JAGIELSKA *et al*, 2006). Finalmente, de acordo com El-Attar e Miller (2006), Hazan *et al* (2005) e Knight (2004, p.2), problemas em requisitos contribuem para a produção de sistemas de informação de baixa qualidade.

A dificuldade da escrita de Casos de Uso de qualidade é devida à grande quantidade de detalhes que devem ser especificados, entre os quais podemos citar: o fluxo básico, com sua seqüência de passos; os fluxos alternativos; os detalhes das informações trocadas entre os Atores e o sistema. Além dessas informações, as regras de negócio associadas às interações de um Caso de Uso devem ser especificadas ou referenciadas. Como consequência, a escrita de Casos de Uso é uma das tarefas que mais consome tempo durante o processo de modelagem de sistemas e seus requisitos (BITTNER, SPENCE, 2002).

A especificação de todos os detalhes para a produção de um Caso de Uso de qualidade é uma tarefa complicada para estudantes e profissionais iniciantes e, como consequência, só podem ser produzidos após um longo período de treinamento. Neste trabalho, estamos interessados em apoiar a produção de Casos de Uso de alta qualidade, no contexto de Casos

de Uso escritos por alunos e profissionais principiantes. Acreditamos que possa haver um modo de auxílio à escrita que aumente a qualidade dos Casos de Uso produzidos e diminua o tempo gasto para o desenvolvimento de especificações detalhadas.

1.2 Objetivos

O objetivo deste trabalho é propor uma abordagem para facilitar o desenvolvimento de Casos de Uso de qualidade baseada no conceito de Fragmentos de Caso de Uso. Além de facilitar a escrita, a abordagem apresenta as vantagens de auxiliar o aprendizado de especificação de requisitos de qualidade além da padronização de sua representação.

A viabilidade da proposta foi verificada com dois experimentos envolvendo alunos do curso de graduação em Ciência da Computação da Universidade Federal do Rio de Janeiro (UFRJ). Os resultados do experimento possibilitam uma comparação inicial da abordagem proposta neste trabalho em relação à técnica artesanal de escrita de Casos de Uso.

1.3 Definição do Escopo do Trabalho

Durante a elaboração de especificações de Casos de Uso para sistemas de informação comerciais, é comum encontrar algumas seqüências recorrentes de interações (JACOBSON, NG, 2004). A seleção de um objeto a partir de uma lista, juntamente com alguns fluxos alternativos e troca de informações, é um exemplo de uma seqüência de interações que ocorre recorrentemente em descrições de Casos de Uso.

Este trabalho apresenta uma abordagem baseada em Fragmentos de Casos de Uso. Um Fragmento de Caso de Uso é definido como sendo a estrutura comum de interações entre o sistema e seus atores, incluindo os fluxos alternativos, focados em atingir um sub-objetivo genérico. Tal estrutura contempla detalhes as diversas partes do texto de um Caso de Uso, como os fluxos, as regras e as estruturas de dados. A partir da disponibilidade de um catálogo

de Fragmentos de Casos de Uso, coletados a partir de especificações recorrentes de diversos sistemas de informação, os alunos e profissionais principiantes poderiam reutilizar trechos de Casos de Uso. Os Fragmentos podem ser utilizados em um processo de composição, contemplando boa parte das descrições de Casos de Uso encontradas em sistemas de informação comerciais.

A elaboração de um Caso de Uso utilizando Fragmentos é análoga ao uso de Padrões de Projeto (*Design Patterns*) e Padrões de Casos de Uso (*Use Case Patterns*), para resolver um problema recorrente de projeto de software e de projeto de Caso de Uso, respectivamente. Essas abordagens permitem que analistas e projetistas não experientes terminem suas tarefas com produtos de maior qualidade. Assim, Fragmentos de Casos de Uso foram criados a partir de boas práticas de escrita de Casos de Uso para serem reutilizados em situações semelhantes de aplicação.

1.4 Delimitação

As tarefas anteriores à escrita do Caso de Uso (como a identificação do escopo do sistema de informação, a identificação dos Casos de Uso, a identificação dos Atores que interagem com o sistema) não são contempladas pela abordagem apresentada deste trabalho. O uso dessa abordagem assume que um modelo inicial contendo Atores, Casos de Uso e seus objetivos principais esteja disponível como resultado de atividades anteriores de elicitação de requisitos, devido à importância de entender apropriadamente o ambiente organizacional e os objetivos gerais. O arcabouço *i**, apresentado por Yu (1995), é um exemplo de tal técnica, que provê um entendimento dos relacionamentos da organização no domínio do negócio.

Portanto, assume-se que os profissionais já possuam um entendimento razoável a respeito do ambiente organizacional e das necessidades das partes interessadas e também que um modelo inicial de Casos e Uso já tenha sido produzido. Seguindo a abordagem descrita

por Santander e Castro (2002), vemos a descrição de Casos de Uso como um refinamento de um modelo inicial.

Embora essas outras abordagens tenham como resultado o entendimento da organização e das necessidades de melhorias, o modelo inicialmente produzido de requisitos contempla apenas uma visão superficial da solução. Esta dissertação busca apresentar uma contribuição para o detalhamento dos Casos de Uso, cujo pré-requisito é o modelo inicial.

1.5 Contribuições

Como principal contribuição deste trabalho, podemos destacar a produção de um Catálogo de Fragmentos de Casos de Uso preliminar, que pode ser utilizado por estudantes e profissionais principiantes para aumentar a qualidade e diminuir o tempo da escrita de Casos de Uso.

Outra contribuição deste trabalho consiste nas diretivas resultantes da análise dos experimentos sobre a utilização de Fragmentos. Tal experimento provê um conjunto de evidências iniciais sobre as vantagens e oportunidades de melhoria da abordagem, assim como das situações onde o uso de Fragmentos não traz benefícios consideráveis.

Este trabalho originou dois trabalhos que foram publicados em anais de congressos. Durante as primeiras etapas de definição do trabalho, foi publicado o artigo Dias *et al* (2007), cuja apresentação possibilitou discussões que contribuíram positivamente para a continuidade da investigação. Posteriormente, a proposta se tornou mais consistente e foi apresentada no artigo Dias *et al* (2008). As idéias e definições utilizadas em ambos os artigos forneceram a base necessária para a elaboração desta dissertação.

1.6 Organização da Monografia

Este trabalho está dividido em seis capítulos. Este primeiro capítulo introduz o contexto e a motivação do trabalho, descrevendo o problema e os objetivos da abordagem, a fim de

encontrar uma solução para as questões apresentadas, incluindo também as contribuições do trabalho realizado. No segundo capítulo são apresentados o referencial teórico e a revisão de literatura associados à engenharia de requisitos e à técnica de Casos de Uso.

A abordagem proposta é discutida em detalhes no capítulo três, onde é incluído um exemplo de utilização dessa abordagem. No quarto capítulo descrevem-se os experimentos realizados para evidenciar a viabilidade do uso da solução proposta. Esse capítulo apresenta também os dados obtidos através do experimento.

O capítulo cinco apresenta uma análise dos resultados desses experimentos e discute a efetividade da abordagem quanto às metas pretendidas a partir do experimento do capítulo quatro. Finalmente, o capítulo seis contempla a conclusão do trabalho, apresentando as contribuições, limitações da abordagem e sugestões de trabalhos futuros.

Esta dissertação inclui também quatro apêndices e um anexo. O apêndice A apresenta o catálogo de Fragmentos; o apêndice B apresenta o enunciado utilizado no experimento; e os apêndices C e D apresentam os dados dos experimentos um e dois, respectivamente. O anexo A apresenta as orientações para detecção de defeitos em descrições de Casos de Uso, originalmente apresentadas por Cox e Phalp (2000), e uma tradução dessas orientações.

2 CASOS DE USO

2.1 Introdução

Este capítulo aborda alguns assuntos referentes à especificação e elaboração de Casos de Uso. A última seção deste capítulo apresenta informações relacionadas às boas práticas da especificação de requisitos de alta qualidade.

O processo de desenvolvimento de software pode ser conceitualmente dividido em quatro fases: engenharia de requisitos, projeto, implementação e testes (ISO, 1995). A engenharia de requisitos deve prover o real entendimento a respeito do problema para que possa ser desenvolvido o sistema adequado, isto é, que corresponda ao que é efetivamente esperado pelas partes interessadas (SOMMERVILLE, SAWYER, 1997; LARMAN, 2001). Tal entendimento deve ser obtido através de uma investigação a respeito das soluções previamente adotadas e do processo de negócio envolvido (FORTUNE, PETER, 2005).

2.2 Engenharia de Requisitos

Requisitos formam a base para qualquer projeto de software. Os requisitos definem não somente o que as partes interessadas necessitam do sistema como também o que este deve fazer para satisfazer às necessidades dos envolvidos. As partes interessadas podem ser: usuários, clientes, fornecedores, desenvolvedores, especialistas de negócio, dentre outros. É essencial que os requisitos sejam entendidos e validados por todas as partes interessadas e, por essa razão, a linguagem utilizada na descrição dos requisitos deve ser a mais próxima possível da linguagem comum a todos (HULL *et al*, 2005, p.2). Além disso, de acordo com McGovern (2002), o entendimento dos requisitos originados no negócio é fundamental para gerenciamento de projetos de software.

Segundo Sommerville e Sawyer (1997), a engenharia de requisitos implica na utilização sistemática e repetível de uma técnica para garantir que os requisitos do software sejam completos, consistentes e relevantes. Para isso, segundo os mesmos autores, a prática sistemática da engenharia de requisitos requer a realização do seguinte conjunto de atividades: elicitação, análise e validação de requisitos.

A elicitação de requisitos tem por objetivo principal a busca pelos requisitos do sistema, através de várias técnicas, entre as quais podemos citar: entrevista com os interessados, documentos de outros sistemas relacionados, estudo de mercado, conhecimento do domínio, dentre outras fontes de informação. Esta atividade deve ser realizada com cautela, uma vez que a identificação dos reais problemas e das possíveis soluções é considerada uma tarefa difícil de ser realizada. O responsável pela elicitação dos requisitos deve ser capaz de entender os pontos de vista de cada um dos interessados, de forma a habilitar a construção de um consenso a respeito dos requisitos do sistema desejado (GOLDSMITH, 2004).

Uma vez elicitados, os requisitos devem ser especificados e documentados na atividade de análise, já que tais documentos serão utilizados tanto na definição da arquitetura do sistema como na elaboração dos testes de aceitação. A atividade de análise deve contemplar o detalhamento de cada requisito desejado e a negociação a respeito de quais requisitos serão efetivamente considerados ou não. A negociação é necessária devido a diversos fatores, como: o custo de implantação do requisito, a inviabilidade da solução, dentre outros motivos que podem levar ao aceite ou rejeição de um requisito inicial.

Além disso, a análise dos requisitos necessita ser validada pelas partes interessadas, de forma a permitir que seja usada como um contrato, isto é, definindo o que se espera da interação entre o sistema e seus usuários. A consequência natural desse fato é que esses documentos sejam escritos em linguagem natural, permitindo a todas as partes interessadas

entendê-los e avaliá-los (SOMMERVILLE, SAWYER, 1997). De acordo com Tjong *et al* (2007), a linguagem natural ainda é a representação mais utilizada para descrever requisitos, apesar de seus problemas intrínsecos, como ambigüidade, inconsistência e redundância. Há diversos trabalhos que visam à diminuição dos problemas da linguagem natural como o trabalho de Kiyavitskaya *et al* (2007), que apresenta uma técnica para identificar e eliminar ambigüidades em especificações de requisitos, o de Leonardi e Mauco (2004), que apresenta uma integração entre requisitos em linguagem natural e transformações entre modelos, e o de Liu *et al* (2004), que apresenta uma metodologia para automatizar a análise de requisitos em linguagem natural para a geração de modelo de classes.

Essas três atividades são esperadas, de alguma forma, em qualquer processo de engenharia de requisitos em sucedido (SOMMERVILLE, SAWYER, 1997). Além dessas três atividades, um processo de engenharia de requisitos requer um gerenciamento de manutenção de requisitos como suporte. A manutenção dos requisitos é uma atividade necessária para que haja consistência entre o comportamento desejado (funcionalidades) e o comportamento realmente implementado. Os requisitos devem ser atualizados ao longo do projeto, pois é muito comum haver solicitações de alteração de requisitos no decorrer do projeto. A atividade de manutenção dos requisitos possibilita atingir uma característica importante do desenvolvimento de um sistema que é a rastreabilidade entre os requisitos e a sua implementação ao longo da construção de sistemas (S. ROBERTSON, J. ROBERTSON, 2006).

Dentre as três atividades apresentadas, a atividade de análise de requisitos, onde é feita a documentação do detalhamento dos mesmos, é destacada neste trabalho como foco a ser analisado. A especificação de requisitos é uma tarefa desafiadora, devido principalmente à dificuldade de se fazer convergir diferentes pontos de vista de diversos interessados, que

raramente têm a oportunidade de conversar entre si sobre seus problemas relacionados ao novo sistema em questão (ALEXANDER, STEVENS, 2002).

O documento de requisitos deve, inicialmente, ser esboçado como resultado da atividade de elicitación de requisitos. Posteriormente, deve ser refinado, visto que ele deve obrigatoriamente explicitar os detalhes de cada requisito com a precisão suficientemente necessária para o desenvolvimento do sistema (YOUNG, 2004).

Diversos trabalhos foram publicados para auxiliar a especificação de requisitos, com base em boas práticas e exemplos de situações reais. Tais trabalhos incluem a apresentação de conjuntos de orientações para que o texto seja elaborado da melhor forma possível.

Alexander e Stevens (2002) apresentaram algumas orientações simples para a escrita de requisitos. Os autores apontam que, embora tais orientações possam não ser aplicáveis em todos os casos, certamente auxiliam os profissionais iniciantes. Além das orientações, os autores também apresentam exemplos de como não se deve escrever requisitos. Como exemplos dessas orientações, temos:

- Use frases em voz direta;
- Use um vocabulário limitado;
- Identifique o tipo de usuário interessado em cada requisito;
- Foque em estabelecer resultados;
- Defina critérios para verificação; dentre outras.

Um conjunto mais completo e detalhado de orientações foi apresentado por Sommerville e Sawyer (1997). Nesse trabalho, os autores apresentaram um extenso conjunto de orientações, separadas de acordo com os níveis: orientações básicas, orientações intermediárias e orientações avançadas. Dentre as diversas orientações documentadas, destacamos as que foram efetivamente utilizadas no presente trabalho:

1. Defina uma estrutura padrão de documentação;

2. Explique como usar a documentação;
3. Elabore a organização visual do documento para facilitar a leitura;
4. Torne o documento fácil de ser alterado;
5. Use aspectos do negócio para auxiliar a elicitação de requisitos;
6. Defina padrões de *templates* para descrição de requisitos;
7. Use uma linguagem simples e concisa;
8. Auxilie a linguagem natural com outras descrições de requisitos;
9. Use um dicionário de dados;
10. Procure por restrições de domínio.

Essas orientações apresentadas foram utilizadas durante toda a etapa de concepção e elaboração da proposta deste trabalho, o que auxiliou a produção de um trabalho com as características minimamente necessárias de qualidade. As seções seguintes apresentam aspectos relacionados à técnica de Casos de Uso e à documentação de requisitos de software de qualidade.

2.3 Técnica de Casos de Uso

No contexto de documentação de funcionalidades de software, a técnica de Casos de Uso, definida na UML, é a mais popularmente utilizada (KULAK, GUINEY, 2003). De acordo com Eriksson e Penker (2000), o interesse pela UML causou um grande impacto no mercado, o que motivou muitas empresas a investir em treinamento dos seus funcionários nessa técnica e a investir em migração de outras linguagens de modelagem para o uso da UML. Um dos principais motivos para isso foi a utilização de uma linguagem de fácil entendimento por parte de analistas de requisitos e pelos clientes, o que facilita a validação dos documentos de requisitos (FOWLER, 2003). A representação gráfica dessa técnica auxilia a visualização da interação entre os atores e os casos de uso, como mostra a Figura 1.

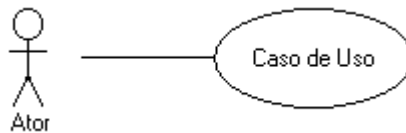


Figura 1. Um diagrama de Casos de Uso contendo um ator e um Caso de Uso.

Uma vantagem do uso dessa técnica é a sua capacidade de participar de diversas visões diferentes sobre a arquitetura do sistema. Durante a especificação de requisitos, os Casos de Uso auxiliam as visões: lógica, de implementação, de implantação e de processos, como representado na Figura 2.

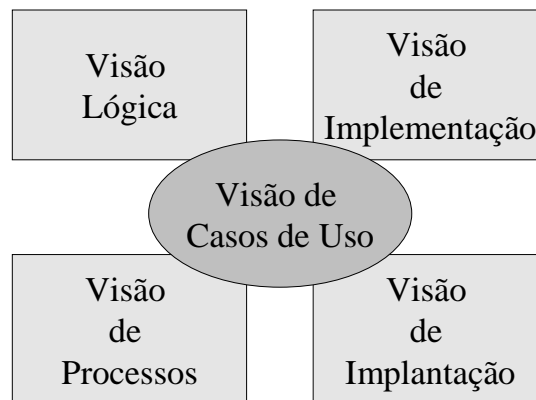


Figura 2. Visões relacionadas a Casos de Uso (MARTINS, 2007)

Como apresentado por Martins (2007) e por Knichten (1995), a visão lógica está associada aos requisitos funcionais do sistema e descreve as principais funcionalidades e interações. A visão de implementação consiste em componentes, códigos-fonte, banco de dados e artefatos computacionais diversos. A visão de processos fornece os detalhes a respeito do fluxo de trabalho da organização, de forma que os requisitos indiquem como o sistema deve auxiliar os processos. A visão de implantação indica a forma como o sistema deverá adequar-se em seu ambiente de instalação e desempenho.

Os Casos de Uso interagem com todas essas visões, tanto fornecendo informações para algumas como obtendo de outras. Como os Casos de Uso explicitam as interações desejáveis, explicitam a lógica de comportamento do sistema. Entretanto, para que tal lógica possa ser entendida, é necessário um conhecimento prévio da visão de processos, de forma

que sejam entendidas todas as etapas da cadeia produtiva da empresa e sejam determinados os pontos que podem ser auxiliados por computadores. Os Casos de Uso serão também importantes para a definição do projeto arquitetural do software e de seus artefatos. O arquiteto de software deve realizar seu trabalho sempre de acordo com os requisitos especificados. Por último, os Casos de Uso permitem uma visão dos pacotes entregáveis do software, facilitando a divisão do software em partes, ou módulos, para que a ordem de implementação possa ser definida pela equipe de projeto do software, como discutido no trabalho de Schmitz *et al* (2008).

A linguagem natural utilizada nos documentos de Casos de Uso facilita a interação com especialistas do negócio. O conhecimento de tais interessados é primeiramente documentado na forma de processos de negócio e, posteriormente, em documentação de requisitos do sistema, sendo os processos a ponte entre o negócio e os sistemas de informação (BALDAM *et al*, 2007).

Esse alinhamento entre a linguagem do negócio e a dos requisitos é possível através da técnica de Casos de Uso, uma vez que tal técnica expressa os requisitos do negócio utilizando a linguagem do negócio. A Figura 3 apresenta como o modelo de Casos de Uso interage com os demais modelos de software. Destacamos o relacionamento entre Casos de Uso e o vocabulário do negócio, mostrando que a técnica utiliza os termos do domínio em sua descrição. O vocabulário do negócio deve ser utilizado explicitamente durante a descrição dos Casos de Uso para que possa representar os reais objetos de negócio.

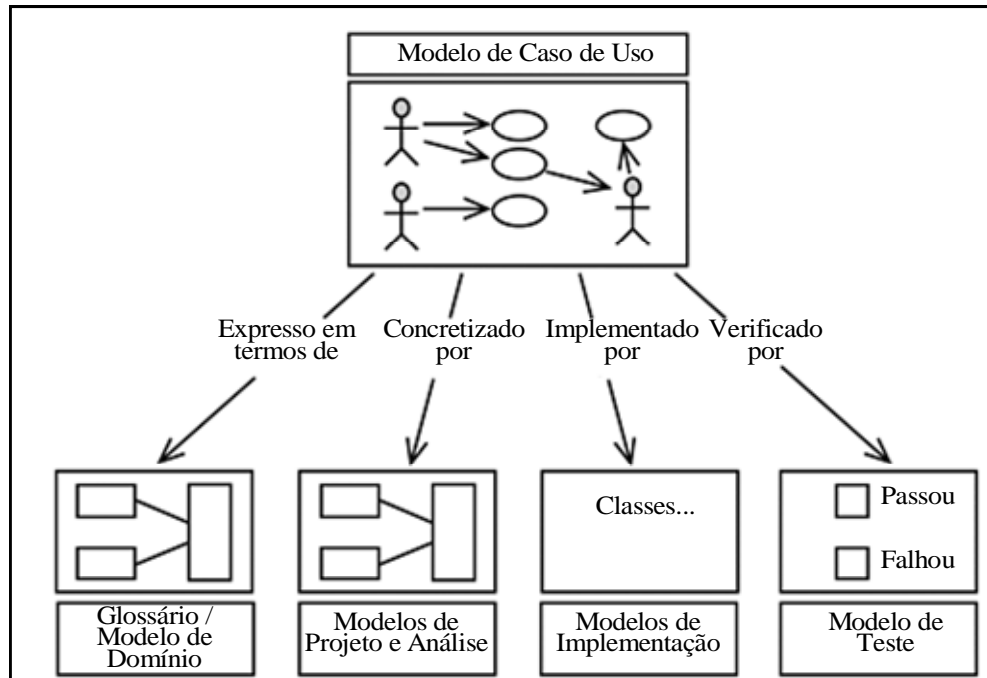


Figura 3. Relacionamentos entre Casos de Uso e outros modelos, traduzidos do trabalho de Bittner e Spence (2002)

Os Casos de Uso podem ser aplicados de forma a captar o comportamento pretendido do software sem haver a necessidade de se explicitar como esse comportamento será implementado (BOOCK *et al*, 2005). Todo Caso de Uso interage com atores, onde estes são todos os elementos externos ao sistema, como usuários humanos, outros sistemas, dispositivos externos, dentre outros.

