



**PODER EXECUTIVO  
MINISTÉRIO DA EDUCAÇÃO  
UNIVERSIDADE FEDERAL DO AMAZONAS  
INSTITUTO DE COMPUTAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA**



**UM MODELO SOBRE AS DIFICULDADES PARA MODELAR  
CASOS DE USO**

**ELIZAMARY DE SOUZA NASCIMENTO**

Manaus, Junho de 2017.

**ELIZAMARY DE SOUZA NASCIMENTO**

**UM MODELO SOBRE AS DIFICULDADES PARA MODELAR CASOS DE USO**

Dissertação submetida ao corpo docente do Programa de Pós-Graduação em Informática da Universidade Federal do Amazonas (PPGI-UFAM) como requisito para obtenção do título de Mestre em Informática.

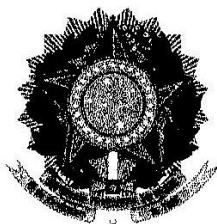
Orientadora: Prof<sup>ª</sup>. Tayana Uchôa Conte, D.Sc.

Manaus, Junho de 2017.

### Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

N244u	Nascimento, Elizamary de Souza Um Modelo sobre as Dificuldades para Modelar Casos de Uso / Elizamary de Souza Nascimento. 2017 154 f.: il. color; 31 cm.  Orientadora: Tayana Uchôa Conte Dissertação (Mestrado em Informática) - Universidade Federal do Amazonas.  1. Casos de Uso. 2. Dificuldades na modelagem de casos de uso. 3. Estudo experimental. 4. Modelo de dificuldades. I. Conte, Tayana Uchôa II. Universidade Federal do Amazonas III. Título
-------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



PODER EXECUTIVO  
MINISTÉRIO DA EDUCAÇÃO  
INSTITUTO DE COMPUTAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA



UFAM

## FOLHA DE APROVAÇÃO

"UM MODELO SOBRE AS DIFICULDADES PARA MODELAR CASOS  
DE USO"

ELIZAMARY DE SOUZA NASCIMENTO

Dissertação de Mestrado defendida e aprovada pela banca examinadora constituída pelos

Professores:

Profa. Tayana Uchôa Conte - PRESIDENTE

Prof. Eduardo Luzeiro Feitosa - MEMBRO INTERNO

Profa. Simone Diniz Junqueira Barbosa - MEMBRO EXTERNO

Manaus, 09 de Junho de 2017

**ELIZAMARY DE SOUZA NASCIMENTO**

**UM MODELO SOBRE AS DIFICULDADES PARA MODELAR CASOS DE USO**

Dissertação submetida ao corpo docente do Programa de Pós-Graduação em Informática da Universidade Federal do Amazonas (PPGI-UFAM) como requisito para obtenção do título de Mestre em Informática.

Banca Examinadora:

---

Prof<sup>ª</sup>. Tayana Uchôa Conte, D.Sc. (Orientadora)

---

Prof. Eduardo Luzeiro Feitosa, D.Sc.

---

Prof<sup>ª</sup>. Simone Diniz Junqueira Barbosa, D.Sc

*Até aqui nos ajudou o Senhor.*

## AGRADECIMENTOS

Chegar até aqui é o resultado de uma longa e difícil caminhada, no entanto, quando fazemos de Deus nosso companheiro de jornada, percebemos que no trajeto mais difícil, Ele nos coloca em Seus ombros e nos transporta para o outro lado e continua a caminhada, com uma diferença: de dia o Sol não nos molesta e a noite as trevas não nos impede de prosseguirmos, pois Ele é nossa luz. Quando vemos o presente como resultado de lutas já ultrapassadas, podemos vislumbrar ao futuro, com mais coragem, acreditando que nosso parceiro de tantas lutas e vitórias, continuará conosco, se assim o permitirmos. Obrigado Senhor, por acreditar nas possibilidades que Tu nos oferece.

A Prof<sup>a</sup>. DSc. Tayana Uchôa Conte, que neste trajeto, não foi apenas uma orientadora, foi mais que isto, foi e é uma grande amiga. Muito obrigada! Por seu espírito voluntarioso e generoso; disposta sempre a ajudar àqueles que acreditam nas possibilidades e fazem da vida seu maior desafio.

A todos os colegas e amigos do grupo USES-UFAM pelo apoio, colaboração, troca de experiências e compartilhamento de conhecimento. Em especial, ao meu companheiro de pesquisa Williamson Silva, um grande amigo e colaborador na minha pesquisa. Agradeço pelas discussões sobre nossos estudos, por nos ajudar em todos os momentos e nos divertimos pelos erros e acertos.

Aos colegas dos grupos ESE/COPPE-UFRJ e OPUS/PUC-Rio pelas contribuições nos estudos realizados. Em especial, agradeço ao Prof. Guilherme Horta Travassos pelo apoio e colaboração no estudo realizado na COPPE-URFJ.

Aos professores Igor Steinmacher e Bruno Gadelha por participarem da minha banca de qualificação, pelas contribuições e colaboração nos estudos realizados.

À todos os participantes dos estudos apresentados nesta dissertação, muito obrigada!

À Universidade Federal do Amazonas e o Instituto de Computação, pelo apoio durante todo o mestrado. Em especial ao Prof. Eduardo Luzeiro Feitosa, Helen Nascimento e Frank Azevedo pela ajuda em todos os momentos.

À CAPES pelo apoio financeiro ao longo do mestrado.

À todos os meus amigos, especialmente àqueles que de uma forma ou de outra nos apoiaram nesta etapa.

Ao Ministério Hiero, que sempre foi e é meu refúgio e minha força. Obrigada Senhor pelo dom que me deste. É um grande privilégio levar alegria e esperança através do louvor.

À família maravilhosa que tenho. Ao meu saudoso pai, Aristonildo Chagas Araújo Nascimento (in memorian), um exemplo de vida. A você paizinho, que sempre nos incentivou a buscar novos horizontes e a lutar pelos nossos sonhos. A você, a minha eterna gratidão, também dedico os méritos desta conquista. Você foi e é meu modelo de vida. Te amo.

À minha querida mãe, Elizabete de Souza Nascimento. A você querida, que sempre acreditou em mim e nos apoiou durante esta caminhada. A você que sempre tem um sorriso, um afeto para dar, que tem fé e acredita numa vida melhor, onde esperamos viver sem sofrimentos ou dor. Te amo querida.

Ao meu amado esposo, Nasson Nascimento dos Passos, que durante todo este tempo soube compreender esta minha luta. Com paciência soube esperar por este momento e juntos poderemos desfrutar o sabor desta vitória que é de todos nós. Soube compreender, principalmente nos momentos de necessidades pessoais e familiares, e haver guardado em silêncio a esperança de um dia termos o direito de vivermos com mais dignidade e usufruir os frutos desta conquista. Te amo.

Dedico esta dissertação ao meu querido esposo e aos meus queridos filhos, por quem encarei esta luta, a eles, Tiago Mendelsson Nascimento dos Passos e Nael Marshall Nascimento dos Passos.



## RESUMO

### UM MODELO SOBRE AS DIFICULDADES PARA MODELAR CASOS DE USO

Orientadora: Tayana Uchôa Conte, D.Sc.

Casos de Uso (Use Cases – UCs) tornaram-se um importante artefato para a especificação dos requisitos de software. No entanto, há várias dificuldades que impedem estudantes e engenheiros de software de especificarem UCs de forma correta. Com o objetivo de explorar e entender as dificuldades em modelar diagrama e especificar UCs, foram realizados quatro estudos experimentais com 195 participantes. Esta dissertação apresenta um modelo sobre as dificuldades na modelagem de UCs encontradas nos estudos realizados. O modelo apresenta as dificuldades relacionadas a modelagem do diagrama e especificação de UC. Estas dificuldades foram classificadas em categorias. Primeiramente, serão apresentadas as categorias relacionadas ao diagrama (D): (D-I) Identificar os atores do sistema, (D-II) Identificar os casos de uso do sistema, (D-III) Relacionamento de generalização e (D-IV) Relacionamentos extend e include. Por fim, as categorias relacionadas a especificação (E): (E-I) Abstração dos requisitos, (E-II) Fluxos do UC, (E-III) Regras de negócio, (E-IV) Referências de fluxos e Regras de negócio, (E-V) Pré e Pós-condição e (E-VI) Dependência de outros UCs. Esta pesquisa fundamentou-se em uma metodologia baseada em evidências, a análise dos dados foi conduzida utilizando procedimentos do método Grounded Theory (GT) e para a avaliação do modelo elaborado utilizou-se a Análise Fatorial Exploratória (AFE). O modelo foi avaliado e mostrou que as dificuldades apresentadas ocorrem com mais de 50% dos participantes. Utilizou-se a AFE para identificar possíveis fatores com dificuldades correlacionadas. A AFE mostrou a correlação entre certas dificuldades agrupadas nas categorias definidas para o modelo. Através dos resultados obtidos dos estudos experimentais, a versão atual do modelo pode ser considerada válida para representar a percepção dos participantes sobre as dificuldades ocorridas em diagrama e especificação de UC. Além disso, o modelo serve como base para futuras pesquisas na área, bem como no apoio à sugestão de práticas para melhorar o processo de ensino/aprendizagem dos alunos em modelagem de UC.

Palavras-chave: Casos de Uso, Dificuldades na modelagem de casos de uso, Estudo experimental, Modelo de dificuldades;

## **ABSTRACT**

### **A MODEL ON DIFFICULTIES IN MODELING USE CASES**

Advisors: Tayana Uchôa Conte, D.Sc.

Use Cases (UCs) have become an important artifact for the specification of software requirements. However, there are several difficulties that prevent students and software engineers from specifying UCs correctly. In order to explore and understand the difficulties in modeling diagram and specify UCs, we carried out four empirical studies with 195 participants. We present in this thesis a model about the difficulties in the modeling of UCs found in the studies carried out. The model presents the difficulties in diagram and specification UC that were classified into categories. First, we present the categories related to the diagram (D), which identify: (D-I) the actors of the system, (D-II) the use cases of the system, (D-III) generalization relationship, (D-IV) included and extend relationship; and categories related to the use-case specification (E): (E-I) abstraction of requirements, (E-II) flows of UC, (E-III) business rules, (E-IV) References of flows and business rules, (E-V) Pre- and Post-condition and (E-VI) dependence of other UCs. This on research is based on an evidence-based methodology, we executed procedures of the Grounded Theory (GT) method to analyze the data; and we use the Exploratory Factor Analysis (EFA) to evaluate the constructed model. The model was evaluated and it showed that the presented difficulties occur with more than 50% of the participants. We used EFA to identify possible factors with correlated difficulties. The EFA did not show that some of the difficulties grouped in the categories defined for the model are correlated. Through the obtained results from the empirical studies, the current version of the model can be considered valid to represent the perception of the participants about the difficulties occurred in diagram and specification of UC. In addition, the model serves as a basis for future research in the field, as well as support for the suggestion of practices to improve the teaching / learning process of students in modeling UC.

**Keywords:** Use Cases, Difficulties in use case modeling, empirical study, Model of difficulties;

## LISTA DE FIGURAS

<b>Figura 1.1:</b> Visão geral da metodologia adotada para a definição e validação do Modelo de Dificuldades em Modelar Casos de Uso. ....	4
Figura 2.1: Exemplo de um Diagrama de UC (Cockburn, 2005).....	9
<b>Figura 3.1:</b> Descrição do requisito do caso de uso. ....	17
<b>Figura 3.2:</b> Exemplo de um caso de uso especificado com UCMModel baseado em Massollar <i>et al.</i> (2012). ....	19
<b>Figura 3.3:</b> Artefatos utilizados no experimento. ....	23
<b>Figura 3.4:</b> Passos utilizados na execução do estudo. ....	25
<b>Figura 3.5:</b> Número de diferentes tipos de defeitos encontrados nos casos de uso gerados pelos formatos gráfico e textual.....	29
<b>Figura 3.6:</b> Número de diferentes tipos de defeitos encontrados nos UCs gerados pelos formatos.....	30
<b>Figura 3.7:</b> Número de diferentes tipos de defeitos encontrados nos UCs gerados pelos formatos.....	31
<b>Figura 3.8:</b> Gráfico de dispersão do tempo e corretude por Grupo. ....	32
<b>Figura 3.9:</b> Boxplots do tempo e corretude por Grupo. ....	33
<b>Figura 3.10:</b> Percepção dos Participantes na utilização do UCMModel.....	36
<b>Figura 3.11:</b> Percepção dos participantes sobre a Preferência dos formatos para especificar UC. ....	38
<b>Figura 4.1:</b> Modelo de Dificuldades sobre Especificação de Casos de Uso.....	56
<b>Figura 5.1:</b> Conceitos relacionados a casos de uso e trabalhos práticos realizados. ....	68
<b>Figura 5.2:</b> Design do segundo trabalho prático na disciplina. ....	69
<b>Figura 5.3:</b> Processo de Construção e Análise dos Artefatos (TP1 e TP2).....	70
<b>Figura 5.4:</b> Total de defeitos identificados pelo Especialista e Equipes.....	72
<b>Figura 5.5:</b> Quantidade de tipos de defeitos encontrados na Inspeção dos Diagramas de UCs. ....	73
<b>Figura 5.6:</b> Quantidade de tipos de defeitos encontrados na Inspeção das Especificações de UCs.....	74
<b>Figura 5.7:</b> Quantidade de tipos de defeitos encontrados na Inspeção Vertical.....	75
<b>Figura 5.8:</b> Dificuldades percebidas a partir das respostas para a especificação textual do UC. ....	76
<b>Figura 5.9:</b> Dificuldades percebidas a partir das respostas para a construção do Diagrama de UCs.....	76
<b>Figura 5.10:</b> Modelo de dificuldades na Modelagem de Casos de Uso.....	81
<b>Figura 6.1:</b> Modelo de Dificuldades em Modelagem de Casos de Uso.....	87
<b>Figura 6.2:</b> Total de dificuldades em Diagrama de UC respondida por todos os participantes ....	98
<b>Figura 6.3:</b> Ranking das questões de dificuldades em Diagrama de UC respondidas pelos participantes.....	99
<b>Figura 6.4:</b> Total de dificuldades em Especificação de UC respondidas pelos participantes. ....	100
<b>Figura 6.5:</b> Ranking das questões de dificuldades em Especificação de UC respondidas pelos participantes.....	101
Figura 6.6: Testes KMO e Bartlett para o Índice de Dificuldades em UCs. ....	103
Figura 6.7: Gráfico de autovalores e fatores relacionados a dificuldades em Diagrama de UC. ....	103

Figura 6.8: Gráfico de autovalores e fatores relacionados a dificuldades em Especificação de UC ..... 104

## LISTA DE TABELAS

<b>Tabela 2.1:</b> Exemplo dos elementos utilizados no diagrama de casos de uso baseado em Cockburn (2005).	10
<b>Tabela 2.2:</b> Exemplo baseado no caso de uso Matricular em Cursos (Cockburn, 2005).	12
<b>Tabela 3.1:</b> Exemplo da especificação do caso de uso “Concluir Compra” de um Sistema de Compra Web.	17
<b>Tabela 3.2:</b> Objetivo do Estudo usando o paradigma GQM (Basili e Rombach, 1988).	20
<b>Tabela 3.3:</b> Questões para avaliar o conhecimento sobre Modelagem e Especificação de Casos de Uso.	23
<b>Tabela 3.4:</b> Itens de Verificação dos Atributos de Modelos de Casos de Uso.	26
<b>Tabela 3.5:</b> Defeitos e Grau de Severidade.	27
<b>Tabela 3.6:</b> Resumo de defeitos e tempo gasto por participante.	31
<b>Tabela 3.7:</b> Preferências dos formatos para especificar UCs.	35
<b>Tabela 3.8:</b> Preferências dos formatos para especificar UCs.	37
<b>Tabela 3.9:</b> Características de Facilidade, Utilidade, Empregabilidade e Dificuldades do formato gráfico.	39
<b>Tabela 3.10:</b> Preferência dos formatos: facilidade, utilidade e usar numa aplicação.	41
<b>Tabela 4.1:</b> Roteiro Semiestruturado.	49
<b>Tabela 4.2:</b> Resumo dos Participantes do Estudo.	50
<b>Tabela 4.3:</b> Dificuldades relacionadas à Abstração do Requisito.	52
<b>Tabela 4.4:</b> Dificuldades relacionadas à Pré e Pós-Condição.	53
<b>Tabela 4.5:</b> Dificuldades relacionadas às Regras de Negócio.	53
<b>Tabela 4.6:</b> Dificuldades relacionadas aos Fluxos do Caso de Uso.	54
<b>Tabela 4.7:</b> Dificuldades relacionadas a Referenciar os Fluxos e RN nos passos do Caso de Uso.	55
<b>Tabela 4.8:</b> Benefícios em Utilizar UC no Desenvolvimento de Software.	56
<b>Tabela 5.1:</b> Classes de defeitos adaptado de Travassos <i>et al.</i> (1999).	65
<b>Tabela 5.2:</b> Questões Objetivas do Questionário de Dificuldades em UCs.	66
<b>Tabela 5.3:</b> Sub-questões do Questionário de Dificuldades em UCs.	67
<b>Tabela 5.4:</b> Defeitos encontrados pelas equipes nos Modelos de UCs.	71
<b>Tabela 5.5:</b> Dificuldades em Modelar o Diagrama e Especificar UCs.	78
<b>Tabela 6.1:</b> Resumo das Dificuldades relacionadas a modelagem do Diagrama de UC.	88
<b>Tabela 6.2:</b> Resumo das Dificuldades relacionadas a especificação de UC.	90
<b>Tabela 6.3:</b> Resumo dos Participantes do Estudo.	93
<b>Tabela 6.4:</b> Questionário do Modelo de Dificuldades (Ver APÊNDICE H)	95
<b>Tabela 6.5:</b> Resultado do Teste Estatísticas de confiabilidade de Cronbach para dificuldades em Diagrama de UC, retirado da ferramenta JASP.	102
<b>Tabela 6.6:</b> Resultado do Teste Estatísticas de confiabilidade de Cronbach para dificuldades em Especificação de UC, retirado da ferramenta JASP.	102
<b>Tabela 6.7:</b> Análise fatorial exploratória sobre as dificuldades no Diagrama de UC, retirado da ferramenta JASP.	104
<b>Tabela 6.8:</b> Análise fatorial exploratória sobre as dificuldades na Especificação de UC, retirado da ferramenta JASP.	106
<b>Tabela 6.9:</b> Citações sobre dificuldades durante a modelagem do diagrama de UC.	108
<b>Tabela 6.10:</b> Citações sobre dificuldades durante a especificação do UC.	109

## SUMÁRIO

<b>CAPÍTULO 1 – INTRODUÇÃO.....</b>	<b>1</b>
1.1 CONTEXTO.....	1
1.2 PROBLEMA.....	1
1.3 OBJETIVOS.....	2
1.4 METODOLOGIA.....	3
1.5 ORGANIZAÇÃO.....	6
<b>CAPÍTULO 2 - BACKGROUND E TRABALHOS RELACIONADOS.....</b>	<b>8</b>
2.1 CASOS DE USO.....	8
2.1.1 Diagrama de Casos de Uso.....	8
2.1.2 Especificação de caso de uso.....	10
2.1.3 Regras de Negócio.....	11
2.2 TRABALHOS RELACIONADOS.....	12
2.3 CONSIDERAÇÕES FINAIS.....	13
<b>CAPÍTULO 3 – ESTUDO EXPERIMENTAL SOBRE OS FORMATOS TEXTUAL E GRÁFICO PARA ESPECIFICAÇÃO DE CASOS DE USO .....</b>	<b>14</b>
3.1 INTRODUÇÃO.....	14
3.2 ESPECIFICANDO CASOS DE USO.....	16
3.3 ESTUDO EXPERIMENTAL.....	20
3.4 RESULTADOS OBTIDOS.....	28
3.5 AMEAÇAS À VALIDADE.....	43
3.6 CONSIDERAÇÕES FINAIS.....	45
<b>CAPÍTULO 4 – ESTUDO QUALITATIVO SOBRE AS DIFICULDADES EM ESPECIFICAÇÃO DE CASOS DE USO.....</b>	<b>48</b>
4.1 INTRODUÇÃO.....	48
4.2 MÉTODO DE PESQUISA.....	49
4.3 RESULTADOS DO ESTUDO.....	51
4.4 DISCUSSÃO DOS RESULTADOS.....	57
4.5 AMEAÇAS À VALIDADE.....	59
4.6 CONSIDERAÇÕES FINAIS.....	60
<b>CAPÍTULO 5 – ESTUDO EXPERIMENTAL SOBRE AS DIFICULDADES INICIAIS EM CASOS DE USO E O APOIO DA INSPEÇÃO PARA O APRENDIZADO NA MODELAGEM DE CASOS DE USO.....</b>	<b>62</b>
5.1 INTRODUÇÃO.....	62
5.2 ESTUDO EXPERIMENTAL.....	63
5.3 RESULTADOS OBTIDOS.....	70
5.4 DISCUSSÃO DOS RESULTADOS.....	80
5.5 AMEAÇAS A VALIDADE.....	82

5.6 CONSIDERAÇÕES FINAIS .....	82
<b>CAPÍTULO 6 – MODELO DE DIFICULDADES EM MODELAR CASOS DE USO E SUA VALIDAÇÃO .....</b>	<b>85</b>
6.1 INTRODUÇÃO .....	85
6.2 MODELO DE DIFICULDADES EM CASOS DE USO .....	86
6.3 ESTUDO EXPERIMENTAL PARA AVALIAÇÃO DO MODELO .....	92
6.4 RESULTADOS OBTIDOS .....	97
6.5 DISCUSSÃO DOS RESULTADOS .....	109
6.6 AMEAÇAS À VALIDADE .....	111
6.7 CONSIDERAÇÕES FINAIS .....	113
<b>CAPÍTULO 7 – CONSIDERAÇÕES FINAIS E PERSPECTIVAS FUTURAS.....</b>	<b>115</b>
7.1 CONCLUSÕES .....	115
7.2 CONTRIBUIÇÕES.....	116
7.3 PERSPECTIVAS FUTURAS.....	118
<b>REFERÊNCIAS .....</b>	<b>119</b>

# CAPÍTULO 1 – INTRODUÇÃO

*Este capítulo apresenta a introdução a esta dissertação de mestrado. Além de contextualizar esta pesquisa, são apresentados a motivação, objetivos e a metodologia seguida. O restante deste capítulo apresenta a estrutura desta dissertação de mestrado sobre o modelo de dificuldades para especificação de casos de uso.*

## 1.1 CONTEXTO

A UML (*Unified Modeling Language*) é uma linguagem visual utilizada para modelar a estrutura e comportamento de sistemas orientados a objetos (Booch *et al.*, 2004). A UML teve início em meados dos anos 90 e tornou-se uma linguagem padrão para a modelagem de sistemas de software (OMG, 2010). Os modelos propostos por essa linguagem são frequentemente utilizados em todas as fases de desenvolvimento de software, que vai desde a fase inicial do levantamento de requisitos à implantação do software (Chaudron *et al.*, 2012). Nesse sentido, a indústria de software vem adotando Casos de Uso (*Use Cases* - UC) para especificar os requisitos de sistemas (Dobing e Parsons, 2006; Anda *et al.* 2009). Jacobson *et al.* (1998) introduziram o conceito de UC como uma forma dos profissionais de software obterem uma melhor compreensão dos requisitos de um sistema. Casos de Uso consistem normalmente de duas partes: o diagrama de UC e a especificação de UC (Tiwari e Gupta, 2015). O diagrama fornece uma visão geral das funcionalidades do sistema (Bolloju, 2006). A especificação representa o comportamento das funcionalidades do sistema detalhando a interação entre os atores e o sistema (Cockburn, 2001).

Para a especificação de casos de uso a literatura oferece diferentes formatos que vão desde descrições informais com textos mais simples para modelos mais formais, tais como, pseudocódigo (Anda *et al.* 2009). Além disso, este modelo pode auxiliar em diferentes atividades do processo de desenvolvimento de software, como, planejamento e estimativa, análise, projeto, desenvolvimento e testes (Anda *et al.* 2001; Anda *et al.* 2009; El-Attar e Miller, J. 2009).

## 1.2 PROBLEMA

Apesar da ampla disseminação dos casos de uso existem algumas dificuldades como especificações de UCs confusas, descrições complexas, os membros do projeto não entendem



ou nem todos utilizam este modelo (Tiwari e Gupta, 2015). Bolloju e Sun (2012) identificaram alguns problemas nas descrições textuais que podem afetar o entendimento da funcionalidade como um todo, por exemplo, especificações muito longas, difíceis de obter uma visão geral, conjunto de muitos cenários que às vezes não podem ser representados de forma clara, devido existir cenários que podem ser chamados dentro de outros cenários.

Especificar UCs pode ser uma atividade demorada e propensa a falhas, pois as especificações geralmente são escritas em linguagem natural (Rago *et al.* 2013). Com isso, estudos têm mostrado que estudantes e profissionais em engenharia de software apresentam dificuldades durante o processo da especificação do UC, tais como: dificuldade em especificar os passos dos fluxos nos UCs, dificuldades em organizar as informações na especificação do UC, dentre outras (Anda *et al.* 2006; Bolloju, 2006, Nascimento *et al.* 2016). Estas dificuldades podem afetar a qualidade do software pois vários tipos de defeitos podem ser inseridos nas especificações de UCs e propagados para os outros artefatos, como informações ambíguas, incompletude de informações devido à ausência de passos nos fluxos e ausência de Regras de Negócio (RN) (Anda *et al.* Bolloju, 2006, Nascimento *et al.* 2016).

Neste contexto, o problema tratado nesta dissertação de mestrado está relacionado com a identificação de problemas na modelagem de casos de uso (diagrama e especificação) durante o processo de ensino/aprendizagem de UCs. Tendo como objetivo responder a seguinte questão de pesquisa “*Quais as dificuldades percebidas ao modelar casos de uso (diagrama e especificação) durante o desenvolvimento de software?*”, foram realizados estudos experimentais para entender a percepção dos estudantes e profissionais de software sobre as dificuldades relacionadas a este processo.

### **1.3 OBJETIVOS**

O objetivo principal desta pesquisa é caracterizar as principais dificuldades encontradas, por estudantes e profissionais de software, ao modelarem o diagrama e especificarem casos de uso. Para alcançar este objetivo geral, buscou-se decompô-lo nos seguintes objetivos específicos:

- Compreender e identificar as dificuldades em especificar casos de uso utilizando diferentes formatos (textual e gráfico) durante o processo de modelagem;
- Formular um modelo de dificuldades da modelagem de casos de uso baseado na percepção de estudantes e profissionais de software;

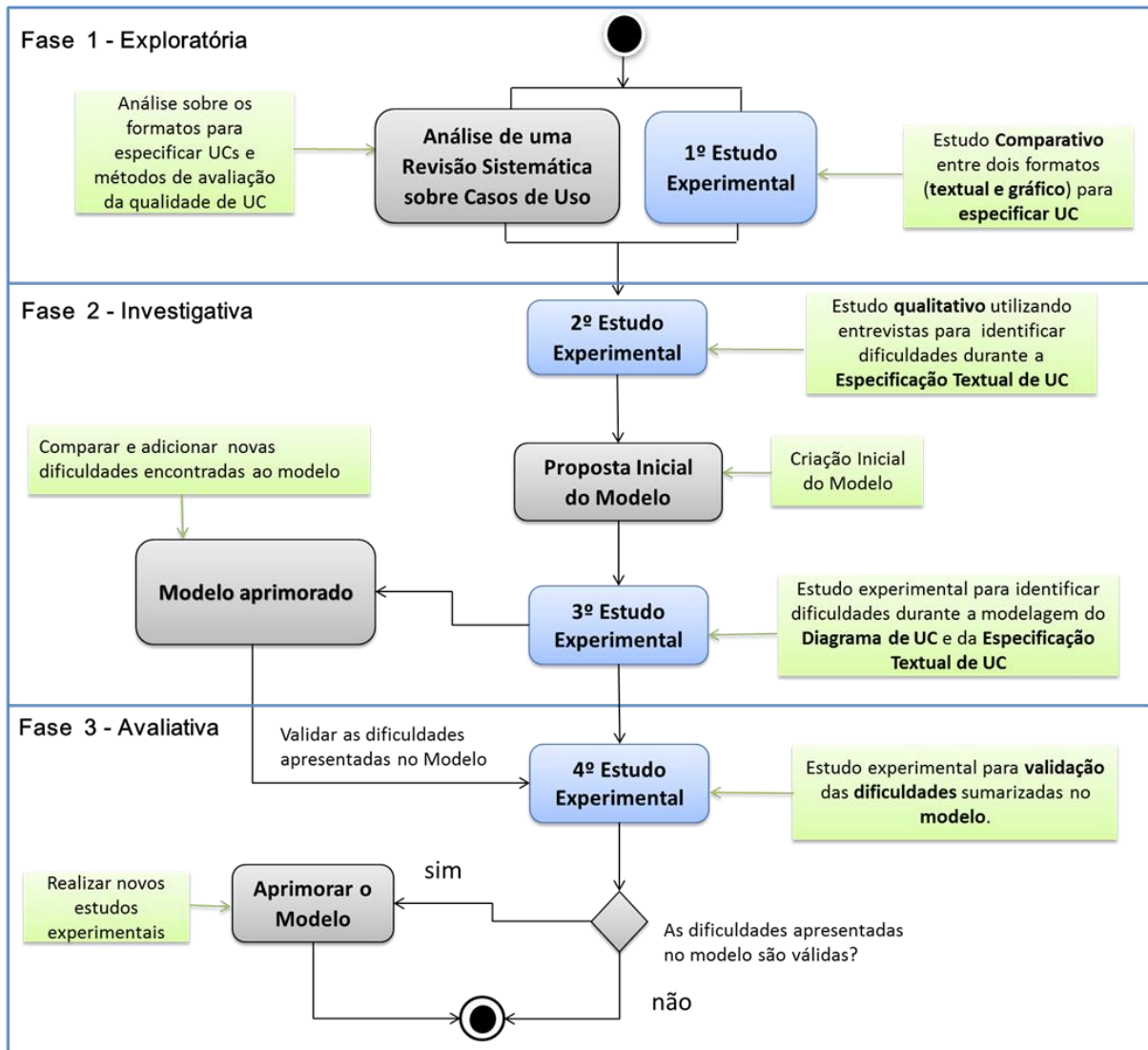
- Avaliar o modelo de dificuldades com avaliação metodológica na presente pesquisa.

O propósito final é que o modelo de dificuldades em modelar casos de uso (diagrama e especificação) sirva para: (a) auxiliar acadêmicos e profissionais de software que utilizam casos de uso na academia e/ou na prática industrial; (b) sugerir práticas para melhorar o processo de ensino/aprendizagem dos alunos na modelagem de casos de uso baseado na literatura; e (c) auxiliar os pesquisadores como base para futuras pesquisas na área, como na construção de técnicas voltadas para o ensino de casos de uso.

## 1.4 METODOLOGIA

Segundo Sjøberg *et al.* (2007) a pesquisa experimental explora, descreve, prevê, e explica os fenômenos naturais, sociais ou cognitivos, utilizando métodos científicos e experiência baseada em observação e evidências. Na pesquisa experimental pode-se incluir métodos qualitativos e quantitativos para a coleta e análise dos dados (Sjøberg *et al.* 2007). De acordo com Dittrich *et al.* (2007), a principal contribuição da pesquisa qualitativa é poder explorar o entendimento prático da engenharia de software. Dessa forma, obtém-se uma melhor compreensão dos métodos sendo utilizados na prática, proporcionando uma base para a sua melhoria (Dittrich *et al.* 2007).

A metodologia baseada em evidências foi dividida em três fases: na Fase 1 (Exploratória) buscar evidências na literatura sobre diferentes formatos para especificar casos de uso e de como avaliar a qualidade dessas especificações. Além disso, será realizado um estudo experimental para avaliar duas formas de especificar casos de uso. Através dos resultados alcançados neste estudo, será obtido um conhecimento sobre as dificuldades encontradas por estudantes com relação ao formato textual utilizado para especificar UC. Na Fase 2 (Investigativa) o objetivo é investigar com mais detalhes, através de estudos experimentais, as dificuldades percebidas na especificação de caso de uso, formular um modelo inicial sobre estas dificuldades e aprimorá-lo. Por fim, na Fase 3 (Avaliativa) o objetivo é avaliar o modelo de dificuldades em casos de uso proposto nesta pesquisa. A seguir são detalhadas as atividades que serão realizadas em cada fase da metodologia, representada na Figura 1.1.



**Figura 1.1:** Visão geral da metodologia adotada para a definição e validação do Modelo de Dificuldades em Modelar Casos de Uso.

**Fase 1 (Exploratória):** buscar na literatura evidências sobre casos de uso, sua utilização e aplicação. Nesse sentido, os objetivos desta primeira fase são:

- Buscar evidências na literatura: durante essa fase encontrou-se uma “Revisão Sistemática sobre Especificação de Casos de Uso” realizada por Tiwari e Gupta (2015). O objetivo dos autores nesse estudo foi buscar na literatura a evolução dos casos de uso, sua aplicação, avaliação da qualidade das especificações de UC e quais as questões de pesquisa em aberto e as futuras direções para as pesquisas em especificação de casos de uso (Tiwari e Gupta, 2015). Assim, com a análise dos resultados da revisão sistemática de Tiwari e Gupta (2015), pretende-se identificar diferentes formatos utilizados para especificar UCs. Além disso, analisar diferentes propostas para avaliação da qualidade das especificações de UCs.

- 1º estudo experimental: em paralelo à análise da revisão sistemática, conduzir um estudo experimental com o objetivo de avaliar a percepção de estudantes em relação a dois formatos utilizados para especificar UCs (ver Figura 1.1). A partir deste estudo, coletar dados sobre como os participantes especificaram UC utilizando dois formatos diferentes e identificar as principais dificuldades, e os pontos positivos e negativos de cada formato. Além disso, para a análise qualitativa dos dados será empregado alguns procedimentos de Grounded Theory (GT) (Strauss e Corbin, 1998).

Assim, a partir dos resultados qualitativos do 1º estudo experimental, identificar os principais problemas relatados pelos participantes durante a especificação de casos de uso construídos pelos formatos utilizados no estudo. Estes resultados servirão de base para realizar a segunda fase da pesquisa, investigando mais detalhadamente qual a percepção de estudantes e profissionais de software sobre a especificação textual de casos de uso.

**Fase 2 (Investigativa):** realizar novos estudos experimentais com estudantes e profissionais de software a fim de coletar dados para formular e propor o modelo de dificuldades, como também aprimorá-lo. Assim, os resultados do 2º estudo experimental servirão de base para formular o modelo de dificuldades e os resultados do 3º estudo experimental servirão para comparar e identificar novas dificuldades para melhorar e aprimorar o modelo. Os objetivos destes estudos experimentais são detalhados abaixo:

- 2º estudo experimental: identificar detalhadamente quais as principais dificuldades encontradas por estudantes, durante a especificação de UCs. Com os resultados deste estudo gerar a versão inicial do modelo de dificuldades em especificação de UCs. Contudo, esta versão inicial do modelo limita-se à especificação textual de UC pois tem como base as dificuldades relacionadas a este formato.
- 3º estudo experimental: identificar as dificuldades durante o processo de modelagem do diagrama de caso de uso e da especificação textual de UCs por estudantes. Além disso, o uso de uma inspeção em UCs será avaliada como forma de apoio/melhoria no aprendizado dos alunos. Os resultados deste estudo servirão de base para aprimorar e gerar uma nova versão do modelo, incluindo as dificuldades identificadas na modelagem do diagrama de UC e em especificação de UCs.

**Fase 3 (Avaliativa):** realizar um novo estudo experimental (4º) para avaliar as dificuldades evoluídas e consolidadas no modelo. As categorias do modelo são baseadas nos resultados dos estudos experimentais realizados anteriormente com estudantes (na fase 2). No

entanto, as dificuldades do modelo serão avaliadas por estudantes de outras instituições diferentes das instituições dos estudos realizados na fase 2. O objetivo deste estudo é identificar o grau de frequência das dificuldades que ocorreram com alunos após o uso de UC, baseado no modelo. Desta forma, será possível identificar se as dificuldades apresentadas no modelo são válidas e ocorrem com outros estudantes de diferentes instituições de outras regiões do país.

Os resultados dos estudos experimentais realizados nesta pesquisa foram publicados no Simpósio Brasileiro de Engenharia de Software (SBES) e no Congresso Ibero-Americano de Engenharia de Software (CIbSE) com os seguintes artigos aceitos:

- *“Is a Picture worth a Thousand Words? A Comparative Analysis of Using Textual and Graphical Approaches to Specify Use Cases”* (Nascimento *et al.* 2016) publicado no 30th Simpósio Brasileiro de Engenharia de Software (SBES 2016). Nesse artigo, foi descrito o 1º estudo experimental comparativo entre dois formatos (Gráfico e Textual) para especificação textual de UC com o objetivo de verificar a corretude e o tempo na geração destas especificações, apresentado no Capítulo 3;
- *“Um Modelo sobre as Dificuldades para Especificar Casos de Uso”* (Nascimento *et al.* 2017) publicado no XX Congresso Ibero-Americano em Engenharia de Software (CIbSE 2017). Nesse artigo, foi descrito o 2º estudo qualitativo em que foi proposto o modelo de dificuldades para especificação de UC, apresentado no Capítulo 4.

## 1.5 ORGANIZAÇÃO

Esta dissertação está organizada em outros sete capítulos, além deste primeiro capítulo que apresentou a motivação e o contexto no qual está inserida esta pesquisa. A organização do texto deste trabalho segue a estrutura abaixo:

**Capítulo 2 – Referencial Teórico:** apresenta uma base teórica dos principais conceitos sobre Casos de Uso, diagramas e especificação de UCs e regras de negócio. Além disso, apresenta alguns trabalhos relacionados sobre dificuldades em modelar o diagrama e especificar casos de uso.

**Capítulo 3 – Estudo Preliminar sobre as diferentes notações para especificação de casos de uso:** apresenta um estudo experimental realizado com dois formatos (textual e gráfico) diferentes para especificação de casos de uso. Este estudo teve como objetivo verificar a percepção dos participantes em relação aos formatos para especificar casos de uso com relação

à facilidade, utilidade e preferência. Além disso, avaliou a corretude e o tempo gasto na geração das especificações de caso de uso.

**Capítulo 4 – Estudo Qualitativo sobre as dificuldades em especificação de casos de uso:** neste capítulo é apresentado um estudo qualitativo realizado com estudantes e profissionais de software de duas universidades localizadas em diferentes regiões do país. Neste estudo, utilizaram-se entrevistas semiestruturadas visando identificar estas dificuldades. Neste capítulo, é apresentada a versão inicial do modelo de dificuldades que sumariza as dificuldades percebidas pelos participantes relacionadas à especificação de UC utilizando um formato textual.

**Capítulo 5 – Estudo experimental sobre as dificuldades iniciais em casos de uso e o apoio da inspeção para o aprendizado na modelagem de casos de uso:** este capítulo apresenta um estudo experimental que buscou analisar a percepção dos alunos sobre os modelos de casos de uso (diagrama e especificação) e as dificuldades percebidas. Além disso, analisou-se o uso da inspeção em casos de uso (diagrama e especificação) com o objetivo de entender a percepção dos alunos sobre possíveis melhorias no diagrama e especificação de casos de uso.

**Capítulo 6 – Modelo de dificuldades em modelar casos de uso e sua validação:** este capítulo apresenta o modelo de dificuldades em modelar casos de uso (diagrama e especificação) e sua evolução baseado nos resultados dos estudos experimentais realizados nesta pesquisa. Além disso, neste capítulo é apresentado um estudo experimental realizado com a finalidade de validar as dificuldades apresentadas no modelo, por alunos de graduação de três universidades, localizadas em diferentes regiões do país.

**Capítulo 7 – Considerações finais e trabalhos futuros:** este capítulo contém as conclusões e contribuições deste trabalho, além de indicar a continuação desta pesquisa através de trabalhos futuros.

## **CAPÍTULO 2 - BACKGROUND E TRABALHOS RELACIONADOS**

*Este capítulo apresenta os principais conceitos sobre casos de uso, sobre os diagramas e especificação de UCs seguindo a Fase I (Exploratória) da metodologia. Além disso, são apresentados alguns trabalhos relacionados a dificuldades em modelagem de diagrama e especificação de casos de uso.*

### **2.1 CASOS DE USO**

Os casos de uso foram propostos por Ivan Jacobson como parte do processo de desenvolvimento de software orientado a objetos (Jacobson, 1992). Como apresentando anteriormente os casos de uso consistem em duas partes: o diagrama e a especificação (Tiwari e Gupta, 2015). Os principais objetivos do diagrama de UC são: (a) auxiliar na comunicação entre os membros da equipe de desenvolvimento (Jacobson, 1992; Anda *et al.*, 2009); e (b) descrever um cenário mostrando as funcionalidades do sistema do ponto de vista do usuário (Jacobson, 1992; Heumann, 2015). E o principal objetivo da especificação de um caso de uso é representar “*a especificação de um conjunto de ações realizadas por um sistema que produz um resultado observável que é, normalmente, de valor para um ou mais atores ou outros stakeholders do sistema*” (OMG, 2010). Um stakeholder é alguém ou algo que tem interesse no comportamento do caso de uso (Cockburn, 2005).

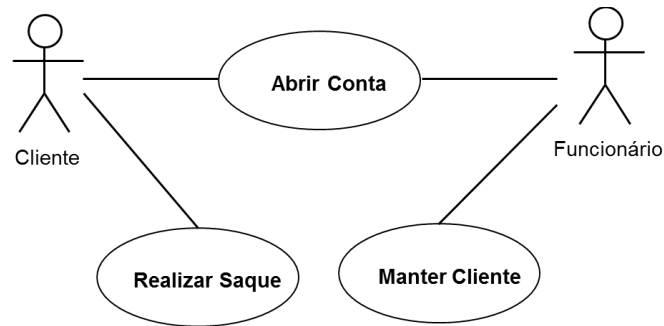
Neste sentido, a especificação de casos de uso descreve uma tarefa como uma ação que pode ser alcançada com uma sequência de interações do usuário com o sistema (Cockburn, 2005). Logo, o sistema deve responder a várias ações que um ou mais atores e ou *stakeholders* realizam em uma funcionalidade específica (Kettenis, 2007).

#### **2.1.1 Diagrama de Casos de Uso**

Os diagramas representam a estrutura dos casos de uso, os atores e as relações entre eles (Cockburn, 2005). Além disso, um diagrama de caso de uso é composto por quatro elementos diferentes que representam o funcionamento de um sistema: o sistema em si, os atores que interagem com o sistema, os serviços (ou casos de uso) necessários para executar o sistema, e as relações entre estes elementos (Misbhauddin e Alshayeb, 2013).

Um ator é a representação do papel desempenhado por algum tipo de usuário em um sistema (Cockburn, 2005). Os atores também podem simbolizar algum hardware especial ou outro software que tenha interação com o sistema (Dutoit e Paech, 2002). O caso de uso

representa as ações que devem suceder quando um ator interage com o sistema e que permite ao ator atingir o seu objetivo (Cockburn, 2005). Desta forma, os casos de uso são possíveis respostas de um determinado sistema de ações (Kettenis, 2007). A notação usada para representar os atores, os casos de uso e as relações entre estes elementos são exemplificados na Figura 2.1.



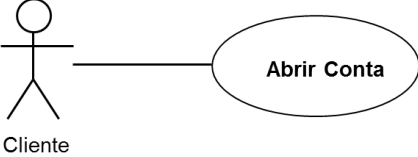
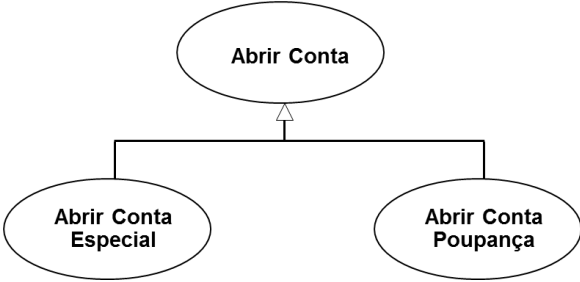
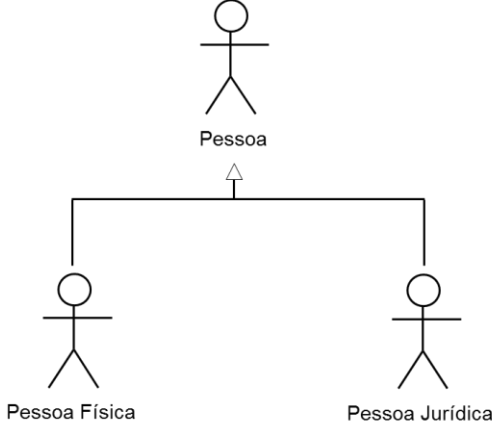
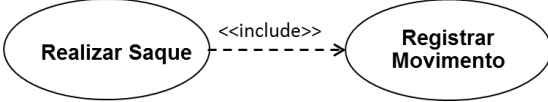
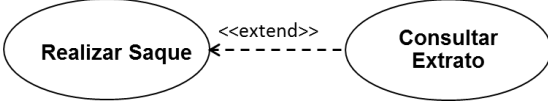
**Figura 2.1:** Exemplo de um Diagrama de UC (Cockburn, 2005).

As relações entre os elementos que constituem o diagrama de casos de uso são: associação, generalização, inclusão e extensão, como exemplificado na Tabela 2.1 (Cockburn, 2005). A associação é uma relação entre um ator e um caso de uso representando a interação do ator com o caso de uso, o seu objetivo é definir a maneira como os elementos estão unidos e se relacionam, compartilhando informações (Kettenis, 2007). A generalização é representada por uma linha com uma seta grossa, que aponta para o elemento geral e na outra extremidade o elemento especializado, pode ser representado entre casos de uso e entre atores. Além disso, existem os relacionamentos de inclusão (<<include>>) e extensão (<<extend>>) (Cockburn, 2005).

O relacionamento de inclusão acontece quando um caso de uso já identificou sua funcionalidade, ou seja, é inserido outro caso de uso como forma de complementação obrigatória de uma atividade (Cockburn, 2005). Por exemplo, um caso de uso “A” para um caso de uso “B” indica que “B” é essencial para o comportamento de “A”. Isto implica que ao executar o caso de uso “A” executa-se também o caso de uso “B”. O relacionamento de extensão permite adicionar um novo comportamento ao caso de uso existente de uma forma não obrigatória, ou seja, a inserção de atividades no fluxo de um caso de uso sempre acontece de forma condicional (Cockburn, 2005).



**Tabela 2.1:** Exemplo dos elementos utilizados no diagrama de casos de uso baseado em Cockburn (2005).

Elementos e Relações	Representação
Associação	
Generalização entre casos de uso	
Generalização entre atores	
Relacionamento de inclusão (include)	
Relacionamento de extensão (extend)	

### 2.1.2 Especificação de caso de uso

A parte mais importante da especificação de um caso de uso é seus fluxos de eventos, que descrevem uma sequência de eventos de um ator para completar um processo. São através dos fluxos de eventos que a sequência de ações entre o ator e o sistema é especificado utilizando a descrição do fluxo principal, fluxos alternativos e os fluxos de exceções (Kruchten, 2003).

O fluxo principal é executado quando o usuário aciona o UC (Misbhauddin e Alshayeb, 2013) e é finalizado quando o objetivo do usuário é alcançado (no fluxo principal ou alternativo) ou quando o objetivo é abortado pelo usuário (Yue *et al.* 2009). Os fluxos alternativos descrevem outros cenários e são iniciados a partir do fluxo principal (Yue *et al.* 2009). Além destes fluxos, existe o fluxo de exceção. Este tipo de fluxo ocorre quando algo inesperado acontece na interação entre o usuário e o sistema (Cockburn, 2005; Misbhauddin e Alshayeb, 2013). Para que os fluxos alternativos e de exceções sejam executados, as condições devem ser indicadas nos passos do fluxo principal ou alternativo do UC (Yue *et al.*, 2009; Misbhauddin e Alshayeb, 2013). Segundo Larman (2012), um caso de uso deve conter apenas um fluxo principal e pode conter vários fluxos alternativos.

Existem diferentes formatos que podem ser usados para descrever casos de uso, como o uso de *templates* (Anda *et al.*, 2009) e o primeiro *template* foi proposto por Jacobson (1992). Nos *templates* são definidos vários atributos do caso de uso: nome do caso de uso, pré-condições, pós-condições, fluxo principal, fluxos alternativos e campos de fluxos alternativos para especificar as necessidades dos requisitos (Ver APÊNDICE A - Resumo dos atributos mais utilizados em templates para especificação de UC).

Esse tipo de formato textual (uso de *template*) tem sido amplamente aceito para especificar os requisitos funcionais do software (Tiwari e Gupta, 2015) por: (a) ser um formato menos formal e simples, (b) possuir uma estrutura pré-definida que pode ser utilizada de acordo com o contexto do software, (Jacobson, 1992; Bolloju e Sun, 2012), e (c) proporcionar flexibilidade aos engenheiros de software para especificar os requisitos (Tiwari e Gupta, 2015). Um exemplo de um *template* utilizado para especificar os requisitos de um caso de uso é apresentado na Tabela 2.2. Na primeira coluna são apresentados os atributos utilizados e na segunda coluna são as informações descritas para cada atributo do caso de uso.

### **2.1.3 Regras de Negócio**

As regras de negócio representam declarações que descrevem procedimentos ou restrições relacionadas à organização ou ao contexto onde a organização atua (Halpin, 2006). As regras de negócio afetam diretamente o comportamento do sistema, ou seja, os Casos de Uso contêm restrições que devem ser seguidas durante sua execução que devem ser identificadas e detalhadas na especificação do UC (Linehan, 2008).

**Tabela 2.2:** Exemplo baseado no caso de uso Matricular em Cursos (Cockburn, 2005).

<b>Nome do Caso de Uso</b>	Matricular em Curso
<b>Breve Descrição</b>	Esse caso de uso permite um estudante matricular-se em cursos oferecidos no semestre corrente.
<b>Atores</b>	Estudante
<b>Pré-condição</b>	Entrar no sistema
<b>Fluxo Principal</b>	<ol style="list-style-type: none"> <li>1. O estudante seleciona “criar horário”.</li> <li>2. O sistema exibe um formulário de horário vazio.</li> <li>3. O sistema retorna uma lista de vagas em cursos disponíveis do sistema de Catálogo de Cursos (FA01- Modificar um Horário)</li> <li>4. O estudante seleciona as vagas disponíveis em cursos básicos e as vagas disponíveis nos cursos alternativos da lista e seleciona a opção “Enviar” (FE01 – Curso não selecionado, RN01).</li> <li>5. O sistema salva o horário.</li> </ol>
<b>Fluxos Alternativos</b>	FA01 – Modificar um Horário <ol style="list-style-type: none"> <li>1. O estudante seleciona “modificar horário”.</li> <li>2. O sistema retorna para o passo 2 do fluxo principal e exibe o horário atual do Estudante.</li> <li>3. O sistema retorna uma lista com todas as vagas em cursos disponíveis para o semestre corrente do Sistema de Catálogo de Curso e exibe essa lista para o Estudante.</li> <li>4. O estudante modifica as seleções dos cursos apagando e adicionando novos cursos e seleciona a opção “Enviar”.</li> <li>5. O sistema salva o horário.</li> </ol>
<b>Fluxos de Exceção</b>	FE01–Curso não selecionado <ol style="list-style-type: none"> <li>1. O sistema verifica que o curso básico e curso alternativo não foram selecionados pelo estudante.</li> <li>2. O sistema notifica o estudante que deve ser selecionado um curso básico e curso alternativo para a opção de enviar.</li> </ol>
<b>Pós-condição</b>	Salvar o horário definido pelo estudante para efetivar a matrícula no curso
<b>Pontos de Extensão</b>	Não há pontos de extensão associados a esse caso de uso
<b>Regra de Negócio</b>	RN01 – Vagas Disponíveis O sistema exibe uma lista com as vagas disponíveis para os cursos: básico e alternativo. São 30 vagas para a o curso básico e 20 vagas para o curso alternativo. Para cada vaga selecionada e confirmada por um aluno o sistema vai contabilizando o número de vagas disponíveis.

## 2.2 TRABALHOS RELACIONADOS

Os estudos apresentados a seguir foram realizados em contextos e com objetivos diferentes. Contudo, apresentam resultados que mostram que os participantes de cada estudo tiveram dificuldades ao utilizar os artefatos da UML.

Anda *et al.* (2006) realizaram um estudo de caso visando identificar as dificuldades percebidas por profissionais de software (desenvolvedores, gerentes de projetos, dentre outros) que estavam atuando em projetos de software que utilizavam UML. Sobre as especificações de UCs, os resultados mostraram que os profissionais tiveram dificuldades tanto para definir o nível de detalhes quanto para organizar as informações na especificação do UC.

Bolloju (2006) realizou um estudo experimental que identificou as dificuldades que analistas novatos em desenvolvimento de software possuem ao utilizar artefatos da UML em projetos de software. Com relação às especificações dos UCs, os resultados mostraram que as

dificuldades mais frequentes na especificação de UC estavam relacionadas à descrição de passos dos fluxos nos UCs, ou seja, houve dificuldade para especificar os passos nos fluxos, dificuldade em especificar as informações de forma não ambígua e dificuldade em referenciar corretamente os fluxos alternativos nos passos do UC.

Siau e Loo (2006) realizaram um estudo visando identificar as dificuldades dos estudantes no processo de aprendizagem dos diagramas da UML. Os participantes não possuíam conhecimento prévio nos diagramas da UML. Eles foram treinados e realizaram trabalhos práticos utilizando os conceitos aprendidos sobre os diagramas da UML. O estudo não focou na especificação do UC e sim nos diagramas da UML. Com relação ao diagrama de UC, os resultados mostraram que as principais dificuldades dos participantes foram em entender o objetivo do diagrama de UC, por exemplo, a relação dos atores do diagrama de casos de uso com outros diagramas da UML. Além disso, os participantes tiveram dificuldades na notação do diagrama de UC, por exemplo, algumas setas são confusas, ao usar o *extend* e *include*.

Dos estudos apresentados acima, três estudos (Anda *et al.* 2006, Bolloju, 2006 e Siau e Loo, 2006) abordam as dificuldades do uso de artefatos da UML utilizados no desenvolvimento de software, apresentando apenas algumas das dificuldades em especificar e modelar diagrama de UCs. Além disso, os estudos apresentados não propõem nenhum modelo de dificuldades, apenas apresentam a percepção das dificuldades encontradas pelos participantes ao utilizar os artefatos da UML (diagrama e especificação de UC). Com o intuito de identificar quais as dificuldades encontradas por estudantes e profissionais de software durante o processo de escrita da especificação do UC e na modelagem do diagrama de UC, os próximos capítulos apresentam os resultados dos estudos experimentais realizados nesta pesquisa e a proposta de um modelo de dificuldades em modelar casos de uso.

## **2.3 CONSIDERAÇÕES FINAIS**

Este capítulo teve como objetivo apresentar conceitos sobre casos de uso, o diagrama e a especificação de casos de uso e as regras de negócio. Além disso, apresentou os trabalhos relacionados com esta pesquisa sobre as dificuldades encontradas durante o uso de casos de uso no processo de desenvolvimento de software. O próximo capítulo apresenta o primeiro estudo experimental que comparou dois formatos (textual e gráfico) utilizados para especificar casos de uso (Nascimento *et al.* 2016). Além disso, são apresentadas as dificuldades percebidas pelos participantes ao utilizarem estes formatos durante a especificação do UC.

