

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**PEDRO OLIVEIRA DE AZEVEDO**

**UMA CONTRIBUIÇÃO AO ENSINO DE MODELAGEM DE  
SOFTWARE PARA ESTUDANTES COM DEFICIÊNCIA  
VISUAL**

**DOIS VIZINHOS**

**2021**

**PEDRO OLIVEIRA DE AZEVEDO**

**UMA CONTRIBUIÇÃO AO ENSINO DE MODELAGEM DE  
SOFTWARE PARA ESTUDANTES COM DEFICIÊNCIA  
VISUAL**

**A CONTRIBUTION TO THE SOFTWARE MODELING  
TEACHING FOR VISUALLY IMPAIRED STUDENTS**

Trabalho de Conclusão de Curso apresentado  
como requisito à obtenção do título de Bacharel  
em Engenharia de Software, da Universidade  
Tecnológica Federal do Paraná.

Orientadora: Prof<sup>ª</sup>. Dr<sup>ª</sup>. Alinne C. Corrêa  
Souza

**DOIS VIZINHOS**

**2021**



Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

**TERMO DE APROVAÇÃO**
**TRABALHO DE CONCLUSÃO DE CURSO - TCC**
**UMA CONTRIBUIÇÃO AO ENSINO DE MODELAGEM DE SOFTWARE PARA ESTUDANTES COM DEFICIÊNCIA VISUAL**

Por

Pedro Oliveira de Azevedo

Monografia apresentada às 16horas 30min. do dia 25 de agosto de 2021 como requisito parcial, para conclusão do Curso de Bacharelado em Engenharia de Software da Universidade Tecnológica Federal do Paraná, Câmpus Dois Vizinhos. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação e conferidas, bem como achadas conforme, as alterações indicadas pela Banca Examinadora, o trabalho de conclusão de curso foi considerado APROVADO.

Banca examinadora:

|                                       |                             |
|---------------------------------------|-----------------------------|
| Prof. Gustavo Jansen de Souza Santos  | Membro                      |
| Prof. Ricardo Ferreira Vilela         | Membro                      |
| Profa. Alinne Cristinne Corrêa Souza  | Orientadora                 |
| Prof. Francisco Carlos Monteiro Souza | Professor responsável TCCII |



Documento assinado eletronicamente por (Document electronically signed by) **ALINNE CRISTINNE CORREA SOUZA, PROFESSOR DO MAGISTERIO SUPERIOR**, em (at) 31/08/2021, às 14:39, conforme horário oficial de Brasília (according to official Brasilia-Brazil time), com fundamento no (with legal based on) art. 4º, § 3º, do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por (Document electronically signed by) **GUSTAVO JANSEN DE SOUZA SANTOS, PROFESSOR DO MAGISTERIO SUPERIOR**, em (at) 31/08/2021, às 14:43, conforme horário oficial de Brasília (according to official Brasilia-Brazil time), com fundamento no (with legal based on) art. 4º, § 3º, do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por (Document electronically signed by) **Ricardo Ferreira Vilela, Usuário Externo**, em (at) 31/08/2021, às 14:44, conforme horário oficial de Brasília (according to official Brasilia-Brazil time), com fundamento no (with legal based on) art. 4º, § 3º, do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por (Document electronically signed by) **Pedro Oliveira De Azevedo, Usuário Externo**, em (at) 31/08/2021, às 18:37, conforme horário oficial de Brasília (according to official Brasilia-Brazil time), com fundamento no (with legal based on) art. 4º, § 3º, do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por (Document electronically signed by) **FRANCISCO CARLOS MONTEIRO SOUZA, PROFESSOR DO MAGISTERIO SUPERIOR**, em (at) 31/08/2021, às 18:38, conforme horário oficial de Brasília (according to official Brasilia-Brazil time), com fundamento no (with legal based on) art. 4º, § 3º, do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no site (The authenticity of this document can be checked on the website) [https://sei.utfpr.edu.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](https://sei.utfpr.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador (informing the verification code) **2217799** e o código CRC (and the CRC code) **187AF3ED**.



## **AGRADECIMENTOS**

Agradeço à força divina que habita em mim e permeia tudo que é vivo.

Agradeço aos meus familiares e amigos pelo suporte e conselhos.

Agradeço à minha orientadora por seu voto de confiança.

*Rictus Sempra.*

ROWLING, J.K (2001).

## RESUMO

A modelagem de software é uma atividade fundamental no processo de desenvolvimento de software. Uma forma de realizar a modelagem é com auxílio da Linguagem de Modelagem Unificada. Entretanto, estudantes portadores de deficiência visual são desafiados ao modelarem, devido essa atividade ser realizada por meio de diagramas, compostos por formas geométricas e conectores. Neste contexto, essa monografia apresenta uma linguagem chamada *Blind Modeling Language (BML)* para padronizar a especificação de requisitos a fim de facilitar a geração automática de diagramas de caso de uso. Para viabilizar a BML foi desenvolvida a ferramenta chamada *Blind Modeling software (B-Model)*, que visa auxiliar estudantes com deficiência visual na atividade de modelagem de software utilizando a Linguagem de Modelagem Unificada. A *BML* foi avaliada por meio de um estudo de caso com três estudantes deficientes visuais. Além disso, também foi realizado um experimento para avaliar a eficácia da ferramenta usando a especificação de requisitos de quatro cenários diferentes especificados por estes estudantes. Os resultados sugerem a aceitação e uso da linguagem *BML* e a eficácia da *B-Model*, pois os diagramas de caso de uso gerados correspondem aos requisitos funcionais especificados utilizando a *BML*.

**Palavras-chave:** Deficientes visuais, Modelagem de software, UML, Linguagem Específica

## ABSTRACT

Software modeling is a fundamental activity in the software development process. One way to accomplish this activity is by the Unified Modeling Language teaching, widely known as UML. However, visually impaired students must overcome their impairment in modeling activity, because it is performed through diagrams that are composed of geometric shapes and connectors. A language called Blind Modeling Language (BML) was created to standardize the requirements specification to facilitate the automatic generation of use case diagrams. For making the use of the BML feasible was developed a Blind Modeling software (B-Model) tool to assist visually impaired students on software modeling activity using Unified Modeling Language. The BML was evaluated through a case study with three blind students and the tool was assessed on an experiment using the requirements specification of four different scenarios performed by these students. The results suggest the acceptance and use of BML language and the effectiveness of the B-Model tool since the generated use case diagrams correspond to the specified functional requirements using the BML.

**Keywords:** Visually impaired, Software Modeling, UML, Specific Language



## LISTA DE FIGURAS

|   |    |
|---|----|
| Figura 1 – Visão geral da metodologia. ....   | 16 |
| Figura 2 – Diagrama de caso de uso para um sistema educacional .....  | 24 |
| Figura 3 – Resultado da análise sintática. ....   | 26 |
| Figura 4 – Definição da <i>string</i> de busca .....  | 30 |
| Figura 5 – Processo de seleção dos estudos primários .....  | 33 |
| Figura 6 – Número de publicações por ano .....  | 34 |
| Figura 7 – Ocorrência de estudos primários por categoria .....  | 34 |
| Figura 8 – Quantidade de estudos primários por tipo de publicação .....   | 35 |
| Figura 9 – Número de estudos incluídos por país .....   | 36 |
| Figura 10– Quantitativo do tipo de validação realizada pelos estudos primários .....  | 37 |
| Figura 11– Números de estudos por diagramas contemplados .....  | 37 |
| Figura 12– Visão geral da ferramenta B-Model. ....  | 46 |
| Figura 13– Gramática referente a linguagem BML. ....  | 49 |
| Figura 14– Exemplo de uma especificação de requisitos utilizando a <i>BML</i> . ....  | 50 |
| Figura 15– Exemplo da geração do diagrama de caso de uso a partir da especificação de requisitos utilizando a <i>BML</i> . .... | 51 |
| Figura 16– c <sub>1</sub> - Sistema de vendas de produtos eletrônicos. ....   | 61 |
| Figura 17– c <sub>2</sub> - Sistema de gerenciamento de festas .....  | 62 |
| Figura 18– c <sub>3</sub> - Sistema de locação de veículos .....  | 63 |
| Figura 19– c <sub>4</sub> - Sistema de gerenciamento de consultas de uma clínica veterinária ...                                | 64 |

## LISTA DE TABELAS

|  |    |
|--|----|
| Tabela 1 – Fontes de busca automática .....  | 30 |
| Tabela 2 – Visão geral dos estudos primários incluídos .....   | 35 |
| Tabela 3 – Visão geral dos estudos primários referente aos diagramas contemplados e contexto .....                                     | 38 |
| Tabela 4 – Categorias definidas para BML com base na POS. ....   | 47 |
| Tabela 5 – Categorias definidas para BML com base nos conceitos do diagrama de caso de uso. ....                                       | 48 |
| Tabela 6 – Variáveis do modelo TAM utilizadas para avaliar a <i>BML</i> . ....   | 54 |
| Tabela 7 – Descrição dos quatro cenários utilizados para avaliar a <i>BML</i> e a <i>B-Model</i> . ....                                | 55 |
| Tabela 8 – Resultados da avaliação referente à facilidade de uso percebida, utilidade percebida e intenção de uso da <i>BML</i> . .... | 57 |

## LISTA DE SIGLAS

|       |  |
|-------|--|
| AABB  | Associação Atlética do Banco do Brasil     |
| UTFPR | Universidade Tecnológica Federal do Paraná |

## SUMÁRIO

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>INTRODUÇÃO</b> .....   | <b>13</b> |
| 1.1      | Motivação .....   | 14        |
| 1.2      | Objetivos .....   | 15        |
| 1.3      | Metodologia de Pesquisa .....   | 15        |
| 1.4      | Estrutura da Monografia .....   | 17        |
| <b>2</b> | <b>ASPECTOS CONCEITUAIS</b> .....   | <b>18</b> |
| 2.1      | Deficiência Visual .....  | 18        |
| 2.2      | Inclusão e Deficiência Visual no âmbito educacional .....   | 19        |
| 2.3      | Tecnologias Assistivas .....  | 20        |
| 2.4      | Engenharia de Requisitos .....  | 21        |
| 2.5      | Modelagem de sistemas .....   | 22        |
| 2.5.1    | Diagrama de Caso de Uso .....   | 23        |
| 2.6      | Processamento de Linguagem Natural .....  | 24        |
| 2.6.1    | Análise léxica .....  | 25        |
| 2.6.2    | Análise Sintática .....   | 26        |
| 2.6.3    | Análise Semântica .....   | 26        |
| 2.7      | Considerações finais .....  | 27        |
| <b>3</b> | <b>MAPEAMENTO SISTEMÁTICO</b> .....   | <b>28</b> |
| 3.1      | Planejamento do Mapeamento Sistemático .....  | 28        |
| 3.1.1    | Estratégia para Busca Automática .....  | 29        |
| 3.1.2    | Critérios para Seleção de Estudos Primários .....   | 30        |
| 3.1.3    | Extração de Dados .....   | 31        |
| 3.2      | Condução do Mapeamento Sistemático .....  | 32        |
| 3.3      | Análise e Síntese dos Estudos Primários .....   | 32        |
| 3.4      | Ameaças à validade .....  | 43        |
| 3.5      | Considerações finais .....  | 43        |
| <b>4</b> | <b>FERRAMENTA <i>BLIND MODELING SOFTWARE (B-MODEL)</i><br/>E LINGUAGEM <i>BLIND MODELING LANGUAGE (BML)</i></b> ... | <b>45</b> |

|            |   |           |
|------------|---|-----------|
| <b>4.1</b> | <b>Ferramenta B-Model</b> .....                                 | <b>45</b> |
| 4.1.1      | Fase 1: Especificação de Requisitos Funcionais (ERF) .....      | 46        |
| 4.1.2      | Fase 2: Interpretação dos Requisitos Funcionais (IRF) .....     | 46        |
| 4.1.3      | Fase 3: Geração do Diagrama (DC) .....                          | 50        |
| <b>4.2</b> | <b>Considerações finais</b> .....                               | <b>51</b> |
| <b>5</b>   | <b>AVALIAÇÃO EXPERIMENTAL E DISCUSSÃO DOS RESULTADOS</b>        |           |
|            | <b>52</b>   |           |
| <b>5.1</b> | <b>Estudo de Caso: Utilidade da linguagem <i>BML</i></b> .....  | <b>52</b> |
| 5.1.1      | Planejamento .....  | 52        |
| 5.1.2      | Condução .....  | 53        |
| <b>5.2</b> | <b>Experimento: Eficácia da ferramenta <i>B-Model</i></b> ..... | <b>54</b> |
| 5.2.1      | Definição do Experimento .....                                  | 55        |
| 5.2.2      | Design do Experimento .....                                     | 56        |
| 5.2.3      | Condução do Experimento .....                                   | 56        |
| <b>5.3</b> | <b>Análise e discussão dos resultados</b> .....                 | <b>56</b> |
| 5.3.1      | Aceitação e uso da linguagem <i>BML</i> .....                   | 57        |
| 5.3.2      | Eficácia da ferramenta <i>B-Model</i> .....                     | 58        |
| 5.3.3      | Ameaças à Validade .....  | 59        |
| <b>5.4</b> | <b>Considerações Finais</b> .....                               | <b>59</b> |
| <b>6</b>   | <b>CONSIDERAÇÕES FINAIS</b> .....                               | <b>65</b> |
|            | <b>REFERÊNCIAS</b> .....  | <b>68</b> |
|            | <b>Apêndice A – ARTIGO PUBLICADO NO ERES 2020</b> .....         | <b>72</b> |

## 1 INTRODUÇÃO

O uso de Tecnologias da Informação e da Comunicação (TICs) é inerente ao cotidiano acadêmico. No entanto, apesar da relevância que há em torno da inclusão social e digital, ainda existem indivíduos que não possuem acesso ou condições físicas para desfrutar os benefícios das TICs.

Segundo o Instituto Brasileiro de Geografia e Estatística - IBGE (2010), 45 milhões de pessoas, ou seja, 23,9% da população total, apresentaram algum tipo de incapacidade ou deficiência. No que diz respeito à limitação visual, o IBGE informa que cerca de 6,5 milhões de brasileiros apresentam esta deficiência, sendo 582 mil cegos e 6 milhões com baixa visão.

Embora a deficiência visual não seja a mais predominante na população brasileira, esses indivíduos enfrentam diversos desafios no plano educacional (ZITKUS et al., 2016); (NETO et al., 2015). Dentre os desafios que estudantes portadores de deficiência visual se deparam é importante destacar a carência de recursos didáticos acessíveis; e a falta de capacitação de profissionais para atuarem neste contexto de ensino (THOMA, 2006).

Para amenizar esses desafios é preciso desenvolver soluções utilizando TICs que promovam a inclusão de pessoas com deficiência visual seja plano social ao digital. Dentre as TICs existentes, destacam-se as Tecnologias Assistivas (TAs). As TAs podem ser aliadas no processo de ensino e aprendizado de estudantes com deficiência visual, uma vez que podem ser vistas como um apoio computacional para minimizar a deficiência, incapacidade ou desvantagem identificada naqueles estudantes.

Nesse contexto, a utilização de T.As nos cursos de computação do ensino superior pode ser considerada como um indicativo de qualidade de ensino, independência e inclusão para estudantes com deficiência visual. Nesse contexto, as T.As contribuem para redução das diferenças nas Universidades, praticando a inclusão social e digital.

Apesar dos diversos esforços observados desde o vestibular até a adequação física de espaços para incluir esses estudantes ao ensino superior, ainda existe um campo vasto

a ser explorado no processo da formação acadêmica. É necessário garantir que esses estudantes ingressantes comecem e finalizem o curso. Nesse cenário, a acessibilidade no ensino superior é crucial para combater a taxa de evasão de estudantes visualmente impedidos, bem como aumentar suas expectativas de empregabilidade.

A inclusão de um estudante com limitação visual é realizada quando são desenvolvidas competências que possibilitem alternativas de acesso ao plano educacional. No entanto, desenvolver essas competências em um estudante com deficiência visual pode ser desafiador no ensino de conteúdos que necessitam de representações gráficas. Nos cursos de computação, por exemplo, essas representações gráficas são utilizadas na modelagem de software.

A modelagem é uma etapa fundamental no processo de desenvolvimento de software. Ela apoia as especificações de um software por meio de diagramas, que permitem a visualização das funcionalidades e seus processos antes mesmo de desenvolvê-los.

## 1.1 Motivação

Um sério desafio que professores, da esfera federal, enfrentam ao incluir estudantes com deficiência visual no plano de ensino é a carência de capacitação pedagógica continuada. No que concerne aos cursos de computação, os deficientes visuais têm historicamente uma participação notória. Essa participação ocorre devido à existência de T.As, tais como leitores de tela, que permitem a compreensão de códigos fonte, pois são essencialmente texto. No entanto, algumas atividades da Engenharia de Software (ES) são dependentes de sentidos específicos, como a visão.

Nesse contexto, a modelagem de software é realizada por meio de diagramas visuais, compostos por formas geométricas e conectores relacionando-as. Para que os estudantes sejam capazes de modelarem um software, o ensino da Linguagem de Modelagem Unificada (do inglês, *Unified Modeling Language - UML*) torna-se fundamental nos cursos de computação. Apesar da sua importância, estudos apontam que o ensino dessa linguagem é complexo e que a maioria dos projetistas encontram obstáculos relacionados às suas especialidades cognitivas, como a metodologia de alfabetização ministrada aos estudantes com deficiência por exemplo (FLINT; GARDNER; BOUGHTON, 2004); (KUZNIARZ; BORSTLER, 2011); (BERA, 2012)).

É importante destacar que os diagramas gerados por meio da UML são utilizados com frequência para explicar conceitos da modelagem, projeto ou mesmo documentar manuais de uso do software. Uma aplicação desses conceitos, seria estudantes de UML

aprenderem que uma funcionalidade do sistema é representada por uma elipse no diagrama de caso de uso (em notação UML, caso de uso), e que essa funcionalidade é executada por uma pessoa ou sistema, que é representado por um boneco palito (em notação UML, ator).

Diante desse cenário, os diagramas se tornam inacessíveis para os estudantes com deficiência visual, devido à sua natureza gráfica. Além disso, é desafiador criar um meio comum em que estudantes portadores ou não da deficiência visual possam trabalhar juntos sobre a mesma modelagem. Portanto, o aprendizado de estudantes deficientes visuais fica prejudicado, sendo necessário o auxílio de outra pessoa tanto para a leitura quanto para a edição desses diagramas.

## 1.2 Objetivos

Diante da motivação apresentada, o presente trabalho tem como objetivo o desenvolvimento de uma linguagem textual específica chamada *Blind Model Language (BML)* para padronizar a especificação de requisitos. A *BML* foi viabilizada por meio da ferramenta nomeada de *Blind Modeling software (B-Model)* com a finalidade de auxiliar estudantes com deficiência visual na modelagem de software, ou seja, permitir a geração automática do diagrama de caso de uso a partir dos requisitos especificados com a *BML*. No contexto acadêmico a geração do diagrama serve ao propósito de avaliação dos discentes na realização da modelagem. Enquanto no contexto profissional, a geração do diagrama viabiliza a comunicação entre deficientes e não deficientes nessa atividade.

Para alcançar o objetivo geral os seguintes objetivos específicos foram delineados:

- desenvolver uma linguagem específica para auxiliar a representação do diagrama de caso de uso;
- desenvolver uma ferramenta para geração de diagramas de caso de uso utilizando a linguagem desenvolvida;
- avaliar por meio de estudo de caso e um experimento a linguagem e a ferramenta desenvolvidas, respectivamente.

## 1.3 Metodologia de Pesquisa

Para alcançar os objetivos definidos foram previstas seis atividades principais, contemplando atividades teóricas e práticas, para o desenvolvimento desse trabalho,



conforme é ilustrado na Figura 1.

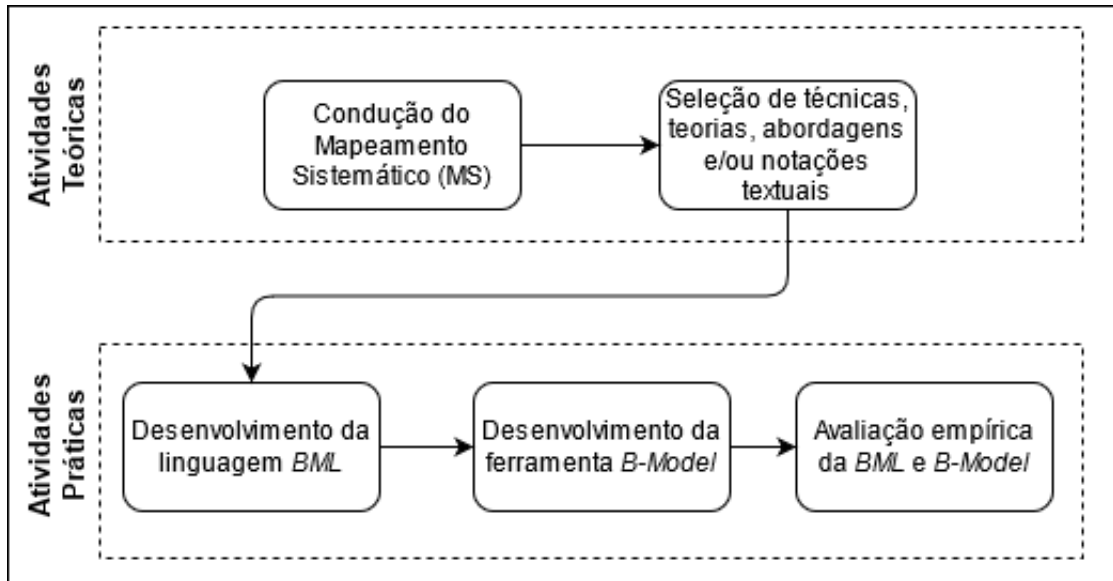


Figura 1 - Visão geral da metodologia.

Fonte: Autoria própria.

- **Condução do Mapeamento Sistemático (MS):** nesta atividade foi conduzido um MS a fim de investigar notações textuais, tecnologias e/ou ferramentas para o ensino de modelagem de software a pessoas deficientes visuais.
- **Seleção de técnicas, teorias, abordagens e/ou notações textuais:** a partir do MS, foram selecionadas as técnicas de Processamento de Linguagem Natural (PLN), do inglês *Natural Language Processing (NLP)*, *Parts of Speech (POS)* e Teoria de *Chomsky* para auxiliar na criação da gramática da BML bem como seu processamento léxico e semântico.
- **Desenvolvimento da linguagem BML:** nesta atividade foi construída a linguagem BML para padronizar a especificação de requisitos a fim de facilitar a geração automática do diagrama de caso de uso.
- **Desenvolvimento da ferramenta B-Model:** nesta atividade foi desenvolvido um protótipo para interpretar os requisitos especificados utilizando a BML para gerar o diagrama de caso de uso.
- **Avaliação empírica da BML e B-Model:** nesta atividade foi conduzido um estudo de caso e um experimento para avaliar a eficácia da BML e da B-Model, respectivamente.