Um Caso de Uso representa um requisito funcional do sistema e sua descrição deve especificar todas as interações entre os atores e o sistema. Essas interações devem mostrar tanto o comportamento do usuário como o comportamento do sistema durante a troca de informações entre ambos (JACOBSON, 1992).

Como o documento que descreve um Caso de Uso é um documento de requisitos que deve ser entendido por todos os interessados, a linguagem utilizada na sua especificação deve ser a linguagem natural, e os termos utilizados devem ser de entendimento de todos. O responsável pelo Caso de Uso deve escrever todas as partes de forma que o texto seja legível e compreensível por todos os interessados. Dentre as partes de um Caso de Uso, destacamos: fluxo de eventos principal, fluxo de eventos alternativos, quando o Caso de Uso começa e

termina, quando o Caso de Uso interage com os atores e quais objetos são transferidos (BOOCH *et al*, 2005). Diversos autores apresentam partes estruturadas para a descrição de Casos de Uso. A Figura 4 apresenta as propriedades de uma descrição completa de Caso de Uso a partir da visão de Bittner e Spence (2002), que utilizaremos neste trabalho.

Ao descrever os fluxos de eventos, devem ser consideradas as diferentes e possíveis condições. O fluxo básico deve conter o fluxo de eventos esperados na maioria dos casos. A partir da descrição do fluxo básico, devem ser considerados todos os casos de escolhas alternativas e de exceções em relação aos eventos principais. Tais casos devem ser explicitados e descritos, de modo que todas as possíveis condições desejadas possam ser estudadas, analisadas, validadas e, posteriormente, implementadas (LEFFINGWELL, WIDRIG, 2003).

Nome da Propriedade	Breve Descrição
Nome	O nome do Caso de Uso. Cada Caso de Uso deve ter um nome que indique o que é alcançado através das suas interações com o(s) ator(es). O nome ter algumas palavras para ser entendido. Dois Casos de Uso não devem ter um mesmo nome.
Breve Descrição	Uma breve descrição do papel e propósito do Caso de Uso.
Fluxo de Eventos	Uma descrição textual do que o sistema deve fazer em consideração ao Caso de Uso (não como problemas específicos são solucionados pelo sistema. A descrição deve ser inteligível pelas partes interessadas. O fluxo dos eventos é estruturado em fluxo básico, fluxos alternativos e sub-fluxos.
Requisitos Especiais	Uma descrição textual a respeito de todos os requisitos, como requisitos não-funcionais, do Caso de Uso que não são considerados nos fluxos de eventos, mas que devem ser considerados nas etapas seguintes de projeto e implementação de software.
Pré-Condições	Uma descrição textual que define restrições do sistema quando o Caso de Uso inicia.
Pós-Condições	Uma descrição textual que define restrições do sistema quando o Caso de Uso tiver terminado.
Pontos de Extensão	Uma lista de locais nos fluxos de eventos do Caso de Uso nos quais um comportamento adicional pode ser inserido.
Relacionamentos	Os relacionamentos, assim como os relacionamentos de comunicação, nos quais o Caso de Uso participa.
Diagramas	Diagramas que ilustrem os aspectos do Caso de Uso, tais como a estrutura do fluxo de eventos ou os relacionamentos envolvendo o Caso de Uso.

Figura 4. Propriedades da descrição de um Caso de Uso, traduzidas da visão de Bittner e Spence (2002)

Em relação aos fluxos alternativos, estes devem descrever tanto os casos em que houver um evento alternativamente desejado por uma interação de um ator como os casos em que houver uma exceção no fluxo básico (BITTNER, SPENCE, 2002). Na Figura 5, é apresentado visualmente o relacionamento entre o fluxo básico de eventos e seus fluxos alternativos. Na descrição dos fluxos alternativos deve ser especificado como o sistema deve interagir com os atores e como eles retornam ao fluxo básico ou terminam o Caso de Uso (ROSENBERG, STEPHENS, 2007).

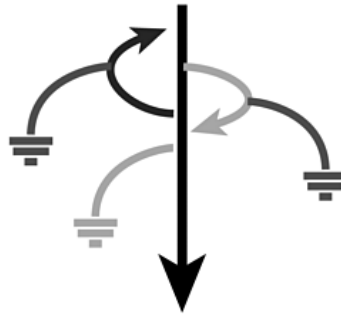


Figura 5. Fluxo básico principal com seus fluxos alternativos (BITTNER, SPENCE, 2002)

Apesar de haver diversos livros e artigos publicados sobre a técnica de Casos de Uso, a sua escrita ainda não é padronizada pela linguagem UML, segundo Morgan (2002). A UML padroniza apenas a notação gráfica do diagrama de Casos de Uso, mas não o conteúdo dos mesmos. Como consequência, o trabalho de escrita de um Caso de Uso torna-se uma tarefa artesanal e subjetiva, como apresentado por Melo (2006) e por Belgamo e Fabbri (2004). Outros trabalhos, como o de Briand *et al* (2005), buscam verificar e propor melhorias à escrita estruturada de Casos de Uso.

2.4 Etapas de Especificação de Casos de Uso

As atividades de engenharia de requisitos apresentadas por Sommerville (1997) são aplicáveis a qualquer abordagem de requisitos, ou seja, não estão vinculadas a nenhuma técnica específica para a engenharia de requisitos de software. Em relação à técnica de Casos de Uso, as atividades de elicitação e documentação são detalhadas em fases mais refinadas, como mostra o trabalho de Windle e Abreo (2002). Os autores apresentaram um modelo para o processo de engenharia de requisitos composto por quatro fases: elicitar, analisar, especificar e verificar. Nessa abordagem, porém, não se encontra a validação dos interessados, sendo abordadas apenas as fases necessárias para a produção do texto do Caso de Uso. A Figura 6 apresenta as fases do processo e as possibilidades de mudança dos requisitos ao longo das fases.

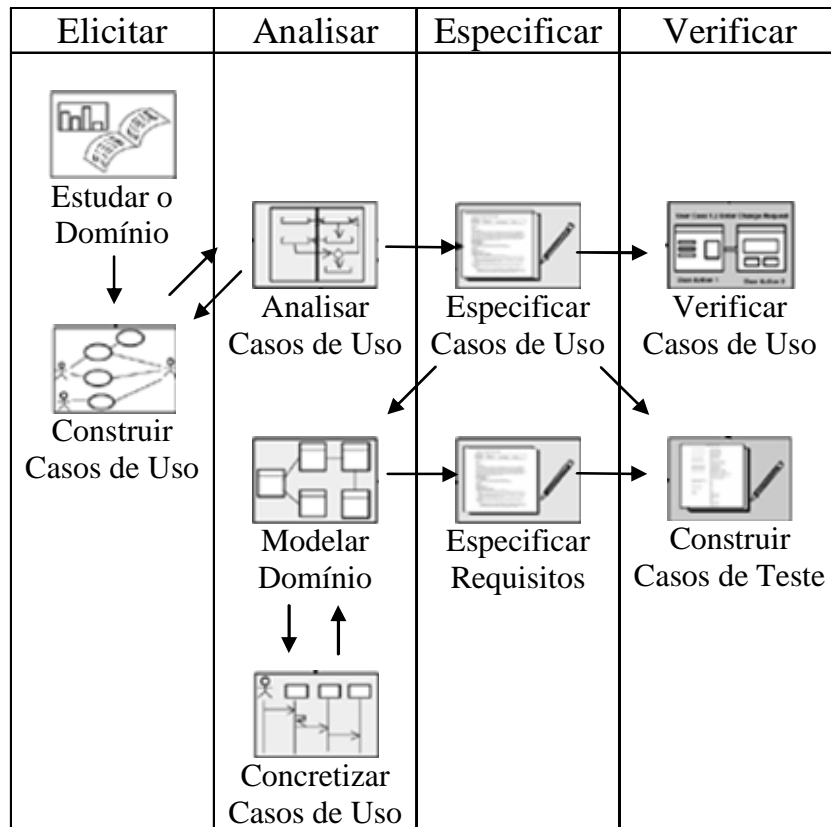


Figura 6. Fases do processo de engenharia de requisitos, traduzidas da visão de Windle e Abreo (2002)

A fase de elicitação envolve o estudo do ambiente do usuário, resultando na elaboração de um documento contendo uma descrição de alto nível dos Casos de Uso e de seus atores. Os atores devem representar tanto os usuários do sistema como os demais sistemas com os quais o sistema principal deve interagir (WAZLAWICK, 2004).

A fase de análise das principais funcionalidades do software torna possível a identificação dos componentes necessários à solução. Tais componentes devem ser formalizados em diagramas de classes, de forma a explicitar os relacionamentos e atributos de cada entidade de negócio.

A terceira etapa contempla a especificação dos Casos de Uso e dos requisitos de software. Os autores sugerem que a especificação dos Casos de Uso deve ser detalhada, de forma a contemplar as interações entre os atores e os Casos de Uso e adicionalmente sejam

documentados também alguns detalhes de interface adicionais para os Casos de Uso existentes.

A fase de verificação se inicia com a verificação dos textos dos Casos de Uso e a elaboração dos casos de teste. Os textos dos Casos de Uso devem ser verificados em relação aos documentos inicialmente produzidos a respeito da organização e dos objetivos dos Casos de Uso. Os casos de teste são úteis para verificar se os requisitos realmente indicam como o software deve se comportar quando recebe alguns estímulos específicos.

A descrição de Casos de Uso passa por dois estados distintos: no primeiro, cada Caso de Uso é identificado e tem seus objetivos brevemente descritos; o segundo inclui todas as interações necessárias para descrever o comportamento do software. Nesta dissertação estamos interessados na etapa de especificação onde os Casos de Uso, já devidamente identificados e com seus objetivos definidos, devem ter seu conteúdo detalhado.

Bittner e Spence (2002) recomendam que os Casos de Uso não sejam apresentados de uma forma hierárquica, em níveis de detalhamento, visto que estes não possuem uma arquitetura textual que possa ser rerepresentada em diversos níveis. Além disso, o uso de níveis de decomposição pode induzir a incursões no nível de projeto de software, o que não deve ocorrer na fase de especificação de requisitos. Os Casos de Uso devem capturar a descrição do que o sistema deve fazer e conter todos os detalhes necessários para explicitar seu comportamento. Além disso, devem ser escritos numa linguagem simples e sem ambigüidade.

Os detalhes do comportamento do sistema, incluídos no Caso de Uso, devem ser obtidos como produto das fases anteriores de elicitação e análise, as quais detalham os processos do negócio, as entidades e as regras de negócio. Usando esse conhecimento e uma lista de Casos de Uso, o analista de requisitos tem a possibilidade de explicitar seu

entendimento quanto às funcionalidades do software (PREECE *et al*, 2005; BITTNER, SPENCE, 2002).

2.5 Qualidade de Casos de Uso: propostas de boas práticas e uso de padrões

Cockburn (2001) define um Caso de Uso com qualidade como sendo aquele que seja aceitável para seu propósito, que é o de especificar precisamente os requisitos desejados. Esse mesmo trabalho mostra uma série de boas práticas para a escrita de Casos de Uso, sempre baseado em exemplos e contra-exemplos práticos.

Dando continuidade a esse mesmo trabalho, Cockburn *et al* (2002) apresentaram um conjunto de padrões para a documentação de Casos de Uso. Os autores admitem que não há um critério absoluto que possa ser utilizado para diferenciar um Caso de Uso de boa ou de má qualidade. Entretanto, garantem que o catálogo de padrões por eles elaborado pode ser utilizado tanto para a construção de Casos de Uso de qualidade como para avaliação de documentos existentes. Tais padrões são aplicáveis a passos, cenários ou até mesmo a um Caso de Uso completo. Esses padrões abordam diversos tipos de problemas recorrentes tais como: nomeação de Casos de Uso, criação de muitos fluxos alternativos, passos em níveis, relacionamentos entre Casos de Uso, dentre outros.

Um exemplo desses padrões de Cockburn *et al* (2002), é o padrão “NomeFraseVerbal” (*VerbPhraseName*). Esse padrão é indicado para a situação em que o objetivo seja único e bem determinado. Sua aplicação requer que o Caso de Uso seja nomeado com um verbo seguido de um substantivo, representando o objetivo do ator ao interagir com esse Caso de Uso. Os autores apresentam o nome “Assegurar Crédito” como exemplo e “Caso de Uso Principal”, “Processo Principal”, “Caso de Uso 2” e “Assuntos do Processo” como contra-exemplos. O mesmo padrão indica que é possível utilizar uma numeração no nome dos Casos

de Uso, mantendo a formatação do padrão, como por exemplo: “Caso de Uso 42: Aprovar Danos”.

O problema de erros em documentos de requisitos motivou também o trabalho de Resenberg e Scott (2001), que apresentaram uma lista contendo os dez erros mais comuns em documentos de Casos de Uso e suas respectivas conseqüências. O trabalho apresenta ainda formas de resolver tais problemas, mostrando as situações onde comumente ocorrem e que não são percebidos pelo analista de requisitos. A lista dos erros mais comuns encontra-se na Figura 7.

# 10	Listar os requisitos funcionais no lugar da descrição textual
# 9	Descrever atributos e métodos ao invés do comportamento do sistema
# 8	Escrever um texto excessivamente conciso
# 7	Ocultar completamente os detalhes dos elementos de negócio, para não explicitar a interface
# 6	Nomear objetos que norteiam a solução, como: tela, janela, <i>pop-up</i>
# 5	Escrever numa perspectiva que não é a do usuário, em voz passiva
# 4	Escrever apenas os estímulos do usuário, esquecendo-se das respostas do sistema
# 3	Omitir o texto para os fluxos alternativos
# 2	Focar em detalhes que não interessam ao caso de uso, por exemplo “como obter uma informação” ou “o que acontece depois”
# 1	Gastar um mês para decidir onde usar inclusão ou extensão

Figura 7. Lista dos dez principais erros encontrados em descrições de Casos de Uso, traduzida do trabalho de Resenberg e Scott (2001)

O trabalho de Övergaard e Palmkvist (2004) apresenta também um catálogo de padrões de Caso de Uso e um conjunto de erros mais comuns. Os padrões de Casos de Uso devem ser utilizados tanto para melhorar o detalhamento de Casos de Uso existentes como para contribuir como fonte de aprendizado para o desenvolvimento de melhores Casos de Uso futuramente. Como exemplos de padrões documentados, temos: múltiplos atores, transferência de mensagens, geração de relatórios, dentre outros. Os autores explicam que os

padrões foram elaborados através da observação de situações recorrentes onde o uso dos padrões propostos tornou o documento de Casos de Uso mais atualizável, compreensível e reutilizável, e com isto, melhorando a qualidade da documentação de requisitos.

Como exemplo do trabalho de Övergaard e Palmkvist (2004), apresentamos o padrão de Caso de Uso chamado “Regras de Negócio”. Sua intenção é extrair informações de regras de negócio a partir dos fluxos de Casos de Uso existentes e descrever essas informações como uma coleção de regras de negócio referenciada a partir dos Casos de Uso originais. Esse padrão pode ser utilizado em soluções estáticas ou dinâmicas. A solução estática do padrão deve ser aplicada quando as regras não necessitam ser alteradas em tempo de execução. Essa solução requer que as regras estáticas sejam migradas para um documento à parte e, onde eram utilizadas, haja apenas uma referência à regra do novo documento. A solução dinâmica deve ser aplicada quando as regras podem ser alteradas em tempo de execução. Essa solução requer a criação de um novo Caso de Uso chamado “Gerenciar Regras”, onde as regras de negócio podem ser criadas, buscadas, alteradas e removidas.

Os trabalhos mencionados nesta seção focam em como o uso de experiências e boas práticas pode contribuir para uma melhoria da qualidade do trabalho de especialistas de requisitos. Entretanto, como explicado por Melo (2006), essas práticas não resolvem as dificuldades encontradas por alunos e profissionais iniciantes, visto que estes não possuem a experiência para iniciar um detalhamento de requisitos.

A dificuldade em escrever Casos de Uso de alta qualidade é discutida em Jagielska *et al* (2006). Neste trabalho, os autores apresentam os resultados de um experimento realizado com alunos de graduação a respeito de defeitos em documentos de Casos de Uso. Os defeitos, comumente encontrados em documentos de analistas de sistemas, dificultam a especificação precisa dos requisitos do sistema, fazendo com que o documento perca o seu propósito.

Cox *et al* (2004) apresentam um *checklist* destinado a encontrar defeitos em Casos de Uso e, com isto, avaliar a qualidade do texto de Casos de Uso. A elaboração desse *checklist*, bastante completo, objetivo e abrangente, foi baseada na avaliação de um extenso conjunto de trabalhos publicados com esse mesmo objetivo.

Cada defeito encontrado num Caso de Uso possui diferentes conseqüências para o processo de engenharia de requisitos. Os defeitos foram categorizados por Cox *et al* (2004) de acordo com o tipo de impacto:

- Impacto na descrição do Caso de Uso – **Defeito de Caso de Uso**: causa efeitos em relação à leitura e à ordenação dos passos de um Caso de Uso, contemplando os defeitos de uso incorreto de tempos verbais, adjetivos, numeração de fluxos, dentre outros;
- Impacto de especificação – **Defeito de Especificação**: relacionado à mistura incorreta de requisitos com níveis de detalhe que não fazem parte de um documento de requisitos, tais como detalhes de implementação, de interface visual, de arquitetura etc.;
- Impacto de requisitos – **Defeito de Requisito**: relacionados ao não entendimento do real problema e necessidade dos interessados. O documento contém defeitos que levam a construção de sistemas que não atendem aos objetivos dos usuários e que se comportem de modo indesejado pelos interessados.

O trabalho de Cox *et al* (2004) foi utilizado como base para o experimento descrito no capítulo 4 desta dissertação. O Anexo A desta dissertação inclui uma tradução do *checklist* para português.

O trabalho de Biddle *et al* (2002) apresenta alguns padrões para a escrita de Casos de Uso Essenciais. Casos de Uso Essenciais foram originalmente apresentados por Constantine e Lockwood (1999), onde são definidos como um Caso de Uso menos detalhado com narrativa

estruturada, dividindo-se as responsabilidades dos atores e do sistema. O texto segue o formato de duas colunas: uma para a ação do usuário e outra para a resposta do sistema, sem que haja um texto escrito em frases. A Figura 8 mostra um exemplo de um Caso de Uso Essencial, elaborado por Biddle *et al* (2002).

Ação do Usuário	Resposta do Sistema
identificar-se	verificar identificação oferecer opções
escolher	
pegar dinheiro	liberar dinheiro

Figura 8. Exemplo de conteúdo de um Caso de Uso Essencial “Retirar Dinheiro”

Além de mostrar alguns padrões genéricos de Casos de Uso Essenciais, o trabalho também apresenta dois padrões específicos para escrita de passos de Casos de Uso Essenciais: pergunta e confirmação. Tais padrões não fixam a forma de escrita, mas sim indicam como deve ser feito um Caso de Uso Essencial que contenha uma pergunta ou uma confirmação. O primeiro padrão é indicado quando houver a necessidade do sistema requisitar uma informação ao usuário. O segundo padrão sugere a requisição de uma confirmação do usuário sempre que uma operação importante deva ser executada, por exemplo, no pagamento de algum produto. Embora esses padrões ofereçam um guia para o desenvolvimento de requisitos de software de modo claro e objetivo, deixam de fornecer um detalhamento preciso da interação do ator com o sistema, omitindo informações importantes, tais como regras de negócio e os objetos de negócio envolvidos nas operações do sistema.

Uma segunda abordagem foi sugerida pelo grupo CREWS (COX, PHALP, 2000). Esse trabalho apresenta um conjunto de diretrizes para o desenvolvimento de Casos de Uso, separadas em dois tipos: diretrizes de estilo e diretrizes de conteúdo.

As seis diretrizes de estilo apresentam uma estrutura padrão para facilitar a leitura dos requisitos. Essas diretrizes de estilo são:

- SG1: Escreva o Fluxo Básico do Caso de Uso como uma lista de ações na forma: <ação#> <descrição da ação>. Cada descrição de ação deve iniciar em uma nova linha. Uma vez que cada ação é atômica, evite frases com mais de duas orações;
- SG2: Use uma ordenação seqüencial das descrições de ação (e também seus identificadores numéricos únicos) para indicar uma seqüência restrita entre ações. CREWS impõe um significado preciso na ordenação de ações em sua lista. Variações devem ser escritas em uma seção separada;
- SG3: Interações e ações concorrentes podem ser expressas numa mesma seção do Caso de Uso, enquanto que ações alternativas devem ser escritas numa outra seção diferente;
- SG4: Use agentes, objetos e nomes de ação consistentes ao longo de toda a descrição de um Caso de Uso. Evite o uso de sinônimos e homônimos, e referências como “ele”, “ela” e “eles”. Seja consistente no uso da terminologia;
- SG5: Use o presente do indicativo e a voz ativa quando descrever ações;
- SG6: Evite o uso de negações, advérbios e verbos auxiliares na descrição de uma ação.

As oito diretrizes de conteúdo apresentam um *template* para fixar a escrita de passos do Caso de Uso. Cada passo deve ser escrito de acordo com um dos *templates*, substituindo-se algumas partes do texto para o contexto do Caso de Uso. As diretrizes de conteúdo são:

- CG1: <agente> <ação de movimentação> <objeto> de <origem> para <destino>;
- CG2: <agente de origem> <ação de posição> <objeto> em <agente de destino>;
- CG3: <agente de destino> <ação de pegar> <objeto> de <agente de origem>;
- CG4: <agente> <ação> <agente>;
- CG5: <agente> <ação> <objeto>;
- CG6: Se <condição alternativa> então <ação>;

- CG7: Enquanto <condição repetitiva> faça <ação>;
- CG8: <ação 1> enquanto <ação 2>.