## **CAPÍTULO 3 – ESTUDO EXPERIMENTAL SOBRE OS FORMATOS TEXTUAL E GRÁFICO PARA ESPECIFICAÇÃO DE CASOS DE USO**

*Este capítulo apresenta o 1º estudo experimental realizado na Fase I (Exploratória). Será detalhado o planejamento, execução e resultados do 1º estudo experimental comparativo realizado entre dois formatos (Textual e Gráfico) para especificação de casos de uso. Este estudo teve como objetivo analisar qual dos formatos apresenta especificações com maior grau de corretude e em menos tempo.*

### **3.1 INTRODUÇÃO**

Existem diferentes formatos e estilos que podem ser utilizados para especificar casos de uso (Tiwari e Gupta, 2015). A escolha por um desses formatos dependerá das necessidades do projeto de software e das características de sua equipe, por exemplo, o nível de experiência dos membros da equipe de projeto, o tamanho do projeto, a organização e a coesão (Anda *et al.*, 2009). Estes formatos podem ser caracterizados de duas formas de maneira genérica: textual e gráfico.

Nos formatos textuais, os casos de uso são normalmente apresentados como um conjunto de cenários, descritos em linguagem natural (narrativa) e obedecendo a uma estrutura sintática (Gutiérrez *et al.* 2008; Anda *et al.* 2009). Utilizando esse tipo de formato, os *stakeholders* familiarizados com a notação não necessitam de treinamento formal e os desenvolvedores não precisam de ferramentas de modelagem específica (Achour *et al.* 1999). No entanto, o comportamento do caso de uso não é apresentado de forma direta e intuitiva (Gutiérrez *et al.* 2008). Além disso, as especificações geradas podem apresentar diferentes defeitos, tais como: descrições ambíguas, que possuem diferentes interpretações (Anda e Sjøberg, 2002); e informações incompletas, nas quais faltam descrições de fluxos ou regras necessárias para o entendimento completo do caso de uso (Tiwari e Gupta, 2015).

Os formatos gráficos utilizam notações gráficas que auxiliam a representar os cenários do caso de uso, por exemplo, por meio de diagramas (Gutiérrez *et al.* 2008). Esse tipo de formato pode ser útil para mostrar de forma mais direta o comportamento de um caso de uso, ou seja, os aspectos dinâmicos (fluxos alternativos, opções do sistema, dentre outros) (Phalp *et al.* 2007; Gutiérrez *et al.* 2008). Contudo, os engenheiros podem ter dificuldade em representar corretamente os requisitos do software no diagrama (Mello *et al.* 2011). Um problema potencial

é a omissão de fluxos/regras de negócios que devem ser descritos ou referenciados no diagrama (El-Attar e Miller, 2009; Mello *et al.* 2011). Com isso, as especificações de casos de uso geradas a partir da representação gráfica podem se tornar confusas, com descrições incompletas/incorrectas e de difícil entendimento (Cockburn, 2001; Phalp *et al.* 2007; Tiwari e Gupta, 2015).

Embora ambos os formatos possuam elementos em comum, o formato textual é considerado adequado para detalhar, validar e testar os requisitos do software (Phalp *et al.* 2007; Tiwari e Gupta, 2015) e o formato gráfico (diagramas) é adequado para visualizar o comportamento global do caso de uso e detalhar os aspectos dinâmicos relacionados à funcionalidade do software (Gutiérrez *et al.* 2008).

No entanto, não há consenso sobre qual formato é mais eficiente em diferentes contextos, assim como não há uma definição sobre o grau de formalismo mais adequado para esses formatos (Tiwari e Gupta, 2015). Se os formatos forem utilizados corretamente e os casos de uso forem bem especificados, eles especificam os requisitos do software de forma completa, correta e de fácil compreensão pelos engenheiros de software e *stakeholders* (Anda *et al.* 2009). Especificar casos de uso de forma clara, correta e coerente é importante para se obter um produto de software com qualidade (Wiegers, 1999). Logo, é importante investigar e comparar a facilidade (de aprendizado e uso) e a utilidade dessas duas formas de especificar casos de uso.

Com o objetivo de compreender a influência destas diferentes representações de casos de uso, este capítulo descreve a condução de um estudo experimental realizado com o objetivo de avaliar dois diferentes formatos para especificação de casos de uso (um textual e um gráfico). Para o formato textual, escolheu-se o formato proposto por Cockburn (2001 (com algumas adaptações, exclusão do campo trigger e palavras chaves), pois esse é um dos formatos mais citados na literatura (Tiwari e Gupta, 2015). Com relação ao formato gráfico, foram selecionados formatos que se baseiam em diagramas de atividades da UML (OMG, 2010), por ser um dos diagramas mais utilizados para representar o comportamento dos UCs (Gutiérrez *et al.* 2008, Yue *et al.* 2010).

Em seguida, esses formatos foram avaliados, sendo o baseado no UCMModel selecionado por oferecer uma estrutura sintática e semântica do UC e apoiar procedimentos de testes funcionais (Massollar *et al.* 2012). Além disso, os dois formatos (textual e gráfico) utilizam os elementos propostos por Jacobson *et al.* (1992) para descrever casos de uso. Para observar a influência de um ou outro formato, verificou-se qual o grau de corretude das especificações geradas a partir dos casos de uso e o tempo gasto por cada participante ao usar cada formato. A

análise da corretude das especificações foi realizada utilizando um método chamado Avaliação de Atributos de Verificação dos Modelos de UC (Anda *et al.* 2009).

Além disso, foi realizada uma análise qualitativa das percepções dos participantes em relação à facilidade (de aprendizado e uso), a utilidade e a preferência em relação aos formatos utilizados para especificar um caso de uso. Com base nos resultados, identificou-se qual formato é mais fácil de aprender e de ser usado para construir uma especificação de caso de uso. Adicionalmente, foram identificadas as dificuldades, os pontos positivos e negativos observados pelos participantes.

Este capítulo está organizado da seguinte forma: a Seção 3.2 aborda sobre os formatos textual e gráfico utilizados para especificar casos de uso. A Seção 3.3 descreve o planejamento, a condução e execução do estudo experimental. A Seção 3.4 apresenta os resultados quantitativos e qualitativos. A Seção 3.5 apresenta as principais ameaças à validade do estudo. Por fim, na Seção 3.6 apresenta as considerações finais do capítulo.

## **3.2 ESPECIFICANDO CASOS DE USO**

Nesta seção serão apresentados os dois formatos utilizados para especificação de casos de uso neste estudo.

### **3.2.1 Especificando de Forma Textual**

O primeiro formato textual para especificar um caso de uso foi proposto por Jacobson *et al.* (1992). Neste formato são incluídos vários elementos do caso de uso, tais como, nome do caso de uso, pré-condições, pós-condições, fluxo principal, fluxos alternativos e campos de fluxos alternativos para especificar informações do caso de uso. Este formato é focado principalmente na descrição do fluxo básico e nos fluxos alternativos de eventos. Para a realização do estudo, não foi definido um modelo específico para a descrição textual do caso de uso. Cada participante escolheu o formato que utilizaria para especificar o caso de uso de acordo com sua preferência ou experiência, contanto que o formato escolhido obrigatoriamente apresentasse os seguintes elementos: Fluxo Principal, Fluxos Alternativos e de Exceção (caso houvesse) e Regras de Negócio. Opcionalmente, poderiam ser descritos no caso de uso seu identificador, nome, atores, objetivo, as pré e pós-condições. Deste modo, assim como é sugerido por Anda *et al.* (2009), a escolha da estrutura deve ser de acordo com a necessidade do projeto de software, sendo que esta escolha deve ser motivada pelas características da equipe de desenvolvimento, por exemplo, o tamanho, organização, experiência e coesão em relação ao modelo que será escolhido.

Para exemplificar a descrição textual de um caso de uso utilizando um formato textual, a Tabela 3.1 mostra a especificação do caso de uso “Concluir Compra” de um sistema de compras via Web. Este formato e os elementos utilizados para especificar este caso de uso são semelhantes aos de Cockburn (2005), que também apresenta os mesmos elementos identificados por Tiwari e Gupta (2015). Os cenários e regras que este caso de uso deve atender estão descritos no seguinte requisito:

Para realizar a compra de um produto via Web é necessário que o cliente adicione seus produtos no carrinho de compras. O cliente poderá a qualquer momento incluir novos produtos ou excluir os produtos do carrinho de compras. Para o cliente concluir a compra na loja virtual, o sistema deve disponibilizar uma opção para que o cliente siga os passos necessários para concluir a compra e efetuar o pagamento. No final o cliente deve ser informado de que o pagamento foi efetuado e validado mostrando o número do pedido.

**Figura 3.1:** Descrição do requisito do caso de uso.

Conforme apresentado na Tabela 3.1, a primeira coluna apresenta os elementos utilizados para descrever o caso de uso e a segunda coluna apresenta as informações que o caso de uso deverá atender.

**Tabela 3.1:** Exemplo da especificação do caso de uso “Concluir Compra” de um Sistema de Compra Web.

<b>Nome do caso de uso:</b>	Concluir Compra
<b>Descrição:</b>	Permitir ao cliente fechar a compra, realizar o pagamento e obter o número do pedido.
<b>Atore(s):</b>	Cliente
<b>Pré-condições:</b>	Cliente deve estar autenticado
<b>Pós-condições:</b>	Cliente realiza o pagamento e obtém o número do pedido
<b>Fluxo Principal:</b>	<ol style="list-style-type: none"> <li>1. O sistema lista os produtos que constam no carrinho de compras e disponibiliza as opções “Efetuar pagamento” e “Excluir item” [A1];</li> <li>2. O cliente seleciona a opção “Efetuar pagamento”;</li> <li>3. O sistema calcula o valor do pedido, apresenta o valor do pedido e solicita os dados de pagamento [R1, R2];</li> <li>4. O cliente fornece os dados de pagamento;</li> <li>5. O sistema valida os dados de pagamento, gera o pedido e apresenta o número do pedido;</li> </ol>
<b>Fluxos Alternativos:</b>	<p>A1 – Excluir item</p> <ol style="list-style-type: none"> <li>1. O cliente seleciona um item da lista de produtos no carrinho de compras e seleciona a opção “Excluir item”;</li> <li>2. O sistema exclui o item selecionado e retorna para o passo 1 do fluxo principal;</li> </ol>
<b>Fluxos de Exceção:</b>	Não aplicável
<b>Regras:</b>	R1 – É obrigatório que o cliente tenha um desconto de 10% no valor do pedido se ele tiver mais de 1000,00 em compras nos últimos 30 dias.



	R2 – É obrigatório que o frete seja gratuito se o valor do pedido for maior que 500,00.
--	-----------------------------------------------------------------------------------------

### 3.2.2 Especificando de Forma Gráfica

Os diagramas de atividades da UML frequentemente são utilizados para especificar e detalhar os requisitos do software (Paydar e Kahani, 2015). Os diagramas de atividades da UML podem ser utilizados para especificar graficamente os casos de uso (Almendros-Jiménez *et al.* 2005; Gutiérrez *et al.* 2008; Mello *et al.* 2011). Massollar *et al.* (2012) propuseram um meta-modelo (UCModel) que utiliza o diagrama de atividades para especificar de forma gráfica os aspectos comportamentais do caso de uso. Este meta-modelo permite a estruturação da especificação dos casos de uso segundo um conjunto de critérios e restrições bem definidos, visando aprimorar o grau de formalismo das especificações dos casos de uso (Massollar *et al.* 2012).

O UCModel estende o diagrama de atividades da UML acrescentando novos elementos para a descrição do comportamento do caso de uso (OMG, 2010). Desses novos elementos adotados, há três tipos que representam as ações (interação) do ator e do sistema, representados através de estereótipos no diagrama (Massollar *et al.* 2012):

(a) <<actor\_action>>: representa uma ação do ator com o sistema na qual este faz uma requisição ao sistema informando os dados necessários;

(b) <<system\_action>>: representa uma ação do sistema cujos os resultados gerados não são diretamente observados pelo ator; e

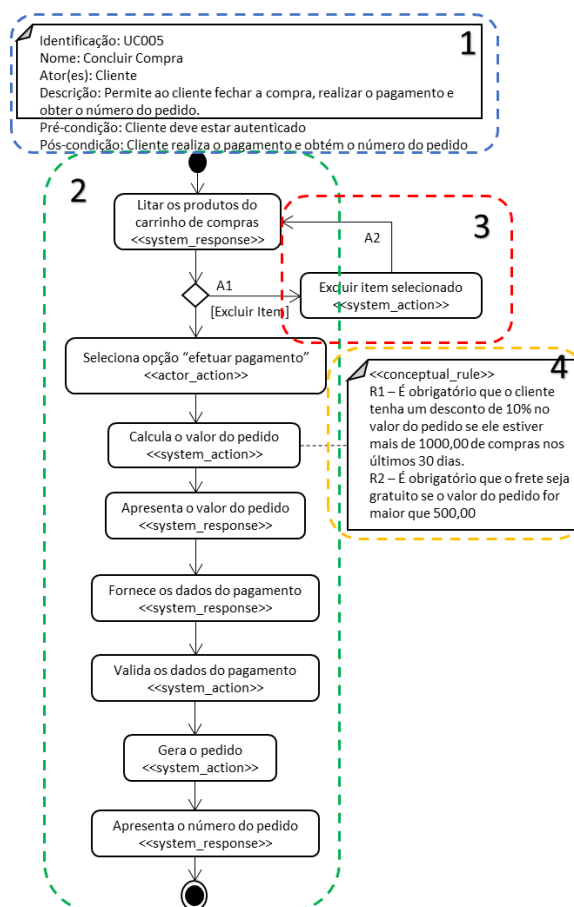
(c) <<system\_response>>: representa uma ação do sistema cujos os resultados são direta ou indiretamente observados pelo ator. Essa ação pode representar os resultados gerados anteriormente ou pode solicitar outros dados ao ator.

Outros elementos utilizados são os tipos de regras associadas ao caso de uso. Essas regras devem estar sempre associadas às ações que a afetam. Os estereótipos que representam essas regras são (Massollar *et al.* 2012):

- <<conceptual\_rule>>: regra relacionada à perspectiva de conceituação – restringe as ações do sistema, estabelecendo condições para sua execução;
- <<navigation\_rule>>: regra relacionada à perspectiva de navegação – restrições de quais caminhos podem ser navegados e quais informações serão solicitadas ao ator;

- <<presentation\_rule>>: regra relacionada à perspectiva de apresentação – restrições de como as informações serão apresentadas ao ator. A Figura 3.2 apresenta o formato gráfico representando uma especificação de um caso de uso gerados partir do UCMModel.

O item 1 da Figura 3.2 representa o conjunto de elementos que define o caso de uso (identificação, nome, atores, descrição, pré e pós-condições). O item 2 mostra o fluxo principal que contém uma sequência de ações que começa do nó inicial até o nó final. Cada ação é identificada pelos estereótipos que representam a ação do ator e sistema: *actor\_action*, *system\_action* e *system\_response*. O item 3 da Figura 3.2 mostra o fluxo alternativo identificado pelo nó de decisão. Neste ponto do diagrama, existe um caminho alternativo que pode ser executado dependendo da condição associada ao fluxo alternativo. No exemplo, o fluxo alternativo é identificado por “A1”. O item 4 da Figura 3.2 refere-se às regras de negócio que estão representadas pelos identificadores R1 e R2 usando os comentários no diagrama de atividades e classificados com os tipos de regras. Além disso, cada regra de negócio está associada à ação que será impactada por ela.



**Figura 3.2:** Exemplo de um caso de uso especificado com UCMModel baseado em Massollar *et al.* (2012).

### 3.3 ESTUDO EXPERIMENTAL

Esta seção apresenta o objetivo, planejamento e execução do estudo experimental. Além disso, apresenta os resultados da análise da corretude dos UCs gerados pelos formatos textual e gráfico.

#### 3.3.1 Planejamento do Estudo

Este estudo teve o objetivo de verificar qual dos dois formatos obtém melhores resultados em termos de: (a) tempo gasto para elaboração da especificação do caso de uso e (b) a corretude das especificações geradas pelos formatos. Além disso, foi realizada uma análise qualitativa das percepções dos participantes em relação à facilidade (de aprendizado e uso), a utilidade e a preferência em relação aos formatos utilizados para especificar um caso de uso. O objetivo deste estudo é apresentado de acordo com o paradigma GQM (Basili e Rombach, 1988), conforme a Tabela 3.2.

**Tabela 3.2:** Objetivo do Estudo usando o paradigma GQM (Basili e Rombach, 1988).

<b>Analisar</b>	Formato Textual (Cockburn, 2001) e Gráfico (UCModel proposto por Massollar et al. 2012) para especificação de casos de uso
<b>Com o propósito de</b>	Caracterizar
<b>Em relação à</b>	<b>Tempo gasto, corretude na especificação dos casos de uso e percepção sobre facilidade de uso e utilidade</b>
<b>Do ponto de vista</b>	Do pesquisador
<b>No contexto da</b>	Especificação de casos de uso de uma aplicação real por alunos de graduação dos cursos de Ciência da Computação e Sistemas de Informação da UFAM.

Considerou-se para análise dos dados quantitativos de **tempo** gasto e a **corretude** dos casos de uso gerados pelos participantes, conforme definidos na subseção 3.3.1.3. Os resultados quantitativos e qualitativos deste estudo são apresentados neste capítulo nas Seções 3.6 e 3.7.

Nesta fase, realizou-se a definição do escopo do estudo (quais formatos seriam utilizados), a preparação dos materiais que foram utilizados (elaboração dos requisitos, protótipos/*mockups* e dos diagramas), a seleção e treinamento dos participantes. Todas as atividades desta fase foram realizadas pelo moderador do estudo e foram revisadas por dois pesquisadores.

##### 3.3.1.1 Hipóteses

O estudo foi planejado e conduzido a fim de colocar à prova as seguintes hipóteses (nula e alternativa, respectivamente) em relação aos indicadores quantitativos:

- H01: O **tempo** para utilização dos formatos textual e gráfico para especificar um caso de uso é igual.
- HA1: O **tempo** necessário para especificar um caso de uso com o formato textual é diferente do formato gráfico.
- H02: A **corretude** da especificação do caso de uso com o formato textual é igual à com o formato gráfico.
- HA2: A **corretude** da especificação do caso de uso com o formato textual é diferente do formato gráfico.

Durante o planejamento, foram definidos os seguintes recursos necessários para a realização do estudo:

#### 3.3.1.2 *Contexto*

O estudo foi executado com requisitos, diagramas de caso de uso e *mockups* de um projeto real para um sistema Web de Agendamento de Veículos. O cenário utilizado para especificar o caso de uso está relacionado com a confirmação do condutor após a execução do serviço de agendamento de veículo. O objetivo é permitir a um condutor pesquisar no sistema seus agendamentos e escolher um deles para concluir. Ao escolher um agendamento o condutor informa os dados de conclusão. O estudo foi conduzido com estudantes de graduação do curso de Ciência da Computação e Sistema de Informação (segundo semestre de 2015) da Universidade Federal do Amazonas (UFAM). Os alunos haviam cursado a disciplina introdutória de “Engenharia de Software” e estavam cursando a disciplina de “Análise e Projeto de Sistemas”.

#### 3.3.1.3 *Variáveis de Seleção*

As variáveis independentes foram os formatos para especificação do caso de uso (tratamentos: gráfico e textual). As variáveis dependentes foram os indicadores de tempo e corretude. O **tempo** é calculado em minutos e representa o tempo total gasto por cada participante para especificar o caso de uso utilizando um formato. O indicador **corretude** verifica o grau de quão correto os casos de uso foram elaborados. Para isso, foi verificado o número de defeitos nos casos de uso criados pelos participantes, utilizando os itens de verificação apresentados na Tabela 3.4.

#### 3.3.1.4 *Ambiente*

O estudo foi realizado em um ambiente acadêmico (*off-line*) dentro de uma disciplina ofertada aos cursos de Ciência da Computação e Sistemas de Informação. Para a execução do estudo foram reservadas duas salas de aula, uma para cada grupo.

#### 3.3.1.5 *Participantes*

Quarenta e quatro (44) participantes assinaram o formulário de consentimento e preencheram o formulário de caracterização. Este formulário captura dados que permitem avaliar o perfil dos participantes em relação à experiência em especificação de casos de uso e conhecimento em modelagem de sistemas, como mostra a Tabela 3.3. Os dados informados foram usados para caracterizar o perfil dos participantes em relação à experiência: nenhuma (N), baixa (B), média (M) e alta (A).

Um participante é considerado com alta experiência se tiver participado em mais de 5 projetos de software na indústria onde modelou diagramas da UML e atuou como analista na especificação de casos de uso. Um participante com média experiência deve ter participado de 1 a 4 projetos na indústria onde modelou diagramas da UML e especificou casos de uso. Um participante com baixa experiência deve ter participado de pelo menos um 1 projeto onde modelou diagramas da UML e especificou casos de uso em sala de aula, com nenhuma experiência o que não tem nenhum conhecimento em modelagem de diagramas da UML ou possui alguma noção sobre a linguagem UML através de leituras/palestras/tutoriais, mas sem experiência prática ou não possui nenhuma experiência prévia em especificação de casos de uso. A Tabela 3.6 (colunas CI e CII) apresenta a caracterização do perfil dos participantes.

#### 3.3.1.6 *Artefatos*

Os instrumentos utilizados neste estudo experimental foram: o Termo de Consentimento Livre e Esclarecido (TCLE), o formulário de caracterização, os diagramas de casos de uso, os requisitos da aplicação, os *mockups* (protótipos de baixa fidelidade do caso de uso), instruções para orientar na modelagem com o formato gráfico (UCModel) e questionários pós-modelagem. Todos os artefatos do estudo foram validados por três autores deste estudo e pelos desenvolvedores do sistema (ver Figura 3.3). Neste estudo a especificação dos casos de uso foi realizada manualmente sem o uso de ferramentas.

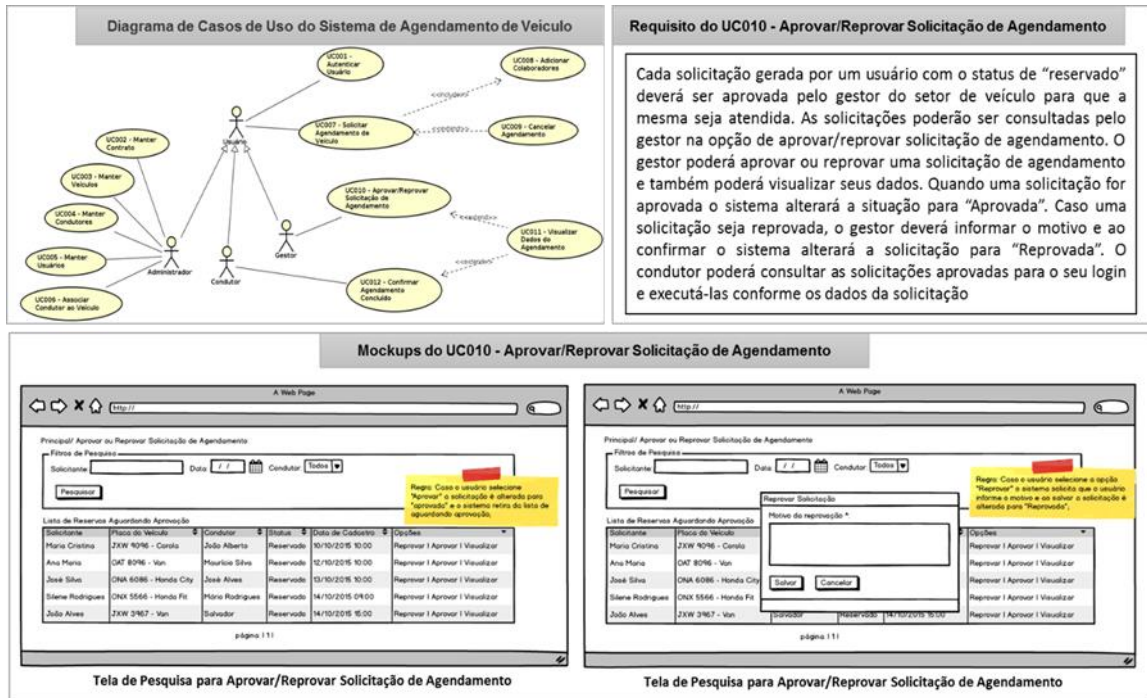


Figura 3.3: Artefatos utilizados no experimento.

Tabela 3.3: Questões para avaliar o conhecimento sobre Modelagem e Especificação de Casos de Uso.

Conhecimento em Modelagem de Sistemas	
Grau de experiência	Como adquiriu
<input type="checkbox"/> <b>NENHUMA EXPERIÊNCIA</b>	Não possui conhecimento em modelagem de diagramas utilizando a UML ou possui alguma noção sobre a linguagem UML através de leituras/palestras/tutoriais, mas sem experiência prática;
<input type="checkbox"/> <b>BAIXA</b>	Participou de pelo menos um 1 projeto onde modelou diagramas da UML em sala de aula;
<input type="checkbox"/> <b>MÉDIA</b>	Participou de 1 a 4 projetos na <b>indústria</b> onde modelou diagramas da UML.
<input type="checkbox"/> <b>ALTA</b>	Participou em mais de 5 projetos de software na <b>indústria</b> onde modelou diagramas da UML.
Experiência em Especificação de Casos de Uso	
<input type="checkbox"/> <b>NENHUMA EXPERIÊNCIA</b>	Não possui experiência prévia em especificação de casos de uso ou possui algumas noções através de leituras em livros, artigos, aulas e outros, mas sem experiência prática na indústria;
<input type="checkbox"/> <b>BAIXA</b>	Participou de pelos menos um projeto em <b>sala de aula</b> especificando caso de uso;
<input type="checkbox"/> <b>MÉDIA</b>	Participou de 1 a 4 projetos na <b>indústria</b> atuando como Analista na especificação de caso de uso;
<input type="checkbox"/> <b>ALTA</b>	Participou em mais de 5 projetos na <b>indústria</b> atuando como Analista na especificação de caso de uso;

### 3.3.1.7 Treinamento

Os participantes receberam treinamento de duas (02) horas em um mesmo ambiente sobre os dois formatos de especificação (gráfico e textual). Também foram realizados exercícios práticos de modelagem para especificar casos de uso com o formato gráfico e

também especificando com o formato textual. Este tempo foi suficiente para apresentar os formatos e exemplificar seu uso.

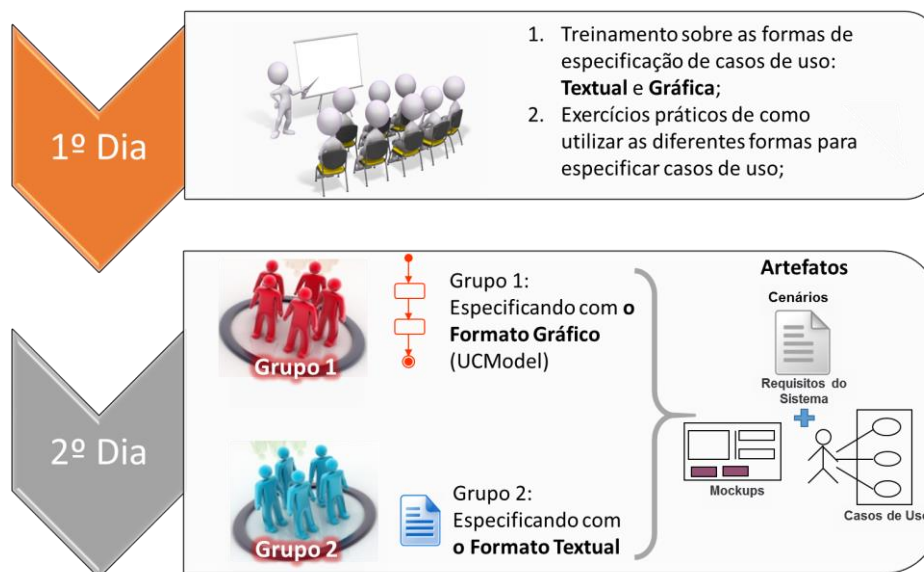
#### *3.3.1.8 Projeto Experimental*

O design aplicado no estudo experimental foi um fator com dois tratamentos (Wöhlin *et al.* 2000). Com base nos dados coletados através do formulário de caracterização (princípio de design de balanceamento conforme Wöhlin *et al.* 2000), os participantes foram divididos em grupos (princípio de design de agrupamento conforme Wöhlin *et al.* 2000). Os participantes foram atribuídos aleatoriamente aos grupos. Todos os participantes alocados em cada grupo especificaram o mesmo cenário. Cada grupo foi formado por 22 participantes. Cada grupo utilizou apenas um tratamento.

#### *3.3.1.9 Execução do Estudo*

Os participantes foram alocados em salas diferentes de acordo com o grupo em que foram atribuídos (Tabela 3.6). O Grupo 1 utilizou formato gráfico e o Grupo 2 utilizou o formato textual. Cada participante recebeu individualmente os artefatos descritos no item **Artefatos**. Todos os participantes (Grupo 1 e Grupo 2) incluíram o tempo total gasto para a elaboração do caso de uso, conforme mostra a Figura 3.4. Após o experimento foi aplicado um questionário pós-modelagem visando capturar a percepção dos participantes após o uso do formato gráfico. Para isto, apenas o grupo 1 respondeu ao questionário pós-modelagem devido ao foco principal de observação ser o uso do formato gráfico.

Além disso, foram selecionados dois monitores para auxiliarem na execução deste estudo experimental. Eles receberam treinamento sobre as atividades que seriam realizadas durante o estudo. Vale ressaltar que durante o processo de construção dos casos de uso não foi permitida a comunicação entre os participantes e também os participantes não receberam nenhum auxílio dos pesquisadores envolvidos.



**Figura 3.4:** Passos utilizados na execução do estudo.

### 3.3.2 Resultados da Análise da Corretude dos Casos de Uso gerados pelos Formatos

Para avaliar a corretude das especificações geradas pelos participantes do estudo, primeiramente três pesquisadores realizaram a análise de 16 métodos encontrados na revisão sistemática da literatura realizada por Tiwari e Gupta (2015) (Ver APÊNDICE C – Análise de Viabilidade das Tecnologias). Cada uma das tecnologias avaliada foi apresentada no Capítulo 2 (Subseção 2.2).

Após esta análise os métodos propostos por Phalp *et al.* (2007) e por Anda *et al.* (2009) foram selecionados. Esses dois métodos foram escolhidos por apresentar itens de verificação que auxiliam a avaliação da corretude das especificações de casos de uso. Para investigar qual dos métodos auxiliava a identificação do maior número de defeitos nas especificações, foram utilizadas quatro especificações de casos de uso, duas para cada formato. A avaliação das especificações foi realizada por um pesquisador e, posteriormente, conferida e discutida por outros dois pesquisadores.

O método escolhido foi de Anda *et al.* (2009) chamado de “atributos de verificação de modelos de casos de uso”. A razão da escolha deste método foi por este apresentar diretrizes que auxiliam de forma mais focada e detalhada de quais informações os modelos UCs devem conter para atender aos requisitos de qualidade de um caso de uso. A taxonomia do método proposto pelos autores auxilia na verificação de defeitos do caso de uso de forma guiada. Para cada atributo de correção são definidas diretrizes que o caso de uso deve abranger. Com isso, a execução da avaliação se torna mais direcionada e orientada para identificar os defeitos da especificação de um caso de uso (Anda *et al.* 2009).



Após a definição do método a ser utilizado para avaliar a corretude das especificações de casos de uso geradas no estudo, retirou-se a referência dos participantes, identificando-os com um código. O processo de avaliação das especificações foi realizado em duas etapas. Na primeira etapa um pesquisador avaliou as especificações que os participantes elaboraram a partir dos artefatos entregues (casos de uso, requisitos e os *mockups*). Na segunda etapa, dois pesquisadores revisaram por completo a avaliação realizada.

Durante a avaliação dos casos de uso houve a necessidade de incluir novos itens para complementar o Checklist de verificação proposto por Anda *et al.* (2009). Por exemplo, no atributo de verificação “Incorretude da funcionalidade” os novos itens serviram para identificar fluxos alternativos, de exceção ou regras de negócio que estavam descritas de forma incompleta. Esses novos itens estão sinalizados em negrito na Tabela 3.4 (coluna “Definição e Execução”). Foram utilizados os seguintes atributos de verificação baseados em Anda *et al.* (2009): Incorretude, Redundância, Falha de Integridade, Inconsistência, Ambiguidade, Falha de Legibilidade, Falha de Nível de Abstração, Falha na Manutenibilidade e Inverificável (ver a lista de atributos de qualidade no APÊNDICE B).

**Tabela 3.4:** Itens de Verificação dos Atributos de Modelos de Casos de Uso.

Atributos de Qualidade	Definição e Execução
1) <b>Incorretude na funcionalidade</b>	Os requisitos identificados devem ser representados corretamente sem informações que mudem as necessidades. Não deve conter: <ul style="list-style-type: none"> <li>1.1. atores com descrições incorretas;</li> <li>1.2. casos de uso que não são independentes ou tarefas distintas;</li> <li>1.3. dependências incorretas entre atores e casos de uso ou entre casos de uso; ou</li> <li>1.4. casos de uso com eventos ou fatos incorretos, fluxos alternativos incorretos ou pré ou pós-condições que não são realmente pré-requisitos para os estados iniciais e finais do caso de uso;               <ul style="list-style-type: none"> <li><b>1.4.1. fluxos alternativos, fluxos de exceção, regras de negócio descritos mas que não foram referenciados em nenhum momento no caso de uso;</b></li> <li><b>1.4.2. referências incorretas de fluxos alternativos, fluxos de exceção e regras de negócio que não foram descritos no caso de uso;</b></li> <li><b>1.4.3. fluxo principal, fluxos alternativos, fluxos de exceção e regras de negócio incompletos, quando ficou faltando descrever algumas informações importantes para sua compreensão;</b></li> </ul> </li> </ul>
2) <b>Redundância</b>	Não deve conter informações superficiais, como: <ul style="list-style-type: none"> <li>2.1. atores que não vão interagir com o sistema;</li> <li>2.2. casos de uso com funcionalidades que estão fora do contexto do sistema;</li> <li>2.3. dependências superficiais entre atores e casos de uso ou entre casos de uso; ou</li> <li>2.4. atores, pré-condições ou eventos superficiais;</li> </ul>
3) <b>Falha de Integridade</b>	Todos os requisitos explícitos devem ser atendidos pelo modelo de casos de uso e não deve faltar: <ul style="list-style-type: none"> <li>3.1. atores na forma de pessoas ou entidades externas que têm objetivos relacionados ao sistema;</li> <li>3.2. funcionalidades que não foram descritas como casos de uso ou os objetivos dos atores que não possuem um caso de uso relacionado;</li> <li>3.3. entradas ou saídas para casos de uso;</li> <li>3.4. eventos que são necessários para a compreensão do caso de uso;</li> <li>3.5. fluxos alternativos; ou</li> <li>3.6. dependências entre o ator e um caso de uso ou entre casos de uso;</li> <li><b>3.7. Faltou descrever o objetivo do caso de uso;</b></li> <li><b>3.8. Faltou descrever algum elemento do modelo que está sendo utilizado como:</b></li> </ul>

Atributos de Qualidade	Definição e Execução
	<b>identificador do caso de uso, nome do caso de uso, pré e pós-condições, regras de negócio;</b>
<b>4) Inconsistência</b>	Deve ter consistência entre todos os elementos do modelo de casos de uso. A estrutura dos casos de uso, o uso da linguagem e gramática devem ser consistentes em todos os casos de uso. Não deve haver nenhuma: <ul style="list-style-type: none"> <li>4.1. descrição de atores incompatíveis com o comportamento do caso de uso;</li> <li>4.2. inconsistências entre as descrições de casos de uso;</li> <li>4.3. inconsistências na numeração dos fluxos alternativos, ou de eventos;</li> <li>4.4. pré e pós-condições que são inconsistentes com as descrições dos casos de uso;</li> </ul>
<b>5) Ambiguidade</b>	Não deve existir ambiguidade nas descrições de casos de uso ou em terminologias como: <ul style="list-style-type: none"> <li>5.1. nomes de atores que não refletem seu papel;</li> <li>5.2. descrições de atores muito grandes;</li> <li>5.3. nomes de casos de uso que não refletem os objetivos do caso de uso;</li> <li>5.4. descrições ambíguas nas pré ou pós-condições;</li> <li>5.5. descrições ambíguas nos eventos;</li> <li>5.6. descrições ambíguas nos fluxos alternativos ou de exceções;</li> <li>5.7. qualquer uso de advérbios, adjetivos, pronomes, sinônimos, homônimos, referências, ou negativos</li> </ul>
<b>6) Falha de Legibilidade</b>	As descrições de casos de uso devem ser corretas e utilizar uma terminologia comum. Não deve haver informações duplicadas, casos de uso muito grandes ou muito pequenos, ou suposição de como um ator irá executar as suas próprias tarefas internas. Cada caso de uso deve seguir um caminho lógico para um objetivo claro. <b>Faltou utilizar um nome ou termo para identificar um fluxo alternativo, de exceção ou regra de negócio, como por exemplo: A1, FA01 para Fluxo Alternativo 1; E1, FE01 para Fluxo de Exceção e RN01, RN ou Regra 1 para Regras de Negócio;</b>
<b>7) Nível de Abstração</b>	No modelo de caso de uso deve estar descrito somente o que o sistema deve fazer, em um nível adequado de granularidade, sem descrições da interface do usuário ou soluções de projeto.
<b>8) Falha de Manutenibilidade</b>	O modelo de caso de uso deve permitir que alterações sejam feitas sem mudar a consistência, a estrutura e o estilo dos casos de uso.
<b>9) Inverificável</b>	A especificação do caso de uso deve permitir que seja testada por uma pessoa, ou seja, quando o sistema estiver pronto, seja possível verificar se os itens descritos foram atendidos. Além disso, também deve ser possível a verificação automática.

Além disso, foram criadas categorias para os principais tipos de defeitos (ver Tabela 3.5). Cada defeito possui um grau de severidade. Por exemplo, o defeito de número quinze “15 – Regras Incompletas” está associado à categoria “Incorretude”, e classificado ao grau de severidade “Médio”. A severidade do tipo “Grave” foi utilizada para classificar defeitos de omissão de fluxos ou regras de negócio que não foram descritos no caso de uso. Os defeitos de severidade “Média” foram utilizados para classificar fluxos ou regras de negócio que não foram descritos por completo, ou que foram descritos de forma incorreta, passos de fluxos que ficaram faltando, fatos incorretos ou informações estranhas descritas do caso de uso. Os defeitos de severidade “Baixa” identificaram aqueles defeitos que não prejudicavam a compreensão e entendimento do caso de uso.

**Tabela 3.5:** Defeitos e Grau de Severidade.

Núm.	Cód.	Definição dos Defeitos	Atributo de Qualidade	Grau de Severidade
01	<b>COR_01</b>	Pré e Pós-condições, Eventos Incompletos	Incorretude	Baixo
02	<b>COR_02</b>	Referências de Fluxos, Regras Incorretas	Incorretude	Baixo
03	<b>COR_03</b>	Faltou referenciar Casos de Uso/Fluxos/Regras no caso de uso	Incorretude	Baixo
04	<b>COR_04</b>	Identificação de Atores e Casos de Uso Incorretos	Incorretude	Baixo

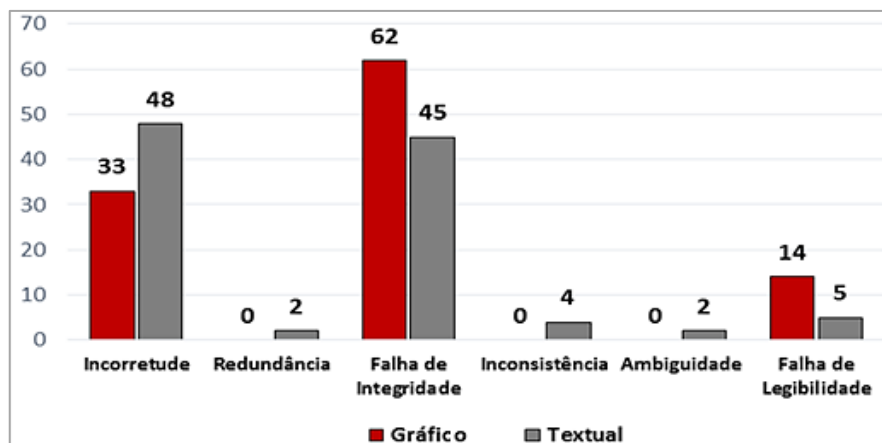
Núm.	Cód.	Definição dos Defeitos	Atributo de Qualidade	Grau de Severidade
05	<b>COR_05</b>	Descrição do Caso de Uso Incorreta	Incorretude	Baixo
06	<b>RED_01</b>	Fluxos, Eventos que realizam a mesma tarefa	Redundância	Baixo
07	<b>INT_01</b>	Faltou identificar/descrever o ator, pré e pós-condição	Falha de Integridade	Baixo
08	<b>CON_01</b>	Fluxos com numeração incorreta	Inconsistência	Baixo
09	<b>LEG_01</b>	Não utilizou uma terminologia comum	Falha de Legibilidade	Baixo
10	<b>LEG_02</b>	Fluxos e Regras não claros (confusos)	Falha de Legibilidade	Baixo
11	<b>CON_03</b>	Eventos Inconsistentes com o comportamento do Caso de Uso	Inconsistência	Baixo
12	<b>AMB_01</b>	Descrições ambíguas nos eventos, Fluxos, Regras	Ambiguidade	Médio
13	<b>CON_02</b>	Regras Inconsistentes com o comportamento do Caso de Uso	Inconsistência	Médio
14	<b>COR_06</b>	Fluxos incorretos com mais de um cenário	Incorretude	Médio
15	<b>COR_07</b>	Regras Incompletas	Incorretude	Médio
16	<b>COR_08</b>	Omissão de Passos (faltam passos em algum fluxo)	Incorretude	Médio
17	<b>COR_09</b>	Fato Incorreto/Informações estranhas descritas nos Fluxos/Regras	Incorretude	Médio
18	<b>INT_02</b>	Omissão de Fluxos/Regras	Falha de Integridade	Grave

### 3.4 RESULTADOS OBTIDOS

Nesta seção são apresentados os resultados quantitativos e qualitativos obtidos no estudo experimental. Na Subseção 3.4.1 são descritos os resultados quantitativos relativos ao tempo para construção e a corretude das especificações dos casos de uso gerados pelos dois formatos. Os resultados qualitativos relativos à percepção dos participantes quanto à facilidade e utilidade de cada formato são apresentados na Subseção 3.4.2.

#### 3.4.1 Resultados Quantitativos

A avaliação dos casos de uso resultou em 215 defeitos, dos quais 109 foram identificados nos casos de uso criados utilizando o formato gráfico e 106 dentre aqueles criados utilizando o formato textual. A Figura 3.5 mostra o número dos diferentes tipos de defeitos identificados na avaliação dos casos de uso, classificados de acordo com os atributos de verificação (Anda *et al.* 2009).

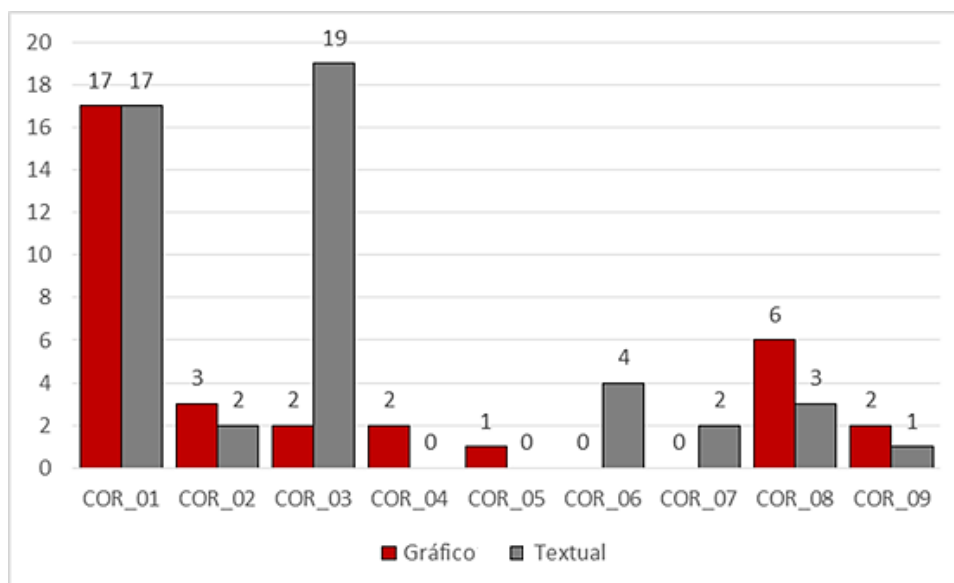


**Figura 3.5:** Número de diferentes tipos de defeitos encontrados nos casos de uso gerados pelos formatos gráfico e textual.

O maior número de defeitos encontrados nas especificações geradas pelo formato gráfico foi de Falha de Integridade (62 defeitos), seguido por Incorretude (33) e Falha de Legibilidade (14). Não foram identificados defeitos de Redundância, Consistência e Ambiguidade nas especificações geradas por esse formato. Nas especificações geradas pelos participantes que utilizaram o formato textual (Figura 3.5), os defeitos mais encontrados foram de Incorretude (48 defeitos), seguidos por Falha de Integridade (45), Falha de Legibilidade, Inconsistência, Redundância e Ambiguidade.

Nos dois formatos, os defeitos de Incorretude (textual – 48 defeitos; e gráfica – 33 defeitos) estavam relacionados a descrições incompletas, por exemplo, pré e pós-condições, referências de fluxos alternativos, regras e outros casos de uso que não foram referenciados na descrição do caso de uso. Já os defeitos de Falha de Integridade (textual – 45 defeitos; e gráfica – 62 defeitos) estavam relacionados à omissão de fluxos ou regras que não foram descritos no caso de uso. A partir da classificação dos defeitos e o grau de severidade mostrados na Tabela 3.5, confrontou-se o número de defeitos encontrados por participante, que são apresentados na Seção 3.6.1.

Para detalhar os defeitos relacionados à Incorretude encontrados em cada formato a Figura 3.7 apresenta a quantidade de defeitos por código de defeito (COR\_01, COR\_02, COR\_03, COR\_04, COR\_05, COR\_06, COR\_07, COR\_08 e COR\_09). Cada código é um defeito da categoria Incorretude, descrito na Tabela 3.5.



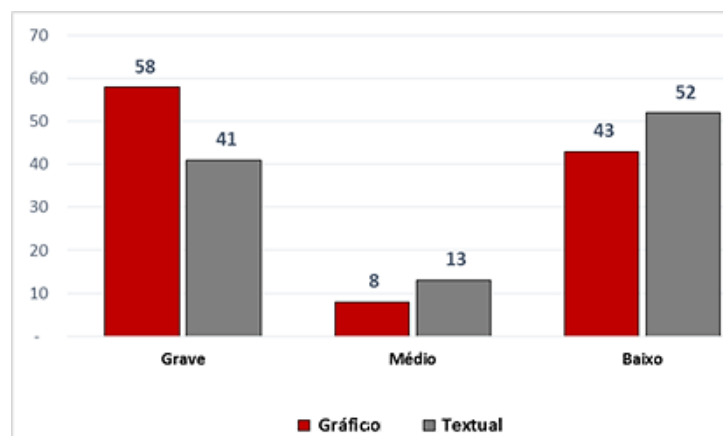
**Figura 3.6:** Número de diferentes tipos de defeitos encontrados nos UCs gerados pelos formatos.

Para o defeito “Pré e Pós-condições, Eventos Incompletos” (COR\_01) os dois formatos obtiveram a mesma quantidade, ou seja, nenhum formato foi melhor que o outro (Textual – 17 defeitos e Gráfico – 17 defeitos).

A Figura 3.6 mostra que o formato gráfico foi melhor para referenciar fluxos, regras e casos de uso no diagrama (COR\_03) do que o formato textual. Também foi a melhor em elaborar fluxos corretos (COR\_06), contendo um cenário, e descrever as regras por completo (COR\_07) do que o formato textual, ou seja, para o formato gráfico não ocorreu nenhum defeito nestes itens.

Conforme mostra Figura 3.6 o formato textual foi melhor do que o formato gráfico nos itens: COR\_04, COR\_05, COR\_06, COR\_08 e COR\_09. Para os itens COR\_04 e COR\_05, não ocorreu nenhum defeito nos casos de uso gerados pelo formato textual, ou seja, os casos de uso continuam a identificação dos atores e a descrição do objetivo do caso de uso.

A Figura 3.7 mostra o número de defeitos encontrados nas especificações geradas por cada formato distribuídos por grau de severidade. Os casos de uso especificados utilizando o formato gráfico tiveram o maior número de defeitos do tipo Grave (58), 17 defeitos a mais que a aqueles especificados com o formato textual (41). Isto mostra que nessas especificações não foram descritos os fluxos ou regras de negócio importantes para a completude do caso de uso. Em comparação com o formato textual, as especificações geradas com o formato gráfico obtiveram menor número de defeitos de grau de severidade Médio e Baixo.



**Figura 3.7:** Número de diferentes tipos de defeitos encontrados nos UCs gerados pelos formatos.

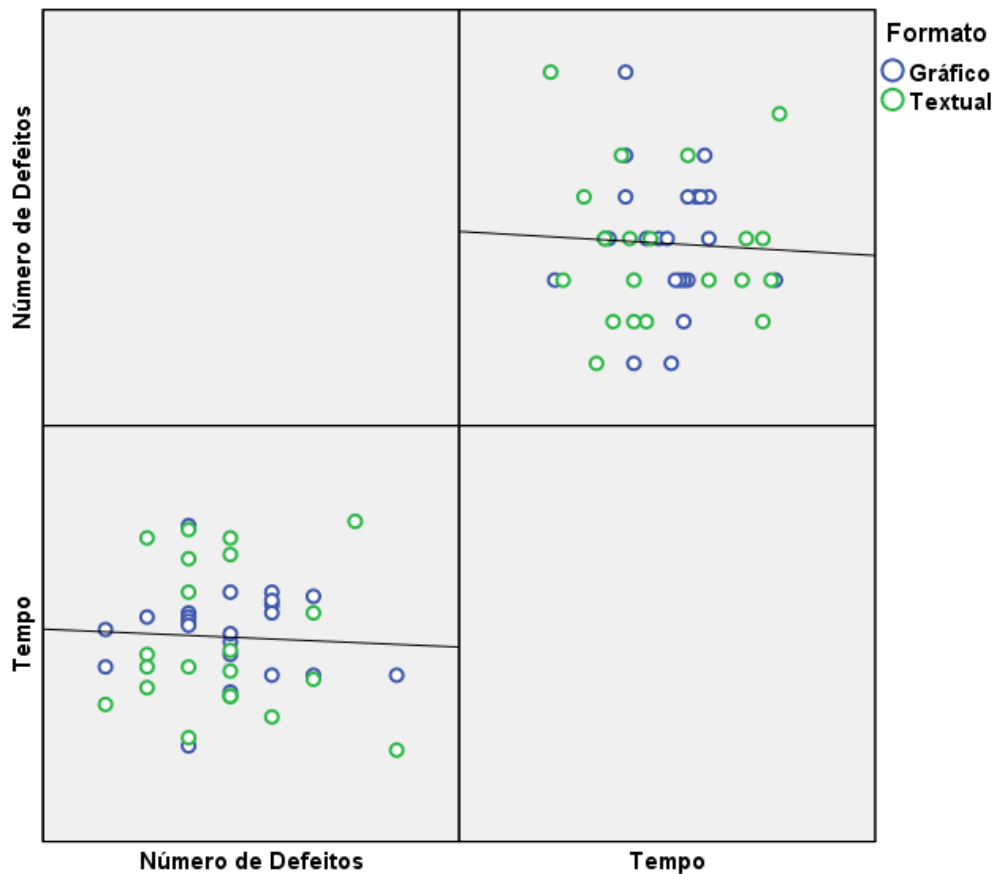
A Tabela 3.6 sumariza os resultados relativos à **quantidade de defeitos** e o **tempo gasto** por participante para elaborar a especificação. As colunas P1 e P2 apresentam o código dos participantes que utilizaram, respectivamente, o formato gráfico e a textual. As colunas ND (1) e ND (2) apresentam respectivamente a quantidade de defeitos encontrados na avaliação dos casos de uso gerados pelos formatos gráfico e textual. As colunas T1 e T2 apresentam o tempo, em minutos, que cada participante utilizou para especificar o caso de uso. As colunas CI e CII apresentam os níveis de conhecimento, respectivamente, em modelagem e em especificação de casos de uso dos participantes.

**Tabela 3.6:** Resumo de defeitos e tempo gasto por participante.

GRUPO 1 – UCModel			GRUPO 2 - Textual		
Partic.	Núm. Defeitos (1)	T1 (min.)	Partic.	Núm. Defeitos (2)	T2 (min.)
S1	4	60	S30	3	50
S2	7	64	S31	5	40
S3	9	45	S32	9	27
S4	6	45	S33	6	35
S5	4	59	S34	4	30
S6	4	58	S35	3	42
S7	4	28	S36	5	40
S8	2	47	S37	2	38
S9	5	41	S38	5	40
S10	5	53	S39	5	46
S11	6	62	S40	3	47
S12	6	65	S41	7	60
S13	4	81	S42	5	74
S14	6	63	S43	4	65
S15	7	45	S44	4	47
S16	4	57	S45	5	51
S17	5	50	S46	4	80
S18	5	55	S47	4	73
S19	3	59	S48	5	78
S20	2	56	S49	3	78
S21	6	60	S50	8	82
S22	5	65	S51	7	44

Legenda: **Partic.** → Participantes  
**Núm. Defeitos (1)** → Número de defeitos dos modelos gerados pela UCModel  
**Núm. Defeitos (2)** → Número de defeitos dos modelos gerados pelo formato Textual  
**T1 (min.)** → Tempo gasto em minutos para especificar com UCModel  
**T2 (min)** → Tempo gasto em minutos para especificar com o formato Textual

Para analisar o tempo e a corretude obtida pelos grupos, foram realizadas análises estatísticas utilizando a ferramenta SPSS V.201. Conduziu-se o teste de normalidade de Shapiro-Wilk com  $\alpha=0.05$  para tempo e corretude. O teste mostrou que a distribuição dos valores de tempo não é normal para ambos os grupos (com  $p=0,328$  para as especificações com o formato gráfico e  $p=0,036$  para as especificações com o formato textual) e que a distribuição dos valores de corretude não é normal (com  $p= 0,318$  para as especificações com o formato gráfico e  $p= 0,093$  para as especificações com o formato textual). Devido a esses resultados, utilizou-se o teste não-paramétrico *Mann-Whitney*. Além disso, foi a Figura 3.8 apresenta o gráfico de dispersão, utilizado para mostrar os pontos no espaço cartesiano (XY) que representam simultaneamente os valores das duas variáveis: tempo (X) e corretude (Y).