## 1.4 Estrutura da Monografia

Esta monografia está organizada em seis capítulos. No Capítulo 2, são definidos os conceitos fundamentais de deficiência visual, o processo de inclusão de deficientes visuais no âmbito educacional, T.As, Engenharia de Requisitos (ER), modelagem de software utilizando UML e PLN para auxiliar na compreensão deste trabalho.

No Capítulo 3 é apresentada a condução do Mapeamento Sistemático (MS), o qual visa identificar evidências do ensino para deficientes visuais da modelagem de software utilizando a UML.

No Capítulo 4 são apresentadas a linguagem *BML* e a ferramenta *B-Model* para auxiliar estudantes com deficiência visual na modelagem de software.

O Capítulo 5 apresenta o estudo de caso e o experimento conduzidos, bem como a discussão dos resultados alcançados. Por fim, no Capítulo 6 são apresentadas as principais conclusões, contribuições, limitações deste trabalho, bem como os trabalhos futuros.

## 2 ASPECTOS CONCEITUAIS

Neste Capítulo os aspectos conceituais pertinentes ao tema de pesquisa são detalhados a fim de contextualizar e introduzir uma problemática: como realizar a inclusão de pessoas com deficiência visual no âmbito educacional.

A organização deste capítulo se dispõe: na Seção 2.1 é apresentado o conceito de deficiência visual; na Seção 2.2 é detalhado o processo de inclusão do estudante com deficiência visual no âmbito educacional; a Seção 2.3 aborda T.As; na Seção 2.4 descreve-se a Engenharia de Requisitos (ER); a Seção 2.5 aborda os conceitos de UML, destacando o diagrama de caso de uso; a Seção 2.6 apresenta PLN e suas etapas; e finalizando com as considerações finais na Seção 2.7.

### 2.1 Deficiência Visual

No último censo realizado pelo IBGE (2010), existem cerca de 6,5 milhões de brasileiros com algum tipo de deficiência visual, sendo 582 mil cegos e 6 milhões com baixa visão. A deficiência visual apresenta uma diversidade conceitual, sendo difícil estabelecer um consenso entre os autores para o estabelecimento das definições (COSTA et al., 2009).

Segundo Tonet (2006), a deficiência visual refere-se à falta, insuficiência ou imperfeição da imagem visual que causa a diminuição ou situação irreversível da visão. Essa deficiência consiste em uma limitação sensorial, na qual os indivíduos que são portadores têm perdas visuais variáveis, podendo ir de resíduos visuais, para ausência da percepção da luz, e até mesmo a total ausência da visão.

Deficiência visual carrega o termo ‘cegueira’ em sua definição, visto no Decreto 5.296/04 Art. 5º (BRASIL, 2004):

“Deficiência visual: cegueira, na qual a acuidade visual é igual ou menor que 0,05 no melhor olho, com a melhor correção óptica; a baixa visão, que significa acuidade visual entre 0,3 e 0,05 no melhor olho, com a melhor correção óptica; os casos nos quais a somatória da medida do campo visual em ambos os olhos for

igual ou menor que 60<sup>o</sup>; ou a ocorrência simultânea de quaisquer das condições anteriores” (BRASIL, 2004).

É importante destacar que segundo Gambarato et al. (2012) existem diferentes termos relacionados à deficiência visual, tais como:

- **Visão subnormal ou baixa visão:** definem como pessoas que são capazes de, mesmo com a limitação sensorial, realizarem atividades cotidianas sem muita dificuldade. Mas para isso, é necessário o auxílio óptico e não-óptico (SUBNORMAL, 2016).
- **Cegueira:** pode ser entendida como o grau mais severo de deficiência que acomete a visão, configurando à pessoa total perda deste sentido. Apesar de irreversível é possível amenizar sua severidade com o auxílio de T.As aliadas às metodologias alternativas de ensino.

Neste contexto, a limitação visual de um estudante não implica na sua exclusão, mas exige um acompanhamento para minimizar os desafios de inclusão. Assim, o papel do professor é fundamental para proporcionar um cenário de ensino mais inclusivo, estimular as potencialidades do estudante com deficiência visual e oportunizar que este aprenda e atue sem distinções em qualquer disciplina. Para isso, é necessário que sejam utilizados meios de acessibilidade para viabilizar sua inclusão no ensino.

## 2.2 Inclusão e Deficiência Visual no âmbito educacional

No contexto da deficiência visual e seus portadores, pouco se entende sobre as diferenças entre inserir e incluir um estudante no âmbito educacional. Segundo Marshall apud Brumer et al. (2004), a inclusão social contempla os direitos e deveres de um cidadão quando pertencente à sociedade. No que tange aos direitos compreendidos pela cidadania, destacam-se os sociais que visam assegurar o bem-estar do indivíduo, ou seja, direitos à saúde, à segurança, ao lazer, ao trabalho, e à educação.

É importante destacar que o processo de inclusão consiste em garantir que um membro da sociedade tenha acesso, sem distinção, a todos os direitos. No entanto, o cenário acadêmico atual não consegue incluir um estudante, apenas inseri-lo no ensino superior, ou seja, permitir que este ingresse em um curso sem garantir sua participação ativa em qualquer disciplina e/ou atividade devido à deficiência visual.

Para Leite e Silva (2006) incluir um estudante visualmente limitado é buscar ouvi-lo a fim de oferecer um ensino equitativo de acordo com suas necessidades. A realização da educação inclusiva é apoiada por leis, decretos e portarias que estabelecem critérios básicos para esse fim. Dentre estas é importante destacar as leis 7.853/89; 9.394/96; 10.098/00 e 10.172/01; bem como os decretos 5.296/04 e 3.298/99 e a portaria 1.679/99 do Ministério da Educação. Infere-se que essa legislação sobre o direito ao acesso da educação por estudantes com deficiência visual, atribui também a responsabilidade de instaurar políticas inclusivas à comunidade acadêmica. Portanto, é necessária a adequação do espaço e a capacitação de professores para receber tais estudantes no ensino superior.

Seguindo este contexto, Dallabona (2011) contribui com noções importantes em seu trabalho a respeito da adequação para recepção e garantia da permanência dos estudantes com deficiência visual no ensino superior por meio do suporte das TICs, especialmente as T.As.

### 2.3 Tecnologias Assistivas

A ideia de Tecnologia nasceu na Grécia antiga a partir do diálogo filosófico de Platão. Ao diferenciar o que é natureza do que é artefato, ele propôs a reflexão sobre *techné*, ou seja, capacidade de produção de um artefato baseado no conhecimento prático, contraposta à episteme, o conhecimento em estado puro, ou seja, ciência (KUSSLER, 2015).

Produzir tecnologia consiste na capacidade humana de construir soluções que auxiliam ou completam a natureza de acordo com as necessidades observadas. Um estudante com deficiência visual, por exemplo, utiliza uma T.A para superar a limitação que acomete o estado natural da visão. Neste cenário, segundo o Comitê de Ajudas Técnicas - CAT (2011), a T.A é conceituada da seguinte forma:

“Tecnologia Assistiva é uma área do conhecimento, de característica interdisciplinar, que engloba produtos, recursos, metodologias, estratégias, práticas e serviços que objetivam promover a funcionalidade, relacionada à atividade e participação, de pessoas com deficiência, incapacidades ou mobilidade reduzida, visando sua autonomia, independência, qualidade de vida e inclusão social”.

É possível aplicar esse conceito no contexto de acessibilidade, uma vez que a T.A é entendida como um mecanismo ou um artefato oferecido ao estudante com deficiência visual a fim de mitigar barreiras no processo de ensino e aprendizado em sala de aula. No campo da TI as seguintes T.As de acesso ao computador são identificadas:

- **Braille Fácil<sup>1</sup> e DOSVOX<sup>2</sup>**: ambos os softwares foram desenvolvidos no Núcleo de Computação Eletrônica (NCE) da Universidade Federal do Rio de Janeiro. O primeiro software é utilizado para transcrição de textos corridos para o braille que podem ser impressos por qualquer impressora que tenha o driver instalado. O segundo consiste num sintetizador de voz que comunica ao usuário os textos digitados/exibidos na tela.
- **JAWS<sup>3</sup>**: do inglês, *Job Access With Speech*, é um leitor de tela que permite ao usuário com baixa visão navegar pela internet, escrever documentos, ler e-mail, além de gerar saídas em braille caso necessário.
- **NVDA<sup>4</sup>**: do inglês, *Non Visual Desktop Access*, é um leitor de telas para *Windows*.
- **Virtual vision<sup>5</sup>**: é um software brasileiro desenvolvido pela Micropower que possibilita a utilização do sistema operacional *Windows* e outros aplicativos por meio da leitura de tela.
- **Braille translator<sup>6</sup>**: é uma ferramenta web que converte textos corridos em braille.

Estudantes e/ou profissionais com deficiência visual podem acessar um computador por meio de T.As que auxiliam na leitura de tela. Tais tecnologias permitem com que essas pessoas possam exercer atividades variadas na área de TI. No que tange ao curso de computação e afins, um leitor de tela pode auxiliar em atividades da Engenharia de Requisitos (ER), por exemplo.

## 2.4 Engenharia de Requisitos

A ES é uma área de conhecimento tecnológico que imprime as diretrizes necessárias para a produção de um sistema sem defeitos, oferecendo soluções aos usuários, e prazo determinado (PRESSMAN, 2016). A ES se preocupa com todos os aspectos de produção do software. Um desses aspectos, contidos no processo de desenvolvimento de software, é a ER.

Dentre as atividades abarcadas pela ER, a análise de requisitos é a que conduz o levantamento e a documentação de informações sobre o domínio para o qual pretende-se

<sup>1</sup>Braille Fácil - NCE/UFRJ <<http://intervox.nce.ufrj.br/brfacil/>>

<sup>2</sup>Projeto DOSVOX - NCE/UFRJ <<http://intervox.nce.ufrj.br/dosvox/>>

<sup>3</sup>JAWS® – Freedom Scientific <<https://www.freedomscientific.com/products/software/jaws/>>

<sup>4</sup>NV Access <<https://www.nvaccess.org/>>

<sup>5</sup>O que é o Virtual Vision? <<https://www.virtualvision.com.br/Virtual-Vision/O-Que-E-O-Virtual-Vision.aspx>>

<sup>6</sup>Braille Translator <<https://www.brailletranslator.org/>>

desenvolver uma solução. Ao elicitar os requisitos do software é possível classificá-los em dois grupos: Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF). Esses requisitos consistem nas descrições das funcionalidades a serem desenvolvidas e restrições operacionais do software, respectivamente (SOMMERVILLE, 2016).

No que tange ao primeiro grupo (RF), estes descrevem o que o software deve fazer para atender as necessidades dos *stakeholders*<sup>7</sup>. Um RF pode ser descrito da perspectiva do usuário, que é uma maneira de explicar o comportamento do software quando o usuário interage com ele; e também pode ser descrito da perspectiva do próprio software, que consiste no detalhamento das funções do software como suas entradas, saídas e exceções.

O processo de levantamento de requisitos apoia, fundamentalmente, o desenvolvimento do software. No entanto, para que essas informações escritas em linguagem natural sejam compreendidas pela equipe de desenvolvimento é crucial realizar a modelagem do software. Essa modelagem consiste na representação gráfica dos requisitos com a finalidade de facilitar a comunicação entre todos os envolvidos no processo de desenvolvimento de software.

## 2.5 Modelagem de sistemas

A importância da modelagem, segundo Booch, Rumbaugh e Jacobson (2006), está diretamente relacionada ao sucesso da empresa de software e à garantia na entrega de um produto final de qualidade aos usuários. Para auxiliar a modelagem dos sistemas utiliza-se a Linguagem de Modelagem Unificada (do inglês, *Unified Modeling Language - UML*).

A UML foi criada por Booch, Jacobson e Rumbaugh em 1994 para especificar, documentar, construir, e visualizar a arquitetura do software. Em 1997 foi adotada pela OMG<sup>8</sup>, instituição que regulariza e aprova padrões abertos para aplicações orientadas a objetos, como linguagem padrão. A UML surgiu com a unificação das notações criadas por esses três autores que desenvolveram metodologias próprias para modelagem de sistemas Orientado a Objetos (OO).

A modelagem além de auxiliar na idealização do software por meio de notações gráficas, também colabora na comunicação daquilo que o software precisa fazer para resolver para os usuários. Essas notações são chamadas de diagramas e permitem compreender desde a estrutura até o comportamento do software a ser desenvolvido.

---

<sup>7</sup>*Stakeholders* são usuários que, direta ou indiretamente, possuem alguma relação de interesse para com as funcionalidades e serviços que o software oferece (PRESSMAN, 2016).

<sup>8</sup>Object Management Group: OMG <<https://www.omg.org/>>

É importante destacar que os diagramas são divididos em: (i) estruturais e (ii) comportamentais. Os diagramas estruturais contribuem com a visualização lógica e estática do software. Dentre os diagramas estruturais é importante destacar: diagramas de classe, objeto, pacotes, componentes e implantação. Por outro lado, os diagramas comportamentais viabilizam a compreensão das funcionalidades a serem desenvolvidas. Os diagramas comportamentais estão classificados em diagramas de caso de uso, atividades, sequência, máquina de estados e comunicação. Dentre comportamentais, o diagrama de caso de uso foi escolhido como foco de trabalho pois este utilizado desde as primeiras fases do desenvolvimento do software.

### 2.5.1 Diagrama de Caso de Uso

Segundo Booch, Rumbaugh e Jacobson (2006) um caso de uso é uma descrição de um conjunto de sequências de ações que um sistema executa para produzir um resultado de valor observável por um ator. A importância do diagrama de caso de uso está no fato deste ser utilizado para demonstrar o comportamento do sistema às partes (atores ou sistemas externos) envolvidas quando interagem com o mesmo.

O diagrama pode ser usado no diálogo com o usuário na descoberta e validação de requisitos, uma vez que nele representam-se as funcionalidades do sistema. Os principais componentes do diagrama de caso de uso são: (i) ator, (ii) caso de uso e (iii) relacionamentos. O ator está associado aos papéis desempenhados por elementos externos ao sistema, podendo ser humano (usuário), organização, dispositivo de hardware ou outro sistema. Um ator é representado por um boneco palito.

O caso de uso representa uma funcionalidade do sistema, ou seja, um RF que visa descrever o que o sistema faz. Um caso de uso, graficamente, é representado por uma elipse. Por fim, os relacionamentos indicam que existe uma interação entre atores e casos de uso, bem como entre os próprios casos de uso. No que tange aos relacionamentos é importante destacar a associação, generalização e dependência (*include* e *extend*). Na Figura 2 é apresentado um exemplo de diagrama de caso de uso para um sistema educacional. Observa-se no diagrama:

- **Atores:** são representados pelo aluno, professor e Instituição de Ensino.
- **Caso de Uso:** representados por Realizar Login, Acessar Fórum, ou seja, todas as elipses.



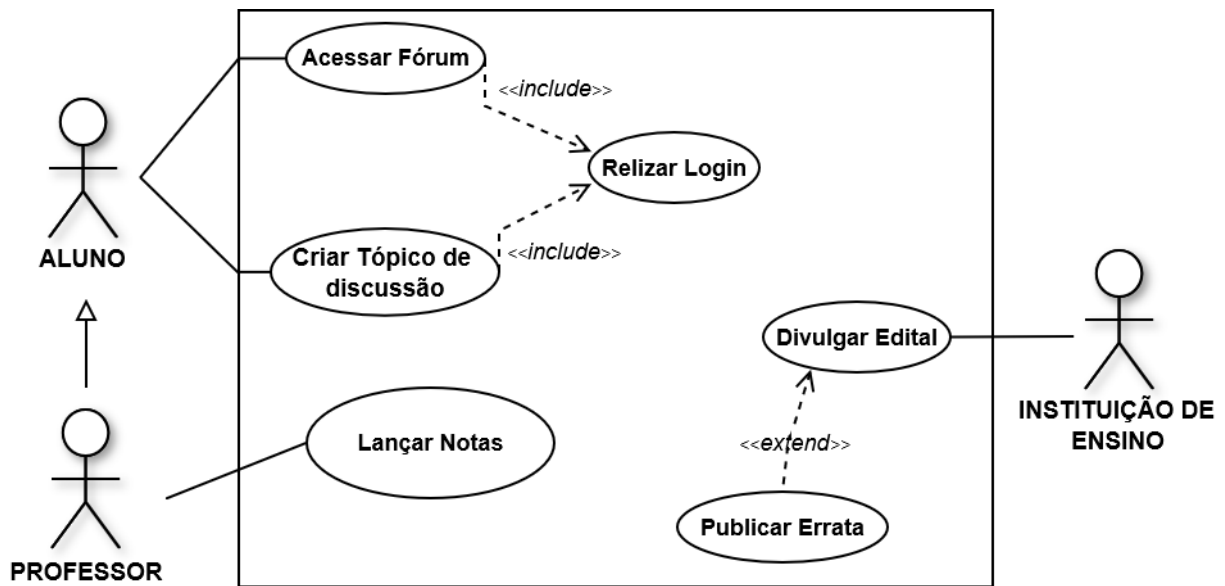


Figura 2 - Diagrama de caso de uso para um sistema educacional

Fonte: Autoria própria

- **Generalização de ator-ator:** A generalização especifica que um ator interage não só com o caso de uso em que está conectado, mas também com aqueles cujo o ator generalizado está relacionado. Professor, por exemplo, além de Realizar Login; Acessar Fórum e Criar Tópicos de discussão, que são herdados do ator Aluno, também é responsável por Lançar Notas.
- **Include:** para que a funcionalidade Acessar Fórum seja realizada, necessariamente o ator Aluno deve executar a funcionalidade Realizar Login.
- **Extend:** após realizar a funcionalidade Divulgar Edital, o ator Instituição de Ensino facultativamente realizaria a funcionalidade Publicar Errata.

Portanto, para a geração automática do diagrama de caso de uso a partir da especificação de requisitos em linguagem natural, diversas técnicas podem ser utilizadas, porém a de Processamento de Linguagem Natural (PLN) foi selecionada afim de apoiar os objetivos desse trabalho.

## 2.6 Processamento de Linguagem Natural

PLN consiste no desenvolvimento de modelos computacionais para a realização de tarefas que dependem de informações expressas em alguma língua natural, ou seja,

tradução e interpretação de textos, busca de informações em documentos e interface homem-máquina (RICH; KNIGHT, 1994).

É importante destacar que umas das principais barreiras é a compreensão da língua humana de uma forma que os computadores consigam reconhecer e interpretar essa mesma linguagem (TERESO, 2019). Neste contexto, o principal objetivo da PLN é estudar problemas de compreensão de linguagem natural humana. Portanto, PLN diz respeito à construção de softwares que sejam capazes de processar (compreender) informações em linguagem natural, conseguindo obter algum significado do texto que por ele é interpretado.

PLN geralmente faz uso de conceitos linguísticos, como partes do discurso (substantivo, verbo, adjetivo, entre outros), e estruturas gramaticais. Neste contexto, para o processo de compreensão de textos escritos é necessário analisadores léxicos-morfológicos, os quais leem um texto e realizam o tratamento a nível de palavras. Em seguida é produzido um conjunto de estruturas sintáticas relacionando os termos de uma frase à sua semântica, afim de criar uma representação final do significado de uma frase. Portanto, segundo Rich e Knight (1994) PLN é composta por três etapas: *(i)* Análise Léxica, *(ii)* Análise Sintática, e *(iii)* Análise Semântica.

### 2.6.1 Análise léxica

Na etapa da análise léxica, ocorre a conversão de uma cadeia de caracteres (o texto da consulta) em uma cadeia de palavras. Uma das funções da análise léxica é a *tokenização*, que consiste na decomposição de um texto em palavras, e estas palavras são denominadas de *tokens*.

Além disso, é realizada a normalização do texto em busca da padronização dos dados de entrada, colocando todo o texto em letras minúsculas, removendo *tokens* indesejados como espaços em branco, ou caracteres especiais (RICH; KNIGHT, 1994). Um exemplo para um melhor entendimento seria na frase: “O usuário realiza o cadastro.”, após a normalização e *tokenização* a frase ficaria da seguinte forma (“o”, “usuário”, “realiza”, “o”, “cadastro”, “.”). A nível de software, cada *token* é armazenado em um vetor ou *array*. Ao final é obtida uma matriz de *tokens* processados, organizados de forma lógica repetindo a gramática da linguagem BML construída.

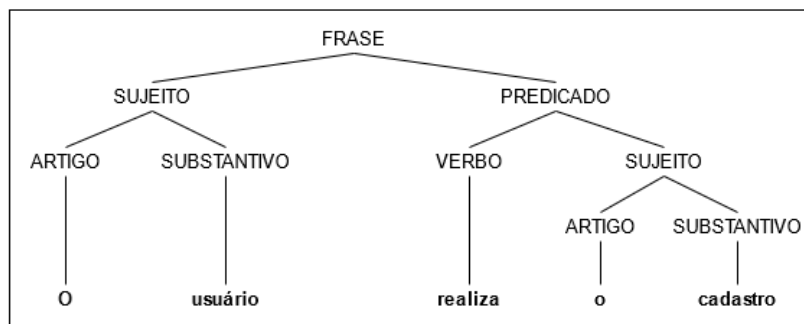
Por fim, junto a análise léxica é realizada a análise morfológica, a qual é responsável pela análise de palavras isoladas, ou seja, definição de classe gramatical também conhecida como *Part-Of-Speech (POS)*. Nesta análise, cada palavra é classificada em uma categoria

de acordo com as regras que regem a língua portuguesa (substantivo, adjetivo, numeral, verbo, advérbio, pronome, artigo, preposição, conjunção e interjeição). Por exemplo, se informar o texto “O usuário realiza o cadastro.”, a análise morfológica tem que saber que “usuário” é um substantivo, e que “realiza” é um verbo.

### 2.6.2 Análise Sintática

Esta etapa visa criar uma descrição estruturada das palavras em uma frase, utilizando como entrada os resultados obtidos com a análise morfológica (RICH; KNIGHT, 1994). Por exemplo, se fornecida à ferramenta a frase: “usuário o realiza o cadastro” como entrada, o analisador sintático a rejeitaria. O erro nesta frase, está em “Usuário o”. O substantivo “Usuário” e o artigo definido “o” estão fora de ordem, pois, segundo a gramática da língua portuguesa, um artigo deve antepor um substantivo.

Por outro lado, se analisada sintaticamente a frase “O usuário realiza o cadastro”, tem-se como resultado uma descrição bem estruturada desta frase, destacando as suas unidades, como artigo, sujeito, verbo, entre outros. Estas unidades são importantes quando for realizada a análise semântica (RICH; KNIGHT, 1994). A Figura 3 apresenta o resultado da análise sintática para o exemplo “O usuário realiza o cadastro”.



**Figura 3 - Resultado da análise sintática.**

**Fonte: Autoria própria.**

### 2.6.3 Análise Semântica

Esta etapa visa fornecer significado às estruturas criadas na análise sintática, ou seja, analisa o significado das palavras em uma frase. Nesse contexto, a análise semântica dos textos é realizada para tentar identificar a importância das palavras dentro da estrutura da frase (RICH; KNIGHT, 1994).