Ridao *et al* (2001) apresentam um trabalho sobre documentação e uso de cenários, independentes de domínio, recorrentemente encontrados na escrita de requisitos. A técnica de escrita baseada em cenários visa à garantia do entendimento e efetiva colaboração entre as partes interessadas no processo de definição dos requisitos. Cenário é apresentado pelos autores como uma narrativa de uma situação que contém uma estrutura parcialmente escrita em linguagem natural, tornando-o compreensivo por qualquer um dos participantes.

Ridao *et al* (2001) também apresentam um conjunto de padrões de cenários encontrados em estudos de casos no Brasil e na Argentina. Tais padrões são: Produção, com somente um ator realizando a atividade; Colaboração e Serviço, que envolvem dois ou mais atores em atividades que podem ser colaborativas; e Negociação, que é uma seqüência de ações coordenadas.

Esses padrões de cenários facilitam o entendimento do problema e o desenvolvimento da respectiva solução. Um exemplo de padrão de cenário é apresentado na Figura 9, onde são especificados os atributos que a solução deve conter.

Negociação completa com produção	
Título:	Execução de uma atividade
Objetivo:	Executar uma atividade, incluindo atividades de produção
Contexto:	Localização geográfica, pré-condições
Atores:	Diversos, pelo menos dois
Recursos:	Pelo menos um, normalmente diversos
Episódios:	Narrativa de interação entre os atores participantes
Exceção:	Circunstâncias que previnem a execução do objetivo

Figura 9. Tradução de um exemplo de um padrão de cenário

Os dois primeiros trabalhos citados nesta seção, Biddle *et al* (2002) e Cox e Phalp (2000), focam na estrutura do texto de passos individuais de Casos de Uso e provêm pouco auxílio à documentação das interações entre o sistema e os atores. A terceira abordagem, apresentada por Ridao *et al* (2001), baseada em cenários, fornece padrões de cenários independentes de domínio que são úteis para cenários semelhantes, não fornecendo os detalhes necessários a um Caso de Uso detalhado.

Os três trabalhos citados foram importantes fontes de informação, construindo importantes fundamentos para o desenvolvimento desta dissertação. A abordagem de Fragmentos de Casos de Uso, tema deste trabalho, é apresentada como uma solução que inclui a escrita de passos bem estruturados, como os dois primeiros trabalhos, e considerando um contexto específico independente de domínio, como o terceiro trabalho.

3 FRAGMENTOS DE CASOS DE USO

3.1 Introdução

Este capítulo apresenta os principais conceitos relacionados com a abordagem de Fragmentos de Casos de Uso. Primeiramente, serão apresentadas as idéias e definições utilizadas na elaboração do catálogo de Fragmentos e, posteriormente, o uso dos Fragmentos é explicado e exemplificado.

3.2 Desenvolvimento da Pesquisa


Esta dissertação teve início examinando a transformação de modelos de negócio em modelos de especificação de sistemas, como Casos de Uso e classes do sistema (DIAS *et al*, 2006). Esse trabalho apresenta uma abordagem que auxilia a elaboração de parte dos requisitos de software a partir de informações de negócio, como processos e definição de termos. Entretanto, tal trabalho limita-se em gerar um diagrama de Casos de Uso, sem fornecer auxílio para a escrita do seu conteúdo. Outras abordagens com objetivos semelhantes, como as de Dijkman e Joosten (2003) e Oscar (2004), também não auxiliam o detalhamento dos Casos de Uso. A abordagem de Santander e Castro (2002) fornece alguns detalhes para a elaboração dos Casos de Uso, o que requer um detalhamento da organização e seus objetivos utilizando a técnica i*.

A dificuldade em desenvolver os Casos de Uso gerados, por parte de alunos e profissionais inexperientes, motivou a pesquisa a respeito de técnicas que ajudassem a conduzir o detalhamento de requisitos de software com alta qualidade. Tal conhecimento deveria ter origem a partir de boas práticas na escrita de Casos de Uso.

Através de um processo de observação empírica de descrições de Casos de Uso, foram extraídos os passos que individualmente ocorriam com frequência. Esses passos foram listados e catalogados na forma de uma documentação. Utilizamos essa documentação como forma de registro e sistematização de dados para possibilitar a análise por parte da pesquisa, como sugerido por Severino (2007).

A documentação de passos foi avaliada através de um procedimento informal com uma turma de quatorze alunos de Mestrado em Informática. A tarefa solicitada foi utilizar a lista de passos para compor as descrições de um conjunto de Casos de Uso como exercício de aula. Como resultado obtido, destacamos a dificuldade de encontrar os passos desejados na lista e de utilizar sistematicamente tais passos. Os alunos também sugeriram melhorias para a qualidade do texto dos passos e indicaram os pontos positivos da abordagem.

O passo seguinte da pesquisa teve como objetivo buscar uma forma para facilitar o uso da lista de passos. Para isso, três páginas de grandes sistemas na internet foram analisadas do ponto de vista do usuário: o Sistema Integrado de Gestão Acadêmica (SIGA²) da UFRJ, a página da Amazon³ de compra e venda de produtos na internet e a página do Banco do Brasil⁴. As principais interações do usuário com as interfaces dessas páginas foram observadas e listadas. A lista de interações possibilitou verificar a ocorrência recorrente de algumas interações, tais como: selecionar um ou mais objetos em listas; informar parâmetros de busca para que o sistema responda com um conjunto de informações; solicitar o envio de arquivos.

² <http://intranet.ufrj.br/>  SIGA (acesso restrito a alunos, professores e funcionários da UFRJ)

³ <http://www.amazon.com/>

⁴ <http://www.bb.com.br/> (acesso restrito a clientes do Banco do Brasil)

As interações mais freqüentes do usuário com o sistema deveriam ser descritas como requisitos de interação com tais sistemas. Por isso, os passos foram agrupados para formar um ou mais Fragmentos de Casos de Uso, onde cada Fragmento tem por finalidade satisfazer a um sub-objetivo do ator em relação ao sistema. O objetivo principal é aquele que será atingido ao término da execução do Caso de Uso como um todo.

3.3 Definição de Fragmentos de Casos de Uso

De acordo com Jacobson (1992), um Caso de Uso é um modo particular de um ator utilizar o sistema para atender a um objetivo de uma parte interessada e pode ser visto como uma coleção de seqüências de interações entre um sistema e seus atores externos, visando o atingimento de um objetivo. Tal objetivo deve agregar valor para os atores que interagem com o mesmo Svetinovic *et al* (2007).

Para atingir um objetivo, podem ser necessárias diversas etapas menores que, ao serem cumpridas, colaboram para que aproxime-se do objetivo principal. Um objetivo de Caso de Uso pode ser decomposto recursivamente em sub-objetivos, como apresentado por Yu *et al* (2008). Cada sub-objetivo corresponde a funcionalidades que estão abaixo do nível principal de interesse dos usuários interessados. Em descrições de Caso de Uso, sub-objetivos são refinados em uma ou mais interações entre o sistema e seus atores (COCKBURN, 2001).

Em sistemas de informação comerciais, a decomposição de objetivos de Casos de Uso costuma gerar diversos sub-objetivos semelhantes, que podem ser vistos como instâncias de um sub-objetivo mais genérico. Como exemplo de um sub-objetivo genérico, que ocorre de forma recorrente em Casos de Uso distintos, podemos citar a seleção de um elemento a partir de um conjunto de elementos existentes para que o sistema apresente seus detalhes ao ator.

Os sub-objetivos recorrentes podem ser refinados em um conjunto de interações que compartilham uma estrutura comum. Essa estrutura comum inclui não somente as interações

presentes no fluxo básico da descrição de um Caso de Uso como também seus fluxos alternativos. Definimos um Fragmento de Caso de Uso, ou simplesmente Fragmento, como sendo a estrutura comum de interações entre o sistema e seus atores, incluindo os fluxos alternativos, focados em atingir um sub-objetivo genérico. A descrição de um Fragmento lembra a de um Padrão de Projeto (*Design Pattern*), contendo a descrição de uma estrutura genérica que precisa necessariamente ser personalizada para compor uma solução real. Padrões de Projeto são utilizados para os casos em que uma mesma solução possa ser reutilizada outras vezes em situações semelhantes. Como exemplo, destacamos o padrão *Singleton*: tal padrão deve ser utilizado sempre que apenas uma instância do objeto deva existir em tempo de execução. Na descrição da solução, é sugerida uma estrutura de classes para que sempre que for instanciado um novo objeto, seja utilizada a mesma instância, caso já tenha sido instanciada anteriormente. Assim, sempre que surgir um problema semelhante em desenvolvimento orientado a objetos pode ser utilizado o padrão *Singleton* para resolver tal problema, diminuindo o tempo para elaborar uma nova solução e aumentando a padronização das implementações.

A Figura 10 apresenta um modelo do Fragmento e de suas partes. Um Fragmento contém: um nome, um sub-objetivo genérico, um fluxo básico, um conjunto de fluxos alternativos, um conjunto de estruturas de dados e um conjunto de regras. Os fluxos, tanto básico como alternativos, são compostos por passos escritos de acordo com um *template* e com pontos de personalização para serem substituídos por termos do domínio. As estruturas de dados e as regras contêm também o texto em *templates* e os pontos de personalização como os fluxos, porém também são compostos por pontos de explicação. Nos pontos de explicação devem ser explicitados em linguagem natural os detalhes das informações contidas nas estruturas de dados e nas regras. Tais informações podem ser: detalhes de atributos dos objetos, modo de acesso, referências a outros documentos e repositórios de informação e

quaisquer outras informações que devam ser explicitadas. Apesar de as estruturas de dados e as regras terem associação com outros documentos, não são explicitados no modelo tais documentos, pois não pertencem ao escopo de Fragmentos. Caso haja documentos externos, estes devem ser referenciados nos pontos de explicação correspondentes. As multiplicidades utilizadas nas associações de composição indicam: i) “0..*” significa zero ou muitos associados; ii) “1..*” significa pelo menos um ou muitos associados; iii) “sem cardinalidade” significa um e somente um associado.

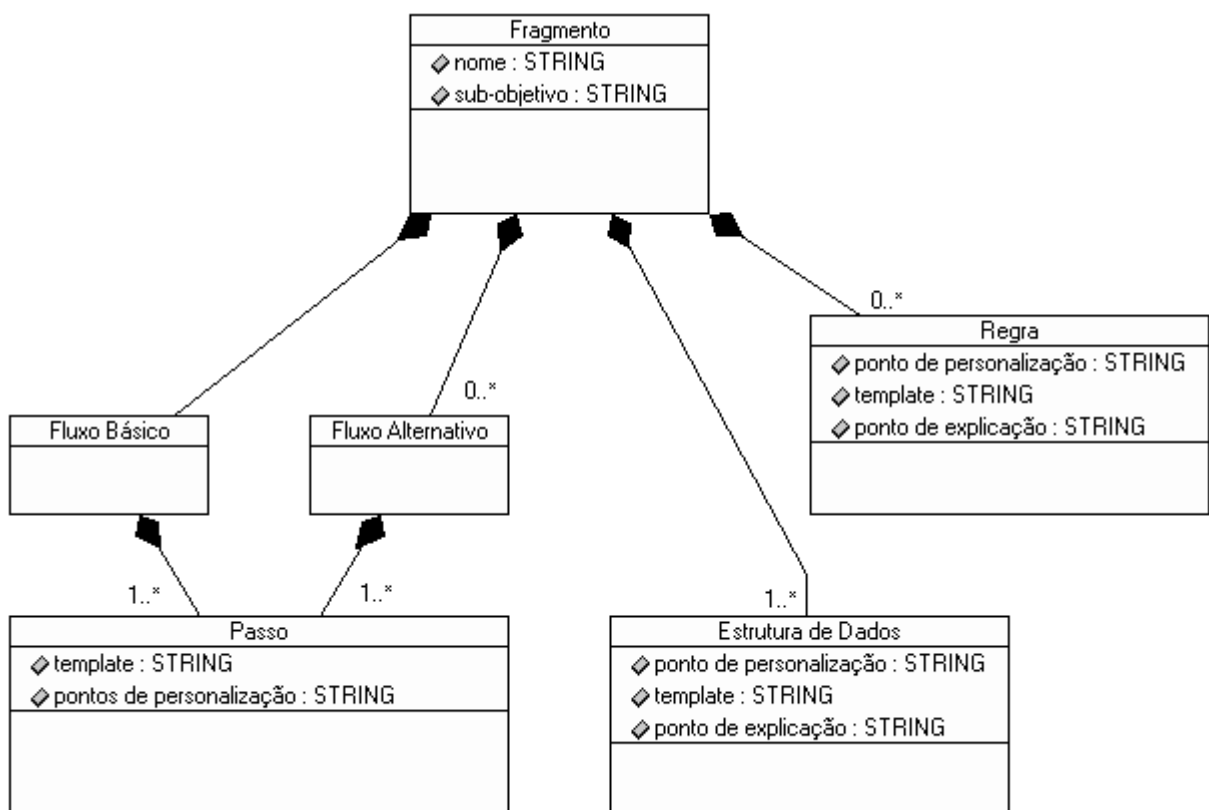


Figura 10. Modelo de explicação de um Fragmento e de suas partes

Fragmentos são recorrentemente encontrados durante a tarefa de descrição de requisitos de sistemas de informação e podem ser usados na maior parte das situações que se assemelhem em relação ao sub-objetivo específico. A recorrência de trechos de interações é comumente observada, como apresentado por Jacobson e Ng (2004). Assim, os Fragmentos são compostos por uma seqüência de tais interações recorrentes.

A escrita de um Fragmento é composta por três tipos de texto: *templates* (texto que não deve ser alterado), pontos de personalização e pontos de explicação. Os *templates* contêm o texto necessário para a descrição das interações requeridas para atingir um sub-objetivo, sendo escrito de acordo com as boas práticas definidas para passos de Casos de Uso. Os pontos de personalização, apresentados entre os sinais de ‘<’ e ‘>’, podem ocorrer no texto dos *templates* e devem ser substituídos por objetos do negócio ou por atores que interagem com o Caso de Uso em questão durante a personalização do Fragmento. A personalização do Fragmento pode requerer uma descrição das Estruturas de Dados e das Regras utilizadas no Caso de Uso. Por isso, os pontos de explicação são referências tanto às propriedades de cada objeto de negócio como à descrição das Regras de Negócio participantes do Caso de Uso.

Os Fragmentos não têm como finalidade restringir a solução de um modo extremo. Para que possam ser flexíveis e reutilizáveis, alguns passos dos Fragmentos podem ser utilizados opcionalmente. O responsável pela personalização do Caso de Uso deve decidir quando tais passos opcionais serão necessários. Essa possibilidade de opção permite a omissão de passos desnecessários, o que torna o uso dos Fragmentos menos rígido e, conseqüentemente, mais abrangente.

Os passos de Caso de Uso contidos nos Fragmentos apresentados neste trabalho foram escritos de acordo com a proposta do grupo CREWS, que pode ser encontrada integralmente no capítulo 2 desta dissertação e originalmente no trabalho apresentado por Cox e Phalp (2000). Como apresentado nesse capítulo, o grupo CREWS propôs dois conjuntos de orientações para a escrita de passos individuais: estilo e conteúdo.

As orientações de estilo provêm uma estrutura para que a escrita de um passo seja fácil de ser lida e compreendida. Essa orientação de estilo é aplicada diretamente no texto dos Fragmentos para indicar a ordem de interações entre os atores e o sistema nos fluxos básico e alternativos. As orientações de conteúdo provêm *templates* para descrever passos de forma

padronizada. Por exemplo, a orientação de conteúdo CG5 é usada para descrever um passo de Fragmento como este: “<ator> informa <parâmetros_para_seleção>”.

Além das orientações do grupo CREWS, as descrições dos Fragmentos também seguiram as orientações de qualidade de requisitos apresentadas por Sommerville e Sawyer (1997). Como exemplo dessas orientações, temos: definir *templates* para descrever requisitos e usar uma linguagem simples, concisa e fácil de ser lida.

O uso de Fragmentos para a escrita de Casos de Uso tem a finalidade de melhorar a qualidade da especificação de requisitos. A qualidade dos Fragmentos é decorrente da qualidade de cada passo do Fragmento e das boas práticas adotadas dos trabalhos anteriores. A documentação e o uso de Fragmentos são análogos ao caso de componentes de software: uma vez que cada parte seja construída com alta qualidade, o sistema como um todo também estará com alta qualidade. Pressman (2001) mostra que os componentes de software devem ser refinados e modificados, de forma a minimizar o retrabalho e melhorar a qualidade dos softwares posteriores.

3.4 Um Fragmento de Caso de Uso

Cada Fragmento de Caso de Uso possui seis seções: nome, sub-objetivo, a descrição da sua parte de Fluxo Básico, o conjunto dos Fluxos Alternativos relativos aos passos do Fluxo Básico, o detalhes das Estruturas de Dados e o detalhes das Regras. A Figura 11 apresenta o Fragmento “Obter Confirmação de Registro”, que terá cada parte explicada nesta seção. Apresentaremos apenas um dos Fragmentos, uma vez que todos possuem as mesmas características, diferenciando-se apenas no seu conteúdo.

Nome do Fragmento: Obter Confirmação de Registro
Sub-Objetivo: Obter Confirmação a respeito do Registro de um conjunto de informações no Sistema.
Fluxo Básico: 1. <ator> solicita o registro de <objeto>. 2. Sistema apresenta dados de <objeto>. 3. <ator> confirma registro de <objeto>. 4. Sistema apresenta os detalhes do recibo de registro.
Fluxos Alternativos: <i>a) registro cancelado.</i> No passo 3 do Fluxo Básico, o <ator> decide cancelar o registro. 1. Sistema apresenta a mensagem: “Registro cancelado”. 2. Fluxo de eventos retorna ao passo 1 do Fluxo Básico. <i>b) regra de registro violada.</i> No passo 4 do Fluxo Básico, o Sistema não permite o registro de <objeto> devido à violação da <regra_de_registro>. 1. Sistema apresenta mensagem de acordo com as regras que tenham sido violadas. 2. Fluxo de eventos retorna ao passo 1 do Fluxo Básico. <i>fluxos opcionais: b)</i>
Detalhes de Estruturas de Dados: a) Dados de <objeto> = <conjunto_de_propriedades> a ser registrado. b) Detalhes do recibo de registro = lista de <propriedades>.
Detalhe de Regras: a) <regra_de_registro> = definição de todas as regras que devem ser aplicadas para validar os parâmetros informados para o registro de um dado objeto. <i>detalhes opcionais: a)</i>

Figura 11. Fragmento “Obter Confirmação de Registro”

A primeira seção do Fragmento é seu nome. O nome deve ser um resumo da sua finalidade, uma vez que será útil para quem estiver analisando o Catálogo em busca de um Fragmento. O nome deve ser único e o mais curto possível.

A segunda seção do Fragmento é o sub-objetivo. Essa seção deve explicar a finalidade do uso de tal Fragmento, de forma que o responsável pela escrita do Caso de Uso possa comparar o seu sub-objetivo específico com o do Fragmento e decidir se utilizará ou não esse Fragmento no seu Caso de Uso. Nesse Fragmento apresentado, o seu sub-objetivo indica que

tal Fragmento deve ser utilizado quando o ator desejar obter uma confirmação após a solicitação de registro de um dado objeto.

A terceira seção contém a descrição de um trecho de Fluxo Básico. Esse trecho não é a descrição completa do Caso de Uso, mas sim a parte de contribuição do Fragmento. A descrição da parte do Fluxo Básico é composta por um conjunto de passos, em seqüenciamento lógico e numerado. Cada passo possui pontos de personalização, que devem ser substituídos por atores ou objetos, e *templates*, que são sugestões de textos formados a partir de boas práticas. No passo “1” do Fluxo Básico apresentado, o primeiro ponto de personalização deve ser substituído pelo ator que solicita o registro e o segundo deve ser substituído pelo objeto a ser registrado. Os demais passos devem ter seus pontos substituídos da mesma forma, formando uma seqüência de interações entre o ator e o sistema.

A quarta seção contém os Fluxos Alternativos. Estes são consequência direta da aplicação do Fluxo Básico sugerido pelo Fragmento e devem ser tratados nessa seção. Cada Fluxo Alternativo possui: uma referência numérica a um passo do Fluxo Básico, uma explicação da situação onde a execução do sistema segue para o Fluxo Alternativo e a seqüência alternativa de ações. Assim como no Fluxo Básico, os pontos de personalização devem ser substituídos pela nomenclatura coerente utilizada em um Caso de Uso. No Fragmento apresentado, há uma sugestão de opção quanto à aplicação do Fluxo Alternativo b). Como este se baseia na violação de uma regra para o registro do objeto, caso não haja tal regra, esse Fluxo Alternativo torna-se desnecessário.

A quinta seção contém os pontos de explicação a respeito das Estruturas de Dados dos objetos que são participantes dos passos do Caso de Uso. No Fragmento apresentado, é solicitada explicação a respeito dos dados do objeto que será registrado e os detalhes do recibo de registro. O ponto de explicação requer que sejam explicitados todos os detalhes e essa explicação deve conter referência a outros documentos externos caso existam, como

Dicionário de Dados. Os pontos de explicação são importantes, pois não basta escrever que um dado objeto será registrado, uma vez que parte ou a totalidade do mesmo podem ser efetivamente registradas. Os requisitos quanto às propriedades dos objetos devem ser explicitados, uma vez que somente o modelo dos objetos não é suficiente para dizer quais propriedades serão utilizadas em cada situação.