**Figura 3.8:** Gráfico de dispersão do tempo e corretude por Grupo.

A Figura 3.9 mostra o gráfico de boxplot com a distribuição de tempo dispendido pelos participantes por grupo. Observou-se que a mediana do grupo que utilizou o formato gráfico é maior que a mediana do grupo que utilizou o formato textual. No entanto, quando as duas amostras foram comparadas utilizando o teste de Mann-Whitney, não foi possível encontrar diferença estatisticamente significativa entre os dois grupos ( $p = 0,341$ ). Esses resultados

apoiam a H01, que afirma que não há diferença entre utilizar o formato textual e o formato gráfico para especificar um caso de uso com relação ao indicador de tempo.

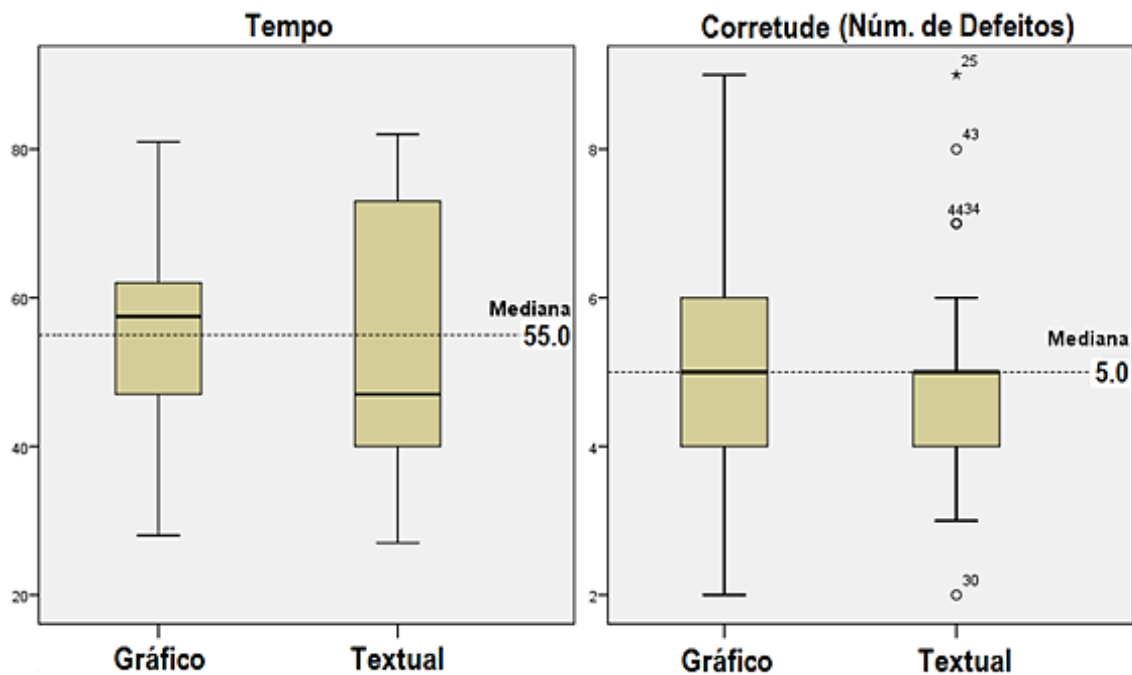


Figura 3.9: Boxplots do tempo e corretude por Grupo.

A Figura 3.9 também mostra o gráfico de *boxplot* comparando a distribuição do número de defeitos (corretude) por formato. Notou-se que a corretude para os dois grupos é similar. O resultado do teste não-paramétrico mostrou que não há diferença estatisticamente significativa entre os grupos ( $p = 0,606$ ). Esses resultados apoiam a H02, que afirma que não há diferença entre utilizar o formato textual e o formato gráfico para especificar um caso de uso com relação ao indicador de corretude.

Os *outliers* (S32, S37, S40, S41, S50, S51) do formato textual apresentaram os mesmos tipos de defeitos (Falha de Integridade, Incorretude) de omissão de fluxos ou regras no caso de uso, além de alguns defeitos de Falha de Legibilidade. Contudo, o participante S32, que possui conhecimento médio em especificação de casos de uso, apresentou o pior desempenho no indicador corretude: obteve nove defeitos distribuídos em Falha de Integridade, Incorretude, Inconsistência e Falha de Legibilidade. Em comparação com os demais *outliers*, o participante 32 foi o que gastou menos tempo (27 min.) para especificar o caso de uso, mas consequentemente omitiu fluxos, regras, passos e não identificou os atores, pré e pós-condições do caso de uso. Já o participante S37, que também possui conhecimento médio em especificação de casos de uso, apresentou melhor desempenho nos dois indicadores (Tempo e Corretude), gastando onze (11) minutos a mais que o participante S32.



Apesar dos participantes S32 e S37 possuírem o mesmo nível de conhecimento em especificação de casos de uso e utilizarem um tempo aproximado, podem não possuir a mesma experiência individual para especificar casos de uso. Segundo Wieggers (1999), uma boa especificação de requisitos é adquirida através de experiências anteriores, em que se aprende com os problemas encontrados no passado.

Dentre todos os participantes do formato textual, o participante S50 gastou mais tempo para especificar o caso de uso e obteve oito defeitos distribuídos em Falha de Integridade, Incorretude e Redundância. Já o participante S51 gastou metade do tempo e apresentou um defeito a menos que o participante S50. Apesar destes participantes (S50 e S51) possuírem o mesmo nível (baixo) de conhecimento em especificação de casos de uso, o nível de conhecimento pode não influenciar no indicador tempo. Isto sugere que um participante com baixo conhecimento e que ainda está aprendendo a especificar casos de uso, o tempo gasto, pode variar para mais ou para menos e isso pode depender da habilidade individual de cada participante.

O participante S3, do formato gráfico, obteve a mesma quantidade de defeitos que o participante S32 do formato textual. Porém, na especificação do caso de uso do participante S3 não ocorreu defeitos de Falha de Legibilidade e Inconsistência, sugerindo que no formato gráfico o pouco uso de textos no diagrama podem minimizar estes tipos de defeitos. Isto sugere que o formato gráfico pode ser útil no aprendizado de especificação de casos de uso para participantes com baixo conhecimento em especificação e modelagem de diagramas (UML). Além disso, o formato gráfico pode reduzir defeitos de inconsistência.

Observou-se que a maioria dos defeitos dos participantes S2, S3, S8 e S15 estão relacionados à Falha de Integridade, ou seja, à omissão de fluxos ou regras no caso de uso. Apesar desse tipo de defeito também ocorrer no formato textual, a quantidade de omissão de fluxos e regras foi maior no formato gráfico, evidenciando que os participantes do formato gráfico construíram o cenário principal (fluxo principal) do caso de uso e deixaram de construir os outros cenários (fluxos alternativos, fluxos de exceção e regras). Isso pode ter ocorrido devido aos participantes apresentarem dificuldades em construir os fluxos alternativos, fluxos de exceção e regras de negócio com esse formato. Há evidências sobre essas dificuldades relatadas por alguns participantes, por exemplo, o participante S17 afirmou “onde inserir fluxos alternativos” e o participante S18 afirmou “... é meio estranho perceber o momento de inserir um fluxo alternativo”.

O participante S8, do formato gráfico, apresentou o melhor desempenho nos dois indicadores (Tempo e Corretude), apesar de possuir baixo conhecimento em modelagem UML e especificação de casos de uso. Comparando o indicador tempo dos participantes S8 e S37, o participante S37, do formato textual, gastou menos tempo na elaboração do caso de uso e obteve melhor desempenho nos dois indicadores (Tempo e Corretude). Isso pode ser justificado pelo fato do participante S37 possuir média experiência em especificação de casos de uso; dessa forma, gastou pouco tempo na elaboração do caso de uso.

O participante S20 obteve a mesma quantidade de defeitos de omissão de fluxos ou regras que o participante S8. Porém, gastou nove minutos a mais para elaborar o caso de uso que o participante S8. Observou-se que o participante S20 mencionou que teve dificuldades em “Arranjar as posições das atividades de forma organizada, uma vez que foi feita a mão ou papel. Organizar os fluxos de forma correta com os componentes corretos”. Nesse sentido, é justificado o motivo da omissão de fluxos e regras que não estavam presentes no caso de uso deste participante. Já o participante S8 relatou dificuldades em “Entender perfeitamente o caso de uso”, sugerindo que o conjunto de requisitos (cenários, mockups e diagramas de casos de uso) que descrevem a funcionalidade do caso de uso podem causar falha de integridade se não estiverem bem descritos. Apesar dos participantes S8 e S20 possuírem o mesmo nível de experiência (baixo) e apresentarem diferentes dificuldades, conseguiram manter um tempo equivalente e com melhor desempenho no indicador corretude.

#### 3.4.1.1 Percepção sobre o Formato Gráfico

A fim de avaliar a percepção dos participantes quanto à facilidade (aprendizado e de uso) e à utilidade do formato gráfico, questionários pós-modelagem foram aplicados e analisados. Os participantes do Grupo 1 forneceram suas respostas em uma escala de seis pontos, baseados nos questionários aplicados por Lanubile et al. (1998) (variando de concordo totalmente a discordo totalmente). Como sugerido por Laitenberger e Dreyer (1998), não foi utilizado um valor neutro como intermediário, pois estes não oferecem informações sobre a tendência dos participantes (concordar ou discordar). Os itens avaliados estão na Tabela 3.7.

**Tabela 3.7:** Preferências dos formatos para especificar UCs.

<b>Facilidade</b>	(Q1) Foi fácil aprender a especificar um caso de uso com este formato?
	(Q2) Foi fácil construir a especificação do caso de uso utilizando este formato?
<b>Utilidade</b>	(Q3) Utilizar este formato aumentou meu entendimento sobre a interação do ator com o sistema?

(Q4) O formato foi útil para especificar um UC?

A Figura 3.10 apresenta a percepção dos participantes com relação à facilidade de especificar casos de uso utilizando o formato gráfico. Na figura o eixo horizontal refere-se ao grau de aceitação dos participantes. Nas barras foram inseridos códigos que representam os participantes (S1, S2, e assim sucessivamente) conforme a Tabela 3.6.

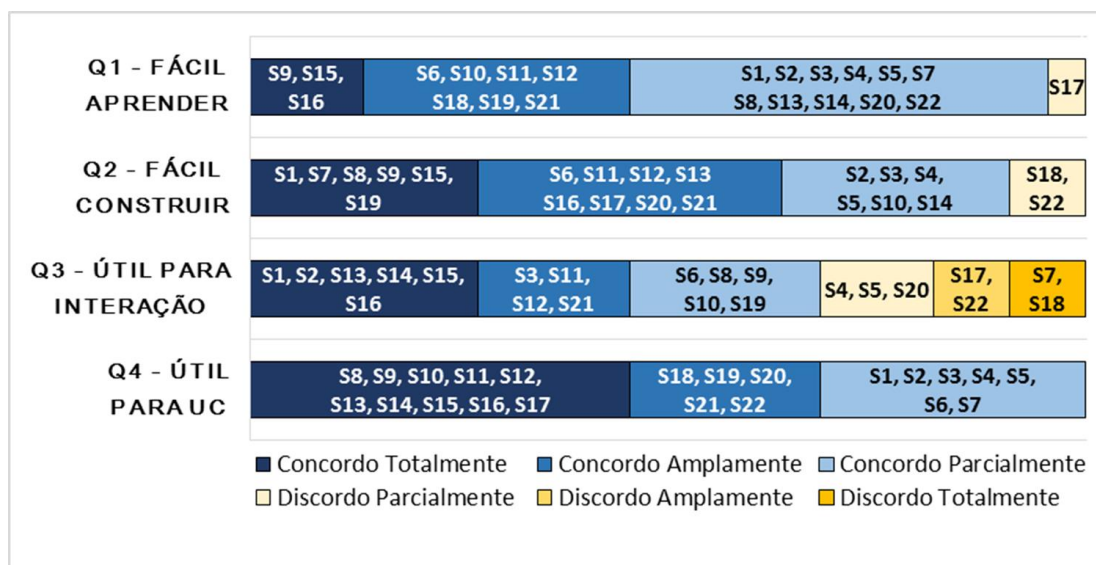


Figura 3.10: Percepção dos Participantes na utilização do UCModel.

Metade dos participantes (11) concordou parcialmente que o formato gráfico é fácil de aprender. Na análise qualitativa, alguns participantes evidenciaram que utilizar o conhecimento do diagrama de atividades facilitou o aprendizado do formato gráfico. No entanto, alguns participantes afirmaram que tiveram dificuldades em diferenciar a notação do formato gráfico (UCModel) do diagrama de atividades. Por exemplo, o participante S1 afirmou “...*consigo usar o conhecimento que tenho com diagramas de atividades, porém ainda confundo algumas restrições de sintaxe entre diagrama de atividade e UCModel*”. Além disso, os principais defeitos gerados pelos participantes utilizando o formato gráfico estão relacionados à Falha de Integridade e Incorretude.

Apenas o participante S17 não concordou que esse formato seja fácil de aprender, e ainda apresentou cinco defeitos, quatro de Falha de Integridade e um defeito de Falha de Legibilidade. Percebeu-se que este participante mencionou não ter dificuldades em aprender, “*Foi fácil, pois eu já sabia como especificar um caso de uso, se tivesse que aprender do zero usando UCModel seria muito mais difícil*”. No entanto, relatou que teve dificuldade em utilizar o formato gráfico – “*Onde inserir fluxos alternativos*” – corroborando a quantidade de omissão

de fluxos (Falha de Integridade) identificados no caso de uso. Desse modo, possuir um conhecimento preliminar sobre a especificação de casos de uso textual (formato textual) pode facilitar o aprendizado do formato gráfico, mas não diminui as dúvidas sobre como elaborar os fluxos alternativos utilizando este formato.

Quanto à facilidade em construir (Q2) oito participantes (S6, S11, S12, S13, S16, S17, S20, S21) concordaram amplamente que o formato gráfico é fácil de construir, porém estes participantes obtiveram 2 ou 3 defeitos de omissão de fluxos (Falha de Integridade) e um defeito de Incorretude (pré, pós-condições e eventos incompletos nos casos de uso). Na análise qualitativa a maioria dos participantes evidenciou não ter dificuldades em construir. No entanto, observou-se que alguns participantes apresentaram dificuldades em utilizar a sintaxe do formato gráfico. Isto sugere que esta dificuldade pode ter contribuído para a geração dos defeitos de omissão de fluxos identificadas nas especificações.

Os participantes S18 e S22 discordaram parcialmente que é fácil construir um caso de uso com o formato gráfico. Apesar dos participantes apresentarem níveis diferentes de conhecimento em modelagem de diagramas UML (S18 - médio e S22 - baixo) obtiveram a mesma quantidade de defeitos. Observou-se, que estes participantes mencionaram diferentes dificuldades como, confusão no entendimento da funcionalidade, modelagem dos fluxos alternativos usando o formato gráfico e diferenciar a notação entre o diagrama de atividades e UCMModel.

Na análise da utilidade do modelo para melhorar o entendimento da interação do caso de uso (Q3), dois participantes (S7 e S18) discordaram totalmente. Apesar dos participantes possuírem níveis de conhecimento diferentes em modelagem de diagramas UML (S7 – baixo e S18 – médio), os participantes relataram dificuldades relacionadas à sintaxe e elaboração dos fluxos alternativos do modelo (UCModel). Isto sugere que estas dificuldades podem ter influenciado a quantidade de defeitos geradas pelos participantes na elaboração do caso de uso.

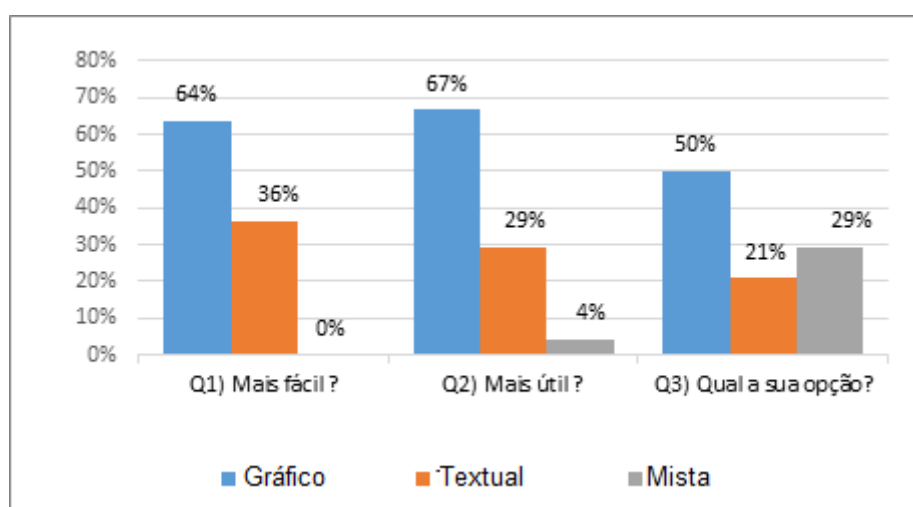
#### 3.4.1.2 *Percepção sobre a preferência dos formatos para especificar casos de uso*

Para uma análise comparativa sobre os formatos os participantes também responderam a três questões, apresentadas na Tabela 3.8.

**Tabela 3.8:** Preferências dos formatos para especificar UCs.

<b>Facilidade</b>	<b>(Q1)</b> Qual o formato mais fácil para especificar um caso de uso?
<b>Utilidade</b>	<b>(Q2)</b> Qual o formato mais útil para especificar um caso de uso?
<b>Empregabilidade</b>	<b>(Q3)</b> Para especificar casos de uso de uma aplicação, qual a sua opção?

A Figura 3.11 apresenta a análise sobre a percepção dos participantes, onde o eixo horizontal refere-se ao grau de preferência dos participantes sobre essas questões. Percebe-se que 64% (14 participantes) preferem utilizar o formato gráfico por ser mais fácil, 67% (16) por ser mais útil e 50% (11) o utilizariam em uma próxima aplicação. No entanto, 36% (8) preferem utilizar o formato textual por ser mais fácil, 29% (6) por ser mais útil e 21% (5) para especificar casos de uso em uma próxima aplicação. Além disso, 4% (1) e 29% (7) preferem utilizar os dois formatos, em conjunto.



**Figura 3.11:** Percepção dos participantes sobre a Preferência dos formatos para especificar UC.

### 3.4.2 Resultados Qualitativos

Para analisar os dados qualitativos sobre a percepção dos participantes (comentários adicionais) contidos nos questionários, utilizou-se alguns procedimentos de codificação (Strauss e Corbin 1998) com auxílio do software Atlas.ti<sup>1</sup>. O objetivo da análise qualitativa foi identificar as dificuldades encontradas pelos participantes no uso do formato gráfico. Além disso, analisaram-se os comentários sobre a preferência de uso dos formatos (textual e gráfico) para especificação.

Os dados qualitativos extraídos do questionário pós-modelagem foram analisados. Inicialmente, criaram-se conceitos (ou códigos) relevantes sobre a percepção dos participantes quanto aos formatos. Em seguida, os códigos foram analisados e foram criados relacionamentos entre eles. Além disso, categorias foram criadas e os códigos foram agrupados nestas categorias. O processo de codificação foi revisado por um pesquisador mais experiente. As citações e

<sup>1</sup> Atlas.ti – The Knowledge Workbench, Scientific Software Development – <http://www.atlasti.com>

códigos foram discutidos e, ao final (após quatro revisões), foram gerados 170 códigos associados a 11 categorias.

Para facilitar a visualização de todas as percepções obtidas sobre o formato gráfico (UCModel), as categorias (Facilidade, Utilidade, Empregabilidade e Dificuldades) e subcategorias foram incluídas na Tabela 3.9. A primeira coluna mostra a percepção sobre a **facilidade**, que está subdividida em aprender e construir. Na sequência apresenta a percepção de **utilidade**, que está subdividida em: melhorar a compreensão da interação e especificar casos de uso. Também, a percepção de **empregabilidade**, subdividida em: casos de uso e projetos de software. A segunda coluna apresenta as características positivas do formato gráfico (códigos), associadas a cada subcategoria. Por fim, a categoria **dificuldades** está subdividida nas subcategorias aprender, construir e utilizar.

**Tabela 3.9:** Características de Facilidade, Utilidade, Empregabilidade e Dificuldades do formato gráfico.

Percepção (Categorias e Subcategorias)	Características do formato gráfico - Citações
Facilidade	<p>Aprender</p> <p>“O modelo abrange todos os aspectos de uma boa especificação como declaração de atores, descrição dos eventos, regras etc”. - S21</p> <p>“É fácil especificar utilizando fluxos” - S19</p> <p>“A ideia é bem simples” - S12</p> <p>“O UCModel utiliza uma abordagem visual que ajuda a enxergar melhor o caso de uso” - S6</p> <p>“É uma maneira bem visual e natural” - S9</p>
	<p>Construir</p> <p>“Por causa da facilidade em estruturar o próprio diagrama” - S9</p> <p>“É fácil, pois a intimidade com os elementos do diagrama ajuda a visualizar melhor as ações” - S20</p> <p>“É fácil quando se consegue visualizar quais as ações que o sistema e o ator possuem” - S13</p> <p>“É fácil exibir as ações sistema-usuário, assim como reparar os fluxos alternativos” - S17</p> <p>“O esquema de diagramação é bem intuitivo e usualmente melhor” - S21</p> <p>“É bastante intuitivo em questão do fluxo das atividades” - S7</p> <p>“Estou acostumado com diagrama utilizando fluxos de atividade” - S19</p> <p>“É fácil, pois a intimidade com os elementos do diagrama ajuda a visualizar melhor as ações” - S20</p> <p>“Sim, muito mais fácil que utilizar a especificação em si, pois é mais visual e tem menos coisas para escrever” - S15</p> <p>“Por ser um diagrama as atividades ficam mais explícitas” - S1</p> <p>“Já conhecíamos algo parecido (Diagrama de Atividades)” - S5</p>

Percepção (Categorias e Subcategorias)		Características do formato gráfico - Citações
Utilidade	Melhorar a Compreensão da Interação	<p>“Sim, visto que em todo evento pode-se especificar o ator e sua ação” - S21</p> <p>“A ideia do Diagrama de Atividades junto com o caso de uso deixou mais clara a interação, devido as tags (apresentadas no UCMoel)” - S2</p> <p>“Ajudou na compreensão e na visualização geral do comportamento do sistema. A leitura feita a partir do diagrama se tornou mais simples e rápida” - S12</p> <p>“Sim, é mais de entender e analisar um processo modelado por esse artifício consequentemente você consegue perceber mais fácil essa interação” - S13</p> <p>“Da para ver claramente o diálogo entre eles.” - S3</p> <p>“Com os fluxos fica fácil visualizar a interação do ator com o sistema” - S11</p> <p>“Facilita a compreensão quando já pronto, por não ter que ter muitos textos” - S20</p>
	Especificar casos de uso	<p>“Foi útil em questão da agilidade em fazer, algo que eu demoraria 30 minutos, foi realizado em 20 (por exemplo)” - S5</p> <p>“Podemos perceber a interação e suas respectivas escolhas e regras” - S14</p> <p>“Como ele é bem dividido (Fluxo Principal, Alternativos) fica bem organizado” - S3</p> <p>“Achei melhor de visualizar do que uma especificação” - S15</p> <p>“Foi útil, pois ajuda a ter uma visualização melhor” - S17</p> <p>“Simplificou o modo de visualizar o problema” - S9</p> <p>“Muito útil, fica bem melhor para manipular e ler também” - S8</p> <p>“Sim, fica bem próximo do que vai ser implementado” - S13</p> <p>“Sim, foi útil para o entendimento geral do caso de uso” - S12</p>
Empregabilidade	Em Casos de Uso	<p>“Na demonstração ao cliente” - S7</p> <p>“Em casos de uso mais complexos ou maiores” - S9</p> <p>“Primeiro porque em termos de gráficos a compreensão aumenta e em texto torna-se cansativo as vezes” - S20</p> <p>“Onde for necessária uma melhor visualização do caso de uso” - S11</p> <p>“Quando precisa de uma modelagem bastante detalhada” - S17</p> <p>“Onde eu consigo perceber sem dificuldade as interações entre ator e sistema” - S13</p> <p>“Empregaria porque especificaria e já modelaria” - S16</p>
	Em Projetos de Software	<p>“Projetos grandes” - S3</p> <p>“Em qualquer projeto de pequeno e médio porte” - S21</p> <p>“Qualquer projeto que necessitasse especificar caso de uso” - S1</p> <p>“Em qualquer sistema que necessita da aplicação de casos de uso” - S22</p>
	Aprender	“Muitos diagramas para aprender, às vezes fica confuso” - S3

Percepção (Categorias e Subcategorias)		Características do formato gráfico - Citações
Dificuldades		<p>“Possui elementos únicos, causando confusão” - S5</p> <p>“Fiquei um pouco confuso quanto a usar diagrama de atividades para especificar o UC” - S20</p>
	Construir	<p>“Logo de início, há uma confusão entre funcionalidade (vem de casos de usos) ou atividade. Mas uma vez, identifica a funcionalidade do sistema, pode-se concluir o diagrama” - S22</p> <p>“Só a escrita que é dificultada, Vale ressaltar que é meio estanho perceber o momento de inserir um fluxo alternativo” - S18</p>
	Utilizar	<p>“As especificações no diagrama e ter um melhor entendimento do processo do caso de uso” - S2</p> <p>“Arranjar as posições das “atividades” de forma organizada, uma vez que foi feita a mão ou papel. Organizar os fluxos de forma correta com os componentes corretos” - S20</p> <p>“Confundi algumas coisas, como para onde voltar o fluxo alternativo, já que me lembra muito o diagrama de atividades” - S16</p> <p>“Onde inserir fluxos alternativos” - S17</p> <p>“Confusão com os outros diagramas ensinados” - S3</p> <p>“Pouca experiência com a sintaxe de UCModel, confundi algumas vezes com a do diagrama de atividades” - S1</p>

### 3.4.2.1 Preferência dos formatos (textual e gráfico) pela facilidade e utilidade

A Tabela 3.10 apresenta os códigos associadas a cada formato. Na primeira coluna, mostra os códigos relacionados à escolha do formato textual. Na segunda coluna, mostra os códigos relacionados à escolha do formato gráfico. E na terceira coluna, mostra os códigos relacionados a escolha de uma abordagem Mista (formato textual e gráfico).

**Tabela 3.10:** Preferência dos formatos: facilidade, utilidade e usar numa aplicação.



<b>Preferência pelo Formato</b>			
	<b>Textual</b>	<b>UCModel</b>	<b>Mista (As duas)</b>
<b>Facilidade</b>	<ul style="list-style-type: none"> <li>• Utiliza tópicos que facilitam a escrita e visualização</li> <li>• Devido a familiaridade</li> <li>• Pois a notação do UCModel é mais complexa</li> <li>• Pois suas regras de sintaxe são simples</li> <li>• Por ser mais simples</li> <li>• Pois é mais organizada e detalhada</li> <li>• UCModel possui uma sintaxe complexa</li> <li>• É melhor para entendimento da interação do ator com o sistema</li> <li>• Por ser mais fácil</li> </ul>	<ul style="list-style-type: none"> <li>• Pois a Tradicional é cansativa</li> <li>• Tendo o conhecimento das interações do UC</li> <li>• Porque é melhor para entender e visualizar a interação dos atores</li> <li>• Pois deixa o caso de uso mais estruturado</li> <li>• Pois é visual</li> <li>• Porque utiliza uma notação para melhor entendimento do UC</li> <li>• Porque tem uma descrição sucinta</li> <li>• Por utilizar uma notação que detalha o UC</li> <li>• Por ser uma especificação gráfica</li> <li>• Por ser um gráfico ficam visíveis seus fluxos</li> <li>• Por dar uma melhor compreensão do caso de uso</li> <li>• Pois mostra as descrições no diagrama</li> </ul>	
<b>Utilidade</b>	<ul style="list-style-type: none"> <li>• Não empregaria o UCModel prefere a especificação Tradicional</li> <li>• Por apresentar informações importantes</li> <li>• Por ser mais rápido de fazer</li> <li>• Pois é objetiva e detalhada</li> <li>• Pois é resumida</li> </ul>	<ul style="list-style-type: none"> <li>• Porque é mais fácil de entender</li> <li>• Por ser visualmente limpo</li> <li>• Pois detalha as ações dos atores</li> <li>• Pois é mais detalhado e melhora o entendimento</li> <li>• Pois mostra a especificação no diagrama</li> <li>• Pois mostra com detalhes as ações do caso de uso</li> <li>• Por ser uma especificação gráfica</li> <li>• Por ser de fácil compreensão</li> <li>• Por ser mais organizado</li> <li>• Porque é visual</li> <li>• Por utilizar elementos visuais (notação)</li> <li>• Pois dá uma compreensão melhor dos fluxos de forma mais claro</li> <li>• Pois dá uma visualização melhor do caso de uso do que a especificação textual</li> <li>• Pois ajuda na implementação do UC</li> <li>• Pois seu modelo mostra o UC de forma completa</li> </ul>	<ul style="list-style-type: none"> <li>• Utilizaria a Tradicional por ser mais completa e UCModel por visualizar melhor as interações do UC.</li> <li>• UCModel é mais útil por ser mais prático do que a abordagem Tradicional</li> </ul>
<b>Usar em uma Aplicação</b>	<ul style="list-style-type: none"> <li>• A forma Tradicional é melhor para o entendimento do caso de uso</li> <li>• Utilizaria a abordagem Tradicional, pois facilita a modelagem do analista para o problema bem detalhado</li> <li>• A forma Tradicional deixa a especificação mais organizada</li> <li>• A forma Tradicional deixa a especificação mais detalhada</li> <li>• Numa aplicação de pequeno porte utilizaria a abordagem Tradicional</li> </ul>	<ul style="list-style-type: none"> <li>• Por ser um modelo mais completo</li> <li>• Por ser de fácil utilização</li> <li>• Para facilitar o entendimento de quem utilizará a especificação</li> <li>• Por ser simples e útil como a abordagem Tradicional</li> <li>• Utilizaria UCModel para casos de uso com texto grande de difícil visualização</li> <li>• Pois mostra as especificações no diagrama</li> <li>• Por ser bem próxima a implementação</li> </ul>	<ul style="list-style-type: none"> <li>• Numa aplicação utilizaria UCModel para casos de uso pequenos</li> <li>• UCModel melhora a visualização do caso de uso</li> <li>• Numa aplicação utilizaria UCModel para UCs mais complexos e a Tradicional para UCs simples</li> <li>• UCModel retarda a escrita do caso de uso</li> <li>• As duas abordagens se completam, mas demanda tempo</li> </ul>

<b>Preferência pelo Formato</b>			
	<b>Textual</b>	<b>UCModel</b>	<b>Mista (As duas)</b>
<b>Opção de usar o Modelo</b>	<ul style="list-style-type: none"> <li>• A abordagem Tradicional é melhor para entendimento da interação do ator com o sistema</li> <li>• Utilizaria a abordagem Tradicional, pois é melhor para especificar</li> <li>• Não empregaria o UCModel prefere a especificação Tradicional</li> </ul>	<ul style="list-style-type: none"> <li>• UCModel permite ter uma visualização do caso de uso</li> <li>• Por ser agradável e prático</li> <li>• Porque gosto de diagrama de atividades</li> <li>• Por estar familiarizado com diagramas</li> </ul>	<ul style="list-style-type: none"> <li>• Primeiro usa a abordagem Tradicional para especificar depois UCModel para visualizar o UC</li> <li>• Utilizaria a Tradicional por ser mais completa e UCModel por visualizar melhor as interações do UC</li> </ul>

Os resultados qualitativos mostram que o formato gráfico baseado no UCModel (Massollar *et al.* 2012) pode auxiliar a melhorar a forma como as informações dos casos de uso são apresentadas. Além disso, pode contribuir para a leitura e compreensão geral do caso de uso facilitando sua realização, uma vez que esse formato utiliza elementos conhecidos pelos participantes. O formato gráfico contribuiu na identificação e visualização das principais funcionalidades e regras que serão implementadas a partir do caso de uso. No entanto, o formato gráfico apresentou algumas dificuldades relacionadas à legibilidade do diagrama, ou seja, a especificação gerada pode ficar visualmente grande e de difícil entendimento para os profissionais de software.

Em relação ao formato textual utilizado (Cockburn, 2005), os resultados mostram que esse formato é útil para detalhar e organizar os cenários, tornando-o mais simples para especificar o caso de uso. Os resultados deste estudo coincidem com os resultados encontrados por Bolloju *et al.* (2012), que também afirmam que o formato textual é mais adequado para mostrar o detalhamento dos passos e para demonstrar a interação dos atores no caso de uso. Contudo, foi observado no estudo que a leitura do caso de uso não é facilmente entendida pelos participantes, pois na maioria das vezes, a descrição não é sucinta e simples.

### **3.5 AMEAÇAS À VALIDADE**

Segundo Travassos *et al.* (2008), há quatro tipos da validade dos resultados do experimento: a validade interna, a validade externa, a validade de conclusão e a validade de

constructo. As principais ameaças relacionadas à validade do estudo realizado estão listadas nesta Seção.

### 3.5.1 Validade Interna

Neste estudo foram consideradas quatro principais ameaças que representam um risco de interpretação imprópria dos resultados: (a) efeitos de treinamento; (b) classificação de experiência dos participantes; (c) medição do tempo de especificação; (d) uso dos cenários; (e) uso do UCMModel; e (f) inclusão de itens no Checklist de Avaliação.

Em relação à primeira ameaça (a), poderá haver um efeito causado pelo treinamento, caso o treinamento em um dos formatos tivesse qualidade inferior ao do outro. Este risco foi controlado ministrando treinamentos equivalentes, incluindo atividades teóricas e exercícios práticos com os mesmos exemplos e os mesmos instrutores. Em relação à classificação da experiência dos participantes (b), ela teve por base o número e tipo de experiências anteriores (Em especificação de Casos de Uso e Modelagem de Diagramas UML). Em relação à medição do tempo de especificação (c), foi solicitado aos participantes que fossem mais precisos possível com relação à medição do tempo gasto nas tarefas de especificação. O moderador conferiu o tempo anotado por cada participante, no momento em que eles iniciaram e finalizaram a especificação.

Em relação ao uso dos cenários (d), estes poderiam afetar o estudo caso os participantes não entendessem o cenário. Esta ameaça foi reduzida utilizando cenários escritos em linguagem natural onde os requisitos e *mockups* (protótipos) deste cenário estavam explícitos, de forma similar aos exercícios realizados durante o treinamento. Com relação ao uso do UCMModel (e), este poderia afetar o estudo caso os participantes não entendessem a aplicação desse formato. Esta ameaça foi reduzida apresentando exemplos práticos usando o formato gráfico durante o treinamento. Além disso, os participantes realizaram exercícios práticos utilizando este formato.

Durante a avaliação dos casos de uso foi verificado que os itens do Checklist de Avaliação não possuíam alguns itens que cobrissem defeitos encontrados nas especificações (f). Por isso foram incluídos novos itens no Checklist para conduzir a verificação desses defeitos. Este risco foi controlado pela avaliação de dois pesquisadores especialistas que validaram a necessidade da inclusão dos novos itens ao Checklist.

### 3.5.2 Validade Externa

Três questões foram consideradas: (a) os participantes do estudo foram estudantes de graduação; (b) o estudo foi realizado em ambiente acadêmico; e, (c) validade dos cenários utilizados. Sobre a questão (a), poucos participantes possuíam experiência em aplicações na indústria, devido ao fato de serem estudantes de graduação. No entanto, o estudo foi realizado em uma disciplina, de Análise e Projeto de Sistemas (APS), que estava relacionada ao experimento, além disso, os participantes já tinham cursado disciplina introdutória, como Introdução a Engenharia de Software (ES). Em relação à questão (b), os artefatos utilizados (diagramas de casos de uso, requisitos do sistema e *mockups*) foram baseados em uma aplicação real o que pode ajudar a simular um ambiente real de desenvolvimento de software. Sobre a questão (c), não é possível afirmar que os requisitos e casos de uso utilizados representem todos os tipos de cenários. Esta é uma limitação dos resultados do estudo.

### 3.5.3 Validade de Conclusão

Neste estudo, o maior problema é o tamanho e a homogeneidade da amostra, por serem todos estudantes de uma mesma instituição. A quantidade de participantes também não é ideal do ponto de vista estatístico. Pequenas amostras são um problema conhecido em experimentos na área de Engenharia de Software e difíceis de serem superados (Conte *et al.*, 2007). Assim, há limitação nos resultados, sendo estes considerados indícios e não conclusivos.

### 3.5.4 Validade de Constructo

Neste tipo de ameaça, considerou-se a definição dos indicadores: (a) tempo gasto para especificar os casos de uso; e (b) corretude (número de defeitos) das especificações de casos de uso. Esses indicadores são semelhantes aos definidos por Cox *et al.* (2004) que avaliaram diferentes tipos de defeitos em casos de uso. Thelin *at al.* (2003) avaliaram a eficiência (número de defeitos/hora) e eficácia (número de defeitos/total de defeitos) comparando duas técnicas de leitura na identificação de defeitos em casos de uso. Anda e Sjøberg (2002) realizaram dois estudos para avaliar a eficiência (número de defeitos) comparando duas técnicas de inspeção em modelos de casos de uso.

## 3.6 CONSIDERAÇÕES FINAIS

Este capítulo apresentou resultados obtidos de um estudo experimental que comparou dois formatos (Textual e Gráfico) utilizados para especificar casos de uso. Para isso buscou-se verificar a facilidade e utilidade destes formatos. Os resultados deste estudo foram publicados

no artigo *“Is a Picture worth a Thousand Words? A Comparative Analysis of Using Textual and Graphical Approaches to Specify Use Cases”* (Nascimento *et al.* 2016).

A análise dos resultados quantitativos mostrou que não há diferença significativa entre os formatos em relação ao tempo necessário para especificar os casos de uso e a corretude obtida nas especificações. No entanto, a avaliação das especificações geradas mostrou que foram encontrados diferentes tipos de defeitos para os dois formatos. Os principais defeitos encontrados nas especificações geradas pelo formato gráfico foram de falha de integridade, incorretude e falha de legibilidade. Isso pode ter sido ocasionado devido ao fato dos participantes: (a) apresentarem dificuldades em modelar no diagrama os fluxos e regras de negócio do caso de uso; e (b) utilizarem um formato (diagrama) que possui alguns elementos diferentes do que estão familiarizados.

No que diz respeito ao formato textual os defeitos mais encontrados foram incorretude e falha de integridade. Isso pode ser justificado pelo fato dos participantes não terem utilizado corretamente os elementos desse formato, o que resultou em descrições incompletas, com omissão de passos/ fluxos/ regras, referências (regras/fluxos) incorretas ou a falta de referências nos passos.

Com relação à percepção dos participantes notou-se que ter conhecimento em diagramas de atividades da UML foi importante para facilitar o aprendizado e construção da especificação do caso de uso utilizando o formato gráfico. A principal característica quanto à facilidade em aprender o formato gráfico foi motivada por ser “visual”, possibilitando aos participantes uma melhor compreensão e entendimento dos casos de uso. A facilidade em construir especificações com esse formato também está relacionada com esta característica (a visualização). Apesar do formato gráfico apresentar defeitos de omissão e incorretude, este parece ser intuitivo para a construção do caso de uso, uma vez que os participantes já estão familiarizados com a modelagem em diagramas.

Houve pouca concordância sobre a utilidade do formato gráfico em relação à melhoria do entendimento da interação do caso de uso. Segundo os participantes, o formato textual e o diagrama de atividades deixam claro esse entendimento. No entanto, a maioria dos participantes (67%), concordou que o formato gráfico é útil para especificar casos de uso. As principais características quanto a esta utilidade foram: mostrar a interação e regras do caso de uso de forma visual, simplificar o problema e dar uma visão geral do caso de uso, melhorar a leitura e manipulação do caso de uso, facilitar a compreensão e entendimento do caso de uso.

Os motivos dos participantes que preferiram utilizar o formato gráfico em uma aplicação foram: por ser prático e mais fácil para visualizar o caso de uso (em forma de diagrama). E a preferência de utilizar o formato textual foi citada por quem prefere ter uma especificação mais detalhada e organizada, sendo melhor para entendimento do caso de uso.

Com isso, percebeu-se que cada formato pode ser útil para atender a diferentes situações ou necessidades de casos de uso e projetos de desenvolvimento de software. Por exemplo, o formato textual foi considerado mais detalhado para especificar o caso de uso e o formato gráfico, por ser visual, foi considerada melhor para a compreensão do caso de uso. E, com base nestes resultados, percebeu-se que existem dificuldades em utilizar o formato gráfico e textual. Por conta disso, o próximo capítulo apresenta um estudo qualitativo que buscou avaliar, de forma mais detalhada, quais as dificuldades encontradas por estudantes ao especificar casos de uso utilizando um formato textual.

## **CAPÍTULO 4 – ESTUDO QUALITATIVO SOBRE AS DIFICULDADES EM ESPECIFICAÇÃO DE CASOS DE USO**

*Este capítulo apresenta a Fase II (Investigativa) detalhando o planejamento, execução e resultados de um estudo qualitativo (2º estudo) realizado em duas universidades (Universidade Federal do Rio de Janeiro - UFRJ e Universidade Federal do Amazonas - UFAM) de diferentes regiões do país. Este estudo teve como objetivo explorar e entender as dificuldades em especificar UCs. Utilizaram-se entrevistas semiestruturadas com estudantes visando identificar estas dificuldades. A análise dos dados foi conduzida utilizando procedimentos do método Grounded Theory (GT). Como resultado, foi proposta uma versão inicial do modelo que apresenta as dificuldades encontradas para especificação de UC.*

### **4.1 INTRODUÇÃO**

No estudo anterior foram identificadas dificuldades percebidas pelos participantes durante a especificação do UC utilizando os formatos textual e gráfico. Estas dificuldades impactaram na qualidade das especificações geradas por estes formatos gerando defeitos relacionados, principalmente, a Falha de Integridade e Incorretude. Os defeitos de Falha de Integridade estavam relacionados à omissão de fluxos ou regras que não foram descritos no caso de uso. Os defeitos de Incorretude estavam relacionados a descrições incompletas, referências de fluxos alternativos, regras e outros casos de uso que não foram referenciados na descrição do caso de uso.

Portanto, para melhor compreender como estas dificuldades ocorrem durante a especificação de UC utilizando um formato textual, foi realizado um estudo qualitativo que tem como objetivo responder a seguinte questão de pesquisa: “*Quais as dificuldades percebidas ao especificar casos de uso durante o desenvolvimento de software?*”. Para responder esta questão de pesquisa, realizaram-se entrevistas semiestruturadas com 24 estudantes (de graduação e pós-graduação) que possuíam experiência prévia em especificação de Casos de Uso. A análise dos dados obtidos foi feita qualitativamente empregando procedimentos do método de Grounded Theory (GT) (Strauss e Corbin, 1998). Com base nos resultados, foram identificadas as dificuldades percebidas pelos estudantes ao especificar UC e foi desenvolvido um modelo que tem por objetivo sumarizar estas dificuldades.

Este capítulo está organizado da seguinte forma: a Seção 4.2 descreve o método de pesquisa, o planejamento, a condução e execução do estudo qualitativo. A Seção 4.3 apresenta os resultados e a versão inicial do modelo sobre dificuldades para especificar UC. A Seção 4.4 apresenta as discussões dos resultados deste estudo. A Seção 4.5 apresenta as principais ameaças à validade do estudo. Por fim, a Seção 4.6 apresenta as considerações finais do capítulo.

## 4.2 MÉTODO DE PESQUISA

Um estudo qualitativo foi realizado com o objetivo de identificar quais as dificuldades percebidas pelos estudantes ao especificar UC. Para realizar a coleta dos dados qualitativos, foram utilizadas entrevistas semiestruturadas. Segundo Manotas *et al.* (2016), as entrevistas são úteis para reunir um conjunto de observações e percepções qualitativas dos participantes. Além disso, entrevistas ajudam a obter uma compreensão ampla do contexto em que os entrevistados atuam (Manotas *et al.* 2016).

### 4.2.1 Planejamento

As perguntas incluídas no roteiro da entrevista foram desenvolvidas colaborativamente por dois pesquisadores e revisadas por outros três pesquisadores. Após a revisão, realizou-se um estudo piloto a fim de avaliar a clareza das perguntas, a estrutura e o tempo total da entrevista. O resultado do estudo piloto mostrou que não eram necessárias alterações importantes no roteiro da entrevista (ver APÊNDICE I). A Tabela 4.1 apresenta o roteiro utilizado durante a condução da entrevista.

**Tabela 4.1:** Roteiro Semiestruturado.

<b>Parte I – Background do Participante</b>
Q1 - Você já participou de projetos de desenvolvimento de software (na indústria ou na academia)?
Q2 - Você já participou de projetos de desenvolvimento de software que utilizaram casos de uso (na indústria ou academia)?
<b>Parte II – Especificação de Casos de Uso</b>
Q3 - Qual a sua percepção sobre a utilização da especificação de casos de uso durante o desenvolvimento de software?
<b>Parte III – Dificuldades em Descrever Casos de Uso</b>
Q4 - Você tem/teve alguma dificuldade para especificar um caso de uso?

Conforme visto na Tabela 4.1, as perguntas foram apresentadas em formato de funil, começando com questões gerais e avançando para questões mais específicas, conforme sugerido por Runeson e Host (2008).



#### 4.2.2 Participante e Contexto do Estudo

O estudo teve como participantes estudantes de graduação e pós-graduação de duas universidades (Universidade Federal do Rio de Janeiro - UFRJ e Universidade Federal do Amazonas - UFAM) de diferentes regiões do país. Todos os alunos tinham cursado anteriormente a disciplina de Engenharia de Software, ou seja, já tinham uma experiência prévia com a especificação de UC. Ao todo, foram entrevistados vinte e quatro participantes, conforme a Tabela 4.2.

**Tabela 4.2:** Resumo dos Participantes do Estudo.

Un.	UFRJ											UFAM												
Par.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20	P21	P22	P23	P24
Per.	A	A	A	I	A	I	I	I	I	I	A	I	A	I	A	A	A	A	A	A	I	I	A	A
Nív.	G	G	G	PG	G	PG	PG	PG	PG	PG	PG	G	G	G	G	G	G	G	G	G	G	G	G	G
<b>Legenda:</b> <b>Par.</b> – Código do Participante; <b>Per.</b> – Perfil do participante: Experiência em Desenvolvimento de Software na <u>Indústria</u> (I) ou na <u>Academia</u> (A); <b>Nív.</b> – Nível de Escolaridade do Participante: <u>G</u> – <u>Graduação</u> ; <u>PG</u> – <u>Pós-Graduação</u> ;																								

Na UFRJ, foram entrevistados onze participantes: quatro eram alunos de graduação, cursavam a disciplina de Desenvolvimento de Software Orientado a Objetos e estavam projetando um sistema Web, cuja especificação de requisitos era baseada em UC, em um ciclo de desenvolvimento contínuo; sete eram alunos de pós-graduação (mestrado e doutorado) e também profissionais de software e já haviam desenvolvido mais de dois projetos de software na indústria que utilizavam UCs.

Na UFAM, foram entrevistados treze alunos de graduação que estavam cursando a disciplina de Análise e Projeto de Sistemas. Quatro dos treze participantes já eram profissionais de software, ou seja, já tinham utilizado UC na indústria. Todos os participantes da UFAM estavam projetando sistemas Web e Mobile baseados em UC.

#### 4.2.3 Coleta de Dados

A fim de atender às exigências éticas da pesquisa, foi explicada a cada participante a finalidade da pesquisa e os seus direitos, através de um Termo de Consentimento Livre e Esclarecido (TCLE), garantindo a confidencialidade dos dados fornecidos e o anonimato do participante. Além disso, as entrevistas foram realizadas individualmente, dentro do tempo previsto (entre quinze e trinta minutos). Assim, os pesquisadores não apressaram a entrevista e o entrevistado pôde se expressar de forma tranquila e sem interrupções.

A coleta e análise dos dados foram realizadas em dois momentos: (i) no primeiro momento, todos os participantes da UFRJ foram entrevistados, as entrevistas foram gravadas e transcritas e o processo de análise dos dados foi iniciado; (ii) no segundo momento, os

participantes da UFAM foram entrevistados, depois todo o processo de análise foi realizado novamente, conforme executado no primeiro momento. Para a análise dos dados, todas as entrevistas foram transcritas e importadas para o Atlas.ti<sup>2</sup>. Durante este processo não houve nenhuma evolução nos instrumentos e procedimentos para a coleta e/ou análise dos dados.

#### **4.2.4 Análise de Dados**

Utilizou-se procedimentos de codificação do método Grounded Theory (GT) para realizar a análise dos dados. GT é um método de pesquisa qualitativa que utiliza um conjunto de procedimentos sistemáticos para criar e avaliar teorias substantivas através das fases de Codificação Aberta, Axial e Seletiva (Strauss e Corbin, 1998). De acordo com Strauss e Corbin (1998), o pesquisador pode usar apenas alguns dos seus procedimentos para atender seus objetivos de pesquisa.

Os dados qualitativos foram analisados utilizando um subconjunto das fases do processo sugerido por Strauss e Corbin (1998): codificação aberta (1ª fase) e codificação axial (2ª fase). Na 1ª fase, os códigos foram criados a partir da análise dos dados qualitativos. Em seguida, os códigos foram agrupados de acordo com suas propriedades e dimensões, criando conceitos que representam categorias e subcategorias. Na 2ª fase, os relacionamentos entre os códigos foram feitos. A 3ª fase é a codificação seletiva, que tem por objetivo integrar a teoria, identificando uma teoria central na qual todas as outras categorias devem ser relacionadas. No entanto, a 3ª fase não foi realizada, pois a saturação teórica ainda não foi alcançada (Strauss e Corbin, 1998).

Após realizar a análise, outro pesquisador com bom nível de conhecimento em análise qualitativa verificou os códigos, as categorias e subcategorias criadas e as associações entre os códigos. Todo o processo de codificação foi revisado por este pesquisador, a fim avaliar os resultados obtidos na análise dos dados.

### **4.3 RESULTADOS DO ESTUDO**

Nesta seção são apresentados os resultados da percepção dos participantes com relação às principais dificuldades que os participantes relataram ao especificar um UC, ao modelo proposto de dificuldades em especificar UC resultante, além dos benefícios percebidos da utilização da especificação de UC.

---

<sup>2</sup> Atlas.ti – The Knowledge Workbench, Scientific Software Development – <http://www.atlasti.com>

### 4.3.1 Dificuldades em Especificar Casos de Uso

A partir da análise dos dados qualitativos, identificaram-se cinco categorias de tipos de dificuldades: Abstração do Requisito, Pré e Pós-Condição, Regras de Negócio, Fluxos dos Casos de Uso, e Referências a Fluxos e Regras de Negócio. Estas categorias serão apresentadas a seguir.

Conforme apresentado na seção anterior, a categoria “**Abstração do Requisito**” emergiu a partir da análise qualitativa, assim como para todas as outras categorias. Nesta categoria foram associados os códigos que mencionam dificuldades em abstrair o requisito durante a especificação do UC. A Tabela 4.3 apresenta os códigos relacionados a esta categoria e as citações dos participantes.

**Tabela 4.3:** Dificuldades relacionadas à Abstração do Requisito.

<b>Códigos (Abstração do Requisito)</b>	<b>Citações</b>
<b>“Entender o domínio do problema”</b>	<p>“Eu acho que a maior dificuldade é o entendimento do problema, mas a dificuldade de escrever (casos de uso) não muito.” – P5</p> <p>“O problema não foi a transcrição para o caso de uso, mas o problema (em si) que não ficou muito claro pra mim.” – P12</p>
<b>“Identificar todas as funcionalidades do sistema e documentá-las”</b>	<p>“Acho que (identificar) algumas funcionalidades pode ter faltado (...) A gente percebeu, esqueceu e eu percebi agora de novo.” – P16</p> <p>“Faltou o caso de uso atender a todos os cenários (...) e percebemos que foi uma questão de omissão mesmo, ou seja, não ter (os cenários) no caso de uso.” – P2</p>

Apesar do participante P12 possuir experiência na indústria e o participante P5 não possuir experiência (ver Tabela 4.2), ambos tiveram dificuldades em entender o domínio do problema para especificar o UC. Observou-se que independente da experiência do participante, compreender os requisitos era fundamental para especificar as informações do UC. Para o segundo código (Identificar todas as funcionalidades do sistema e documentá-las), percebeu-se que devido a pouca experiência prática em especificar UCs, os participantes P16 e P2 (alunos de graduação e sem experiência na indústria) tiveram dificuldades para identificar algumas funcionalidades necessárias para o sistema, logo, omitiam alguns fluxos (cenários) na especificação do UC.

Além de dificuldades com a abstração do problema, os participantes também relataram problemas com elementos da sintaxe do UC. Certas dificuldades citadas estão relacionadas à descrição da pré e pós-condição no UC. Foi então criada a categoria “**Pré e Pós-Condição**”, cujos códigos são apresentados na Tabela 4.4.

**Tabela 4.4:** Dificuldades relacionadas à Pré e Pós-Condição.

<b>Códigos (Pré e Pós-Condição)</b>	<b>Citações</b>
<b>“Definir a Pré-condição do UC”</b>	<p>“A primeira dificuldade foi mais na pré-condição (...) eu colocava algumas coisas que não faziam sentido.” – P15</p> <p>“A dificuldade que eu tive logo quando eu fui fazer a primeira especificação era com relação as pré-condições, as pós-condições, o fluxo principal.” – P14</p>
<b>“Definir a Pós-condição do UC”</b>	<p>“O que eu ficava com dificuldade era aquele negócio de pós-condição depois que tinha feito o caso de uso, depois que aconteceu aquele comportamento, eu tinha que pensar nas pós-condições daquilo então até hoje eu sinto muita dificuldade.” – P23</p>

Sobre as dificuldades na Pré e Pós-Condição, os resultados qualitativos mostram que apesar dos participantes P14, P15 e P23 possuírem perfis diferentes (P14 – com experiência na indústria; P15 e P23 – sem experiência), estes participantes tiveram dificuldades em descrever/entender quais informações eram adequadas para definir o que ficaria na pré e pós-condição do UC. Por exemplo, durante a especificação do UC, são descritas informações que podem não estar relacionadas a pré-condição do UC, como Regras de Negócio (RN) ou Requisitos Não-Funcionais (RNF).

Também foram mencionadas dificuldades em identificar e descrever as RNs associadas ao UC. Por esta razão, foi criada a categoria “Regras de Negócio”, com os códigos apresentados na Tabela 4.5.

**Tabela 4.5:** Dificuldades relacionadas às Regras de Negócio.

<b>Códigos (Regras de Negócio)</b>	<b>Citações</b>
<b>“Identificar as Regras de Negócio para o UC”</b>	<p>“Eu tento imaginar as regras de negócio do caso de uso (...) Por exemplo validar o login a senha, o cpf ou alguma coisa que pode ter mais de tantas letras. Imaginar as regras de negócio eu acho que foi isso, imaginar as restrições do caso de uso na hora.” – P14</p> <p>“As regras de negócio ficam meio intrínsecas, você não consegue separar ou tem dificuldade para separar e identificar uma regra de negócio que está relacionada aquele caso de uso (...) Talvez havia uma necessidade de identificar qual era a regra de negócio que estava relacionada ao caso de uso (...)” – P6</p>
<b>“Confundir as RNs com Requisitos Não-Funcionais”</b>	<p>“Particularmente eu confundo requisitos não funcionais com regras de negócio.” – P1</p> <p>“(…) às vezes você fica meio: o que é regra de negócio é requisito não funcional. Então quando você vê pela primeira vez você confunde onde vai localizar aquela regra. É funcional ou não funcional?” – P3</p>

Códigos (Regras de Negócio)	Citações
<p><b>“Confundir as RNs com os Fluxos do UC”</b></p>	<p><i>“Muitas vezes eu confundo um pouco fluxos e regras, porque quando eu olho, isso daqui faz parte do fluxo principal, mas também parece uma regra de negócio, então eu confundo às vezes... eu tento imaginar um pouco a situação para tentar abstrair o que é o que.” – P20</i></p>

Para o primeiro código, os participantes P14 e P6 mencionaram que tiveram dificuldades em identificar as regras de negócio do UC. Isto pode estar relacionado ao fato de que, durante a especificação do UC, nem todas as informações, como RNs, estão descritas ou claras na descrição dos requisitos do sistema. Para o segundo código, os participantes P1 e P3 confundiram as regras de negócio com os requisitos funcionais ou RNF do sistema. Esta dificuldade pode ter ocorrido devido os participantes ainda possuírem dúvidas sobre estes conceitos (RN e RNF) durante a especificação do UC. Para o terceiro código, o participante P20 afirmou que confundiu a descrição dos fluxos do UC com a descrição das RNs.