Por fim, existem as Gramáticas que são construídas ao final da análise semântica. É importante frisar que não é necessário criar estas gramáticas, mas quando usadas, há vantagens para PLN. Para Rich e Knight (1994), as principais vantagens são que com uma gramática já construída, é possível utilizar sua estrutura diretamente na aplicação de PLN. Outra vantagem bastante evidente é com relação às ambiguidades léxicas que são evitadas, pois a gramática já deixa bem definido o significado de cada palavra. Portanto, uma ferramenta de PLN precisa ter amplo conhecimento da linguagem natural que irá ser manipulada, e para realizar o processamento da linguagem é preciso ter fontes de informações.

Nesse contexto, é necessário o emprego de técnicas que auxiliem na mecânica e no desenvolvimento de um sistema capaz de interpretar a linguagem natural humana. Dentre as técnicas de pré-processamento previstas pela PLN, destacam-se as utilizadas nesse trabalho:

- (i) **Normalização:** consiste no tratamento da remoção de caracteres especiais, conversão de maiúsculo para minúsculo, bem como a categorização morfológica das palavras estruturando o texto em partes lexicais.
- (ii) **Remoção de stopwords:** visa a identificação e a remoção de palavras que não possuam relevância no domínio da especificação de requisitos, ou geração de caso de uso.
- (iii) **Stemização e Lematização:** implica na redução das palavras ao seu radical, a fim de reduzir o vocábulo e abstrair o significado delas.

## 2.7 Considerações finais

Com os conceitos apresentados neste Capítulo é possível afirmar que, a inclusão de estudantes com deficiência visual ao ensino superior e às atividades que envolvam manipulação de representações gráficas, devem ser estudados para viabilizar a estes estudantes, em sua carreira acadêmica e profissional, acesso às atividades que envolvem o desenvolvimento de software, a exemplo da modelagem de software utilizando a UML.

Para este público é crucial que, visando a plena atuação no âmbito educacional, sejam ofertados meios de acessibilidade para aprendizagem e prática da modelagem de software utilizando UML. No próximo Capítulo é apresentado um Mapeamento Sistemático (MS) com a finalidade de enriquecer as motivações deste trabalho.

### 3 MAPEAMENTO SISTEMÁTICO

A fim de contribuir cientificamente com aqueles quem também pesquisam, foram reunidas evidências referentes ao ensino de UML a estudantes visualmente impedidos. Além de reunidas as evidências, também foram realizadas sínteses, avaliações e divulgação dos dados coletados. Nesse contexto, existem diferentes técnicas da Engenharia de Software Baseada em Evidências - ESBE (do inglês, *Evidence-Based Software Engineering*) que auxiliam neste processo (KITCHENHAM et al., 2015).

Dentre tais técnicas é importante destacar o MS (do inglês, *Systematic Mapping Study - SM*). O MS executado neste trabalho conduziu a busca por referências que propuseram soluções possíveis a fim de incluir estudantes com deficiência visual no ensino a modelagem de software utilizando UML. Neste MS foi utilizado o método proposto por Kitchenham et al. (2010), que sugere três fases: *(i)* Planejamento; *(ii)* Condução; *(iii)* Análise.

A organização do MS está estruturada da seguinte maneira: na Seção 3.1 descreve-se o planejamento do MS, destacando a Questão de Pesquisa (QP) elaborada; a estratégia e as fontes de busca utilizadas; bem como critérios de seleção dos estudos; a concepção das categorias dos estudos e a extração de dados. Na Seção 3.2 é apresentado o processo de condução do MS de acordo com a QP. A Seção 3.3 aborda a síntese e análise dos estudos primários. Na Seção 3.4 são apresentadas as ameaças à validade deste MS. E por fim, a Seção 3.5 dispõe as considerações finais deste Capítulo.

#### 3.1 Planejamento do Mapeamento Sistemático

Nesta fase é necessário estabelecer um protocolo que oriente a condução do MS. O planejamento descreve os processos e métodos revisados nessa atividade como: definição da QP, as bases de busca consultadas, a *string* de busca, os critérios para inclusão e exclusão de estudos, e então da forma de extração de dados.

Segundo Kitchenham et al. (2010), uma QP deve ser concebida levando em

consideração a técnica *PICO* (*Population, Intervention, Comparison e Outcomes*). Sendo que:

- **População (*Population*):** é campo de pesquisa sobre o qual se está buscando estudos relacionados. No contexto vigente a população consiste na modelagem de sistemas utilizando UML para deficientes visuais.
- **Intervenção (*Intervention*):** consiste em soluções encontradas na literatura para aquele domínio alvo. Para esse MS a intervenção poderá ser considerada como uma metodologia, ferramenta, técnica, notação textual ou abordagem que visa auxiliar deficientes visuais nas atividades de modelagem de sistemas.
- **Comparação (*Comparison*):** não se aplica no contexto deste MS.
- **Resultados (*Outcomes*):** é o que se espera/busca observar nos estudos primários retornados. Para esse contexto foram retornados estudos propondo metodologia, ferramenta, técnica, notação textual ou abordagem no domínio supracitado.

Logo, a partir desses conceitos, foi criada a seguinte Questão de Pesquisa:

- **QP: Quais são as evidências existentes referentes ao ensino da modelagem de sistemas utilizando UML para estudantes com deficiência visual?**

Com esta QP foi possível identificar estudos propondo metodologias, ferramentas, técnicas, notação textual e/ou abordagens para apoiar estudantes com deficiência visual na modelagem de sistemas utilizando UML.

### 3.1.1 Estratégia para Busca Automática

A estratégia desenvolvida consistiu na seleção das fontes de busca automática e na geração da *string*. A recuperação de estudos pertinentes ao tema ocorre por meio da aplicação da *string* nas fontes. A *string* de busca original sofreu variações de acordo com a necessidade de cada fonte de busca. Para construir a *string* de busca foram seguidos quatro passos: *(i)* identificação dos termos (*keywords*) relevantes à QP; *(ii)* identificação dos sinônimos desses termos em inglês; *(iii)* utilização do operador booleano “**OR**” entre os sinônimos; e *(iv)* utilização do operador booleano “**AND**” para concatenar os principais termos, conforme pode ser visto na Figura 4.

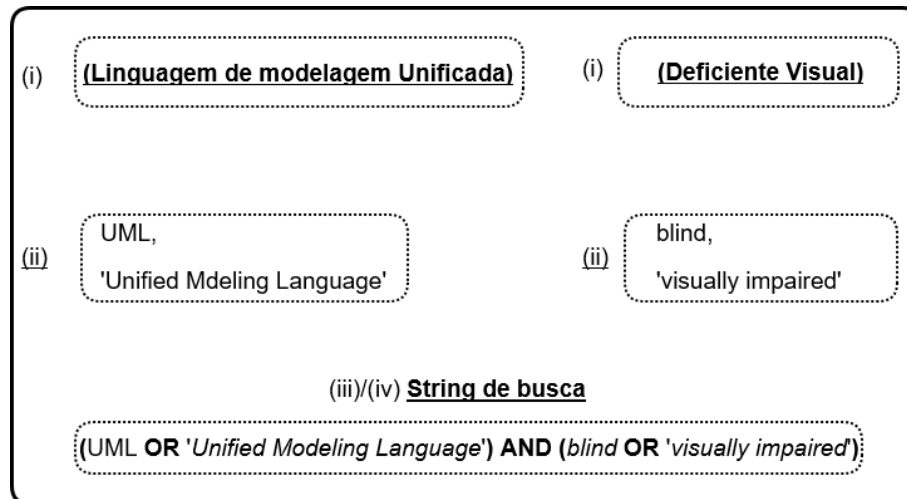


Figura 4 - Definição da *string* de busca

Fonte: Autoria própria

Em seguida foram definidas as fontes de busca, nas quais a *string* construída foi aplicada para execução da busca automática. Para tal, baseou-se nas diretrizes fornecidas por Brereton et al. (2007) e na mentoria da professora orientadora. Na Tabela 1 estão listadas as fontes de busca utilizadas.

| Tabela 1 - Fontes de busca automática |  |
|---------------------------------------|--|
| Fonte/Base de Busca                   | Endereço Eletrônico  |
| Engineering Village                   | <a href="http://www.engineeringvillage.com">www.engineeringvillage.com</a> |
| Scopus                                | <a href="http://www.scopus.com">www.scopus.com</a>                         |
| IEEE                                  | <a href="http://www.ieeexplore.com">www.ieeexplore.com</a>                 |
| Springer                              | <a href="https://link.springer.com">https://link.springer.com</a>          |
| ACM                                   | <a href="https://dl.acm.org/">https://dl.acm.org/</a>                      |

Fonte: Autoria própria

Adicionalmente foi realizada uma calibragem da *string* de busca. Essa atividade consistiu em validar se o estudo de Grillo e Fortes (2014), definido como controle, retornara ao rodar a *string* de busca. A calibragem foi aplicada na fonte *Engineering Village*, uma vez que o estudo controle é indexado nessa fonte.

### 3.1.2 Critérios para Seleção de Estudos Primários

Após a criação e aplicação da *string* nas fontes de busca, o resultado da pesquisa deve passar por um processo de seleção. Este processo é conduzido, inicialmente, por alguns critérios a serem delineados. Eles conferem segurança ao processo de filtragem, que

consiste na inclusão ou exclusão, dos estudos primários que realmente possam contribuir para a elucidação da QP. No contexto desse MS a seleção seguiu os seguintes critérios:

- **Critérios de Inclusão (CI)**

- **CI<sub>1</sub>**: estudos primários que apresentam alguma metodologia, ferramenta, técnica, notação textual ou abordagem que viabilize o desenvolvimento de diagramas em UML a deficientes visuais.
- **CI<sub>2</sub>**: estudos primários que evidenciam a inclusão de alunos com algum nível de deficiência visual ao ensino da modelagem de sistemas utilizando UML.

- **Critérios de Exclusão (CE)**

- **CE<sub>1</sub>**: estudos primários que não sejam *full paper* ou *short paper* (propostas, pôsteres, tutoriais).
- **CE<sub>2</sub>**: estudos primários com textos incompletos.
- **CE<sub>3</sub>**: estudos primários indisponíveis para *download*.
- **CE<sub>4</sub>**: estudos primários escritos em outra língua que não sejam portuguesa ou inglesa.

### 3.1.3 Extração de Dados

Para a conduzir a extração dos dados dos estudos foram realizados os seguintes passos: *(i)* leitura do título e *abstract*; *(ii)* leitura da introdução e conclusão; e *(iii)* leitura na íntegra.

Após a leitura integral de cada estudo foi criada uma planilha utilizando a ferramenta Planilhas do *Google Drive*<sup>1</sup>, a fim de organizar a extração de dados como: título; ano da publicação; objetivo; autores; país de origem; tipo da publicação; soluções propostas, diagramas contemplados; tipo de avaliação e contexto investigado (academia e mercado). Essa coleta de dados apoiou a construção de gráficos discutidos na Seção 3.3.

A partir da extração do objetivo de cada estudo primário incluído foi possível organizá-los em quatro categorias primárias. São elas:

- **C1 - Ferramentas**: reúne estudos que focam na criação de uma ferramenta, protótipo, sistema, software, ou *framework* para o desenvolvimento de diagramas em UML.

---

<sup>1</sup><https://www.google.com.br/drive/apps.html>.



- **C2 - Notação textual:** contempla estudos que cujo a temática é a criação de uma linguagem de domínio específico (do inglês, *Domain Specific Language (DSL)*, notação textual, ou linguagem específica para o desenvolvimento de diagramas utilizando UML.
- **C3 - Contribuição:** contém estudos cujo objetivo é realizar uma contribuição, avaliação, catalogação ou *guideline* de soluções pesquisadas sobre o estado da arte para o desenvolvimento de diagramas em UML.
- **C4 - Representação física:** aborda estudos cujo foco foi desenvolver representações físicas e sonoras, *haptics*<sup>2</sup>, impressões 3D, versões táteis para o desenvolvimento de diagramas em UML.

Em alguns estudos, após a leitura na íntegra, foi possível associá-los à mais de uma das categorias criadas. As uniões de categorias observadas foram: (i) C1 e C2; (ii) C1, C2 e C3; e (iii) C2 e C4.

### 3.2 Condução do Mapeamento Sistemático

A partir da execução da busca automática foi reunido um total de 13 estudos primários que atendiam à QP nesse MS, sendo que em cada fonte de busca retornaram: 70 estudos na *Engineering Village*; 47 na *Scopus*; 24 na *IEEE*; 80 na *Springer*; e 443 na *ACM*.

Na Figura 5 é ilustrada a visão geral da condução deste MS. A partir da execução da *string* de busca nas fontes um total de 664 estudos foram retornados. A partir destes foram excluídos 22 estudos repetidos. Dos estudos restantes (642) foram lidos títulos e *abstract*, aliado à aplicação dos critérios de inclusão e exclusão, sendo incluídos somente 22 estudos. No passo 2 foram lidas a introdução, a conclusão e aplicados os critérios de inclusão e exclusão aos 22 estudos, restando 15 estudos incluídos. Por fim, no passo 3 dos 15 estudos lidos na íntegra, 13 estudos foram incluídos.

### 3.3 Análise e Síntese dos Estudos Primários

A partir da execução da busca automática nesse MS foram identificados 13 estudos primários abordando soluções que tinham como objetivo auxiliar alunos deficientes visuais

---

<sup>2</sup>Do grego, *haptikós*; referente ao toque. *Haptics* são tecnologias relativas ao trato da percepção por meio daquilo que é tátil.

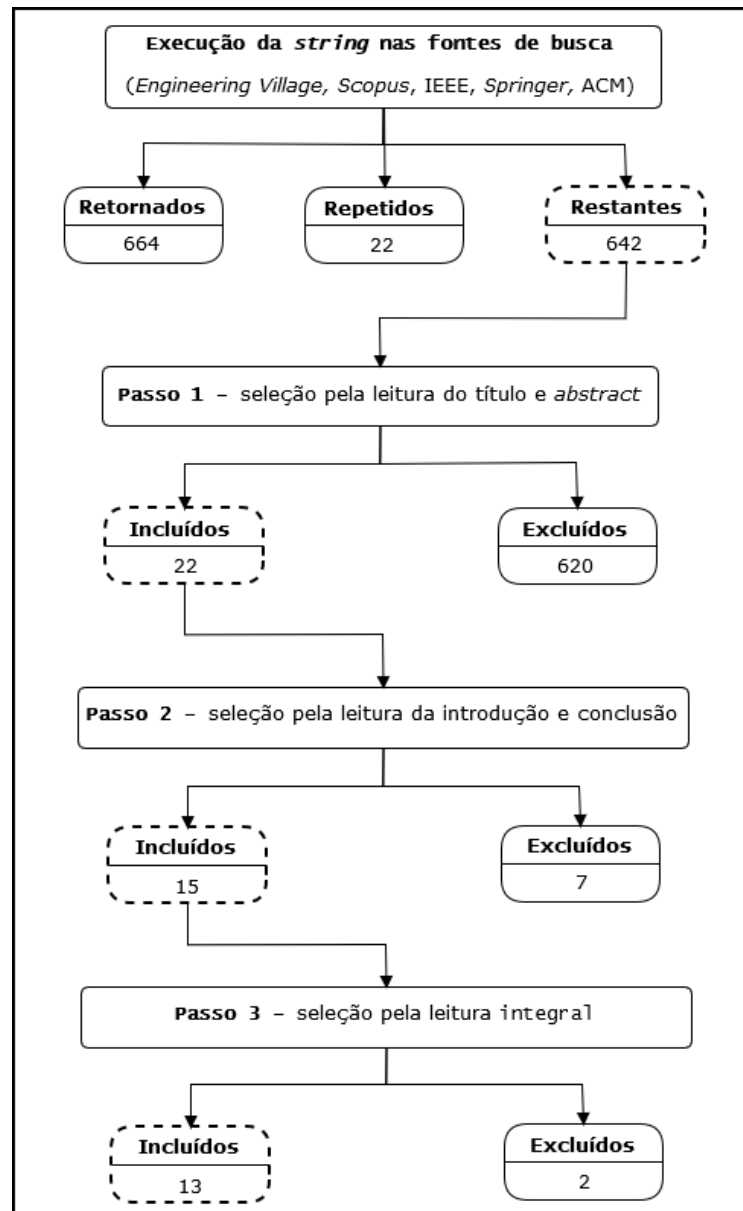
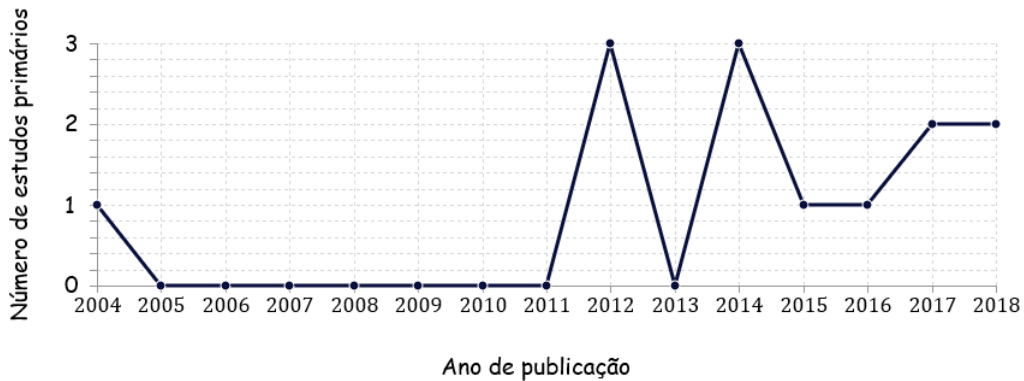


Figura 5 - Processo de seleção dos estudos primários

Fonte: Autoria própria

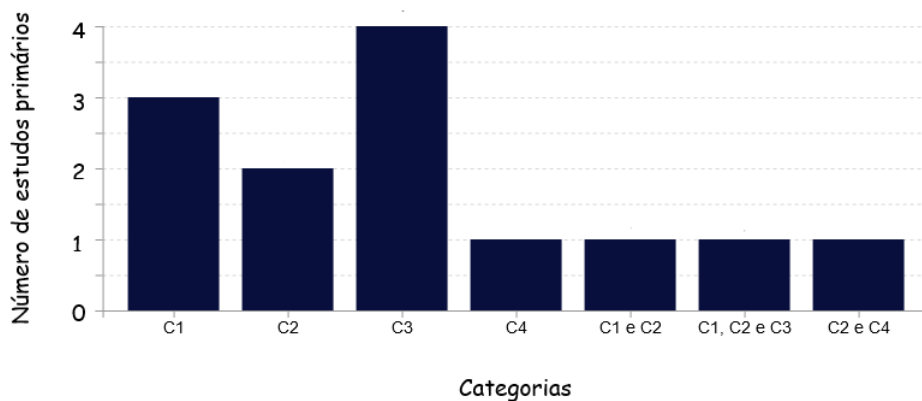
na modelagem de software utilizando UML tanto no âmbito educacional quanto profissional. Foi possível analisar que este tema começou a ser investigado em 2004, passando por um longo hiato até 2011, conforme pode ser observado na Figura 6.

No que tange às categorias criadas na Subseção 3.1.3, foi visto que 15% (2/13) dos estudos primários incluídos contemplam a categoria C2 (Notação textual). Uma DSL, por exemplo, pode ser uma alternativa a ser explorada quando se trata de acessibilidade e modelagem de sistemas com diagramas em UML. Na Figura 7 observa-se as ocorrências de estudos por categorias.



**Figura 6 - Número de publicações por ano**

Fonte: Autoria própria



**Figura 7 - Ocorrência de estudos primários por categoria**

Fonte: Autoria própria

As publicações científicas são divulgadas em diferentes formas, como: congressos, *workshops*, simpósios, periódico, conferências, entre outros. Dentre os tipos citados, neste MS foram identificados quatro tipos de estudos de publicação: conferência, simpósio, *workshop*, e periódico. Sendo que conferência compreende 60% (8/13), enquanto *periódico* e *workshop* 8% (1/13) dos estudos incluídos, como pode ser observado na Figura 8.

Das cinco fontes de busca, a *Engineering Village (EV)* contve a maioria dos estudos incluídos neste MS, contabilizando 70% (9/13). Nas *Springer* e *ACM* foram coletados dois estudos em cada. Quanto às demais bases (*IEEE* e *Scopus*) nenhum estudo foi incluído durante o processo de condução. A Tabela 2 exibe uma visão geral desses estudos primários, contendo a classificação dos estudos de acordo com suas categorias; os



Figura 8 - Quantidade de estudos primários por tipo de publicação

Fonte: Autoria própria

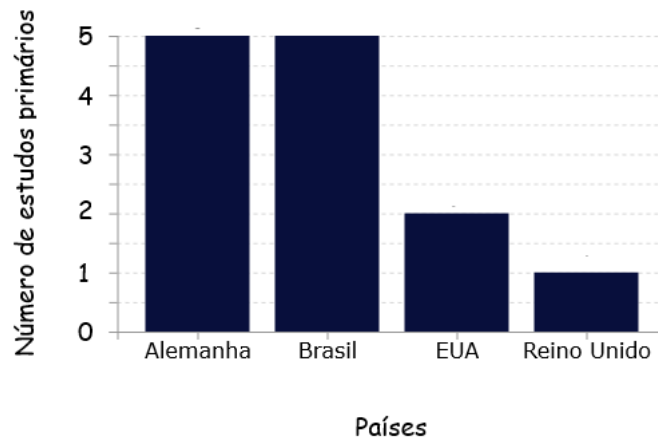
estudos, os autores conduziram alguma forma de avaliação das soluções propostas; e a fonte/base de busca da qual foram selecionados.

Tabela 2 - Visão geral dos estudos primários incluídos

| Categorias  | Estudos Primários           | Validação       | Fontes          |
|-------------|-----------------------------|-----------------|-----------------|
| C1          | Doherty e Cheng (2015)      | Não especificou | <i>EV</i>       |
|             | King et al. (2004)          | Estudo de Caso  | <i>EV</i>       |
|             | Pansanato et al. (2012)     | Estudo de Caso  | <i>ACM</i>      |
| C2          | Loitsch et al. (2018)       | <i>Survey</i>   | <i>EV</i>       |
|             | Vieritz et al. (2012)       | <i>Survey</i>   | <i>ACM</i>      |
| C3          | Luque et al. (2014)         | Estudo de Caso  | <i>EV</i>       |
|             | Petrausch et al. (2016)     | <i>Survey</i>   | <i>EV</i>       |
|             | Seifermann e Groenda (2016) | Não especificou | <i>Springer</i> |
|             | Luque et al. (2018)         | <i>Survey</i>   | <i>Springer</i> |
| C4          | Owen et al. (2014)          | Estudo de Caso  | <i>EV</i>       |
| C1 e C2     | Grillo e Fortes (2014)      | Estudo de Caso  | <i>EV</i>       |
| C1, C2 e C3 | Luque et al. (2017)         | Estudo de Caso  | <i>EV</i>       |
| C2 e C4     | Müller (2012)               | Não especificou | <i>EV</i>       |

Fonte: Autoria própria

Abordando o cenário internacional foram identificados três países, além do Brasil, explorando soluções que visam auxiliar estudantes deficientes visuais na modelagem de software utilizando UML. São eles: Alemanha, EUA e Reino Unido. A classificação dos estudos levou em conta o país de origem do primeiro autor. A Figura 9 a seguir mostra a quantidade de estudos incluídos por país.