Finalmente, a sexta seção contém os pontos de explicação em relação às Regras referenciadas no Fragmento. Assim como nos detalhes de Estruturas de Dados, os pontos de explicação de Regras devem explicitar as restrições que ocorrem nos passos e nas condições para desvio de execução para os Fluxos Alternativos. Comumente, tais Regras são necessárias devido a alguma restrição do negócio que utilizará o software e, por isso, a explicação de Regras deve conter referências para o documento externo que contém as Regras de Negócio. No Fragmento apresentado, o detalhe de Regras solicita que sejam explicitadas as restrições de validação para quem um determinado objeto possa ser registrado. Essas restrições devem conter tanto as restrições de formato dos dados como referências às restrições de negócio.

A estrutura de todos os Fragmentos de Caso de Uso é a mesma, de forma que o Catálogo contenha todos os Fragmentos em um formato de apresentação padronizado. Assim, à medida que novos estudos forem conduzidos, será possível a expansão do Catálogo e a manutenção da sua padronização.

O Catálogo contendo todos os detalhes a respeito dos Fragmentos de Casos de Uso desenvolvidos até o momento encontra-se no Apêndice A. Os fragmentos, juntamente com seus sub-objetivos, são:

- Seleção Única de uma Lista: selecionar um elemento de um conjunto de elementos existentes;
- Seleção Múltipla de uma Lista: selecionar zero, um ou mais elementos de um conjunto de elementos existentes;

- Obter Detalhes de um Objeto: obter os detalhes de um objeto fornecendo seu identificador;
- Obter Confirmação de Registro: obter confirmação a respeito do registro de um conjunto de informações no sistema;
- Obter Confirmação de Envio de Arquivo: obter confirmação a respeito do envio de um arquivo ao sistema;
- Obter Confirmação de Exclusão de Objeto: obter confirmação a respeito da exclusão de um determinado objeto do sistema;
- Buscar Objetos: buscar por instâncias de um dado objeto de interesse

3.5 Como usar os Fragmentos de Casos de Uso

Para usar os Fragmentos de Casos de Uso, é necessário seguir uma seqüência de etapas. As entradas do macro-processo são: atores, objetivos, Casos de Uso identificados e definição dos termos do negócio. As saídas do macro-processo são os Casos de Uso com suas descrições completas. A Figura 12 apresenta o diagrama do macro-processo, contendo a ordem de execução e as principais etapas para usar os Fragmentos de Casos de Uso.

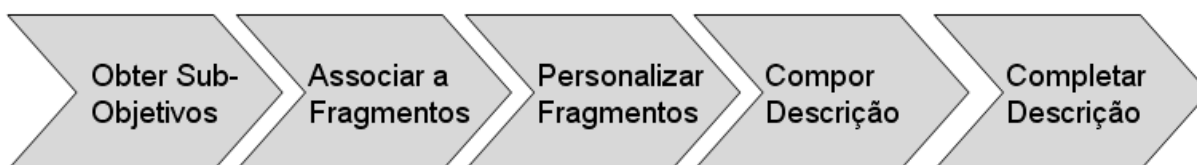


Figura 12. Macro-processo para o uso dos Fragmentos

Cada uma das etapas do macro-processo possui atividades próprias para contribuir de alguma forma com a produção do texto do Caso de Uso. As atividades de cada etapa são descritas nas próximas seções.

3.5.1 Obter Sub-Objetivos

- Entrada: objetivo principal do Caso de Uso
- Saída: lista de sub-objetivos ordenados

Os Fragmentos de Casos de Uso estão relacionados a sub-objetivos recorrentemente encontrados em requisitos de sistemas de informação. Por isso, o primeiro passo a ser seguido na construção de Casos de Uso com Fragmentos consiste na decomposição do objetivo do Caso de Uso em seus respectivos sub-objetivos, de forma que o objetivo principal seja atingido assim que os seus sub-objetivos também o forem. Para usar Fragmentos, não há restrição quanto à técnica utilizada para realizar tal decomposição. Yu et al (2008) apresentam uma dessas possíveis técnicas para modelagem de objetivos e sub-objetivos.

3.5.2 Associar a Fragmentos

- Entrada: lista de sub-objetivos ordenados e Catálogo de Fragmentos de Casos de Uso
- Saída: lista ordenada de pares [sub-objetivo, Fragmento]

De posse dos sub-objetivos, deve-se buscar o Fragmento do catálogo que seja mais adequado a cada sub-objetivo. Essa associação deve ser feita de acordo com a percepção e entendimento a respeito do sub-objetivo e dos Fragmentos, sendo necessário um estudo do Catálogo de Fragmentos para escolher o Fragmento que melhor se adéque à situação. O esquema da decomposição de um objetivo em sub-objetivos e da associação destes com Fragmentos é indicado na Figura 13.

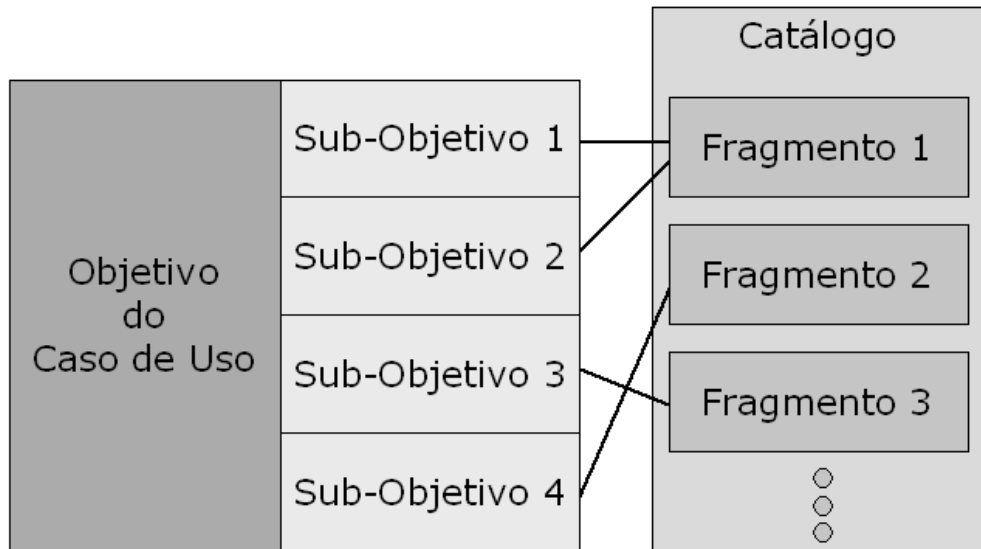


Figura 13. Associar cada sub-objetivo ao Fragmento mais adequado

Um Fragmento tem permissão para ser associado a um ou mais sub-objetivos. Porém, um sub-objetivo deve ser associado a no máximo um Fragmento. Essa atividade produz um conjunto de pares [sub-objetivo, Fragmento] que serão utilizados na etapa seguinte do processo.

3.5.3 Personalizar Fragmentos

- Entrada: lista ordenada de pares [sub-objetivo, Fragmento], modelo de objetos de negócio e atores
- Saída: Fragmentos personalizados (trechos)

Após a identificação dos pares [sub-objetivo, Fragmento], cada Fragmento deve ser personalizado através da substituição dos pontos de personalização por objetos do domínio do negócio e da expansão dos pontos de explicação. O resultado dessa tarefa são trechos contendo os passos e detalhes necessários para a composição de um Caso de Uso, como mostra a Figura 14

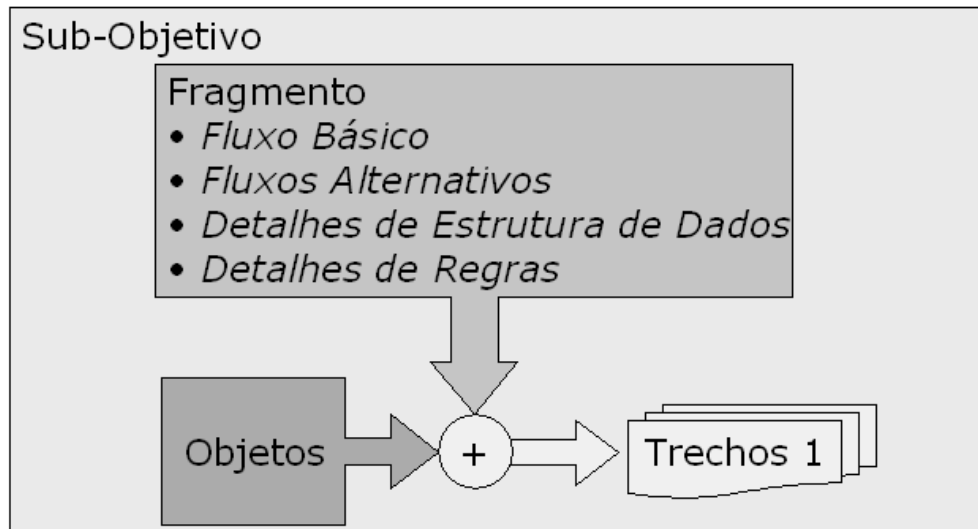


Figura 14. Personalizando os Fragmentos com objetos do domínio do negócio

Ao personalizar um Fragmento, dois detalhes importantes devem ser respeitados:

- o texto original (*template*) deve ser mantido ao personalizar o trecho de Fluxo Básico e os trechos de Fluxos Alternativos, sendo permitida apenas a inserção de objetos de negócio nos pontos de personalização;
- a personalização dos pontos de explicação requer a sua definição textual, já que não há *template* para esse tipo de descrição. O detalhamento da Estrutura de Dados e das Regras de Negócio depende, por sua vez, de um conhecimento prévio do modelo conceitual dos objetos do negócio e de suas interações.

Os detalhes de Estrutura de Dados devem conter todas as informações necessárias para que as propriedades de cada objeto sejam devidamente indicadas. Quando num passo de Caso de Uso é necessário apresentar os detalhes de um objeto, o detalhamento da Estrutura de Dados deve explicar quais são as propriedades que deverão ser apresentadas, visto que tais detalhes dependem de diversos fatores, tais como: o ator que estiver interagindo com o sistema e o momento da execução em que os detalhes serão apresentados. A explicação da Estrutura de Dados deve conter uma referência a um documento padrão de dados, como um Dicionário de Dados. O Dicionário de Dados contém o esquema e as informações necessárias para a apresentação dos objetos (SILBERSCHATZ *et al*, 2005). Os requisitos relacionados

diretamente ao domínio do sistema podem ser tratados de acordo com outras abordagens, como a apresentada por Moon *et al* (2005).

Os detalhes de Regras devem explicitar o comportamento do sistema em relação aos estímulos dos atores. Na explicação, deve haver uma referência ao repositório de Regras de Negócio, se houver, de forma que os Casos de Uso descrevam apenas o comportamento do software através das restrições das Regras. A definição das Regras de Negócio deve seguir alguma abordagem que diminua a sua ocorrência ao longo de documentos de requisitos. Um exemplo de abordagem para especificação de Regras de Negócio é o arcabouço SBRV, definido na especificação de OMG ⁵(2008).

3.5.4 Compor Descrição

- Entrada: Fragmentos personalizados (trechos) e lista ordenada de pares [sub-objetivo, Fragmento]
- Saída: descrição do Caso de Uso composta por trechos

Cada trecho contém informações de partes distintas de um Caso de Uso. Tais informações devem ser agrupadas de acordo com o seu tipo, com o objetivo de formar uma descrição completa. Somente os trechos por si só não são suficientes para uma especificação completa de requisitos, visto que os trechos atendem a apenas um sub-objetivo. Como deseja-se atingir o objetivo principal do Caso de Uso, os trechos devem ser agrupados na seqüência lógica necessária para tal e cada parte dos trechos deve ser agrupada de acordo com sua finalidade: trechos de Fluxo Básico devem ser agrupados em um único Fluxo Básico na ordem especificada na decomposição por sub-objetivos; os trechos de Fluxos Alternativos devem ser

⁵ Grupo de Gerenciamento de Objetos (Object Management Group, OMG). Ver: <http://www.omg.org/>.

agrupados em uma seção de Fluxos Alternativos; os detalhes de Estrutura de Dados e de Regras devem ser agrupados em suas seções respectivas.

A Figura 15 mostra a composição dos trechos em uma descrição completa de Caso de Uso. O símbolo de “+” representa o agrupamento de cada trecho de acordo com sua finalidade e não simplesmente um agrupamento em blocos. A Figura 16 representa uma descrição completa contendo os trechos agrupados de acordo com o tipo de informação. O Fluxo Básico é composto pelos trechos de Fluxo Básico, ocorrendo o mesmo com as demais partes do Caso de Uso.

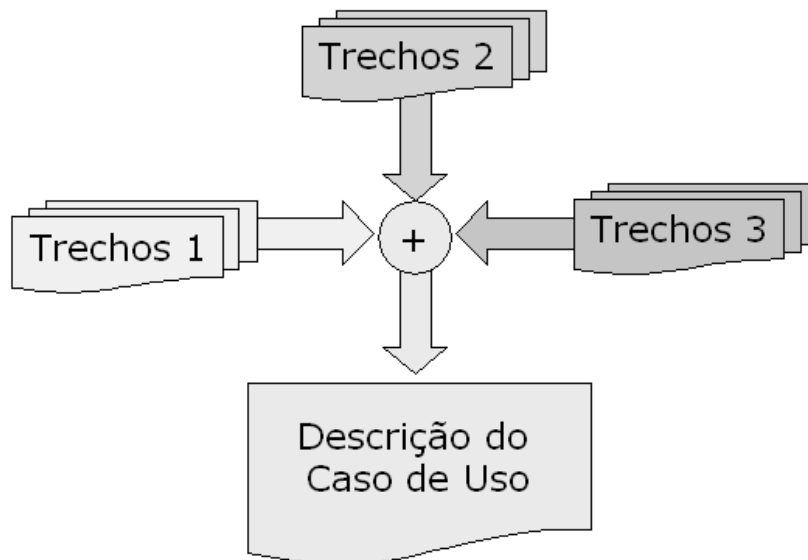


Figura 15. Compondo a descrição do Caso de Uso

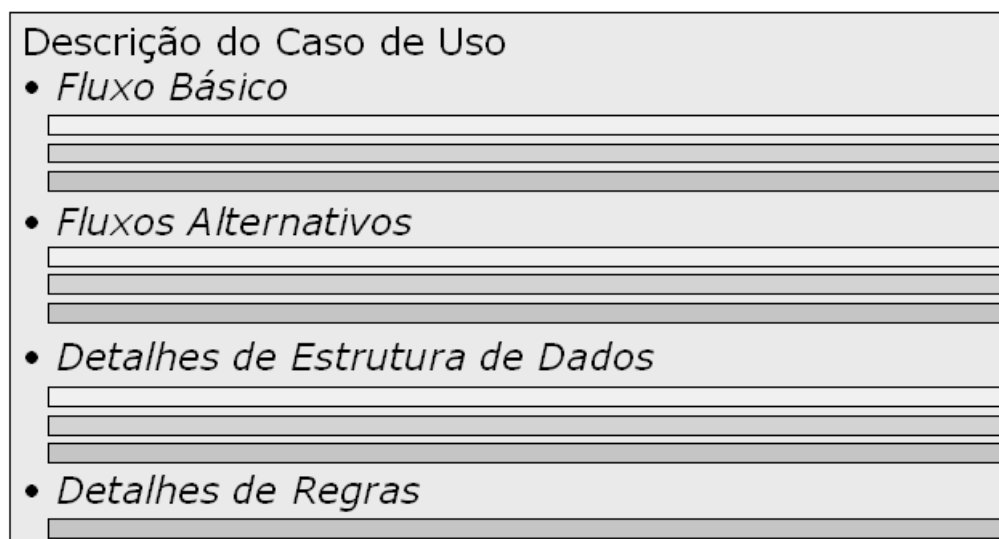


Figura 16. Visualização das partes da descrição do Caso de Uso, composta a partir dos trechos de Fluxos e detalhes de Estrutura de Dados e Regras

Após compor a descrição do Caso de Uso, é necessário ajustar ordenação dos passos, de forma que a referência dos fluxos alternativos esteja corretamente associada aos passos do fluxo básico correspondentes. No Fragmento, o *template* apresenta uma numeração que se inicia em “1”. Porém, ao compor a descrição, a numeração dos passos deve ser ajustada para se tornar incremental ao longo dos fluxos. Além da numeração dos passos, devem ser feitos ajustes nas numerações de referência dos Fluxos Alternativos. Como cada Fluxo Alternativo possui uma referência numérica para um passo, tal referência deve ser ajustada para que os Fluxos Alternativos sejam coerentes aos passos. A atividade de atualização numérica de passos pode ser facilitada pelo uso de ferramentas computacionais que forneçam referência cruzada, de forma que ao alterar um índice, sua referência seja atualizada sem a necessidade de busca manual.

3.5.5 Completar Descrição

- Entrada: descrição do Caso de Uso composta por trechos e objetivo principal do Caso de Uso
- Saída: descrição completa do Caso de Uso

A etapa de completar a descrição é necessária para que sejam inseridas as informações adicionais que não tenham sido contempladas na personalização dos Fragmentos. Como os Fragmentos são genéricos para uso em diversos tipos de sistemas e domínios genéricos, há situações específicas em que algumas adaptações e acréscimo de informação são necessários. Essa última etapa é responsável por garantir que haja coerência no texto final da descrição em relação ao objetivo principal do Caso de Uso.

3.6 Exemplo: Caso de Uso escrito com Fragmentos

Vamos exemplificar o uso de Fragmentos na especificação de um Caso de Uso do negócio de Aluguel de Carros. Esse exemplo encontra-se originalmente no trabalho de Dias *et al* (2008). Para mostrar o uso dos Fragmentos, o exemplo seguirá as etapas descritas na seção 3.5 desta dissertação.

O negócio de Aluguel de Carros necessita de um sistema de informações para suportar o processo de aluguel imediato, ou seja, na situação em que o cliente deseja alugar um carro sem ter feito uma reserva prévia. Tal necessidade foi estudada através da elicitação de requisitos, que teve como produto o modelo dos termos do negócio e o diagrama de Casos de Uso do sistema desejado. A Figura 17 apresenta os termos de negócio, seus relacionamentos e seus atributos. As associações entre termos, neste exemplo, utilizam apenas duas multiplicidades: muitos (símbolo “*”) e um e somente um (sem multiplicidade). A Figura 18 apresenta o diagrama de Casos de Uso, que contém o ator “Recepcionista” interagindo com o Caso de Uso “Registrar Aluguel Imediato”.

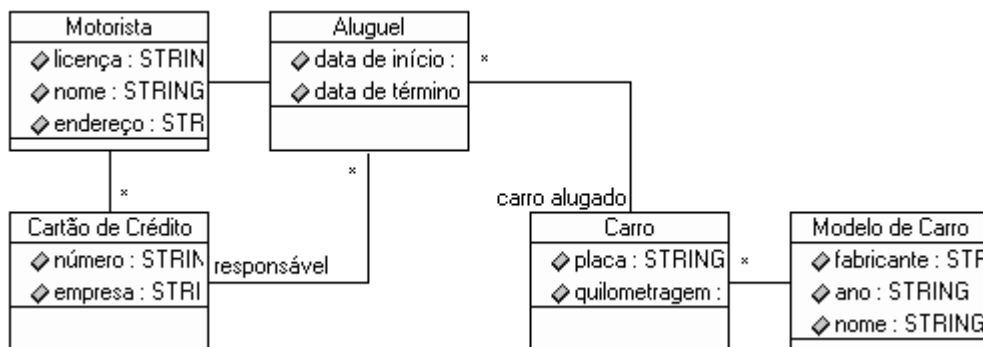


Figura 17. Diagrama de classes que representa o modelo conceitual dos termos de negócio



Figura 18. Diagrama de Casos de Uso do exemplo de Aluguel de Carros

Esse Caso de Uso possui um objetivo principal para o ator, que é “Registrar um aluguel imediato”. Essa é a necessidade do recepcionista ao utilizar o sistema quando desejar registrar o aluguel de um carro.

Sem utilizar alguma abordagem específica, o analista de requisitos deveria começar a descrever o Caso de Uso neste momento, após a identificação dos mesmos. Entretanto, como apontado na revisão de literatura, escrever Casos de Uso não é trivial de ser realizada por alunos e profissionais iniciantes. O analista inexperiente pode ter o conhecido das partes de um Caso de Uso, mas não possui a habilidade necessária para elaborar os passos detalhados de interações sem partir de um texto inicial. Nesta seção, estamos interessados em exemplificar o uso dos Fragmentos como uma forma de auxiliar o analista inexperiente a elaborar, integral ou parcialmente, o texto de seu Caso de Uso.

3.6.1 Obter Sub-Objetivos

O objetivo principal do Caso de Uso pode ser decomposto em sub-objetivos. A decomposição é o primeiro passo para o entendimento a respeito dos conjuntos de passos que serão necessários para descrever a interação entre o recepcionista e o sistema. Nesta etapa, os sub-objetivos necessários para registrar um aluguel são:

1. Obter detalhes do motorista: identificar o motorista interessado;
2. Selecionar um modelo de carro: escolher qual modelo de carro, dentre os disponíveis, será registrado para entrega ao motorista;
3. Selecionar um cartão de crédito: a empresa realiza alugueis apenas através da apresentação de um cartão de crédito, que deve ser selecionado dentre os cartões de crédito disponíveis no cadastro do motorista;
4. Obter confirmação de registro do aluguel: o aluguel deve ser registrado e o recepcionista deve receber uma confirmação de que tal operação foi realizada, para poder liberar o carro para locação.