Os participantes mencionaram dificuldades em identificar e/ou descrever os fluxos ao UC. A Tabela 4.6 apresenta os códigos relacionados à categoria **“Fluxos do Caso de Uso”** e algumas das citações associadas.

**Tabela 4.6:** Dificuldades relacionadas aos Fluxos do Caso de Uso.

Códigos (Fluxos do UC)	Citações
<p><b>“Identificar e Descrever os Fluxos no UC”</b></p>	<p><i>“Inicialmente há a dificuldade de escrever e identificar quais fluxos alternativos serão descritos.” – P16</i></p> <p><i>“Eu não pensava muito no erro do sistema, então era mais na parte certa eu não pensava em erros, só na parte certa (...) no caminho certo, então eu errava muito nisso (...) Eu não sabia muito bem o que era fluxo alternativo, aí eu precisei pesquisar e ver o que era fluxo alternativo (...)” – P19</i></p> <p><i>“Isso de primeira foi complicado, organizar. Esse daqui é meu fluxo principal e esse daqui é meu alternativo (...). Eu não conseguia ver muito bem o que ia ser o alternativo, ou quando ia dar erro (no caso de uso).” – P20</i></p>
<p><b>“Descrever a interação entre ator e sistema para os Fluxos do UC”</b></p>	<p><i>“A maior dificuldade foi definir como seria a interação entre o sistema e o ator, eu acho que não tem algo específico (...) para todos os fluxos.” – P15</i></p>
<p><b>“Confundir as RNs com os Fluxos do UC” “Descrever a interação de outros sistemas que interagem com o UC”</b></p>	<p><i>“(…) quando tem sistemas externos interagindo com o sistema aí eu acho meio difícil de colocar isso. Porque quando eu estou especificando eu não sei que tipo de resposta aquilo vai me dar (...) Saber como ele ia devolver e saber como eu ia tratar isso no meu caso de uso.” – P22</i></p>

Para o primeiro código, os participantes que mencionaram a dificuldade na identificação e descrição dos fluxos (principal e alternativos) foram alunos de graduação e que não possuem experiência na indústria. Além disso, percebeu-se que o participante P16 afirmou anteriormente que teve dificuldade em abstrair as funcionalidades do sistema. Esta dificuldade pode ter influenciado na dificuldade em identificar quais seriam os fluxos alternativos da especificação do UC. O segundo e o terceiro códigos apresentam dificuldades mais específicas. No segundo código, o participante P15 mencionou que teve dificuldade em descrever a interação do ator e sistema nos fluxos do UC. No terceiro código, apesar do participante P22 possuir experiência em UCs na indústria, o participante mencionou que teve dificuldade em descrever a interação do sistema com outros sistemas que interagem com o UC.

Para a categoria “**Referências de Fluxos e Regras de Negócio**”, foram associados os códigos que mencionam dificuldades em referenciar os fluxos e RN nos passos dos fluxos do UC. A Tabela 4.7 apresenta os códigos e algumas citações relacionadas.

**Tabela 4.7:** Dificuldades relacionadas a Referenciar os Fluxos e RNs nos passos do Caso de Uso.

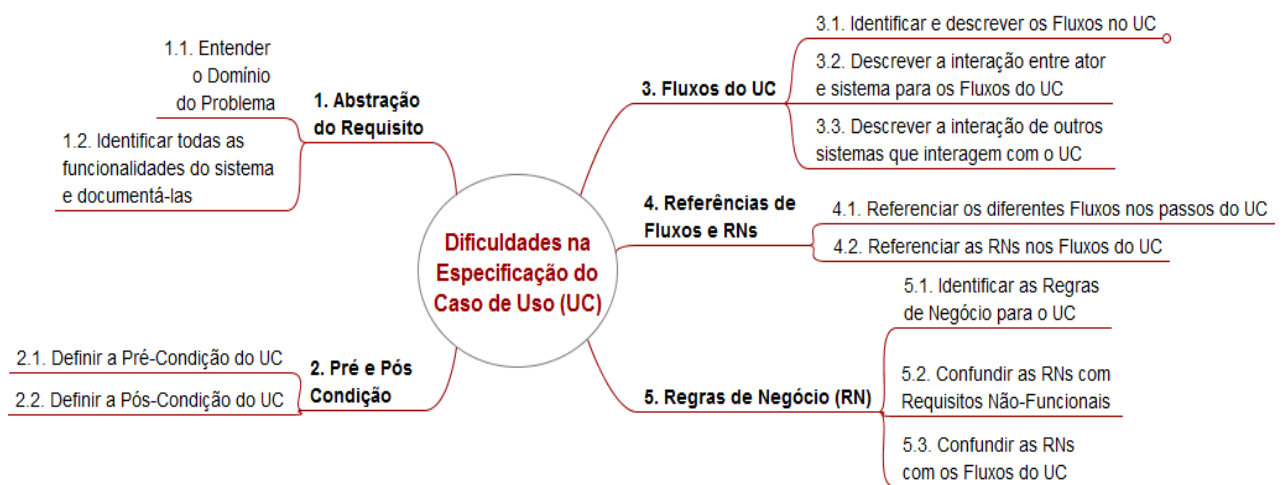
<b>Códigos (Referenciar os Fluxos e RNs)</b>	<b>Citações</b>
<b>“Referenciar os diferentes Fluxos nos passos do UC”</b>	<p><i>“(…) os membros da equipe esqueceram de incluir o fluxo alternativo e esqueceram de linkar.” – P13</i></p> <p><i>“Eu fico com dúvida. Ah! Mas o fluxo alternativo vai vir na hora que o ator informa ou na hora que o sistema válida? Então eu acho que a minha maior dificuldade é onde chamar o fluxo alternativo.” – P23</i></p>
<b>“Confundir as RNs com Requisitos Não-Funcionais”</b>	<p><i>“(…) questões de regras de negócio, às vezes eram mal explicadas ou não deveriam estar naquela subseção (de regras de negócio do UC).” – P10</i></p>

Sobre as dificuldades relacionadas ao primeiro código, os participantes P13 e P23 tiveram dificuldades em referenciar os fluxos (principal e ou/alternativos) nos passos do UC e por conta disso ocorreram dúvidas em saber onde referenciar o fluxo alternativo nos passos do UC. No segundo código, o participante P10 teve dificuldade em referenciar as RNs nos passos dos fluxos principal e/ou alternativos do UC.

Com base nos resultados encontrados na análise qualitativa, foi elaborada uma representação gráfica (Figura 4.1), apresentando a primeira versão do modelo com as dificuldades percebidas durante o processo de especificação de UC. Logo, as cinco categorias definidas no modelo foram extraídas da análise qualitativa. O processo de construção do modelo foi feito em duas etapas. Primeiro extraiu-se as categorias do modelo em seguida os códigos (ou as dificuldades) foram relacionados a essas categorias, ou seja, os códigos foram agrupados

em suas categorias correspondentes. Por exemplo, o código “Entender o domínio do Problema” foi associado à categoria “Abstração do Requisito”, e assim foi feito para todas as categorias.

A partir do modelo observou-se que podem existir possíveis relações entre as dificuldades apresentadas, discutidas na Seção 4.4. O modelo tem por objetivo: (a) auxiliar alunos e profissionais de software que utilizam UC na academia e/ou na prática na indústria; (b) indicar pontos de melhoria em diretrizes, técnicas e abordagens que auxiliam profissionais de software na especificação de UC; e (c) auxiliar os pesquisadores na construção de técnicas voltadas para o ensino da especificação de UC.



**Figura 4.1:** Modelo de Dificuldades sobre Especificação de Casos de Uso.

### 4.3.2 Benefícios em Utilizar a Especificação Casos de Uso

Além de relatar suas dificuldades, os participantes também comentaram quais suas percepções sobre os benefícios de Casos de Uso. Foi então criada uma categoria à parte, chamada “Benefícios em Utilizar UC no Desenvolvimento de Software”. A Tabela 4.8 apresenta seus códigos e exemplos de citações relacionadas.

**Tabela 4.8:** Benefícios em Utilizar UC no Desenvolvimento de Software.

Códigos (Benefícios em Utilizar UC)	Citações
<p><i>“Ajuda a descrever o comportamento e ver o que vai ser desenvolvido para o caso de uso”</i></p>	<p><i>“Porque você tem (...) uma visão geral do caso de uso, você tem os fluxos principais, os fluxos alternativos, as exceções, então é um detalhamento, que ao meu ver é necessário para o desenvolvimento.” – P10</i></p> <p><i>“A especificação do caso de uso (em si) chega mais perto do software (...) pois mostra o passo a passo, então nesse ponto eu acho que ela ajuda a (...) pensar na sequência de passos para alcançar o resultado do software” – P12</i></p>

Códigos (Benefícios em Utilizar UC)	Citações
<p><i>“É importante para todos da equipe visualizarem o que foi pensado para o sistema”</i></p>	<p><i>“(...) ajuda, por exemplo, a estruturar a ideia dos requisitos, de forma que o time de desenvolvimento consiga realmente entender a ideia...” – P7</i></p> <p><i>“...a descrição do caso de uso serve para o desenvolvedor ou para a equipe de desenvolvimento mostrar, como um todo, como vão desenvolver.” – P23</i></p>

No primeiro código, os participantes P10 e P12, por possuírem experiência na indústria, mencionaram que a especificação de UC é necessária, pois detalha os fluxos (principal, alternativos e exceções). Além disso, afirmaram que a especificação do UC é útil para detalhar a interação do ator com o sistema, pois descreve uma sequência lógica dos passos que serão desenvolvidos e assim alcançar o resultado esperado. No segundo código, os participantes P7 e P23 mencionaram que a especificação do UC ajuda a equipe de projeto a entender o que será desenvolvido para uma funcionalidade específica do sistema, pois apresenta de forma estruturada toda a ideia do UC.

#### 4.4 DISCUSSÃO DOS RESULTADOS

Nesta seção são discutidos os resultados obtidos baseado nas categorias resultantes da análise apresentada na seção anterior. Além disso, são apresentadas sugestões para melhorar o aprendizado e a prática de UC, com base nas dificuldades definidas para o modelo.

Os benefícios percebidos da utilização da especificação de UC no desenvolvimento de software estão relacionados ao detalhamento do comportamento do UC (através dos fluxos principal, alternativos e exceções). A especificação de UC ajuda a estruturar a ideia do UC, facilitando o entendimento da funcionalidade que será desenvolvida pela equipe de projeto. Esses resultados são similares aos de Nascimento *et al.* (2016), pois evidenciam que a especificação do UC foi útil para detalhar e organizar os cenários do UC.

As dificuldades relacionadas à categoria “**Abstração do Requisito**” somente ocorreram quando o participante iniciou o processo de especificação do UC, ou seja, quando o aluno precisou, por exemplo, identificar quais os fluxos e/ou RNs estavam relacionadas e descritas no UC. Assim, estas dificuldades podem afetar outras categorias do modelo, como “**Fluxos do UC**” e “**Regras de Negócio**”, uma vez que é necessário que os requisitos do sistema sejam compreendidos, a fim de transcrevê-los corretamente na especificação do UC. Esses resultados são semelhantes aos de Nascimento *et al.* (2016), pois evidenciam que os participantes tiveram dificuldades para entender o requisito e seguir os passos corretamente para a especificação do



UC. Além disso, no contexto real, a definição correta dos fluxos e das RNs é necessária para a completude da especificação do UC.

Na categoria “**Pré e Pós-Condição**”, as dificuldades ocorreram quando os participantes estavam definindo as pré e/ou pós-condições do UC, ou seja, os participantes incluíam informações como RNs ou requisitos não-funcionais que não estão relacionados a pré e pós-condição. Para minimizar e corrigir este tipo de dificuldade é preciso mostrar exemplos de pré e pós-condição mais claros, ou seja, exemplos onde os alunos consigam perceber a importância das pré e pós-condições e como que estas informações impactam na especificação do software.

Na categoria “**Fluxos do UC**”, os participantes tiveram dúvidas quando estavam organizando e descrevendo os fluxos (principal e alternativos) para o UC. Estes resultados são semelhantes às dificuldades apresentadas em Bolloju (2006) e Nascimento *et al.* (2016). Bolloju (2006) afirma que houve dificuldades em identificar e descrever os fluxos. No estudo de Nascimento *et al.* (2016), os resultados mostraram que ocorreram dificuldades em identificar e descrever os fluxos (alternativos ou exceções) no UC. Nesta categoria, também ocorreram dificuldades mais específicas, como descrever os passos da interação do ator e sistema nestes fluxos e descrever os passos da interação de outros sistemas que interagem com o UC. As dificuldades desta categoria podem impactar no detalhamento do comportamento do UC, pois a organização e descrição dos fluxos são importantes para obter a compreensão de toda a funcionalidade. Portanto, é necessário que os alunos pratiquem a atividade de especificação de UC, com retorno do professor sempre que possível, para decidir com clareza quais fluxos (principal e alternativos) serão descritos no UC.

Na categoria “**Referências de Fluxos e RNs**”, as dificuldades ocorreram após a definição e descrição dos fluxos. Por exemplo, os participantes tiveram dúvidas em identificar quais fluxos e RN seriam referenciados nos passos dos fluxos descritos no UC. Na categoria “**Regras de Negócio**”, as dificuldades ocorreram quando os participantes foram descrever as RNs no UC. Por exemplo, os alunos confundiram a descrição das RNs com a descrição dos requisitos não-funcionais ou confundiram com a descrição dos fluxos do UC. Esta confusão entre RNs e RNF pode estar relacionada ao fato do aluno ainda possuir dúvidas sobre restrições do sistema, uma vez que uma RN foca nas restrições do domínio do problema e o RNF está relacionado em como os atributos de qualidade atendem às funcionalidades que serão exibidas pelo sistema.

Como forma de minimizar ou evitar as dificuldades descritas, sugere-se a adoção de algumas práticas recomendadas pela literatura, como forma de melhorar a aprendizagem dos

alunos. A primeira prática sugerida é a inclusão de inspeção durante as aulas, por ser um meio eficiente que auxilia na identificação de defeitos e também ajuda a melhorar a qualidade dos documentos de software (Anda e Sjøberg, 2002). A inclusão da inspeção auxilia os alunos a relembrar os conceitos ensinados sobre UCs, a compreender os principais tipos de defeitos nas especificações e a evitar que estes sejam repetidos em outras modelagens. Outra prática que pode ser adotada é ensinar utilizando anti-padrões (Balaban *et al.* 2015), ou seja, apresentar aos alunos más soluções de modelagem. Assim como na inspeção, adotar anti-padrões durante as aulas sobre UCs melhora a capacidade dos alunos de identificar os problemas de modelagem, entender a forma errada para solucionar determinado cenário e desenvolver soluções corretas. Outra prática que pode ser utilizada é especificar uma funcionalidade de um software já desenvolvido (Ali, 2005; Pavlov *et al.* 2007). Desta forma, os alunos conseguem perceber todos os aspectos que devem ser especificados em um UC. A adoção desta prática pode auxiliar os alunos a visualizar melhor os vários cenários que compõem um UC e a entender na prática a importância de alguns conceitos, por exemplo, as pré e pós-condições e as regras de negócio.

Espera-se que a identificação das categorias de dificuldades apoie o processo de ensino-aprendizagem, além de motivar a melhoria de técnicas, métodos e abordagens no ensino/aprendizagem da especificação de UC, para que os alunos possam desenvolver diferentes habilidades, como abstração de diferentes contextos de sistemas e na escrita da especificação do UC. Assim, as dúvidas e dificuldades relacionadas às categorias do modelo poderão ser reduzidas e corrigidas, preparando os alunos para enfrentarem os desafios no desenvolvimento de software com UC na indústria.

## **4.5 AMEAÇAS À VALIDADE**

Neste estudo, existiram algumas ameaças que podem afetar a validade dos resultados. Nesta seção, são discutidas as principais ameaças que podem ser classificadas em: validade interna, validade externa, validade de constructo e confiabilidade (Wöhlin *et al.* 2012; Petersen e Gencel, 2013).

### **4.5.1 Validade Interna**

Neste estudo foram consideradas duas principais ameaças que representam um risco de interpretação imprópria dos resultados: (I) influência e experiência dos moderadores; e (II) participantes não relatarem dificuldades. Em relação à ameaça (I), poderá ocorrer um efeito diferente nos resultados das entrevistas, pois estas foram individuais e conduzidas por dois pesquisadores separadamente. Contudo, o segundo pesquisador acompanhou a condução da

entrevista piloto juntamente com o primeiro autor. Esse acompanhamento permitiu que as conduções das entrevistas fossem realizadas de modo semelhante.

Em relação à ameaça (II), para evitar que o participante não relatasse nenhuma dificuldade, os pesquisadores solicitavam que o participante exemplificasse alguma dúvida que teve durante a especificação de UC. Isto foi possível, pois os participantes estavam usando UC em trabalhos práticos desenvolvidos durante a disciplina ou em projetos de software que estavam participando e/ou já haviam atuado.

#### **4.5.2 Validade Externa**

Neste estudo, considerou-se como ameaça a comparação com outros contextos, devido existirem fatores relacionados ao background dos participantes e a grade curricular dos cursos nas universidades. No entanto, os diferentes perfis dos participantes ajudaram a capturar diferentes visões sobre as dificuldades em especificação de UCs. E os cursos apresentam disciplinas semelhantes nas quais os participantes desenvolveram projetos de software com UCs.

#### **4.5.3 Validade de Constructo**

Neste tipo de ameaça, considerou-se o roteiro de entrevistas induzirem o participante a ter resultados favoráveis ao pesquisador. Contudo, o roteiro de entrevista era semiestruturado e, portanto, continha perguntas abertas para capturar as dificuldades na especificação de UC. Para minimizar esta ameaça, o roteiro de entrevista foi avaliado por três pesquisadores. Após a avaliação, realizou-se uma entrevista piloto com um estudante de graduação com experiência prévia em especificação de UC, a fim de verificar se o roteiro atingiria seu objetivo. Os resultados da entrevista foram avaliados em conjunto com outro pesquisador e não foi preciso realizar melhorias no roteiro.

#### **4.5.4 Confiabilidade**

Neste estudo, considerou-se como ameaça o viés do pesquisador na interpretação dos dados qualitativos. Para tratar esta ameaça, outro pesquisador com alto nível de conhecimento em análise qualitativa e especificação de UC acompanhou todas as etapas da análise dos dados qualitativos.

### **4.6 CONSIDERAÇÕES FINAIS**

Este capítulo apresentou um estudo qualitativo que teve como objetivo investigar *“Quais as dificuldades percebidas ao especificar casos de uso durante o desenvolvimento de*

*software?*”. Para isso, foram conduzidas entrevistas semiestruturadas com estudantes de duas universidades distintas, com conhecimento prévio em especificação de UC. A análise dos dados foi realizada utilizando alguns procedimentos do método GT. Além disso, os resultados deste estudo foram publicados no artigo “*Um Modelo sobre as Dificuldades para Especificar Casos de Uso*” (Nascimento et al. 2017).

Visando responder à questão de pesquisa, os resultados da análise qualitativa mostraram que os estudantes (com e sem experiência prévia com UCs na indústria) apresentam dificuldades na especificação de UC em: Abstrair os Requisitos, Pré e Pós-Condições, Fluxos do UC, Regras de Negócio e Referências de Fluxos e RNs. Percebeu-se que existem dificuldades que são comuns tanto para estudantes que estão aprendendo, quanto para os estudantes que possuem experiência com UCs na indústria, estas dificuldades podem estar relacionadas ao (i) aprendizado prático da descrição textual de UCs e a (ii) falta de informações do sistema, necessárias para a descrição do UC.

Além disso, percebeu-se que algumas dificuldades ocorreram apenas com os estudantes que não possuíam experiência na indústria, tais como dificuldades na identificação e descrição dos fluxos (principal e alternativos) e confundir RNs com os RNFs do sistema. Já os estudantes que possuíam experiência apresentaram dificuldades na descrição da interação do sistema com outros sistemas que interagem no UC e em referenciar as RNs nos passos dos fluxos do UC. Espera-se que estas dificuldades encontradas no estudo e sintetizadas no modelo sirvam de base para futuras pesquisas na área, bem como apoio à sugestão de práticas baseado na literatura para melhorar o processo de ensino/ aprendizagem dos alunos na especificação de UC.

O próximo capítulo apresenta um estudo experimental realizado para identificar dificuldades na modelagem do diagrama de UC, devido ao foco principal deste estudo ser na especificação textual e não no diagrama de UC. Além disso, foi investigada a correlação entre os tipos de dificuldades e outros fatores que podem influenciar as dificuldades apresentadas no modelo. Assim, com os resultados deste novo estudo, o modelo poderá ser aprimorado e aperfeiçoado. Também, foi investigado o uso da inspeção de UCs com uso de *checklist* com o propósito de investigar a percepção dos alunos utilizando esta prática na melhoria do ensino/aprendizagem de UC.

## **CAPÍTULO 5 – ESTUDO EXPERIMENTAL SOBRE AS DIFICULDADES INICIAIS EM CASOS DE USO E O APOIO DA INSPEÇÃO PARA O APRENDIZADO NA MODELAGEM DE CASOS DE USO**

*Este capítulo apresenta o 3º estudo realizado na Fase II (Investigativa). Assim, será detalhado o planejamento, execução e resultados deste estudo realizado no contexto de uma disciplina de Engenharia de Software do Curso de Ciência da Computação da Universidade Federal do Amazonas (UFAM). Este estudo teve como objetivo analisar a percepção dos alunos sobre as dificuldades iniciais encontradas durante a modelagem do diagrama e da especificação textual do UC. Além disso, foi feita uma comparação das novas dificuldades com as anteriores para aprimorar e evoluir o modelo. Finalmente, foi analisada a correlação entre os tipos de dificuldades do modelo apresentado no capítulo anterior com as novas dificuldades identificadas neste estudo.*

### **5.1 INTRODUÇÃO**

No estudo anterior foram identificadas dificuldades sobre a especificação textual de UC, coletadas através de entrevistas semiestruturadas com participantes de duas universidades (Universidade Federal do Rio de Janeiro - UFRJ e Universidade Federal do Amazonas - UFAM) localizadas em diferentes regiões do país. Os participantes possuíam conhecimento em especificação de UC e durante a entrevista poderiam expressar sua percepção de acordo com sua experiência prática (na academia/indústria). As dificuldades relatadas pelos participantes foram analisadas qualitativamente e serviram de base para a criação da primeira versão do modelo de dificuldades em UCs.

Contudo, devido ao principal foco do estudo anterior ser na especificação textual e não no diagrama de UC, foi realizado um novo estudo experimental para investigar as dificuldades iniciais que os alunos encontram tanto para a modelagem do diagrama de UC, quanto para a especificação textual do UC. Além disso, investigou-se a correlação entre os tipos de dificuldades e os fatores que podem influenciar estas dificuldades durante a modelagem do diagrama e especificação do UC.

Desta forma, com os resultados deste novo estudo foi possível comparar as dificuldades encontradas com as dificuldades do estudo anterior para aprimorar e evoluir o modelo. Além

disso, foi analisado o uso de inspeção em casos de uso (diagrama e especificação) com o objetivo de entender a percepção dos alunos sobre possíveis melhorias na modelagem do diagrama e na especificação textual de casos de uso, como forma (prática) de melhorar o processo de ensino/aprendizagem de UCs.

Este capítulo está organizado da seguinte maneira: na Seção 5.2 descreve o planejamento, a condução e execução do estudo experimental. A Seção 5.3 apresenta a análise dos resultados quantitativos e qualitativos do estudo. A Seção 5.4 apresenta a discussão dos resultados. A Seção 5.5 apresenta as ameaças à validade deste estudo. E, por fim, a Seção 5.6 apresenta as considerações finais deste capítulo.

## **5.2 ESTUDO EXPERIMENTAL**

Esta seção apresenta o objetivo, planejamento e execução do estudo experimental. Além disso, apresenta os resultados da análise dos trabalhos práticos sobre UCs realizado no contexto de uma disciplina de Introdução à Engenharia de Software (IES) do curso de Ciência da Computação da UFAM.

### **5.2.1 Planejamento do Estudo**

O objetivo do estudo experimental foi compreender as dificuldades iniciais percebidas pelos alunos durante a modelagem do diagrama e da especificação de UC. Além disso, avaliar a percepção dos alunos quanto ao uso da inspeção em modelos de casos de uso (diagrama e especificação) como prática na melhoria do ensino/aprendizagem de UC. Através da compreensão destas percepções, foi possível entender o que os impede e/ou os ajuda a atingir os objetivos educacionais em relação aos casos de uso. Portanto, este estudo buscou responder as seguintes questões de pesquisa:

- RQ1 – *Quais as principais dificuldades de alunos durante a modelagem do diagrama e especificação de casos de uso?*
- RQ2 – *A atividade de inspecionar modelos de casos de uso pode ajudar alunos a compreenderem melhor casos de uso?*

Nesta fase, realizou-se a definição do escopo do estudo (quais etapas da ementa da disciplina o estudo seria conduzido), a preparação dos materiais que foram utilizados (exercícios práticos sobre diagrama, especificação e inspeção de UC, e *checklists* para inspeção dos modelos de UC) durante as aulas e nos trabalhos práticos, e a seleção das equipes para os trabalhos práticos (TP): no primeiro TP (TP1) realizou-se a modelagem do diagrama e especificação dos UCs (modelos) e no segundo TP (TP2) foi feita a inspeção dos modelos de

UCs. Todas as atividades desta fase foram realizadas pelo moderador do estudo e foram revisadas por dois pesquisadores. Durante o planejamento, foram definidos os seguintes recursos necessários para a realização do estudo:

#### 5.2.1.1 Contexto da Disciplina

O estudo foi executado durante a disciplina de Introdução à Engenharia de Software (IES) do curso de Ciência da Computação na UFAM. O foco da disciplina é apresentar os principais conceitos e práticas utilizadas na Engenharia de Software. A ementa da disciplina é dividida em 6 etapas, que são: (i) Ciclo de Vida e Processos utilizados no desenvolvimento de software; (ii) Elicitação de Requisitos durante o Desenvolvimento de Software; (iii) Casos de Uso; (iv) Inspeção de Software; (v) Teste de Software e (vi) Manutenção de Software. Nessa edição da disciplina, 43 alunos estavam matriculados.

Em relação à etapa de **(iii) Casos de Uso**, foi ensinado aos alunos a modelar o diagrama e a descrição textual de casos de uso. Para a modelagem foi utilizado o diagrama de casos de uso da UML (OMG, 2010) e para a descrição textual foi utilizado como exemplo o modelo proposto por Cockburn (2001) (com algumas adaptações, não sendo necessário a inclusão dos elementos trigger e palavras-chaves), por ser um dos modelos mais citados na literatura (Tiwari e Gupta, 2015).

Na etapa de **(iv) Inspeção de Software**, adotaram-se técnicas de inspeção para compreender a percepção dos alunos sobre a melhoria de casos de uso modelados. Além disso, com as técnicas de inspeção, as dificuldades com os modelos de casos de uso e seu apoio no aprendizado dos alunos poderiam ser melhor compreendidas. Para isto, adotou-se o processo de Sauer *et al.* (2000) utilizado para aplicar a inspeção nos modelos de casos de uso. Tal processo vem sendo adotado para apoiar inspeções em diferentes etapas do processo de desenvolvimento (Rigby e Storey, 2011; Bacchelli e Bird, 2013). O processo proposto por Sauer *et al.* (2000) possui as seguintes etapas:

- a) Planejamento: atividade onde é definido o escopo da inspeção (escolha dos principais módulos dos modelos), preparação do roteiro, seleção dos inspetores, preparação da infraestrutura, treinamento dos inspetores (profissionais que realizam a inspeção) e atribuição das tarefas a cada inspetor.
- b) Detecção: cada inspetor executa individualmente essa atividade, a qual consiste na busca de discrepâncias (problemas identificados no diagrama) nos modelos.
- c) Coleção: eliminação de discrepâncias repetidas (encontradas por mais de um inspetor), gerando uma lista de discrepâncias únicas (sem duplicatas).

- d) Discriminação: Classificação das discrepâncias em defeitos reais. As discrepâncias não classificadas como defeitos são consideradas como falso-positivos.
- e) Retrabalho: autor do documento realiza as alterações necessárias para corrigir os defeitos, produzindo um relatório explicando o trabalho realizado.

Adotou-se as classes de defeitos de Travassos *et al.* (1999) que comumente são encontrados em artefatos de software. Contudo, as classes de defeitos de Travassos *et al.* (1999) foram adaptadas para os tipos de defeitos encontrados em digramas e especificação de casos de uso, como apresentadas na Tabela 5.1.

**Tabela 5.1:** Classes de defeitos adaptado de Travassos *et al.* (1999).

Classes de Defeitos	Descrição
<b>Omissão</b>	Informações necessárias foram omitidas do caso de uso.
<b>Fato Incorreto</b>	Algumas informações no caso de uso contradizem a lista de requisitos ou do conhecimento geral do domínio do sistema
<b>Inconsistência</b>	As informações em uma parte do caso de uso estão inconsistentes com outras no caso de uso (ou em outras partes da Especificação).
<b>Ambiguidade</b>	As informações no caso de uso são ambíguas, isto é, é possível ao cliente, desenvolvedor ou testador interpretar as informações de diferentes maneiras podendo não levar a uma implementação correta
<b>Informação Estranha</b>	As informações fornecidas não são necessárias para o caso de uso.

#### 5.2.1.2 Ambiente

Não foi definido um ambiente específico para a realização do estudo, pois os alunos tiveram duas semanas para realizar o primeiro trabalho prático e mais duas semanas para realizar o segundo trabalho prático. O primeiro trabalho prático foi conduzido na etapa **(iii) Casos de Uso** da ementa da disciplina de IES o segundo trabalho prático foi conduzido na etapa **(iv) Inspeção de Software**.

#### 5.2.1.3 Participantes

Quarenta e três (43) participantes assinaram o formulário de consentimento livre e esclarecido, a fim de atender às exigências éticas da pesquisa, explicando aos participantes sobre a confidencialidade dos dados fornecidos e do seu anonimato. Vale ressaltar que os alunos não tinham experiência prévia na modelagem de casos de uso (diagrama e especificação) na indústria.

#### 5.2.1.4 Artefatos

Os artefatos utilizados neste estudo experimental foram validados por três pesquisadores, são eles:



- Termo de Consentimento Livre e Esclarecido (TCLE);
- Conjunto de Artefatos 1 (CA1) do TP1: lista de requisitos do sistema com descrição dos requisitos funcionais e os casos de uso (diagrama e a especificação textual dos UCs);
- Conjunto de Artefatos 2 (CA2) do TP2: *checklists* para inspeção dos diagramas e especificações de UCs (ver APÊNDICE G) e a planilha de discriminação de defeitos;
- Questionário pós-estudo.

O **questionário pós-estudo** foi aplicado com a finalidade de compreender as dificuldades relatadas pelos alunos no desenvolvimento dos trabalhos práticos TP1 e TP2. Assim, os alunos foram convidados a responder um questionário após a entrega do TP2. O questionário foi disponibilizado aos alunos por meio do Google Forms. Um e-mail foi enviado aos alunos, contendo um link para acesso ao questionário e dado um prazo para responderem as perguntas relacionadas às dificuldades que tiveram durante a modelagem de casos de uso. O questionário era composto por 4 questões objetivas. A Tabela 5.2 detalha as 4 questões objetivas e seus objetivos.

**Tabela 5.2:** Questões Objetivas do Questionário de Dificuldades em UCs.

Questões	Objetivo
Q1) Na sua opinião pra que serve modelo de casos de uso (o diagrama e a especificação)? No que ajuda e no que atrapalha?	Capturar a opinião dos alunos quanto ao uso de casos de uso no desenvolvimento de software;
Q2) Durante a especificação do caso de uso você tem/teve alguma dificuldade em descrever? Se sim, que tipo de dificuldades foram estas?	Capturar possíveis dificuldades que o aluno teve durante a especificação dos casos de uso;
Q3) Durante a modelagem do diagrama de caso de uso você tem/teve alguma dificuldade em modelar? Se sim, que tipo de dificuldades foram estas?	Capturar possíveis dificuldades que o aluno teve durante a modelagem do diagrama;
Q4) A atividade de inspecionar os documentos (Diagrama e da Especificação de Casos de Uso) ajudou no seu aprendizado de casos de uso? Caso sim, como ajudou?	Capturar a percepção do aluno se a atividade de inspecionar influenciou no aprendizado de casos de uso.

Nas questões (Q2) e (Q3) havia sub-questões com os tipos de dificuldades que poderiam ocorrer durante a modelagem do diagrama e especificação de casos de uso como apresentado na Tabela 5.3. O aluno poderia selecionar mais de uma sub-questão, sendo obrigatório selecionar pelo menos uma de cada questão (Q2) e (Q3). Além disso, ao selecionar essas sub-questões o aluno relatava detalhadamente sobre cada dificuldade. Este questionário foi revisado

e validado por três pesquisadores com alto nível de conhecimento em inspeção de software antes de ser disponibilizado aos alunos. De 43 alunos da disciplina, 33 alunos responderam ao questionário (77%).

**Tabela 5.3:** Sub-questões do Questionário de Dificuldades em UCs.

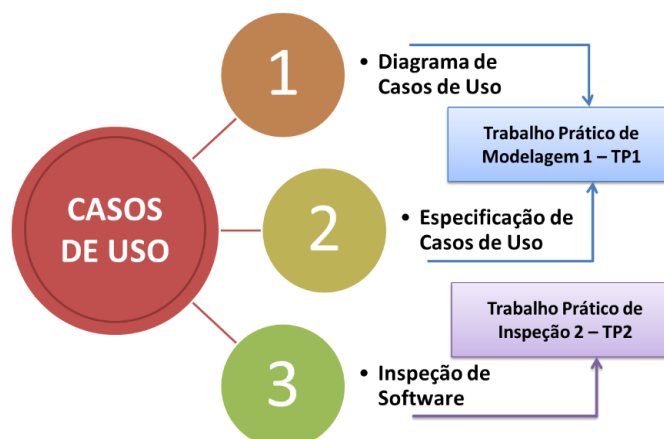
Questões	Sub-questões
<b>Dificuldades que o aluno teve durante a especificação dos casos de uso (Q2)</b>	Q2-A. Em entender (abstrair) o requisito e descrever no caso de uso? Q2-B. Em identificar e organizar os fluxos (principal, alternativos e exceções) e descrever os cenários para cada fluxo? Q2-C. Em identificar as regras de negócio do caso de uso e descrevê-las? Você confundiu regras de negócio com outros elementos do caso de uso? Se sim, quais? Q2-D. Em referenciar as regras de negócio e os fluxos (alternativos e de exceções) nos passos dos fluxos do caso de uso? Q2-E. Em saber onde incluir a dependência (extend e/ou include) de outros casos de uso (se existir) nos fluxos? Q2-F. Outra dificuldade? Qual?
<b>Dificuldades que o aluno teve durante a modelagem do diagrama (Q3)</b>	Q3-A. Em identificar os atores do sistema? Q3-B. Em utilizar a generalização entre os atores identificados? Q3-C. Em identificar as principais tarefas que cada ator necessita no sistema (os casos de uso)? Q3-D. Em utilizar os relacionamentos extend e include entre os casos de uso? Q3-E. Outra dificuldade? Qual?

#### 5.2.1.5 Execução do Estudo

Inicialmente foi feito um acompanhamento com os alunos durante as aulas sobre casos de uso (referente à etapa ‘iii – Casos de Uso’ da ementa). Em seguida, os alunos realizaram o TP1 de modelagem de casos de uso. Após esse trabalho, foram apresentados conceitos e exercícios práticos de inspeção em modelos de casos de uso (referente à etapa ‘iv – Inspeção de Software’ da ementa). Na sequência, os alunos realizaram o TP2, ou seja, inspecionaram os modelos gerados no TP1 de modelagem de casos de uso (diagrama e especificação). Por fim, os alunos foram convidados a responder um questionário pós-estudo.

A Figura 5.1 apresenta a execução do estudo quanto à ordem dos conceitos ensinados em relação aos casos de uso e inspeção de software, que são: (1) Diagrama de Casos de Uso, (2) Especificação de Casos de Uso e (3) Inspeção de Software. Além disso, a Figura 5.1 apresenta a ordem dos trabalhos práticos (TP1 e TP2) realizados durante a disciplina. Os trabalhos práticos foram realizados por equipes de até 5 alunos. A seguir as atividades que os alunos realizaram em cada trabalho são detalhadas:

Conceitos Ensinados sobre Casos de Uso

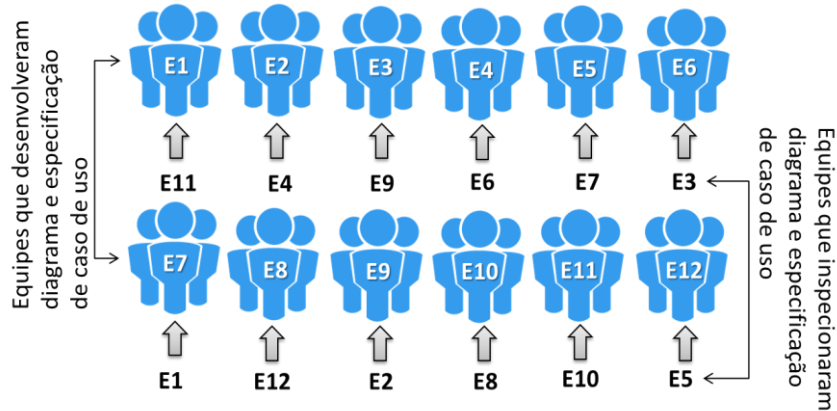


**Figura 5.1:** Conceitos relacionados a casos de uso e trabalhos práticos realizados.

- **Modelagem do Diagrama e Especificação de Caso de Uso (Trabalho Prático de Modelagem 1 – TP1)** – Esse trabalho foi solicitado após serem ministrados os conceitos e atividades práticas realizadas em sala de aula sobre diagrama e especificação de casos de uso seguindo a etapa (iii) da ementa. Os alunos realizaram a modelagem dos casos de uso (diagrama e especificação) baseado nos requisitos definidos na etapa (ii) da ementa. Os alunos haviam realizado a elicitação de requisitos através de entrevistas com usuários reais baseado na etapa (ii) Elicitação de Requisitos (da ementa) durante o Desenvolvimento do Software. Os alunos tiveram duas semanas para concluir este trabalho.
- **Inspeção de Diagramas e Especificação de Caso de Uso (Trabalho Prático de Inspeção 2 – TP2)** – Esse trabalho foi solicitado após serem ministrados os conceitos e atividades práticas realizadas em sala de aula sobre (iv) Inspeção de Software, definido na ementa. Além disso, após a finalização do TP1, foram definidas diferentes equipes que inspecionariam os modelos de casos de uso, como mostra a Figura 5.2. Neste trabalho, foi solicitado aos alunos que utilizassem as seguintes etapas do processo de inspeção de Sauer *et al.*, (2000): Planejamento, Detecção, Coleção e Discriminação. Além disso, os alunos deveriam apresentar um relatório apresentando datas e tempo de reunião de todas as atividades realizadas.

Apesar dos trabalhos práticos (TP1 e TP2) serem realizados em equipes, todos os alunos participaram. Foi solicitado de cada equipe um relatório técnico descrevendo as atividades executadas por cada membro da equipe informando os casos de uso modelados e inspecionados. Desta forma, todos os alunos modelaram e especificaram no TP1 e inspecionaram os modelos

de outros alunos diferentes dos modelos modelados por sua equipe no TP2. Além disso, a quantidade de casos de uso de modelados e inspecionados pelas equipes eram equivalentes.



**Figura 5.2:** Design do segundo trabalho prático na disciplina.

Para os diagramas e especificações de casos de uso gerados no TP1 os alunos receberam três *checklists* de inspeção (ver APÊNDICE G). A técnica de *checklist* foi escolhida por ser uma das mais utilizadas para inspecionar casos de uso (Anda *et al.*, 2009). Assim, foram elaborados três *checklists* de inspeção com base em Travassos *et al.* (1999) e Anda *et al.* (2009) com o objetivo de atender diferentes propósitos. Os propósitos dos *checklists* referem-se à detecção de defeitos nos (1) digramas de caso de uso; (2) especificação de caso de uso; e (3) consistência dos digramas, especificação de caso de uso e requisitos. Com base no trabalho de Travassos *et al* (1999) foi chamado de “Inspeção Vertical” o *checklist* que tinha o propósito de verificar a consistência dos diagramas com a especificação de casos de uso e os requisitos. Os três *checklists* foram revisados por dois pesquisadores antes de serem entregues aos alunos para o TP2. Para este trabalho (TP2) os alunos tiveram duas semanas para concluir esta atividade.

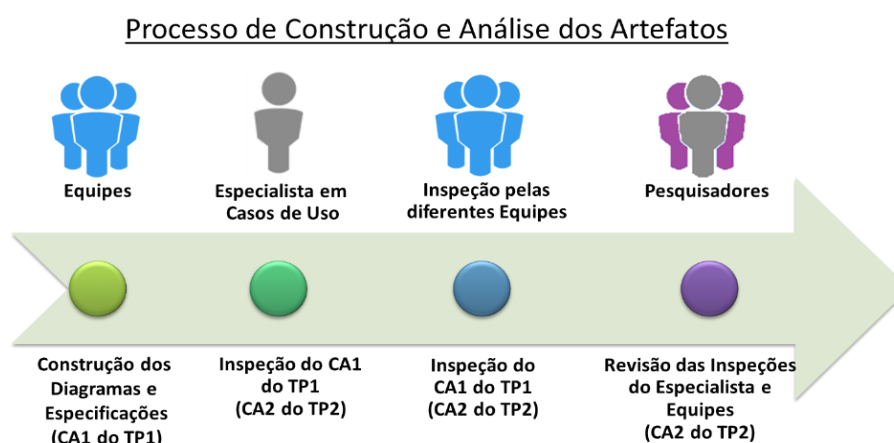
### 5.2.2 Resultados da Análise dos Artefatos (CA1 e CA2) gerados nos Trabalhos Práticos: TP1 e TP2

O primeiro conjunto de artefatos (CA1) analisados foi gerado no TP1 que continha: lista de requisitos do sistema (descrição dos requisitos funcionais) e os casos de uso (diagrama e a especificação). O CA1 foi analisado com o objetivo de entender a maneira como os alunos desenvolveram o diagrama e especificação de UCs. O segundo conjunto de artefatos (CA2) analisado foi gerado no TP2 que continha: os *checklists* para inspeção dos diagramas e especificações de UCs e a planilha de discriminação.

O CA2 foi analisado com o objetivo de compreender o entendimento dos defeitos identificados em relação aos diagramas e especificações de casos de uso de outras equipes. O

processo de análise dos artefatos gerados pelas equipes no TP1 e TP2 foi planejado para ser executado em duas etapas como mostra a Figura 5.3. Na primeira etapa, para o TP1, as inspeções nos casos de uso (diagramas e especificações) foram realizadas por um especialista em casos de uso com 10 anos de experiência na indústria. O especialista utilizou os *checklists* na inspeção dos diagramas e especificações de casos de uso. Além de realizar a Inspeção Vertical, verificando a consistência dos diagramas com a especificação de casos de uso e os requisitos.

Na segunda etapa do TP2 dois pesquisadores realizaram a revisão dos defeitos apontados pelo especialista e pelas equipes que desenvolveram o TP1. O objetivo foi retirar as discrepâncias repetidas, falsos-positivos (discrepâncias que não eram defeitos) e verificar o número de defeitos encontrados. Além disso, foi realizada uma análise comparativa das respostas dos alunos no questionário para compreender suas dificuldades relatadas em relação aos trabalhos práticos TP1 e TP2.



**Figura 5.3:** Processo de Construção e Análise dos Artefatos (TP1 e TP2).

## 5.3 RESULTADOS OBTIDOS

Nesta subseção serão apresentados os resultados quantitativos (Na Subseção 5.3.1) e qualitativos (Na Subseção 5.3.2) obtidos no estudo. Além disso, na Subseção 5.3.2 são discutidos os resultados sobre a percepção dos alunos em relação à modelagem dos casos de uso e a atividade de inspeção.

### 5.3.1 Resultados Quantitativos

Na Subseção 5.3.1.1 são apresentados os resultados quantitativos do TP2, mostrando o total de defeitos encontrados pelas equipes e pelo especialista em casos de uso. Nas Subseções

5.3.1.2, 5.3.1.3 e 5.3.1.4 são descritos os resultados quantitativos relativos aos defeitos encontrados no TP2. Esses resultados estão relacionados aos diagramas, especificação de casos de uso e a comparação dos diagramas com a especificação de caso de uso e requisitos. A Subseção 5.3.1.5 mostra as principais dificuldades obtidas a partir das respostas dos alunos no questionário pós-estudo.

#### 5.3.1.1 Resultados do Trabalho de Inspeção realizado pelas equipes (TP2) e especialista em modelos de casos de uso

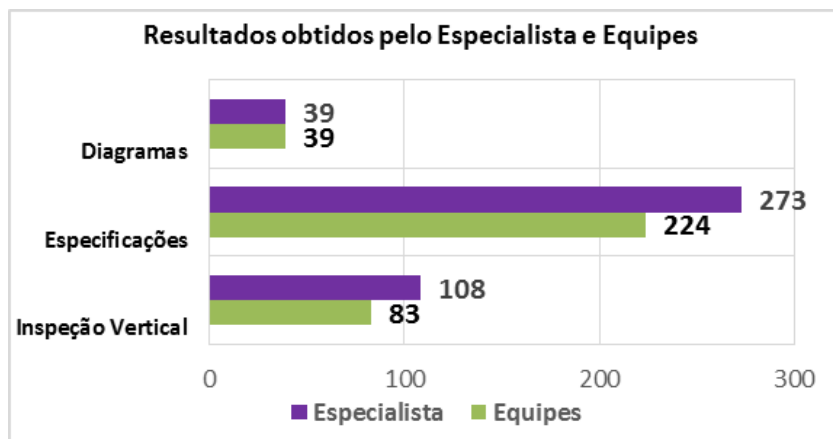
Os resultados quantitativos relacionados à inspeção realizada pelas equipes foram 39 Defeitos nos Diagramas (DD), 224 Defeitos nas Especificações (DE) e 83 Defeitos na comparação do diagrama, especificação de caso de uso e requisitos (DV) como mostra a Tabela 5.4.

**Tabela 5.4:** Defeitos encontrados pelas equipes nos Modelos de UCs.

Equipes	DD		DE		DV	
	Eq.	Esp.	Eq.	Esp.	Eq.	Esp.
<b>E1</b>	1	2	11	27	<b>9</b>	<b>9</b>
<b>E2</b>	1	1	9	26	<b>2</b>	<b>2</b>
<b>E3</b>	0	4	7	17	4	6
<b>E4</b>	2	2	<b>48</b>	<b>11</b>	10	7
<b>E5</b>	2	2	16	37	4	14
<b>E6</b>	<b>7</b>	<b>5</b>	<b>35</b>	<b>24</b>	3	11
<b>E7</b>	2	2	<b>45</b>	<b>32</b>	4	8
<b>E8</b>	3	2	15	26	4	7
<b>E9</b>	<b>9</b>	<b>5</b>	-	-	16	14
<b>E10</b>	3	3	7	21	9	8
<b>E11</b>	2	3	3	19	3	11
<b>E12</b>	7	8	28	33	15	11

**Legenda:**  
**Eq.** – Equipes;  
**Esp.** – Total de defeitos encontrados pelo Especialista em Casos de Uso;  
**DD** – Total de defeitos encontrados na inspeção dos diagramas de casos de uso;  
**DE** – Total de defeitos encontrados na inspeção das especificações de casos de uso;  
**DV** – Total de defeitos encontrados na inspeção vertical.

Na Tabela 5.4 observou-se que algumas equipes como E6 e E9, encontraram defeitos nos diagramas a mais que o especialista. Na coluna DE o número de defeitos encontrados pelas equipes nas especificações também são semelhantes, e algumas equipes (E4, E6 e E7) encontraram defeitos a mais que o especialista nas especificações. Na coluna DV mostra que o número de defeitos encontrados pelas equipes e especialista também são equivalentes, como encontrados pelas equipes E1 e E2. A Figura 5.4 mostra o número total de defeitos encontrados pelas equipes e especialista em relação ao DD, DE e DV.



**Figura 5.4:** Total de defeitos identificados pelo Especialista e Equipes.

Com objetivo de verificar se existiam diferenças significativas entre a quantidade de defeitos encontrados nos diagramas e especificações de casos de uso pelo especialista e equipes da disciplina de IES, as seguintes hipóteses (nula e alternativa, respectivamente) foram testadas:

- **H<sub>0</sub>:** Não há diferença na quantidade de defeitos detectados na inspeção dos modelos pelo especialista e equipes.
- **H<sub>1</sub>:** Há diferença na quantidade de defeitos detectados na inspeção dos modelos pelo especialista e equipes.

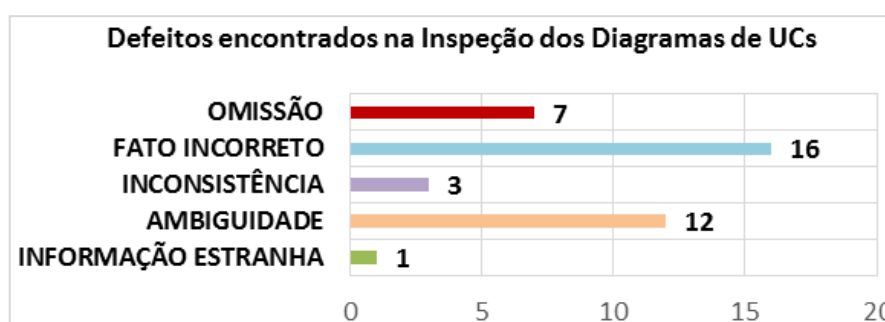
Para testar as hipóteses (H<sub>0</sub> e H<sub>1</sub>), realizou-se testes estatísticos apoiados pela ferramenta SPSS V. 23. Primeiramente foi feito o teste de normalidade de Shapiro-Wilk, com  $\alpha=0,05$ , para o número de defeitos encontrados em relação ao DD, DE e DV. O teste de normalidade mostrou que a distribuição dos valores é normal para DD (com  $p= 0,024$  para os resultados do especialista;  $p= 0,032$  para os resultados das equipes) e não é normal para DE (com  $p= 0,087$  para os resultados do especialista;  $p= 0,613$  para os resultados das equipes) e DV (com  $p= 0,021$  para os resultados do especialista e  $p= 0,632$  para os resultados das equipes). Devido a esses resultados, utilizou-se o t-teste para analisar os resultados do DD e o teste não paramétrico de Mann-Whitney para DE e DV (McCrum-Gardner, 2008).

Com base nos resultados dos testes, não existe diferença estatística entre a quantidade de defeitos encontrados pelo especialista e equipes nos diagramas e especificações de casos de uso ( $p=0,238$  para DD;  $p=0,298$  para DE e  $p=0,212$  para DV, suportando a H<sub>0</sub>). Estes resultados mostram que as técnicas de inspeção do tipo *checklist* auxiliaram tanto inspetores novatos quanto o especialista. Além disso, foram explorados os principais defeitos inspecionados, apresentados nas próximas subseções.

### 5.3.1.2 Resultados sobre a Inspeção nos Diagramas de Casos de Uso (TP1)

Os resultados quantitativos relacionados à inspeção nos diagramas de casos de uso gerados pelas equipes no TP1 resultaram em 39 defeitos, divididos em defeitos de Omissão (7), Fato Incorreto (16), Inconsistência (3), Ambiguidade (12) e Informação Estranha (1) como mostra a Figura 5.5. Os principais defeitos encontrados estão relacionados às classes de **Fato Incorreto**, **Omissão** e **Ambiguidade**.

Os defeitos de **Fato Incorreto** nos diagramas estão relacionados à sintaxe como: (a) o nome do caso de uso que não era único, (b) casos de uso divididos erroneamente em partes menores, (c) a direção da seta para representar a associação de inclusão (include) ou extensão (extend) de outro caso de uso não estava correta no diagrama.



**Figura 5.5:** Quantidade de tipos de defeitos encontrados na Inspeção dos Diagramas de UCs.

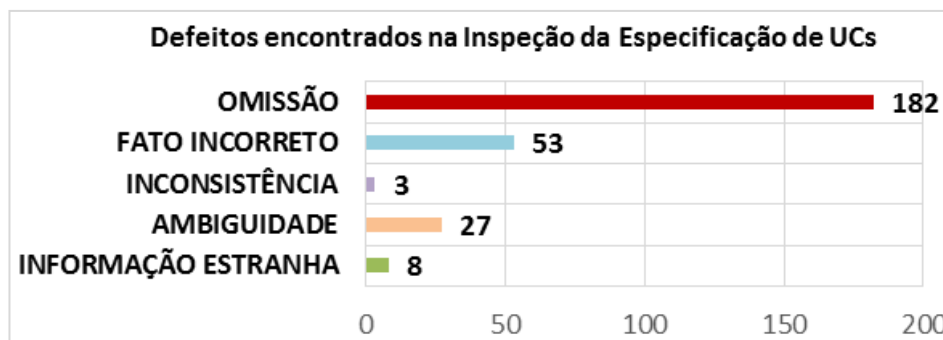
Os defeitos de **Ambiguidade** nos diagramas foram: (a) nome dos atores e/ou casos de uso que não estavam claros permitindo ter dupla interpretação, (b) casos de uso que executavam a mesma funcionalidade e as (c) associações dos atores com os casos de uso não estavam claras.

Os defeitos de **Omissão** nos diagramas foram: (a) não identificar outros sistemas (atores) que se comunicava com o sistema a ser construído, (b) nem todas as funcionalidades do sistema foram representadas no diagrama, (c) faltou incluir alguma associação entre um ator com o caso de uso no diagrama ou (d) faltou incluir um ator que herdasse o perfil de outro ator (Especialização) realizando algo a mais no sistema.

### 5.3.1.3 Resultados sobre a Inspeção das Especificações dos Casos de Uso (TP1)

Os resultados quantitativos relacionados à inspeção realizada nas especificações de casos de uso, geradas pelas equipes no TP1, resultaram em 273 defeitos. A Figura 5.6 apresenta os principais defeitos encontrados na seguinte ordem, Omissão (182 defeitos), Fato Incorreto (53), Ambiguidade (27), Inconsistência (3) e Informação Estranha (8).