**Figura 9 - Número de estudos incluídos por país**

**Fonte: Autoria própria**

Alemanha com cinco estudos ((MÜLLER, 2012), (LOITSCH et al., 2018), (PETRAUSCH et al., 2016), (SEIFERMANN; GROENDA, 2016), (VIERITZ et al., 2012)) e Brasil também com cinco estudos ((PANSANATO et al., 2012), (LUQUE et al., 2014), (LUQUE et al., 2017), (GRILLO; FORTES, 2014), (LUQUE et al., 2018)) lideram o ranking de publicações incluídas nesse MS.

Sobre a validação realizada nos estudos primários incluídos destaca-se que o estudo de caso foi conduzido em 46,2% (6/13) dos estudos, seguido pelo *Survey* com 30,8% (4/13), e por fim tiveram estudos que não especificaram como foi realizada a validação 23,1% (3/13), conforme pode ser visto na Figura 10.

Destaca-se que os estudos de casos realizados contaram com um público heterogêneo, tais como: estudantes; profissionais (com deficiência ou não), bem como professores e pesquisadores da área de TI. Estudos categorizados em C1, como por exemplo de Pansanato et al. (2012); Grillo e Fortes (2014); Luque et al. (2017), aplicaram tarefas de modelagem e questionários para avaliar a solução proposta.

Outro aspecto relevante percebido neste MS é o fato de os estudos primários incluídos concentrarem esforços e soluções para auxiliar o desenvolvimento de diagrama de Classe. A Figura 11 ilustra a quantidade de estudos, dentre os 13, que abordam cada tipo de diagrama identificado.

Como pode ser observado, pelo menos 9 dos 13 estudos primários investigam meios para auxiliar o desenvolvimento de diagrama de classes. Enquanto os diagramas

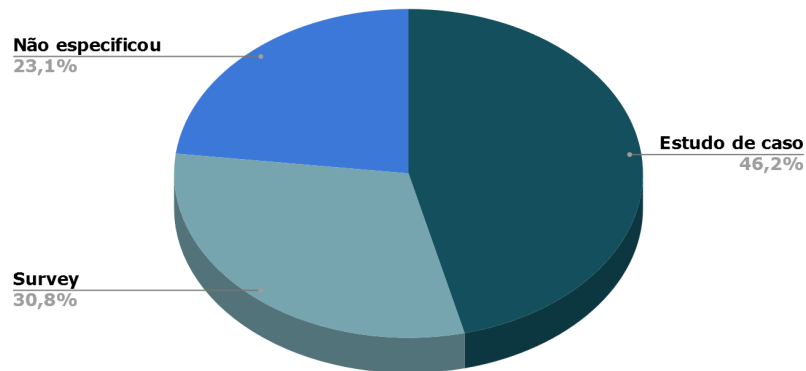


Figura 10 - Quantitativo do tipo de validação realizada pelos estudos primários

Fonte: Autoria própria

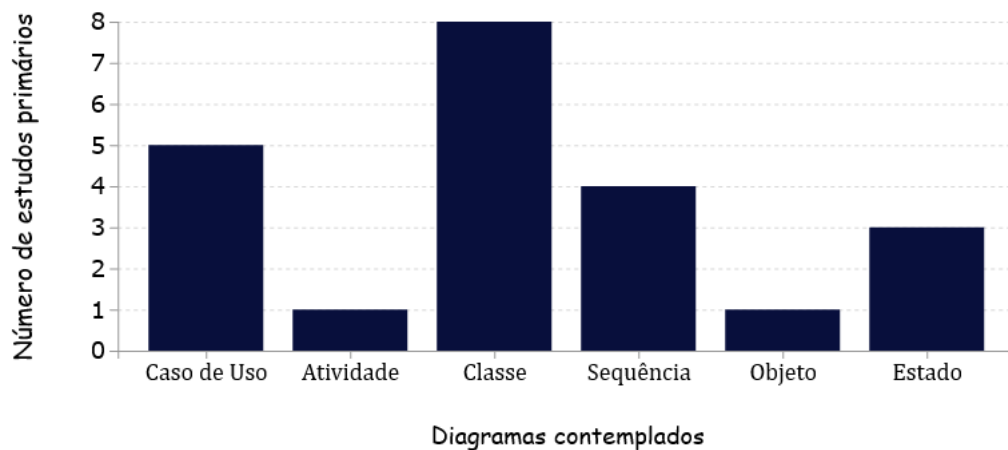


Figura 11 - Números de estudos por diagramas contemplados

Fonte: Autoria própria

de atividade e objeto são tratados somente por Vieritz et al. (2012) e Owen et al. (2014), respectivamente, conforme é apresentado na Tabela 3.

Acredita-se que o foco no diagrama de classes é justificado por este carregar informações de alto e baixo nível utilizadas no desenvolvimento de software; além de ser amplamente ministrado no ensino de disciplinas voltadas para Orientação a Objetos (OO).

Também, salienta-se que o estudo proposto por King et al. (2004) investigou sozinho quatro diagramas diferentes (caso de uso, classe, sequência e estado), sendo o

**Tabela 3 - Visão geral dos estudos primários referente aos diagramas contemplados e contexto**

| <b>Categorias</b> | <b>Estudos Primários</b>    | <b>Diagramas</b>                       | <b>Contexto</b>     |
|-------------------|-----------------------------|--|---------------------|
| C1                | Doherty e Cheng (2015)      | classe e sequência                     | academia            |
|                   | King et al. (2004)          | caso de uso, classe, sequência, estado | academia, indústria |
|                   | Pansanato et al. (2012)     | caso de uso                            | indústria           |
| C2                | Loitsch et al. (2018)       | caso de uso, classe, sequência         | academia, indústria |
|                   | Vieritz et al. (2012)       | classe, atividade                      | -                   |
| C3                | Luque et al. (2014)         | caso de uso, classe                    | indústria           |
|                   | Petrausch et al. (2016)     | classe, sequência                      | academia            |
|                   | Seifermann e Groenda (2016) | -                                      | -                   |
|                   | Luque et al. (2018)         | -                                      | -                   |
| C4                | Owen et al. (2014)          | classe, objeto, estado                 | academia            |
| C1 e C2           | Grillo e Fortes (2014)      | classe                                 | academia            |
| C1, C2 e C3       | Luque et al. (2017)         | caso de uso                            | academia            |
| C2 e C4           | Müller (2012)               | classe, sequência, estado              | academia            |

**Fonte: Autoria própria**

estudo com a maior cobertura observada. Enquanto que os estudos de Pansanato et al. (2012) e Luque et al. (2017) abordaram somente o diagrama de caso de Uso.

A QP proposta neste MS orientou o processo de busca por estudos primários a fim de identificar soluções desenvolvidas no ambiente educacional e/ou profissional que incluíssem indivíduos com baixa visão ou cegueira total às práticas de modelagem de sistemas utilizando diagramas UML.

A partir da busca e análise conduzidas, foi possível concluir que os estudos evidenciam baixa participação do público deficiente visual em atividades de modelagem de sistemas utilizando diagramas em UML. Quando participam, normalmente colaboram apenas com a confecção de documentos de requisitos. Alguns estudos relataram sobre deficientes visuais possuem, devido à compensação sensorial, habilidade lógica superior em relação aos colegas de turma e/ou equipe de desenvolvimento. Por isso, comumente assumem cargo de desenvolvedor nos projetos, já que a utilização de uma T.A como leitores de tela viabiliza essa função.

É importante ressaltar o desafio com que professores lidam em sala de aula ao ministrar conteúdos dependentes da visão, dada sua natureza gráfica, para alunos visualmente debilitados. Esse desafio é acentuado devido à falta de capacitação continuada aos educadores, e investimentos para criação de núcleos de ensino especializados no espaço

educacional, básico ou superior. Portanto, é necessário viabilizar soluções que visam mitigar desafios e barreiras existentes para professores, alunos e profissionais deficientes visuais no acesso ao ensino ou mercado de trabalho da Engenharia de Software e campos afins.

Com o fim de atenuar essas barreiras e desafios observados, foram identificados estudos como (DOHERTY; CHENG, 2015), (KING et al., 2004) e (PANSANATO et al., 2012), pertencentes à categoria C1, propondo ferramentas que viabilizam a modelagem de sistemas por meio diagramas em UML.

Doherty e Cheng (2015) abordaram a necessidade do deficiente visual compreender a disposição espacial dos nós de um diagrama por meio do tato. Para isso foi proposto no estudo a criação de uma ferramenta chamada PRISCA capaz de gerar versões 3D de diagramas de classe e sequência, utilizando como entrada uma versão 2D oriunda de outra ferramenta de modelagem. Posteriormente a saída gerada pode ser fornecida impressa por meio de uma impressora 3D ao estudante. A técnica utilizada na interpretação e conversão dos diagramas consistiu em transformar a entrada 2D num formato XML (do inglês, *eXtensible Markup Language*) que é convertida para uma saída 3D utilizando a biblioteca da ferramenta *Visual Paradigm*.

King et al. (2004) relataram um projeto chamado *Technical Diagram Understanding for the Blind (TeDUB)* que busca tornar diagramas de classe e estado em UML acessíveis para deficientes visuais. A estratégia consistiu na conversão de cada nó do diagrama de entrada para um modelo XML conceitual. A ferramenta interpreta o arquivo XML e então exibe informações hierarquicamente estruturadas numa interface, estilo Windows, que o usuário navega utilizando o teclado com o auxílio de leitores de tela.

Pansanato et al. (2012) seguiram o contexto da acessibilidade para usuários cegos desenvolvendo um protótipo de interação com diagramas de caso de uso. O projeto denominado de *Diagram For All (D4ALL)*, foi uma iniciativa que buscou permitir o acesso e a manipulação de diagramas por estudantes de ES com algum nível de deficiência visual no Campus de Cornélio Procópio da Universidade Tecnológica Federal do Paraná (UTFPR). A ferramenta desenvolvida permite o acesso da representação do diagrama de caso de uso por meio de uma tabela que o usuário percorre auxiliado por leitores de tela. Também provê um editor do diagrama convertido para que o usuário possa fazer as devidas alterações navegando entre os elementos com auxílio do teclado. O diagrama é percorrido da esquerda para direita e de cima para baixo no plano cartesiano.

Outros estudos como (LOITSCH et al., 2018) e (VIERITZ et al., 2012) também



buscaram propor soluções para auxiliar o desenvolvimento de diagramas UML, as quais pertencem à categoria C2. Nesta categoria são identificados estudos cujo foco foi o desenvolvimento de uma notação textual para modelagem de sistemas.

Loitsch et al. (2018) contemplaram três diagramas em seu estudo: caso de uso, classe, e sequência. A DSL desenvolvida foi avaliada com 14 participantes. Nessa população, sete apresentavam algum nível de deficiência visual (baixa visão ou cegueira total); e sete não apresentavam nenhum tipo de deficiência. Além disso, um aspecto a ser ressaltado é que no estudo foi observada a estratégia de se criar prefixos identificadores genéricos para os três tipos de diagramas citados. Tais prefixos garantiram uma padronização da DSL e rápida identificação do tipo de diagrama que o usuário está manipulando ou acessando.

No estudo de Vieritz et al. (2012) foram contemplados dois diagramas distintos: atividade e classe. Nesse estudo foi desenvolvida uma notação textual chamada *Human - usable Textual Notation (HUNT)* que utiliza o princípio de metamodelo proposto por OMG para modelar todo um diagrama.

Os estudos de Luque et al. (2014), Petrausch et al. (2016), Seifermann e Groenda (2016) e Luque et al. (2018) pertencem à categoria C3, pois relataram contribuições científicas a partir de pesquisas e avaliações para auxiliar na modelagem de sistemas utilizando UML.

Luque et al. (2014) realizaram uma revisão da literatura a fim de avaliar a disponibilidade e acessibilidade das soluções que tratam sobre a inclusão de pessoas com deficiência visual em atividades focadas na criação e edição de diagramas em UML na área da educação e desenvolvimento de software. Ao final do estudo os autores concluíram que as ferramentas disponíveis não proporcionam acessibilidade, nem são apropriadas para uso na indústria de software.

Petrausch et al. (2016) relataram a ausência de métricas que sirvam como base para avaliação da acessibilidade e compreensão no desenvolvimento de notações textuais para modelagem utilizando UML. Por isso, foi conduzida uma entrevista e aplicação avaliativa de questionário sobre quatro notações textuais com seis participantes, sendo dois cientistas da computação cegos, dois pesquisadores da área de T.A, e dois pesquisadores da computação. O resultado obtido consistiu num guia cobrindo cinco categorias avaliadas: usabilidade; acessibilidade; concepção semântica; construção sintática; e interação entre notação - ferramenta. O guia foi direcionado ao diagrama de classe, mas pode ser aplicado a outros diagramas.

Seifermann e Groenda (2016) conduziram uma revisão bibliográfica a fim de levantar as notações textuais para modelagem UML disponíveis, e avaliar a aplicabilidade em times de desenvolvimento de software. Devido a falta de padronização das 21 notações identificadas, os autores perceberam que essa solução é pouco usada. O diferencial do estudo é fornecer estimativas quanto à cobertura de diagramas UML, experiência do usuário na edição, e aplicabilidade nos times de desenvolvimento. Ao final do estudo, os autores concluíram que existe alta fragmentação da cobertura de diagramas UML. A seleção de uma notação por cenário de uso específico demanda tempo e análise do *trade-off* para a projeção do software.

Luque et al. (2018) realizaram a condução de uma pesquisa das práticas educacionais e percepções de alunos com deficiência visual no processo de inclusão dos mesmos ao sistema de ensino da computação e afins. A partir do *survey* realizado foi possível fazer um levantamento da ocorrência do público com baixa visão ou cegueira total que busca participar das esferas de ensino no Brasil. Além disso, foi possível averiguar as estratégias tomadas pelos educadores ao se depararem com o desafio de ensinar conteúdos gráficos a estudantes com deficiência visual, como diagramas utilizando UML. No trabalho foi discutido o resultado dos questionários aplicados aos 56 professores e aos 19 alunos atuantes nos cursos de TI. Os autores apontam a relevância do estudo para o planejamento estratégico de ações inclusivas no ensino de alunos visualmente impedidos. E concluem o trabalho alegando que, apesar dos esforços até o momento, ainda é necessário mais desenvolvimento tecnológico que promova um cenário inclusivo.

O estudo de Owen et al. (2014), pertencente à categoria C4, evidencia as estratégias tomadas por um professor no ensino da disciplina Programação Orientada a Objetos (POO) a partir do relato experimental com uma aluna cega no curso Ciência da Computação. Foi observado pelo autor uma elevada limitação das ferramentas quanto acessibilidade, disponibilidade para download, e usabilidade para modelagem utilizando UML. A solução encontrada foi uma miscelânea de alternativas. Partindo desde a utilização de ferramentas assistivas até a construção de diagramas táteis com auxílio de pinos, elásticos, *post-it* presos a um quadro de boletins que viabilizaram: a manipulação de diagramas de classe e objeto, e a conceitualização do paradigma Orientação a Objetos para a aluna. Ao final do trabalho o professor relata suas impressões sobre o semestre que passou, dizendo que em alguns momentos era necessário improvisar na transmissão de conteúdo, além de realçar a limitação das ferramentas disponíveis.

No estudo de Grillo e Fortes (2014) foram identificadas duas categorias criadas

nesse MS, C1 e C2. Os autores apresentam o estudo de caso realizado para avaliar a efetividade de um protótipo, *AWMo* (do inglês, *Accessible Web Modeling*), que permite a modelagem do diagrama de classe a partir de duas perspectivas: gráfica e textual. A primeira focada no público sem deficiência visual; a segunda desenvolvida para o público com baixa visão ou cegueira total. Ambos podem trabalhar colaborativamente sobre o mesmo modelo, já que a ferramenta é capaz de transformar gráfico em notação textual e vice versa. Para modelagem a partir da perspectiva textual os autores desenvolveram uma DSL para construir o diagrama de classe. Quanto à validação, primeiro foi conduzido um estudo piloto que teve a participação de um usuário com apenas 10% da visão. Em seguida, foi conduzida uma segunda avaliação que teve a participação de um voluntário totalmente cego. Apesar da ferramenta e DSL terem sido desenvolvidas especificamente para o diagrama de classe, a avaliação conduzida revelou boa usabilidade, forneceu acessibilidade, e compreensão da atividade de modelagem.

No estudo de Luque et al. (2017) foram identificadas três categorias: C1, C2 e C3. Os autores desenvolveram uma ferramenta chamada *Model2gether*, uma DSL para gerar o diagrama, e conduziram a avaliação da percepção dos usuários na modelagem de casos de uso para um sistema bibliotecário. O objetivo do estudo foi construir um ambiente colaborativo, onde usuários com e sem deficiência visual pudessem modelar juntos. Após conduzida a validação da ferramenta, realizada a atividade de modelagem, e a avaliação da percepção do usuários, os autores observaram que, entre colaboradores, não foi possível distinguir a participação de deficientes visuais no grupo de trabalho.

Por fim, o estudo de Müller (2012), categorizado em C2 e C4, teve como objetivo propor estratégias para auxiliar o desenvolvimento de diagramas de classe e sequência por meio de análise dos materiais de aulas (slides) utilizados no curso Ciência da Computação em duas universidades diferentes. A partir dessa análise, os autores desenvolveram uma DSL e representações físicas dos diagramas citados. A autora afirma que para uma participação efetiva do deficiente visual em atividade de modelagem este deve: *(i)* ter noção espacial do diagrama com o tato; *(ii)* entender a semântica da representação física; *(iii)* realizar um *design* alternativo dos layouts; *(iv)* comunicar-se com o time de desenvolvimento; *(v)* orientar-se por meio das alterações feitas por outros colaboradores; *(vi)* ser capaz de construir diagramas independentemente; e *(vii)* saber traduzir UML em código.

### 3.4 Ameaças à validade

As principais ameaças à condução de um MS estão relacionadas ao fator humano que pode introduzir viés no processo de seleção dos estudos primários, além de provocar imprecisão na extração dos dados. Neste MS foram identificadas as seguintes ameaças à validade:

- **validade de conclusão:** é referente à relação estatisticamente significativa entre o tratamento e os resultados (WOHLIN et al., 2012). Buscando reduzir a ameaça de seleção dos estudos por apenas um pesquisador, foi realizada análise de 10% dos estudos primários incluídos e excluídos pela professora orientadora, sem interferir na qualidade do MS. No que tange à extração dos dados, devido a omissão de informações importantes em vários estudos, houve a necessidade do pesquisador buscar outras publicações relacionadas a fim de auxiliar na análise dessas informações. Contudo, a fim de assegurar a validade dos estudos, foi consultado um terceiro pesquisador em caso de discordância entre os dois primeiros (aluno e orientadora). Assim garantindo a confiabilidade do MS.
- **validade interna:** refere-se à relação causal entre o tratamento e os resultados (WOHLIN et al., 2012). A ameaça à validade interna é observada quando alguns estudos, devido sua indisponibilidade de acesso ou download, não puderam ser incluídos no processo de condução do MS.
- **validade de construção:** diz respeito à relação entre teoria e aplicação (WOHLIN et al., 2012). A ameaça à validade de construção é identificada quando estudos relevantes para QP são excluídos. Uma solução adotada nesse MS consistiu no desenvolvimento de um protocolo de pesquisa seguindo as diretrizes propostas por Kitchenham et al. (2010).
- **validade externa:** refere-se à generalização de resultados num campo externo ao estudo (WOHLIN et al., 2012). Diz respeito à possibilidade de publicações externas referenciam o presente MS de acordo com o grau de relevância deste ao estudo proposto.

### 3.5 Considerações finais

Ao longo deste capítulo foram apresentados os resultados do MS conduzido, a fim de identificar evidências existentes do ensino de modelagem de sistemas utilizando

UML para alunos com deficiência visual. A partir da leitura e análise realizada dos 13 estudos primários, foi possível inferir limitações sobre as ferramentas, técnicas, abordagens ou notações textuais propostas. Apesar das contribuições identificadas, as mesmas não são capazes de garantir plena inclusão de um estudante deficiente visual em atividades de modelagem utilizando UML. É necessário conhecer mais de uma tecnologia ou estratégias de apoio para que conteúdos sejam transmitidos de maneira acessível e a modelagem seja efetivada.

Destaca-se que um dos diferenciais deste trabalho, em relação aos estudos identificados na literatura, é o desenvolvimento de uma solução que apoie a modelagem de software voltada a partir da especificação de software. Em outras palavras, promover acessibilidade na atividade de elicitação e documentação de requisitos para um sistema. A relevância dessa solução é viabilizar melhorias no ensino-aprendizagem dos alunos deficientes visuais e, bem como a inserção destes no mercado de trabalho.

## 4 FERRAMENTA *BLIND MODELING SOFTWARE (B-MODEL)* E LINGUAGEM *BLIND MODELING LANGUAGE (BML)*

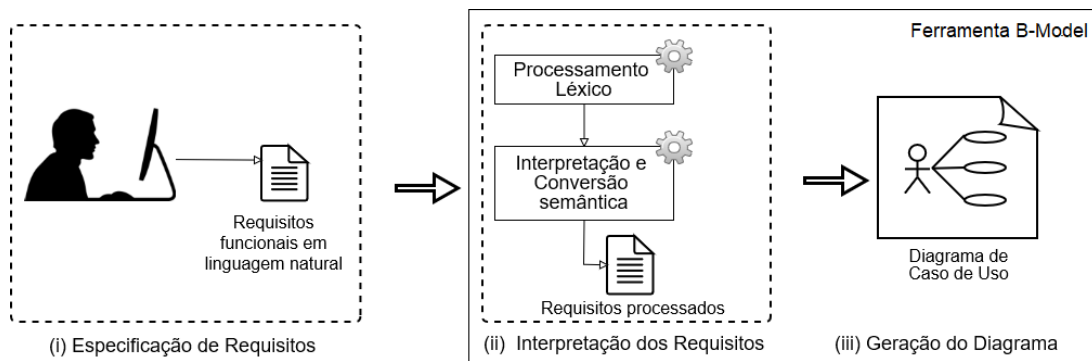
O ensino superior na área de computação oferta disciplinas que focam na modelagem de software utilizando a linguagem UML. Essa linguagem tem como propósito auxiliar a modelagem, por meio de notações gráficas a partir dos requisitos funcionais especificados. Contudo, percebe-se que o aprendizado/participação de estudantes com deficiência visual no processo de modelagem de software é difícil, sobretudo quando há falta de um apoio especializado. Diante deste cenário, nota-se a necessidade de soluções que viabilizem a acessibilidade da modelagem de software utilizando UML.

Neste Capítulo será apresentada a ferramenta B-Model e a linguagem BML. Na Seção 4.1 é apresentada e detalhada as fases da ferramenta, bem como a linguagem BML. Por fim, na Seção 4.2 são apresentadas as considerações finais do capítulo.