O detalhamento em sub-objetivos é importante para determinar os conjuntos mais relevantes de passos do Caso de Uso. Sem essa etapa, o analista de requisitos deveria começar a escrever o Caso de Uso sem seguir uma forma estruturada e lógica, dependendo apenas de sua capacidade, o que somente é possível após experiências com a técnica. A lista, ordenada, de sub-objetivos é a saída desta etapa. A ordem é importante para que os passos sigam uma seqüência lógica de forma a atingir o objetivo do Caso de Uso.

3.6.2 Associar a Fragmentos

Os sub-objetivos obtidos da etapa anterior devem ser associados a Fragmentos do Catálogo. Esta etapa de associação é importante para que sejam identificados os Fragmentos mais adequados à especificação em particular. Tal etapa é subjetiva e a correta escolha dos Fragmentos depende de um prévio entendimento do sub-objetivo e dos Fragmentos disponíveis no Catálogo. A associação dos sub-objetivos a Fragmentos feita neste exemplo é apresentada no Quadro 1. A lista de pares ordenados de sub-objetivos e Fragmentos é a saída desta etapa.

Quadro 1. Associação entre sub-objetivos do Caso de Uso e Fragmentos do Catálogo

Sub-objetivo	Fragmento selecionado
1. Obter detalhes do motorista	Obter os Detalhes de um Objeto
2. Selecionar um modelo de carro	Seleção Única de uma Lista
3. Selecionar um cartão de crédito	Seleção Única de uma Lista
4. Obter confirmação de registro do aluguel	Obter Confirmação de Registro

3.6.3 Personalizar Fragmentos

Cada Fragmento selecionado na etapa anterior deve ser personalizado nesta etapa. A personalização consiste na substituição dos pontos de personalização por termos do negócio. Além disso, as descrições das estruturas de dados e das regras devem ser devidamente explicadas para que as informações utilizadas nos passos sejam detalhadas.

Para explicitar a personalização, utilizaremos o Fragmento “Obter Confirmação de Registro”, apresentado na Figura 11. Ao personalizar esse Fragmento, os pontos <ator> e <objeto> são substituídos, respectivamente, pelo ator participante do Caso de Uso e pelos objetos que fazem parte da interação em cada passo dos fluxos básico e alternativos. Posteriormente, os detalhes de estrutura de dados são detalhados para indicar quais propriedades de cada termo devem ser utilizados.

A substituição dos pontos de personalização e o detalhamento das estruturas de dados devem considerar o sub-objetivo “Obter Confirmação de Registro do Aluguel”. A personalização do Fragmento deve produzir um trecho do Caso de Uso que possibilite ao ator atingir ao sub-objetivo desejado. No exemplo, o fluxo alternativo opcional b) e o detalhe de regra a) serão desconsiderados, pois são opcionais e no exemplo não consideraremos como relevantes. A Figura 19 apresenta o Fragmento personalizado para atender ao sub-objetivo em questão, onde as substituições são indicadas em negrito e as partes opcionais descartadas são indicadas com um risco.

<p>Fluxo Básico:</p> <ol style="list-style-type: none"> 1. Recepcionista solicita o registro do aluguel. 2. Sistema apresenta dados do aluguel. 3. Recepcionista confirma registro do aluguel. 4. Sistema apresenta os detalhes do recibo do registro.
<p>Fluxos Alternativos:</p> <p><i>a) registro cancelado.</i> No passo 3 do Fluxo Básico, o Recepcionista decide cancelar o registro.</p> <ol style="list-style-type: none"> 1. Sistema apresenta a mensagem: “Registro cancelado”. 2. Fluxo de eventos retorna ao passo 1 do Fluxo Básico. <p><i>b) regra de registro violada.</i> No passo 4 do Fluxo Básico, o Sistema não permite o registro de <objeto> devido à violação da <regra_de_registro>.</p> <ol style="list-style-type: none"> 1. Sistema apresenta mensagem de acordo com as regras que tenham sido violadas. 2. Fluxo de eventos retorna ao passo 1 do Fluxo Básico. <p><i>fluxos opcionais: b)</i></p>
<p>Detalhes de Estruturas de Dados:</p> <p>a) Dados de aluguel = motorista.nome, responsável.número, data de início, data de término, “carro alugado”.placa.</p> <p>b) Detalhes do recibo de registro = motorista.nome, responsável.número, data de início, data de término, “carro alugado”.placa, aluguel.data de início, aluguel.data de término.</p>
<p>Detalhe de Regras:</p> <p>a) <regra_de_registro> = definição de todas as regras que devem ser aplicadas para validar os parâmetros informados para o registro de um dado objeto.</p> <p><i>detalhes opcionais: a)</i></p>

Figura 19. Personalização do Fragmento “Obter Confirmação de Registro” para o sub-objetivo “Obter Confirmação de Registro do Aluguel”. As áreas riscadas correspondem ao que não será utilizado

A personalização desse Fragmento de Caso de Uso para o sub-objetivo 4, originou o Trecho 4. Cada personalização de Fragmento para cada sub-objetivo deve ser feita de forma análoga, produzindo um conjunto de Trechos. Os Trechos gerados nesta etapa serão utilizados para a composição da descrição do Caso de Uso na etapa seguinte.

3.6.4 Compor Descrição

Os Fragmentos personalizados devem ter suas partes comuns agrupadas, de forma a comporem a descrição completa do Caso de Uso. Para compor a descrição, são necessários os trechos de Fragmentos, personalizados na etapa anterior, e a lista ordenada de pares [sub-objetivo, Fragmento].

A partir da lista ordenada de pares, os trechos devem ter suas partes agrupadas de acordo com o tipo de informação. O fluxo básico do Caso de Uso deve ser composto pelos fluxos básicos dos trechos na ordem definida na lista. O mesmo deve ocorrer com os fluxos alternativos e com os detalhes de estruturas de dados e de regras. Respeitar a ordem definida pela lista é importante para que a seqüência de passos seja coerente para atingir ao objetivo principal do Caso de Uso.

O fluxo básico do Caso de Uso resultante da junção dos trechos possui sua numeração de acordo com cada Fragmento. Portanto, a numeração dos passos deve ser reconstruída para que haja consistência da seqüência de interações e dos fluxos alternativos em relação aos passos do fluxo básico. Na Figura 20 é mostrada a composição dos fluxos básicos num único fluxo básico. O mesmo deve ocorrer para as demais partes do Caso de Uso.

3.6.5 Completar Descrição

No exemplo mostrado, todos os sub-objetivos foram associados a Fragmentos do Catálogo, o que facilita a composição da especificação do Caso de Uso. Entretanto, isso não significa que o Caso de Uso esteja completo; podem ser necessárias adaptações ou acréscimo de informações à descrição para que o objetivo seja atingido corretamente. Nesta etapa, é verificado se o Caso de Uso satisfaz às necessidades dos atores envolvidos, ou seja, é verificado se através da descrição, o atendente está possibilitado a registrar um aluguel imediato de um carro. No exemplo mostrado, houve a necessidade apenas de inserir o primeiro e o último passos, para explicitar o início e o fim do Caso de Uso.

A descrição completa do Caso de Uso “Registrar Aluguel Imediato” é mostrada a seguir: Figura 20, Figura 21, Figura 22 e Figura 23, contendo o detalhamento dos passos dos fluxos básico e alternativos como também os detalhes de dados e de regras de negócio. Cada parte do Caso de Uso segue a ordem da lista de objetivos e seu conteúdo de acordo com o texto dos Fragmentos personalizados.

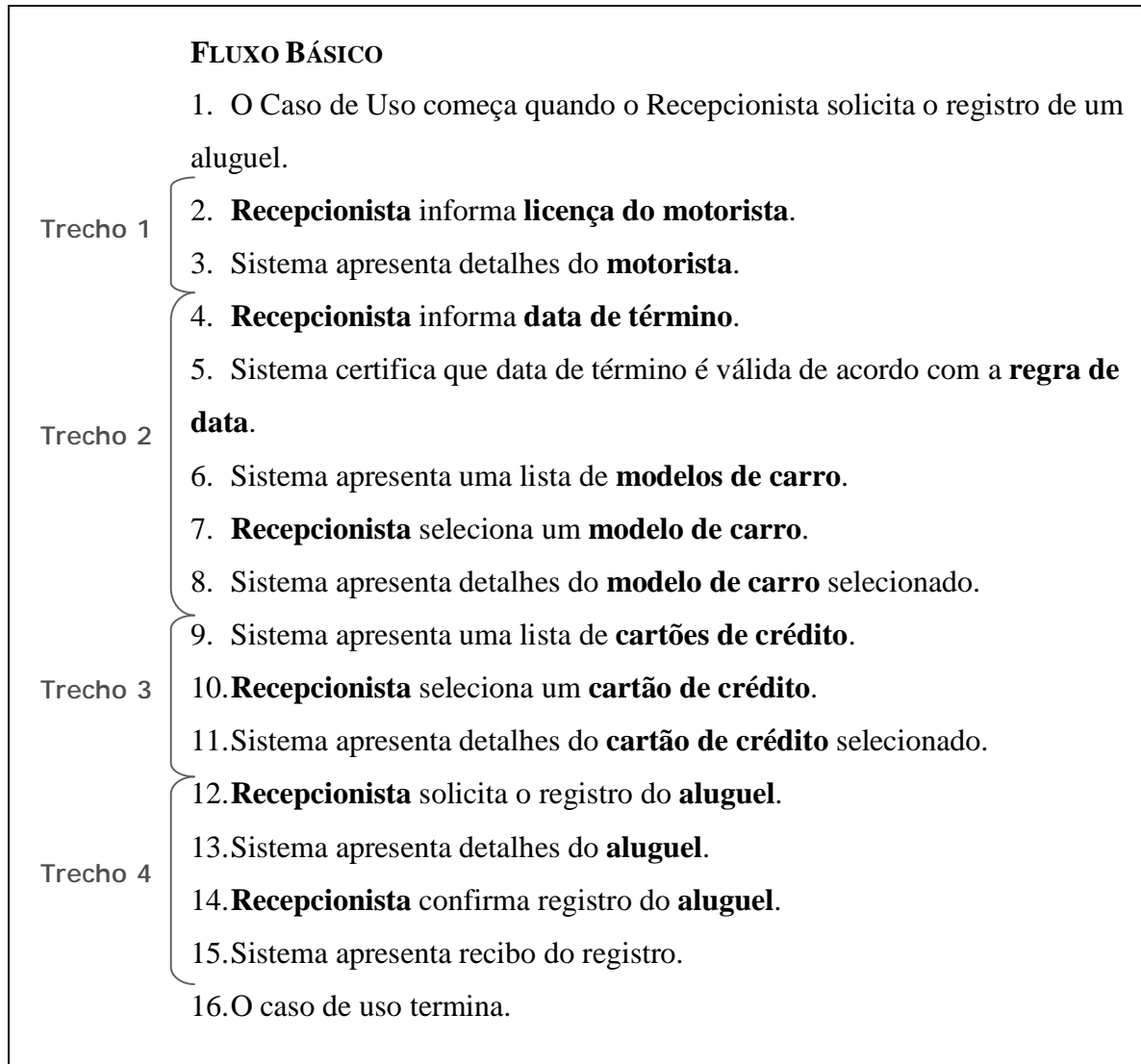


Figura 20. Fluxo Básico resultante da composição dos trechos de Fluxos Básicos

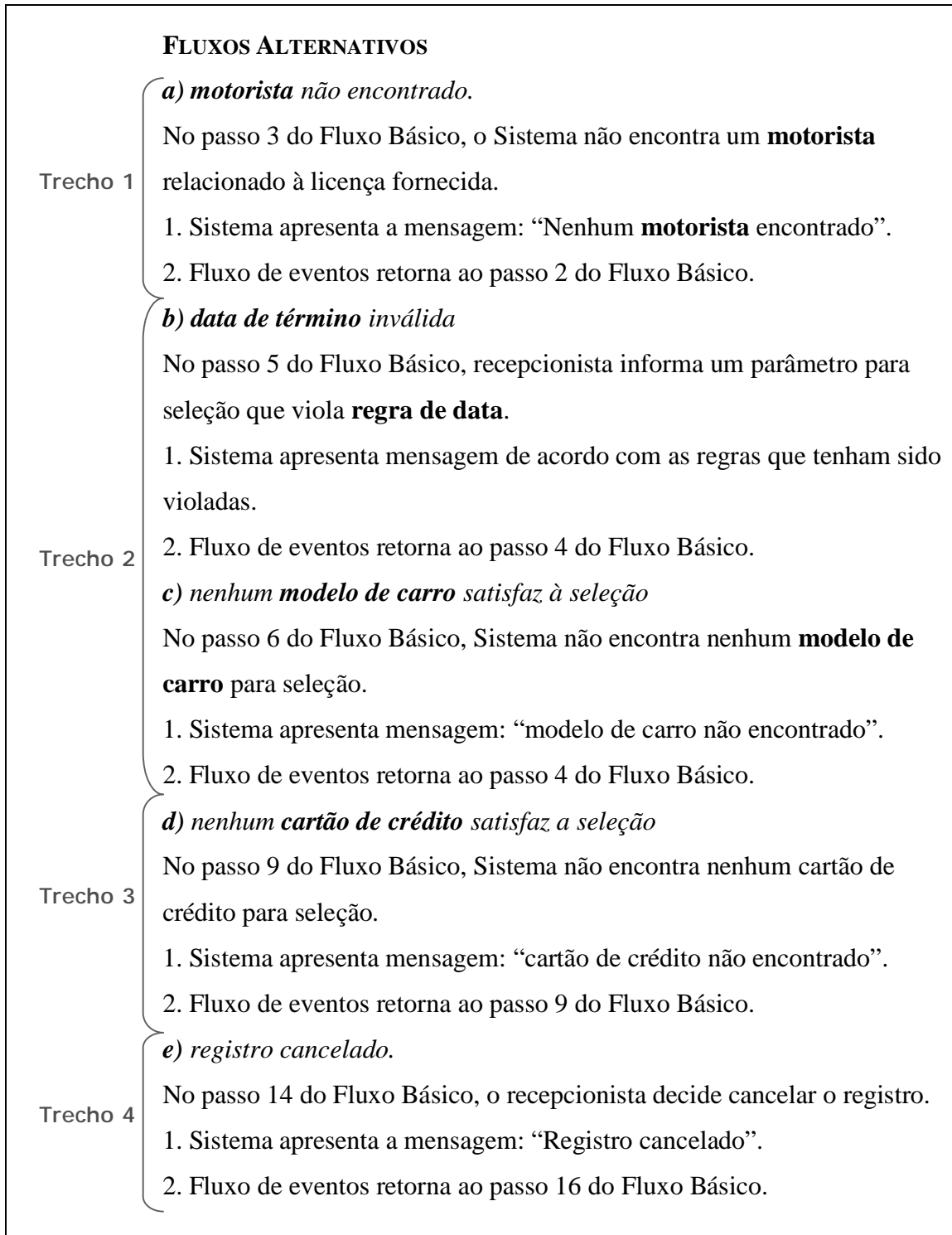


Figura 21. Fluxos Alternativos associados aos passos do Fluxo Básico

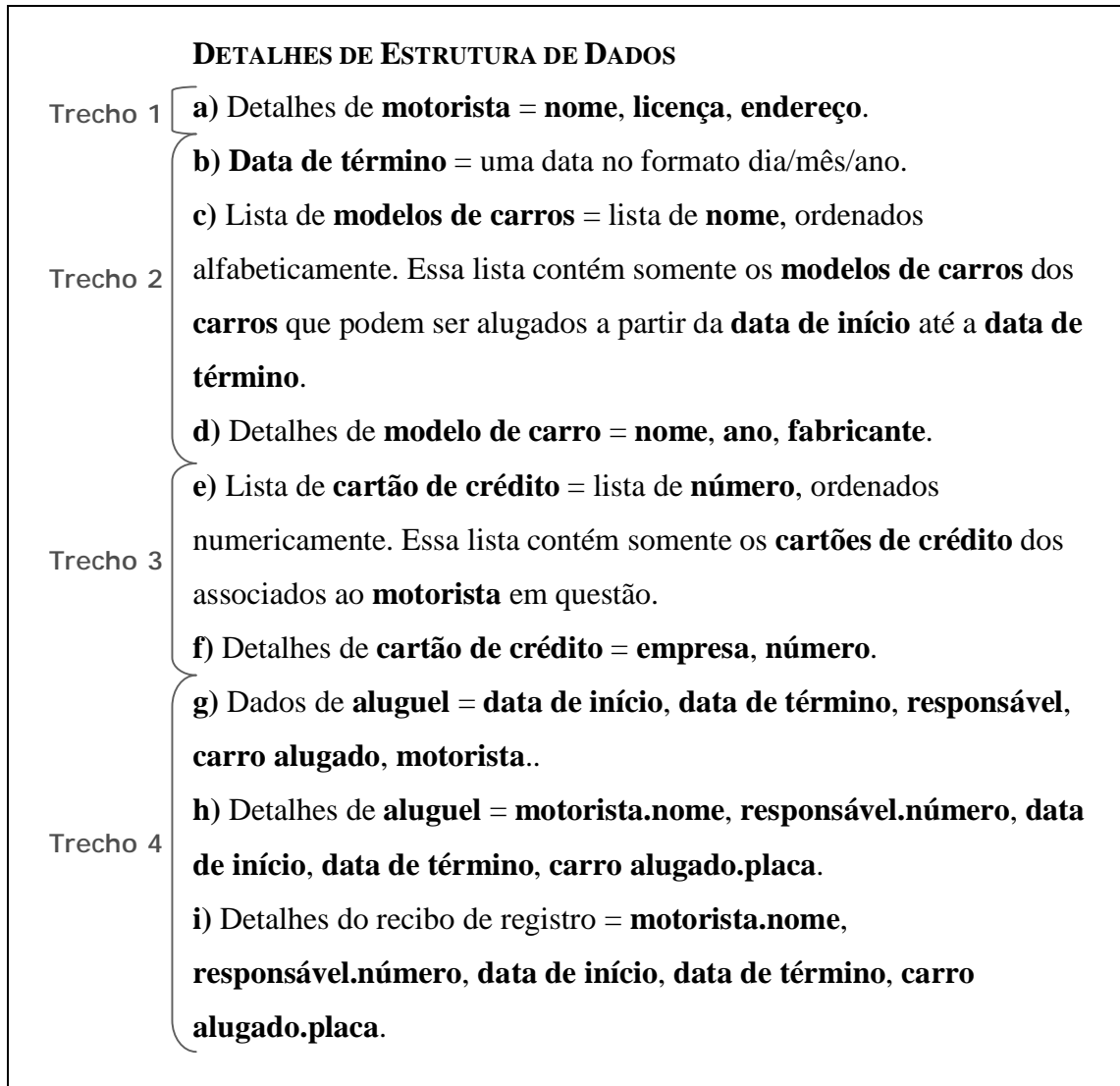


Figura 22. Detalhes das Estruturas de Dados de todos os termos de negócio participantes do Caso de Uso

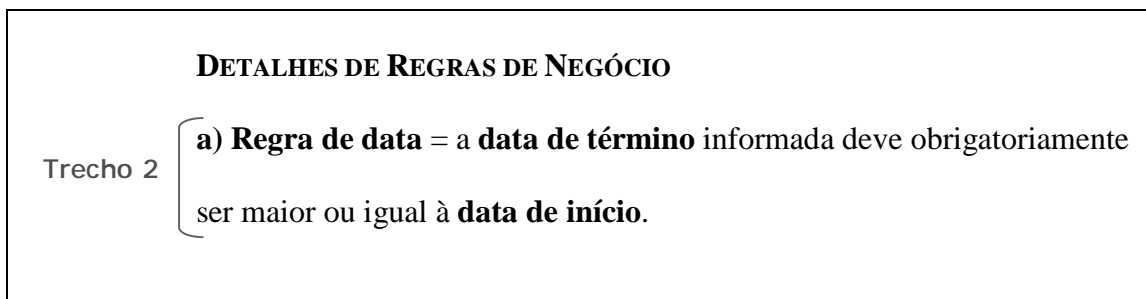


Figura 23. Detalhes das Regras de Negócio que restringem os termos de negócio

4 AVALIAÇÃO

4.1 Materiais e Métodos

4.1.1 Descrições Gerais

Uma avaliação da viabilidade da solução proposta neste trabalho foi realizada através de dois experimentos com alunos, em nível de graduação, do curso de Ciência da Computação da Universidade Federal do Rio de Janeiro.

O projeto de um experimento, de acordo com Wohlin *et al* (2000), deve conter: participantes, contexto, instrumentos, planejamento, hipóteses, variáveis e validação. Um único projeto de experimento foi elaborado para ser aplicado em dois contextos distintos.

4.1.2 Participantes

Os participantes, como já mencionado, foram unicamente alunos de graduação. Todos os alunos foram informados a respeito do experimento e aceitaram fazer parte do trabalho.

4.1.3 Contexto

O contexto continha uma situação fictícia de especificação de requisitos de software. Esse contexto era conhecido pelos estudantes: inscrição em disciplinas. Todos os alunos são da mesma universidade e do mesmo departamento, o que auxilia o conhecimento de tal contexto devido à necessidade de inscrição em disciplina todos os semestres. Os alunos tiveram conhecimento de que teriam que escrever um único Caso de Uso para o sistema proposto.

4.1.4 Instrumentos

O enunciado do problema consistia de uma breve descrição, em linguagem natural, usando a linguagem do cliente, dos requisitos de um software para aluguel (imediate) de carros. Além disso, foram fornecidos um diagrama de Casos de Uso e um modelo com as classes do domínio da aplicação. O enunciado completo encontra-se no Apêndice B desta dissertação. Além do enunciado do problema, foram entregues um Catálogo de Fragmentos de Caso de Uso, os documentos de resposta e um questionário, que pode ser encontrado no Apêndice E. O Catálogo de Fragmentos de Casos de Uso encontra-se no Apêndice A deste trabalho. Todos os alunos tiveram acesso a um computador com editor de texto para uso exclusivo.