**Figura 5.6:** Quantidade de tipos de defeitos encontrados na Inspeção das Especificações de UCs.

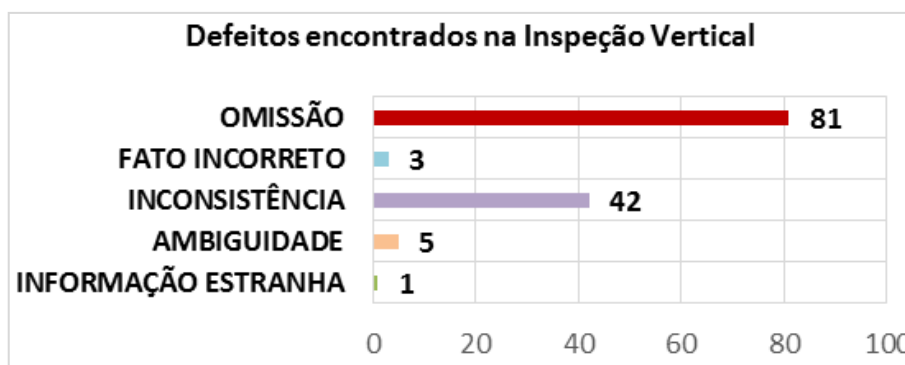
Os defeitos de **Omissão** nas especificações foram: (a) faltou identificar ou descrever o nome dos atores no caso de uso, (b) faltou descrever na pré-condição alguma condição indispensável para iniciar o caso de uso, (c) faltou descrever algum fluxo de evento (alternativo e/ou exceção) necessário para a completude do caso de uso, (d) faltou descrever alguma regra de negócio necessária para completude do caso de uso, (e) faltou referenciar alguma regra de negócio, fluxos alternativos ou de exceção no caso de uso e (f) faltou incluir os pontos de extensão de outros casos de uso.

Os defeitos de **Fato Incorreto** nas especificações estavam relacionados com: (a) as identificações dos fluxos (alternativo – A1, A2 e exceção – E1, E2) e regras de negócio (RN) que não eram representados no caso de uso ou não estavam sendo referenciados corretamente, (b) a descrição dos fluxos não estavam completas ou corretas e a (c) descrição das regras de negócio não estavam completas e corretas.

Os defeitos de **Ambiguidade** nas especificações foram: (a) descrições de pré e pós-condições confusas (quando existiam), (b) descrições dos fluxos de eventos e das regras de negócios não claras, (c) fluxos de eventos (principal, alternativo e exceção) não seguiam um caminho lógico e claro, (d) também as informações trocadas entre o ator e o sistema não estavam bem definidas nos fluxos.

#### 5.3.1.4 Resultados sobre a Inspeção Vertical (TP1)

Os resultados da comparação de inspeção do digrama, especificação de caso de uso e requisitos, resultaram em 132 defeitos. A Figura 5.7 mostra os defeitos identificados na seguinte ordem: Omissão (81 defeitos), Inconsistência (42), Ambiguidade (5), Fato Incorreto (3) e Informação Estranha (1).



**Figura 5.7:** Quantidade de tipos de defeitos encontrados na Inspeção Vertical.

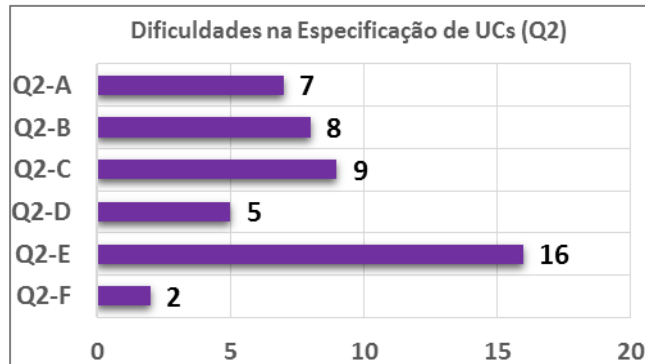
Em relação aos defeitos de **Omissão** na inspeção vertical: (a) faltou identificar os relacionamentos de extensão ou inclusão na especificação do caso de uso conforme o diagrama e (b) faltou identificar algum caso de uso no diagrama para atender uma funcionalidade descrita na lista de requisitos.

Os defeitos de **Inconsistência** na inspeção vertical: (a) a especificação do caso de uso não estava consistente com o diagrama, (b) a associação para estender o caso de uso (*extend*) não estava consistente e (c) o diagrama de casos de uso não estava consistente com a lista de requisitos.

### 5.3.1.5 Resultados do Questionário

A Figura 5.8 mostra a quantidade de respostas dos participantes nos questionários sobre as dificuldades na especificação de UCs. As principais dificuldades foram:

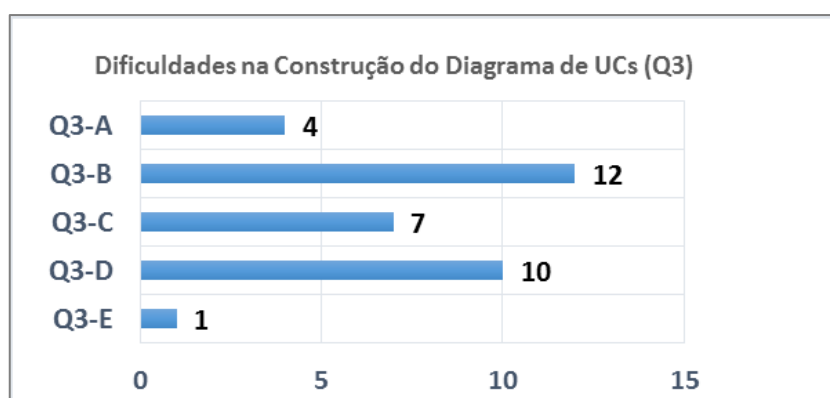
- Uso dos **relacionamentos de *extend* e *include*** no UC (Q2-E: 16 respostas). Por exemplo, o participante P9 relatou “*Em que momento tem que fazer um extend, onde fazer, como fazer, etc*”;
- Saber **identificar as regras de negócio** e descrevê-las no UC (Q2-C: 9). Por exemplo, o participante P13 mencionou “*Às vezes a regra de negócio não fica muito clara e eu não sei como descrevê-las*”;
- **Identificar, organizar e descrever os fluxos** (principal, alternativos e exceções) no UC (Q2-B: 8). O participante P15 afirmou “*No início tive dificuldades em entender quando algo era um fluxo alternativo...*”;
- **Entender (abstrair) o requisito** e descrever no UC (Q2-A: 7). O participante P1 relatou “*A única dificuldade foi em entender o requisito e descrever no caso de uso...*”.



**Figura 5.8:** Dificuldades percebidas a partir das respostas para a especificação textual do UC.

Quanto às dificuldades relatadas pelos alunos no questionário para a construção do diagrama de UC, são apresentadas na Figura 5.9. As principais dificuldades observadas foram:

- Uso da **generalização entre os atores** identificados (Q3-B: 12 respostas). Por exemplo, o participante P4 marcou essa dificuldade e mencionou “*Tive dificuldade em entender a generalização na primeira vez que a vi*”;
- Uso dos **relacionamentos *extend* e *include*** entre os casos de uso (Q3-D: 10). Por exemplo, o participante P29 marcou essa dificuldade e afirmou “*Me confundi um pouco com os includes e extends...*”.
- **Identificação das principais tarefas** que cada ator necessita no sistema (Q3-C: 7). Por exemplo, o participante P30 relatou “*... foi complicado entender o que cada caso necessitava para que o caso se apresentasse completo, ou seja, as ações necessárias para tal completude*”.



**Figura 5.9:** Dificuldades percebidas a partir das respostas para a construção do Diagrama de UCs.

### 5.3.2 Resultados Qualitativos

Para a análise qualitativa das dificuldades citadas pelos alunos ao responderem o questionário foram utilizados alguns procedimentos de codificação (Strauss e Corbin, 1998)

com auxílio da ferramenta Atlas.ti<sup>3</sup>. Os comentários descritos pelos alunos sobre as dificuldades relatadas durante a modelagem do diagrama e na especificação de casos de uso foram analisados.

Os dados extraídos de cada resposta do questionário foram analisados e criaram-se códigos relevantes sobre a percepção dos alunos quanto o uso de modelos de casos de uso (diagrama e especificação) no desenvolvimento de software. A partir dos códigos, as categorias e relacionamentos entre os códigos foram criados. O processo de codificação foi revisado por um pesquisador com 15 anos de experiência com o ensino de modelagem e 7 anos de experiência com análise qualitativa. As citações e códigos foram discutidos e, ao final, foram gerados 59 códigos, associados a diferentes categorias e subcategorias. Através da discussão destes códigos, analisaram-se as percepções dos alunos em relação à modelagem de casos de uso (diagrama e especificação). Também foi analisada a percepção dos alunos sobre a atividade de inspeção como apoio no ensino de casos de uso.

A subseção 5.3.2.1 apresenta os resultados das percepções dos alunos sobre a Modelagem de UCs e a subseção 5.3.2.2 apresenta os resultados das percepções dos alunos sobre o uso da inspeção como apoio ao ensino/aprendizagem de UCs.

#### 5.3.2.1 *Percepção dos Alunos sobre Modelagem de Casos de Uso*

Na categoria “Percepção sobre Modelos de UCs no desenvolvimento de software” foram associados códigos que mostram que os casos de uso (diagrama e especificação) podem ajudar no desenvolvimento de software, por exemplo, o código “Facilita o entendimento do sistema e na comunicação dos *stakeholders*”. Nesse sentido, três alunos citaram alguns aspectos que o modelo pode ajudar:

*“Ajuda quem vai programar a ter uma ideia de como deve ser o comportamento do sistema diante das necessidades dos atores.” – P5*

*“...O diagrama sozinho não é uma ferramenta poderosa, no entanto ao juntá-lo com a Especificação não somente eu consigo decidir o rumo das funcionalidades como também consigo saber o que cada diagrama vai se tornar posteriormente no código.” – P10*

*“...ajuda a estimar o esforço necessário para o desenvolvimento.” – P32*

Os três alunos mencionaram que o diagrama de UC pode ajudar e não atrapalhar no desenvolvimento de software. No entanto, o participante P10 mencionou que para o caso de

---

<sup>3</sup> Atlas.ti – The Knowledge Workbench, Scientific Software Development – <http://www.atlasti.com>

uso ser completo é preciso ter a modelagem do diagrama e da especificação do UC juntos. Segundo o participante P10 cada um dá uma visão do que será projetado para o sistema nas próximas fases do desenvolvimento do software.

Os códigos relacionados à subcategoria “Dificuldade no Diagrama de UC” foram criados a partir dos códigos que relatavam dificuldades com o diagrama de casos de uso, por exemplo, o código “Em definir os atores do sistema”. E para a subcategoria “Dificuldade em Especificar UC” foram criados códigos que relatavam dificuldades na especificação do caso de uso, como o código “Saber identificar as RNs e descrevê-las”. Essas subcategorias estão associadas à categoria “Dificuldades em Modelagem de Casos de Uso”.

Algumas das dificuldades com diagramas estão relacionadas em usar os relacionamentos *extend*, *include* e generalização. E algumas das dificuldades na especificação de casos de uso foram em incluir os pontos de extensão para outros casos de uso.

**Tabela 5.5:** Dificuldades em Modelar o Diagrama e Especificar UCs.

Aluno	Dificuldade durante a Construção do Diagrama	Dificuldade durante a Especificação do Caso de Uso
P1	<i>“No início era confuso a utilização dos relacionamentos extend e include, pois não sabia como relacionar um caso de uso com o outro de forma correta ou que realmente necessitasse...”</i>	<i>“A única dificuldade foi em entender o requisito e descrever no caso de uso. Foi uma dificuldade que tive no início e após várias práticas, fui conseguindo entender e interpretar melhor o que o sistema gostaria que fizesse.”</i>
P13	<i>“Muitas vezes me confundi com qual ator herdaria as funcionalidades de um determinado ator.”</i>	<i>“Tive dificuldade em saber onde referenciar um extend ou include na especificação do caso de uso.”</i>
P18	<i>“Às vezes me confundo no tipo de relacionamento entre casos de uso.”</i>	<i>“Em relação aos includes e extends em alguns casos dependendo do cenário.”</i>
P25	<i>“...mesmo sabendo as funcionalidades e atores, como representá-los, e dividir em casos de uso.”</i>	<i>“... dúvida foi em relação a como estruturar/dividir as funcionalidades em casos de uso, e.g, esta funcionalidade vai ser quebrada em quantos casos de uso? e como?”</i>
P28	<i>“... foi em identificar quando um conjunto de passos pode virar um extend e quando deve permanecer apenas como fluxo alternativo.”</i>	<i>“...uma dúvida que surgiu foi em como referenciar extends que representam diferentes opções para se realizar uma tarefa”</i>

### 5.3.2.2 Percepção dos Alunos sobre a Inspeção como apoio no aprendizado de Casos de Uso

Na categoria “Inspeccionar Modelos de Casos de Uso ajuda no aprendizado de UCs” foram associados códigos que indicam que a atividade de inspeção ajudou no aprendizado de casos de uso, por exemplo, o código “Ajudou a ver outras formas de especificar e melhorar a especificação de UCs”. Foram encontradas algumas afirmativas que mostram que a inspeção

auxiliou a lembrar os conceitos sobre casos de uso e evitar futuros erros em próximas modelagens, como:

*“... Auxiliou a lembrar alguns dos principais problemas que podem ocorrer e, desta forma, posso evitá-los no futuro.” – P5*

*“...Nesse trabalho pude identificar erros cometidos por mim também quando fiz o meu. Esses erros ficam mais evidentes e nos ajudam a lembrar o que e como algo deve ser feito, evitando-os.” – P15*

*“Me fez perceber alguns erros que eu talvez tenha cometido nas minhas especificações e também me fez pensar melhor em como os descrever numa próxima vez, de um modo que venha justamente a evitar que eu cometa as ambiguidades que encontrei nas descrições que avaliei.” – P24*

Também foram encontradas evidências de que aprender observando os erros nos diagramas e especificações de UCs inspecionados foi uma forma positiva para aprender os conceitos, como mencionado por dois alunos:

*“...a inspeção pode nos mostrar o que não fazer, o que na minha opinião é o melhor jeito de se aprender algo novo, ou seja, o aprendizado a base de erros para mim é melhor.” – P30*

*“Ajudou porque aprendemos com os erros dos outros e a fixar os principais erros através do checklist” – P16*

A atividade de inspecionar os casos de uso (diagrama e especificação) foi útil para os alunos lembrarem os conceitos ensinados em sala de aula e a evitar futuros erros nas próximas modelagens. Além disso, observou-se que o uso de *checklist* guiou os alunos a identificarem defeitos nos diagramas e especificações de UCs de forma sistemática.

Os resultados qualitativos mostram que os diagramas e especificações de casos de uso podem auxiliar o desenvolvimento de software sobre a identificação de funcionalidades do sistema. Além desses aspectos, também foi mencionado por alguns alunos que os casos de uso ajudam a pensar na interface e nas interações dos atores e sistema. Também foi mencionado que os casos de uso servem para organizar as funcionalidades de forma visual (diagrama) e estruturada (especificação). Estes resultados indicam que os conceitos ensinados sobre casos de uso e com o auxílio das técnicas de inspeção houve uma compreensão aperfeiçoada pelos alunos. Além disso, a atividade de inspeção melhorou o entendimento dos alunos em relação aos defeitos que devem ser evitados durante a construção do diagrama e especificação de casos de uso.

## 5.4 DISCUSSÃO DOS RESULTADOS

A Figura 5.10 mostra uma nova versão do modelo proposto no capítulo anterior. Esta nova versão é baseada nos resultados deste estudo experimental, que apresenta um resumo das dificuldades relatadas pelos alunos durante a construção dos modelos de casos de uso (Diagrama e Especificação). A sequência dessas dificuldades é apresentada a seguir:

- **Diagrama de UCs (D):** (D-I) utilizar os relacionamentos de *extend* e *include*; (D-II) utilizar a generalização entre os atores; (D-III) identificar as tarefas de cada ator; e (D-IV) identificar os atores do sistema.
- **Especificação de UCs (E):** (E-I) incluir a dependência de outros casos de uso; (E-II) referenciar RNs e Fluxos; (E-III) identificar as Regras de Negócio (RNs); (E-IV) identificar e organizar os fluxos; e (E-V) abstrair o requisito.

Em relação às dificuldades identificadas pelo especialista através da inspeção nos diagramas modelados no TP1, percebeu-se que os erros dos alunos, por exemplo, envolviam: o uso incorreto dos relacionamentos de *include* e *extend*, omissão de relacionamentos de atores com casos de uso, e omissão de funcionalidades que estavam na lista de requisitos. Além disso, os alunos que inspecionaram os diagramas gerados no TP1 também foram capazes de identificar esses tipos de defeitos, identificados pelo especialista, com o auxílio do *checklist*. Quanto ao questionário pós-estudo os alunos relataram algumas dificuldades sobre o uso do modelo, foram as seguintes: o uso incorreto (defeito de Fato Incorreto) da direção da seta para representar os relacionamentos de inclusão (*include*) e extensão (*extend*); omissões (defeito de Omissão) de relacionamentos dos atores com os casos de uso.

Nas especificações de casos de uso geradas no TP1 foi percebido que os alunos erraram, por exemplo, em não descrever fluxos alternativos e regras de negócio para a completude do caso de uso. No entanto, quando os alunos inspecionaram as especificações de casos de uso no TP2 perceberam, por exemplo, as omissões na descrição de fluxos alternativos e regras de negócio (defeito de Omissão). Além disso, os alunos encontraram os outros defeitos que são classificados em outros tipos como defeitos de Ambiguidade, Inconsistência e Informação Estranha. Logo, o uso dos *checklists* serviram como guia durante a atividade inspeção para alunos e especialista. E os mesmos defeitos encontrados pelo especialista também foram percebidos pelos alunos.

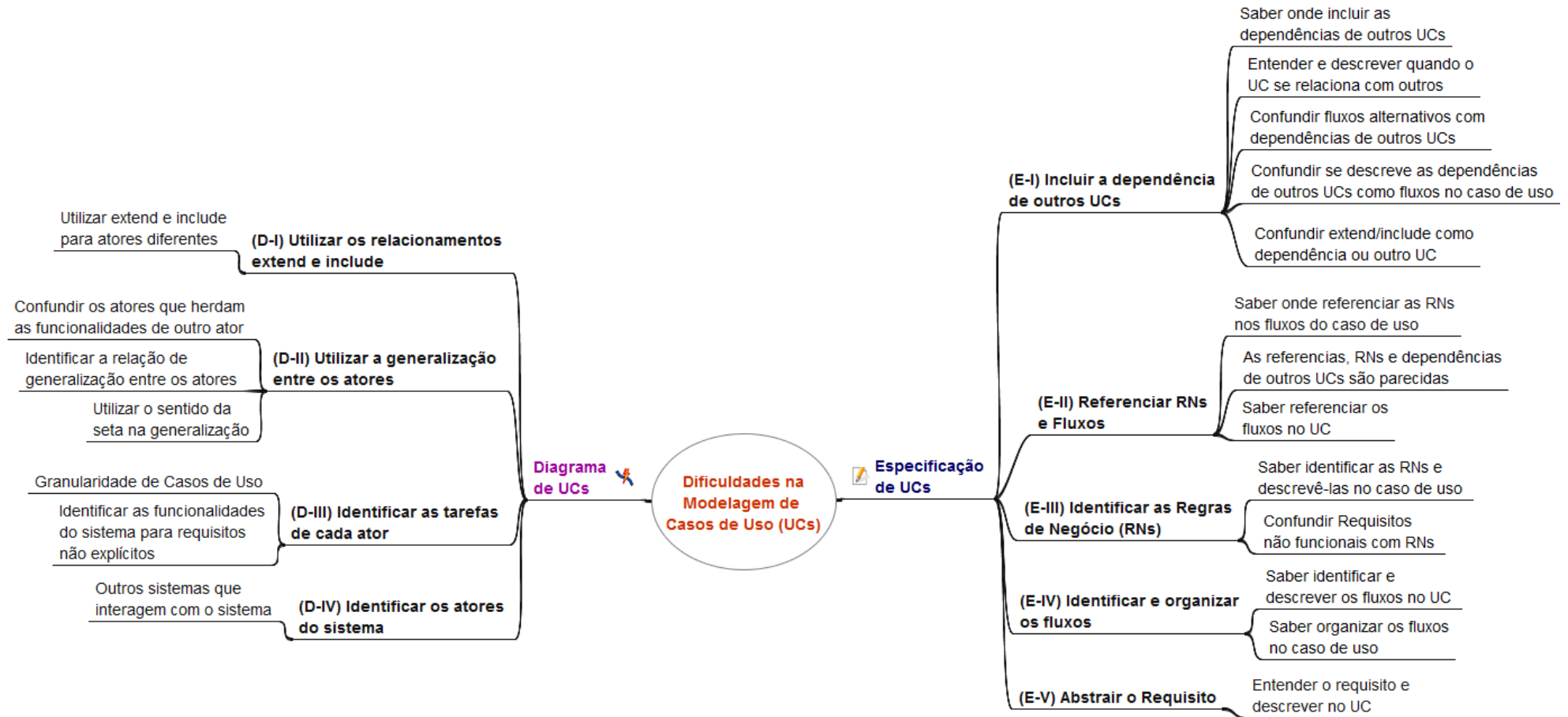


Figura 5.10: Modelo de dificuldades na Modelagem de Casos de Uso.



## 5.5 AMEAÇAS A VALIDADE

Neste estudo foram consideradas cinco principais ameaças que representam um risco de interpretação imprópria dos resultados: (a) itens dos *checklists* utilizados para inspecionar; (b) inspeção em grupo; (c) questionário de dificuldades; (d) viés do pesquisador na interpretação dos dados; e (e) tamanho da amostra. Em relação à primeira ameaça, os itens dos *checklists* podem não cobrir todos os defeitos em modelos de casos de uso. No entanto, estes itens foram revisados por mais dois especialistas e os alunos poderiam incluir novos itens aos *checklists* caso identificassem novos defeitos durante a inspeção dos modelos.

Em relação à inspeção em grupo, as equipes que inspecionaram os modelos seguiram o processo de inspeção de Sauer *et al.* (2000), ou seja, cada membro da equipe fez a inspeção individual e depois fizeram a discriminação em grupo, como sugere o processo de inspeção. Assim cada aluno realizou a atividade de inspecionar casos de uso, colocando em prática os conceitos aprendidos em sala de aula. Com relação ao questionário de dificuldades, as questões foram objetivas e algumas sub-questões direcionavam possíveis dificuldades que os alunos poderiam encontrar durante a modelagem dos casos de uso. Contudo, estas sub-questões que direcionavam possíveis dificuldades foram baseadas nas observações realizadas durante os exercícios práticos de modelagem em sala de aula e durante a inspeção dos modelos.

Com relação ao viés do pesquisador na interpretação dos dados qualitativos, outro pesquisador com alto nível de conhecimento em análise qualitativa e especificação de UC acompanhou todas as etapas da análise dos dados qualitativos. Em relação ao tamanho e a homogeneidade da amostra, todos estudantes eram de uma mesma instituição e a quantidade de participantes não é ideal do ponto de vista estatístico. Contudo, pequenas amostras são um problema conhecido em experimentos na área de Engenharia de Software e difíceis de serem superados (Conte *et al.* 2007). Assim, há limitação nos resultados, sendo estes considerados indícios e não conclusivos.

## 5.6 CONSIDERAÇÕES FINAIS

Este capítulo apresentou um estudo experimental em que foi analisada a percepção dos alunos sobre o ensino de casos de uso e as dificuldades iniciais no aprendizado deste modelo. Além disso, foi analisado o uso da inspeção em casos de uso para compreender melhor o seu auxílio no ensino de casos de uso.

Para responder a questão de pesquisa RQ1 (*Quais as principais dificuldades de alunos durante a modelagem do diagrama e especificação de casos de uso?*), os resultados da análise quantitativa mostraram que os principais defeitos detectados na inspeção dos modelos de casos de uso estavam relacionados à sintaxe do modelo, por exemplo, no diagrama de casos de uso, houve defeitos como, uso incorreto dos relacionamentos *extend*, *include* e generalização.

Na especificação de casos de uso, a maioria dos defeitos estava relacionada à Omissão, como fluxos e/ou regras de negócio necessários para a completude do caso de uso. No entanto, os tipos de defeitos encontrados nos modelos de casos de uso foram melhor compreendidos pelas respostas no questionário, pois os alunos relataram as dificuldades que tiveram durante a especificação de casos de uso; por exemplo, houve dúvidas em saber onde incluir as dependências de outros casos de uso nos fluxos. Outra dúvida era saber identificar os fluxos e regras de negócio abstraídos dos requisitos para descrevê-los no caso de uso.

Além disso, as dificuldades identificadas no estudo anterior foram comparadas com os resultados deste estudo, mostrando que algumas dificuldades foram semelhantes, por exemplo, na categoria “Abstrair o Requisito” do modelo, a dificuldade entender o requisito para descrever no UC também ocorreu com os participantes que estavam aprendendo casos de uso. A partir dos resultados obtidos neste estudo foi possível aprimorar o modelo de dificuldades e criar novas categorias relacionadas às dificuldades em especificação textual do UC como, a categoria “Incluir a dependência de outros UCs”. Também foram criadas novas categorias relacionadas às dificuldades na modelagem do diagrama de UC, já que no estudo anterior o foco não foi no diagrama de UC.

As novas categorias adicionadas ao modelo foram: (D-I) utilizar os relacionamentos de *extend* e *include*; (D-II) utilizar a generalização entre os atores; (D-III) identificar as tarefas de cada ator; e (D-IV) identificar os atores do sistema. Percebeu-se que as dificuldades em saber utilizar os relacionamentos de *extend* e *include* no diagrama podem estar correlacionadas com as dificuldades de incluir a dependência de outros UCs na especificação textual do UC, pois os participantes que relataram esta dificuldade na modelagem do diagrama também relataram dificuldades semelhantes na especificação textual do UC.

O uso de inspeção nos modelos de casos de uso mostrou que os defeitos encontrados pelas equipes foram equivalentes aos do especialista. Desse modo, visando responder a questão de pesquisa RQ2 (*A atividade de inspecionar modelos de casos de uso pode ajudar alunos a compreenderem melhor casos de uso?*), observou-se que a prática da inspeção com o uso dos *checklists* auxiliaram tanto inspetores novatos quanto o especialista em casos de uso durante a

inspeção de defeitos nos modelos de casos de uso. Além disso, a inspeção apresentou aos alunos uma visão sobre os defeitos que podem ocorrer nos modelos de casos de uso e como evitá-los no futuro. Com os resultados desse estudo conclui-se que a atividade de inspecionar modelos de casos de uso ajudou no aprendizado de casos de uso.

Assim, a inspeção (leitura nos diagramas e especificação de casos de uso) feita pelos alunos ajudou-os na compreensão sobre como construir casos de uso (escrever/modelar melhor). Com estes resultados, apresentando-se indícios de que ler casos de uso ajuda a escrevê-los melhor, ou seja, a prática da inspeção em modelos de UCs pode auxiliar no ensino/aprendizagem de casos de uso.

## **CAPÍTULO 6 – MODELO DE DIFICULDADES EM MODELAR CASOS DE USO E SUA VALIDAÇÃO**

*Este capítulo apresenta a versão final do modelo de dificuldade em modelar casos de uso, consolidado a partir dos resultados dos estudos experimentais realizados anteriormente na Fase I (Exploratória) e II (Investigativa). Além disso, é apresentado o planejamento, execução e resultados do 4º estudo experimental realizado com o objetivo de validar as dificuldades apresentadas no modelo, seguindo a Fase III (Avaliativa) da metodologia desta pesquisa. Desta forma, as dificuldades identificadas nos estudos anteriores e sumarizadas no modelo foram avaliadas por estudantes de três diferentes instituições.*

### **6.1 INTRODUÇÃO**

Nos capítulos anteriores (Capítulo 4 e Capítulo 5), foram apresentados estudos experimentais que tinham como objetivo identificar dificuldades percebidas durante na modelagem de UC (diagrama e especificação). Assim, a partir da análise qualitativa dos dados destes estudos, foi possível identificar, comparar e aprimorar o modelo de dificuldades em modelar casos de uso. O modelo teve a versão inicial criada e apresentada no final do capítulo 4 (2º estudo), nesta versão foi apresentado cinco categorias e as dificuldades associadas a estas categorias. Contudo, esta versão limitou-se a dificuldades percebidas durante a especificação textual de casos de uso, devido o foco do 2º estudo ser a identificação detalhada das dificuldades em especificação.

No final do capítulo 5 (3º estudo) apresentou-se uma nova versão do modelo de dificuldades, nesta versão foram criadas quatro categorias relacionadas há dificuldades percebidas durante a modelagem do diagrama de casos de uso. Além disso, foi identificada uma nova categoria para a especificação textual (E-I: Incluir a dependência de outros UCs) e assim, associadas as dificuldades desta categoria. As outras categorias já identificadas na primeira versão do modelo foram mantidas, ou seja, as dificuldades percebidas no 2º estudo também foram percebidas no 3º estudo. Neste estudo (3º) o foco foi identificar as dificuldades tanto em diagramas de UCs como na especificação.

As dificuldades sumarizadas e visualmente apresentadas no modelo têm por objetivo: (a) auxiliar acadêmicos e profissionais de software que utilizam casos de uso na academia e/ou na prática industrial; (b) sugerir práticas para melhorar o processo de ensino/aprendizagem dos alunos na modelagem de casos de uso, baseadas na literatura; e (c) auxiliar os pesquisadores

como base para futuras pesquisas na área, como na construção de técnicas voltadas para o ensino de casos de uso.

Neste capítulo será apresentada a versão final do modelo em que tomou como base a junção da primeira versão e segunda versão do modelo apresentadas anteriormente nos capítulos 4 e 5. Além disso, será apresentado um estudo experimental que foi executado com o propósito de validar o modelo de dificuldades em modelar casos de uso, conforme apresentada na metodologia deste trabalho (Capítulo 1, Subseção 1.4). Este 4º estudo avaliou se as dificuldades apresentadas no modelo são percebidas pelos alunos de graduação dos cursos de Ciência da Computação e Sistemas de Informação de outras três universidades do país localizadas em duas diferentes regiões do país. Para isso, os participantes responderam a um questionário com o objetivo de obter os dados quantitativos e qualitativos sobre as dificuldades apresentadas no modelo. Os resultados quantitativos, em conjunto com os resultados qualitativos, proporcionaram informações relevantes para a identificação de oportunidades de melhoria no modelo.

Este capítulo está organizado da seguinte maneira: na Seção 6.2 é apresentado o modelo de dificuldades em casos de uso e seu processo de construção. Na Seção 6.3 é descrito como o estudo experimental foi planejado e conduzido. A Seção 6.4 apresenta a análise dos resultados quantitativos encontrados no estudo, além de apresentar a percepção dos participantes sobre as dificuldades na modelagem de casos de uso. A Seção 6.5 apresenta as discussões com relação aos resultados. A Seção 6.6 apresenta as ameaças à validade deste estudo. E, por fim, a Seção 6.7 apresenta as considerações finais deste capítulo.

## **6.2 MODELO DE DIFICULDADES EM CASOS DE USO**

A Figura 6.1 mostra o modelo de dificuldades consolidado a partir dos resultados dos estudos experimentais realizados anteriormente. Estas dificuldades foram relatadas por alunos de duas universidades (UFAM e UFRJ) durante a modelagem dos diagramas e da especificação de casos de uso em dois estudos experimentais realizados em contextos diferentes já detalhados nos capítulos 4 e 5.

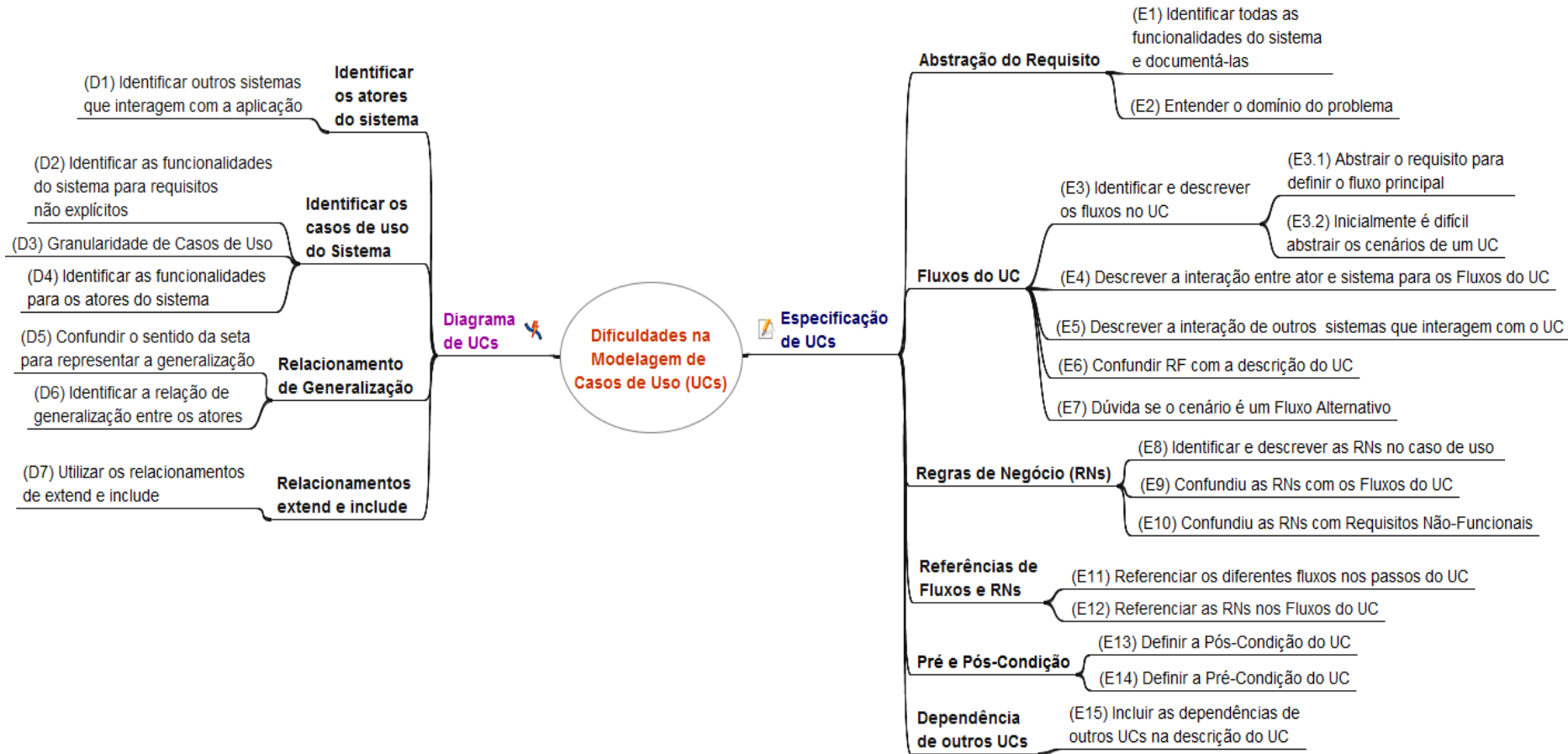


Figura 6.1: Modelo de Dificuldades em Modelagem de Casos de Uso.

Com base nos resultados qualitativos dos estudos realizados, o processo de construção do modelo seguiu duas etapas: (i) extração das dificuldades; (ii) criação das categorias e o relacionamento das dificuldades às categorias. Primeiramente, foram (i) extraídas todas as dificuldades identificadas na análise qualitativa dos estudos (2º e 3º). Em seguida, as dificuldades encontradas foram relacionadas e agrupadas em categorias de acordo com o tipo de dificuldade (ii), como mostram a Tabela 6.1 e a Tabela 6.2. Por exemplo, as dificuldades relacionadas aos fluxos do UC, foram agrupadas na categoria *Fluxos do UC*.

Além disso, cada dificuldade recebeu um identificador, por exemplo, para as dificuldades relacionadas ao Diagrama de UCs recebeu um identificador de D1, D2,...,D7 (dificuldade no Diagrama de UC – D1), e assim sucessivamente. Da mesma maneira as dificuldades relacionadas à Especificação de UCs receberam um identificador de E1, E2,...,E15 (dificuldade na Especificação do UC – E1). Esses identificadores serviram para visualizar e organizar os tipos de dificuldades no modelo. Assim, é possível compreender quais os tipos de dificuldades que estão relacionadas ao Diagrama e a Especificação de UC.

O processo da análise qualitativa dos estudos e a construção do modelo de dificuldades foram realizados por um pesquisador e, posteriormente, conferido e discutido por outros dois pesquisadores. Assim, serão apresentadas as evidências relacionadas às dificuldades na modelagem do diagrama de UC e em seguida as evidências relacionadas às dificuldades na especificação textual do UC.

Inicialmente a Tabela 6.1 mostra um resumo com todas as citações referentes às dificuldades apresentadas nas categorias do modelo para o Diagrama de UCs. A primeira coluna apresenta as categorias do modelo. A segunda coluna mostra os códigos referentes às dificuldades agrupadas na categoria do modelo para o Diagrama. A terceira coluna apresenta as citações relacionadas às dificuldades (no diagrama de UC) mencionadas pelos participantes. Por fim, a quarta coluna mostra o código do estudo experimental em que a dificuldade foi identificada durante a análise qualitativa, por exemplo, o código “ES03” refere-se ao 3º estudo experimental realizado e o código “ES02” refere-se ao 2º estudo experimental, como apresentados nas Tabelas 6.1 e 6.2.

**Tabela 6.1:** Resumo das Dificuldades relacionadas a modelagem do Diagrama de UC.

<b>Categorias do Modelo</b>	<b>Códigos (Dificuldades)</b>	<b>Citações</b>	<b>Estudo</b>
<b>Identificar os Atores do Sistema</b>	<i>“Identificar outros sistemas que interagem com a aplicação” (D1)</i>	<i>“Em identificar os atores do sistema, principalmente nas vezes em que há comunicação com outros sistemas, pois ficou mais fácil de aprender que os atores são sempre pessoas, e isso quase ficou como regra para mim, quando há atores que são outros sistemas, sinto certa dificuldade.” – ES03_P29</i>	ES03

Categories do Modelo	Códigos (Dificuldades)	Citações	Estudo
		“Às vezes esqueço de sistemas externos que interagem com o sistema que estou modelando.” – ES03_P30	
Identificar os Casos de Uso do Sistema	“Identificar as funcionalidades do sistema para requisitos não estão explícitos” (D2)	“A minha maior dificuldade é atentar aos requisitos que não estão “explícitos”. O que acho que tem mais a ver com o levantamento de requisitos em si.” – ES03_P21	ES03
	“Granularidade de Casos de Uso” (D3)	“Mesmo sabendo as funcionalidades e atores, como representá-los, e dividir em casos de uso” – ES03_P23	ES03
	“Identificar as funcionalidades para os atores do sistema” (D4)	“Foi complicado entender o que cada caso necessitava para que o caso se apresentasse completo, ou seja, as ações necessárias para tal completude.” – ES03_P28  “Talvez a dificuldade ocorra em identificar as principais tarefas que serão representadas no diagrama.” – ES03_P3	ES03
Relacionamento de Generalização	“Confundir o sentido da seta para representar a generalização” (D5)	“Eu sempre esqueço o sentido da seta de generalização: aponta do específico para o geral ou do geral para o específico?” – ES03_P10  “Lembrar o lado correto da “setinha”. – ES03_P25  “No caso da generalização, tive dificuldade em entender o padrão generalização/especialização (direção da seta).” – ES03_P30	ES03
	“Identificar a relação de generalização entre os atores” (D6)	“Não sabia identificar a relação de generalização que havia entre os atores, não sei se este fato é decorrente da subjetividade que existe na especificação dos casos de uso ou da má elaboração do mesmo.” – ES03_P3  “As principais dúvidas que eu tinha era na hora de verificar se uma generalização de atores era realmente necessária e, se sim, criar um ator mais genérico que descrevesse bem as funcionalidades utilizadas por ele.” – ES03_P6  “Entender quais atores fazem a mesma função o que diferencia cada um deles no sistema.” – ES03_P8  “Muitas vezes me confundi com qual ator herdaria as funcionalidades de um determinado ator.” – ES03_P11  “A princípio não entendia como um ator incorporava as ações de outro(s) ator(es), mesmo com aquela identificação de uma seta(linha) ligando um ator no outro.” – ES03_P15  “Na prática foi mais complicado do que pareceu, definir o que cada ator terá e quem irá herdar dele foram as maiores dificuldades.” – ES03_P18	ES03
Relacionamento extend e include	“Utilizar os relacionamentos de extend e include” (D7)	“No início era confuso a utilização dos relacionamentos extend e include, pois não sabia como relacionar um caso de uso com o outro de forma correta ou que realmente necessitasse” – ES03_P1  “Às vezes me confundo no tipo de relacionamento entre casos de uso.” – ES03_P16  “De certa forma a forma de utilização dos relacionamentos podem causar certa confusão quanto a forma de compreender o direcionamento das setas de cada relacionamento e a forma de leitura que se dá para cada relacionamento estabelecido.” – ES03_P20  “Para atores diferentes, não tenho certeza de como utilizar os relacionamentos existentes quando há tarefas que não são comuns aos dois, mas necessitam de algum extend/include uma da outra.” – ES03_P22  “Eu me confundi um pouco com os includes e extends.” – ES03_P27	ES03



A Tabela 6.2 mostra todas as citações referentes às dificuldades apresentadas nas categorias do modelo relacionadas às dificuldades em Especificação de UCs. A tabela está organizada da mesma forma que foram apresentadas as dificuldades relacionadas ao Diagrama de UC.

**Tabela 6.2:** Resumo das Dificuldades relacionadas a especificação de UC.

<b>Categorias do Modelo</b>	<b>Códigos (Dificuldades)</b>	<b>Citações</b>	<b>Estudo</b>
<b>Abstração do Requisito</b>	<b>“Identificar todas as funcionalidades do sistema e documentá-las” (E1)</b>	<p>“Acho que algumas funcionalidades podem ter faltado (...) A gente percebeu, esqueceu e eu percebi agora de novo.” ES02_P16</p> <p>“Faltou o caso de uso atender a todos os cenários (...) sendo uma questão de omissão mesmo, de não ter no caso de uso.” – ES02_P2</p>	ES02
	<b>“Entender o domínio do problema” (E2)</b>	<p>“Eu acho que a maior dificuldade é o entendimento do problema, mas a dificuldade de escrever não muito.” – ES02_P5</p> <p>“Se for de uma área que eu domine não tenho dificuldade, mas se for de uma área que eu não tenha muito conhecimento acho que eu teria dificuldade sim. (...) quando é uma coisa que você não conhece muito bem fica meio complicado, você não consegue imaginar como pode ser aquele passo a passo você se embola” – ES02_P9</p> <p>“O problema não foi a transcrição para o caso de uso mas o problema que não ficou muito claro pra mim.” – ES02_P12</p>	ES02
		<p>“A única dificuldade foi em entender o requisito e descrever no caso de uso. Foi uma dificuldade que tive no início e após várias práticas, fui conseguindo entender e interpretar melhor o que o sistema gostaria que fizesse.” – ES03_P1</p> <p>“Tive que parar um tempo maior para entender o que a especificação estava pedindo, outro tempo para saber quais fluxos estavam relacionados com aquele caso de uso e assim por diante (...)” – ES03_P3</p>	ES03
<b>Fluxos do UC</b>	<b>“Identificar e descrever os fluxos no caso de uso” (E3)</b>	<p>“Eu tive mais dificuldade na parte dos fluxos alternativos e exceções porque tudo depende de como foi redigido os requisitos né.” – ES02_P11</p> <p>“Inicialmente tem a dificuldade de escrever e identificar quais fluxos alternativos serão descritos.” – ES02_P16</p> <p>“Isso de primeira foi complicado, organizar. Esse daqui é meu fluxo principal e esse daqui é meu alternativo, essa coisa é uma forma mágica pelo menos pra mim. O fluxo principal eu não conseguia ver muito bem o que ia ser o alternativo, quando ia dar erro.” – ES02_P20</p>	ES02
		<p>“No início tive dificuldades em entender quando algo era um fluxo alternativo. Tive dificuldades em especificar CRUD's no início, e seus fluxos alternativos.” – ES03_P13</p> <p>“Houve dificuldade em saber organizar para se saber, que o que estava em minha mente, se encaixava em qual fluxo exatamente.” – ES03_P19</p> <p>“Em fluxos como CRUD, em que temos vários possíveis caminhos, é confuso saber qual será o principal e quais serão os alternativos.” – ES03_P32</p>	ES03
	<b>“Descrever a interação entre o ator e sistema para os fluxos do UC” (E4)</b>	<p>“A maior dificuldade foi decidir como seria a interação entre o sistema e o ator, eu acho que não tem algo específico (...) para todos os fluxos.” – ES02_P15</p>	ES02

Categories do Modelo	Códigos (Dificuldades)	Citações	Estudo
		<i>"A maior dificuldade foi decidir como seria a interação entre o sistema e o ator, para todos os fluxos"</i> – ES02_P15	
	<b>"Descrever a interação de outros sistemas que interagem com o UC"</b> (E5)	<i>"(...) quando tem sistemas externos interagindo com o sistema aí eu acho meio difícil de colocar isso. Porque quando eu estou especificando eu não sei que tipo de resposta aquilo vai me dar (...) Saber como ele ia devolver e saber como eu ia tratar isso no meu caso de uso."</i> – ES02_P22	ES02
	<b>"Confundir RF com a descrição do UC"</b> (E6)	<i>"Às vezes dependendo da situação eu sinto um pouco de dificuldade de passar para a descrição os requisitos funcionais e as vezes eu me confundo um pouco com a própria descrição."</i> – ES02_P13	ES02
	<b>"Dúvida se o cenário é um fluxo alternativo"</b> (E7)	<i>"Às vezes tem alternativo mas isso é um alternativo mesmo, vai ser só um, eu fico com um pouco de dúvida mas, o fluxo principal a chance de leitura é um pouco menor, você imagina como o usuário vai se comportar pela aplicação, eu acho que quando você está fazendo as regras as vezes eu tenho um pouco de dificuldades."</i> – ES02_P5  <i>"(...) eu tenho muita insegurança se eu entendi da forma correta. Por exemplo, tem um requisito e aí depois que eu desenvolvo da forma que eu imaginei que seria no caso, mas eu tenho dúvida em relação a fluxos alternativos, só isso mesmo"</i> – ES02_P23	ES02
Regras de Negócio (RNs)	<b>"Identificar e descrever as RNs no caso de uso"</b> (E8)	<i>"A maior dificuldade estaria em detalhar as regras de negócio e o funcionamento. Por exemplo, se no cadastro há uma validação de dados, é necessário explicar como ela é feita, a senha tem que ter no mínimo 8 caracteres? O meu endereço é obrigatório?"</i> – ES03_P9  <i>"Nem todos os requisitos apresentam as regras de negócio claramente, o que leva ao esquecimento na elaboração da especificação."</i> – ES03_P16	ES03
		<i>"Eu tento imaginar as regras de negócio do caso de uso (...) Por exemplo validar o login a senha, o cpf ou alguma coisa que pode ter mais de tantas letras. Imaginar as regras de negócio eu acho que foi isso, imaginar as restrições do caso de uso na hora"</i> – ES02_P14  <i>"As regras de negócio ficam meio intrínsecas você não consegue separar ou tem dificuldade pra separar e identificar uma regra de negócio que está relacionada aquele caso de uso (...) Talvez havia um necessidade de identificar qual era a regra de negócio que estava relacionado ao caso de uso e o que é uma regra de negócio de fato."</i> – ES02_P6  <i>"Dificuldade na (...) parte de identificar e imaginar as restrições no caso de uso para colocar como regra de negócio."</i> – ES02_P14	ES02
	<b>"Confundi as RNs com os fluxos do UC"</b> (E9)	<i>"Muitas vezes eu confundo um pouco fluxos e regras, porque quando eu olho, isso daqui faz parte do fluxo principal, mas também parece uma regra de negócio, então eu confundo as vezes... eu tento imaginar um pouco a situação para tentar abstrair quem é quem."</i> – ES02_P20	ES02
	<b>"Confundi as RNs com Requisitos Não-Funcionais"</b> (E10)	<i>"(...) essa parte não está muito clara para mim, mas a gente está conseguindo aos poucos incluir (...) Particularmente eu confundo requisitos não funcionais com regras de negócio."</i> – ES02_P1  <i>"(...) às vezes você fica meio o que é regra de negócio é requisito não funcional. Então quando você ver pela primeira vez você confunde onde vai localizar aquela regra. É funcional ou não funcional?"</i> – ES02_P3	ES02
		<i>"Com relação as regras de negócio, a dúvida que eu mais tive foi na hora se era ou não necessário pegá-lo de requisitos não-funcionais, pois eles parecem descrever as mesmas coisas."</i> – ES03_P6	ES03

Categories do Modelo	Códigos (Dificuldades)	Citações	Estudo
		<i>“Por vezes, tenho dúvidas se deveria escrever requisitos não-funcionais como regras de negócio de alguns casos de uso ou apenas listá-los no documento de visão.” – ES03_P30</i>	
Referências de Fluxos e RNs	<i>“Referenciar os diferentes fluxos nos passos do UC” (E11)</i>	<i>“(…) esquecerem de incluir o fluxo alternativo e esqueciam de linkar.” – ES02_P13</i>  <i>“Eu fico com dúvida, há mas o fluxo alternativo vai vir na hora que o ator informa ou na hora que o sistema válida? Então eu acho que a minha maior dificuldade é onde colocar o fluxo alternativo.” – ES02_P23</i>	ES02
		<i>“A principal dúvida foi em como referenciar fluxos que podem acontecer a qualquer momento durante a execução do caso de uso.” – ES03_P26</i>	ES03
	<i>“Referenciar as RNs nos fluxos do UC” (E12)</i>	<i>“Inicialmente, ficava confusa se as regras de negócio só eram referenciadas juntamente aos fluxos de exceção ou poderiam estar também nos fluxos principal e alternativos, junto à referência ao fluxo de exceção” – ES03_P5</i>	ES03
Pré e Pós-Condição	<i>“Definir a pós-condição do UC” (E13)</i>	<i>“O que eu ficava com dificuldade era aquele negócio de pós-condição depois que tinha feito o caso de uso, depois que aconteceu aquele comportamento, eu tinha que pensar nas pós-condições daquilo então até hoje eu sinto muita dificuldade.” – ES02_P23</i>	ES02
	<i>“Definir a pré-condição do UC” (E14)</i>	<i>“A dificuldade que eu tive logo quando eu fui fazer a primeira especificação era com relação as pré-condições, as pós-condições, o fluxo principal, (...) eu nem lembrava que tinha fluxo alternativo. Aí eu percebi que era um negócio um pouco maior do que estava parecendo e aí tive que imaginar realmente, delinear o sistema para aquela ação né.” – ES02_P14</i>	ES02
Dependência de outros UCs	<i>“Incluir as dependências de outros casos de uso na descrição do UC” (E15)</i>	<i>“Em certas ocasiões não sabia como encaixar as devidas dependências. Um exemplo que havia sido visto em sala era o de pagamento via cartão, boleto ou depósito. Como escolher o caminho? Onde eles ficam expostos para os projetistas?” – ES03_P7</i>  <i>“Tive dificuldade em saber onde referenciar um extend ou include na especificação do caso de uso.” – ES03_P11</i>  <i>“A dificuldade foi entender se um caso de uso que estende outro deveria ser descrito como fluxo alternativo na especificação do caso de uso estendido, enquanto fosse identificado com associação extend no diagrama, ou se, além de estar identificado com associação extend no diagrama, deveria ser descrito como um outro caso de uso na especificação e referenciado nos fluxos do caso de uso estendido.” – ES03_P10</i>  <i>“Não sei se include/extend pode ser feito em qualquer parte da especificação ou apenas no começo.” – ES03_P12</i>  <i>“Os extend e include geraram dúvidas quanto à forma de uso adequado, uma questão de memorizar melhor como é feito o uso de cada uma das dependências.” – ES03_P20</i>  <i>“O questionamento era, devemos especificar uma forma padrão para realizar essa parte do caso de uso, como ‘pagar com dinheiro’ e referenciar os extends em fluxos alternativos ou podemos fazer uma estrutura estilo ‘if/else’ para referenciar os extends, sem uma forma principal de execução?” – ES03_P26</i>	ES03

### 6.3 ESTUDO EXPERIMENTAL PARA AVALIAÇÃO DO MODELO

Uma vez que o modelo de dificuldades estava consolidado, fez-se necessário avaliá-lo, verificando se as dificuldades identificadas através da análise qualitativa dos estudos

apresentados nos capítulos anteriores eram válidas. Assim, o estudo experimental apresentado neste capítulo foi dividido em três etapas: (a) Planejamento, (b) Execução do Estudo e (c) Análise dos dados. Nas próximas subseções será apresentado em mais detalhes como foi realizada cada etapa do estudo.

### 6.3.1 Planejamento do Estudo

Realizou-se um estudo experimental com o objetivo de validar as dificuldades apontadas no modelo de dificuldades em casos de uso (diagrama e especificação). Com o propósito de validar estas dificuldades, alunos dos cursos de Ciência da Computação (CC) e Sistemas de Informação (SI) de três universidades responderam a um questionário, a fim de validá-las. Através dos resultados deste questionário, foi possível validar quais dificuldades apontadas no modelo são relevantes para melhorar o ensino/aprendizagem de casos de uso. Conseqüentemente, as percepções encontradas podem auxiliar na adoção de outras práticas, abordagens ou métodos no ambiente acadêmico. Portanto, este estudo buscou responder as seguintes questões de pesquisa:

- RQ1 – Com qual frequência as dificuldades apresentadas no modelo são percebidas pelos alunos após a modelagem de casos de uso?
- RQ2 – Existem dificuldades que estão relacionadas ao uso de casos de uso e que não foram contempladas pelo modelo?

### 6.3.2 Execução do Estudo

No início do estudo, esclareceu-se aos alunos a intenção do pesquisador de analisar as respostas do questionário em uma pesquisa acadêmica. Após estes esclarecimentos, todos os alunos assinaram os TCLEs (Termos de Consentimento Livre e Esclarecido), cedendo os dados para análise. Os recursos necessários para analisar a percepção dos alunos em relação às dificuldades na modelagem de casos de uso (diagrama e especificação) apresentadas no modelo são definidos e detalhados a seguir:

- a) **Ambiente de Execução** - o ambiente de execução do estudo foi realizado na sala de aula, das respectivas universidades.