### 4.1 Ferramenta B-Model

A ferramenta *B-Model* visa auxiliar estudantes com deficiência visual no aprendizado da modelagem de software utilizando UML. Conforme pode ser visto na Figura 12, a *B-Model* é composta por três fases: (i) Especificação dos Requisitos Funcionais (ERF); (ii) Interpretação dos Requisitos Funcionais (IRF); e (iii) Geração do Diagrama (GRD). No entanto, somente as duas últimas fases são realizadas na ferramenta.

Na primeira fase da (ERF) os requisitos funcionais devem ser especificados em linguagem natural pelo estudante. Na segunda fase (IRF) os requisitos especificados são utilizados como entrada para obter uma especificação padronizada. Por fim, na terceira fase (GRD), a partir dos requisitos funcionais normalizados e processados na segunda fase, é gerado o diagrama de caso de uso correspondente. Cada fase da ferramenta será detalhada nas próximas seções.



**Figura 12 - Visão geral da ferramenta B-Model.**

**Fonte: Autoria própria.**

#### 4.1.1 Fase 1: Especificação de Requisitos Funcionais (ERF)

Os requisitos de software representam as necessidades apresentadas pelos usuários ou clientes, que por sua vez devem ser contempladas pelo sistema a ser desenvolvido (SOMMERVILLE, 2016). Neste contexto, a especificação de requisitos consiste em descrever funcionalidades do sistema que sejam essenciais e também desejáveis a serem implementadas.

Nesta fase o estudante deverá, em linguagem natural, especificar textualmente os requisitos de um cenário disponibilizado pelo professor, por exemplo. Neste contexto, somente serão considerados os requisitos funcionais, uma vez que estes são necessários para a geração do diagrama de casos de uso. As especificações dos requisitos deverão ser escritas em português e salvas em um arquivo de texto, o qual será utilizado como entrada na próxima fase da *B-Model*.

#### 4.1.2 Fase 2: Interpretação dos Requisitos Funcionais (IRF)

Para auxiliar a interpretação dos requisitos foi definida e desenvolvida uma linguagem chamada *Blind Modeling Language (BML)*. A *BML* é uma linguagem que permite a padronização da especificação dos requisitos para gerar o diagrama de caso de uso. A *BML* foi viabilizada por meio da ferramenta *B-Model*.

Nesta fase são realizados o processamento léxico, sintático e semântico dos requisitos especificados com a *BML* utilizando a técnica Processamento de Linguagem Natural (PLN). No processamento léxico da *BML*, suas entradas são classificadas em categorias, conhecidas como *Parts-Of-Speech (POS)* (ALLEN, 1995). As categorias definidas são apresentadas na Tabela 4. Na primeira coluna são apresentadas as categorias

utilizadas, na segunda coluna a descrição dessas categorias, na terceira coluna como as categorias são representadas na gramática, e por fim, na quarta coluna é ilustrado um exemplo.

**Tabela 4 - Categorias definidas para BML com base na POS.**

| <b>Categoria</b>   | <b>Descrição</b>  | <b>Representação na Gramática</b> | <b>Exemplo</b>                                  |
|--------------------|---|-----------------------------------|---|
| <i>Determiners</i> | Representam as palavras que determinam o significado de um nome. Para a <i>BML</i> foram considerados os artigos definidos, os quais devem ser especificados no singular e, maiúsculo ou minúsculo. | DET                               | O, o, A, a                                      |
| <i>Noun</i>        | Representa o sujeito que realiza a ação e deve ser escrito no singular e minúsculo.   | ATOR                              | usuário, sistema                                |
| <i>Verbs</i>       | Representa a ação que será executada pelo usuário ou pelo sistema. Os verbos devem ser especificados no infinitivo.   | VERBO                             | realizar, vender, emitir                        |
| <i>Objeto</i>      | Representam os objetos sucedidos após o verbo. São substantivos e podem ser especificados no singular e/ou no plural.   | OBJ                               | realizar <b>login</b> , consultar <b>festas</b> |
| <i>Aux</i>         | Visa auxiliar na definição do caso de uso. Para isso, são utilizados as palavras “deve” e “devem” juntamente com o verbo.   | AUX                               | <b>deve</b> realizar                            |

**Fonte: Autoria própria.**

Além das categorias de *POS* apresentadas na Tabela 4, também foram definidas categorias com base no domínio de diagrama de caso de uso. Na Tabela 5 são apresentadas as categorias utilizadas na definição da linguagem *BML*. Adicionalmente, a teoria da Estrutura de Constituintes, proposta por Chomsky (CHOMSKY, 1956), foi adicionada no desenvolvimento da *BML*. Esta teoria assume que as sentenças podem ser divididas em *Noun Phrase (NP)* e em *Verb Phrase (VP)*. As frases podem ser divididas em constituintes que podem ser uma palavra ou uma sentença (grupo de palavras).



A teoria de Chomsky foi aplicada devido utilizar a estrutura de sentenças como: sujeito, verbo e predicado; e que conseqüentemente, facilita a especificação de requisitos uma vez que é próxima da linguagem natural.

**Tabela 5 - Categorias definidas para BML com base nos conceitos do diagrama de caso de uso.**

| Categorias    | Descrição  | Representação na Gramática | Exemplo  |
|---------------|--|----------------------------|--|
| Caso de uso   | Representa as funcionalidades do sistema.  | VERBO OBJ                  | realizar login.  |
| Associação    | Refere-se ao relacionamento de associação entre o usuário e um caso de uso.  | -                          | O usuário deve realizar cadastro.                                      |
| Generalização | <b>Generalização de Ator</b> é representada pela conjunção “e” que permite a ligação entre dois atores. O último ator herda as ações do primeiro ator mencionado.      | GENA                       | O atendente e gerente  |
|               | <b>Generalização de casos de uso</b> é representado pela sentença a partir do símbolo “;”  | GENC                       | O usuário deve realizar pagamento: <b>pagar cartão: pagar dinheiro</b> |
| Inclusão      | Representa o relacionamento conhecido como <i>include</i> e indica obrigatoriedade. O <i>include</i> é representado pela sentença a partir do símbolo “;”              | INC                        | O usuário deve realizar cadastro, <b>informar e-mail</b>               |
| Extensão      | Representa o relacionamento conhecido como <i>extend</i> e é utilizado para descrever cenários opcionais. O <i>extend</i> é representado pela sentença a partir do “;” | EXT                        | O usuário deve realizar saque; <b>imprimir extrato</b>                 |

**Fonte: Autoria própria.**

Neste contexto, para a linguagem *BML* foi desenvolvida uma gramática que consiste de frases que são estruturadas de acordo com as categorias e regras apresentadas nas Tabelas 4 e 5. A gramática da *BML* foi definida como uma Gramática Livre de Contexto (do inglês, *Context Free Grammar-CFG*), representada na notação de *Extended Backus-Naur Form (EBNF)*, conforme pode ser visto na Figura 13. As palavras em maiúsculo denotam símbolos terminais que correspondem às categorias léxicas apresentadas nas Tabelas 4 e 5.

|                             |   |   |
|-----------------------------|---|---|
| <b>DriagramaCasoUso</b>     | → | ((Associacao   Generalizacao?   Inclusao?   Extensao?)+ |
| <b>NounPhrase</b>           | → | DET Noun  |
| <b>Noun</b>                 | → | ATOR  |
| <b>VerbPhrase</b>           | → | AUX CasoUso   |
| <b>CasoUso</b>              | → | VERBO OBJ   |
| <b>Associacao</b>           | → | NounPhrase VerbPhrase                                   |
| <b>Generalizacao</b>        | → | ((GeneralizacaoAtor?   GeneralizacaoCasoUso?)+)         |
| <b>GeneralizacaoAtor</b>    | → | NounPhrase GENA ATOR VerbPhrase                         |
| <b>GeneralizacaoCasoUso</b> | → | NounPhrase VerbPhrase ((GENC CasoUso)+)                 |
| <b>Inclusao</b>             | → | NounPhrase VerbPhrase INC CasoUso                       |
| <b>Extensao</b>             | → | NounPhrase VerbPhrase EXT CasoUso                       |

Figura 13 - Gramática referente a linguagem BML.

Fonte: Autoria própria.

O símbolo inicial da gramática é o **DiagramaCasoUso**. O **DiagramaCasoUso** é composto por *Associacao* e *Generalizacao*, *Inclusao* e *Extensao* que representam os relacionamentos que podem ocorrer no diagrama de caso de uso. Todos os relacionamentos podem ocorrer mais de uma vez e os relacionamentos de *Generalizacao*, *Inclusao* e *Extensao* podem ser opcionais.

A *Associacao* representa o relacionamento mais simples no diagrama de caso de uso (existe uma comunicação direta entre o ator e o caso de uso). A *Associacao* compreende um *NounPhrase* (um **ATOR** precedido por um **DETER**) e um *VerbPhrase* que inicia com um **AUX** seguido pelo *CasoUso* que consiste de um **VERBO** sucedido de um **OBJT** (objeto).

A *Generalizacao* é dividida entre a *GeneralizacaoAtor* e *GeneralizacaoCasoUso*. A *GeneralizacaoAtor* ocorre quando dois ou mais atores podem se comunicar com o mesmo conjunto de casos de uso. A *GeneralizacaoAtor* compreende um *NounPhrase* seguido por **GENA** (representado pela conjunção “e”) seguido por um **ATOR** e um *VerbPhrase*. Já a *GeneralizacaoCasoUso* ocorre quando o comportamento e significado de um caso de uso geral são herdados por outros casos de uso. A *GeneralizacaoCasoUso* inicia com *NounPhrase*, seguido do *VerbPhrase*, um **GENC** (representado por “:”) e *CasoUso*.

A *Inclusao* indica um relacionamento de obrigatoriedade e compreende *NounPhrase*, seguido do *VerbPhrase*, um **INC** (representado por “;”) e *CasoUso*. Por fim, a *Extensao* inicia com *NounPhrase*, sucedido por um *VerbPhrase*, um **EXT** (representado por “;”) e *CasoUso*. Portanto, a gramática da *BML* consiste em frases estruturadas de acordo com

algumas regras que permitam especificar os requisitos para a geração do diagrama de caso de uso. A Figura 14 apresenta um exemplo de uma especificação de requisitos usando a *BML*.

```
O usuário deve realizar cadastro.
O gerente e o atendente devem contar estoque.
O atendente e deve registrar venda, realizar login.
O atendente deve gerar relatório; realizar impressão.
O atendente deve selecionar pagamento: pagar cartão: pagar dinheiro.
```

**Figura 14 - Exemplo de uma especificação de requisitos utilizando a *BML*.**

**Fonte: Autoria própria.**

#### 4.1.3 Fase 3: Geração do Diagrama (DC)

Nesta fase, o diagrama de caso de uso é gerado a partir do processamento léxico e da conversão semântica dos requisitos funcionais especificados. Ao realizar a especificação dos requisitos, o arquivo texto deve ser salvo no mesmo diretório em que se encontra o código fonte da ferramenta. A ferramenta possui três módulos: o primeiro localiza o arquivo texto, e faz a leitura do arquivo, salvando todo o texto em vetores. O segundo módulo realiza a etiquetagem do texto. E finalmente o terceiro módulo realiza a geração do diagrama no plano cartesiano.

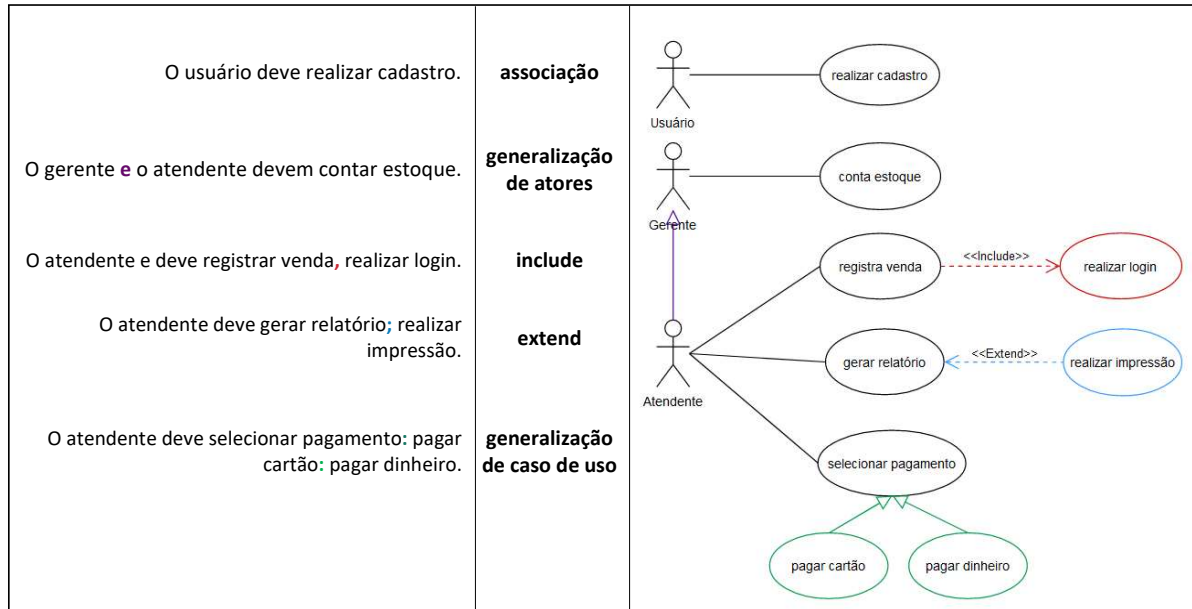
Para o desenvolvimento da ferramenta *B-Model* fora utilizada a linguagem *python* e duas bibliotecas dessa linguagem: (i) *Spacy*<sup>1</sup>; e (ii) *Tkinter*<sup>2</sup>. A biblioteca *Spacy* foi utilizada para realizar a etiquetagem das palavras em cada sentença da especificação de requisitos, buscando por partes do texto que representassem os atores, casos de uso, e os seus respectivos relacionamentos. Apesar da existência de inúmeras bibliotecas similares, a seleção da *Spacy* foi devido permitir o processamento da língua portuguesa.

Após a etiquetagem das palavras, o processamento e a interpretação dos requisitos especificados são realizados utilizando PLN por meio da biblioteca *Spacy*. Cada parte de um requisito especificado é associado a um elemento que irá compor o diagrama de caso de uso, conforme pode ser visto na Figura 15. Por fim, com a biblioteca *Tkinter* foi

<sup>1</sup>*Spacy* documentation - <https://spacy.io/api/doc>

<sup>2</sup>*Tkinter* documentation - <https://docs.python.org/3/library/tk.html>

possível gerar o diagrama, uma vez que permite rotular os atores, os casos de uso e os relacionamentos inserindo rótulos aos respectivos componentes.



**Figura 15 - Exemplo da geração do diagrama de caso de uso a partir da especificação de requisitos utilizando a BML.**

**Fonte: Autoria própria.**

A partir do diagrama acima é possível observar que a ferramenta pode, ainda em fase de testes, entender todos os atores especificados, bem como seus respectivos casos de uso relacionados, além de reconhecer a regra gramatical que expressa a generalização de atores e casos de uso, e também os relacionamentos de *include* e *extend* no texto.

## 4.2 Considerações finais

O presente Capítulo foca na linguagem *BML* com o objetivo de facilitar a especificação de requisitos. Para viabilizar o uso da linguagem *BML*, a ferramenta *B-Model* foi implementada como apoio ferramental. Portanto a linguagem e a ferramenta apresentadas visam auxiliar alunos com deficiência visual no ensino e aprendizado de modelagem de software. Assim, com a utilização da linguagem e da ferramenta os requisitos podem ser especificados de uma forma padronizada e o diagrama de caso de uso gerado automaticamente, independentemente da aplicação. No próximo Capítulo é apresentada a avaliação experimental da linguagem *BML* e ferramenta *B-Model*, bem como os seus respectivos resultados.

## 5 AVALIAÇÃO EXPERIMENTAL E DISCUSSÃO DOS RESULTADOS

Durante a condução deste trabalho, foram desenvolvidos artefatos e coletadas informações a fim de avaliar a linguagem e a ferramenta propostas. Nesta direção, este Capítulo apresenta dois estudos experimentais conduzidos para investigar a utilidade da linguagem *BML*, bem como a eficácia da ferramenta *B-Model* apresentada no Capítulo 4.

Na Seção 5.1 é apresentado o planejamento e a condução do estudo de caso. Na Seção 5.2 é descrito o processo de condução do experimento. Na Seção 5.3 são apresentadas as discussões acerca dos resultados e as ameaças à validade relacionadas ao estudo de caso e ao experimento. Por fim, na Seção 5.4 são apresentadas as considerações finais deste Capítulo.

### 5.1 Estudo de Caso: Utilidade da linguagem *BML*

Este estudo experimental consiste em um estudo de caso exploratório que foi conduzido com objetivo de avaliar a aceitação e o uso da linguagem *BML*. É importante enfatizar que o estudo foi conduzido para revelar possíveis problemas relacionados à aceitação e ao uso da *BML*. Para a condução deste estudo de caso foram utilizadas as diretrizes propostas por Wholin et al. (WOHLIN et al., 2012).

#### 5.1.1 Planejamento

O estudo de caso foi realizado com três discentes deficientes visuais de graduação de três universidades diferentes no período de agosto a dezembro de 2019. O grupo de discentes foi constituído por uma mulher e dois homens. Estes discentes serão identificados ao longo do trabalho como discentes “A”, “B” e “C”.

O discente “A” é do curso de Bacharelado em Engenharia de Software, o discente “B” do curso de Ciência de Dados - Tecnólogo em Sistemas para Internet, e o “C” Bacharelado em Ciência da Computação. O estudo de caso foi conduzido com a discente “A” na disciplina

de Análise Orientada a Objeto e com os demais na disciplina de ES. É importante destacar que algumas melhorias na *BML* foram realizadas durante o período da disciplina a partir dos *feedbacks* dos estudantes.

A avaliação da *BML* foi realizada utilizando a métrica Aceitação e Uso com base no Modelo de Aceitação de Tecnologia (do inglês, *Technology Acceptance Model* (TAM) (DAVIS; BAGOZZI; WARSHAW, 1989). O TAM baseia-se no pressuposto de que a intenção de utilizar um dispositivo tecnológico é determinada pela percepção do usuário sobre sua utilidade e sua facilidade de uso. Neste contexto, foram utilizadas três variáveis que estão compreendidas no questionário deste trabalho:

- **Facilidade de Uso Percebida:** consiste na percepção do usuário quanto a linguagem *BML* em termos de facilidade de aprendizado e de uso.
- **Utilidade Percebida:** consiste no grau em que o usuário acredita que o uso da linguagem *BML* pode melhorar o seu desempenho acadêmico;
- **Intenção de Uso:** consiste na intenção do usuário usar novamente a linguagem *BML*.

De acordo com as três variáveis foram definidas 11 perguntas, as quais foram avaliadas utilizando a escala *Likert* que consiste em cinco níveis, sendo o nível mais alto representado por “Extremamente” e o nível mais baixo “Nem um pouco” (LIKERT, 1932). Na Tabela 6 são apresentadas as variáveis e as suas respectivas perguntas.

### 5.1.2 Condução

Após a definição e o planejamento do estudo de caso, sua fase de condução foi realizada em duas etapas:

1. **Especificação de requisitos usando a *BML*:** nesta etapa foi realizada a especificação dos requisitos de quatro cenários diferentes ( $C = c_1, c_2, c_3$  e  $c_4$ ) utilizando as regras previamente estabelecidas na *BML* e a gramática apresentada na Figura 13. Além disso, esta etapa foi precedida por duas atividades:

- 1.1 **Explicação da *BML*:** os docentes responsáveis pelas disciplinas explicaram a linguagem para os e estudantes e disponibilizaram um material com explicações e exemplos das regras da *BML*.

Tabela 6 - Variáveis do modelo TAM utilizadas para avaliar a *BML*.

| Variáveis                   | Perguntas  |
|-----------------------------|--|
| Facilidade de Uso Percebida | 1. A sua interação com a linguagem <i>BML</i> foi clara e compreensível?<br>2. Interagir com a linguagem <i>BML</i> exigiu muito esforço mental?<br>3. Você considera a linguagem <i>BML</i> fácil de usar?<br>4. Você considera a linguagem <i>BML</i> fácil de ser aprendida?  |
| Utilidade Percebida         | 5. A linguagem <i>BML</i> melhorou o seu desempenho?<br>6. A linguagem <i>BML</i> permitiu aumentar a sua produtividade?<br>7. A linguagem <i>BML</i> foi útil para o aprendizado?<br>8. A linguagem <i>BML</i> lhe permitiu realizar trabalhos em grupo?<br>9. A linguagem <i>BML</i> pode lhe ajudar no mercado de trabalho? |
| Intenção de uso             | 10. Você pretende usar novamente a linguagem <i>BML</i> em outras disciplinas?<br>11. Você recomendaria a linguagem <i>BML</i> para outras pessoas?  |

1.2 **Disponibilização dos cenários:** após a explicação realizada pelos docentes, a descrição dos cenários foi disponibilizada aos estudantes, conforme apresentado na Tabela 7. Os requisitos de cada cenário foram especificados em um arquivo de texto.

2. **Análise da aceitação e uso da *BML*:** a partir dos requisitos especificados, foi aplicado um questionário para coletar a percepção dos estudantes a respeito da aceitação e uso da *BML*. O questionário foi elaborado com 11 questões em um arquivo de texto, as quais foram agrupadas de acordo com as variáveis apresentadas na Tabela 6.

## 5.2 Experimento: Eficácia da ferramenta *B-Model*

Nessa seção será detalhado o experimento conduzido para avaliar a eficácia da ferramenta proposta na geração do diagrama de caso de uso a partir da especificação de requisitos utilizando a linguagem *BML*.

**Tabela 7 - Descrição dos quatro cenários utilizados para avaliar a *BML* e a *B-Model*.**

**c<sub>1</sub>. Sistema de vendas de produtos eletrônicos:** uma loja possui notebooks e celulares para venda. A loja também possui um gerente cuja função administrar o estoque. A loja possui um atendente cuja função é atender os clientes, registrar a venda dos produtos para os clientes, registrar pagamento que pode ser no boleto, dinheiro ou cartão e também administra estoque assim como o gerente. Para registrar as compras, o atendente deve estar logado. Além disso, após a compra o sistema gera a nota fiscal do produto que pode ou não ser impressa.

**c<sub>2</sub>. Sistema de gerenciamento de festas:** um sistema online deve gerenciar a compra e venda de ingressos de festa. Neste sistema tanto o administrador como o gerente podem gerar relatório, gerenciar as festas e consultar festas. Além disso, o gerente pode adicionar patrocínios. O usuário do sistema realiza cadastro e compra ingresso. Para realizar a compra o usuário deve estar logado. Após a compra do ingresso, a Administradora do cartão autentica a venda. Ao final, o sistema gerar o ingresso, ele pode ou não ser enviado por email.

**c<sub>3</sub>. Sistema de locação de veículos:** um sistema online deve gerenciar a locação de veículos de diferentes marcas e modelos. O cliente realiza cadastro, gerencia seu perfil e realiza locação do automóvel. Para realizar a locação o cliente deve estar logado. Tanto o gerente como o funcionário podem gerenciar automóveis e consultar automóveis. O funcionário registra o pagamento, podendo ser no cartão ou dinheiro e registra a devolução do automóvel. Após a devolução, o funcionário pode ou não gerar nota fiscal.