4.1.5 Planejamento

Os experimentos foram conduzidos da seguinte forma:

1. o plano do experimento foi apresentado aos alunos utilizando um projetor;
2. os alunos foram separados aleatoriamente em dois grupos distintos. Esta separação foi feita utilizando-se um programa desenvolvido pelo autor desta dissertação com a finalidade gerar dois grupos balanceados aleatoriamente a partir de uma lista de alunos. A diferença de pessoas entre os grupos não foi maior que um aluno e a escolha dos mesmos foi aleatória;
3. um grupo recebeu o rótulo de Controle e o outro de Tratamento;
4. o grupo Controle se retirou para uma outra sala, onde seus componentes iniciaram a tarefa de ler o enunciado do problema e escrever o Caso de Uso para o sistema proposto;

5. o grupo Tratamento recebeu orientações breves sobre como utilizar o Catálogo de Fragmentos de Casos de Uso. Esse grupo recebeu o mesmo material do grupo controle, acrescido do próprio Catálogo de Fragmentos, e tiveram a mesma tarefa de escrever o Caso de Uso. Não foram impostas condições para a escrita do Caso de Uso, sendo solicitado apenas que esses alunos tentassem reutilizar os Fragmentos ao escreverem seus textos;
6. todos os alunos realizaram a tarefa de escrita do Caso de Uso de modo individual e sem consulta a qualquer outro material adicional;
7. ao terminar de escrever o Caso de Uso, preencheram o questionário e foram liberados da atividade.

A distribuição do material aos participantes utilizou um repositório de acesso comum, disponível na rede local. Todo o material entregue e recebido durante o experimento foi elaborado em meio eletrônico. O recolhimento do material elaborado pelos participantes utilizou o mesmo repositório, onde foram adicionados repositórios individuais com o nome de cada participante. O objetivo do questionário foi coletar informações subjetivas a respeito do participante e do andamento do experimento sendo que seu preenchimento foi feito somente após o término do experimento.

Ao longo do experimento nenhum outro material foi acessado pelos alunos. Os participantes puderam esclarecer dúvidas e o fizeram principalmente no início, quanto à condução do experimento. O grupo Tratamento teve uma breve explicação sobre como utilizar o Catálogo, pois não houve como fazê-lo num momento anterior, uma vez que os alunos foram divididos em grupos no momento de ocorrer o experimento.

A divisão em grupos foi feita de forma aleatória. Os nomes dos participantes foram utilizados como entrada de um software, que aleatoriamente dividiu os alunos em dois grupos com quantidade equivalente de alunos.

Para estimular a presença dos alunos ao experimento, foi oferecida uma pontuação extra na matéria. Portanto, os alunos não foram obrigados a participar, porém aqueles que o fizeram receberam uma gratificação de pontuação na disciplina que estavam cursando. A pontuação extra foi concedida igualmente a todos os que participaram, ou seja, não foi proporcional a nenhum outro critério, mas apenas por ter participado.

4.1.6 Hipóteses

A avaliação de um experimento requer a elaboração de hipóteses nulas, sendo desejável que os resultados dos experimentos venham a rejeitá-las. Hipóteses nulas definem se as diferenças entre os grupos do experimento são devidas ao acaso, ou seja, são aquelas que o interessado pelo experimento deseja rejeitar com a maior significância possível. As hipóteses nulas elaboradas para os experimentos são:

- H1₀: O uso de Fragmentos requererá uma quantidade de **Tempo** semelhante que uma abordagem sem restrições.
- H2₀: O uso de Fragmentos encontrará um número de **Defeitos Totais** semelhante ao de uma abordagem sem restrições.
- H3₀: O uso de Fragmentos encontrará um número de **Defeitos de Casos de Uso** semelhante ao de uma abordagem sem restrições.
- H4₀: O uso de Fragmentos encontrará um número de **Defeitos de Especificação** semelhante ao de uma abordagem sem restrições.
- H5₀: O uso de Fragmentos encontrará um número de **Defeitos de Requisitos** semelhante ao de uma abordagem sem restrições.
- H6₀: O uso de Fragmentos requererá um número de **Palavras**, para a escrita do Caso de Uso, semelhante ao de uma abordagem sem restrições.

O número de **Defeitos Totais** é calculado como o somatório dos três tipos de defeito: de **Casos de Uso**, de **Especificação** e de **Requisitos**. Os **Defeitos de Casos de Uso** estão relacionados à técnica de Casos de Uso, como por exemplo: erros em numeração de passos e de referência para fluxos alternativos. Os **Defeitos de Especificação** estão relacionados à forma de especificar os requisitos, como por exemplo: uso de terminologia inadequada, uso de sinônimos e depois problemas ao descrever requisitos. Os **Defeitos de Requisitos** são os mais graves, onde há dificuldade de entendimento do problema e da solução. Esse tipo de defeito pode ser encontrado principalmente em seqüências de passos e referências à fluxos alternativos incoerentes com o problema ou a solução (COX *et al*, 2004). A separação dos defeitos possibilita o estudo sobre os pontos fortes e fracos da aplicação dos Fragmentos.

A busca pelos erros foi feita pessoalmente pelo autor desta dissertação. Todas as respostas dos alunos foram avaliadas num único dia, pela mesma pessoa e utilizando o *checklist* apresentado por Cox *et al* (2004). Apesar de ser uma avaliação subjetiva, os critérios são definidos e categorizados, como mostra o Anexo A deste trabalho.

Um exemplo de erro é encontrado na frase: “*Sistema armazena as disciplinas inscritas pelo aluno na base de dados*”. O uso da expressão “base de dados” implica no erro de número três do *checklist*, pois se referencia a um detalhe interno do sistema. Outro exemplo: “*O sistema mostra uma lista de disciplinas disponíveis para a inscrição em uma tela pop-up*”. Nesse caso, ocorre o erro de número 3, pois faz referência a uma interface visual do sistema. Ambos os erros mostrados contabilizam como **Defeito de Especificação**, como categorizado no trabalho de Cox *et al* (2004).

4.1.7 Variáveis

As variáveis medidas foram: tempo gasto na escrita do Caso de Uso, quantidade total de defeitos encontrados, quantidade de defeitos de Caso de Uso, quantidade de defeitos de especificação, quantidade de defeitos de requisitos e quantidade de palavras. A partir dos

valores obtidos, foi possível aplicar o Teste U de Mann-Whitney (BUSSAB, MORETTIN, 2002) para diferença de grupos, com o nível de significância (*p-level*) igual a 0,05.

As quantidades de defeitos foram obtidas através do *checklist* que se encontra no Anexo A deste trabalho. A quantidade de tempo foi obtida através dos horários de início e fim fornecidos por cada aluno no questionário.

O Teste U é um teste não-paramétrico equivalente ao Teste-T de Student. Entretanto, Teste U não define restrições a respeito da normalidade dos dados como também é aplicável na avaliação de variável de medidas usando uma escala ordinal, como explicado por Juristo e Moreno (2001). O cálculo do Teste U foi feito pela ferramenta STATISTICA'99⁶, que possui diversas funcionalidades para análise estatística e geração de resultados gráficos (STATSOFT, 1999).

A escolha do nível de significância (*p-level*) de 0,05 é devida ao fato desse valor ocorrer recorrentemente em experimentos de Engenharia de Software, sendo sugerido por Wohlin *et al* (2000) e encontrado em trabalhos similares tais como os de Cox e Phalp (2000), de Correa (2006) e de Fogelström e Gorschek (2007).

4.1.8 Validação

Na etapa de validação, os resultados obtidos do experimento são analisados para verificar se tais resultados estão consistentes em relação ao esperado. Ou seja, os casos de uso gerados e os questionários foram analisados para verificar se tudo as informações solicitadas foram informadas. Não houve ocorrência de ausência de preenchimento ou informações indevidas. Isso se deve ao fato de durante o experimento os alunos terem sido alertados ao terminar as tarefas solicitadas, o que evitou o esquecimento de alguma informação.

⁶ <http://www.statsoft.com>

4.2 Os Experimentos

Em ambos experimentos os alunos estavam cientes que participariam de um experimento. Porém, não tiveram conhecimento dos critérios de avaliação, de forma que estiveram livres para escrever o Caso de Uso de acordo com a forma que lhes fora solicitada, sem pensarem nos defeitos que poderiam cometer. Os alunos do Experimento 1 tinham bom entendimento sobre Casos de Uso e já haviam desenvolvido outros trabalhos de especificação de requisitos ao longo da disciplina. Os alunos do Experimento 2 não tinham bons conhecimentos sobre Casos de Uso.

4.2.1 Experimento 1

O Experimento 1 ocorreu no dia 21 de novembro de 2007, às 10h da manhã, e foi realizado com alunos da disciplina eletiva de Modelagem de Sistemas de Informação 2, cuja ementa abordava Engenharia de Requisitos de Software como assunto principal. Os alunos, no momento da realização do experimento, tinham passado por um treinamento específico sobre especificação de requisitos e, portanto, possuíam conhecimento das boas práticas sobre elaboração de requisitos de software com Casos de Uso. Os grupos Controle e Tratamento continham ambos oito participantes.

4.2.2 Experimento 2

O Experimento 2 ocorreu no dia 28 de novembro de 2007, às 13h da tarde, e foi realizado com alunos da disciplina obrigatória de Fundamentos da Engenharia de Software, cuja ementa abordava requisitos de software como um dentre diversos outros assuntos. Os alunos não tinham experiência na especificação de requisitos e não tampouco estavam habituados às boas práticas sobre a elaboração de requisitos de software. Os grupos Controle e Tratamento continham dezoito e dezessete participantes respectivamente.

4.3 Resultados

Esta seção apresenta os dados referentes às variáveis observadas, que foram coletados após a execução dos experimentos. A variável **Defeitos Totais** foi calculada como o somatório dos demais defeitos. As variáveis **Conhecimento** e **CRA** (Coeficiente de Rendimento Acumulado) foram informadas pelos próprios alunos através do Questionário e foram utilizadas no experimento apenas para averiguar se havia diferença significativa entre os grupos Controle e Tratamento.

4.3.1 Experimento 1

Os dados coletados a partir dos Casos de Uso entregues e dos questionários referentes ao Experimento 1 encontram-se detalhados no Apêndice D, separados por variável e por aluno. A Tabela 1 apresenta as medianas dos valores obtidos para cada variável observada e a probabilidade de os valores terem sido ao acaso.

Os valores obtidos mostram que não há diferença significativa entre os grupos (*p-level* > 0,05) em relação às variáveis: CRA, CONHECIMENTO, TEMPO e DEFEITOS DE CASO DE USO. As demais variáveis apresentaram uma baixa probabilidade (*p-level* ≤ 0,05) de os valores terem sido ao acaso.

Tabela 1. Resultados do Experimento 1 obtidos a partir da aplicação do teste U de Mann-Whitney.

	Mediana	Mediana	
	CONTROLE	TRATAMENTO	p-level
CRA	6,0	6,0	0,752
CONHECIMENTO	6,0	6,5	0,916
PALAVRAS	297,5	390,0	0,016
TEMPO	45,0	57,5	0,083
DEFEITO – TOTAL	8,0	2,0	0,002
DEFEITO - CASO DE USO	0,0	1,0	0,600
DEFEITO – ESPECIFICAÇÃO	2,0	0,0	0,031
DEFEITO – REQUISITO	5,0	0,5	0,007

4.3.2 Experimento 2

Os dados coletados a partir dos Casos de Uso entregues e dos questionários referentes ao Experimento 2 encontram-se detalhados no Apêndice E, separados por variável e por aluno. A Tabela 2 apresenta as medianas dos valores obtidos para cada variável observada e a probabilidade de os valores terem sido ao acaso.

Os valores obtidos mostram que não há diferença significativa entre os grupos (*p-level* > 0,05) em relação às variáveis: CRA, CONHECIMENTO, DEFEITOS DE CASO DE USO, DEFEITOS TOTAL e DEFEITOS DE REQUISITO. As demais variáveis apresentaram uma baixa probabilidade (*p-level* ≤ 0,05) de os valores terem sido ao acaso.

Tabela 2. Resultados do Experimento 2 obtidos a partir da aplicação do teste U de Mann-Whitney

	Mediana	Mediana	
	CONTROLE	TRATAMENTO	p-level
CRA	6,0	5,0	0,109
CONHECIMENTO	6,0	6,0	0,632
PALAVRAS	283,0	451,0	0,00008
TEMPO	41,5	60,0	0,00003
DEFEITO – TOTAL	4,5	3,0	0,138
DEFEITO – CASO DE USO	1,0	1,0	0,792
DEFEITO – ESPECIFICAÇÃO	1,0	0,0	0,010
DEFEITO – REQUISITO	2,5	2,0	0,283

5 DISCUSSÃO

5.1 Análise Estatística

5.1.1 Critério para rejeição de hipóteses

Ambos os experimentos permitiram, com base na probabilidade de os resultados não terem sido ao acaso, rejeitar algumas das hipóteses nulas. Os resultados foram semelhantes, porém não iguais. As seções seguintes explicam algumas particularidades dos resultados de cada experimento em separado, visto que cada um foi realizado em situações e com participantes distintos.

Uma hipótese é rejeitada quando a diferença de grupos tenha probabilidade $p \leq 0,05$ de ter ocorrido ao acaso. Consideramos, então, que os grupos provêm de populações diferentes, indicando que o tratamento (uso de Fragmentos) gera uma diferenciação entre os membros de um conjunto inicial homogêneo.

5.1.2 Experimento 1

A Tabela 3 apresenta as avaliações sobre a rejeição de cada hipótese. As conclusões sobre as hipóteses devem considerar também as informações contidas na Tabela 1 e também o contexto do Experimento 1.

Primeiramente, a rejeição da hipótese nula H_{10} indica que o tempo para escrever Casos de Uso é diferente nos grupos Controle e Tratamento. O grupo Controle escreve Casos de Uso em menor tempo do que o grupo Tratamento.

Tabela 3. Situação de cada hipótese nula do Experimento 1

Hipótese Nula	Situação	p
H1 ₀	Rejeitada	0,083
H2 ₀	Rejeitada	0,002
H3 ₀	Não Rejeitada	0,600
H4 ₀	Rejeitada	0,031
H5 ₀	Rejeitada	0,007
H6 ₀	Rejeitada	0,016

A rejeição da hipótese nula H2₀ mostra que o total de defeitos em Casos de Uso encontrados nos grupos é de fato diferente. O grupo Controle possui uma quantidade quatro vezes maior de defeitos que o grupo Tratamento, indicando que o uso de Fragmentos reduz significativamente a ocorrência geral de defeitos em Casos de Uso.

A rejeição das hipóteses nulas H4₀ e H5₀ indica que as quantidades de defeitos tanto de requisitos como de especificação são diferentes nos grupos Controle e Tratamento. O grupo Controle possui maior quantidade de erros de requisitos e de especificação que o grupo Tratamento. Portanto, o resultado indica que o uso de Fragmentos de Casos de Uso auxilia a boa qualidade de especificação e o entendimento de requisitos de software.

A rejeição da hipótese nula H6₀ mostra que a quantidade de texto usada nos Casos de Uso é diferente para os grupos Controle e Tratamento. A quantidade de texto do grupo Tratamento é cerca de duas vezes maior que a do grupo Controle, indicando que o uso de Fragmentos aumenta o volume de texto de um Caso de Uso em comparação com a não utilização de nenhuma técnica.

A não rejeição da hipótese nula H3₀ indica que a diferença de valores entre os grupos ocorreu ao acaso, isto é, os grupos Controle e Tratamento apresentaram o mesmo número de

defeitos. Esse resultado indica que os Fragmentos de Casos de Uso não auxiliam na redução de defeitos de Casos de Uso, tais como numeração no fluxo básico e de fluxos alternativos.

Os resultados desse experimento apresentaram que, nesse grupo de alunos, os Fragmentos de Casos de Uso reduziram significativamente a ocorrência de **Defeitos de Requisitos** e de **Defeitos de Especificação**, a ocorrência de **Defeitos de Casos de Uso** é estatisticamente a mesma com ou sem Fragmentos.

5.1.3 Experimento 2

A Tabela 4 apresenta as avaliações sobre rejeição de cada hipótese. As conclusões sobre as hipóteses devem considerar também as informações contidas na Tabela 2 e também o contexto do Experimento 2.

Primeiramente, a rejeição da hipótese nula $H1_0$ indica que o tempo para escrever Casos de Uso é diferente nos grupos Controle e Tratamento, isto é, o grupo Controle escreve Casos de Uso mais rapidamente do que o grupo Tratamento.

Tabela 4. Situação de cada hipótese nula do Experimento 2

Hipótese Nula	Situação	p
$H1_0$	Rejeitada	0,00003
$H2_0$	Não Rejeitada	0,138
$H3_0$	Não Rejeitada	0,792
$H4_0$	Rejeitada	0,010
$H5_0$	Não Rejeitada	0,283
$H6_0$	Rejeitada	0,00008

A rejeição da hipótese nula $H4_0$ indica que o número de defeitos de especificação é diferente entre os grupos Controle e Tratamento, com o grupo Controle apresentando uma

maior quantidade de erros de especificação que o grupo Tratamento. Podemos concluir que o uso de Fragmentos de Casos de Uso auxilia a boa qualidade da escrita de requisitos de software.

A rejeição da hipótese nula H_{60} mostra que os grupos Controle e Tratamento possuem quantidade de texto diferente para a descrição de um Caso de Uso. A quantidade de texto do grupo Tratamento é cerca de duas vezes maior que a do grupo Controle, indicando que o uso de Fragmentos aumenta o volume de texto de um Caso de Uso em comparação com a não utilização de nenhuma técnica.

A não rejeição das hipóteses nulas H_{20} , H_{30} e H_{40} mostra que o total de defeitos e os defeitos de Casos de Uso e de requisitos não possuem diferença significativa em ambos os grupos. Os resultados do Experimento 2 mostram que o uso de Fragmentos não auxiliou os alunos na redução tanto dos defeitos de requisitos e de Casos de Uso como a quantidade total de defeitos.

Portanto, os resultados desse experimento mostram que, nesse grupo de alunos, os Fragmentos de Caso de Uso auxiliam a diminuir significativamente somente a ocorrência de **Defeitos de Especificação**. Não há, nesse contexto, diferença de ocorrência de **Defeitos de Casos de Uso** e de **Defeitos de Requisitos**.

5.2 Discussão sobre os Experimentos

5.2.1 Conclusões sobre os resultados

Os resultados dos experimentos permitiram perceber os pontos positivos do uso dos Fragmentos de Casos de Uso, bem como suas limitações e oportunidades de melhorias. Nesta seção vamos discutir cada resultado obtido dos experimentos.

Primeiramente, os resultados do questionário mostram que nem o CRA⁷ nem o Conhecimento dos alunos participantes influenciaram os experimentos. Essa informação é útil para mostrar que os resultados não são tendenciosos quanto à habilidade prévia nem ao rendimento escolar dos alunos.

A segunda observação é quanto ao volume do texto e o tempo de escrita: o grupo Controle escreveu menos e mais rapidamente que o grupo Tratamento. Isso pode ser explicado pela necessidade maior de escrita imposta pelo uso de Fragmentos no grupo Tratamento. A velocidade menor apresentada em ambos os experimentos pelos alunos do grupo Tratamento pode ser explicada pelo tempo gasto pelos alunos tanto no aprendizado da técnica de Fragmentos como na consulta e leitura do Catálogo.

O grupo Controle apresentou uma maior velocidade de escrita, porém com uma qualidade de especificação inferior. Isso pode ser confirmado no Experimento 1, que mostra uma diferença entre os grupos em relação aos defeitos de requisitos e de especificação, e no Experimento 2, que mostra uma diferença em relação aos defeitos de especificação.

O tempo da escrita de uma primeira versão do Caso de Uso é maior utilizando-se os Fragmentos de Casos de Uso. Entretanto, acreditamos que o aumento da qualidade dessa primeira versão possa reduzir o retrabalho global da Engenharia de Requisitos. Essa assertiva poderia ser verificada num outro experimento futuro que buscaria mostrar a redução do retrabalho global e sua conseqüente economia de tempo.

Os dois experimentos constataram que o uso dos Fragmentos não auxilia na diminuição dos defeitos do tipo **Defeitos de Caso de Uso**. Entretanto, em ambos os experimentos foi observado que o uso de Fragmentos auxilia na diminuição de **Defeitos de**

⁷ CRA = Coeficiente de Rendimento Acumulado. É calculado como a média das notas obtidas nas disciplinas já cursadas, ponderadas de acordo com a quantidade de créditos de cada disciplina.

Especificação, ou seja, os textos pré-escritos dos Fragmentos auxiliam a escrita de frases objetivas e com o uso de termos adequados ao tipo de documento, além de sugerir a inserção de outros termos no momento apropriado da especificação de requisitos.

A última observação é relacionada ao fato de não ter sido encontrada diferença entre os grupos em relação à quantidade de **Defeitos de Requisitos** encontrados no Experimento 2. Nossa explicação para esse fato é que os alunos participantes do Experimento 2 tinham muito pouca experiência com a escrita de Casos de Uso. Isso nos permite crer que um conhecimento mínimo sobre como escrever corretamente os requisitos de um software ajuda a melhor utilizar a abordagem de Fragmentos.

Lembramos que os resultados encontrados são diretamente dependentes das condições e dos participantes de ambos os experimentos. Portanto, é possível que novos experimentos possuam outras conclusões diferentes das encontradas neste trabalho, assim como os resultados entre os Experimentos 1 e 2 não foram exatamente os mesmos.

5.2.2 Comparação com outros Trabalhos

A respeito da escrita de Casos de Uso, Cox e Phalp (2000) replicaram o experimento sobre as diretrizes (*guidelines*) do grupo CREWS. Nesse experimento foram usados quatorze alunos de tempo integral de Mestrado em Engenharia de Software. Com quatorze alunos, foram montados três grupos:

- Grupo A: quatro alunos, sem nenhuma técnica adicional para escrever Casos de Uso;
- Grupo B: quatro alunos, para os quais foi dado o *CREWS Style Guideline*;
- Grupo C: três alunos, para os quais foi dado o *CREWS Content Guideline*;
- Grupo D: três alunos, para os quais foram dados ambos os *CREWS Style* e *Content Guidelines*.