**Tabela 6.3:** Resumo dos Participantes do Estudo.

UNINORTE (TURMA A)																							
Par.	PA1	PA2	PA3	PA4	PA5	PA6	PA7	PA8	PA9	PA10	PA11	PA12	PA13	PA14	PA15	PA16	PA17	PA18	PA19	PA20	PA21	PA22	PA23
Exp.	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	S	N	N	N
Proj.	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	I	A	A	A
Par.	PA24	PA25	PA26	PA27	PA28	PA29	PA30	PA31	PA32	PA33	PA34	PA35	PA36	PA37	PA38	PA39	PA40	PA41	PA42	PA43	PA44	PA45	PA46
Exp.	N	N	N	S	N	N	N	S	N	S	N	N	N	N	N	N	N	N	N	N	S	N	N
Proj.	A	A	A	I	A	A	A	I	A	I	A	A	A	A	A	A	A	A	A	A	I	A	A

UTFPR (TURMA B)																				
Par.	PB1	PB2	PB3	PB4	PB5	PB6	PB7	PB8	PB9	PB10	PB11	PB12	PB13	PB14	PB15	PB16	PB17	PB18	PB19	PB20
Exp.	N	N	N	S	N	N	N	N	N	N	N	N	N	N	N	N	N	N	S	N
Proj.	A	A	A	I	A	A	A	A	A	A	A	A	A	A	A	A	A	A	I	A
PUCRS (TURMA C)																				
Par.	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12	PC13	PC14	PC15	PC16	PC17	PC18		
Exp.	N	N	N	N	S	N	S	N	N	N	N	N	N	N	N	N	N	N		
Proj.	A	A	A	A	I	A	I	A	A	A	A	A	A	A	A	A	A	A		

**Legenda:**  
**Par.** – Código do Participante;  
**Exp.** – Experiência em Desenvolvimento de Software com UCs? Sim (S) – Na Indústria; ou Não (N) – Na Academia;  
**Proj.** – Participação em projetos de software: Indústria (I) ou Academia (A);

b) **Participantes** – ao todo participaram do estudo oitenta e quatro (84) estudantes. Na Universidade Tecnológica Federal do Paraná (UTFPR), foram vinte (20) alunos de graduação do curso de CC, matriculados na disciplina Engenharia de Software. Na Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS), foram dezoito (18) alunos de graduação do curso de SI, matriculados na disciplina Engenharia de Requisitos. No Centro Universitário do Norte (Uninorte) em Manaus, foram quarenta e seis (46) alunos de graduação do curso de CC, matriculados na disciplina Análise de Sistemas.

Vale ressaltar que os alunos tinham experiência prévia em modelagem de casos de uso em sala de aula, pois estavam finalizando disciplinas que abordavam assuntos relacionados a casos de uso (diagrama e especificação). A Tabela 6.3 apresenta um resumo sobre a experiência dos participantes no desenvolvimento de software com casos de uso.

c) **Questionário sobre o modelo de dificuldades em casos de uso** – Foi aplicado um questionário com a finalidade de avaliar as dificuldades relatadas pelos alunos no uso de UCs após o desenvolvimento de trabalhos práticos realizados nas disciplinas de cada curso. Desta forma, os alunos foram convidados a responder o questionário no final da disciplina. Os participantes forneceram suas respostas em uma escala de cinco pontos, baseados nos questionários aplicados por Laitenberger e Dreyer (1998). No entanto, os itens foram adaptados permanecendo um valor neutro (usando os seguintes valores: sempre, quase sempre, algumas vezes, raramente e nunca) para que os participantes expressassem mais claramente sua tendência com relação a dificuldades mais ou menos frequentes durante a modelagem do diagrama e especificação do caso de uso.

No mesmo questionário, os alunos responderam questões relacionadas à sua experiência em desenvolvimento de software com UCs, as dificuldades relacionadas à modelagem de UCs (diagrama e especificação). O questionário era composto por três partes, como detalhado na Tabela 6.4. Na parte I, o aluno incluía sua experiência com relação ao

desenvolvimento de software com casos de uso, na indústria ou na academia, se já havia participado de projetos de desenvolvimento de software e de quantos. Nas partes II e III, o aluno poderia incluir uma nova dificuldade que não estava listada nas questões apresentadas sobre dificuldades em diagrama/especificação do UC.

**Tabela 6.4:** Questionário do Modelo de Dificuldades (Ver APÊNDICE H)

<b>Parte I – Background do Aluno</b>	<b>Objetivo</b>
<p><b>Q1</b> – Você já participou de projetos de desenvolvimento de software que utilizaram casos de uso (na indústria ou na academia)?  <input type="checkbox"/> SIM   <input type="checkbox"/> NÃO</p> <p><b>Q2</b> – De quantos projetos você participou?  <input type="checkbox"/> Somente projetos em sala de aula  <input type="checkbox"/> De 1 a 4 projetos na indústria  <input type="checkbox"/> Mais de 5 projetos na indústria</p>	<p>Identificar a experiência dos participantes com relação à: (i) desenvolvimento de software com UCs e (ii) a quantidade de projetos de software que já havia participado utilizando UCs.</p>
<b>Parte II – Dificuldades no Diagrama de UCs</b>	
<p><b>D1</b> – Foi difícil identificar outros sistemas que interagem com o sistema que será desenvolvido?  <b>D2</b> – Foi difícil identificar os casos de uso quando as necessidades de negócio não estão claras?  <b>D3</b> – Foi difícil dividir os casos de uso para atender as funcionalidades do sistema?  <b>D4</b> – Foi difícil identificar quais funcionalidades se relacionam com cada ator do sistema?  <b>D5</b> – Você já confundiu o sentido da seta para representar uma generalização?  <b>D6</b> – Você teve dúvidas em identificar os relacionamentos de generalização entre os atores do sistema?  <b>D7</b> – Você teve dúvidas ao utilizar os relacionamentos de <i>extend</i> e <i>include</i> (extensão e inclusão)?</p>	<p>Avaliar as respostas das dificuldades em Diagramas de UC, em uma escala de cinco pontos, variando de sempre, quase sempre, algumas vezes, raramente e nunca.</p>
<b>Parte III – Dificuldades na Especificação de UCs</b>	
<p><b>E1</b> – Foi difícil entender a funcionalidade do sistema (para o caso de uso que você escreveu)?  <b>E2</b> – Foi difícil entender o domínio/contexto da aplicação que seria desenvolvida?  <b>E3</b> – Foi difícil identificar e descrever os fluxos do caso de uso?  <b>E4</b> – Foi difícil descrever a interação entre o ator e sistema nos fluxos do caso de uso?  <b>E5</b> – Foi difícil descrever a interação de outros sistemas que interagem com o sistema que será desenvolvido?  <b>E6</b> – Você já confundiu a descrição/especificação dos casos de uso com o detalhamento dos requisitos funcionais do sistema?  <b>E7</b> – Você teve dúvidas se o fluxo que você descreveu para o caso de uso era um fluxo alternativo?  <b>E8</b> – Foi difícil identificar as regras de negócio e descrevê-las nos casos de uso?  <b>E9</b> – Você já confundiu as regras de negócio com os fluxos do caso de uso?  <b>E10</b> – Você já confundiu as regras de negócio com os requisitos não funcionais do sistema?  <b>E11</b> – Foi difícil referenciar os fluxos alternativos nos passos do caso de uso?  <b>E12</b> – Foi difícil referenciar as regras de negócio nos passos do caso de uso?  <b>E13</b> – Foi difícil definir a(s) pré-condição (ões) do caso de uso?  <b>E14</b> – Foi difícil definir a(s) pós-condição (ões) do caso de uso?  <b>E15</b> – Foi difícil incluir as dependências (<i>include</i> e/ou <i>extend</i>) de outros casos de uso na descrição do caso de uso que você escreveu?</p>	<p>Avaliar as respostas das dificuldades em Especificação de UC, em uma escala de cinco pontos, variando de sempre, quase sempre, algumas vezes, raramente e nunca.</p>

Vale ressaltar que as perguntas do questionário foram desenvolvidas por um pesquisador e revisadas por três pesquisadores. Após a revisão, foi realizado um estudo piloto, com dois estudantes de graduação e com experiência prévia em UCs (na indústria e academia), a fim de validar a clareza das perguntas e o tempo total que seria gasto para responder o questionário.

O resultado do estudo piloto mostrou que foi necessário realizar alterações em algumas questões para o participante obter uma melhor compreensão e clareza das questões. Por exemplo, algumas das questões relacionadas às dificuldades em especificação de UCs foram unificadas, a questão “Foi difícil identificar as funcionalidades de um sistema e descrevê-las na especificação do Caso de Uso?” e a questão “Foi difícil entender os requisitos definidos para o sistema (no contexto do Caso de Uso)?” ficou em uma única questão “E1 - Foi difícil entender a funcionalidade do sistema (para o caso de uso que você escreveu)?” (ver Tabela 6.4).

### **6.3.3 Análise dos Dados**

Nesta etapa, foi realizada a atividade de coleta e análise dos resultados obtidos no estudo. Vale ressaltar que a coleta dos dados (aplicação dos questionários) foi realizada por outros pesquisadores que eram professores das disciplinas nas referidas universidades. Após a coleta dos dados, os questionários respondidos pelos participantes foram analisados pelo pesquisador. Para a realização deste processo, retirou-se a referência dos participantes e foi inserido um código para representá-los, por exemplo, o participante “José” recebeu o código P1.

Para a validação do questionário foi conduzida uma análise fatorial pelo método de extração dos eixos principais, com rotação Varimax, revelando algumas correlações entre os fatores (Laros e Pasquali, 2005; Huang *et al.* 2006). O método de rotação Varimax é o método mais amplamente aceito e utilizado (Laros e Pasquali, 2005; Huang *et al.* 2006). Procedeu-se ainda à análise de consistência interna das questões apresentadas no questionário de dificuldades pelo alfa de Cronbach (Keller, 1987). O coeficiente do alfa de Cronbach indica indiretamente o grau em que um conjunto de itens mede um único fator de qualidade. Desta forma, buscou-se saber se o questionário de dificuldades mede o mesmo fator de qualidade. Geralmente, valores de alfa de Cronbach variando de 0,70 a 0,95 são relatados como aceitáveis (DeVellis, 2003).

Para analisar se os itens do questionário de dificuldades podem ser submetidos a um processo de análise fatorial foi utilizado o índice de Kaiser-Meyer-Olkin (KMO) e o teste de esfericidade de *Barlett*, pois são os testes mais utilizados (Brown, 2006; Hair *et al.* 2009).

Segundo Hair *et al.* (2009) ambos os testes indicam o grau de ajuste dos dados à análise fatorial, ou seja, qual é o nível de confiança esperado dos dados quando do seu tratamento. O índice KMO mede que a amostra entre valores 0,5 e 1,0 é adequada para a análise fatorial, ou seja, um valor de índice menor que 0,5 indicam que a análise fatorial pode ser inadequada (Brown, 2006; Hair *et al.* 2009). E o teste de esfericidade de Barlett também indica se a análise fatorial é apropriada com valores de nível de significância  $<0,05$  são considerados aceitáveis (Brown, 2006; Hair *et al.* 2009).

Além disso, para análise dos comentários adicionais incluídos pelos participantes no questionário utilizaram-se procedimentos de codificação do método Grounded Theory (GT) (Strauss e Corbin, 1998). Vale ressaltar que todo o processo de codificação foi revisado por outro pesquisador, a fim avaliar os resultados obtidos na análise dos dados.

## **6.4 RESULTADOS OBTIDOS**

Nesta seção serão apresentados os resultados quantitativos do estudo e a percepção dos participantes sobre as dificuldades do Modelo.

### **6.4.1 Resultados Quantitativos**

Na Subseção 6.4.1.1 são apresentados os resultados quantitativos que mostra o total de dificuldades respondidas para o diagrama de UC. Na Subseção 6.4.1.2 são apresentados os resultados quantitativos relativos às dificuldades em especificação de UC, obtidas a partir das respostas dos alunos no questionário. Por fim, a Subseção 6.4.1.3 apresenta os resultados da análise fatorial exploratória.

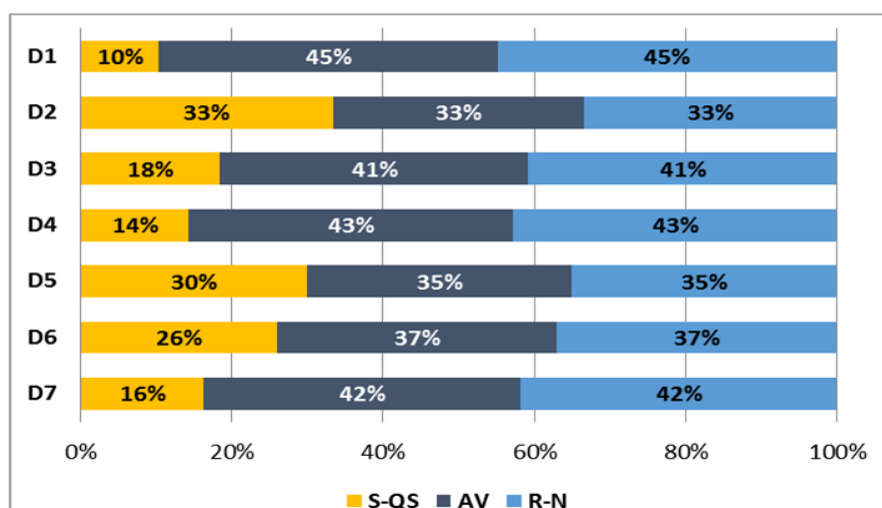
#### *6.4.1.1 Resultados do Questionário – Dificuldades em Diagramas de UCs*

A Figura 6.2 mostra a quantidade de respostas dos participantes quanto às dificuldades ocorridas durante a modelagem no Diagrama de UC. Na figura o eixo horizontal refere-se à intensidade das percepções dos participantes com relação as dificuldades apresentadas (Sempre e Quase Sempre (S-QS), Algumas Vezes (AV), Raramente e Nunca (R-N)) para cada questão. As respostas S e QS de cada questão foram unificadas (somadas) para mostrar que os participantes já tinham tido alguma das dificuldades apresentadas no questionário durante a modelagem do caso de uso, como apresentado na primeira coluna do gráfico abaixo. Da mesma forma as respostas R e N foram unificadas (somadas) para mostrar que os participantes não tinham dificuldade em algumas das questões, como apresentadas na terceira coluna do gráfico. Nas barras foi inserido o total (%) de respostas de todos participantes do estudo.



Para fins do estudo considerou-se que tanto as respostas da escala S-QS quanto da escala AV, em todos esses casos, o participante relatou tendência em ter dificuldades. Desta forma, observa-se na Figura 6.2 que na soma das respostas (S, QS e AV) as dificuldades ocorreram acima de 50%. Porém, algumas dificuldades na modelagem do diagrama de UC alcançaram um percentual acima de 60% na soma das respostas S-QS e AV, tais como:

- **(D2) Identificar os UCs quando as necessidades de negócio não estão claras – 66%** do total de participantes afirmaram que esta dificuldade ocorre S-QS (33%) e AV (33%).
- **(D5) Confundir o sentido da seta para representar uma generalização – 65%** dos participantes responderam S-QS (30%) e AV (35%).
- **(D6) Identificar os relacionamentos de generalização entre os atores do sistema – 63%** dos participantes responderam S-QS (26%) e AV (37%).



**Figura 6.2:** Total de dificuldades em Diagrama de UC respondida por todos os participantes

A Figura 6.3 apresenta um Ranking de respostas por turma, no qual considerou-se como dificuldades relevantes as questões que estão acima de 50% (representadas na cor vermelho) para a construção do diagrama de UCs, estas questões somam as respostas S-QS e AV (Sempre e Quase Sempre, mais as respostas para Algumas Vezes). As questões que somam as respostas R-N (Raramente e Nunca) estão representadas na cor verde na Figura 6.3. Estas questões que tiveram respostas menores que 50% foram questões que os participantes não consideraram como dificuldades.

Conforme apresentado na Figura 6.3 observou-se que a maioria dos participantes que responderam S-QS somando com as respostas AV (Sempre, Quase Sempre – S-QS e Algumas

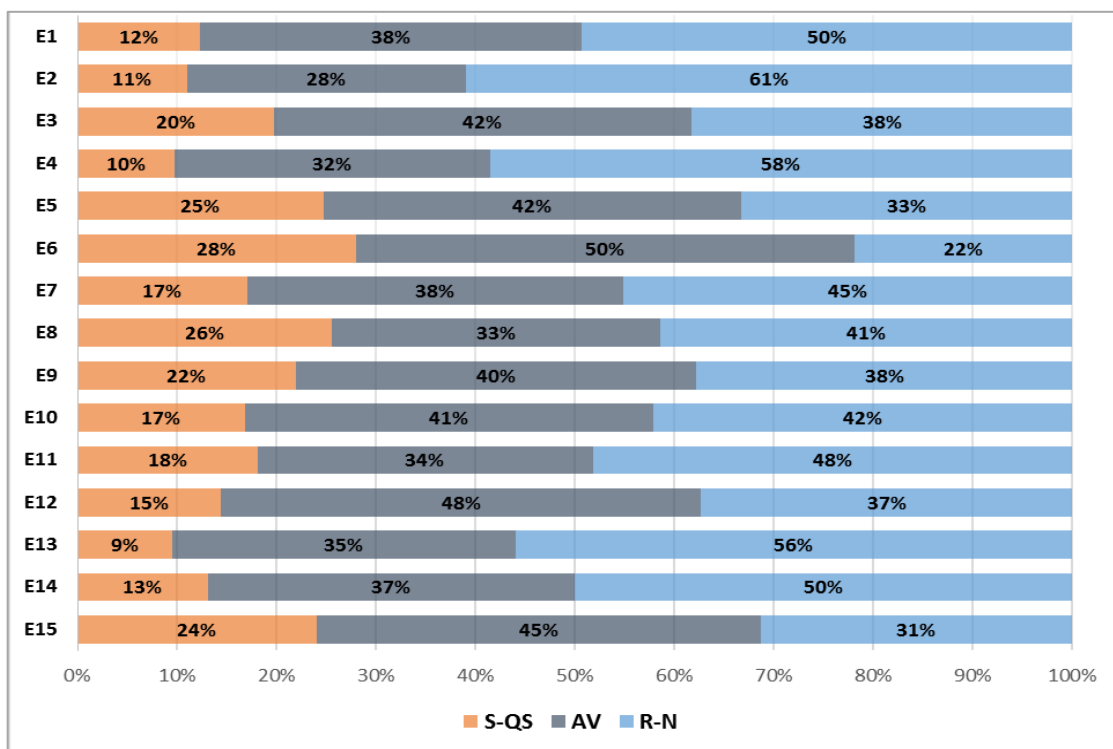
Vezes-AV) para a questão **D2** foi a Turma B (95% dos participantes) e a Turma C (83% dos participantes). Na questão **D5** as mesmas turmas se destacaram nestas respostas, com 70% cada turma (Turmas B e C) e 52% para a Turma A. Estas turmas são de universidades que estão localizadas na mesma região do país, porém uma é federal (Turma B) e outra é particular (Turma C). Apesar de serem instituições diferentes (Federal e Particular), os participantes apresentaram o mesmo percentual de dificuldade. Além disso, percebeu-se que a questão **D4** (*Foi difícil identificar quais funcionalidades se relacionam com cada ator do sistema?*) não foi uma dificuldade significativa para os participantes de cada turma (Turmas A, B e C). Contudo, na soma total das turmas a questão D4 mostrou um percentual de 57%, ou seja, pode ser considerada uma dificuldade.

TURMAS															
Turma A				Turma B				Turma C				TODOS			
Ranking	Questões	S-QS e AV	R-N	Ranking	Questões	S-QS e AV	R-N	Ranking	Questões	S-QS e AV	R-N	Ranking	Questões	S-QS e AV	R-N
1º	D2	78	22	1º	D2	95	5	1º	D2	83	17	1º	D2	67	33
2º	D1	70	30	2º	D5	70	30	2º	D1	78	22	2º	D5	65	35
3º	D6	61	39	3º	D1	65	35	3º	D5	71	29	3º	D6	63	37
4º	D7	59	41	4º	D7	60	40	4º	D7	61	39	4º	D3	59	41
5º	D3	59	41	5º	D3	50	50	5º	D6	56	44	5º	D7	58	42
6º	D5	52	48	6º	D6	47	53	6º	D3	47	53	6º	D4	57	43
7º	D4	37	63	7º	D4	32	68	7º	D4	28	72	7º	D1	55	45

**Figura 6.3:** Ranking das questões de dificuldades em Diagrama de UC respondidas pelos participantes.

#### 6.4.1.2 Resultados do Questionário – Dificuldades em Especificação de UCs

A Figura 6.4 mostra a quantidade de respostas dos participantes nos questionários sobre as dificuldades ocorridas durante a Especificação do UC. Da mesma forma que foi apresentada para as questões relacionadas ao Diagrama de UC, também considerou-se na primeira coluna do gráfico (abaixo) a soma das respostas S e QS de cada questão para mostrar que os participantes já tinham tido alguma das dificuldades durante a especificação do UC. A segunda coluna do gráfico mostra que os participantes já tinham tido algumas vezes (AV) a dificuldade. A terceira coluna do gráfico mostra a soma das respostas R e N, mostrando que os participantes não tinham dificuldade em algumas questões. Nas barras do gráfico foi inserido o total (%) de respostas de todos participantes do estudo.



**Figura 6.4:** Total de dificuldades em Especificação de UC respondidas pelos participantes.

Para fins do estudo considerou-se que tanto as respostas da escala S-QS quanto da escala AV, em todos esses casos, o participante relatou tendência em ter dificuldades durante a especificação do UC. Desta forma, observa-se na Figura 6.4 que na soma das respostas (S, QS e AV) as dificuldades em especificação de UCs ocorreram acima de 50%. No entanto, algumas dificuldades alcançaram respostas acima de 60% na soma das respostas S-QS e AV, como:

- **(E6)** *Confundir RF com a descrição do UC* – **78%** do total de participantes afirmaram que a dificuldade ocorre S-QS (28%) e AV (50%).
- **(E15)** *Incluir as dependências de outros UCs na descrição do UC* – **69%** do total de participantes afirmou que a dificuldade ocorre S-QS (24%) e AV (45%).
- **(E5)** *Descrever a interação de outros sistemas que interagem com o UC* – **67%** do total de participantes afirmou que esta dificuldade ocorre S-QS (25%) e AV (42%).
- **(E12)** *Referenciar as RNs nos fluxos do UC* – **63%** do total de participantes afirmou que a dificuldade ocorre S-QS (15%) e AV (48%).
- **(E9)** *Confundiu as RNs com os fluxos do UC* – **62%** do total de participantes afirmou que esta dificuldade ocorre S-QS (22%) e AV (40%).
- **(E3)** *Identificar e descrever os fluxos no UC* – **62%** do total de participantes afirmou que a dificuldade ocorre S-QS (20%) e AV (42%).

TURMAS															
Turma A		%		Turma B		%		Turma C		%		TODOS		%	
Ranking	Questões	S-QS e AV	R-N	Ranking	Questões	S-QS e AV	R-N	Ranking	Questões	S-QS e AV	R-N	Ranking	Questões	S-QS e AV	R-N
1º	E6	80	20	1º	E5	84	16	1º	E15	83	17	1º	E6	78	22
2º	E7	70	30	2º	E6	80	20	2º	E6	72	28	2º	E15	69	31
3º	E12	67	33	3º	E15	75	25	3º	E10	67	33	3º	E5	67	33
4º	E3	65	35	4º	E12	75	25	4º	E8	67	33	4º	E12	63	37
5º	E9	64	36	5º	E9	75	25	5º	E5	67	33	5º	E9	62	38
6º	E11	62	38	6º	E3	75	25	6º	E9	44	56	6º	E3	62	38
7º	E15	60	40	7º	E14	65	35	7º	E12	39	61	7º	E8	59	41
8º	E10	60	40	8º	E4	60	40	8º	E3	39	61	8º	E10	58	42
9º	E14	59	41	9º	E13	55	45	9º	E1	39	61	9º	E7	55	45
10º	E5	59	41	10º	E11	55	45	10º	E7	28	72	10º	E11	52	48
11º	E8	57	43	11º	E8	55	45	11º	E2	28	72	11º	E1	51	49
12º	E13	52	48	12º	E10	45	55	12º	E11	22	78	12º	E14	50	50
13º	E1	48	52	13º	E7	45	55	13º	E14	11	89	13º	E13	44	56
14º	E4	45	55	14º	E1	44	56	14º	E13	11	89	14º	E4	41	59
15º	E2	44	56	15º	E2	37	63	15º	E4	11	89	15º	E2	39	61

**Figura 6.5:** Ranking das questões de dificuldades em Especificação de UC respondidas pelos participantes.

Conforme mostra a Figura 6.5 observou-se que a maioria dos participantes que responderam acima de 74% (soma das respostas de S, QS e AV) para estas questões **E6**, **E15**, **E5**, **E12**, **E9** e **E3** foram alunos da Turma B. No entanto, para a questão **E6** as Turmas A e B tiveram o mesmo percentual (80%). Na questão **E15**, a Turma C ficou em primeiro lugar com 83% das respostas dos participantes para essa questão. Além disso, percebeu-se que as questões **E2** (39%), **E4** (41%), **E13** (44%) e **E14** (50%) não foram dificuldades significativas para os participantes, como mostra o ranking de respostas por turmas na soma geral (TODOS) apresentado na Figura 6.5. Neste ranking considerou-se as respostas que estão acima de 50% como dificuldades relevantes para a especificação de UCs, as questões que somam as respostas S-QS e AV (na cor vermelho). As questões que somam as respostas R e N estão na cor verde, foram às respostas que os participantes não consideraram como dificuldades relevantes.

#### 6.4.1.3 Análise Fatorial Exploratória (AFE)

Realizou-se uma AFE para verificar possíveis relações entre as dificuldades apresentadas no modelo com auxílio da ferramenta JASP versão 0.8.0.1<sup>4</sup>. Segundo Tabachnick e Fidell (1996), a AFE é uma técnica aplicada a um conjunto de variáveis sobre o qual se deseja

<sup>4</sup> JASP A Fresh Way to Do Statistics - <https://jasp-stats.org/>

descobrir quais destas formam subconjuntos coerentes e independentes entre si. A AFE também proporciona valiosas informações sobre a estrutura multivariada de um instrumento de medição, identificando os construtos teóricos (Laros e Pasquali, 2005). Segundo Laros e Pasquali (2005), na AFE os fatores são estimados para explicar as covariâncias entre as variáveis observadas, portanto, os fatores são considerados como as causas das variáveis observadas.

Para verificar se as questões apresentadas no questionário de dificuldades eram confiáveis foi feito o teste de Cronbach (Keller, 1987). Inicialmente o teste foi realizado nas sete (7) questões relacionadas ao Diagrama de UC, em seguida o teste foi feito nas quinze (15) questões relacionadas a Especificação de UC. No primeiro teste, o resultado mostrou que o alfa de Cronbach obteve um bom índice (0,70) de confiabilidade para as questões, sobre as dificuldades em diagramas de casos de uso, como apresentado na Tabela 6.5.

**Tabela 6.5:** Resultado do Teste Estatísticas de confiabilidade de Cronbach para dificuldades em Diagrama de UC, retirado da ferramenta JASP.

Cronbach's $\alpha$
0.70
<i>Nota.</i> A escala consiste nos itens: D1, D2, D3, D4, D5, D6, D7

O resultado do segundo teste, para as quinze questões relacionadas a especificação de UC, mostrou que o alfa de Cronbach apresentou um bom índice (0,86) de confiabilidade como apresentado na Tabela 6.6.

**Tabela 6.6:** Resultado do Teste Estatísticas de confiabilidade de Cronbach para dificuldades em Especificação de UC, retirado da ferramenta JASP.

Cronbach's $\alpha$
0.86
<i>Nota.</i> A escala consiste nos itens: E1, E2, E3, E4, E5, E6, E7, E8, E9, E10, E11, E12, E13, E14, E15

Geralmente, os valores de alfa de Cronbach, variando de 0,70 a 0,95 são relatados como aceitáveis (DeVellis, 2003). Logo, os resultados dos testes de confiabilidade de Cronbach mostrou que as questões (relacionadas ao diagrama e especificação do UC) aplicadas no estudo foram confiáveis e válidas.

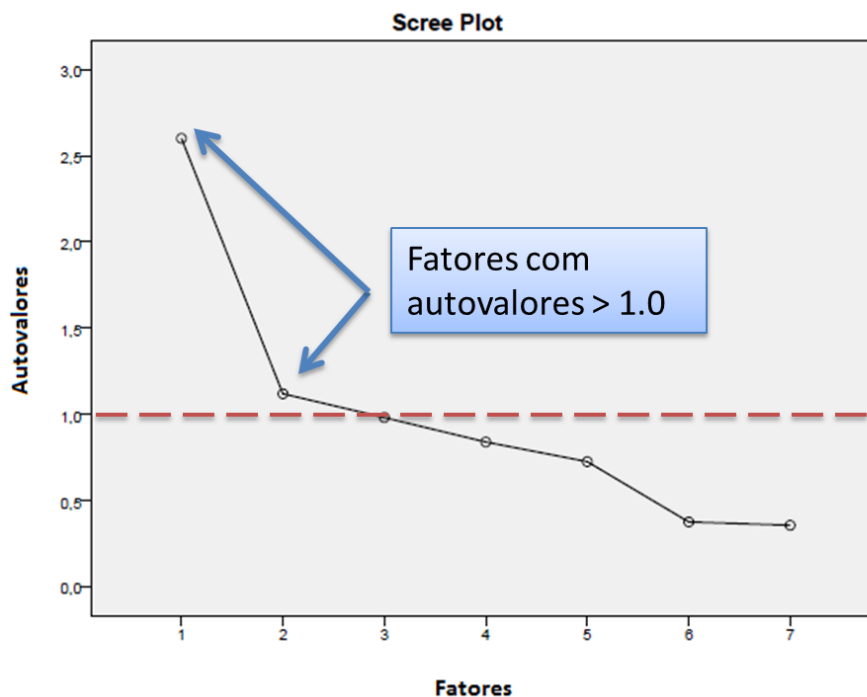
Além disso, utilizou-se o índice de Kaiser-Meyer-Olkin (KMO) e o teste de esfericidade de Bartlett (Brown, 2006; Hair *et al.* 2009) para verificar se os itens do questionário de dificuldades poderiam ser submetidos a uma análise fatorial. Analisando o conjunto de questões do questionário de dificuldades (Diagrama e Especificação de UC) obteve-se um índice de

KMO de 0,756 e um nível de significância menor que 0,001 indicando que a análise fatorial pode ser realizada, como mostra a Figura 6.6.

Kaiser-Meyer-Olkin Measure of Sampling Adequacy.		,756
Bartlett's Test of Sphericity	Approx. Chi-Square	695,199
	df	231
	Sig.	0,001

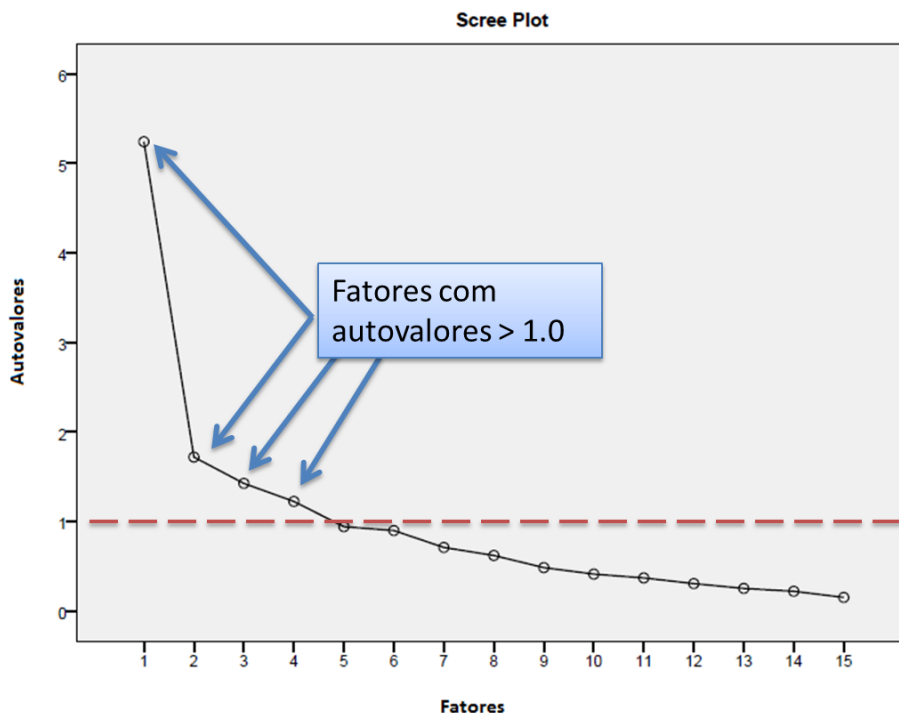
**Figura 6.6:** Testes KMO e Bartlett para o Índice de Dificuldades em UCs.

Além disso, foi utilizado o critério de Kaiser-Guttman para realizar a análise fatorial, pois geralmente é o método mais usado (Brown, 2006). Este método afirma que o número de fatores é igual para o número de autovalores acima de 1 (Brown, 2006). O autovalor refere-se ao valor da variância de todas as questões que é explicado por um fator (Brown, 2006). O gráfico Scree Plot (Figura 6.7 e Figura 6.8) mostra o autovalor para cada número de fator (representando cada questão), ou seja, no eixo Y são os valores dos autovalores e no eixo X mostra o número sequencial de fatores (Laros e Pasquali, 2005). Segundo Laros e Pasquali (2005) o gráfico Scree Plot separa os fatores triviais de fatores não triviais por intermédio de uma inspeção visual, em que os fatores triviais são considerados os autovalores que estão acima de 1 e os não triviais estão abaixo desse valor e não são considerados na análise fatorial.



**Figura 6.7:** Gráfico de autovalores e fatores relacionados a dificuldades em Diagrama de UC.

Seguindo o Kaiser-Guttman (Brown, 2006), os resultados mostraram que 2 fatores devem ser mantidos para as questões relacionadas ao Diagrama de UC (Figura 6.7) e 4 fatores foram mantidos para as questões relacionadas a Especificação de UC (Figura 6.8).



**Figura 6.8:** Gráfico de autovalores e fatores relacionados a dificuldades em Especificação de UC.

Para identificar as cargas fatoriais dos itens extraídos da AFE foi utilizado o método dos eixos principais, com rotação Varimax com Kaiser Normalization, sendo este o método de rotação mais amplamente aceito e utilizado (Cohen, 1998). O maior fator de carga de cada item está em negrito, indicando a que fator o item está mais relacionado, como mostra a Tabela 6.7. Primeiramente serão apresentados os itens relacionados as dificuldades em Diagrama de UC. Na primeira coluna, são apresentadas as categorias do modelo relacionadas às dificuldades em Diagrama de UC. A segunda coluna, são as questões que os participantes responderam. As demais colunas apresentam os resultados da AFE realizada.

**Tabela 6.7:** Análise fatorial exploratória sobre as dificuldades no Diagrama de UC, retirado da ferramenta JASP.

<i>Categorias do Modelo</i>	<i>Questões</i>	<i>Fator 1</i>	<i>Fator 2</i>	<i>Fator 3</i>	<i>Fator 4</i>	<i>Singularidade</i>
<b>Identificar os atores do sistema</b>	D1 - Foi difícil identificar outros sistemas que interagem com o sistema que será desenvolvido?				<b>0.583</b>	0.619

<i>Categorias do Modelo</i>	<i>Questões</i>	<i>Fator 1</i>	<i>Fator 2</i>	<i>Fator 3</i>	<i>Fator 4</i>	<i>Singularidade</i>
<b>Identificar os casos de uso do Sistema</b>	<b>D2</b> - Foi difícil identificar os casos de uso quando as necessidades de negócio não estão claras?			<b>0.808</b>		0.249
	<b>D3</b> - Foi difícil dividir os casos de uso para atender as funcionalidades do sistema?		<b>0.903</b>			0.005
	<b>D4</b> - Foi difícil identificar quais funcionalidades se relacionam com cada ator do sistema?		<b>0.519</b>			0.587
<b>Relacionamento de Generalização</b>	<b>D5</b> - Você já confundiu o sentido da seta para representar uma generalização?	<b>0.613</b>				0.403
	<b>D6</b> - Você teve dúvidas em identificar os relacionamentos de generalização entre os atores do sistema?	<b>0.852</b>				0.196
<b>Relacionamentos de extend e include</b>	<b>D7</b> - Você teve dúvidas ao utilizar os relacionamentos de extend e include (extensão e inclusão)?					0.920

Observou-se que dois fatores (Fator 1 e Fator 2) tiveram dificuldades agrupadas e dois fatores (Fator 3 e Fator 4) apresentaram dificuldades separadas. Em relação ao Fator 1 mostrou as questões D5 e D6 correlacionadas, ou seja, os participantes que responderam a questão D5, também responderam a questão D6. Estas questões estão relacionadas com o uso da generalização no diagrama que estão agrupadas na mesma categoria (Relacionamento de Generalização) do Modelo, o que reforça tal relação. Quanto ao Fator 2 os participantes responderam de forma conjunta as questões D3 e D4. Estas questões (D3 e D4) estão agrupadas na mesma categoria *Identificar os Casos de Uso do Sistema* do modelo, porém os participantes responderam que não tiveram tanta dificuldade em dividir os casos de uso e identificar as funcionalidades para cada ator do sistema, reforçando a correlação destas questões. Com relação ao Fator 3, apenas a questão D2 foi extraída, e apesar de não estar relacionada com nenhuma outra dificuldade (questão) foi a questão que os participantes responderam ter maior dificuldade, ou seja, mais de 65% de respostas como apresentado no ranking de dificuldades em diagrama de UC (Figura 6.3). Com relação ao Fator 4, também houve apenas uma questão (D1) que não teve nenhuma outra dificuldade (questão) relacionada, porém foi uma questão que os participantes responderam não possuir tanta dificuldade, sendo a última questão na escala do ranking de dificuldades em Diagrama de UC. A questão D7, não foi extraída em nenhum fator, no entanto, apresentou o maior valor na singularidade (0.920), ou seja, foi uma questão que mostrou uma dificuldade comum a maioria dos participantes.



Da mesma forma que foi apresentado os itens relacionados ao Diagrama de UC, a Tabela 6.8 apresenta as cargas fatoriais dos itens relacionados à Especificação de UC em que o maior fator de carga de cada item, está em negrito, indicando a que fator o item está mais relacionado. Na primeira coluna, são apresentadas as categorias do modelo relacionadas às dificuldades em Especificação de UC. A segunda coluna, são as questões que os participantes responderam e as demais colunas apresentam os resultados da AFE realizada.

**Tabela 6.8:** Análise fatorial exploratória sobre as dificuldades na Especificação de UC, retirado da ferramenta JASP.

<i>Categorias do Modelo</i>	<i>Questões</i>	<i>Fator 1</i>	<i>Fator 2</i>	<i>Fator 3</i>	<i>Fator 4</i>	<i>Singularidade</i>
<b>Abstração do Requisito</b>	<b>E1</b> - Foi difícil entender a funcionalidade do sistema (para o caso de uso que você escreveu)?		<b>0.680</b>			0.406
<b>Regras de Negócio</b>	<b>E10</b> - Você já confundiu as regras de negócio com os requisitos não funcionais do sistema?			<b>0.553</b>		0.541
<b>Referências de Fluxos e RNs</b>	<b>E11</b> - Foi difícil referenciar os fluxos alternativos nos passos do caso de uso?					0.549
	<b>E12</b> - Foi difícil referenciar as regras de negócio nos passos do caso de uso?					0.609
<b>Pré e Pós-Condição</b>	<b>E13</b> - Foi difícil definir a(s) pré-condição (ões) do caso de uso?				<b>0.874</b>	0.182
	<b>E14</b> - Foi difícil definir a(s) pós-condição (ões) do caso de uso?				<b>0.819</b>	0.188
<b>Dependência de outros UCs</b>	<b>E15</b> - Foi difícil incluir as dependências (inclui e/ou extend) de outros casos de uso na descrição do caso de uso que você escreveu?		<b>0.556</b>			0.580
<b>Abstração do Requisito</b>	<b>E2</b> - Foi difícil entender o domínio/contexto da aplicação que seria desenvolvida?		<b>0.571</b>			0.498
<b>Fluxos do UC</b>	<b>E3</b> - Foi difícil identificar e descrever os fluxos do caso de uso?	<b>0.547</b>				0.520
	<b>E4</b> - Foi difícil descrever a interação entre o ator e sistema nos fluxos do caso de uso?	<b>0.667</b>				0.352
	<b>E5</b> - Foi difícil descrever a interação de outros sistemas que interagem com o sistema que será desenvolvido?		<b>0.624</b>			0.541
	<b>E6</b> - Você já confundiu a descrição/especificação dos casos de uso com o detalhamento dos requisitos funcionais do sistema?					0.759
	<b>E7</b> - Você teve dúvidas se o fluxo que você descreveu para o caso de uso era um fluxo alternativo?	<b>0.809</b>				0.299
<b>Regras de Negócio</b>	<b>E8</b> - Foi difícil identificar as regras de negócio e descrevê-las nos casos de uso?			<b>0.671</b>		0.517

<i>Categorias do Modelo</i>	<i>Questões</i>	<i>Fator 1</i>	<i>Fator 2</i>	<i>Fator 3</i>	<i>Fator 4</i>	<i>Singularidade</i>
	E9 - Você já confundiu as regras de negócio com os fluxos do caso de uso?			0.677		0.449

A Tabela 6.8 mostra que quatro fatores foram extraídos da AFE, havendo dificuldades que se correlacionaram e outras que ficaram separadas. Com relação ao Fator 1 inclui um grupo de três questões (E3, E4 e E7) sobre as dificuldades na descrição dos fluxos do caso de uso. Isso pode ter ocorrido devido os participantes, terem dificuldades durante a especificação dos fluxos do caso de uso, reforçando tal relação. Com relação ao Fator 2 inclui um grupo de quatro questões E1, E15, E2 e E5, mas apenas duas questões (E1 e E2) fazem parte da mesma categoria (Abstração do Requisito) o que reforça o agrupamento destas questões. Porém, as questões E2 e E15 são de categorias diferentes do Modelo (Dependência de outros UCs e Fluxos do UC) reforçando que estas dificuldades podem ocorrer em contextos diferentes. Além disso, a questão E15 foi a segunda maior dificuldade apresentada no ranking de dificuldades em especificação de UC, ou seja, 67% dos participantes responderam que tiveram dificuldades em incluir a dependência de outros casos de uso na especificação. No entanto, a questão E2, foi a última questão no ranking, mostrando que os participantes não possuem dificuldades em entender o domínio/contexto do problema para o sistema que será desenvolvido. Com relação ao Fator 3 inclui um grupo de três questões E10, E9 e E8 que fazem parte da mesma categoria do modelo (Regras de Negócio), reforçando tal relação, ou seja, são dificuldades diferentes mas que fazem parte do mesmo contexto. Com relação ao Fator 4 inclui duas questões (E13 e E14) que estão correlacionadas e agrupadas na mesma categoria do modelo (Pré e Pós-Condição), o que reforça tal agrupamento. Além disso, os participantes responderam para estas questões (E13 e E14) que não possuem dificuldades em definir a pré e pós-condições no caso de uso, reforçando a correlação destas questões.

Percebeu-se que apesar das questões E11, E12 e E6 não serem extraídas em nenhum fator e não apresentar uma correlação com outras questões, observou-se que estas questões foram dificuldades relevantes para os participantes do estudo. Por exemplo, as questões E11 e E12 tiveram respostas acima de 60% na escala de ranking de dificuldades em especificação de UC apresentada na Figura 6.5. Além disso, a questão E6 apresentou maior valor na singularidade (0.759). Assim, a questão E6 pareceu ser uma dificuldade comum para os participantes, reforçando ser a primeira questão na escala de ranking de respostas (apresentado anteriormente na Figura 6.5).

### 6.4.2 Resultados Qualitativos

Após a análise quantitativa, foram analisados os questionários que continham comentários adicionais de dificuldades relatadas pelos participantes, pois desejava-se avaliar se havia novas dificuldades que não estavam descritas no questionário respondido pelos participantes.

Os dados extraídos das respostas adicionais do questionário foram analisados e criaram-se códigos relevantes sobre a percepção dos alunos quanto às dificuldades no diagrama e especificação de UCs. A partir dos códigos, criaram-se as categorias e relacionamentos entre os códigos. O processo de codificação foi revisado por um pesquisador com 15 anos de experiência com o ensino de modelagem e 7 anos de experiência com análise qualitativa.

Na categoria “**Dificuldades na Modelagem do Diagrama de UC**” foram associados códigos que mostram algum tipo de dificuldade relacionada à modelagem do diagrama UC. A Tabela 6.9 mostra os códigos e as respectivas citações. Na coluna “Códigos”, recebem uma numeração, além de estarem relacionados com as dificuldades existentes no modelo e representadas por D1 (Dificuldade de número um no Diagrama de UC), respectivamente. Por exemplo, o código 1 está associado à dificuldade D1 do modelo. Estes resultados confirmam as dificuldades identificadas no estudo anterior (Capítulo 5) relacionadas ao diagrama de UC. Neste resultado, apenas uma nova dificuldade pôde ser identificada: o participante PA10 afirmou que teve dúvida sobre o uso da notação para inclusão de relacionamentos no diagrama de casos de uso. Esta nova dificuldade pode ser adicionada em uma nova categoria no modelo, por exemplo, categoria de dificuldades em “Notação”.

**Tabela 6.9:** Citações sobre dificuldades durante a modelagem do diagrama de UC.

Códigos (Diagrama)		Citações
1 (D1)	<i>Dificuldade em saber quando um sistema pode ser um ator no diagrama de UC</i>	<i>“Saber quando um sistema pode ser um ator, ou seja, quando interage com outros sistemas.” – PA41</i>
2 (D4)	<i>Dificuldade em saber qual funcionalidade/UC deve ser incluída no diagrama de UC</i>	<i>“Saber qual funcionalidade, caso de uso é relevante para colocar no diagrama.” – PB18</i>
3 (D4)	<i>Dificuldade em identificar os stakeholder do sistema</i>	<i>“Dificuldade em identificar os stakeholder do sistema.” – PC01</i>
4 (D5)	<i>Dificuldade em entender o relacionamento de generalização no Diagrama de UC</i>	<i>“Na hora que o ator herda ou herdado tive dificuldade para entender, mas busquei ajuda para entender” – PA15</i>

Códigos (Diagrama)		Citações
5 (D7)	<i>Dificuldade em utilizar os relacionamentos (extend e/ou include)</i>	<p>“Nosso trabalho chamava outros sistemas, usamos muitos includes e não estava acostumada em utilizar.” – PC13</p> <p>“Nas ligações entre os casos de uso, não sei quando pode e quando não pode.” – PB18</p>
6	<i>“Dúvida na notação do Diagrama de UC”</i>	“A principal dúvida foi diferenciar as linhas de associação e relacionamento.” – PA10

Na categoria “**Dificuldades na Especificação de UC**” foram associados códigos que mostram algum tipo de dificuldade percebida durante a especificação do UC. Da mesma forma, na coluna códigos, recebe uma numeração com as dificuldades já existentes no modelo, representadas por E1 (Dificuldade de número um na Especificação de UC), respectivamente. Na Tabela 6.10 mostra os códigos e citações relacionadas a estas dificuldades.

**Tabela 6.10:** Citações sobre dificuldades durante a especificação do UC.

Códigos (Especificação)		Citações
1 (E1)	<i>Dificuldade em transferir os requisitos para a especificação do UC</i>	“Dificuldade ao passar os requisitos para aplicar na especificação.” – PA11
2 (E3)	<i>Dificuldade nos Fluxos do UC</i>	“Dificuldade em diferenciar fluxo alternativo do fluxo de exceção na especificação de UC.” – PA24
3 (E12)	<i>Dificuldade em associar as RNs</i>	“Na regra de negócio, de como iria associar a um caso de uso.” – PA40

Estes resultados também confirmam as dificuldades sobre especificação de UCs identificadas nos estudos anteriores (Capítulo 4 e Capítulo 5), por exemplo, dificuldades com os fluxos do UC e com regras de negócio. No entanto, os resultados não mostraram nenhuma nova dificuldade identificada para a Especificação de UC.

## 6.5 DISCUSSÃO DOS RESULTADOS

Os resultados obtidos mostram evidências de que a construção do questionário de dificuldades é confiável, válido e aceitável como modelo para a avaliação de dificuldades na modelagem e especificação de UCs. Em termos do teste de confiabilidade de Cronbach, os resultados da análise indicaram uma boa aceitação para as questões sobre as dificuldades em Diagrama (alfa de Cronbach A = 0,70) e Especificação (alfa de Cronbach A = 0,86) de UC, indicando um nível satisfatório de consistência interna do questionário. Desta forma,

respondendo a RQ1, considera-se que os itens do questionário de dificuldades são consistentes em relação à avaliação de dificuldades em modelagem do diagrama e especificação do UC.

Em relação à percepção dos participantes quanto à frequência das dificuldades que ocorreram durante a modelagem do diagrama e especificação do UC (RQ1), identificou-se que as questões que pertencem à mesma categoria, em geral, apresentam maior correlação. No entanto, identificaram-se algumas questões do questionário que não apresentaram nenhuma correlação (por exemplo, D1, D2 e D7; e E11, E12 e E6) o que indica que essas questões são consistentes e captura uma dificuldade específica do modelo. Contudo, às questões que ficaram agrupadas e mostraram ter uma correlação precisam ser revisadas pois talvez capturaram a mesma dificuldade mas de uma forma diferente ou tiveram o mesmo sentido e entendimento para os participantes.

Quanto aos resultados quantitativos sobre as dificuldades no diagrama de UC apresentadas no modelo, as dificuldades que mais ocorreram com os participantes estão relacionadas ao uso do relacionamento de generalização (D5 e D6) e com a identificação das funcionalidades do sistema quando os requisitos não estão explícitos (D2). A AFE confirmou este resultado, mostrando que as dificuldades D5 e D6 estão correlacionadas. No entanto, a dificuldade D2 não teve uma correlação com outra dificuldade, mas foi a dificuldade com maior percentual de respostas dos participantes e a carga fatorial foi excelente (0,808). Assim, a questão D2 mostrou ser uma questão consistente e única que captura uma dificuldade específica do modelo.

Com relação às dificuldades em especificação de UCs na AFE os fatores extraídos, corresponderam a Fluxos do UC (Fator 1), Dependência de outros UCs/Abstração do Requisito (Fator 2), Regras de Negócio (Fator 3) e Pré e Pós-Condição (Fator 4). Com relação ao Fator 1 apresentou um grupo de questões (E3, E4 e E7) que estão relacionadas com dificuldades sobre os fluxos da especificação do UC, o que pode reforçar essa relação. No entanto, a E7 não está correlacionada com nenhuma questão mas apresentou uma excelente carga fatorial (0,809). Desta forma, a questão E7 também mostrou ser uma questão consistente e que captura uma dificuldade específica.

No Fator 2 apresentou um grupo de questões (E1, E15, E2 e E5), mas apenas duas estão correlacionadas (E15 e E2). No entanto, na questão E15 a maioria dos participantes responderam que tiveram dificuldades em incluir a dependência de outros UCs na especificação do UC e na questão E2 os participantes responderam que não tiveram dificuldades em entender

o domínio do problema. Além disso, estas questões são apresentadas no modelo em categorias diferentes.

No Fator 3 apresentou um grupo de três questões (E10, E8 e E9), apesar da E10 não estar correlacionada com outra questão e apresentar uma carga fatorial baixa (0,553), no ranking de dificuldades 58% dos participantes responderam que confundiram as regras de negócio com os requisitos não funcionais do sistema o que justifica ser uma questão independente e consistente para o modelo. Além disso, essas questões deste fator estão agrupadas na mesma categoria do modelo, o que reforça tal relação. As questões E8 e E9 estão na mesma categoria do Modelo (Regras de Negócio) e estão correlacionadas, e obtiveram um nível de carga fatorial muito semelhante (E8 – 0,671 e E9 – 0,677) o que reforça a correlação das questões. Assim, estas questões (E8 e E9) podem ser unificadas para capturar uma única dificuldade.

No Fator 4, as questões E13 e E14 estão relacionadas, contudo, os participantes responderam que não foi difícil definir a pré e pós-condições para os casos de uso e a carga fatorial foi excelente (E13 – 0,874 e E14 – 0,819) o que reforça a relação destas questões. Quanto aos resultados quantitativos para as dificuldades relacionadas à especificação de UCs, as dificuldades que mais ocorreram com os participantes foram: E6, E15, E5, E12, E9 e E3. Além disso, os resultados da AFE, confirma algumas destas questões extraídas nos fatores 1, 2 e 3. Por exemplo, a E5 e E15 foram dificuldades extraídas para o mesmo fator (Fator 2). A AFE também confirma questões que estão agrupadas na mesma categoria do modelo, por exemplo, a E3, E5 e E6 na categoria Fluxos do UC. Além disso, apresenta questões que tiveram correlação com outras dificuldades, por exemplo, a E3 se correlaciona com a E4 que estão na mesma categoria do modelo.

## **6.6 AMEAÇAS À VALIDADE**

Em todos os estudos experimentais, existem ameaças que podem afetar a validade dos resultados. Nesta seção são apresentadas as ameaças relacionadas a este estudo que podem ser classificadas em três categorias: validade interna, validade externa e validade de conclusão (Wöhlin *et al.* 2000).

### **6.6.1 Validade Interna**

Neste estudo foram consideradas duas principais ameaças que representam um risco de interpretação inapropriada dos resultados: (1) Questionário e (2) Fadiga dos participantes. Em relação à primeira ameaça, as questões incluídas no questionário poderiam não ser

compreendidas pelos participantes e assim não obter um resultado esperado. Contudo, o questionário foi avaliado por três pesquisadores. Após a avaliação, realizou-se um estudo piloto com duas estudantes de pós-graduação com experiência prévia em UCs (diagrama e especificação), a fim de verificar se as questões atingiriam seu objetivo. Os resultados do estudo piloto foram avaliados em conjunto com outro pesquisador e foi preciso realizar melhorias em algumas questões para que as mesmas fossem melhor compreendidas pelos participantes. Em relação à segunda ameaça, poderia haver influência nos resultados, devido ao fato das turmas serem no horário noturno e os participantes estarem cansados de suas atividades extraclasses. No entanto, esta ameaça foi controlada utilizando um curto período de tempo para responderem as questões, sendo que os questionários foram aplicados no início da aula. Além disso, através do estudo piloto realizado foi possível avaliar o tempo gasto que seria utilizado para responder ao questionário (entre 10 a 20 minutos) evitando a fadiga dos alunos.

#### **6.6.2 Validade Externa**

Três questões foram consideradas: (1) os participantes não possuem experiência prévia com UCs; (2) grades curriculares diferentes; e (3) background dos professores. Sobre a questão (1), com base nas questões de caracterização, todos os participantes possuíam experiência com UCs em sala de aula e outros na indústria. Em relação à questão (2), os cursos são diferentes, porém as disciplinas possuem conteúdos semelhantes que apresentam os mesmos conceitos sobre casos de uso. Em relação à questão (3), apesar dos professores possuírem níveis de conhecimentos/ experiências diferentes sobre casos de uso, os mesmos ministram conteúdos semelhantes, ou seja, a grade é semelhante, proporcionando para os alunos de universidades de duas regiões diferentes do país o aprendizado do mesmo conhecimento sobre UCs.

#### **6.6.3 Validade de Conclusão**

Neste estudo, o maior problema é o tamanho e a homogeneidade da amostra. No entanto, este estudo teve diferentes alunos de diferentes instituições. A quantidade de participantes ainda não é ideal do ponto de vista estatístico. Porém, este é um estudo inicial para a validação do modelo de dificuldades. Futuramente, pretende-se realizar novos estudos com mais participantes de outras universidades. Devido a estes fatos, há limitação nos resultados, sendo estes considerados indícios e não resultados conclusivos.