**c<sub>4</sub>. Sistema de gerenciamento de consultas de uma clínica veterinária:** um sistema deve gerenciar o agendamento e a realização de consultas. O cliente por meio do sistema online agenda a consulta e para isso precisa estar logado. Ao chegar com o seu pet na clínica veterinária, a secretária deverá verificar o cadastro do pet. Caso o pet não seja cadastrado a secretária deverá cadastrá-lo. A secretária deverá registrar o pagamento. Este pagamento pode ser em dinheiro ou em cartão. O veterinário deve registrar a consulta realizada. Durante a consulta, o veterinário pode ou não solicitar exames.

### 5.2.1 Definição do Experimento

Para a definição do experimento, o modelo *Goal-Question-Metric (GQM)* proposto por Basili e Weiss (BASILI; WEISS, 1986) foi utilizado para definir os objetivos deste experimento que consiste em:

*“Analisar a ferramenta B-Model com o propósito de avaliar com respeito à eficácia do ponto de vista de pesquisadores no contexto de quatro cenários diferentes.”*

Para alcançar o objetivo, a seguinte Questão de Pesquisa (QP) foi investigada:

**QP<sub>1</sub>: Quão eficaz é a ferramenta *B-Model* para especificar requisitos e gerar o diagrama de caso de uso correspondente?**

Para responder esta QP, a eficácia da abordagem foi mensurada por meio da



relação do diagrama de caso de uso gerado a partir da especificação de requisitos utilizando a linguagem *BML*.

### 5.2.2 Design do Experimento

Quanto ao design do experimento foram analisadas: (i) uma variável independente - especificação dos requisitos; e (ii) uma variável dependente - diagrama de caso de uso correspondente.

Para realizar a avaliação da ferramenta *B-Model* foram utilizados os quatro cenários apresentados na Tabela 7. Apesar dos cenários serem fictícios, os mesmos compreendem os elementos necessários que um estudante deve abstrair para a modelagem de software. O experimento foi realizado por meio de um *notebook* com processador *Intel Core i7 2.4GHz*, memória de 8GB e o sistema operacional *Windows 10*.

### 5.2.3 Condução do Experimento

O procedimento realizado durante a execução do experimento consistiu nas seguintes etapas:

1. **Especificação de requisitos:** as especificações dos requisitos  $R = (r_1; r_2; \dots; r_n)$  para cada cenário ( $c_i$ ) foram realizadas pelos discentes deficientes visuais usando a linguagem *BML*;
2. **Importação da especificação de requisitos:** os requisitos especificados foram adicionados na ferramenta *B-Model*;
3. **Geração do diagrama de caso de uso:** a partir dos requisitos especificados utilizando a linguagem *BML* foi gerado o diagrama de caso de uso correspondente na ferramenta *B-Model*.

## 5.3 Análise e discussão dos resultados

Nesta seção são detalhados os resultados alcançados para a linguagem *BML* e para a ferramenta *B-Model*.

### 5.3.1 Aceitação e uso da linguagem *BML*

A aceitação e o uso da *BML* foram avaliados por meio das respostas dos três discentes em relação a facilidade de uso percebida, utilidade percebida e intenção de uso. Os resultados são apresentados na Tabela 8. Na primeira coluna são apresentadas as categorias relacionadas às métricas Aceitação e Uso do modelo TAM. Na segunda coluna são listadas as 11 perguntas do questionário. Por fim, na terceira coluna são apresentadas as respostas de cada discente (“Disc. A”, “Disc. B”, “Disc. C”)

**Tabela 8 - Resultados da avaliação referente à facilidade de uso percebida, utilidade percebida e intenção de uso da *BML*.**

| Categorias        | Perguntas   | Avaliação |         |         |
|-------------------|---|-----------|---------|---------|
|                   |   | Disc. A   | Disc. B | Disc. C |
| Facilidade de uso | 1. A sua interação com a linguagem BML foi clara e compreensível?       | 4         | 4       | 3       |
|                   | 2. Interagir com a linguagem BML exigiu muito esforço mental?           | 2         | 1       | 2       |
|                   | 3. Você considera a linguagem BML fácil de usar?                        | 5         | 5       | 4       |
|                   | 4. Você considera a linguagem BML fácil de ser aprendida?               | 4         | 4       | 4       |
| Utilidade         | 5. A linguagem BML melhorou seu desempenho?                             | 5         | 5       | 5       |
|                   | 6. A linguagem BML permitiu aumentar sua produtividade?                 | 5         | 5       | 5       |
|                   | 7. A linguagem BML foi útil para o aprendizado?                         | 5         | 5       | 5       |
|                   | 8. A linguagem BML lhe permitiu realizar trabalhos em grupo?            | 5         | 5       | 5       |
|                   | 9. A linguagem BML pode lhe ajudar no mercado de trabalho?              | 5         | 5       | 4       |
| Intenção de uso   | 10. Você pretende usar novamente a linguagem BML em outras disciplinas? | 4         | 4       | 4       |
|                   | 11. Você recomendaria a linguagem BML para outras pessoas?              | 5         | 5       | 5       |

**Fonte: Autoria própria.**

Os resultados apresentados na Tabela 8 sugerem que a *BML* é uma alternativa útil que pode ser utilizada para a especificação de requisitos e para a geração de diagrama

de caso de usos. Um dos principais benefícios da *BML* é prover uma especificação mais próxima da linguagem natural, garantindo assim, uma especificação mais consistente, inequívoca e completa para gerar os diagramas de caso de uso.

### 5.3.2 Eficácia da ferramenta *B-Model*

Nesta seção são apresentados os resultados alcançados em relação à eficácia da ferramenta *B-Model*. Nas Figuras 16 e 17, 18 e 19 são apresentadas as especificações de requisitos e os diagramas de caso de uso gerados para os cenários  $c_1$  (sistema de vendas de produtos eletrônicos),  $c_2$  (sistema de gerenciamento de festa),  $c_3$  (sistema de locação de veículos), e  $c_4$  ( sistema de gerenciamento de consultas de uma clínica veterinária), respectivamente.

Conforme pode ser observado os diagramas gerados contemplam os elementos de caso de uso e correspondem as especificações de requisitos. Na Figura 17, por exemplo, a ferramenta *B-Model* foi capaz de reconhecer todos os atores e suas respectivas relações com seus casos de uso. Além disso, diferenciou corretamente atores com identificadores similares, como é o caso de “Administrador” e “Administradora”; e atribuiu a generalização identificada aos atores que a possuem, considerando a sequência dos mesmos na especificação de requisitos. Portanto, os resultados alcançados indicam que a ferramenta é eficaz, e consequentemente viável para auxiliar estudantes com deficiência visual na modelagem de software.

Ainda é válido destacar a importância da *B-Model* no âmbito real de ensino e aprendizagem, uma vez que estudantes com deficiência visual utilizam ferramentas de edição de texto para inserção do conteúdo e leitores de tela para verificação. Especificamente no contexto de modelagem de software, o processo de ensino e desenvolvimento de diagramas é realizado por meio de uma descrição textual, uma vez que estruturas visuais podem não fazer sentido para estudantes que são deficientes visuais desde o nascimento.

Portanto, não existe problema em realizar a modelagem de software por meio de uma descrição textual, pois é uma prática comum entre os estudantes com essa deficiência.. No entanto, um artefato visual, como diagramas, é importante para inclusão destes estudantes em diferentes âmbitos, tais como:

- **Ensino:** a modelagem de software atualmente é realizada utilizando a linguagem universal UML. Dessa forma, o artefato visual se torna imprescindível para a avaliação do estudante, bem como a sua interação com os demais colegas sem deficiência.

- **Mercado:** assim como no ensino, os diagramas de UML são utilizados como documentos e artefatos para auxiliar no processo de desenvolvimento de software. Portanto, por meio de uma ferramenta de geração de diagramas via linguagem natural é possível que estudantes com deficiência visual tenham as mesmas oportunidades para ingressar em um cargo de engenheiro de software, por exemplo.

### 5.3.3 Ameaças à Validade

Nesta seção são detalhadas as possíveis ameaças à validade que podem afetar os valores e as conclusões dos estudos experimentais conduzidos seguindo as diretrizes de Wohlin (WOHLIN et al., 2012).

As ameaças à validade externa estão relacionadas à generalização dos resultados. A representatividade dos cenários pode ser um problema, pois não existem teorias que garantam que um determinado conjunto de cenários selecionados seja uma amostra representativa para a condução dos estudos empíricos. Com a finalidade de minimizar esse problema, foram utilizados quatro cenários pertencentes a diferentes contextos. No entanto, não é possível afirmar que os resultados podem ser generalizados para qualquer cenário. Neste contexto, replicações futuras são necessárias para corroborar os resultados obtidos.

As ameaças à validade por construção estão preocupadas com a relação entre a teoria e o que é observado. No desenvolvimento da ferramenta proposta bem como na análise dos cenários, possíveis equívocos podem ter sido cometidos. Para mitigar esse risco foram realizados diversos testes das regras a fim de assegurar que durante a condução do experimento a linguagem e a ferramenta tivessem êxito na sua execução.

Por fim as ameaças à validade interna caracterizam o grau de confiabilidade entre os resultados esperados e os resultados obtidos. Todas as variáveis do experimento foram controladas para mitigar possíveis ameaças. Além disso, para ampliar a confiabilidade dos resultados, os dados obtidos foram validados junto às regras definidas para assegurar que os resultados apresentados sejam realmente ortogonais, coerentes e interpretados de forma apropriada.

## 5.4 Considerações Finais

Neste Capítulo foram apresentados dois estudos experimentais conduzidos: um estudo de caso exploratório e um experimento controlado. O estudo de caso foi realizado

para avaliar a aceitação e uso da linguagem *BML* para auxiliar a especificação de requisitos. O experimento controlado foi conduzido para verificar a eficácia da ferramenta *B-Model* na geração do diagrama de caso de uso a partir dos requisitos especificados utilizando a linguagem *BML*.

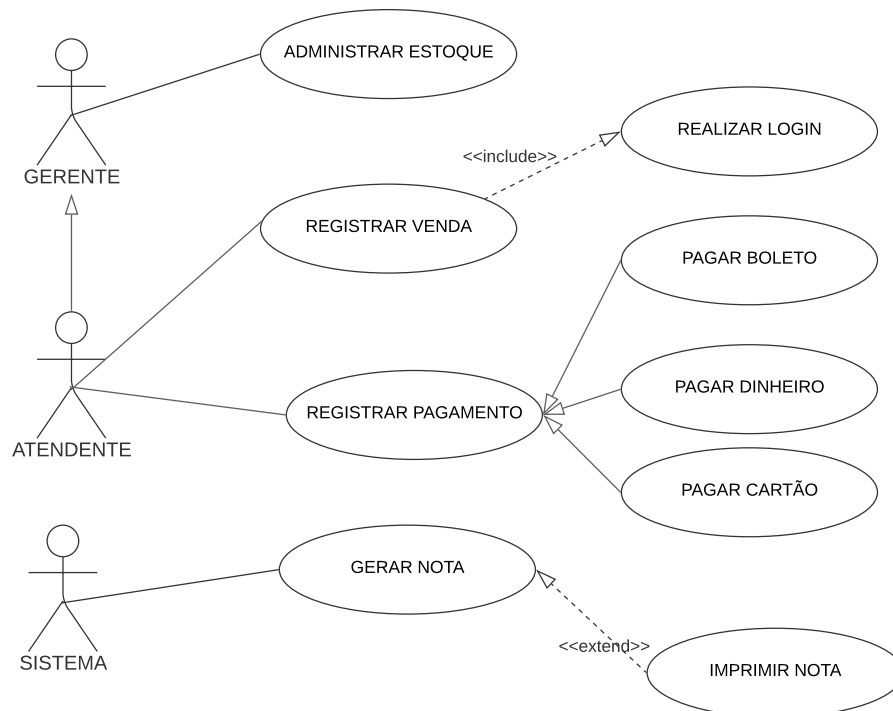
Os resultados do estudo de caso confirmam que a linguagem foi considerada fácil de ser utilizada e para o aprendizado, bem como contribuiu para a melhoria do desempenho acadêmico dos discentes. Portanto, os resultados alcançados indicam a intenção de uso da linguagem *BML* por parte dos discentes. Quanto ao experimento que verificou a eficácia da ferramenta *B-Model*, os resultados indicam que a ferramenta é eficaz, uma vez que gera o diagrama de caso de uso correspondente à especificação de requisitos e, conseqüentemente, auxilia os discentes deficientes visuais no processo de modelagem de software.

[Especificação de requisitos]

O gerente e atendente devem administrar estoque  
 O atendente deve registrar venda, realizar login  
 O atendente deve registrar pagamento: pagar boleto:  
 pagar dinheiro: pagar cartão  
 O sistema deve gerar nota; imprimir nota

[Respectivo

diagrama de caso de uso]



**Figura 16 - c<sub>1</sub> - Sistema de vendas de produtos eletrônicos.**

**Fonte: Autoria Própria.**

## [Especificação de requisitos]

- O administrador e gerente devem gerar relatório
- O administrador e gerente devem consultar festas
- O administrador e gerente devem gerenciar festas
- O gerente deve adicionar patrocínios
- O usuário deve realizar cadastro
- O usuário deve comprar ingresso, realizar login
- A administradora deve autenticar venda
- O sistema deve gerar ingresso; enviar email

[Respectivo

diagrama de caso de uso]

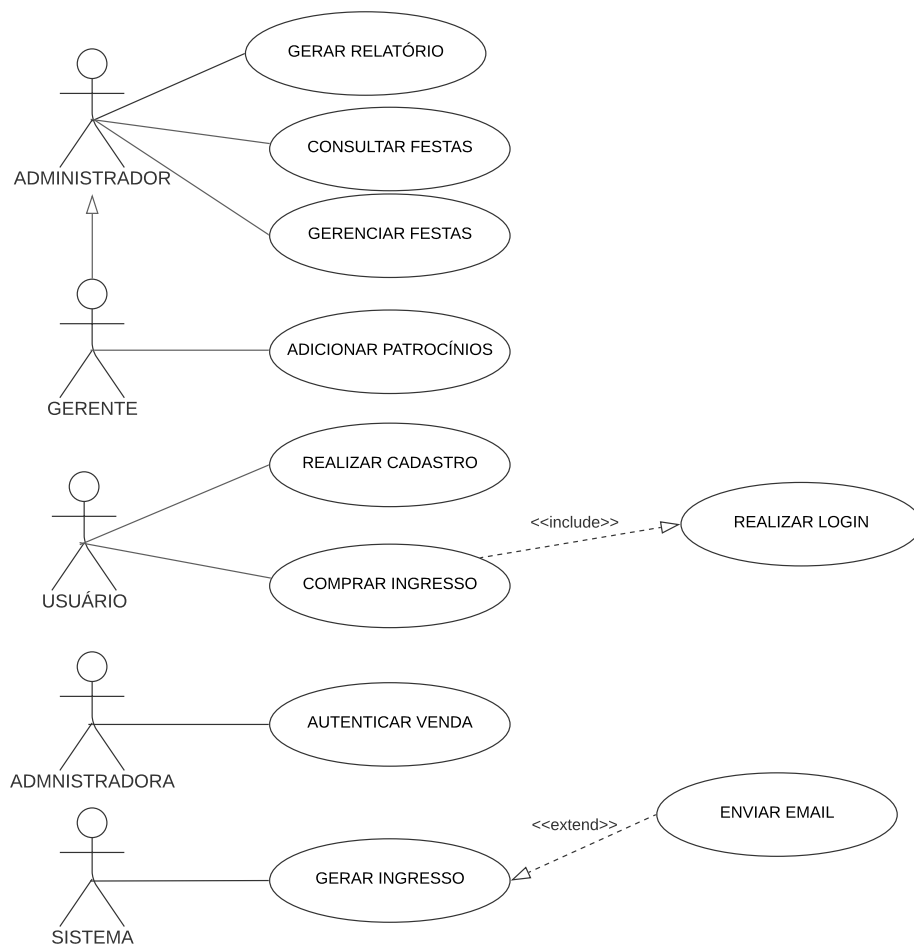


Figura 17 - c2 - Sistema de de gerenciamento de festas

Fonte: Autoria Própria.

## [Especificação de requisitos]

- 0 cliente deve realizar cadastro
- 0 cliente deve gerenciar perfil
- 0 cliente deve realizar locação, realizar login
- 0 gerente e funcionário devem gerenciar automóveis
- 0 gerente e funcionário devem consultar automóveis
- 0 funcionário deve registrar pagamento: pagar cartão: pagar dinheiro
- 0 funcionário deve registrar devolução, gerar nota

[Respectivo

diagrama de caso de uso]

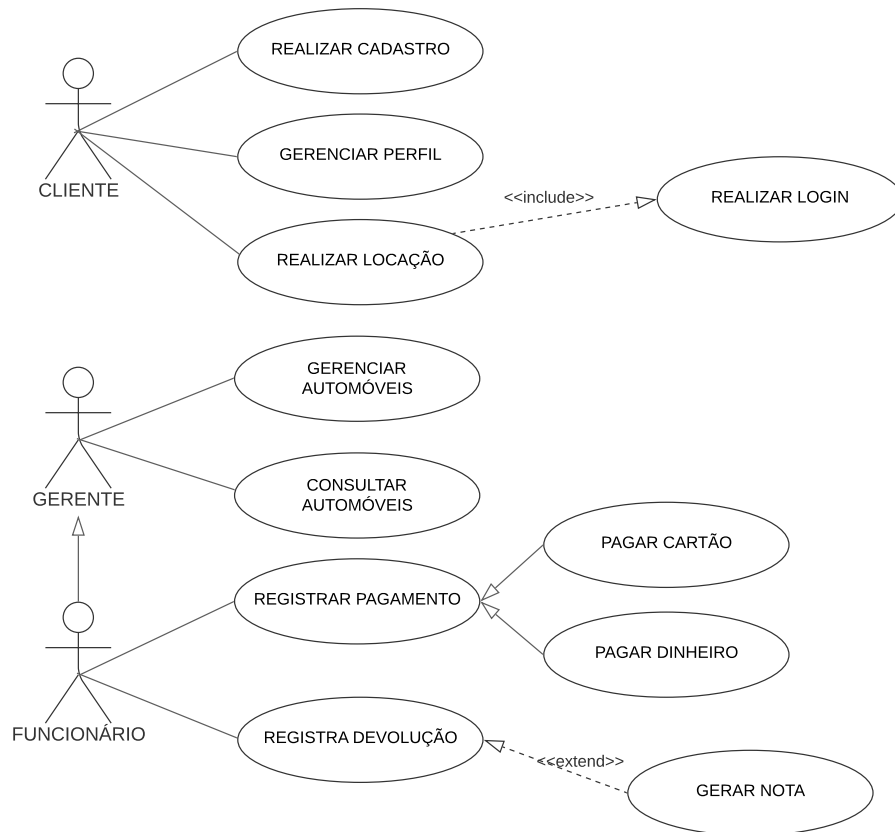


Figura 18 - c3 - Sistema de locação de veículos

Fonte: Autoria Própria.



## [Especificação de requisitos]

O cliente deve agendar consulta, realizar login

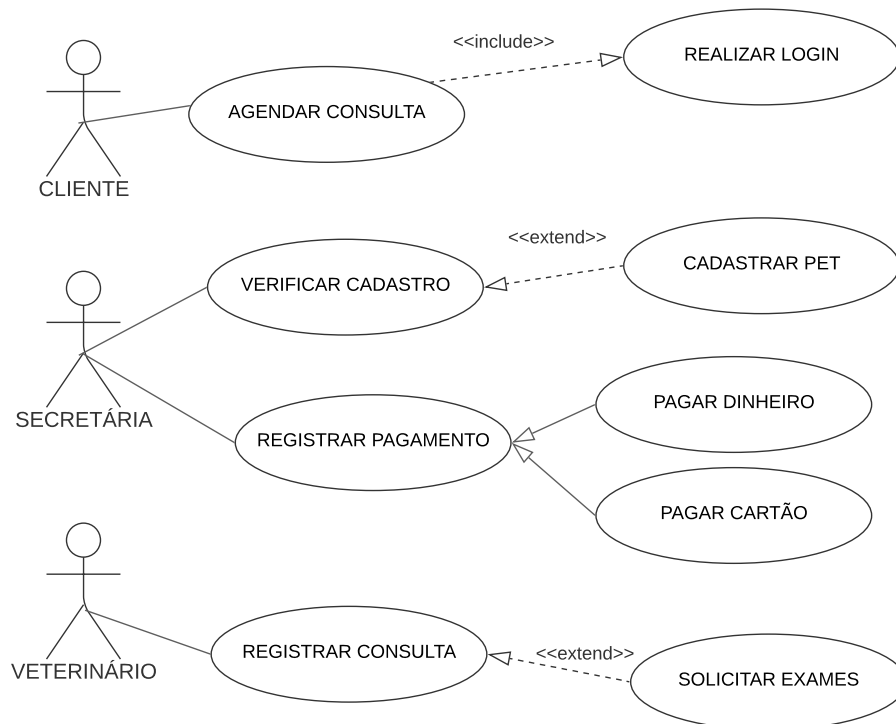
A secretária deve verificar cadastro; cadastrar pet

A secretária deve registrar pagamento: pagar dinheiro:  
pagar cartão

O veterinário deve registrar consulta; solicitar exames

[Respectivo

diagrama de caso de uso]



**Figura 19 - c4 - Sistema de gerenciamento de consultas de uma clínica veterinária**

**Fonte: Autoria Própria.**

## 6 CONSIDERAÇÕES FINAIS

Este trabalho teve como principal objetivo a apresentação da linguagem *BML* para auxiliar estudantes deficientes visuais na especificação de requisitos e da ferramenta *B-Model* para auxiliar a geração do diagrama de caso de uso correspondente a partir dos requisitos especificados. Após a definição de todos os conceitos relacionados ao âmbito deste trabalho, foi realizado um MS com objetivo de analisar o cenário atual das possíveis metodologias, ferramentas, técnicas, notação textual e/ou abordagens para apoiar estudantes com deficiência visual na modelagem de software utilizando UML.

A partir da condução do MS, foi possível identificar limitações sobre as ferramentas, técnicas, abordagens ou notações textuais propostas. Apesar das contribuições identificadas, as mesmas não são capazes de garantir plena inclusão de um estudante deficiente visual em atividades de modelagem utilizando UML. Estas limitações contribuíram para a reafirmação da relevância deste trabalho, para que fosse válido o desenvolvimento da linguagem *BML* e da ferramenta *B-Model* para este contexto.

Com os resultados alcançados a partir do MS, foram propostas a linguagem *BML* e a ferramenta *B-Model*. Para o desenvolvimento da linguagem foram utilizadas a técnica de PLN, juntamente com a definição da classe gramatical conhecida como *POS*, categorias com base no domínio de diagrama de caso de uso e a teoria da Estrutura de Constituintes, proposta por (CHOMSKY, 1956). Por fim, a ferramenta foi desenvolvida em python utilizando as bibliotecas *Spacy* no processamento e interpretação dos requisitos especificados, e a biblioteca *Tkinter* para a geração do diagrama de caso de uso.