Dentre as conclusões desse trabalho, destacamos que o uso de ambos os *Style* e *Content Guidelines* não ajudaram a manter a terminologia consistente na escrita de Casos de Uso e tampouco auxiliaram a estruturação dos mesmos. Os problemas relacionados à terminologia estão contidos no conjunto de **Defeitos de Especificação**.

Os experimentos sobre o uso de Fragmentos de Caso de Uso indicam que o uso de Fragmentos diminui a quantidade desses **Defeitos de Especificação** e, conseqüentemente, os problemas quanto ao domínio, à especificação de interface, aos detalhes internos de projeto e demais problemas de inconsistência de linguagem. Além disso, o uso de Fragmentos por definição obriga a estruturação do texto.

Num outro experimento, Jagielska *et al* (2006) categorizaram os diferentes tipos de defeitos encontrados na escrita de Casos de Uso (*jargon-contaminated*) e discutiram os impactos possíveis na qualidade de especificações de requisitos. Para isso, realizaram um experimento com vinte e dois alunos finalizando o curso de Ciência da Computação. Os defeitos encontrados pertencem à categoria de defeitos de especificação. Os resultados de nossos experimentos mostraram que o uso de Fragmentos diminui a quantidade desse tipo de defeito.

6 CONCLUSÃO

Este capítulo contempla as considerações finais, apresentando as contribuições e as limitações da solução proposta neste trabalho. Em seguida, aponta as sugestões de trabalhos que possam dar continuidade a esta pesquisa, trazendo novas contribuições ou soluções para as limitações da atual proposta.

6.1 Considerações Finais

Este trabalho apresenta a proposta de Fragmentos de Casos de Uso que objetiva diminuir a dificuldade de escrita de Casos de Uso de qualidade por parte de estudantes e profissionais não experientes. A diminuição da dificuldade foi quantificada e observada através da medição do número de defeitos de requisitos, especificação e de Casos de Uso encontrados em dois experimentos realizados com alunos de graduação.

Como resultado desses experimentos, é possível concluir que a proposta de Fragmentos de Casos de Uso:

- auxilia o entendimento de requisitos para um aluno que já possui um conhecimento básico sobre especificação de requisitos. Entretanto, os Fragmentos não substituem capacidade intelectual do aluno necessária para elaborar os requisitos de um sistema;
- auxilia a qualidade da especificação de requisitos, uma vez que, em ambos experimentos, os alunos que usaram Fragmentos produziram documentos com menor quantidade de defeitos de escrita;
- não auxilia na redução dos pequenos defeitos sintáticos de Casos de Uso, como numeração de fluxos e passos

O desenvolvimento deste trabalho envolveu a produção de um Catálogo de Fragmentos de Casos de Uso que pode ser utilizado para a escrita de Casos de Uso de qualidade. Este catálogo contém um conjunto de sete Fragmentos independentes de domínio, podendo ser aplicado em qualquer contexto de especificação de sistemas de informação que utilize a técnica de Casos de Uso.

6.2 Trabalhos Futuros

O uso dos Fragmentos de Casos de Uso, como apresentado nesta dissertação, foi feita de forma manual, o que requer treinamento e leitura do catálogo para a escolha da solução mais adequada. A construção de um artefato computacional que apóie essa abordagem poderia reduzir o tempo de aprendizado, facilitar a escolha do Fragmento adequado e reduzir a quantidade de defeitos sintáticos de Casos de Uso, uma vez que a numeração dos fluxos e passos poderia ser feita automaticamente.

Este trabalho apresentou um pequeno exemplo de uso de Fragmentos. A elaboração de um estudo de caso mais completo e detalhado seria um trabalho futuro útil para a especificação de mais Fragmentos.

A abordagem de Fragmentos pode ser aplicada em qualquer especificação de requisitos de software que tenha sub-objetivos associáveis aos Fragmentos. Mesmo assim, os Fragmentos apresentados neste trabalho são genéricos e de número limitado. Acreditamos que a criação de conjuntos de Fragmentos destinados a domínios específicos permitirá aumentar a facilidade e a qualidade da escrita de Casos de Uso por profissionais sem grande experiência na especificação de requisitos de software.

REFERÊNCIAS

- ALEXANDER, I. F.; STEVENS, R. *Writing Better Requirements*. Pearson Education, 2002.
- BALDAM, R.; VALLE, R.; PEREIRA, H.; HILST, S.; ABREU, M.; SOBRAL, V. *Gerenciamento de Processos de Negócio: BPM – Business Process Management*. Editora Érika, 2007. 240 p.
- BELGAMO, A.; FABBRI, S. GUCCRA. Contribuindo para a Identificação de Defeitos em Documentos de Requisitos Durante a Construção de Modelos de Casos de Uso. *Proceedings of the VII Workshop on Requirements Engineering (WER'04)*, Tandil, Argentina, p. 100-111, 2004.
- BIDDLE, R.; NOBLE, J.; TEMPERO, E. "Patterns for Essencial Use Case Bodies". *3rd Asian Pacific Conference on Pattern Language of Programs (KoalaPLoP)*, Melbourne, Australia, 2002.
- BITTNER, K.; SPENCE, I. *Use Case Modeling*. Addison-Wesley, 2002.
- BOOCH, G., RUMBAUGH, J. and JACOBSON, I. *The Unified Modeling Language User Guide*. 2ª ed. Addison-Wesley, 2005.
- BRIAND, C. L.; LABICHE, Y.; PENTA, M. D.; YAN-BONDOC, H. D. *An Experimental Investigation of Formality in UML-Based Development*. IEEE Transactions on Software Engineering, IEEE Computer Society, v. 31, n. 10, Out., 2005.
- BUSSAB, W. O.; MORETTIN, P. A. *Estatística Básica*. 5ª ed. Saraiva, 2002.
- COCKBURN, A. *Writing Effective Use Cases*. Addison-Wesley, 2001.
- COCKBURN, A.; ADOLPH, S.; BRAMBLE, P.; POLS, A. *Patterns for Effective Use Cases*. Addison-Wesley, 2002.
- CONSTANTINE, L. L.; LOCKWOOD, L. A. *Software for Use*. Addison-Wesley, 1999.
- CORREA, A. L. *Reestruturando Especificações de Restrições de Modelos Elaboradas em OCL*. 2006. 252 f. Tese de Doutorado, Universidade Federal do Rio de Janeiro, 2006.
- COX, K.; AURUM, A.; JEFFERY, R. *An Experiment in Inspecting the Quality of Use Case Descriptions*. Journal of Research and Practice in Information Technology, v. 36, n. 4, 2004.
- COX, K.; PHALP, K. *Replicating the CREWS Use Case Authoring Guidelines Experiment*. Empirical Software Engineering Journal, v. 5, n. 3, p. 245-267, 2000.
- CRUZ, P. O. S. *Heurísticas para Identificação de Requisitos de Sistemas de Informações a partir de Modelos de Processos*. 2004. 91 f. Dissertação de Mestrado, Universidade Federal do Rio de Janeiro, 2004.

DENNEY, R. *Succeeding with Use Cases: Working Smart to Deliver Quality*. Addison-Wesley Professional, 2005.

DIAS, F. G.; MORGADO, G. P.; OSCAR, P.; SILVEIRA, D.; ALENCAR, A. J.; LIMA, P. M. V.; SCHMITZ, E. A. Uma Abordagem para a Transformação Automática do Modelo de Negócio em Modelo de Requisitos. *Proceedings of the IX Workshop on Requirements Engineering*, Rio de Janeiro, Brasil, 2006.

DIAS, F. G.; SCHMITZ, E. A.; CAMPOS, M. L. M. Padrões de Fragmento de Casos de Uso. *Anais do XII Workshop de Teses e Dissertações (WTES'07) do XXI Simpósio Brasileiro de Engenharia de Software (SBES'07)*, João Pessoa, Paraíba, Brasil, 2007.

DIAS, F. G.; SCHMITZ, E. A.; CAMPOS, M. L. M.; CORREA, A. L.; ALENCAR, A. J. Elaboration of Use Case Specifications: an approach based on Use Case Fragments. *Proceedings of the 23rd Annual ACM Symposium on Applied Computing, 1st Requirements Engineering Track*, Fortaleza, Ceará, p. 614-618, 2008.

DIJKMAN, R. M.; JOOSTEN, S. M. M. Deriving Use Case Diagrams from Business Process Models. *Proceedings of the International Conference on Conceptual Modeling (ER'03)*, Chicago, Illinois, 2003.

EL-ATTAR, M.; MILLER, J. Matching Antipatterns to Improve the Quality of Use Case Models. *Proceeding of the 14th IEEE International Requirements Engineering Conference (RE'06)*, p. 99-108, 2006.

ERKSSON, H.; PENKER, M. *Business Modeling with UML: Business Patterns at Work*. OMG Press, John Wiley & Sons, 2000. 459 p.

FANTECHI, A.; GNESI, S.; LAMI, G. MACCARI, A. *Applications of linguistic techniques for use case analysis*. Requirement Engineering Journal, v. 8, Springer, 2003.

FOGELSTRÖM, N. D.; GORSCHKEK, T. Test-case Driven versus Checklist-based Inspections of Software Requirements – An Experimental Evaluation. *Proceedings of the X Workshop on Requirements Engineering (WER'07)*, Toronto, Canadá, p. 116-126, 2007.

FORTUNE, J.; PETERS, G. *Information Systems: Achieving Success by avoiding failure*. John Wiley & Sons, 2005. 220 p.

FOWLER, M. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. 3^a ed. Addison-Wesley Professional, 2003. 208 p.

GOLDSMITH, R. F. *Discovering Real Business Requirements for Software Project Success*. Artech House, 2004.

HAZAN, C.; BERRY, D. M.; LEITE, J. C. S. P. É possível substituir processos de Engenharia de Requisitos por Contagem de Pontos de Função? *Proceedings of the VIII Workshop on Requirements Engineering (WER'05)*, Porto, Portugal, p. 197-208, 2007.

HULL, E.; JACKSON, K.; DICK, J. *Requirements Engineering*. 2^a ed. Springer, 2005.

ISO - International Standard Organization. *Information Technology - Software Life Cycle Processes*. ISO/IEC 12207, 1995.

- JAGIELSKA, D.; WERNICK, P.; WOOD, M.; BENNETT, S. How Natural is Natural Language? How Well do Computer Science Students Write Use Cases? *Conference on Object Oriented Programming Systems Languages and Applications (OOPSLA '06)*, Portland, Oregon, EUA, 2006.
- JACOBSON, I. *Object-Oriented Software Engineering*. Addison-Wesley, 1992.
- JACOBSON, I.; NG, P. *Aspect-Oriented Software Development with Use Cases*. Addison-Wesley, 2004.
- JURISTO, N.; MORENO, A. *Basics of Software Engineering Experimentations*. Springer, 2001.
- KIYAVITSKAYA, N.; ZENI, N.; MICH, L.; BERRY, D. M. Requirements for Tools for Ambiguity Identification and Measurement in Natural Language Requirements Specifications. *Proceedings of the X Workshop on Requirements Engineering (WER'07)*, Toronto, Canadá, p. 197-206, 2007.
- KNIGHT, D. M. *Elicitação de Requisitos de Software a partir do Modelo de Negócio*. Dissertação de Mestrado, Universidade Federal do Rio de Janeiro, 2004.
- KULAK, D.; GUINEY, E. *Use Cases: Requirements in Context*. 2ª ed. Addison-Wesley, 2003.
- LARMAN, C. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process*. 2ª ed. Prentice Hall, 2001.
- LEFFINGWELL, D.; WIDRIG, D. *Managing Software Requirements: A Use Case Approach*, 2nd ed., Addison-Wesley, 2003. 544 p.
- LEONARDI, M. C.; MAUCO, M. V. Integrating Natural Language Oriented Requirements Models into MDA. *Proceedings of the VII Workshop on Requirements Engineering (WER'04)*, Tandil, Argentina, p. 65-76, 2004.
- LIU, D.; SUBRAMANIAM, K.; EBERLEIN, A.; FAR, B. H. *Natural Language Requirements Analysis and Class Model Generation Using UCDA*. Lecture Notes in Computer Science, Innovations in Applied Artificial Intelligence, v. 3029/2004, p. 295-304, Springer, 2004.
- MARTÍNES, A.; PASTOR, O.; ESTRADA, H. A Pattern Language to join Early and Late Requirements. *Proceedings of the VII Workshop on Requirements Engineering (WER'04)*, Tandil, Argentina, p. 51-64, 2004.
- MARTINS, J. C. C. *Técnicas para Gerenciamento de Projetos de Software*. Brasport, 2007. 432 p.
- MCGOVERN, F. *Managing Software Projects with Business-Based Requirements*. IEEE IT-Pro Journal, Set./Out., p. 18-23, 2002.
- MELO, A. C. *Exercitando Modelagem em UML*. Rio de Janeiro: Brasport, 2006. 123 p.

MERRICK, P.; BARROW, P. *Testing the predictive ability of a requirements pattern language*. Requirements Engineering Journal, Springer, Abril, p. 85-94, 2005.

MOON, M.; YEOM, K.; CHAE, H. S. *An Approach to Developing Domain Requirements as a Core Asset*. IEEE Transactions on Software Engineering, IEEE Computer Society, v. 31, n. 7, Julho, 2005.

MORGAN, T. *Business Rules and Information Systems: Aligning IT with Business Goals*. Addison-Wesley, 2002.

OMG. *Semantics of Business Vocabulary and Business Rules Specification*. 2008. Disponível em: <<http://www.omg.org/spec/SBVR/1.0/>>, acesso em 06 set. 2009, 16:30:00.

ÖVERGAARD, G.; PALMKVIST, K. *Use Cases: Patterns and Blueprints*. Addison-Wesley, 2004.

KNICHTEN, P. The 4+1 View Model of Architecture. *IEEE Softloare*, v. 12, n. 6, p. 42– 50, 1995.

NOVember, 1995. PREECE, J.; ROGERS, Y.; SHARP, H. *Design de Interação: Além da Interação Homem-Computador*. Bookman, 2005.

PRESMAN, R. S. *Software Engineering: A Practitioner's Approach*. 5ª ed. McGrall-Hill, 2001.

RIDAO, M.; DOORN, J.; LEITE, J. C. S. do P. Domain Independent Regularities in Scenarios. *Proceedings of the IEEE Joint International Conference on Requirements Engineering (RE'01)*, p. 120-127, 2001.

ROBERTSON, S.; ROBERTSON, J. *Mastering the Requirements Process*. 2ª ed. Addison-Wesley Professional, 2006.

ROSENBERG, D.; SCOTT, K. *Applying Use Case Driven Object Modeling with UML*. Addison-Wesley, 2001.

ROSENBERG, D.; STEPHENS, M. *Use Case Driven Object Modeling with UML*. Apress, 2007. 438 p.

SANTANDER, V. F. A.; CASTRO, J. F. B. Deriving Use Cases from Organizational Modeling. *Proceedings of the IEEE Joint International Conference on Requirements Engineering (RE'02)*, p. 32-41, 2002.

SCHMITZ, E. A.; ALENCAR, A. J.; AZEVEDO, C. M. Defining the Implementation Order of Software Projects under Uncertain Environments. *Proceedings of the 10th International Conference on Enterprise Information System (ICEIS'08)*, v. I, p.100 – 105, Barcelona, Espanha, 2008.

SEVERINO, A. J. *Metodologia do Trabalho Científico*. 23ª ed. Editora Cortez, 2007.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. *Sistema de Banco de Dados*. 3ª ed. Pearson Makron Books, 2005.

SOMMERVILLE, I.; SAWYER, P. *Requirements Engineering: A Good Practice Guide*. Wiley, 1997.

STATSOFT, Inc. *STATISTICA for Windows*. Manual de Programa de Computador, 1999.

SVETINOVIC, D.; BERRY, D. M.; DAY, N. A.; GODFREY, M. W. *Unified Use Case Statecharts: Case Studies*. Requirements Engineering Journal, Springer, Junho, 2007.

TJONG, S. F.; HARTLEY, M.; BERRY, D. M. Extended Disambiguation Rules for Requirements Specifications. *Proceedings of the X Workshop on Requirements Engineering (WER'07)*, Toronto, Canadá, p. 97-106, 2007.

WAZLAWICK, R. S. *Análise e Projeto de Sistemas de Informação Orientados a Objetos*. Elsevier, 2004.

WEILL, P.; BROADBENT, M. *Leveraging the New Infrastructure: How Market Leaders Capitalize on Information Technology*. Harvard Business School Press. 1998.

WINDLE, D. R.; ABREO, L. R. *Software Requirements Using the Unified Process: A Practical Approach*. Prentice Hall, 2002.

WOHLIN, C.; RUNESON, P.; HÖST, M.; OHLSSON, M.; REGNELL, B.; WESSLÉN, A. *Experimentation in Software Engineering – An Introduction*. Kluwer Academic Publishers, 2000.

YOUNG, R. R. *The Requirements Engineering Handbook*. Artech House, 2004.

YU, E. *Modeling Strategic Relationships for Process Reengineering*. 1995. Tese de Doutorado, Universidade de Toronto, 1995.

YU, Y.; LEITE, J. C. S. P.; LAPOUCHNIAN, A.; MYLOPOULOS, J. Configuring Features with Stakeholder Goals. *Proceedings of the 23rd ACM Symposium on Applied Computing (SAC'08)*, Fortaleza, Ceará, Brasil, p. 645-649, 2008.

7 APÊNDICE A – CATÁLOGO DE FRAGMENTOS DE CASOS DE USO

7.1 Fragmento: Seleção Única de uma Lista

Nome do Fragmento: Seleção Única de uma Lista
Sub-Objetivo: Selecionar um elemento de um conjunto de elementos existentes.
<p>Fluxo Básico:</p> <ol style="list-style-type: none"> 1. <ator> informa <parâmetros_para_seleção>. 2. Sistema certifica que <parâmetros_para_seleção> são válidos de acordo com <regras_relacionadas_aos_parâmetros_para_seleção>. 3. Sistema apresenta uma lista de <objeto>. 4. <ator> seleciona um <objeto>. 5. Sistema apresenta detalhes de <objeto> selecionado. <p><i>passos opcionais:</i> 1, 2 e 5</p>
<p>Fluxos Alternativos:</p> <p><i>a) <parâmetros_para_seleção> inválidos</i> No passo 1 do Fluxo Básico, <ator> informa um parâmetro para seleção que viola <regras_relacionadas_aos_parâmetros_para_seleção>.</p> <ol style="list-style-type: none"> 1. Sistema apresenta mensagem de acordo com as regras que tenham sido violadas. 2. Fluxo de eventos retorna ao passo 1 do Fluxo Básico. <p><i>b) nenhum <objeto> satisfaz à seleção</i> No passo 3 do Fluxo Básico, Sistema não encontra nenhum <objeto> para seleção.</p> <ol style="list-style-type: none"> 1. Sistema apresenta mensagem: “<objeto> não encontrado”. 2. Fluxo de eventos retorna ao passo 1 o Fluxo Básico. <p><i>fluxos opcionais:</i> a)</p>
<p>Detalhes de Estruturas de Dados:</p> <p>a) <parâmetros_para_seleção> = definição de cada informação que pode ser informada pelo ator.</p> <p>b) Lista de <objeto> = definição de</p> <ul style="list-style-type: none"> • lista de <propriedades> extraídas de um objeto que devem ser apresentadas ao ator. • objetos dentre os quais as propriedades serão extraídas. Precisa especificar como os parâmetros para seleção influenciam essa decisão, se necessário. • informação sobre a ordem dos elementos da lista. <p>c) Detalhes de <objeto> = lista das propriedades extraídas de <objeto>.</p> <p><i>detalhes opcionais:</i> a) e c)</p>
<p>Detalhe de Regras:</p> <p>a) <regras_relacionadas_a_parâmetros_para_seleção> = definição de todas as regras que devem ser aplicadas para validar os parâmetros informados para a seleção dos objetos.</p> <p><i>detalhes opcionais:</i> a)</p>

7.2 Fragmento: Seleção Múltipla de uma Lista

<p>Nome do Fragmento: Seleção Múltipla de uma Lista</p>
<p>Sub-Objetivo: Selecionar zero, um ou mais elementos de um conjunto de elementos existentes.</p>
<p>Fluxo Básico:</p> <ol style="list-style-type: none"> 1. <ator> informa <parâmetros_para_seleção>. 2. Sistema certifica que <parâmetros_para_seleção> são válidos de acordo com <regras_relacionadas_aos_parâmetros_para_seleção>. 3. Sistema apresenta uma lista de <objeto>. 4. <ator> seleciona zero, um ou mais <objeto>. 5. Sistema apresenta detalhes de <objeto> selecionado. <p><i>passos opcionais:</i> 1, 2 e 5</p>
<p>Fluxos Alternativos:</p> <p><i>a) <parametros_para_seleção> inválidos</i> No passo 1 do Fluxo Básico, <ator> informa um parâmetro para seleção que viola <regras_relacionadas_aos_parâmetros_para_seleção>.</p> <ol style="list-style-type: none"> 1. Sistema apresenta mensagem de acordo com as regras que tenham sido violadas. 2. Fluxo de eventos retorna ao passo 1 do Fluxo Básico. <p><i>b) nenhum <objeto> satisfaz à seleção</i> No passo 3 do Fluxo Básico, Sistema não encontra nenhum <objeto> para seleção.</p> <ol style="list-style-type: none"> 1. Sistema apresenta mensagem: “<objeto> não encontrado”. 2. Fluxo de eventos retorna ao passo 1 o Fluxo Básico. <p><i>fluxos opcionais:</i> a)</p>
<p>Detalhes de Estruturas de Dados:</p> <p><i>a) <parâmetros_para_seleção> = definição de cada informação que pode ser informada pelo ator.</i></p> <p><i>b) Lista de <objeto> = definição de</i></p> <ul style="list-style-type: none"> • lista de <propriedades> extraídas de um objeto que devem ser apresentadas ao ator. • objetos dentre os quais as propriedades serão extraídas. Precisa especificar como os parâmetros para seleção influenciam essa decisão, se necessário. • informação sobre a ordem dos elementos da lista. <p><i>c) Detalhes de <objeto> = lista das propriedades extraídas de <objeto>.</i></p> <p><i>detalhes opcionais:</i> a) e c)</p>
<p>Detalhe de Regras:</p> <p><i>a) <regras_relacionadas_a_parâmetros_para_seleção> = definição de todas as regras que devem ser aplicadas para validar os parâmetros informados para a seleção dos objetos.</i></p> <p><i>detalhes opcionais:</i> a)</p>