## 6.7 CONSIDERAÇÕES FINAIS

Este capítulo apresentou um estudo experimental que teve como objetivo validar o modelo de dificuldades em casos de uso com relação às dificuldades percebidas por estudantes de graduação dos cursos de Ciência da Computação e Sistemas de Informação de três universidades localizadas em duas diferentes regiões do país. Além disso, apresentou os resultados qualitativos realizado com a finalidade de identificar novas dificuldades percebidas durante a modelagem do diagrama e especificação de casos de uso.

Através da análise dos resultados quantitativos do estudo, foi possível responder a questão de pesquisa RQ1 (*Com qual frequência as dificuldades apresentadas no modelo são percebidas pelos alunos após a modelagem de casos de uso?*). De acordo com estes resultados observou-se que todas as sete dificuldades apresentadas no modelo para diagrama de UCs são válidas. Conforme o ranking apresentado, as principais dificuldades para modelar o diagrama de UCs (as três primeiras) são: (D2) Identificar os UCs quando as necessidades de negócio não estão claras, (D5) Confundir o sentido da seta para representar uma generalização e (D6) Identificar os relacionamentos de generalização entre os atores do sistema.

Para as dificuldades em Especificação de UC e respondendo a RQ1, as quinze dificuldades apresentadas no modelo também mostraram ser válidas, apesar de apenas doze questões apresentarem um percentual acima de 60% de ocorrência de dificuldade. As dificuldades que mais se destacaram conforme o ranking (as três primeiras dificuldades) foram: (E6) confundir requisito funcional com a descrição do UC, (E15) incluir as dependências de outros UCs na descrição do UC e (E5) descrever a interação de outros sistemas que interagem com o UC. As dificuldades E5 e E6 estão relacionadas à descrição dos fluxos do UC e a E15 está relacionada à dependência de outros UCs. As dificuldades que apresentaram um percentual de ocorrência igual ou abaixo de 50% foram: (E13 e E14) definição de pré e pós-condições na especificação do UC, (E2) entender o domínio do problema e (E4) descrever a interação entre ator e sistema. Observou-se que estas dificuldades não parecem ser um problema para os alunos durante a especificação do caso de uso.

Em relação à percepção dos participantes sobre as dificuldades apresentadas no modelo, observou-se que os participantes mencionaram as mesmas dificuldades identificadas nos estudos anteriores. É importante ressaltar que todo o estudo está relacionado com as dificuldades percebidas pelos participantes e esta percepção pode ser diferente das dificuldades observadas na prática. E respondendo a RQ2 (*Existem dificuldades que estão relacionadas ao uso de casos de uso e que não foram contempladas pelo modelo?*) percebeu-se que houve



apenas uma nova dificuldade em Diagrama de UCs mencionada por um participante do estudo. Para as dificuldades em Especificação de UCs não foi identificada nenhuma nova dificuldade. Mas, percebeu-se que os participantes mencionaram possuir dúvidas que parecem não estar relacionadas com dificuldades que ocorrem durante a especificação de UC e sim com outras dificuldades fora desse contexto, por exemplo, dúvidas sobre diferenciar requisitos funcionais de requisitos não-funcionais. Assim, é preciso avaliar o questionário com um número maior de estudantes para validação das dificuldades em casos de uso identificadas até o momento. Os resultados da AFE deste estudo mostraram que existem dificuldades que podem ser reagrupadas ou reformuladas no modelo, por exemplo, as questões relacionadas a pré e pós-condições da especificação de UC (questões Q) podem ficar em uma só questão por tratar de dois elementos da especificação que são comuns e que não parecem ser difíceis para os participantes, conforme apresentando na AFE.

As vantagens de como o modelo foi avaliado foram: a participação de outros alunos que não fizeram parte da construção do modelo através da análise qualitativa, ou seja, representam um grupo de alunos distintos de três universidades de duas diferentes regiões do país. Além disso, uma outra vantagem foi utilizar a análise fatorial exploratória (AFE) para avaliar a confiabilidade e validade das questões do modelo. Desta forma, as questões que emergiram da análise qualitativa e apresentadas no modelo foram avaliadas por outros participantes. Além disso, a AFE mostrou que existem dificuldades que são distintas e outras que talvez sejam reagrupadas em uma única questão pois apresentaram uma correlação. Outra vantagem foi capturar novas dificuldades, ou não relatadas pelos participantes. Assim, é possível deixar o modelo consolidado e tentar saturar os dados (as dificuldades).

A desvantagem encontrada foi ter um número pequeno de participantes para validar as dificuldades apresentadas no modelo. Outra possível desvantagem é número de questões que podem fadigar o participante durante as respostas no questionário.

## **CAPÍTULO 7 – CONSIDERAÇÕES FINAIS E PERSPECTIVAS FUTURAS**

*Neste capítulo são apresentadas as conclusões desta dissertação, resumindo sua motivação, propostas e apresentando as suas contribuições. As perspectivas futuras fornecem a direção para que seja dada continuidade ao trabalho relacionado ao modelo de dificuldade para modelagem de casos de uso.*

### **7.1 CONCLUSÕES**

Esta dissertação de mestrado apresentou uma pesquisa sobre as dificuldades que ocorrem durante a modelagem de casos de uso (diagrama e especificação), no processo de aprendizado de participantes que estão iniciando a modelagem de casos de uso e participantes que possuem uma experiência prévia desenvolvimento de software com UCs. Com este propósito, foi desenvolvido um modelo apresentando as dificuldades encontradas nos estudos experimentais realizados. Estas dificuldades foram classificadas em categorias, agrupadas no Diagrama de UC e na Especificação de UC, como apresentado abaixo:

- Diagrama de UC: Identificar os atores do sistema, Identificar os casos de uso do Sistema, Relacionamento de Generalização e Relacionamentos extend e include;
- Especificação de UC: Abstração dos Requisitos, Fluxos do UC, Regras de Negócio, Referências de Fluxos e RNs, Pré e Pós-Condição e Dependência de outros UCs.

Para a proposta, avaliação e evolução do modelo de dificuldades em modelar casos de uso, esta pesquisa se baseou em uma metodologia baseada em evidências (Mafra *et al.* 2006; Shull *et al.* 2001) e utilizou alguns procedimentos do método Grounded Theory na análise qualitativa dos dados (Strauss e Corbin, 1998). O modelo de dificuldades foi proposto com base no resultado dos estudos experimentais realizados com o objetivo de identificar dificuldades que ocorrem com engenheiros de software (acadêmicos e profissionais de software) durante a modelagem de casos de uso (diagrama e especificação).

No primeiro estudo realizado, percebeu-se a necessidade de aprofundar o entendimento sobre as dificuldades ocorridas durante a especificação dos casos de uso utilizando um formato textual. Com isso, realizou-se um estudo qualitativo que buscou encontrar novas dificuldades que os engenheiros de software encontram durante o uso de UC. Os resultados deste estudo auxiliaram no processo de construção da primeira versão do modelo de dificuldades em modelar

casos de uso. Além disso, realizou-se outro estudo experimental que buscou identificar novas dificuldades na modelagem do diagrama de UC, devido o foco principal do estudo qualitativo ter sido na especificação textual e não no diagrama de UC. Os resultados deste estudo auxiliaram no aprimoramento do modelo de dificuldades em modelar casos de uso, gerando uma nova versão do modelo. Além disso, comparou-se as dificuldades identificadas anteriormente com as novas dificuldades, consolidando os resultados obtidos com os resultados dos estudos anteriores, e melhorando o modelo de dificuldades.

Até o presente momento, foi realizado um estudo experimental que validou as dificuldades do modelo proposto. O estudo visou responder se as dificuldades apresentadas no modelo, que foram identificadas a partir dos estudos realizados, de fato ocorrem durante a modelagem de casos de uso, conforme a percepção de outros alunos dos cursos de ciência da computação e sistema da informação de outras universidades localizadas em duas regiões do país, como apresentado no Capítulo 6. Os resultados mostraram que o modelo de dificuldades em modelar casos de uso é válido. No entanto, algumas dificuldades apresentadas no modelo não obtiveram resultados acima de 60% de ocorrência, mostrando indícios de que existem algumas dificuldades que não parecem ser difíceis no aprendizado destes alunos. Além disso, os resultados dos questionários respondidos sobre as dificuldades na modelagem do diagrama e especificação de casos de uso mostrou que há uma grande quantidade de alunos que possuem dificuldades nas questões relacionadas à modelagem de casos de uso, mostrando que houve aplicação das dificuldades sumarizadas no modelo.

As principais dificuldades encontradas por nós durante a execução desta pesquisa foram a condução e realização dos estudos experimentais. Muitas vezes os participantes não comparecem na hora (exata) marcada para realização do estudo, isso dificulta o balanceamento de grupos e pode prejudicar na análise dos resultados. Também pode ocorrer de pessoas que não querem participar ou não realizam a tarefa como se é esperado. Assim, é necessário incluir estratégias que chamem a atenção dos participantes para participarem de forma voluntária com a pesquisa, entendendo que sua participação é extremamente importante para a pesquisa na área de Engenharia de Software.

## **7.2 CONTRIBUIÇÕES**

As principais contribuições desta dissertação são:

- Desenvolvimento de um modelo de dificuldades em modelar casos de uso (diagrama e especificação) que serve de base para auxiliar no

ensino/aprendizagem de casos de uso. Além disso, o modelo pode indicar pontos de melhoria em diretrizes e técnicas voltadas para casos de uso que auxiliam engenheiros de software na construção de UCs;

- Definição de pacotes para planejamento, execução e análise de estudos que permitiram:
  - A avaliação entre dois formatos (Gráfico e Textual) utilizados para a especificação de UCs;
  - Estudo qualitativo sobre as dificuldades em especificar UCs;
  - A avaliação do apoio da Inspeção para o aprendizado em modelagem de casos de uso (Diagrama e Especificação);
  - A avaliação e evolução do modelo de dificuldades em modelar casos de uso;
  - Disseminação do conhecimento sobre realização de estudos experimentais.
- Disseminação dos resultados obtidos nesta pesquisa. Ao longo deste trabalho de mestrado, foram publicados artigos descrevendo os resultados dos estudos experimentais realizados nesta pesquisa. A seguir é apresentada a lista de artigos aceitos descrevendo seu foco:
  - O artigo “*Is a Picture worth a Thousand Words? A Comparative Analysis of Using Textual and Graphical Approaches to Specify Use Cases*” (Nascimento *et al.*, 2014a) foi publicado no 30th Simpósio Brasileiro de Engenharia de Software (SBES 2016). Nesse artigo, foi descrito um estudo comparativo entre dois formatos (Gráfico e Textual) para especificação textual de UC com o objetivo de verificar a corretude e o tempo na geração destas especificações, conforme é apresentado no Capítulo 3;
  - O artigo “Um Modelo sobre as Dificuldades para Especificar Casos de Uso” (Nascimento *et al.*, 2017b) foi aceito no XX Congresso Ibero-Americano em Engenharia de Software (CIbSE 2017). Nesse artigo, foi descrito um estudo qualitativo em que foi proposto o modelo de dificuldades para especificação de UC, conforme apresentado no Capítulo 4.

### **7.3 PERSPECTIVAS FUTURAS**

A realização desta pesquisa resultou no desenvolvimento de um modelo de dificuldades na modelagem de casos de uso visando apresentar categorias que indicam os principais problemas enfrentados pelos estudantes durante o ensino/aprendizado de UCs. Estes resultados abrem novas perspectivas de pesquisa que podem ser exploradas em trabalhos futuros. Deste modo, espera-se que a partir destas dificuldades encontradas e sintetizadas no modelo, sirvam de base para futuras pesquisas na área, bem como no apoio à sugestão de práticas para melhorar o processo de ensino/ aprendizagem dos engenheiros de software (acadêmicos e profissionais de software) na modelagem de UC.

Com base nos resultados do primeiro estudo experimental para validação do modelo, pretende-se realizar melhorias e criar uma nova versão do modelo, por exemplo, identificar as dificuldades (questões) que estão correlacionadas (pela AFE) e talvez possam ser unificadas para capturar uma dificuldade específica, como o uso da notação de generalização para representar a relação de atores no diagrama de UCs. No entanto, observou-se que algumas dificuldades identificadas nesta pesquisa podem ser conceituais, por exemplo, a definição de regras de negócio e requisitos não-funcionais. Assim, as dificuldades apresentadas são diferentes e podem ser divididas em dificuldades conceituais e dificuldades quanto ao uso da notação do modelo de casos de uso. Desta forma, as próximas etapas para continuação desta pesquisa é identificar as dificuldades conceituais e de notação com base nos dados dos resultados dos estudos para aprimorar o modelo e apresentar diferentes visões de dificuldades.

Além disso, a divulgação de estudos experimentais apontando as dificuldades que são mais evidentes durante o aprendizado de casos de uso, pode contribuir para direcionar a utilização de práticas baseado na literatura que melhorem o entendimento e aprendizado de casos de uso.

## REFERÊNCIAS

- Achour C.B.; Rolland C.; Souveyet C.; Maiden N.A. (1999). Guiding use case authoring: results of an empirical study, in: Proceedings of the 4th IEEE International Symposium on Requirements Engineering, RE '99, IEEE Computer Society, Washington, DC, USA, pp. 36–43.
- Adolph, S.; Cockburn, A.; Bramble, P. (2002). Patterns for Effective Use Cases, Addison-Wesley Longman Publishing Co., Inc., 2002.
- Almendros-Jiménez, J.M.; Iribarne, L. (2005). Describing Use Cases with Activity Charts. Will, U.K. (ed.) In International Symposium on Metainformatics. Springer Berlin Heidelberg, pp. 141–159.
- Anda, B.; Dreiem, H.; Sjøberg, D.; Jørgensen, M. (2001). Estimating software development effort based on use cases – experiences from industry, in: M. Gogolla, C. Kobryn (Eds.), UML 2001 The Unified Modeling Language. Modeling Languages, Concepts, and Tools, Lecture Notes in Computer Science, vol. 2185, Springer, Berlin Heidelberg, pp. 487–502.
- Anda, B.; Hansen, K.; Sand, G. (2009). An investigation of use case quality in a large safety-critical software development project. In Information. Software Technology, vol. 51, n.12, pp. 1699–1711.
- Anda, B.; Jørgensen, M. (2000). Understanding use case models, in: Proceedings of Beg, Borrow and Steal Workshop, International Conference on Software Engineering (ICSE), Limerick, Ireland, 2000, pp. 94-102.
- Anda, B.; Sjøberg, D. (2002). Towards an inspection technique for use case models. In Proceedings of the 14th international conference on Software engineering and knowledge engineering, ser. SEKE '02. New York, ACM, 2002, pp. 127–134, 2002.
- Anda, B.; Sjøberg, D. (2003). Applying use cases to design versus validate class diagrams – a controlled experiment using a professional modelling tool. In Proceedings of the 2003 International Symposium on Empirical Software Engineering, ISESE '03, IEEE Computer Society, Washington, DC, USA, 2003, pp. 50–60.
- Anda, B.; Sjøberg, D. (2005). Investigating the role of use cases in the construction of class diagrams. In Empirical Software Engineering, vol. 10 n. 3, pp. 285–309.
- Anderlin, N. A.; Araújo, C.; Oliveira, H. A. B. F; Conte, T. U. (2010). Utilizando Grounded Theory para Compreender a Aceitação de uma Técnica de Elicitação de Requisitos. In VI Workshop, 2010, Belém. Anais do VI WOSSES - IX Simpósio Brasileiro de Qualidade de Software. Porto Alegre: SBC - Sociedade Brasileira de Computação, 2010.
- Arisholm, E.; Briand, L.; Hove, S.; Labiche, Y. (2006). The impact of UML documentation on software maintenance: an experimental evaluation, IEEE Transactions on Software Engineering, vol. 32, n. 6, pp. 365–381.
- Armour, F.; Miller, G. (2000). Advanced Use Case Modeling: Software Systems, Pearson Education, 2000.
- Balaban, M., Maraee, A., Sturm, A., Jelnov, P. (2015). A pattern-based approach for improving model quality. In Software & Systems Modeling, vol. 14, n.4, pp. 1527-1555.

- Bandeira-De-Mello, R.; Cunha, C. (2003). Operacionalizando o Método da Grounded Theory nas Pesquisas em Estratégia: Técnicas e Procedimentos de Análise com Apoio do Software Atlas/ti. Encontro de Estudos em Estratégia. Curitiba, Brasil.
- Bandeira-De-Mello, R.; Cunha, C. (2006). Grounded Theory. In: Godoi, C. K., Bandeira-de-Mello, R., Silva, A. B. d. (eds), Pesquisa Qualitativa em Estudos Organizacionais: Paradigmas, Estratégias e Métodos, Capítulo 8, São Paulo, Saraiva.
- Basili, V.; Rombach, H. (1988). The TAME Project: Towards Improvement Oriented Software Environments. In IEEE Transactions on Software Engineering, vol. 14, pp. 758 - 773.
- Bézivin, J; Muller, P. (1999). UML: The Birth and Rise of a Standard Modeling Notation. In International Conference on the Unified Modeling Language. Springer Berlin Heidelberg, 1999. p. 1-8.
- Bolloju, N. (2006). Exploring Quality Dependencies among UML Artifacts Developed by Novice Systems Analysts. In 12th Americas Conference on Information Systems (AMCIS 2006) pp. 472.
- Brace, W.; Ekman, K. (2014). Coramod: a checklist-oriented model-based requirements analysis approach, Requirement Engineering, vol. 19, n. 1, pp. 1–26.
- Brown, T. A. Confirmatory factor analysis for applied research. New York: The Guilford Press, 2006.
- Chaudron, M. R. V.; Heijstek, W.; Nugroho, A. (2012). How effective is UML modeling? An empirical perspective on costs and benefits. In Proceedings of Software and System Modeling, pp. 571-580.
- Cockburn, A. (2001). Writing Effective Use Cases, vol. 1, Addison-Wesley, Boston, 2001.
- Cockburn, A. (2005). Escrevendo casos de uso eficazes. Trad. Roberto Vedoato. Porto Alegre: Bookman, 2005.
- Conte, T., Massollar, J., Mendes, E., Travassos, G. H. (2007). Usability Evaluation Based on Web Design Perspectives. In Proceedings of the First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007), Madrid, Spain. September, 2007.
- Conte, T.; Cabral, R.; Travassos, G. H. (2009). Aplicando Grounded Theory na Análise Qualitativa de um Estudo de Observação em Engenharia de Software – Um Relato de Experiência. 5º Workshop Um Olhar Sociotécnico sobre a Engenharia de Software (WOSES 2009), pp. 26-37.
- Comrey, A. L., & Lee, H. B. (1992). Interpretation and application of factor analytic results. Comrey AL, Lee HB. A first course in factor analysis, 2, 1992.
- Cohen, J. Statistical Power Analysis for the Behavioral Sciences. Routledge Academic, 1998.
- Cox, K.; Aurum, A.; Jeffery, R. (2004). An experiment in inspecting the quality of use case descriptions, Journal of Research and Practice in Information Technology, vol. 36, n. 4, pp. 211–229.
- Cox, K.; Jeffery, R.; Aurum, A. (2004). A Use Case Description Inspection Experiment, University of New South Wales, School of Computer Science and Engineering, Technical report, 2004.

- Cox, K.; Phalp, K. (2001). Replicating the crews use case authoring guidelines experiment, *Empirical Software Engineering*, vol. 5, n. 3, pp. 245–267.
- Cox, K.; Phalp, K. (2003). Exploiting use case descriptions for specification and design an empirical study. In *Proceedings of the 7th International Conference on Empirical Assessment in Software Engineering*, Keele, UK, 2003, pp. 8–10.
- Cronbach, J. L. Coefficient alpha and the internal structure of tests, vol. 16. No. 3, pp. 297-334, *Psychometrika*, Setembro de 1951.
- DeVellis, R. F. *Scale development: theory and applications*. SAGE Publications, 2003.
- Detienne, F. (1988). Une application de la théorie des schémas a la compréhension de programmes. *Le Travail Humain*, tome Vol. 51, n. 4, pp. 335-350.
- Dittrich, Y.; John, M.; Singer, J.; Tessen B. (2007). For the Special issue on Qualitative Software Engineering Research. *Journal Information and Software Technology*, vol. 49, n. 6, pp. 531-539.
- Dobing, B., & Parsons, J. (2006). How UML is used. In *Communications of the ACM*, vol. 49 n.5, pp. 109-113.
- Dutoit, A.; Paech, B. (2002). Rationale-based use case specification, *Requirements Engineering*, vol. 7, n. 1, pp. 3–19.
- El-Attar, M.; Miller, J. (2009). A subject-based empirical evaluation of SSUCDs performance in reducing inconsistencies in use case models, *Empirical Software Engineering*, vol. 14, n. 5, pp. 477-512.
- El-Attar, M.; Miller, J. (2010). Improving the quality of use case models using anti patterns, *Software System Model*, vol. 9, n. 2, pp. 141–160.
- El-Attar, M.; Miller, J. (2012). Constructing high quality use case models: a systematic review of current practices. *Requirements Engineering*, vol. 17, n. 3, pp. 187–201.
- Fleisch, W. (1999). Applying use cases for the requirements validation of component based real-time software, in: *Proceedings of the 2nd IEEE International Symposium on Object-Oriented Real-Time Distributed Computing, ISORC '99*, pp. 75–84.
- Graham, I. (1998). *Requirements Engineering and Rapid Development: An Object-Oriented Approach*, ACM Press/Addison-Wesley Publishing Co, 1998.
- Gutiérrez, J. J.; Nebut, C.; Escalona, M. J.; Mejías, M.; Ramos, I. M. (2008). Visualization of Use Cases through Automatically Generated Activity Diagrams. In *11th International Conference on Model Driven Engineering Languages and Systems (MODELS)*, LNCS, vol. 5301, pp. 83-96.
- Harwood, R. (1997). Use case formats: requirements, analysis, and design, In *Journal of Object-Oriented Programming*, vol. 9, No. 8, pp. 54-57.
- Hasling, B.; Goetz, H. And Beetz, K. (2008). Model based testing of system requirements using UML use case models. In *Software Testing, Verification, and Validation, 2008 1st international conference on. IEEE, 2008*, pp. 367-376.
- Halpin, T. Business rule modality. (2006). In *International Conference on Advanced Information Systems Engineering (CAiSE'06 Workshops)*, pp. 383-94.
- Haumer, P. (2004). *Use Case-Based Software Development*, IBM Rational Rose, 2004, pp. 1–27.



- Hair, J. F.; Black, W. C.; Babin, B. J.; Anderson, R. E.; Tatham, R. L. *Análise Multivariada de Dados*. 6. ed. Porto Alegre: Bookman, 2009.
- Heumann, J. (2015). Tips for writing good use cases. In IBM Rational Software, May, 2008. Disponível em: <ftp://ftp.software.ibm.com/software/rational/web/whitepapers/RAW14023-USEN-00.pdf>. Acessado em: 02 de nov. 15.
- Huang, W; Huang, W.; Diefes-Dux, H.; Imbrie, P. K. A Preliminary Validation of Attention, Relevance, Confidence and Satisfaction Model-Based Instructional Material Motivational Survey in a Computer-Based Tutorial Setting, *British Journal of Educational Technology*, vol. 37, no. 2, 2006, pp. 243-259.
- Insrán, E.; Pastor, O.; Wieringa, R. (2002). Requirements engineering-based conceptual modelling, *Requirement Engineering*, vol. 7, n. 2, pp. 61–72.
- Ishii, N.; Suzuki, Y.; Fujiyoshi, H.; Fujii, T. (2010). Fostering UML Modeling Skills and Social Skills through Programming Education. In 23rd IEEE Conference on Software Engineering Education and Training (CSEE&T), pp. 25–32.
- Jaaksi, A. (1998). Our cases with use cases. In *Journal of Object-Oriented Programming*, vol. 10, n. 9, pp. 58-65.
- Jacobson, I. (1987). Object-oriented development in an industrial environment. In *Proceedings of the Conference on Object-Oriented Programming, Systems, Languages & Applications*, ACM, pp. 183–191.
- Jacobson, I. (2004). Use cases – yesterday, today, and tomorrow. In *Software System Modeling*. Vol. 3, n. 3, pp. 210–220.
- Jacobson, I.; Christerson, M.; Jonsson, P.; Overgaard, G. (1992). *Object-Oriented Software Engineering: A Use-Case Driven Approach*, Addison-Wesley, Reading, MA, 1992 Edition.
- Jacobson, I., Booch G., Rumbaugh J. (1998). *The Unified Software Development Process*, Addison-Wesley, New York.
- Jantunen S.; Gause D.C. (2014). Using a grounded theory approach for exploring software product management challenges. *The Journal of Systems and Software* 95 (2014) pp. 32–51.
- Keller, J. “Development and use of the ARCS model of motivational design”, *Journal of Instructional Development*, vol. 10, no. 3, 1987, pp. 2-10.
- Kettenis, J. (2007). *Getting Started With Use Case Modeling: White Paper*, Oracle Corporation, USA, 2007.
- Kruchten, P. (2003). *The Rational Unified Process: An Introduction*, Addison-Wesley, Boston, 2003.
- Kulak, D.; Guiney, E. (2012). *Use Cases: Requirements in Context*, Addison-Wesley, 2012.
- Laitenberger, O.; H. M. Dreyer. (1998). Evaluating the Usefulness and the Ease of Use of a Web-based Inspection Data Collection Tool. In *Software Metrics Symposium, 1998. Metrics 1998. Proceedings. Fifth International. IEEE, 1998*. pp. 122-132.
- Lanubile, F., Mallardo, T., Calefato, F., (2003). Tool support for Geographically Dispersed Inspection Teams”. In *Software Process Improvement and Practice*, vol. 8, 2003, pp. 217 – 231.

- Larman, C. (2012). *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*, 3/e, Pearson Education India, 2012.
- Laros, Jacob A., and L. Pasquali (2005). O uso da análise fatorial: algumas diretrizes para pesquisadores. *Análise fatorial para pesquisadores*, pp. 163-184.
- Leite, J.; Hadad, G.; Doorn, J.; Kaplan, G. (2000). A scenario construction process. *Requirements Engineering Journal*, vol. 5, n. 1, pp. 38–61.
- Linehan, M. H. SBVR use cases. (2008) In *International Workshop on Rules and Rule Markup Languages for the Semantic Web*, Springer Berlin Heidelberg, pp. 182-196.
- Mafra, S. N.; Barcelos, R. F.; Travassos, G. H. (2006). Aplicando uma metodologia baseada em evidência na definição de novas tecnologias de software. In *Anais do XX Simpósio Brasileiro de Engenharia de Software (SBES'06)*, v. 1, pp. 239-254.
- Manotas, I., Bird, C., Zhang, R., Shepherd, D., Jaspan, C., Sadowski, C., Clause, J. (2016). An empirical study of practitioners' perspectives on green software engineering. In *38th International Conference on Software Engineering*, pp. 237-248.
- Massollar, J. L.; De Mello, R. M.; Travassos, G. H. (2012). Structuring and Verifying Requirements Specifications through Activity Diagrams to Support the Semi-automated Generation of Functional Test Procedures. In *Quality of Information and Communications Technology (QUATIC) 2012*, pp. 239-244, Lisboa, Portugal.
- Mattingly, L.; Rao, H. (1998). Writing effective use cases and introducing collaboration cases. In *Journal of Object-Oriented Programming*, vol. 11, n. 6, pp. 77–87.
- Mayrhauser, V. A.; Lang, S. (1999). A Coding Scheme to Support Systematic Analysis of Software Comprehension". *IEEE Transactions on Software Engineering*, vol. 25, n. 4, pp. 526-540.
- Mello, R. M.; Massollar, J.; Travassos, G. H. (2011). Técnica de inspeção baseada em checklist para identificação de defeitos em diagramas de atividades. *Simpósio Brasileiro de Qualidade de Software (SBQS)*, pp. 168-177.
- Miller, J.; Wood, M.; Roper, M. (1998). Further experiences with scenarios and checklists, *Empirical Software Engineering*, vol. 3, n. 1, pp. 37–64.
- Misbhauddin, M., Alshayeb M., Extending the UML use case metamodel with behavioral information to facilitate model analysis and interchange. In *Software & Systems Modeling*, v. 4, n. 2, pp. 813-838 (2013).
- Mohagheghi, P.; Anda, B.; Conradi, R. (2005). Effort estimation of use cases for incremental large-scale software development. In *Proceedings of the 27th International Conference on Software Engineering (ICSE '05)*, pp. 303–311.
- Montoni, M.; Rocha, A. (2007). A Methodology for Identifying Critical Success Factors That Influence Software Process Improvement Initiatives: An Application in the Brazilian Software Industry. *14º European Conference Software Process Improvement (EuroSPI 2007)*, Potsdam, Alemanha, Springer Berlin / Heidelberg. LNCS 4764, pp. 175-186.
- Nascimento, E. S.; Silva, W., Conte, T. U., Steinmacher, I., Massollar, J., Travassos, G. H. Is a Picture worth a Thousand Words? A Comparative Analysis of Using Textual and Graphical Approaches to Specify Use Cases. In *30th Brazilian Symposium on Software Engineering (SBES '16)*, pp. 93-102 (2016).

- Nascimento, E. S.; Silva, W.; Franca, B. B. N.; Gadelha, B.; Conte, T. Um Modelo sobre as Dificuldades para Especificar Casos de Uso. In Conference Ibero-American on Software Engineering, 2017, Argentina. Proceedings of the XX Ibero-American Conference on Software Engineering (CIBSE 2017), 2017.
- OMG Unified Modelling Language Superstructure - version 2.3. <http://www.omg.org/spec/UML/2.3/>, 2010.
- Paydar, S.; Kahani, M. (2015). A semi-automated approach to adapt activity diagrams for new use cases. *Information and Software Technology*, vol. 57, pp. 543-570.
- Petersen, K.; Gencel, C. (2013). Worldviews, research methods, and their relationship to validity in empirical software engineering research. In Joint Conference of the 23rd International Workshop on Software Measurement and the Eighth International Conference on Software Process and Product Measurement (IWSM-MENSURA), pp. 81-89.
- Petri, G.; Von Wangenheim, Gresse, C.; Borgatto, A. F. MEEGA+: An Evolution of a Model for the Evaluation of Educational Games. 2016.
- Phalp, K.; Cox, K. (2001). Guiding use case driven requirements elicitation and analysis In X. Wang, R. Johnston, S. Patel (Eds.), *OOIS 2001*, Springer, London, 2001, pp. 329–332.
- Phalp, K.T.; Vincent, J.; Cox, K. (2007). Assessing the quality of use case descriptions, *Software Quality Journal*, vol. 15, n. 1, pp. 69–97.
- Phalp, K.T.; Vincent, J.; Cox, K. (2007a). Improving the quality of use case descriptions: empirical assessment of writing guidelines, *Software Quality Control*, vol. 15, n. 4, pp. 383–399.
- Porter, A.; Votta, L.G.; Basili, V. (1995). Comparing detection methods for software requirements inspections: a replicated experiment, *IEEE Transactions on Software Engineering*, vol. 21, n. 6, pp. 563–575.
- Pressley, M; McCormick, C. B. (1995) *Advanced Educational Psychology for Educators, Researchers and Policymakers*. Harper Collins, 1995.
- Rago, A., Marcos, C., Diaz-Pace, J. (2013). Uncovering quality-attribute concerns in use case specifications via early aspect mining. In *Requirements Engineering*, v.18, n.1, pp. 67–84.
- Rolland, C.; Achour, C.B. (1998). Guiding the construction of textual use case specifications, *Data Knowledge Engineering*, vol. 25, n. 12, pp. 125–160.
- Runeson, P., Host, M. (2008). Guidelines for conducting and reporting case study research in software engineering. In *Empirical Software Engineering*, 14, n. 2, p. 131-164.
- Savic, D., Antovic, I., Vlajic, S., Stanojevic, V., Milic, M. (2011). Language for use case specification. In *34th IEEE Software Engineering Workshop (SEW)*, pp. 19-26.
- Schneider, G.; Winters, J. (1998). *Applying Use Cases A Practical Guide*, Addison-Wesley, 1998.
- Seaman, C. B. (1999). Qualitative Methods in Empirical Studies of Software Engineering. In *IEEE Transactions on Software Engineering*, vol. 25, no. 4, pp. 557–572, Jul. 1999.
- Shull, F.; Carver, J.; Travassos, G. H. (2001). An empirical methodology for introducing software processes. *ACM SIGSOFT Software Engineering Notes*, vol. 26, n. 5, pp. 288-296.

- Siau, Keng, Poi-Peng Loo (2006). Identifying difficulties in learning UML. *Information Systems Management*, vol. 23, n. 3, pp. 43-51.
- Sjøberg, D. I. K.; Dyba, T.; Jørgensen, M. (2007). The Future of Empirical Methods in Software Engineering Research. *Proceedings of the Future of Software Engineering (FOSE '07)*. pp. 358-378: IEEE. 2007.
- Somé, S. (2005). Supporting use case based requirements engineering, *Information Software Technology*, vol. 48, n. 1, pp. 43–58.
- Strauss, A.; Corbin, J. (1998). *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. Thousand Oaks: SAGE Publications, 1998.
- Sutton, R. I.; Staw, B. M. (1995). What theory is not. *Administrative Science Quarterly*, vol. 40, pp. 371-384.
- Tabachnick, b. G.; fidell, l. S. *Using Multivariate Statistics*. 3. ed. New York: Harper Collins, 1996.
- Thelin, T.; Runeson, P.; Wohlin, C. (2003). An experimental comparison of usage-based and checklist-based reading, *IEEE Transactions on Software Engineering*. Vol. 29, n. 8, pp. 687–704.
- Tiwari, S.; Gupta, A. (2015). A systematic literature review of use case specifications research in *Information and Software Technology*, Vol. 67, pp. 128–158.
- Tiwari, S.; Gupta, A. (2014). Does increasing formalism in the use case template help?, in: *Proceedings of the 7th India Software Engineering Conference, ISEC 2014, ACM, NY, USA, 2014*, pp. 6:1–6:10.
- Tiwari, S.; Gupta, A. (2013). A controlled experiment to assess the effectiveness of eight use case templates. In *Proceedings of the 20th Asia–Pacific Software Engineering Conference, APSEC 2013, December 2013*, pp. 207–214.
- Tiwari, S.; Rathore, S.; Gupta, S.; Gogate, V.; Gupta, A. (2012). Analysis of use case requirements using SFTA and SFMEA techniques. In *Proceedings of the 17th International Conference on Engineering of Complex Computer Systems, ICECCS 2012, July 2012*, pp. 29–38.
- Toro, A.D.; Jiménez, B.B.; Cortés, A.R.; Bonilla, M.T. (1999). A requirements elicitation approach based in templates and patterns. In *WER, 1ST, 1999*, pp. 17–29.
- Travassos, G.; Shull, F.; Fredericks, M.; Basili, V., “Detecting defects in object-oriented designs: using reading techniques to increase software quality”, In *Proceedings of XIV ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications, 1999*, pp. 47-56.
- Travassos, G. H.; Santos, P. S. M. D.; Mian, P. G., Dias Neto, A. C.; Biolchini, J. C. D. A. (2008). An Environment to Support Large Scale Experimentation in Software Engineering, *Proceedings of the 13th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS'08)*, pp. 193-202.
- Van Den Berg, K.; Simons, A. (1999). Control-flow semantics of use cases in UML, *Inf. Software Technology*, vol. 41, n. 10, pp. 651–659.
- Wiegers, K.E. (1999). Writing quality requirements, In *Software Develop*, vol. 7, n. 5, pp. 44–48.

- Wöhlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A (2012). Experimentation in software engineering. Springer Science & Business Media.
- Yue, T., Briand, L. C., Labiche, Y. (2009). A use case modeling approach to facilitate the transition towards analysis models: Concepts and empirical evaluation. In International Conference on Model Driven Engineering Languages and Systems. Springer Berlin Heidelberg, pp. 484-498.
- Yue, T.; Briand, L.C.; Labiche, y. (2013). Facilitating the transition from use case models to analysis models: Approach and experiments. ACM Transactions on Software Engineering and Methodology (TOSEM), vol. 22, n. 1, pp. 5:1–5:38.



Autores e seus Templates	Elementos utilizados no Template							
	#Nome UC	#Obj	#AP	#Pré-Cond.	#Pós-Cond.	#FP	#FA	#PE
fluxo de eventos como (ator) para especificar os requisitos do usuário								
Somé (2005) criou um template para gerar modelos de domínio de análise	X	X	X	X	X	X	X	
Kettenis (2007) criou um template para ser usado para especificar os requisitos do usuário em um formato linear	X	X	X	X	X	X	X	X
El-Attar e Miller (2009) criaram um template que faz uso de algumas palavras-chaves específicas, como <i>include</i> e <i>extend</i> para reduzirem as inconsistências das especificações de caso de uso	X		X			X		X
Kulak e Guiney (2012) criaram um template para especificar os requisitos específicos do usuário, mas não permite o uso de pseudocódigo e construções condicionais	X	X		X	X	X	X	
Tiwari <i>et al.</i> (2012) criaram um template para validação dos requisitos do caso de uso	X	X	X	X	X	X	X	X
Yue <i>et al.</i> (2013) criaram uma proposta para geração do modelo de análise	X	X	X	X	X	X	X	X
<b>Legenda:</b> #Nome UC – Nome do caso de uso; #Obj – Objetivo ou descrição do caso de uso; #Pré-Cond. – pré- condição do caso de uso; #Pós-Cond. – pós-condição para o caso de uso; #FP – Fluxo principal do caso de uso; #FA – Fluxo alternativo e #PE – ponto de extensão para o caso de uso								

## APÊNDICE B – LISTA DE ATRIBUTOS DE QUALIDADE

*Este apêndice apresenta uma lista de atributos de qualidade e suas definições baseado em Tiwari e Gupta (2015) para avaliar a qualidade das especificações de casos de uso.*

Atributos	Autores que citaram esse atributo	Definição
<b>Compleitude</b>	Tiwari e Gupta (2013) El-Attar e Miller (2010) Anda <i>et al.</i> (2009) El-Attar e Miller (2009)  Wieggers (1999)	Deve conter na especificação do caso de uso os elementos: Atores, Pré e Pós-condições, Fluxo de Eventos (Principal, Alternativos, Exceção, Subfluxos) e relacionamento entre casos de uso e atores
<b>Exatidão</b>	Tiwari e Gupta (2013) El-Attar e Miller (2010) El-Attar e Miller (2009) Anda <i>et al.</i> (2009)	Não deve conter na especificação do caso de uso: → Descrições incorretas sobre os atores e seus relacionamentos; → Fluxo de eventos e alternativos incorretos; → Pré e Pós-condições incorretos;
<b>Consistência</b>	El-Attar e Miller (2010) El-Attar e Miller (2009) Anda <i>et al.</i> (2009)	Não deve conter na especificação do caso de uso: → Descrições de atores que sejam incompatíveis com o seu comportamento no caso de uso; A estrutura dos casos de uso e o uso da linguagem e gramática devem ser consistentes em todos os casos de uso: → Uso inconsistente de palavras-chaves; → Incoerências entre as descrições de casos de uso com o fluxo de eventos; → Inconsistências no uso da numeração dos fluxos alternativos ou de eventos; → Pré e Pós-condições que são inconsistentes com as descrições de casos de uso;
<b>Compreensível</b>	Phalp <i>et al.</i> (2007) Anda <i>et al.</i> (2001)	Não devem faltar na especificação do caso de uso: → A especificação deve ser legível; → As informações contidas nas descrições de casos de uso devem ser exatas e compreensíveis;
<b>Ambiguidade</b>	Tiwari e Gupta (2013) Anda <i>et al.</i> (2009) Wieggers (1999)	Não devem conter na especificação do caso de uso: → Atores que não refletem seu papel; → Pré ou Pós-condições ambíguas; → Fluxos de eventos com descrições ambíguas; → Condições dos fluxos alternativos ambíguas; ou → Qualquer uso de advérbios, adjetivos, pronomes, sinônimos, referências, ou negatividades; → Fluxos de Exceções ou variações com descrições ambíguas;
<b>Nível de Abstração</b>	El-Attar e Miller (2010) El-Attar e Miller (2009) Anda <i>et al.</i> (2009)	As especificações devem ser escritas e um nível consistente de abstração, sem detalhar solução de projeto ou implementação.
<b>Manutenibilidade</b>	Anda <i>et al.</i> (2009) Arisholm <i>et al.</i> (2006)	Os casos de uso devem ser fáceis de serem alterados e podem ser realizados de forma consistente ao longo de sua especificação, sem alterar a estrutura semântica do caso de uso.
<b>Verificável</b>	Anda <i>et al.</i> (2009) Cox e Phalp (2003)	Facilidade que determine se as especificações têm qualidade ou defeitos. Utilizando <i>checklist</i> , processamento da linguagem natural ou validação da estruturação de cada elemento do caso de uso.
<b>Tempo de Resposta</b>		→ Tempo necessário para identificar as informações dos casos de uso; → Tempo necessário para desenvolver as especificações do caso de uso de forma completa, correta e consistente;
<b>7C's Comunicável</b>	Phalp <i>et al.</i> (2007)	Os casos de uso devem conter: → Cobertura: tudo que é necessário para responder ao problema, informações extras são desnecessárias; → Convincente: <ul style="list-style-type: none"> <li>• Texto ordenado: O caso de uso deve seguir um caminho lógico;</li> </ul>



Atributos	Autores que citaram esse atributo	Definição
		<ul style="list-style-type: none"> <li>• Dependências: O caso de uso concluir com uma transação <i>end-to-end</i> (podendo ser fluxos alternativos ou exceção);</li> <li>• A lógica da descrição do caso de uso deve fornecer uma resposta plausível para o problema;</li> </ul> <p>→ Coerente: o caso de uso deve ter uma coerência local e global;</p> <p>→ Consistência na Abstração: o caso de uso deve ter um nível de abstração consistente;</p> <p>→ Consistência na Estrutura: o caso de uso deve ter uma estrutura consistente:</p> <ul style="list-style-type: none"> <li>• Variações: os eventos alternativos e de exceção devem ser excluídos do fluxo principal e devem estar em uma seção separada;</li> <li>• Sequência: a numeração de eventos do fluxo principal deve ser consistente;</li> </ul> <p>→ Consistência na Gramática: o caso de uso deve ter uma estrutura consistente;</p> <ul style="list-style-type: none"> <li>• Usar o verbo no tempo presente no modo simples;</li> <li>• Evitar o uso de advérbios, adjetivos, pronomes, sinônimos e negativos;</li> </ul> <p>→ Considerações sobre os fluxos alternativos: o caso de uso deve ter uma estrutura consistente;</p> <ul style="list-style-type: none"> <li>• Viáveis: os fluxos alternativos e de exceção devem ter sentido e devem ser completos;</li> <li>• Numeração: as numerações dos fluxos alternativos e de exceção devem coincidir com os números do fluxo principal;</li> </ul>

## APÊNDICE C – TABELA COM ANÁLISE DE VIABILIDADE DAS TECNOLOGIAS PARA AVALIAÇÃO DA QUALIDADE DAS ESPECIFICAÇÕES DE UC

*Este apêndice apresenta um resumo das tecnologias utilizadas para avaliar a qualidade de casos de uso baseada na revisão sistemática de Tiwari e Gupta (2015). A análise das características destas tecnologias mostra o motivo de ter sido selecionada ou não para avaliar a qualidade das especificações geradas neste estudo. A primeira coluna apresenta os autores e na segunda coluna são as tecnologias ou os experimentos realizados por estes. A terceira e quarta coluna estão relacionados com o resultado da análise das tecnologias descrevendo o motivo de inclusão ou exclusão para esta pesquisa.*

<b>Autores</b>	<b>Tecnologia/Experimento</b>	<b>S/N</b>	<b>Motivo de Inclusão ou Exclusão</b>
<b>Porter et al. (1995)</b>	Técnicas de avaliação baseado em cenários, em checklist e <i>Ad hoc</i>	Não	Os atributos de qualidade utilizados nessa pesquisa foram às primeiras investigações na área. Portanto, os autores utilizam uma notação antiga.
<b>Miller et al. (1998)</b>	Técnica de avaliação da qualidade de Requisitos de Software com Cenários	Não	As abordagens apresentadas no experimento – Cenário e Checklist – são para validação de requisitos e não da especificação de casos de uso.
<b>Wieggers (1999)</b>	Características necessárias para descrever requisitos de qualidade	Não	Embora o autor apresente características de qualidade que devem ser atendidas quando se escreve requisitos, ele não descreve como essas características devem ser avaliadas.
<b>Van den Berg e Simons (1999)</b>	Diretrizes que auxiliam a descrever os detalhes de fluxos de controle de casos de uso	Não	O foco é criar diretrizes para detalhar os fluxos de controle que estão descritos nos casos de uso.
<b>Anda e Jørgensen (2000)</b>	Avaliaram a utilização do conhecimento da psicologia cognitiva sobre como os seres humanos compreendem textos - investigaram métodos para medir a compreensão do modelo de Caso de Uso e analisar as diferenças de entendimento.	Não	Os autores avaliaram como as especificações de casos de uso geradas eram compreendidas por diferentes papéis de engenheiros de software e analisaram essas diferenças.
<b>Anda et al. (2001)</b>	Diferentes conjuntos de orientações (Orientações menores, Orientações por Modelos e Orientações por Estilos) para construir e documentar modelos de caso de uso.	Não	O objetivo dos autores foi identificar qual conjunto de diretrizes para especificação de casos de uso apresentou melhor resultado no entendimento dos participantes. O foco não foi avaliar a qualidade das especificações geradas e sim o conjunto de diretrizes que melhor auxiliava na construção de casos de uso.
<b>Anda e Sjøberg (2002)</b>	Técnica de inspeção baseada em lista de verificação utilizando uma taxonomia de defeitos específica para modelos de casos de uso.	Não	Os autores apresentam uma taxonomia de defeitos e uma lista de verificação para detectar defeitos nos modelos de casos de uso gerados pelos participantes. A taxonomia é evoluída em Anda et al. (2009).
<b>Cox e Phalp (2003)</b>	Diretrizes que orientam na especificação de casos de uso e que produzem descrições mais úteis para ajudar no projeto de software – Regras CP [Cox et al., 2001] e Orientações CREWS [Achour et al., 1999]	Não	Os autores avaliaram diretrizes que produziam descrições de casos de uso mais úteis para projetos de software. Para isso foram utilizados os atributos de qualidade clareza e consistência para verificar quais das diretrizes produziam descrições com maior qualidade.
<b>Thelin et al. (2003)</b>	Técnica de Leitura baseado em Uso (UBR - <i>Use-Based Reading</i> )	Não	As abordagens utilizadas no experimento – Técnica de Leitura baseado em Uso (UBR) e a Técnica de leitura baseada em Checklist – utilizaram uma notação antiga baseada em Porter et al. (1995)
<b>Cox et al. (2004)</b>	Checklist no processo de inspeção das descrições de casos de uso	Não	Os 7Cs de Comunicabilidade – a técnica foi evoluída e é apresentada na investigação de Phalp et al. (2007)

<b>Autores</b>	<b>Tecnologia/Experimento</b>	<b>S/N</b>	<b>Motivo de Inclusão ou Exclusão</b>
<b>Arisholm et al. (2006)</b>	Impacto da documentação da UML com a manutenção de software - Avaliaram os benefícios da disponibilidade, correção e o tempo necessário para possíveis alterações nessa documentação.	Não	Os autores utilizaram atributos de qualidade para documentação UML que atendia especificamente a manutenção de software.
<b>Phalp et al. (2007)</b>	Os 7 C's de Comunicabilidade – um conjunto de heurísticas que podem ser úteis na validação do caso de uso de como escrevê-los atendendo a esses atributos de qualidade (Cobertura, Convincente, Coerente, Consistência na Abstração, Consistência na Estrutura, Consistência na Gramática e Considerações Alternativas)	Não	Por que a técnica proposta pelos autores apresenta características amplas do que o caso de uso deve conter, mas não deixa claro como os elementos do caso de uso devem ser avaliados, por exemplo: se os atores do caso de uso foram identificados e descritos corretamente, se as relações de dependências com outros casos de uso foram identificadas e incluídas. Como não houve uma descrição clara da forma de avaliação de determinados itens, isto inviabilizou o uso desta tecnologia.
<b>El-Attar e Miller (2009)</b>	SSUCD ( <i>Simple Structured Use Case Descriptions</i> ) que consiste em um conjunto de palavras-chaves e regras sintáticas. É uma estrutura projetada para melhorar a consistência em modelos de casos de uso.	Não	O objetivo da SSUCD é criar modelos de casos de uso que melhorem a consistência das informações descritas nos casos de uso. Para isso os autores utilizaram os atributos de qualidade propostos por Anda et al. (2001) e adaptaram ao SSUCD para verificar se os modelos gerados pela técnica apresentaram melhoria na qualidade dos casos de uso.
<b>Anda et al. (2009)</b>	Atributos de qualidade utilizados para avaliar a qualidade dos modelos de UC	Sim	Porque apresentam diretrizes que auxiliam de forma mais focada e detalhada de quais informações os modelos UC's devem conter para atender aos requisitos de qualidade. A taxonomia auxilia na avaliação de defeitos do caso de uso de forma guiada. Para cada atributo de qualidade são definidos os <i>guidelines</i> que o caso de uso deve apresentar. Com isso, a execução da avaliação se torna mais direcionada e orientada para identificar os defeitos da especificação de um caso de uso.
<b>Tiwari e Gupta (2013)</b>	Avaliaram a utilidade de oito tipos de modelos de casos de uso em conjunto com cinco critérios de avaliação: completude, consistência, compreensão, redundância e propensão a falhas.	Não	O objetivo dos autores foi avaliar a utilidade de diferentes modelos para especificação de casos de uso. E para avaliar a qualidade de cada modelo os autores utilizaram cinco critérios de avaliação que também são apresentados por Anda et al. (2009) e Anda e Sjøberg (2002).
<b>Brace e Ekman (2014)</b>	CORAMOD : uma abordagem de análise de requisitos baseado em modelos orientado a lista de verificação	Não	É uma abordagem específica para inspecionar os requisitos gerados e modelados a partir da SysML - uma linguagem baseada na UML, voltada para o processo de análise de modelagem de requisitos.

De acordo com Tiwari e Gupta (2015), a qualidade das especificações de casos de uso pode ser avaliada de três maneiras: 1) Inspeção; 2) Avaliações Experimentais; e 3) Atributos de Qualidade. As Inspeções são realizadas para assegurar que as especificações de casos de uso possuem uma quantidade mínima de defeitos (Anda et al., 2009). De acordo com Anda et al. (2009), os Atributos de Qualidade contêm procedimentos que avaliam detalhadamente a corretude das descrições dos casos de uso. As Avaliações Experimentais visam analisar diferentes propostas de formatos textuais (com uso de *templates*), buscando identificar seus principais pontos positivos e negativos de cada proposta (Tiwari e Gupta, 2015). Nos próximos tópicos serão apresentadas, de forma resumida, os conceitos e as tecnologias relacionadas a **Inspeção** e aos **Atributos de Qualidade** utilizadas para avaliar a qualidade das especificações

de casos de uso e apresentadas na tabela acima. As Avaliações Experimentais, não foram utilizadas devido o foco dos estudos experimentais realizados nesta dissertação não ser a avaliação dos diferentes formatos textuais para identificação das vantagens e desvantagens.

- **Inspeção**

A inspeção é um método que pode ser utilizado para assegurar a redução de defeitos em artefatos (Travassos *et al.*, 1999). Com a inspeção é possível reduzir esforços em relação ao retrabalho na correção de artefatos durante o desenvolvimento de software (Mello *et al.*, 2009). De acordo com Mello *et al.* (2009), a inspeção pode ser realizada por meio de técnicas *ad-hoc* (o inspetor recebe o documento para realizar a revisão sem nenhuma tecnologia, direção ou foco) ou outras técnicas mais elaboradas como técnicas de leitura (caracterizada com uma série de passos para a análise individual específico de um artefato) e *checklist* (fornece apoio ao inspetor através de uma lista de perguntas, cujas respostas auxiliam na identificação de defeitos). A técnica de inspeção que tem sido mais utilizada para avaliar a qualidade das especificações de casos de uso é o *checklist* (Anda *et al.*, 2009). O uso de *checklist* tem sido uma prática comum para melhorar a qualidade das especificações de casos de uso, após a elaboração dessa documentação (Tiwari e Gupta, 2015).

Anda e Sjøberg (2002) apresentam uma taxonomia de defeitos específica para especificação de casos de uso e propõe uma técnica de inspeção baseada em lista de verificação para detectar esses defeitos. Os autores avaliaram esta técnica de inspeção em dois estudos experimentais com alunos de graduação. Os resultados das avaliações indicaram que tais técnicas foram úteis para detectar defeitos em especificação de casos de uso que utilizam o formato textual com uso de *templates*.

Porter *et al.* (1995) apresentam um método baseado em cenário, em que cada revisor utiliza diferentes técnicas. Os autores acreditam que a utilização das diferentes técnicas sistemáticas para encontrar diferentes classes e erros específicos terão uma taxa maior de sucesso. Participaram do experimento 48 estudantes de pós-graduação em ciência da computação. Eles foram separados em grupos de três. Cada equipe inspecionou duas especificações de requisitos de software (SRS – Software Requirements Specifications) usando uma combinação dos métodos *ad-hoc*, Checklist e Cenários. Os resultados mostraram que: 1) o método utilizando Cenários teve uma taxa de detecção de falhas maior do que qualquer um dos métodos *ad-hoc* ou Checklist, 2) as inspeções com Cenários foram mais eficazes na detecção de falhas dos cenários projetados e foram mais eficazes na detecção de outras falhas

quando comparado aos métodos *ad-hoc* e Checklist, 3) as Revisões com Checklist não foram mais eficazes do que a revisão com *ad-hoc*.

Cox et al. (2004) apresentaram resultados de um experimento que investigou sobre a aplicação de uma lista de verificação no processo de inspeção das descrições de casos de uso. Os autores fizeram a comparação da lista de verificação com outras abordagens da literatura. No projeto experimental simples foi utilizado um grupo de controle que utilizou a abordagem *ad-hoc* e o grupo do tratamento. Os defeitos identificados no experimento foram classificados em três níveis: 1) Impacto Mínimo no Caso de Uso; 2) Impacto na especificação; e 3) Impacto nos Requisitos.

Thelin et al. (2003) realizaram um estudo que comparou a Técnica de Leitura baseada em Uso (UBR - *Usage-based Reading*) e a Técnica baseada em Checklist. Este estudo foi baseado na notação proposta por Porter et al. (1995). Os resultados mostraram que os revisores que utilizaram a UBR foram mais eficientes e eficazes na identificação de defeitos críticos. Brace e Ekman (2014) apresentam uma abordagem utilizando uma lista de verificação de modelagem orientada a análise de requisitos (CORAMOD - *Checklist-Oriented Requirements Analysis Modelling*). A CORAMOD é uma metodologia para análise dos modelos de requisitos de sistemas complexos modelados pela SysML (*Systems Modelling Language* - Linguagem de Modelagem de Sistemas). A SysML é uma linguagem baseada na UML, conhecida e utilizada na academia e indústria, voltada para o processo de análise de modelagem de requisitos. A lista de verificação apoia uma abordagem consciente e sistemática para identificar esses requisitos. Essa abordagem ajuda na identificação desses requisitos derivados das necessidades do cliente. Além disso, melhora na validação da construção e representação dos modelos. A CORAMOD é uma abordagem específica para inspecionar os requisitos gerados e modelados a partir da SysML.