Foi realizado um estudo de caso com três discentes deficientes visuais para avaliar a aceitação e uso da linguagem *BML* para auxiliar a especificação de requisitos. O experimento controlado foi conduzido com quatro cenários fictícios para verificar a eficácia da ferramenta *B-Model* na geração do diagrama de caso de uso a partir dos requisitos especificados utilizando a linguagem *BML*.

Portanto, com base nos resultados alcançados as principais contribuições deste

trabalho são:

- (i) visão geral sobre o estado da arte relacionado às metodologias, ferramentas, técnicas, notação textual e/ou abordagens existentes para apoiar estudantes com deficiência visual na modelagem de sistemas utilizando UML;
- (ii) desenvolvimento da linguagem *BML* que facilita os estudantes deficientes visuais na elaboração de uma especificação padronizada dos requisitos para auxiliar a modelagem de software utilizando UML;
- (iii) desenvolvimento da ferramenta *B-Model*, a qual foi eficaz na interpretação e processamento dos requisitos especificados e a sua conversão no respectivo diagrama de caso de uso;
- (iv) publicação e premiação como melhor artigo intitulado “Uma abordagem para auxiliar estudantes com deficiência visual na modelagem de sistemas: um estudo piloto” na IV Escola Regional de Engenharia de Software em 2020, conforme apresentado no Apêndice A.
- (v) O artigo foi convidado para uma versão estendida, em uma edição especial, na revista Revista Eletrônica de Iniciação Científica em Computação (REIC). Atualmente, a edição ainda não foi liberada.

Apesar das contribuições alcançadas, existem algumas limitações a respeito da linguagem *BML* e da ferramenta *B-Model*.

- A linguagem *BML* é próxima da linguagem natural, auxiliando na especificação de requisitos. No entanto, alguns relacionamentos do diagrama de caso de uso como *include*, *extend* e generalização de casos de uso são representados por pontuações, o que ocasiona uma dificuldade para os discentes deficientes visuais durante a especificação de requisitos.
- A ferramenta *B-Model* auxilia na geração do diagrama de caso a partir dos requisitos especificados com a linguagem *BML*. No entanto, a ferramenta não permite que o diagrama possa ser salvo, apenas gerado.

Com base no trabalho conduzido foi possível perceber a ausência de uma ferramenta que viabilize a inserção do estudantes deficientes visuais tanto no âmbito educacional quanto no profissional. Portanto, como trabalhos futuros, serão realizadas melhorias na linguagem e na ferramenta proposta tais como:

- (i) modificação da representação do relacionamento generalização de caso de uso, substituindo “:” pela conjunção “ou”;
- (ii) modificação da representação do relacionamento *include*, substituindo “;” pela palavra “incluindo”;
- (iii) modificação da representação do relacionamento *extend*, substituindo “;” pela palavra “estendendo”;
- (iv) inclusão da funcionalidade de salvar o diagrama gerado; e
- (v) criação de uma interface acessível para facilitar a interação dos estudantes deficientes visuais com a ferramenta.

## REFERÊNCIAS

- ALLEN, J. **Natural Language Understanding**. 15<sup>th</sup>. ed. [S.l.]: Addison-Wesley, 1995.
- BASILI, V. R.; WEISS, D. M. A methodology for collecting valid software engineering data. **IEEE Transactions on Software Engineering**, v. 10, n. 6, p. 728–738, 1986.
- BERA, P. Analyzing the cognitive difficulties for developing and using uml class diagrams for domain understanding. **J. Database Manage.**, IGI Global, v. 23, n. 3, p. 1–29, 2012.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML: guia do usuário**. 2<sup>nd</sup>. ed. Rio de Janeiro: Elsevier, 2006. 474 p.
- BRASIL. "Lei de Acessibilidade - Decreto 5.296, de dezembro de 2004. Brasília, DF, 2004. Disponível em: <[http://www.planalto.gov.br/ccivil\\_03/\\_Ato2004-2006/2004/Decreto/D5296.htm](http://www.planalto.gov.br/ccivil_03/_Ato2004-2006/2004/Decreto/D5296.htm)>. Acesso em: 20 mai. 2019.
- BRERETON, P. et al. Lessons from applying the systematic literature review process within the software engineering domain. **Systems and Software**, New York, USA, v. 80, p. 571–583, 2007.
- BRUMER, A. et al. Saindo da “escuridão”: perspectivas da inclusão social, econômica, cultural e política dos portadores de deficiência visual em porto alegre. **Sociologias**, p. 300–327, 2004. Disponível em: <<http://www.scielo.br/pdf/soc/n11/n11a13.pdf>>. Acesso em: 25 mai. 2019.
- CAT. **Tecnologia assistiva - Subsecretaria Especial dos Direitos Humanos da Presidência da República**. Brasília, DF: [s.n.], 2011. 138 p. Disponível em: <<http://www.abed.org.br/congresso2011/cd/66.pdf>>. Acesso em: 29 mai. 2019.
- CHOMSKY, N. Three models for the description of language. **RE Transactions on Information Theory**, v. 2, n. 3, p. 113–124, 1956.
- COSTA, C. S. L. et al. Análise do conceito de deficiência visual: Considerações para a prática de professores. In: **Educação Especial: aspectos conceituais e emergentes**. [S.l.]: EDUFSCar, 2009. p. 47–62.
- DALLABONA, K. **A inclusão de deficientes visuais no curso superior na educação a distância**. 2011. Disponível em: <<http://www.abed.org.br/congresso2011/cd/66.pdf>>. Acesso em: 29 mai. 2019.
- DAVIS, F. D.; BAGOZZI, R. P.; WARSHAW, R. P. User acceptance of computer technology: A comparison of two theoretical models. **Manage. Sci.**, v. 35, n. 8, p. 982–1003, 1989.

DOHERTY, B.; CHENG, B. H. C. UML modeling for visually-impaired persons. In: **Proceedings of the 1st International Workshop on Human Factors in Modeling co-located with 18th International Conference on Model Driven Engineering Languages and Systems**. [S.l.: s.n.], 2015. p. 4–10.

FLINT, S.; GARDNER, H.; BOUGHTON, C. Executable/translatable uml in computing education. In: **Proceedings of the 6th Australasian Conference on Computing Education**. [S.l.]: Springer International Publishing, 2004. p. 69–75.

GAMBARATO, V. et al. Uso de tecnologias assistivas na educação superior tecnológica. **Tekhne e Logos**, 2012. Disponível em: <[www.fatecbt.edu.br/seer/index.php/tl/article/view/126/113](http://www.fatecbt.edu.br/seer/index.php/tl/article/view/126/113)>. Acesso em: 25 abr. 2019.

GRILLO, F. D. N.; FORTES, R. P. de M. Accessible modeling on the web: A case study. **Procedia Computer Science**, v. 27, p. 460–470, 2014.

IBGE. **Características Gerais da População, Religião e Pessoas com Deficiência**. Rio de Janeiro, RJ, 2010. Disponível em: <[https://biblioteca.ibge.gov.br/visualizacao/periodicos/94/cd\\_2010\\_religiao\\_deficiencia.pdf](https://biblioteca.ibge.gov.br/visualizacao/periodicos/94/cd_2010_religiao_deficiencia.pdf)>. Acesso em: 16 mar. 2019.

KING, A. et al. Presenting uml software engineering diagrams to blind people. In: **Proceedings of the International Conference on Computers for Handicapped Persons**. [S.l.: s.n.], 2004. p. 522–529.

KITCHENHAM, B. et al. Systematic literature reviews in software engineering - a tertiary study. **Inf. Softw. Technol.**, Butterworth-Heinemann, Newton, MA, USA, p. 792–805, 2010.

KITCHENHAM, B. et al. **Evidence-Based Software Engineering and Systematic Reviews**. [S.l.]: Chapman and Hall/CRC, 2015. 433 p.

KUSSLER, L. M. Técnica, tecnologia e tecnociência: da filosofia antia à filosofia contemporânea. In: **Kínesis: Revista de estudos dos pós-graduandos em Filosofia**. [S.l.]: Universidade Estadual Paulista, 2015. p. 187–202.

KUZNIARZ, L.; BORSTLER, J. Teaching modeling — an initial classification of related issues. In: **Proceedings of the 7th Modeling in Education**. [S.l.]: Electronic Communications of the EASST, 2011. p. 69–75.

LEITE, M.; SILVA, G. Inclusão da pessoa com deficiência visual nas instituições de educação superior de belo horizonte. In: **30º encontro da EnANPAD**. [S.l.: s.n.], 2006. p. 1–16.

LIKERT, R. A technique for the measurement of attitudes. **Journal Archives of Psychology**, v. 22, n. 40, p. 1–55, 1932.

LOITSCH, C. et al. UML4ALL syntax – a textual notation for UML diagrams. In: **Proceedings of the 16th International Conference on Computers Helping People with Special Needs**. [S.l.]: Springer, 2018. (ICCHP'18), p. 598–605.

LUQUE, L. et al. Inclusion in computing and engineering education: Perceptions and learning in diagram-based e-learning classes with blind and sighted learners. In: **Proceedings of the IEEE Frontiers in Education Conference**. [S.l.: s.n.], 2017. (FIE'17), p. 1–8.

LUQUE, L. et al. On the inclusion of learners with visual impairment in computing education programs in brazil: practices of educators and perceptions of visually impaired learners. **Journal of the Brazilian Computer Society**, Springer, v. 24, n. 4, p. 1–12, 2018.

LUQUE, L. et al. Can we work together? on the inclusion of blind people in uml model-based task. In: LANGDON, P. et al. (Ed.). **Inclusive Designing**. [S.l.]: Springer, 2014. p. 223–233.

MÜLLER, K. How to make unified modeling language diagrams accessible for blind students. In: **Proceedings of the 13th International Conference on Computers Helping People with Special Needs**. [S.l.]: Springer-Verlag, 2012. (ICHP'12), p. 186–190.

NETO, L. de S. B. et al. A wearable face recognition system built into a smartwatch and the blind and low vision users. In: HAMMOUDI, S. et al. (Ed.). **Enterprise Information Systems**. [S.l.]: Springer International Publishing, 2015. p. 515–528.

OWEN, C. B. et al. Teaching modern object-oriented programming to the blind: An instructor and student experience. In: **Proceedings of the 121st American Society for Engineering Education**. [S.l.: s.n.], 2014. (ASSE'14), p. 1–13.

PANSANATO, L. T. E. et al. Projeto D4ALL: acesso e manipulação de diagramas por pessoas com deficiência visual. In: **Proceedings of the 11th Brazilian Symposium on Human Factors in Computing Systems**. [S.l.: s.n.], 2012. (IHC'12), p. 33–36.

PETRAUSCH, V. et al. Guidelines for accessible textual uml modeling notations. In: **Proceedings of the 16th International Conference on Computers Helping People with Special Needs**. [S.l.]: Springer, 2016. (ICHP'16), p. 167–74.

PRESSMAN, R. S. **Software engineering: A practitioner's approach**. 8<sup>th</sup>. ed. [S.l.]: McGraw-Hill, 2016. 976 p.

RICH, E.; KNIGHT, K. **Inteligência artificial**. 2<sup>nd</sup>. ed. [S.l.]: São Paulo: Makron Books do Brasil, 1994.

SEIFERMANN, S.; GROENDA, H. Survey on textual notations for the unified modeling language. In: **4th International Conference on Model-Driven Engineering and Software Development**. [S.l.: s.n.], 2016. (MODELSWARD'16), p. 28–39.

SOMMERVILLE, I. **Software Engineering**. 10<sup>th</sup>. ed. [S.l.]: Pearson Addison-Wesley, 2016. 816 p.

SUBNORMAL, V. **Sociedade Brasileira de Visão Subnormal**. 2016. Disponível em: <<http://www.visaosubnormal.org.br/oquee.php>>. Acesso em: 26 de maio de 2019.

TERESO, M. Análise de sentimento a companhias aéreas norte americanas. **ISLA Multidisciplinary e-Journal**, v. 2, n. 1, p. 52–65, 2019.

THOMA, A. **Inclusão no ensino superior: “ninguém foi preparado para trabalhar com esses alunos(...) isso exige certamente uma política especial...”**. 2006. Disponível em: <<http://29reuniao.anped.org.br/trabalhos/trabalho/GT15-2552-Int.pdf>>. Acesso em: 17 de março de 2019.

TONET, L. **Pesquisa das Ferramentas de Acessibilidade Computacional para Deficientes Visuais e suas Recomendações do W3C**. 2006. Disponível em: <<https://docplayer.com.br/2672022-Pesquisa-das-ferramentas-de-acessibilidade-computacional-para-deficientes-visuais-e-as-recomendacoes-do-w3c.html>>. Acesso em: 25 de abril 2019.

VIERITZ, H. et al. Access to uml diagrams with the HUTN. In: **Proceedings of the 14th International ACM Sigaccess Conference on Computers and Accessibility**. [S.l.: s.n.], 2012. (ASSETS'12), p. 237–238.

WOHLIN, C. et al. **Experimentation in Software Engineering: An Introduction**. 1<sup>st</sup>. ed. [S.l.]: Springer-Verlag Berlin Heidelberg, 2012.

ZITKUS, E. et al. Accessibility and usability of websites intended for people with disabilities: A preliminary study. In: **Proceedings of the International Conference of Design, User Experience, and Usability**. [S.l.]: Springer International Publishing, 2016. p. 678–688.



**APÊNDICE A - ARTIGO PUBLICADO NO ERES 2020**

# Uma abordagem para auxiliar estudantes com deficiência visual na modelagem de sistemas: um estudo piloto

Pedro O. de Azevedo<sup>1</sup>, Vinícius S. L. Vieira<sup>1</sup>, Alinne C. Correa Souza<sup>1</sup>,  
Francisco Carlos M. Souza<sup>1</sup>

<sup>1</sup>Coordenação de Engenharia de Software  
Universidade Tecnológica Federal do Paraná – Dois Vizinhos – PR – Brasil

azevedo.2015@alunos.utfpr.edu.br, vsantos.lvieira@gmail.com

{franciscosouza, alinnesouza}@utfpr.edu.br

**Abstract.** *In recent years there has been a growing interest in attempts to include people, with special needs, performing Software Engineering activities such as programming and systems modeling. In this context, we present a tool called Blind Modeling system (B-Model) which aims to assist visually impaired people on system modeling activity supported by Unified Modeling Language (UML). For making the use of the B-Model approach feasible, we implemented a language called Blind Modeling Language (BML) and a prototype. The BML was evaluated with a blind student and the prototype was assessed through a pilot study using the requirements specification of two different scenarios performed by blind student for use case diagrams generation. The result shows the effectiveness of the B-Model approach, since the generated diagrams correspond to the specified functional requirements using the BML language.*

**Resumo.** *Nos últimos anos houve um interesse crescente em tentar incluir pessoas com necessidades especiais nas atividades de Engenharia de Software, como programação e modelagem de sistemas. Nesse contexto, neste artigo é apresentada uma ferramenta chamada Blind Modeling system (B-Model), que visa auxiliar pessoas com deficiência visual na atividade de modelagem de sistema utilizando a Linguagem de Modelagem Unificada. Para viabilizar o uso da abordagem B-Model foi desenvolvida uma linguagem chamada Blind Modeling Language (BML) e implementado um protótipo. A BML foi avaliada com uma estudante deficiente visual e o protótipo foi avaliado por meio de um estudo piloto usando a especificação de requisitos de dois cenários diferentes realizada pela estudante deficiente visual para geração de diagramas de casos de uso. O resultado mostra a eficácia da B-Model, pois os diagramas gerados correspondem aos requisitos funcionais especificados utilizando a linguagem BML.*

## 1. Introdução

Segundo o Instituto Brasileiro de Geografia e Estatística [IBGE 2010], 45 milhões de pessoas, ou seja, 23,9% da população total, apresentaram algum tipo de incapacidade ou deficiência. No que diz respeito à deficiência visual, atualmente, existem 6,5 milhões de brasileiros, sendo 582 mil cegos e 6 milhões com baixa visão. Neste contexto, a inclusão de estudantes com baixa visão ao ensino superior tem sido considerada desafiante.

Esse desafio é decorrente da baixa disponibilidade de livros e/ou materiais em formato acessível, leitores textuais, além das dificuldades no aprendizado de notações gráficas e a falta de capacitação de profissionais [Konecki et al. 2016].

Um obstáculo que estudantes com deficiência visual enfrentam no ensino superior é o aprendizado de disciplinas que exijam interpretações visuais [Giroto et al. 2012]. Dentre tais disciplinas é importante destacar Interação Humano Computador, programação para web, programação para dispositivos móveis e Análise Orientada a Objetos (AOO).

A disciplina de AOO visa demonstrar como é realizada a modelagem de sistemas utilizando a Linguagem de Modelagem Unificada (do inglês, *Unified Modeling Language - UML*). Essa modelagem apoia as especificações de um sistema por meio de diagramas, que permitem a compreensão das funcionalidades e comportamentos do software antes de desenvolvê-lo. Neste contexto, é importante destacar que os diagramas são utilizados com frequência para explicar conceitos, projetar sistemas ou mesmo documentar sistemas existentes. Diante deste cenário, os diagramas se tornam inacessíveis para os estudantes com deficiência visual devido à sua natureza gráfica.

Este artigo visa apresentar uma ferramenta chamada *Blind Modeling system (B-Model)*, a fim de auxiliar o aprendizado de estudantes com deficiência visual na modelagem de sistemas utilizando UML. A ferramenta é viabilizada por meio de uma linguagem chamada *Blind Modeling Language (BML)* para a especificação de requisitos, que a partir da qual possibilita gerar automaticamente o diagrama de caso de uso correspondente.

Este artigo está organizado da seguinte forma: na Seção 2 são apresentados os trabalhos relacionados. Na Seção 3 é detalhado a ferramenta B-Model. Na Seção 4 são apresentados a forma como foi conduzido o estudo de caso e experimento piloto. Na Seção 5 são apresentados e discutidos os resultados alcançados. Por fim, na Seção 6 as considerações finais e trabalhos futuros são apresentados.

## **2. Trabalhos Relacionados**

A modelagem de sistemas é realizada por meio de diagramas que encapsulam informações visuais, compostos por formas geométricas e conectores relacionando-as. Diversos estudos tem investigado diferentes formas de auxiliar o ensino e a aprendizagem de modelagem de sistemas para deficientes visuais, tais como: ferramentas [Doherty and Cheng 2015], [King et al. 2004], [Pansanato et al. 2012]; representações físicas [Owen et al. 2014]; e notações textuais [Vieritz et al. 2012], [Grillo and de M. Fortes 2014] e [Loitsch et al. 2018], sendo esta última forma o foco deste artigo.

Apesar dos três estudos abordarem notação textual, somente dois [Harmain and Gaizauskas 2000] e [Elallaoui et al. 2018] utilizaram PLN para a geração de diagramas. No estudo de Harmain e Gaizauskas [Harmain and Gaizauskas 2000] foi desenvolvido um protótipo para geração do diagrama de classes a partir de uma especificação de requisitos em linguagem natural. Por outro lado, o trabalho de Elallaoui, Nafil e Touahni [Elallaoui et al. 2018] apresentou um processo de transformação de histórias de usuários em casos de uso utilizando técnicas de PLN juntamente com o analisador *TreeTagger*.

As diferenças da ferramenta B-Model em relação aos trabalhos apresentados são: (i) o foco no diagrama de caso de uso, uma vez que a partir deste é possível gerar o diagrama de atividades e por estar diretamente relacionado aos requisitos funcionais; (ii) o diagrama gerado está seguindo as notações da UML; e (iii) o uso da teoria de Chomsky [Chomsky 1956].

### 3. Ferramenta B-Model

Nesta seção serão delineados os detalhes da ferramenta B-Model cujo propósito é auxiliar estudantes com deficiência visual no aprendizado da modelagem de sistemas utilizando UML. Conforme pode ser visto na Figura 1, a B-Model é composta por três fases: (i) especificação dos requisitos funcionais em linguagem natural; (ii) interpretação dos requisitos; e (iii) geração do diagrama de caso de uso. Cada fase será detalhada nas próximas seções.

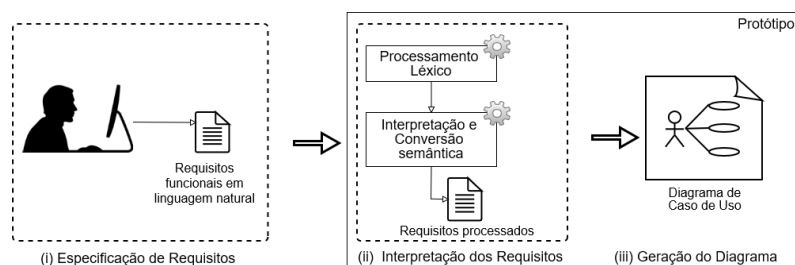


Figura 1. Visão geral da ferramenta B-Model

#### 3.1. Fase 1: Especificação de Requisitos

Os requisitos de software representam as necessidades apresentadas pelos usuários ou clientes, que por sua vez devem ser contempladas pelo sistema a ser desenvolvido. Nesta fase o estudante deverá, em linguagem natural, especificar textualmente os requisitos de um cenário disponibilizado pelo professor, por exemplo.

Para o contexto deste artigo, somente serão considerados os requisitos funcionais, uma vez que estes são necessários para a geração do diagrama de casos de uso. As especificações dos requisitos serão escritos em português salvos em um arquivo de texto que será utilizado como entrada na próxima fase da B-Model.

#### 3.2. Fase 2: Interpretação dos Requisitos

Para auxiliar a interpretação dos requisitos foi definida e desenvolvida uma linguagem chamada *Blind Modeling Language (BML)*. BML é uma linguagem que permite uma padronização na especificação dos requisitos para gerar o diagrama de caso de uso. Nesta fase é realizado o processamento e interpretação dos requisitos especificados utilizando a técnica PLN. A BML foi dividida em duas etapas: (i) processamento Léxico; e (ii) interpretação e conversão semântica.

No processamento léxico (etapa (i) da BML), os requisitos funcionais (entradas) são classificados em categorias conhecidos como *Parts Of Speech (POS)* [Allen 1995]. Nesta etapa foram consideradas as seguintes categorias:

- *determiners*: palavras que determinam o significado de um nome. Neste caso foram considerados os artigos [artigo] definidos no singular (O, o, a, A);

- *nouns*: representam os atores [ator], por exemplo usuário, os objetos [obj], por exemplo login, sistema. Ambos são substantivos e devem ser especificados sempre no singular.
- *verbs*: verbos no infinitivo, por exemplo realizar, que representam a ação que será executada no sistema.
- *prep*: representam as preposições para auxiliar a generalização entre casos de uso, por exemplo entre, em, com.