7.3 Fragmento: Obter Detalhes de um Objeto

Nome do Fragmento: Obter Detalhes de um Objeto
Sub-Objetivo: Obter os detalhes de um objeto fornecendo o identificador desse objeto.
Fluxo Básico: <ol style="list-style-type: none"> 1. <ator> informa <identificador> de <objeto>. 2. Sistema apresenta detalhes de <objeto>.
Fluxos Alternativos: <p><i>a) <objeto> não encontrado.</i></p> <p>No passo 2 do Fluxo Básico, o Sistema não encontra um <objeto> relacionado ao <identificador> fornecido.</p> <ol style="list-style-type: none"> 1. Sistema apresenta a mensagem: “Nenhum <objeto> encontrado”. 2. Fluxo de eventos retorna ao passo 1 do Fluxo Básico.
Detalhes de Estruturas de Dados: <p>a) Detalhes de <objeto> = lista de <propriedades>.</p>
Detalhe de Regras: <p><i>não há</i></p>

7.4 Fragmento: Obter Confirmação de Registro

<p>Nome do Fragmento: Obter Confirmação de Registro</p>
<p>Sub-Objetivo: Obter Confirmação a respeito do Registro de um conjunto de informações no Sistema.</p>
<p>Fluxo Básico:</p> <ol style="list-style-type: none"> 1. <ator> solicita o registro de <objeto>. 2. Sistema apresenta dados de <objeto>. 3. <ator> confirma registro de <objeto>. 4. Sistema apresenta os detalhes do recibo de registro.
<p>Fluxos Alternativos:</p> <p><i>a) registro cancelado.</i> No passo 3 do Fluxo Básico, o <ator> decide cancelar o registro.</p> <ol style="list-style-type: none"> 1. Sistema apresenta a mensagem: “Registro cancelado”. 2. Fluxo de eventos retorna ao passo 1 do Fluxo Básico. <p><i>b) regra de registro violada.</i> No passo 4 do Fluxo Básico, o Sistema não permite o registro de <objeto> devido à violação da <regra_de_registro>.</p> <ol style="list-style-type: none"> 1. Sistema apresenta mensagem de acordo com as regras que tenham sido violadas. 2. Fluxo de eventos retorna ao passo 1 do Fluxo Básico. <p><i>fluxos opcionais:</i> b)</p>
<p>Detalhes de Estruturas de Dados:</p> <ol style="list-style-type: none"> a) Dados de <objeto> = <conjunto_de_propriedades> a ser registrado. b) Detalhes do recibo de registro = lista de <propriedades>.
<p>Detalhe de Regras:</p> <p>a) <regra_de_registro> = definição de todas as regras que devem ser aplicadas para validar os parâmetros informados para o registro de um dado objeto.</p> <p><i>detalhes opcionais:</i> a)</p>

7.5 Fragmento: Obter Confirmação de Envio de Arquivo

Nome do Fragmento: Obter Confirmação de Envio de Arquivo
Sub-Objetivo: Obter confirmação a respeito do envio de um arquivo ao Sistema.
<p>Fluxo Básico:</p> <ol style="list-style-type: none"> 1. <ator> informa a localização física do arquivo. 2. Sistema certifica que a localização física do arquivo é válida de acordo com <regras_de_envio_de_arquivo>. 3. <ator> solicita envio do arquivo. 4. Sistema armazena o arquivo enviado por <ator>. 5. Sistema apresenta os detalhes do recibo de armazenamento.
<p>Fluxos Alternativos:</p> <p><i>a) localização física do arquivo inválido.</i> No passo 2 do Fluxo Básico, <ator> informa uma localização física inválida de acordo com <regras_de_envio_de_arquivo>.</p> <ol style="list-style-type: none"> 1. Sistema apresenta uma mensagem de acordo com as regras violadas.
<p>Detalhes de Estruturas de Dados:</p> <p>a) Localização física = definição da estrutura esperada para a localização física do arquivo. b) Detalhes do recibo de armazenamento = lista de <propriedades>.</p>
<p>Detalhe de Regras:</p> <p>a) < regras_de_envio_de_arquivo > = definição de todas as regras que devem ser aplicadas para validar a localização física do arquivo a ser enviado.</p>

7.6 Fragmento: Obter Confirmação de Exclusão de Objeto

Nome do Fragmento: Obter Confirmação de Exclusão de Objeto
Sub-Objetivo: Obter confirmação a respeito da exclusão de um determinado objeto do sistema.
Fluxo Básico: <ol style="list-style-type: none"> 1. <ator> solicita a exclusão de <objeto>. 2. <ator> confirma exclusão de <objeto>. 3. Sistema exclui <objeto>. 4. Sistema armazena o arquivo enviado por <ator>. 5. Sistema apresenta os detalhes do recibo de exclusão.
Fluxos Alternativos: <p><i>a) exclusão de <objeto> cancelada.</i></p> <p>No passo 2 do Fluxo Básico, <ator> cancela a exclusão de <objeto>.</p> <ol style="list-style-type: none"> 1. Sistema apresenta a mensagem: “Exclusão cancelada”. 2. Fluxo de eventos retorna ao passo 2 do Fluxo Básico.
Detalhes de Estruturas de Dados: <p>a) Detalhes do recibo de exclusão = lista de <propriedades>.</p>
Detalhe de Regras: <i>não há</i>

7.7 Fragmento: Buscar Objetos

Nome do Fragmento: Buscar Objetos
Sub-Objetivo: Buscar por instâncias de um dado objeto de interesse.
<p>Fluxo Básico:</p> <ol style="list-style-type: none"> 1. <ator> informa conteúdo relacionado a <objeto>. 2. <ator> informa <parâmetros_para_busca>. 3. <ator> solicita busca. 4. Sistema apresenta uma lista de <objeto> de acordo com <regras_de_busca>. <p><i>passos opcionais: 2</i></p>
<p>Fluxos Alternativos:</p> <p><i>a) nenhum <objeto> satisfaz à busca.</i></p> <p>No passo 4 do Fluxo Básico, Sistema não encontra nenhum <objeto> para o resultado.</p> <ol style="list-style-type: none"> 1. Sistema apresenta mensagem: “<objeto> não encontrado”. 2. Fluxo de eventos retorna ao passo 4 o Fluxo Básico.
<p>Detalhes de Estruturas de Dados:</p> <p>a) <parâmetros_para_busca> = definição de cada informação que pode ser informada pelo ator para influenciar na busca.</p> <p>b) Lista de <objeto> = definição de</p> <ul style="list-style-type: none"> • lista de <propriedades> extraídas de um objeto que devem ser apresentadas ao ator. • objetos dentre os quais as propriedades serão extraídas. Precisa especificar como os parâmetros para seleção influenciam essa decisão, se necessário. • informação sobre a ordem dos elementos da lista. <p><i>detalhes opcionais: a)</i></p>
<p>Detalhe de Regras:</p> <p>a) <regras_de_busca> = definição de todas as regras que devem ser aplicadas para realizar a busca por objetos que satisfaçam à informação do ator.</p> <p><i>detalhes opcionais: a)</i></p>

APÊNDICE B – ENUNCIADO DO EXPERIMENTO

Caso de Uso: Registrar Inscrição em Disciplinas

Breve Descrição

Este Caso de Uso permite a um Aluno registrar uma inscrição nas disciplinas oferecidas no semestre corrente. Há um catálogo que provê uma lista de todas as ofertas de disciplinas para o semestre corrente. Antes de selecionar as disciplinas desejadas, o aluno deve selecionar a matrícula (graduação, mestrado etc) para qual deseja selecionar disciplinas. O aluno somente tem permissão para se inscrever nas disciplinas relacionadas ao seu curso. Considere que o aluno já esteja devidamente autenticado ao acessar essa funcionalidade do *software*. Regras para permitir uma inscrição: *i*) o aluno não tem permissão para se inscrever em duas ou mais disciplinas no mesmo horário, *ii*) o somatório de créditos das disciplinas selecionadas deve obrigatoriamente ser menor ou igual a 36 créditos e *iii*) a inscrição deve obrigatoriamente estar relacionada a pelo menos uma disciplina.

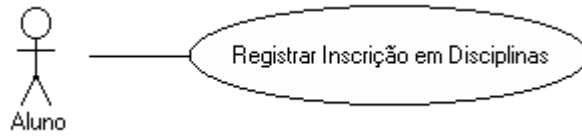
Objetivo Principal do Aluno

Registrar uma inscrição em disciplinas

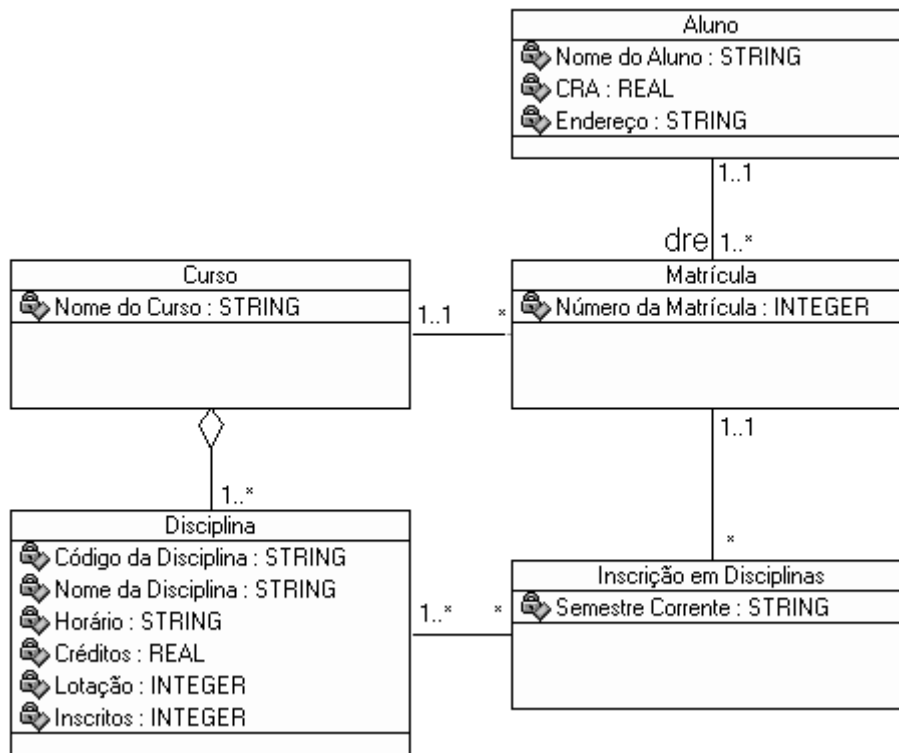
Sub-Objetivos do Aluno

1. Selecionar uma Matrícula dentre as Matrículas do aluno em questão.
2. Selecionar as Disciplinas desejadas, dentre as Disciplinas disponíveis relacionadas à Matrícula em questão.
3. Obter a confirmação de registro da Inscrição em Disciplinas

Diagrama de Casos de Uso



Termos do Negócio



APÊNDICE C – DADOS COLETADOS NO EXPERIMENTO 1

Tabela 5. Dados obtidos do questionário e do exercício no Experimento 1

Aluno	Grupo	Tempo	CRA	Conhecimento	Palavras
Aluno 1	Controle	58	8.3	8,5	365
Aluno 2	Tratamento	61	6.6	8	370
Aluno 3	Tratamento	46	4.5	5	515
Aluno 4	Controle	40	7.2	5	193
Aluno 5	Controle	45	6	0	383
Aluno 6	Controle	75	5.7	6	351
Aluno 7	Tratamento	61	6.1	6	382
Aluno 8	Tratamento	53	7.8	7	440
Aluno 9	Controle	40	5.5	5	260
Aluno 10	Tratamento	44	5.8	5	342
Aluno 11	Tratamento	69	6	5	412
Aluno 12	Controle	46	6	6	335
Aluno 13	Tratamento	54	6.1	7	307
Aluno 14	Controle	40	8.2	8	210
Aluno 15	Tratamento	65	6	7	398
Aluno 16	Controle	45	6.5	7	197

Tabela 6. Defeitos encontrados a partir da aplicação do *Checklist* no Experimento 1

Aluno	Grupo	1.1	1.2	2.1	2.2	2.3	3.	4.1	4.2	5.	6.1	6.2	Def_Total	Def_UC	Def_Esp	Def_Req
Aluno 1	Controle		1				1				2	2	6	2	1	3
Aluno 2	Tratamento											3	3	3	0	0
Aluno 3	Tratamento		1								1		2	0	0	2
Aluno 4	Controle	3				1					1		5	0	0	5
Aluno 5	Controle	3					3				1		7	0	3	4
Aluno 6	Controle		3			1	4				1		9	0	4	5
Aluno 7	Tratamento					1							1	0	0	1
Aluno 8	Tratamento											2	2	2	0	0
Aluno 9	Controle	3	1			5	1				1		11	0	1	10
Aluno 10	Tratamento	2				3					1		6	0	0	6
Aluno 11	Tratamento					1	3					2	6	2	3	1
Aluno 12	Controle	4	1			1	12			1	2	2	23	3	12	8
Aluno 13	Tratamento											1	1	1	0	0
Aluno 14	Controle	2					7						9	0	7	2
Aluno 15	Tratamento											1	1	1	0	0
Aluno 16	Controle	3				2						2	7	2	0	5

APÊNDICE D – DADOS COLETADOS NO EXPERIMENTO 2

Tabela 7. Dados obtidos do questionário e do exercício no Experimento 2

Aluno	Grupo	Tempo	CRA	Conhecimento	Palavras
Aluno 1	Controle	37	4.9	5	182
Aluno 2	Controle	42	7.3	6	174
Aluno 3	Controle	50	7.7	4	441
Aluno 4	Controle	34	7.9	5	272
Aluno 5	Tratamento	60	6.4	3	363
Aluno 6	Controle	40	8.5	5	372
Aluno 7	Tratamento	45	7.1	6	521
Aluno 8	Controle	42	6.6	7	484
Aluno 9	Tratamento	86	5	6	482
Aluno 10	Controle	60	6	6	307
Aluno 11	Controle	42	6.4	8,5	247
Aluno 12	Controle	36	6.8	6	331
Aluno 13	Tratamento	62	6.4	3	605
Aluno 14	Tratamento	57	6.5	7	604
Aluno 15	Tratamento	59	6.2	3	394
Aluno 16	Tratamento	84	5.3	5	472
Aluno 17	Controle	30	6.8	8	319
Aluno 18	Tratamento	54	6.3	6	637
Aluno 19	Tratamento	58	8.1	6	465
Aluno 20	Controle	46	8.5	8	307
Aluno 21	Controle	53	5.5	3	294
Aluno 22	Controle	63	8.5	4	215
Aluno 23	Tratamento	70	8.7	8	374
Aluno 24	Controle	35	6	7	227

Aluno 25	Tratamento	60	7.5	7	415
Aluno 26	Tratamento	66	5.5	8,5	586
Aluno 27	Controle	41	7.5	0	182
Aluno 28	Controle	48	5.1	4	425
Aluno 29	Controle	40	7.5	6	231
Aluno 30	Tratamento	62	6.3	8	264
Aluno 31	Controle	35	6.5	7	230
Aluno 32	Tratamento	69	8.8	5	421
Aluno 33	Tratamento	60	6.5	7	451
Aluno 34	Tratamento	110	6.7	7	434
Aluno 35	Tratamento	45	5	5	433

Tabela 8. Defeitos encontrados a partir da aplicação do *Checklist* no Experimento 2

Aluno	Grupo	1.1	1.2	2.1	2.2	2.3	3.	4.1	4.2	5.	6.1	6.2	Def_Total	Def_UC	Def_Esp	Def_Req
Aluno 1	Controle	1								2			3	2	0	1
Aluno 2	Controle	2											2	0	0	2
Aluno 3	Controle	2					1				1		4	0	1	3
Aluno 4	Controle	2								6			8	6	0	2
Aluno 5	Tratamento												0	0	0	0
Aluno 6	Controle		1										1	0	0	1
Aluno 7	Tratamento												0	0	0	0
Aluno 8	Controle					1	1				1		3	0	1	2
Aluno 9	Tratamento								1			2	3	3	0	0
Aluno 10	Controle					1					1	2	4	2	0	2
Aluno 11	Controle	3	1				1					3	8	3	1	4
Aluno 12	Controle	2									1		3	0	0	3
Aluno 13	Tratamento	5	5								2	2	14	2	0	12
Aluno 14	Tratamento	3				1						1	5	1	0	4
Aluno 15	Tratamento	1	1										2	0	0	2

Aluno 16	Tratamento	1	2	1		1							5	0	0	5
Aluno 17	Controle												0	0	0	0
Aluno 18	Tratamento	3				1						2	6	2	0	4
Aluno 19	Tratamento												0	0	0	0
Aluno 20	Controle	3				1			1				5	1	1	3
Aluno 21	Controle		2			2			8				12	8	2	2
Aluno 22	Controle	3	2						1				6	1	0	5
Aluno 23	Tratamento												0	0	0	0
Aluno 24	Controle	4		1		4			2				11	2	4	5
Aluno 25	Tratamento	1			1						1		3	1	0	2
Aluno 26	Tratamento	1	3							1	1		6	1	0	5
Aluno 27	Controle	5				2							7	0	2	5
Aluno 28	Controle	2				1	1			1			5	1	1	3
Aluno 29	Controle	1				2							3	0	2	1
Aluno 30	Tratamento	3	1			1	1				1	1	8	1	1	6
Aluno 31	Controle	5				5			1		3		14	4	5	5
Aluno 32	Tratamento	1	1								2		4	2	0	2
Aluno 33	Tratamento										2		2	2	0	0
Aluno 34	Tratamento												0	0	0	0
Aluno 35	Tratamento	1							1		2		4	3	0	1

APÊNDICE E – QUESTIONÁRIO DO EXPERIMENTO

Nome:

DRE:

Horário de início do experimento:

Horário de término do experimento:

Grupo do experimento (controle ou tratamento):

1. Qual o seu CRA (aproximadamente)?

2. De 0 a 10, que nota você dá para seu conhecimento sobre a escrita de casos de uso? (0 – nunca escrevi nenhum, 10 – tenho muita experiência na escrita de casos de uso)

3. Você já escreveu casos de uso reais, fora do ambiente da universidade? Se positivo, onde?

4. De 0 a 10, que nota você dá para a facilidade de ter realizado o experimento? (0 – quase impossível de se fazer, 10 – foi extremamente fácil!)

5. De 0 a 10, que nota você dá para seu interesse por essa área de pesquisa, de Requisitos? (0 – não se interessa nada, 10 – se interessa muito!)

6. Críticas, elogios ou sugestões sobre o Experimento:

ANEXO A – *CHECKLIST* EM PORTUGUÊS

Neste Anexo A é apresentado o *checklist* para a busca por defeitos de descrições de Casos de Uso. O *checklist* original pode ser encontrado no trabalho de Cox *et al* (2004).

1. Cobertura.

1.1. Completeza: O recurso deverá conter todos os casos que são necessários para responder ao problema. Isto é, se houver informações suficientes na descrição ou se algum detalhe faltando.

1.2. Escopo: O uso caso deveria conter apenas detalhe relevante para o problema. Informação fornecida desnecessariamente fora do âmbito da aplicação e do problema não é necessária.

2. Lógica.

2.1 Ordem do Texto: A utilização deve seguir um caminho lógico. Este caminho é lógico ou há eventos na descrição de forma incorreta?

2.2 Dependências: A transação deve ter um fim (que podem incluir fluxos alternativos / excepcionais). O ator chega a uma situação que impede a operação de encerrar como é esperado?

2.3 Resposta Racional: A lógica do caso deverá proporcionar uma resposta plausível para o problema. Existem alguns eventos que parecem estar fora do lugar ou se reconhecem como incorretos?

3. Abstração Consistente: o Caso de Uso deve conter um nível de abstração coerente. Mistura de níveis de abstração (problema de domínio, especificação de interface, detalhes internos de projeto) irá causar dificuldade de compreensão. Abstração é consistente?

4. Estrutura consistente.

4.1 Variações: Acontecimentos excepcionais e alternativos devem ser excluídos do fluxo principal e devem estar em uma seção separada.

4.2 Seqüência: A numeração de eventos no fluxo principal deve ser consistente. Há alguma incoerência?

5. Linguagem consistente: tempo verbal no presente do indicativo deve ser usado durante todo o tempo. Advérbios, adjetivos, pronomes, sinônimos e negativos devem ser evitados. Eles têm sido utilizados?

6. Fluxos de Eventos Alternativos e de Exceção.

6.1 Viável: Fluxos alternativos e de exceções devem fazer sentido e devem ser completos. Fazem sentido e são completos?

6.2 Numeração: Numeração de fluxos alternativos e de exceção devem ser iguais ao número no fluxo principal. Há incoerência nessa numeração?