Miller et al. (1998) realizaram um experimento para avaliar se a abordagem baseada em Cenários é superior à abordagem baseada em *Checklist*. Esta pesquisa é uma continuação de estudos realizados anteriormente e visava comparar os resultados deste estudo com os anteriores. Os resultados identificados pelos autores foram às relações adicionais entre o desempenho acadêmico e o desempenho da inspeção individual e entre a ausência de reuniões e o desempenho da inspeção em grupo. As análises dos autores sugerem que o componente “realizar reuniões para inspeção de software” não é um mecanismo eficaz na detecção de defeitos. Essas abordagens são específicas para validação de requisitos e não da especificação de casos de uso.

- **Atributos de Qualidade**

Wiegers (1999) apresenta as características necessárias para descrever requisitos com qualidade (correto, possível, necessário, priorizado, explícito e testável) e características necessárias para especificar requisitos com qualidade (completo, consistente, modificável e rastreável). O autor apresenta diretrizes sobre como escrever bons requisitos para produzir melhores documentos que apoiem na construção do Software. Algumas dessas diretrizes são: 1) Mantenha frases e parágrafos curtos. Use a voz ativa; 2) Ainda precisa de esclarecimentos adicionais do autor para entender a especificação do caso de uso? É suficiente para projetar e implementar?; 3) Cuidado para não agrupar vários requisitos de forma agregada em um único parágrafo; 4) Ao documentar um requisito, faça em um nível consistente de detalhes. Além disso, é sugerido utilizar casos de teste para comparar com os requisitos especificados. Os casos de teste irão mostrar de forma clara o comportamento do produto, de como as condições especificadas estão sendo atendidas, e isso pode revelar imprecisão, omissões e ambiguidades. Embora o autor apresente um método de como descrever requisitos com qualidade, não foi descrito como essas características devem ser avaliadas.

Van den Berg e Simons (1999) analisaram os fluxos de controle de casos de uso para definir diretrizes que descrevessem detalhes desses fluxos na especificação. A partir da análise dos gráficos de fluxos dos casos de uso, os autores derivaram sete diretrizes. Essas diretrizes são guias de como descrever alguns atributos ou semânticas nas especificações de casos de uso, que serão úteis para melhorar e facilitar o raciocínio dos fluxos de controle nos diagramas de sequência, como: 1) Definir para cada caso de uso e seu diagrama de sequência o ponto de entrada e o ponto de saída. Estes pontos são pré-requisitos para um fluxo de controle bem definido. Esses possuem relacionamentos com outros casos de uso e relacionamentos estendidos; 2) Para cada caso de uso o ponto de extensão deve ser exato, ou seja, identificar em que ponto será utilizado; 3) Para cada caso de uso de extensão (relação extends) deixar explícito if-then (if-else). Descrevendo no caso de uso a condição de extensão e o ponto de extensão e - quando aplicável - descrever qual é a extensão alternativa. Os autores focaram em investigar os fluxos de controle de casos de uso, portanto as diretrizes são aplicáveis apenas para avaliar este tipo de conteúdo.

Anda e Jørgensen (2000) avaliaram a utilização do conhecimento da psicologia cognitiva sobre como os seres humanos compreendem textos. Para isso os autores utilizaram a *Theory Schema*, uma teoria desenvolvida por Barlett (1932) de como os seres humanos compreendem textos, a fim de verificar como essa teoria poderia ser aplicada na compreensão

de modelos de casos de uso. Devido ao modelo ser utilizado para diferentes partes interessadas – *stakeholders* – na documentação de um projeto de software, os esquemas são desenvolvidos de formas diferentes e conseqüentemente podem ocorrer diferentes interpretações do Modelo de Casos de Uso. Nesse sentido, os autores realizaram um experimento com dois objetivos: 1) investigar métodos para medir a compreensão do modelo de Caso de Uso; e 2) analisar as diferenças de entendimento. Para isso os autores utilizaram diferentes métodos, como *Think-Aloud Protocol*, questionários e reprodução para avaliar as interpretações dos participantes quanto ao entendimento desses modelos (Mayrhauser e Lang, 1999; Detienne, 1988; Barlett, 1932). No final avaliaram a qualidade desses modelos verificando sua consistência, para isso consideraram a falta de precisão e se houve diferença quanto a informações ausentes identificadas pelos *stakeholders*, além de verificar se existia alguma diferença na forma como os erros eram encontrados pelos *stakeholders* (Pressley e McCormick, 1995).

Anda *et al.* (2001) realizaram um estudo exploratório, onde verificaram três diferentes conjuntos de orientações (Orientações menores, Orientações por Modelos e Orientações por Estilos) para construir e documentar modelos de caso de uso. Os resultados do experimento mostraram que as orientações baseadas em modelos, são mais fáceis de entender, do que as orientações sem detalhes específicos de como documentar os casos de uso. Além disso, mostraram também que pode ser útil unir as orientações do modelo com o conjunto de diretrizes. Essas podem ensinar e orientar na especificação dos fluxos de eventos dos casos de uso. Nesse estudo, os autores apresentam alguns atributos de qualidade (Exatidão, Completude, Consistência, Compreensível e Nível de Abstração) utilizados para avaliar a qualidade das especificações geradas pelos diferentes conjuntos de orientações. No entanto, o objetivo deste estudo foi identificar qual conjunto de diretrizes apresentou melhor resultado no entendimento dos participantes.

Cox e Phalp (2003) realizaram um estudo experimental com o objetivo de comparar diretrizes que orientam na especificação de casos de uso e que produzem descrições mais úteis para ajudar no projeto de software. As regras CP (Cox *et al.*, 2001) e orientações CREWS (Achour *et al.*, 1999) foram novamente comparadas. Os resultados mostraram que as orientações com regras CP relatadas anteriormente, tiveram mais êxito na produção de especificações de casos de uso, mas não provaram ser mais úteis na fase de projeto. Os autores utilizaram os atributos de clareza e consistência para avaliar a qualidade na compreensão das especificações geradas pelos participantes. Porém o principal objetivo foi avaliar se as descrições de casos de uso foram úteis para a fase de projeto.

Arisholm *et al.* (2006) investigaram o impacto da documentação da UML com a manutenção de software em diferentes localidades (Oslo e Ottawa). Os autores avaliaram os benefícios da disponibilidade, correção e o tempo necessário para possíveis alterações na documentação. Para isso utilizaram alguns atributos específicos como corretude e compreensão para atender somente a documentação UML voltado para a manutenção de software.

Phalp *et al.* (2007) apresentam um Checklist de Qualidade para descrição de casos de uso, que atua como um controle sobre a qualidade da escrita desses casos de uso. Os autores apresentam os '7Cs' (os 7 Cs de Comunicabilidade) que é um conjunto de heurísticas que podem ser úteis na validação do caso de uso de como escrevê-los atendendo a esses atributos de qualidade (Cobertura, Convincente, Coerente, Consistência na Abstração, Consistência na Estrutura, Consistência na Gramática e Considerações Alternativas). Segundo os autores, essas heurísticas podem ajudar a ter melhores resultados na clareza e entendimento da especificação de casos de uso.

Anda *et al.* (2009) apresentaram um estudo empírico das alterações feitas nos modelos de casos de uso durante a fase de análise em um projeto de software de grande porte. Os resultados mostraram: a) quais foram os aspectos mais difíceis na modelagem dos casos de uso desse projeto; e (b) como a qualidade dos requisitos funcionais - corretude, integridade e clareza - foram melhoradas no processo de modelagem dos casos de uso. Além disso, os autores apresentam atributos de qualidade que foram utilizados para avaliar a qualidade dos modelos gerados. Alguns desses atributos também foram utilizados em estudos anteriores (Anda e Sjøberg, 2002; Anda *et al.*, 2001).

El-Attar e Miller (2009) avaliaram uma estrutura projetada para melhorar a consistência em modelos de casos de uso, com o objetivo de verificar se existia algum impacto em sua utilização. Essa estrutura é chamada de SSUCD (*Simple Structured Use Case Descriptions*) que consiste em um conjunto de palavras-chaves e regras sintáticas. Esse conjunto é utilizado para apoiar na especificação de casos de uso. Nesse estudo, os autores ainda utilizaram os atributos de qualidade propostos por Anda *et al.* (2001) para avaliar se os casos de uso gerados pela SSUCD melhoravam sua consistência.

Tiwari e Gupta (2013) avaliaram a utilidade de oito tipos de *templates* de casos de uso em conjunto com cinco critérios de avaliação: completude, consistência, compreensão, redundância e propensão a falhas. Os modelos foram utilizados por estudantes de graduação com diferentes especificações de casos de uso. Os resultados avaliaram que o uso do modelo de Yue *et al.* (2013) mostrou ser mais consistente e menos propenso a falhas. O modelo de



Cockburn (2001) mostrou ser mais completo e de melhor compreensão. O modelo de Tiwari *et al.* (2012) foi o menos redundante em relação ao modelo de análise proposto por Yue *et al.* (2013) e ao modelo detalhado em formato de tabela proposto por Cockburn (2001), embora os resultados não foram estatisticamente significativos. Os critérios de avaliação utilizados para avaliar a qualidade dos casos de uso desta pesquisa, também são apresentados em Anda e Sjøberg (2002) e Anda *et al.* (2009) (No APÊNDICE B - apresenta um resumo sobre os atributos de qualidade utilizados para avaliar a especificação de casos de uso).

## APÊNDICE D – QUESTIONÁRIO PÓS-MODELAGEM I

*Este apêndice apresenta o questionário Pós-Modelagem I utilizado no primeiro estudo descrito no capítulo 3.*

QUESTIONÁRIO PÓS-MODELAGEM						
Nome:						
Por gentileza, responda as questões a seguir considerando sua experiência durante a modelagem utilizando o <b>UCModel</b> para especificar o caso de uso						
	Concordo Totalmente	Concordo Amplamente	Concordo Parcialmente	Discordo Parcialmente	Discordo Amplamente	Discordo Totalmente
1. Foi fácil aprender a especificar um caso de uso utilizando o UCModel.						
<b>Explique sua resposta</b>						
	Concordo Totalmente	Concordo Amplamente	Concordo Parcialmente	Discordo Parcialmente	Discordo Amplamente	Discordo Totalmente
2. Foi fácil construir a especificação do caso de uso utilizando o UCModel.						
<b>Explique sua resposta</b>						
	Concordo Totalmente	Concordo Amplamente	Concordo Parcialmente	Discordo Parcialmente	Discordo Amplamente	Discordo Totalmente
3. Utilizar o UCModel aumentou meu entendimento sobre a interação do ator com o sistema (acredito ter maior compreensão para modelar o comportamento do caso de uso do que antes de utilizar o modelo).						
<b>Explique sua resposta</b>						
	Concordo Totalmente	Concordo Amplamente	Concordo Parcialmente	Discordo Parcialmente	Discordo Amplamente	Discordo Totalmente
4. O UCModel foi útil para especificar um caso de uso						
<b>Explique sua resposta</b>						
Quais foram as principais dificuldades que você encontrou ao especificar o caso de uso utilizando o UCModel?						
Onde você empregaria o modelo UCModel para especificar casos de uso.						

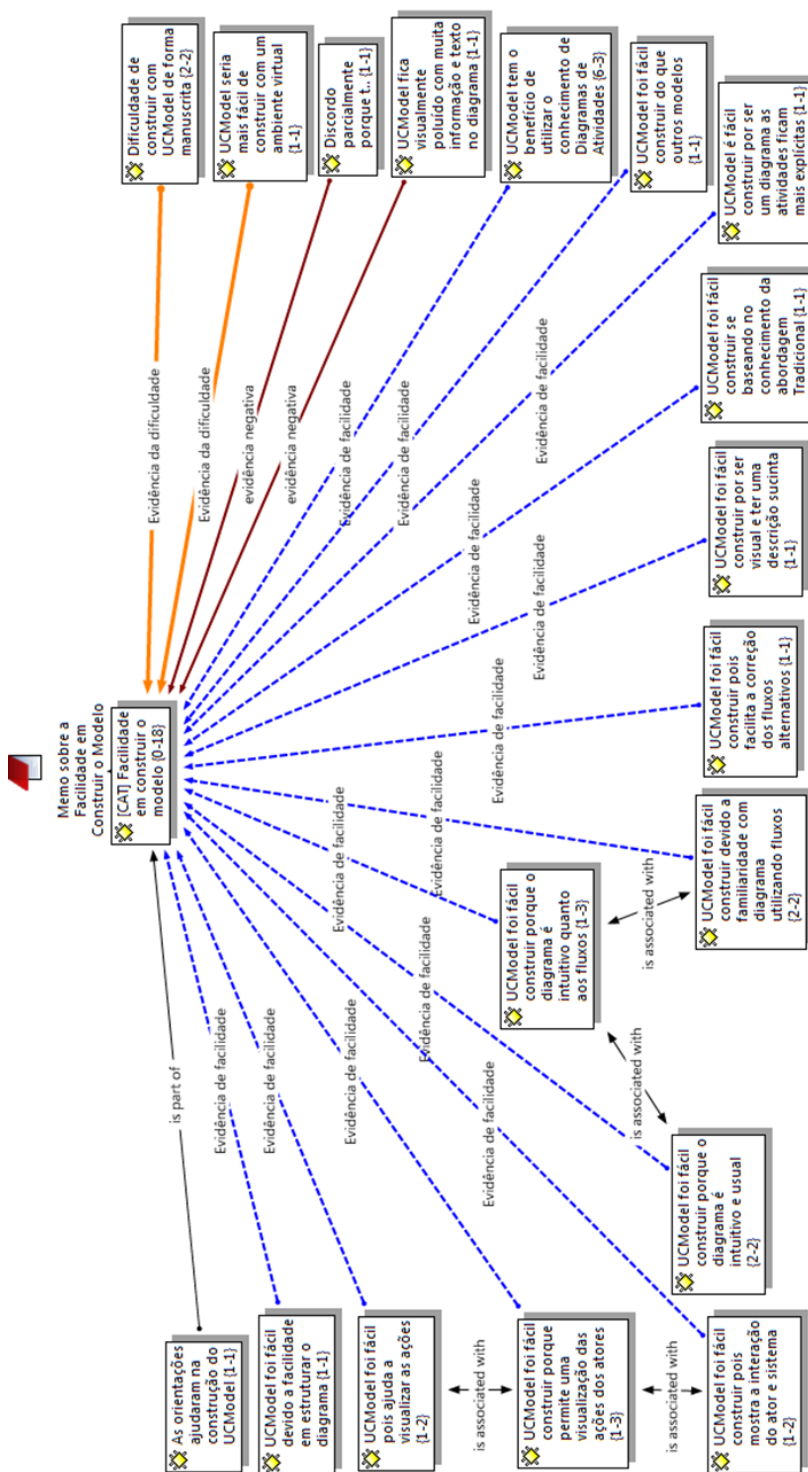
## APÊNDICE E – QUESTIONÁRIO PÓS-MODELAGEM II

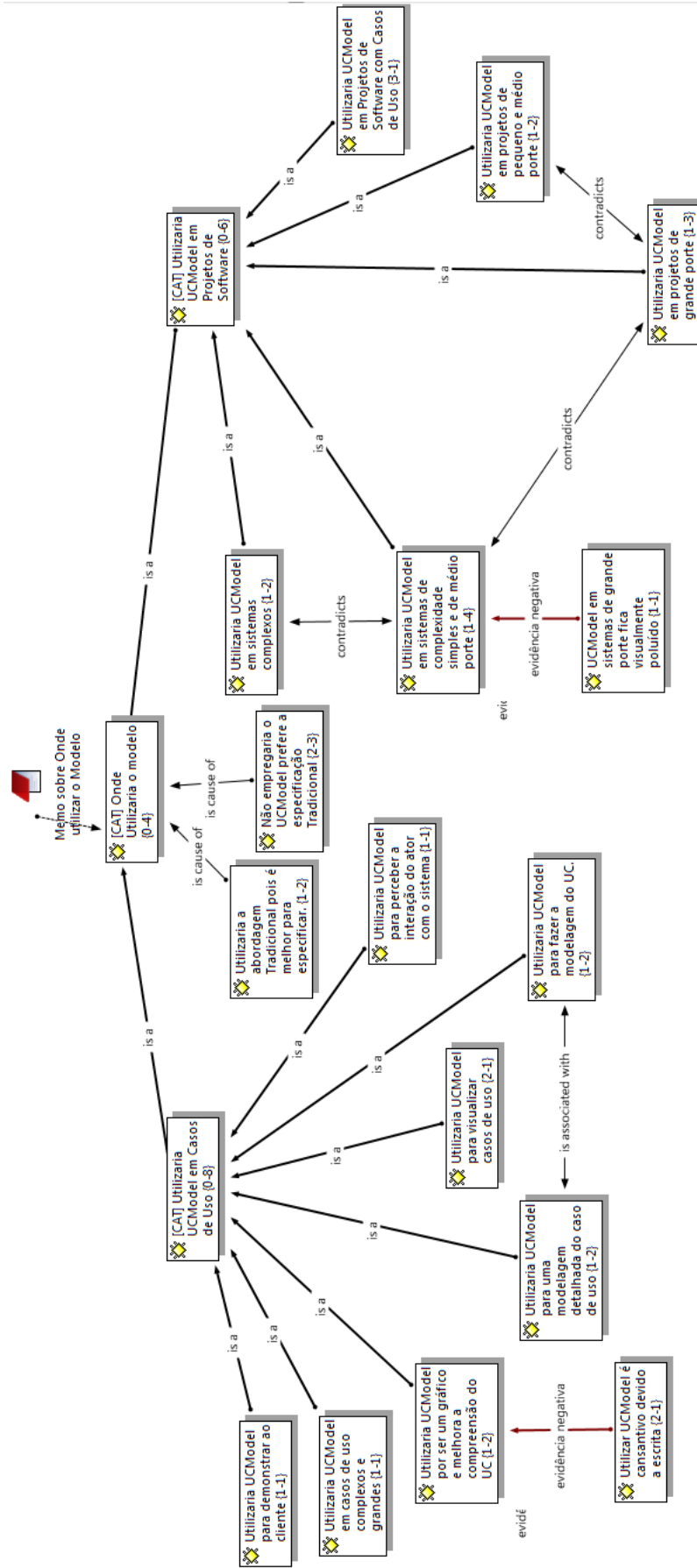
*Este apêndice apresenta o questionário Pós-Modelagem II utilizado no primeiro estudo descrito no capítulo 3.*

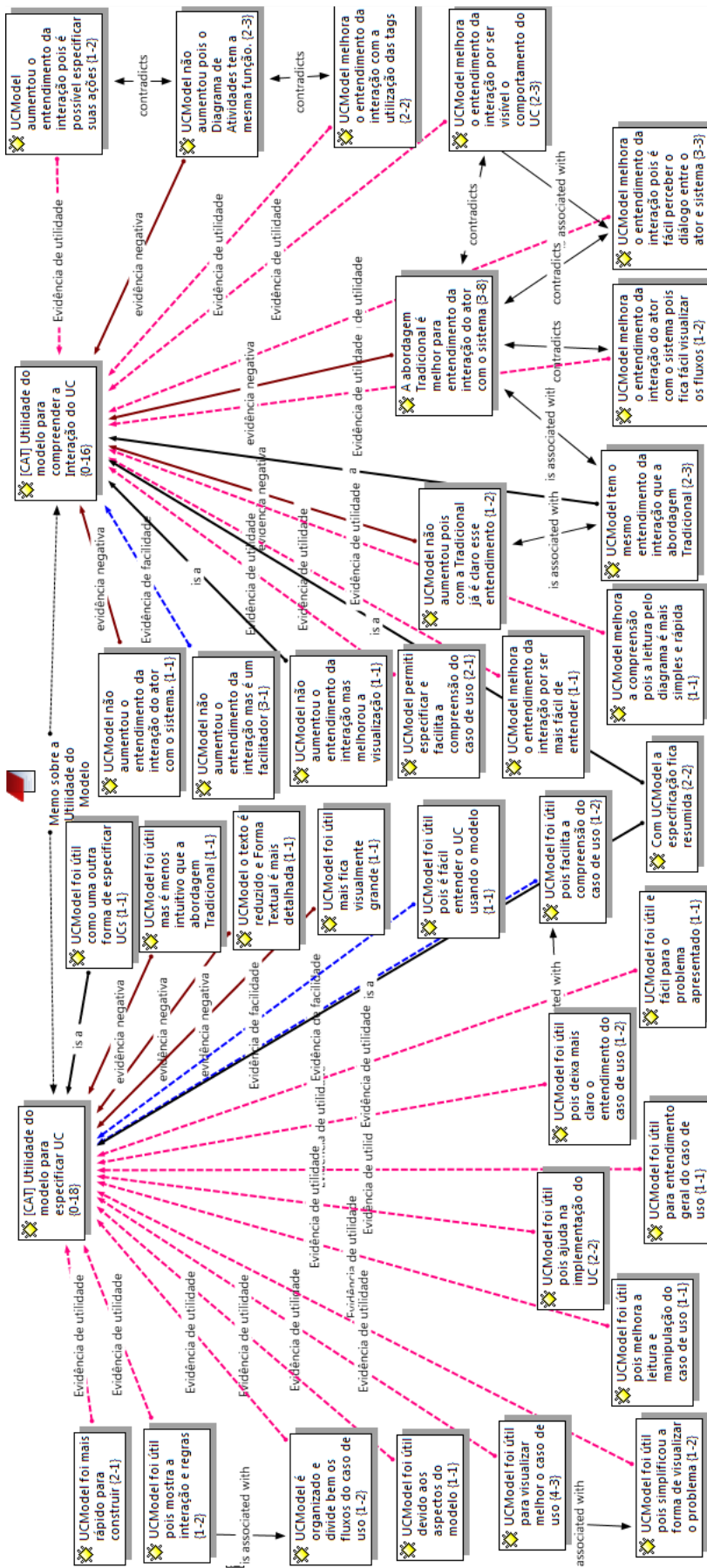
<b>QUESTIONÁRIO PÓS-MODELAGEM</b>	
Por gentileza, responda as questões a seguir do ponto de vista comparativo das abordagens utilizadas.	
<b>1. Qual a abordagem mais fácil para especificar um caso de uso?</b>	
<input type="checkbox"/> Especificar casos de uso utilizando o Diagrama de Atividades do UCMModel.	
<input type="checkbox"/> Especificar casos de uso utilizando o modelo tradicional “textual”.	
<input type="checkbox"/> Especificar casos de uso utilizando outra abordagem. Qual? _____	
<b>Comente a razão de sua escolha.</b>	
<b>2. Qual a abordagem mais útil para especificar casos de uso?</b>	
<input type="checkbox"/> Especificar casos de uso utilizando o Diagrama de Atividades do UCMModel.	
<input type="checkbox"/> Especificar casos de uso utilizando o modelo tradicional “textual”.	
<input type="checkbox"/> Especificar casos de uso utilizando outra abordagem. Qual? _____	
<b>Por quê? Comente a razão de sua escolha.</b>	
<b>3. Para especificar casos de uso de uma aplicação, qual a sua opção?</b>	
<input type="checkbox"/> Eu especificaria os casos de uso com Diagrama de Atividades do UCMModel.	
<input type="checkbox"/> Eu especificaria os casos de uso utilizando um modelo tradicional “textual”.	
<input type="checkbox"/> Eu especificaria os casos de uso utilizando uma abordagem mista, com Diagrama de Atividades do UCMModel e o modelo tradicional “textual”. Em que ocasiões? _____	
<input type="checkbox"/> Eu especificaria casos de uso com base em outro modelo ou artefato. Qual? _____	
<b>Por quê? Comente a razão de sua escolha.</b>	

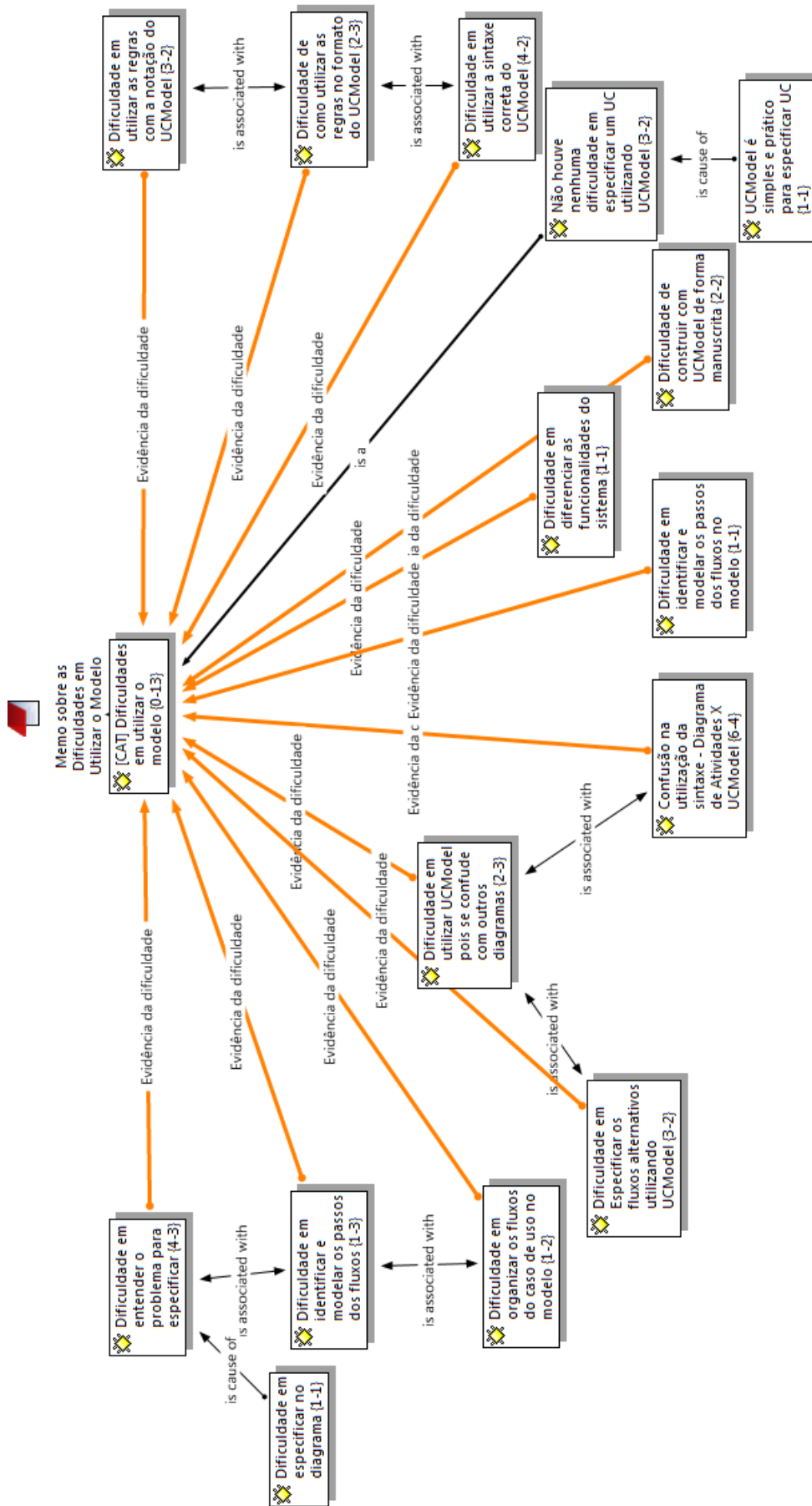
## APÊNDICE F – NETWORKS DA ANÁLISE QUALITATIVA DO PRIMEIRO ESTUDO

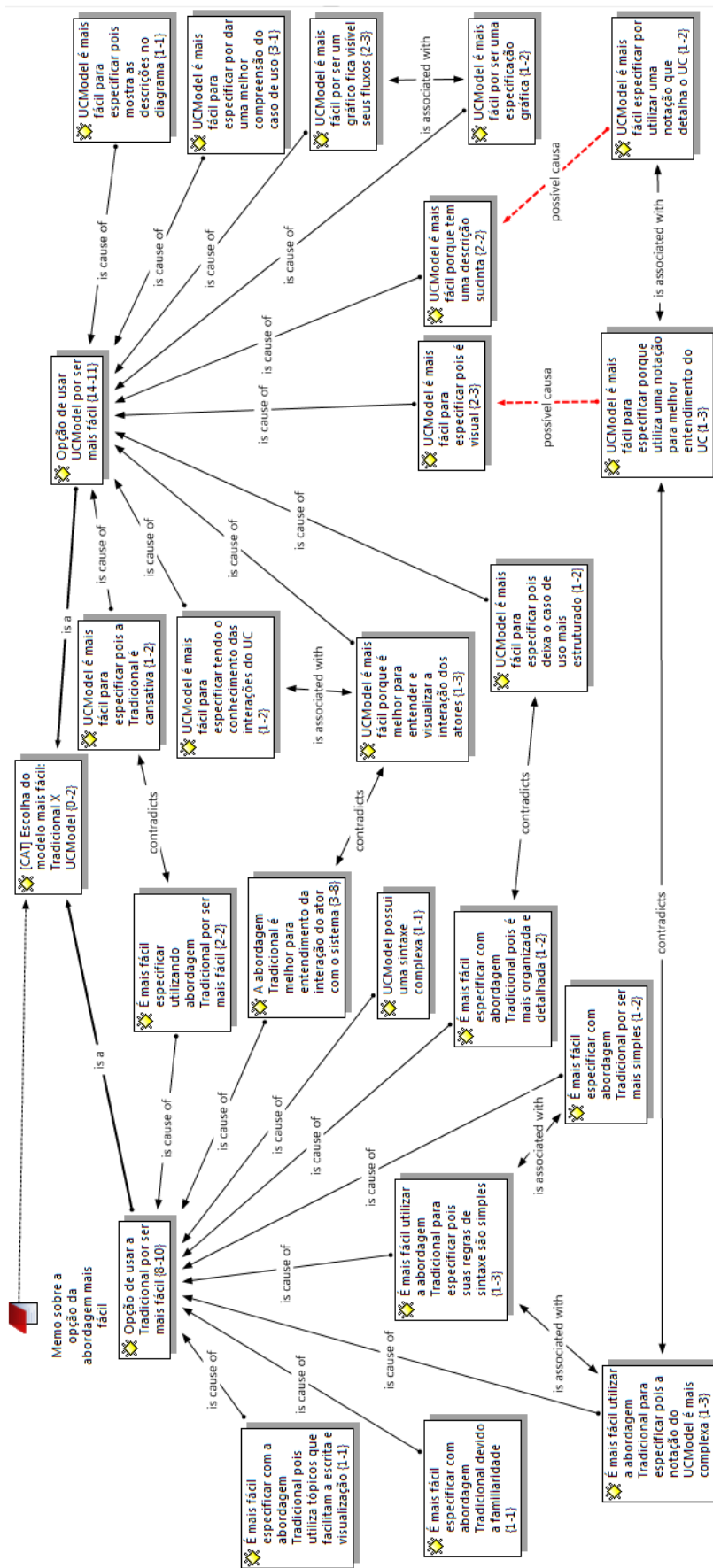
Este apêndice apresenta as networks geradas a partir da análise qualitativa realizada no primeiro estudo descrito no Capítulo 3.



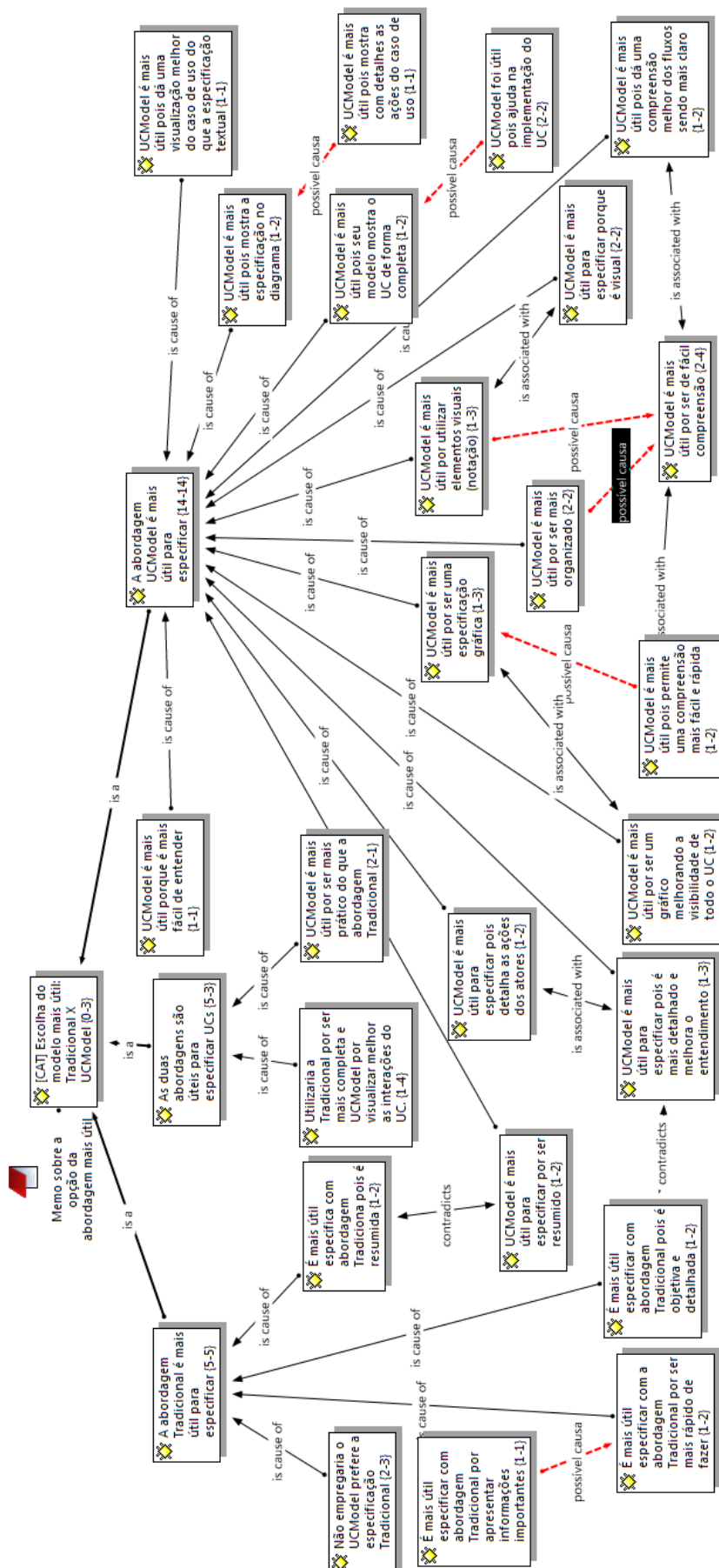


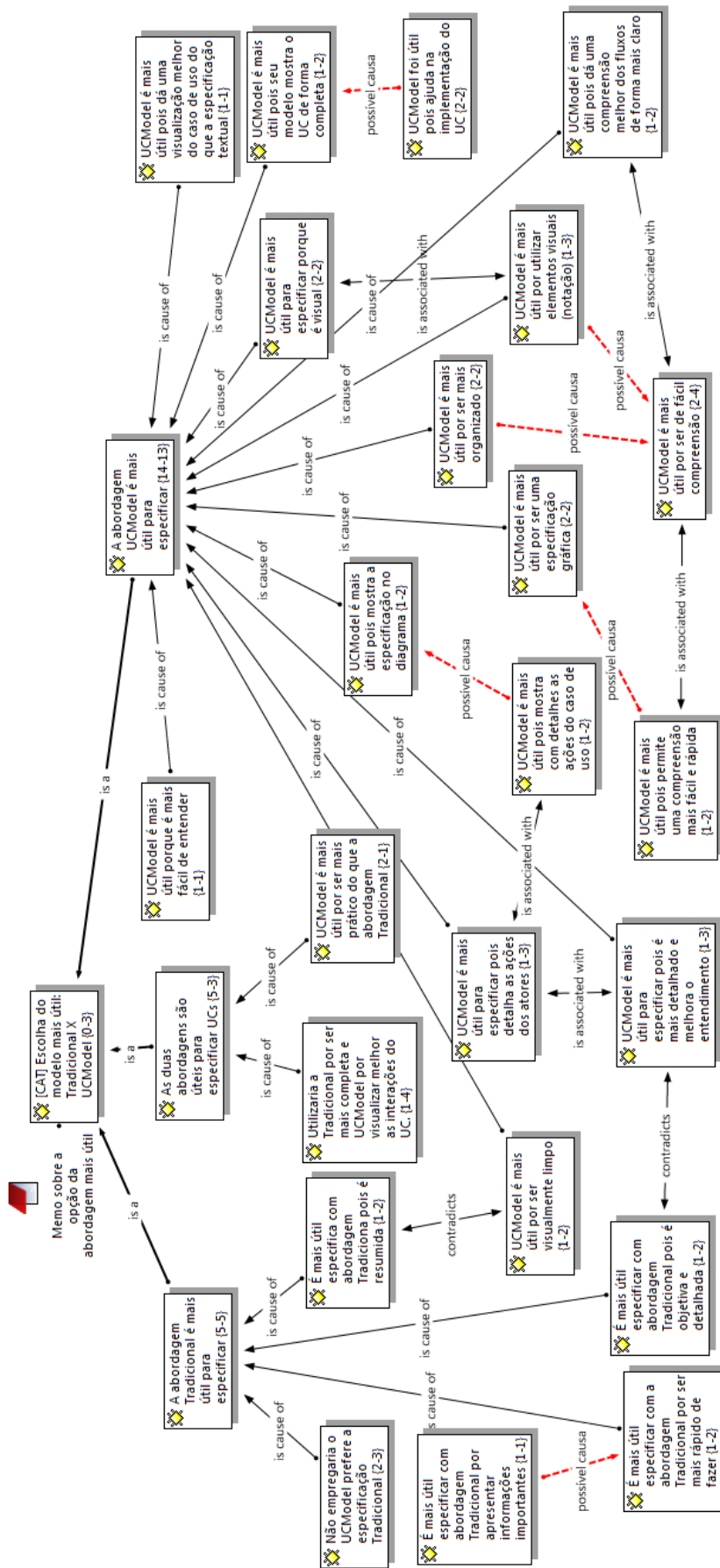








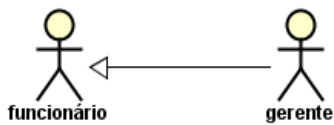
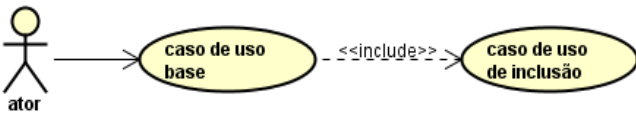
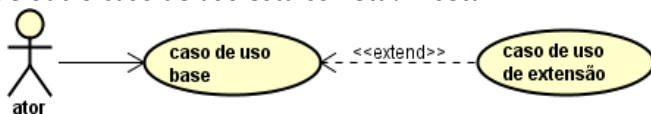




## APÊNDICE G – CHECKLISTS ELABORADOS PARA INSPEÇÃO NOS CASOS DE USO

Este apêndice apresenta os três checklists elaborados para inspecionar os casos de uso que foram utilizados pelos participantes do estudo descrito no Capítulo 5. Os checklists utilizaram como base as classes de defeitos de Travassos *et al.* (1999).

### Checklist para Inspeção nos Diagramas de Casos de Uso

CATEGORIA	CÓD.	ITENS DE VERIFICAÇÃO
OMISSÃO	OMI_01	Os atores (principais usuários e sistemas) que executam as principais tarefas do software foram identificados?
	OMI_02	Faltou identificar outros sistemas que irão se comunicar com o sistema a ser construído? Estão identificados como atores?
	OMI_03	Todas as funcionalidades do sistema foram representadas no diagrama?
	OMI_04	Faltou identificar alguma funcionalidade que será executada (automaticamente/ programada) pelo software?
	OMI_05	Faltou incluir alguma associação de um ator com o caso de uso no diagrama?
	OMI_06	Faltou incluir o nome da associação de um caso de uso com outro caso de uso ( <i>include</i> ou <i>extend</i> )?
	OMI_07	Faltou incluir um ator que herda o perfil de outro ator mas realizando algo a mais no sistema? (Generalização)
FATO INCORRETO	FIN_01	O nome do caso de uso expressa corretamente o objetivo que será realizado pelo ator no sistema?
	FIN_02	O nome dos atores e casos de uso são únicos?
	FIN_03	Existem casos de uso que foram quebrados em partes pequenas mas que não estão corretos isoladamente? Ou seja, que deveriam estar agrupados em um caso de uso para manter a funcionalidade completa?
	FIN_04	A direção da seta utilizada para representar o relacionamento entre atores ( <b>Generalização</b> ) está correta?
		
	FIN_05	A direção da seta utilizada para representar a associação de inclusão ( <b>include</b> ) de um caso de uso está correta?
		
FIN_06	A direção da seta utilizada para representar a associação da extensão ( <b>extend</b> ) de outro caso de uso está correta? A seta	
		
INCONSISTÊNCIA	INC_01	O nome do caso de uso é coerente com o objetivo das funcionalidades que serão executadas pelos atores?
	INC_02	A associação de um ator com um caso de uso está consistente com o contexto da tarefa do ator?

AMBIGUIDADE	AMB_01	O nome dos atores e/ou casos de uso estão claros ou podem levar a dupla interpretação?
	AMB_02	Existe algum ator que possui o mesmo papel no sistema com nome diferente?
	AMB_03	Existe algum caso de uso que executa a mesma funcionalidade?
	AMB_04	As associações dos atores com os casos de uso estão claras?
INFORMAÇÃO ESTRANHA	IES_01	Os atores fazem parte do contexto do sistema?
	IES_02	Os casos de uso fazem parte do contexto do sistema?

### Checklist para Inspeção nas Especificações de Casos de Uso

CATEGORIA	CÓD.	ITENS DE VERIFICAÇÃO
OMISSÃO	OMI_01	Faltou identificar/descrever o nome do(s) ator(es)?
	OMI_02	Faltou descrever na pré-condição alguma condição que é <b>indispensável</b> para iniciar o caso de uso?
	OMI_03	Faltou descrever algum fluxo de evento (alternativo e/ou exceção) necessário para a completude do caso de uso?
	OMI_04	Faltou descrever alguma regra de negócio necessária para a completude do caso de uso?
	OMI_05	Faltou referenciar alguma regra de negócio, fluxos alternativos ou fluxos de exceção em algum passo dos fluxos de eventos?
	OMI_06	Faltou incluir a dependência de outros casos de uso ( <i>include</i> ou <i>extend</i> ) que estão associados ao caso de uso? (conforme o diagrama de casos de uso)
FATO INCORRETO	FIN_01	O nome dado ao caso de uso expressa corretamente seu objetivo?
	FIN_02	As identificações dos fluxos alternativos (A1, A2, etc), fluxos de exceção (E1, E2 etc) e regras de negócio (RN1, RN2 etc) são únicos? Estão referenciados no caso de uso corretamente?
	FIN_03	A descrição dos fluxos principal, alternativos e exceções estão completos e corretos?
	FIN_04	A descrição das regras de negócio estão completas e corretas?
INCONSISTÊNCIA	INC_01	A pré-condição está coerente com o comportamento do caso de uso? É realmente uma condição que impede o caso de uso iniciar?
	INC_02	A descrição do caso de uso está consistente com o comportamento do caso de uso?
	INC_03	O sequenciamento nos fluxos principal, alternativos e de exceções estão coerentes?
AMBIGUIDADE	AMB_01	Os nomes de atores refletem seus papéis? Ou estão ambíguos (podem levar a dupla interpretação)?
	AMB_02	O nome do caso de uso permite interpretações diferentes do seu objetivo?
	AMB_03	Há mais de um caso de uso que possui o mesmo nome?
	AMB_04	A descrição de pré e pós-condições ( <i>quando existem</i> ), estão claros e sem ambiguidades?
	AMB_05	A descrição dos fluxos de eventos e das regras de negócios estão claros e sem ambiguidades?
	AMB_06	Os fluxos de eventos (principal, alternativo e exceção) seguem um caminho lógico e claro?
	AMB_07	As informações trocadas entre o ator e o sistema estão claras e bem definidas?
INFORMAÇÃO ESTRANHA	IES_01	Informações descritas nos passos dos fluxos de eventos (principal, alternativos e/ou exceções), fazem parte do contexto do caso de uso?

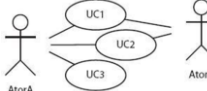
	IES_02	Informações descritas nas regras de negócio fazem parte do contexto do caso de uso?
	IES_03	Atores ou casos de uso descritos fazem parte do contexto do caso de uso?

### Checklist Vertical: Requisitos X Diagrama de UCs X Especificação de Casos de Uso

CATEGORIA	CÓD.	ITENS DE VERIFICAÇÃO
OMISSÃO	OMI_01	Faltou identificar algum <b>ator</b> no <b>diagrama</b> de casos de uso que está descrito na lista de requisitos?
	OMI_02	Faltou identificar algum <b>caso de uso</b> no <b>diagrama</b> para atender uma funcionalidade descrita na lista de requisitos?
	OMI_03	Faltou identificar o <b>nome</b> do(s) <b>caso(s) de uso</b> na <b>especificação</b> conforme o diagrama?
	OMI_04	Faltou identificar/descrever o <b>nome</b> do(s) <b>ator(es)</b> na <b>especificação</b> do caso de uso conforme o diagrama e lista de requisitos?
	OMI_05	Faltou identificar os relacionamentos de <b>extensão</b> ou <b>inclusão</b> na <b>especificação</b> do caso de uso conforme o diagrama?
FATO INCORRETO	FIN_01	Existe algum ator/caso de uso incluído no diagrama de casos de uso que não está correto de acordo com a lista de requisitos?
	FIN_02	Existe alguma regra de negócio descrita na especificação do caso de uso que não está correta de acordo com a lista de requisitos?
	FIN_03	Existe alguma informação descrita nos fluxos de eventos na especificação do caso de uso que não está correta de acordo com a lista de requisitos?
INCONSISTÊNCIA	INC_01	O diagrama de casos de uso está consistente com a lista de requisitos?
	INC_02	A especificação do caso de uso está consistente com o diagrama?
	INC_03	A associação para estender o caso de uso (extend) está consistente? Ou seja, quando o caso de uso base for executado o caso de uso de estendido poderá ser executado também?
	INC_04	A associação para inclusão do caso de uso (include) está consistente? Ou seja, quando o caso de uso base for executado o caso de uso de inclusão será executado?
AMBIGUIDADE	AMB_01	O diagrama de casos de uso mostra de forma clara os atores e funcionalidades que foram descritos na lista de requisitos?
	AMB_02	A especificação do caso de uso atende às funcionalidades descritas na lista de requisitos de forma clara e sem ambiguidade?
INFORMAÇÃO ESTRANHA	IES_01	Existe algum ator e/ou caso de uso no diagrama que não está descrito na lista de requisitos? Ou seja, não fazem parte do contexto e foram incluídos no diagrama?
	IES_02	Atores ou casos de uso que não estão no diagrama e não fazem parte do contexto foram descritos na especificação do caso de uso?
	IES_03	Existe alguma regra de negócio que não faz parte do contexto do sistema, conforme a lista de requisitos, mas foi descrita na especificação de casos de uso?

## APÊNDICE H – QUESTIONÁRIO SOBRE AS DIFICULDADES EM CASOS DE USO (DIAGRAMA E ESPECIFICAÇÃO)

Este apêndice apresenta o questionário de dificuldades de casos de uso que os participantes do estudo descrito no Capítulo 5 responderam.

Informe seu nome: _____							
<input type="checkbox"/> Concordo com a utilização dos dados deste formulário na pesquisa. Toda informação coletada é confidencial <sup>5</sup>							
<b>Em relação ao grau da sua experiência em desenvolvimento de software com casos de uso, marque os itens abaixo que melhor refletem sua experiência.</b>							
Você já participou de projetos de desenvolvimento de software que utilizaram casos de uso (na indústria ou na academia)? ( ) SIM ( ) NÃO							
De quantos projetos você participou? ( ) Somente projetos em sala de aula ( ) De 1 a 4 projetos na indústria ( ) Mais de 5 projetos na indústria							
Identificação	<b>Ao aprender a modelar o diagrama de casos de uso...</b>		Sempre	Quase Sempre	Algumas vezes	Raramente	Nunca
D1	Foi difícil identificar outros sistemas que interagem com o sistema que será desenvolvido?						
D2	Foi difícil identificar os casos de uso quando as necessidades de negócio não estão claras?						
D3	Foi difícil dividir os casos de uso para atender as funcionalidades do sistema?						
D4	Foi difícil identificar quais funcionalidades se relacionam com cada ator do sistema?						
D5	Você já confundiu o sentido da seta para representar uma generalização?						
D6	Você teve dúvidas em identificar os relacionamentos de generalização entre os atores do sistema?						
D7	Você teve dúvidas ao utilizar os relacionamentos de <i>extend</i> e <i>include</i> (extensão e inclusão)?						
	Você teve outra dificuldade, que não foi mencionada acima, durante a modelagem do diagrama de Casos de Uso? Se sim, qual?						
Identificação	<b>Ao aprender a descrever/ especificar o caso de uso...</b>	<div style="border: 1px solid black; padding: 2px;">                     Fluxo Principal:                      1. O cliente chega ao tabuleiro com as regras que deverão jogar.                      2. O cliente informa seu nome e o nome do time ao funcionário.                      3. O funcionário registra o nome do cliente e inicia o jogo.                      4. O funcionário registra cada vez que o time.                      5. O funcionário finaliza a jogada. Quando as regras do cliente e do sistema são registradas, o valor do jogo é calculado.                      6. O cliente se emociona com as regras.                 </div>	Sempre	Quase Sempre	Algumas vezes	Raramente	Nunca
E1	Foi difícil entender a funcionalidade do sistema (para o caso de uso que você escreveu)?						
E2	Foi difícil entender o domínio/contexto da aplicação que seria desenvolvida?						
E3	Foi difícil identificar e descrever os fluxos do caso de uso?						
E4	Foi difícil descrever a interação entre o ator e sistema nos fluxos do caso de uso?						
E5	Foi difícil descrever a interação de outros sistemas que interagem com o sistema que será desenvolvido?						
E6	Você já confundiu a descrição/especificação dos casos de uso com o detalhamento dos requisitos funcionais do sistema?						
E7	Você teve dúvidas se o fluxo que você descreveu para o caso de uso era um fluxo alternativo?						
E8	Foi difícil identificar as regras de negócio e descrevê-las nos casos de uso?						
E9	Você já confundiu as regras de negócio com os fluxos do caso de uso?						
E10	Você já confundiu as regras de negócio com os requisitos não-funcionais do sistema?						
E11	Foi difícil referenciar os fluxos alternativos nos passos do caso de uso?						
E12	Foi difícil referenciar as regras de negócio nos passos do caso de uso?						
E13	Foi difícil definir a(s) pré-condição(ões) do caso de uso?						
E14	Foi difícil definir a(s) pós-condição(ões) do caso de uso?						
E15	Foi difícil incluir as dependências (include e/ou extend) de outros casos de uso na descrição do caso de uso que você escreveu?						
	Você teve outra dificuldade, que não foi mencionada acima, durante a descrição/especificação do Caso de Uso? Se sim, qual?						

<sup>5</sup> Em caso de dúvida, por favor consulte o professor.

## APÊNDICE I – ROTEIRO PARA ENTREVISTA SOBRE AS DIFICULDADES PARA DESCREVER CASOS DE USO

Questão de pesquisa: Quais as dificuldades percebidas por estudantes ao descrever Casos de Uso em um desenvolvimento de software contínuo?

I. Observações importantes (Contextualizar a pesquisa ao entrevistado):

- A pesquisa busca verificar quais as dificuldades que os estudantes encontram ao descrever casos de uso utilizando um formato textual;
- Entregar o termo de consentimento e explicar a sua importância;
- <Pedir autorização para gravar a entrevista e informar que de forma alguma os dados serão utilizados para outros fins fora do âmbito da pesquisa>.

II. Questões

### Background do Aluno

- 1) Você já participou de projetos de desenvolvimento de software (na indústria ou na academia)?  
<Se a resposta for SIM>
  - a. Os projetos foram realizados na indústria ou na academia?
  - b. De quantos projetos você participou?
    - i. Você participou de um ou mais projetos?
  - c. Qual a duração de cada projeto?
    - i. Os projetos duraram seis meses, um ano ou mais?
- 2) Você já participou de projetos de desenvolvimento de software que utilizaram casos de uso (na indústria ou academia)?  
<Se a resposta for SIM>
  - a. Os projetos foram realizados na indústria ou na academia?
  - b. De quantos projetos você participou?
    - i. Você participou de um ou mais projetos?
  - c. Qual a duração de cada projeto?
    - i. Os projetos duraram seis meses, um ano ou mais?
  - d. Como você descrevia os casos de uso neste projeto? Utilizava algum tipo de template usando texto e/ou diagrama (diagrama de atividades ou de sequência, por exemplo)?  
<Se a resposta for que o participante descreveu com template textual e/ou diagrama, deve-se fazer as seguintes perguntas>
    - I. Em sua opinião, descrever o caso de uso em um template textual é mais adequado em que situação? Explique um pouco.
    - II. Em sua opinião, descrever o caso de uso em um template que utiliza algum diagrama é mais adequado em que situação? Explique um pouco.
    - III. Você prefere utilizar um template textual/diagrama para descrever um caso de uso? Explique um pouco.

### Descrição de Casos de Uso

- 3) Qual a sua percepção sobre a utilização da descrição de casos de uso durante o desenvolvimento de software?  
Esta pergunta visa a obter do entrevistado as respostas para as seguintes questões:
  - a. A descrição de casos de uso é uma maneira de comunicar aos stakeholders (Cliente e Equipe de Desenvolvimento) o que o sistema deve fazer?
  - b. A descrição de casos de uso ajuda a descrever o comportamento específico de um caso de uso?
  - c. O template que você utilizou precisou de alguma adaptação?  
<Se a resposta for SIM>
    - i. O que foi adaptado neste template? Explique o porquê.
  - d. Em sua opinião, quais informações são mais importantes ao descrever um caso de uso e não podem deixar de ter no template utilizado? Explique um pouco sua resposta.

### **Dificuldade em Descrever Casos de Uso**

4) Você tem/teve alguma dificuldade para descrever um caso de uso?

< Se a resposta for SIM fazer as perguntas I e II >

I. Quais dificuldade você sentiu ao descrever o caso de uso?

II. Como você conseguiu contornar essas dificuldades?

Obs.: Caso o entrevistado não consiga apontar alguma dificuldade, utilizar os seguintes exemplos:

- i. Você teve dificuldade para entender (abstrair) o requisito para descrever o caso de uso?
- ii. Você teve dificuldade para identificar os fluxos (principal, alternativos e exceções) e as regras de negócio ao descrever o caso de uso?
- iii. Você teve dificuldade para organizar os fluxos (principal, alternativos e exceções) ao descrever o caso de uso?
- iv. Você teve dificuldade para utilizar o template ao descrever o caso de uso?

5) Durante a execução do projeto, você teve alguma dificuldade em criar outros modelos (diagrama de classes, diagrama de sequência, diagrama de estados, diagrama de atividades, etc.)?

< Se a resposta for SIM fazer as perguntas I e II >

I. Quais modelos você sentiu dificuldade em modelar?

II. Como você conseguiu contornar essas dificuldades?

6) No projeto que participou você faria algo diferente ao descrever um caso de uso? Ou o que você não faria ao descrever um caso de uso?

Obrigada pela colaboração!