

# ***B-Model: Uma ferramenta para auxiliar estudantes com deficiência visual na modelagem de sistemas***

**Pedro O. de Azevedo<sup>1</sup>, Francisco Carlos M. Souza<sup>1</sup>,  
Alinne C. Correa 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

{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 the Blind Modeling system (B-Model) tool, which aims to assist visually impaired students on system modeling activity supported by Unified Modeling Language (UML). For making the use of the B-Model tool feasible, we implemented a language called Blind Modeling Language (BML) to standardize the requirements specification to facilitate the automatic generation of use case diagrams. 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.*

**Resumo.** *Nos últimos anos houve um interesse crescente na busca pela inclusão de pessoas com necessidades especiais nas atividades de Engenharia de Software, como programação e modelagem de sistemas. Neste contexto, o artigo apresenta uma ferramenta chamada Blind Modeling system (B-Model), que visa auxiliar estudantes com deficiência visual na atividade de modelagem de sistema utilizando a Linguagem de Modelagem Unificada.<sup>1</sup> Para viabilizar o uso da ferramenta B-Model foi desenvolvida 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. A BML foi avaliada por meio de um estudo de caso com três estudantes deficientes visuais e a ferramenta foi avaliada por meio de um experimento 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.*

---

<sup>1</sup>O presente artigo se trata de uma versão expandida do artigo intitulado de “Uma abordagem para auxiliar estudantes com deficiência visual na modelagem de sistemas: um estudo piloto publicado na IV Escola Regional de Engenharia de Software.”

## 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, o censo retorna 6,5 milhões de brasileiros. Sendo 582 mil cegos, e 6 milhões que apresentam baixa visão. Neste contexto, a inclusão de estudantes com impedimento visual 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 continuada de docentes [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, Engenharia de Software (ES) e Análise Orientada a Objetos (AOO).

As disciplinas de ES e AOO, por exemplo, visam desenvolver nos estudantes a competência da 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)* que foi desenvolvida para 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 padronizar a especificação de requisitos a fim de facilitar a geração automática do 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 linguagem *BML* e a ferramenta *B-Model*. Na Seção 4 são apresentados a forma como foi conduzido o estudo de caso e experimento. 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 (ERF); (ii) Interpretação dos Requisitos Funcionais (IRF); e (iii) Geração do Diagrama (GD).

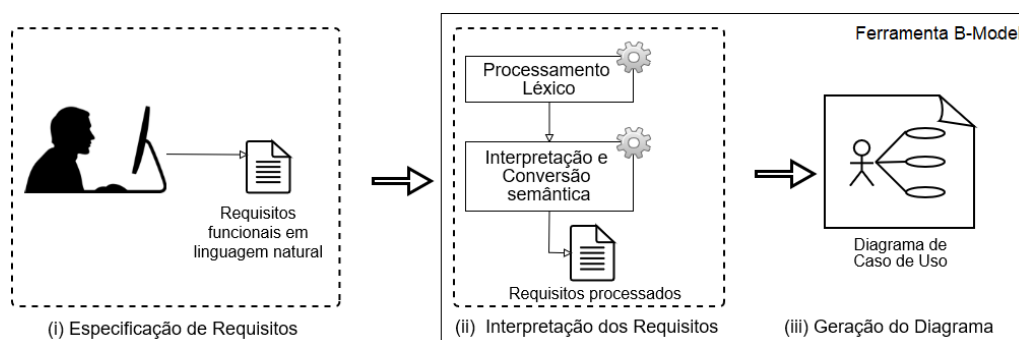


Figura 1. Visão geral da ferramenta B-Model

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

#### 3.1. 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 salvos em um arquivo de texto, o qual será utilizado como entrada na próxima fase da *B-Model*.

### 3.2. 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 específica que permite a padronização da especificação dos requisitos para gerar o diagrama de caso de uso. 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 1.

Além das categorias de *POS* apresentadas na Tabela 1, também foram definidas categorias com base no domínio de diagrama de caso de uso. Na Tabela 2 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.

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 1 e 2, as quais definem as estruturas sintáticas. 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 2. As palavras em maiúsculo denotam símbolos terminais que correspondem às categorias léxicas apresentadas nas Tabelas 1 e 2.

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* inicia com um **AUX** seguido pelo *CasoUso* que consiste de um **VERBO** sucedido de um **OBJT** (objeto).

**Tabela 1. 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                            |

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 3 apresenta um exemplo de uma especificação de requisitos usando a *BML*.