Além disso, algumas palavras-chaves foram definidas com base no domínio de caso de uso. Para isso, foram criadas três categorias especiais:

- *aux*: auxiliar na definição do nome do caso de uso por meio das palavras “deve” e “devem”;
- *conj*: conjunção para auxiliar na generalização de atores e de caso de uso, por meio das palavras “e”, “ou”, respectivamente;
- *dep*: representam os relacionamentos de *include* e *extend* por meio das palavras “incluindo” e “estendendo”, respectivamente;
- *símbolo*: representado por “;”;

Para a interpretação e conversão semântica dos requisitos funcionais foi utilizada a teoria de Chomsky [Chomsky 1956] que assume que as sentenças podem ser divididas em *Noun Phrase (NP)* e em *Verb Phrase (VP)*. Essa teoria foi aplicada devido utilizar a estrutura de sentenças como: sujeito, verbo e predicado; e facilitar a especificação de requisitos uma vez que é próxima da linguagem natural.

Além da teoria de Chomsky, regras foram definidas para auxiliar a interpretação e a conversão semântica. É importante ressaltar que a definição dessas regras é fundamental, pois permitem normalizar a escrita de uma sentença, ou seja, padronizar os requisitos especificados. A Tabela 1 apresenta as regras da BML utilizadas para geração do diagrama de caso de uso com base na especificação dos requisitos.

**Tabela 1. Regras específicas da BML para a geração do diagrama de caso de uso com base na especificação dos requisitos**

| Elementos                    | Regras   | Exemplo  |
|------------------------------|--|--|
| Ator                         | [artigo] + [ator]  | O usuário  |
| Caso de uso                  | [verbo] + [obj]  | realizar login   |
| Requisito                    | [artigo] + [ator]+ [aux] + [verbo] + [obj]   | O usuário deve realizar login.   |
| Generalização de ator        | [artigo] + [ator] + [conj] + [ator]+ [aux] + [verbo] + [obj]   | O atendente e gerente devem realizar login.                                  |
| Generalização de caso de uso | [artigo] + [ator] + [aux] + [verbo] + [obj] + [símbolo] + [verbo] + [prep] + [obj] + [conj] + [verbo] + [prep] + [obj] | O atendente deve realizar pagamento, pagar com dinheiro ou pagar com cartão. |
| <i>Include</i>               | [artigo] + [ator]+ [aux] + [verbo] + [obj] + [dep] + [verbo] + [obj]   | O atendente deve gerar relatório incluindo realizar login.                   |
| <i>Extend</i>                | [artigo] + [ator]+ [aux] + [verbo] + [obj] + [dep] + [verbo] + [obj]   | O atendente deve gerar relatório estendendo realizar impressão.              |

### 3.3. Fase 3: Geração do Diagrama

Para a geração do diagrama, o protótipo realizará o processamento léxico e a conversão semântica a partir dos requisitos funcionais especificados em língua portuguesa. É importante destacar que neste estudo os diagramas gerados contemplam atores, casos de uso, suas respectivas relações, e a generalização de atores.

O protótipo foi desenvolvido utilizando a linguagem python e duas bibliotecas dessa linguagem: (i) Spacy<sup>1</sup>; e (ii) Tkinter<sup>2</sup>. A biblioteca Spacy foi utilizada para realizar a etiquetagem das palavras em cada sentença buscando por partes do texto que representam os atores, os casos de uso e seus respectivos relacionamentos. Apesar da existência de outras bibliotecas similares, a seleção da *Spacy* foi devido permitir o processamento da língua portuguesa.

Após a etiquetagem das palavras, o processamento e a interpretação dos requisitos especificados são realizados utilizando PLN por meio da biblioteca *Spacy*. Cada parte de um requisito especificado é associado a um componente que irá compor o diagrama de caso de uso. Por fim, com a biblioteca *Tkinter* foi possível gerar o diagrama, uma vez que permite rotular os atores e casos de uso inserindo *labels* aos respectivos componentes.

## 4. Avaliação experimental

Dois estudos experimentais pilotos foram realizados: (i) estudo de caso para avaliar a a utilidade da linguagem BML; (ii) experimento piloto para avaliar a eficácia da ferramenta B-Model. Em ambos os estudos foram utilizados as diretrizes propostas por Wholin et al. [Wholin et al. 2012].

### 4.1. Estudo de Caso

Este estudo foi realizado para avaliar a aceitação e uso da linguagem BML com uma estudante deficiente visual do curso de Bacharelado em Engenharia de Software durante a disciplina de AOO no período de agosto a dezembro de 2019. É importante destacar que algumas melhorias na BML foram realizadas durante o período da disciplina a partir dos *feedbacks* da estudante.

A avaliação da BML foi realizada com base no Modelo de Aceitação de Tecnologia (do inglês, *Technology Acceptance Model (TAM)* [Davis et al. 1989]. O modelo é baseado no pressuposto de que a intenção de utilizar um dispositivo tecnológico é determinada pela percepção do usuário sobre sua utilidade e sua facilidade de uso.

Após a definição e planejamento do estudo de caso, sua fase de condução foi realizada em duas etapas: (1) especificação de requisitos usando a BML e (2) análise da aceitação e uso da BML a partir dos requisitos especificados na primeira etapa. A primeira etapa visa especificar os requisitos de dois cenários diferentes ( $C = c_1$  e  $c_2$ ) utilizando as regras previamente estabelecidas na BML apresentada na Tabela 1. Esta etapa foi precedida por duas atividades. Na primeira atividade, o docente responsável pela disciplina explicou a linguagem para a estudante e criou um material com explicações e exemplos das regras da BML. Na segunda atividade, a descrição dos cenários foram disponibilizados à estudante, conforme apresentado na Tabela 2. Os requisitos de cada cenário foram especificados em um arquivo de texto.

Por fim, a etapa de análise da aceitação e uso está relacionada às respostas obtidas por meio de um questionário. O questionário foi elaborado com 11 questões em um arquivo de texto, as quais foram agrupadas em três categorias: (i) facilidade de uso da linguagem BML; (ii) utilidade da linguagem BML; (iii) intenção de usar a linguagem

---

<sup>1</sup>Spacy documentation - <https://spacy.io/api/doc>

<sup>2</sup>Tkinter documentation - <https://docs.python.org/3/library/tk.html>

**Tabela 2. Descrição dos dois cenários utilizados**

---

**c<sub>1</sub>. Sistema de vendas de produtos eletrônicos:** uma loja possui notebooks e celulares para venda. A loja possui um atendente cuja função é vender os produtos para os clientes e gerar relatórios semanais. A loja também possui um gerente cuja função é administrar o estoque e administrar fornecedores. Na ausência do atendente, o gerente também realiza as suas atividades. Além disso, após a compra o sistema gera a nota fiscal do produto.

---

**c<sub>2</sub>. Sistema de gerenciamento de festas:** um sistema online deve gerenciar a compra e venda de ingressos de festa. Nesse sistema tanto o administrador como o gerente podem gerenciar as festas, gerar relatório e consultar festas. Além disso, o gerente pode adicionar patrocínios. O usuário do sistema realiza cadastro, compra ingresso e emite ingresso. Durante a compra do ingresso, a Administradora do cartão autentica a venda. Ao final da compra, o sistema envia um email com ticket do ingresso.

---

BML no futuro. A estudante realizou a avaliação da BML utilizando a escala *Likert* que consiste em cinco níveis, sendo o nível mais alto representado por “Extremamente” e o nível mais baixo “Nenhum pouco” [Likert 1932].

#### **4.2. Experimento Piloto**

Após a avaliação da BML, foi conduzido um experimento piloto para avaliar a eficácia da ferramenta proposta. Para a definição do experimento, o modelo *Goal-Question-Metric (GQM)* proposto por Basili e Weiss [Basili and Weiss 1986] foi utilizado para definir os objetivos deste experimento que consiste em: “*Analisar a ferramenta B-Model com o propósito de avaliar com respeito à eficácia do ponto de vista de pesquisadores no contexto de dois cenários diferentes.*”

Para alcançar o objetivo, a seguinte Questão de Pesquisa (QP) foi investigada: **O quão eficaz é a ferramenta B-Model para especificar requisitos e gerar o diagrama de caso de uso correspondente?** Para responder essa QP, a eficácia da abordagem foi mensurada por meio da relação do diagrama de caso de uso gerado com a especificação de requisitos utilizando a linguagem BML. Para o contexto deste experimento piloto, a primeira versão do protótipo foi avaliada somente com alguns os conceitos que contemplam o diagrama de caso de uso: (i) ator; (ii) caso de uso; (iii) associação; e (iv) generalização de atores. Portanto, os demais conceitos como *include*, *extend* e generalização de casos de uso serão incluídos na próxima versão do protótipo.

Para realizar a avaliação da ferramenta B-Model foram utilizados os dois cenários apresentados na Tabela 2. Apesar dos cenários serem fictícios, os mesmos compreendem os elementos necessários que um estudante deve abstrair para a modelagem de sistema. Por fim, para a condução do experimento piloto foram utilizadas como entrada as especificações dos requisitos dos cenários, realizadas pela estudante com deficiência visual utilizando a BML. Essas especificações foram inseridas no protótipo, as quais foram processadas e interpretadas para a geração do diagrama de caso de uso correspondente.

### **5. Análise e Discussão dos Resultados**

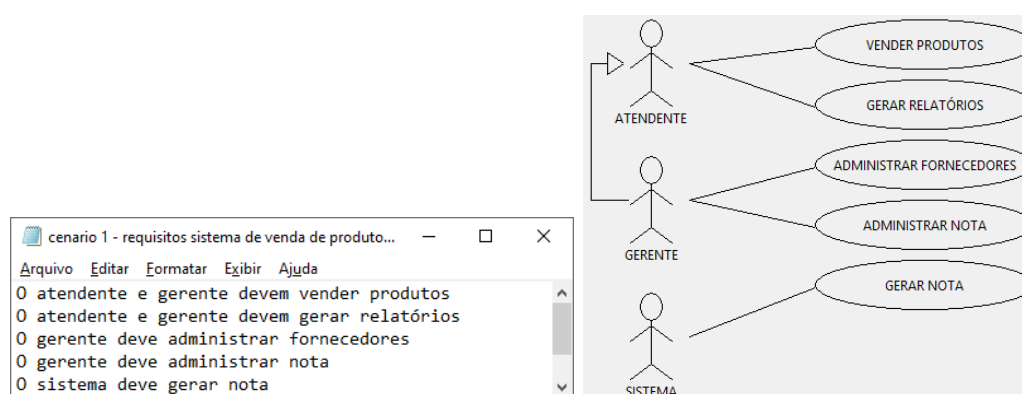
Nesta seção são detalhados os resultados alcançados para a linguagem BML e para a ferramenta B-Model. Na Tabela 3 são apresentados os resultados das respostas da estudante em relação as categorias apresentadas na Seção 4.1. Os resultados sugerem que a BML é uma alternativa eficaz que pode ser utilizada para a especificação de requisitos e geração de diagrama de caso de usos. Um dos principais benefícios da BML é prover uma

especificação mais próxima da linguagem natural, garantindo assim, uma especificação mais consistente, inequívoca e completa para gerar os diagramas de caso de uso.

**Tabela 3. Resultados referentes à facilidade de uso, utilidade e intenção de uso no futuro da BML**

| Categoria                 | Perguntas   | Avaliação |
|---------------------------|---|-----------|
| Facilidade de uso         | 1. A sua interação com a linguagem BML foi clara e compreensível?       | 4         |
|                           | 2. Interagir com a linguagem BML exigiu muito esforço mental?           | 2         |
|                           | 3. Você considera a linguagem BML fácil de usar?                        | 5         |
|                           | 4. Você considera a linguagem BML fácil de ser aprendida?               | 4         |
| Utilidade                 | 5. A linguagem BML melhorou seu desempenho?                             | 5         |
|                           | 6. A linguagem BML permitiu aumentar sua produtividade?                 | 5         |
|                           | 7. A linguagem BML foi útil para o aprendizado?                         | 5         |
|                           | 8. A linguagem BML lhe permitiu realizar trabalhos em grupo?            | 5         |
|                           | 9. A linguagem BML pode lhe ajudar no mercado de trabalho?              | 5         |
| Intenção de uso no futuro | 10. Você pretende usar novamente a linguagem BML em outras disciplinas? | 4         |
|                           | 11. Você recomendaria a linguagem BML para outras pessoas?              | 5         |

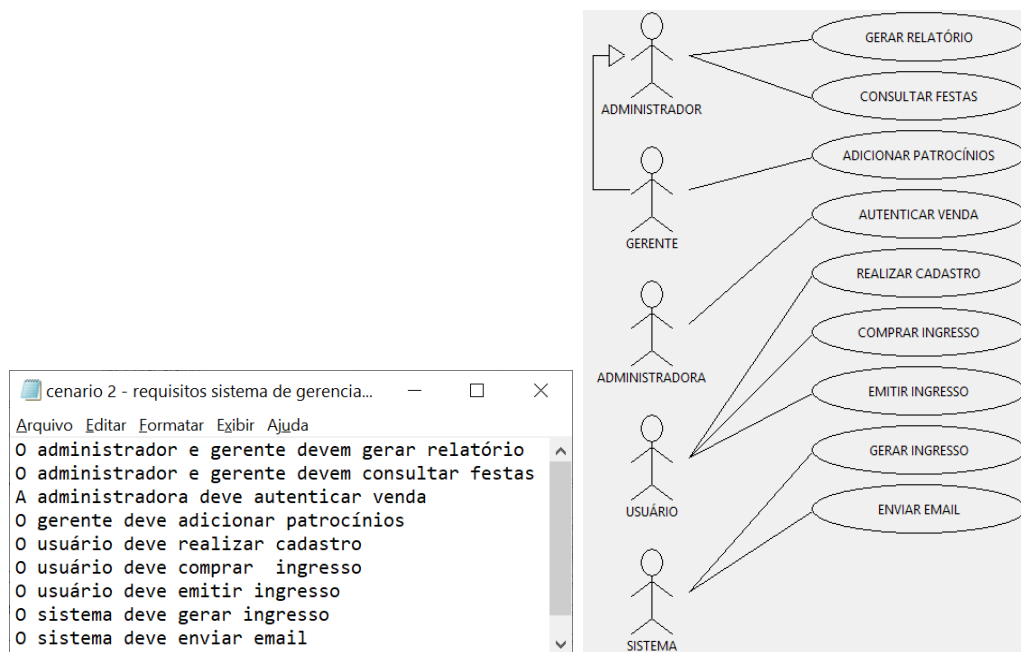
No que diz respeito ao experimento piloto, nas Figuras 2 e 3 são apresentadas as especificações de requisitos e os diagramas de caso de uso gerados para os cenários  $c_1$  (sistema de vendas de produtos eletrônicos) e  $c_2$  (sistema de gerenciamento de festa), respectivamente.



**Figura 2. Especificação de requisitos do sistema de vendas de produtos eletrônicos e Diagrama de caso de uso correspondente gerado**

É importante destacar que o enfoque deste experimento piloto foi corroborar se o protótipo gera o diagrama de caso de uso correspondente à especificação de requisitos utilizando a BML. Para isso, primeiramente, foi realizado um estudo de caso com uma estudante deficiente visual para a avaliação da BML. Nesse contexto, os resultados alcançados indicam que a ferramenta é eficaz, e consequentemente viável para auxiliar estudantes com deficiência visual na modelagem de sistemas. Conforme pode ser observado nas Figuras 3, por exemplo, a ferramenta B-Model foi capaz de reconhecer todos os atores e suas respectivas relações com seus casos de uso; diferenciou corretamente atores com identificadores similares, como é o caso de “Administrador” e “Administradora”; e atribuiu a generalização identificada aos atores que a possuem, considerando a sequência dos mesmos na especificação de requisitos.





**Figura 3. Especificação de requisitos do sistema de gerenciamento de festa e o seu respectivo diagrama de caso de uso gerado**

No âmbito real de ensino e aprendizagem, por exemplo, estudantes com deficiência visual utilizam ferramentas de edição de texto para inserção do conteúdo e leitores de tela para verificação. Especificamente no contexto de modelagem de sistemas, o processo de ensino e construção de diagramas é realizado por meio de uma descrição textual, uma vez que estruturas visuais podem não fazer sentido ou nenhum sentido para estudantes que são deficientes visuais de nascença.

Portanto, não existe problema em realizar uma modelagem de sistemas por meio de uma descrição textual, pois é uma prática comum entre os estudantes com essa deficiência, seja no ensino, médio ou superior. No entanto, um artefato visual, como diagramas, é importante para inclusão destes estudantes em diferentes âmbitos, tais como:

- **ensino:** a modelagem de sistemas atualmente é realizada utilizando o padrão universal UML. Dessa forma, o artefato visual se torna imprescindível para a avaliação do estudante, bem como a sua interação com os demais colegas sem deficiência.
- **indústria:** assim como no ensino, os diagramas de UML são utilizados como documentos e artefatos para auxiliar no processo de desenvolvimento de software. Portanto, por meio de uma ferramenta de geração de diagramas via linguagem natural é possível que estudantes com deficiência visual tenham as mesmas oportunidades para ingressar em um cargo de engenheiro de software, por exemplo.

### 5.1. Ameaças a Validade

Nesta seção são detalhadas as possíveis ameaças à validade que podem afetar os valores e as conclusões do experimento piloto conduzido. As ameaças à validade externa estão relacionadas à generalização dos resultados. A representatividade dos cenários pode ser um problema, pois não existem teorias que garantam que um determinado conjunto

de cenários selecionados seja uma amostra representativa para a condução dos estudos empíricos. Com a finalidade de minimizar esse problema, foram utilizados dois cenários pertencentes a diferentes contextos. No entanto, não é possível afirmar que os resultados podem ser generalizados para qualquer cenário. Neste contexto, replicações futuras são necessárias para corroborar os resultados obtidos.

As ameaças à validade por construção estão preocupadas com a relação entre a teoria e o que é observado. No desenvolvimento da ferramenta proposta bem como na análise dos cenários, possíveis equívocos podem ter sido cometidos. Para mitigar esse risco foram realizados diversos testes das regras a fim de assegurar que durante a condução do experimento a linguagem e a ferramenta tivessem êxito na sua execução.

Por fim as ameaças à validade interna caracterizam o grau de confiabilidade entre os resultados esperados e os resultados obtidos. Todas as variáveis do experimento piloto foram controladas para mitigar possíveis ameaças. Além disso, para ampliar a confiabilidade dos resultados, os dados obtidos foram validados junto as regras definidas para assegurar que os resultados apresentados sejam realmente ortogonais, coerentes e interpretados de forma apropriada.

## 6. Considerações Finais

Neste artigo foram apresentados e discutidos os resultados da ferramenta B-Model que visa auxiliar estudantes com deficiência visual na modelagem de sistemas utilizando UML. Para a geração do diagrama de caso de uso, os requisitos funcionais especificados utilizando a linguagem BML foram processados por meio da técnica PLN juntamente com a teoria de Chomsky. Considerando a importância do diagrama de caso de uso, o foco que este recebeu ao longo do trabalho é justificado pelo fato de ser o principal diagrama utilizado no diálogo com o usuário ao identificar e validar os requisitos. Os casos de uso constituem elementos que estruturam todas as etapas do processo de software e, a partir deste, é possível gerar o diagrama de atividades, por exemplo.

Portanto, os resultados indicam que a ferramenta é promissora e eficaz pelas seguintes razões: (i) permite a descrição dos requisitos por meio de uma linguagem específica reduzindo alguns riscos presentes na especificação de requisitos como incompletude e ambiguidade; e (ii) gera automaticamente o diagrama de caso de uso baseado nos requisitos especificados utilizando a linguagem. Como trabalhos futuros, pretende-se: (i) avaliar a BML com outros estudantes deficientes visuais; (ii) incluir na ferramenta os relacionamentos de dependência do tipo *include* e *extend* e generalizações entre casos de uso; e (iii) validar a ferramenta com estudantes deficientes visuais.

## Referências

- Allen, J. (1995). *Natural Language Understanding*. Addison-Wesley, 15<sup>th</sup> edition.
- Basili, V. R. and Weiss, D. M. (1986). A methodology for collecting valid software engineering data. *IEEE Transactions on Software Engineering*, 10(6):728–738.
- Chomsky, N. (1956). Three models for the description of language. *RE Transactions on Information Theory*, 2(3):113–124.
- Davis, F. D., Bagozzi, R. P., and Warshaw, R. P. (1989). User acceptance of computer technology: A comparison of two theoretical models. *Manage. Sci.*, 35(8):982–1003.

- Doherty, B. and Cheng, B. H. C. (2015). UML modeling for visually-impaired persons. In *Proceedings of the 1st International Workshop on Human Factors in Modeling co-located with 18th International Conference on Model Driven Engineering Languages and Systems*, pages 4–10.
- Elallaoui, M., Nafil, K., and Touahni, R. (2018). Automatic transformation of user stories into uml use case diagrams using nlp techniques. *Procedia Computer Science*, 130:42–49.
- Giroto, C. R. M., Poker, R. B., and Omote, S. (2012). *As tecnologias nas práticas pedagógicas inclusivas*. Editora Unesp, São Paulo.
- Grillo, F. D. N. and de M. Fortes, R. P. (2014). Accessible modeling on the web: A case study. *Procedia Computer Science*, 27:460–470.
- Harmain, H. M. and Gaizauskas, R. (2000). Cm-builder: an automated nl-based case tool. In *Proceedings in 15th IEEE International Conference on Automated Software Engineering*, pages 45–53.
- IBGE (2010). *Características Gerais da População, Religião e Pessoas com Deficiência*. IBGE, Rio de Janeiro, RJ.
- King, A., Blenkhorn, P., Crombie, D., Dijkstra, S., Evans, G., and Wood, J. (2004). Presenting uml software engineering diagrams to blind people. In *Proceedings of the International Conference on Computers for Handicapped Persons*, pages 522–529.
- Konecki, M., Ivković, N., and Kaniški, M. (2016). Making programming education more accessible for visually impaired. In *Proceedings of the 39th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO*, pages 887–890.
- Likert, R. (1932). A technique for the measurement of attitudes. *Journal Archives of Psychology*, 22(40):1–55.
- Loitsch, C., Müller, K., Seifermann, S., Henß, J., Krach, S., and Stiefelhagen, G. J. R. (2018). UML4ALL syntax – a textual notation for UML diagrams. In *Proceedings of the 16th International Conference on Computers Helping People with Special Needs, ICCHP'18*, pages 598–605. Springer.
- Owen, C. B., Coburn, S., and Castor, J. (2014). Teaching modern object-oriented programming to the blind: An instructor and student experience. In *Proceedings of the 121st American Society for Engineering Education, ASSE'14*, pages 1–13.
- Pansanato, L. T. E., Bandeira, A. L. M., dos Santos, L. G., and do Prado Pereira, D. (2012). Projeto D4ALL: acesso e manipulação de diagramas por pessoas com deficiência visual. In *Proceedings of the 11th Brazilian Symposium on Human Factors in Computing Systems, IHC'12*, pages 33–36.
- Vieritz, H., Schilberg, D., and Jeschke, S. (2012). Access to uml diagrams with the HUTN. In *Proceedings of the 14th International ACM Sigaccess Conference on Computers and Accessibility, ASSETS'12*, pages 237–238.
- Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2012). *Experimentation in Software Engineering: An Introduction*. Springer-Verlag Berlin Heidelberg, 1<sup>st</sup> edition.