**Tabela 2. Categorias definidas para BML com base nos conceitos de caso de uso.**

| <b>Categorias</b> | <b>Descrição</b>   | <b>Representação na Gramática</b> | <b>Exemplo</b>   |
|-------------------|--|-----------------------------------|--|
| Caso de uso       | Representa as funcionalidades do sistema.  | VERBO OBJ                         | realizar login.  |
| Associação        | Refere-se ao relacionamento de associação ente 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>validar senha</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>                 |

### 3.3. 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.

Para o desenvolvimento da ferramenta *B-Model* foi utilizado a linguagem python e duas bibliotecas dessa linguagem: (i) *Spacy*<sup>2</sup>; e (ii) *Tkinter*<sup>3</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 inúmeras bibliotecas similares, a seleção da *Spacy* foi devido essa 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

<sup>2</sup>*Spacy* documentation - <https://spacy.io/api/doc>

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

|                             |   |   |
|-----------------------------|---|---|
| <b>DiagramaCasoUso</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 2. Gramática referente a linguagem 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 3. Exemplo de uma especificação de requisitos utilizando a BML.**

um requisito especificado é associado a um componente que irá compor o diagrama de caso de uso, conforme pode ser visto na Figura 4. Por fim, com a biblioteca *Tkinter* foi possível gerar o diagrama, uma vez que permite rotular os atores, os casos de uso e os relacionamentos inserindo *labels* aos respectivos componentes.

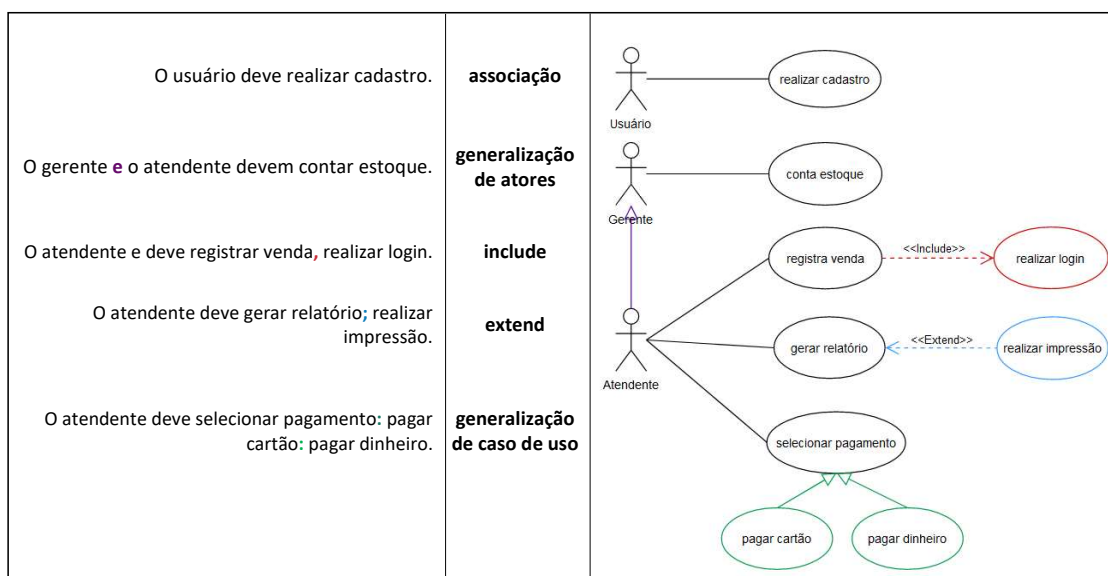
## 4. Avaliação experimental

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

### 4.1. Estudo de Caso

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

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



**Figura 4. Exemplo da geração do diagrama de caso de uso a partir da especificação de requisitos.**

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 et al. 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.

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 2. 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*.
  - 1.2 **Disponibilização dos cenários:** após a explicação realizada pelos docentes, a descrição dos cenários foi disponibilizada aos estudantes, conforme apresentado na Tabela 3. 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 em três categorias: (i) facilidade de uso da linguagem *BML*; (ii) utilidade da linguagem *BML*; (iii) intenção de usar a linguagem *BML* no futuro.

Por fim, a avaliação da *BML* pelos estudantes foi realizada 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].



**Tabela 3. Descrição dos quatro cenários utilizados**

---

**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.

---

## 4.2. Experimento

Após a avaliação a respeito da aceitação e uso da *BML*, foi conduzido um experimento 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 quatro cenários diferentes.”*

O experimento piloto foi realizado por meio de um notebook com processador Intel Core i7 2.4GHz, memória de 8GB e o sistema operacional Windows 10. Para alcançar o objetivo, a seguinte Questão de Pesquisa (QP) foi investigada:

**QP<sub>1</sub>: O 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*.

Quanto ao design do experimento foi manipulada as especificações de requisitos a partir dos cenários como variável independente e o diagrama de caso de uso correspondente como variável dependente. Para realizar a avaliação da ferramenta *B-Model* foram utilizados os quatro cenários apresentados na Tabela 3. 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 foram utilizadas como entrada as especificações dos requisitos dos cenários realizados pelos estudantes com deficiência visual utilizando a *BML*. Essas especificações foram inseridas na ferramenta *B-Model*, 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*.

### **5.1. Aceitação e uso da linguagem *BML***

A aceitação e o uso da *BML* foram avaliados por meio das respostas dos estudantes em relação a facilidade de uso, utilidade e intenção de uso no futuro. Os resultados são apresentados na Tabela 4. 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 estudante (“Est. A”, “Est. B”, “Est. C”)

Os resultados sugerem que a *BML* é uma alternativa eficaz 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.

### **5.2. Eficácia da ferramenta *B-Model***

Nesta seção são apresentados os resultados alcançados em relação a eficácia da ferramenta *B-Model*. Nas Figuras 5 e 6, 7 e 8 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 6, 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

**Tabela 4. Resultados da avaliação referente à facilidade de uso, utilidade e intenção de uso da BML**

| Categoria         | Perguntas   | Avaliação |        |        |
|-------------------|---|-----------|--------|--------|
|                   |   | Est. A    | Est. B | Est. 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      |

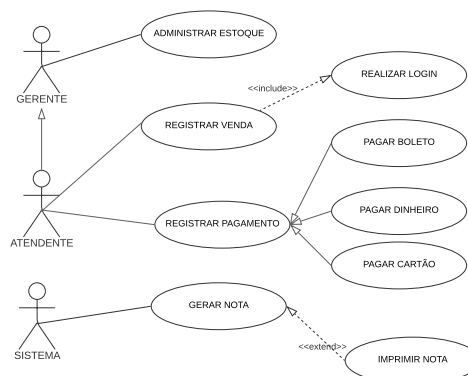
ferramenta é eficaz, e conseqüentemente viável para auxiliar estudantes com deficiência visual na modelagem de sistemas. Ainda é válido destacar a importância da *B-Model* no âmbito real de ensino e aprendizagem, uma vez que estudantes com deficiência visual utilizam ferramentas de edição de texto para inserção do conteúdo e leitores de tela para verificação. Especificamente no contexto de modelagem de sistemas, 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 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 diagrama, 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.

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

(a) Especificação de requisitos

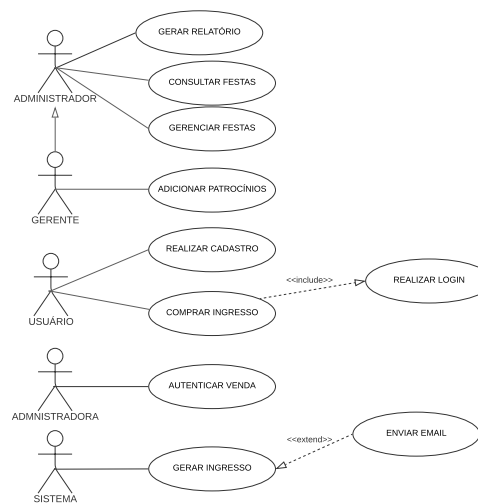


(b) Respectivo diagrama de caso de uso

**Figura 5. c<sub>1</sub> - Sistema de vendas de produtos eletrônicos.**

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

(a) Especificação de requisitos



(b) Respectivo diagrama de caso de uso

**Figura 6. c<sub>2</sub> - Sistema de de gerenciamento de festas**

- 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.

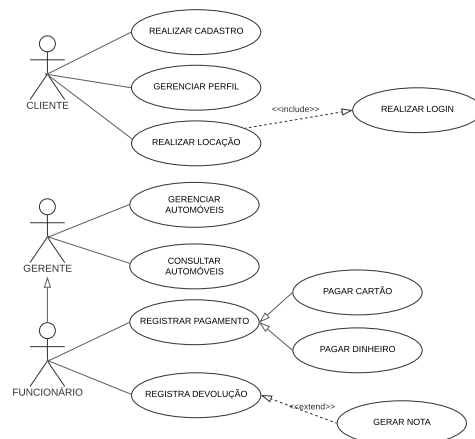
### 5.3. Ameaças a Validade

Nesta seção são detalhadas as possíveis ameaças à validade que podem afetar os valores e as conclusões dos estudos experimentais conduzidos.

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

- 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

(a) Especificação de requisitos

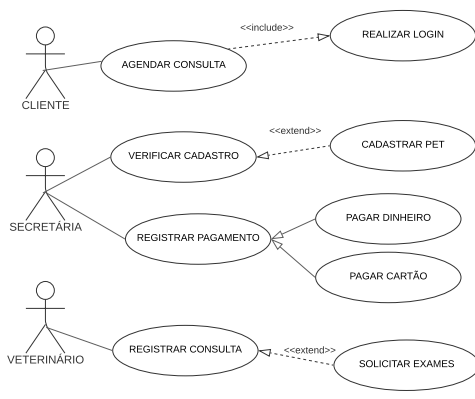


(b) Respectivo diagrama de caso de uso

**Figura 7. c3 - Sistema de locação de veículos**

- 0 cliente deve agendar consulta, realizar login
- A secretária deve verificar cadastro; cadastrar pet
- A secretária deve registrar pagamento: pagar dinheiro; pagar cartão
- 0 veterinário deve registrar consulta; solicitar exames

(a) Especificação de requisitos



(b) Respectivo diagrama de caso de uso

**Figura 8. c4 - Sistema de gerenciamento de consultas de uma clínica veterinária**

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 a 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. *IRE 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.
- Sommerville, I. (2016). *Software Engineering*. Pearson Addison-Wesley, 10<sup>th</sup> edition.
- 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.