

UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS
DEPARTAMENTO DE COMPUTAÇÃO

DANIELA ELISA PRALON
Orientadora: Dayanne Gouveia Coelho

**MATBRAILLE: UMA FERRAMENTA PARA TRANSCRIÇÃO DE
TEXTOS E EXPRESSÕES MATEMÁTICAS EM BRAILLE**

Ouro Preto, MG
2021

UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS
DEPARTAMENTO DE COMPUTAÇÃO

DANIELA ELISA PRALON

**MATBRAILLE: UMA FERRAMENTA PARA TRANSCRIÇÃO DE TEXTOS E
EXPRESSÕES MATEMÁTICAS EM BRAILLE**

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.

Orientadora: Dayanne Gouveia Coelho

Ouro Preto, MG
2021

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

P897m Pralon, Daniela Elisa .

Matbraille [manuscrito]: uma ferramenta para transcrição de textos e expressões matemáticas em Braille. / Daniela Elisa Pralon. - 2021.
77 f.: il.: color., tab..

Orientadora: Profa. Dra. Dayanne Gouveia Coelho.

Monografia (Bacharelado). Universidade Federal de Ouro Preto.
Instituto de Ciências Exatas e Biológicas. Graduação em Ciência da Computação .

1. Projeto de acessibilidade. 2. Braille (Sistema de escrita). 3. Matemática - Fórmulas . I. Coelho, Dayanne Gouveia. II. Universidade Federal de Ouro Preto. III. Título.

CDU 004.5:51

Bibliotecário(a) Responsável: Celina Brasil Luiz - CRB6-1589



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS
DEPARTAMENTO DE COMPUTAÇÃO



FOLHA DE APROVAÇÃO

Daniela Elisa Pralon

MatBRaille: Uma Ferramenta On-line para Transcrição de Textos e Expressões Matemáticas em Braille

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Ciência da Computação

Aprovada em 27 de Agosto de 2021.

Membros da banca

Dayanne Gouveia Coelho (Orientadora) - Doutora - Universidade Federal de Ouro Preto
Rodrigo Geraldo Ribeiro (Examinador) - Doutor - Universidade Federal de Ouro Preto
Camila Freitas de Miranda (Examinadora) - Especialista - Universidade Federal de Ouro Preto

Dayanne Gouveia Coelho, Orientadora do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 27/08/2021.



Documento assinado eletronicamente por **Dayanne Gouveia Coelho, PROFESSOR DE MAGISTERIO SUPERIOR**, em 27/08/2021, às 16:21, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0210306** e o código CRC **600BAE10**.

Referência: Caso responda este documento, indicar expressamente o Processo nº 23109.008217/2021-98

SEI nº 0210306

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35400-000
Telefone: 3135591692 - www.ufop.br

Agradecimentos

Agradeço à todos que contribuíram para a realização deste trabalho e/ou em minha graduação em especial: meus pais, minha irmã e meu namorado; companheiros de curso por dicas e sugestões de melhorias; a toda equipe do Núcleo de Educação Inclusiva da Universidade Federal de Ouro Preto que me apoiaram e me inspiraram em especial Adriene, Marcelo e Wallisson. Agradeço também todos os professores do ICEB que indiretamente ou diretamente contribuíram na minha formação acadêmica em especial as professoras e orientadoras de projetos e monografia Amanda e Dayanne.

Tudo é vário. Temporário. Efêmero.

Nunca somos, sempre estamos. (Chico Buarque)

Resumo

No Brasil existem aproximadamente 582 mil pessoas cegas e 6 milhões de pessoas com baixa visão e a educação em todos os níveis acadêmicos é garantido por lei. Entretanto, há dificuldades na preparação e adaptação das aulas para que essas se tornem inclusivas. Neste contexto, este trabalho tem por objetivo a proposição do MatBRaille, que consiste em uma ferramenta executável voltada para a transcrição de símbolos alfanuméricos para o Sistema Braille. A ideia surgiu da demanda da transcrição de materiais com cunho matemático no ensino superior de forma gratuita e do pequeno número de programas gratuitos que realizam a transcrição para o Sistema Braille. Neste, referido estudo primeiramente buscamos introduzir noções do Sistema Braille em conformidade com as normas brasileiras de transcrição para o Sistema Braille, noções a respeito das teorias usadas na construção da aplicação, trabalhos relacionados, desenvolvimento da aplicação e resultados obtidos após a realização de testes preliminares a respeito receptividade da aplicação proposta. Utilizando o modelo de avaliação SUS, obteve-se uma pontuação de 83.04, comprovando que a ferramenta será útil para o seu propósito, contribuindo de maneira positiva para a diminuição da defasagem de aplicações de transcrição braille e promovendo uma maior acessibilidade.

Palavras-chave: Inclusão; Fórmulas matemáticas; Braille; Transcritor.

Abstract

In Brazil there are approximately 582,000 blind people and 6 million people with low vision and education at all academic levels is guaranteed by law. However, there are difficulties in preparing and adapting classes so that they become inclusive. In this context, this work aims to propose MatBRaille, which is an executable tool aimed at transcribing alphanumeric symbols to the Braille System. The idea arose from the demand for free transcription of materials with a mathematical nature in higher education and the small number of free programs that carry out the transcription for the Braille System. In this study, we first sought to introduce notions of the Braille System in accordance with the Brazilian standards for the Braille System, notions about the theories used in the construction of the application, related works, application development, and results obtained after conducting preliminary tests regarding receptivity of the proposed application. Using the SUS assessment model, a score of 83.04 was obtained, proving that the tool will be useful for its purpose, contributing positively to reducing the gap in Braille transcription applications and promoting greater accessibility.

Keywords:Inclusion; Mathematical formulas; Braille; Transcriber.

Lista de Ilustrações

Figura 2.1 – Exemplo de uma célula braille.	8
Figura 2.2 – Exemplo de uma célula braille 8 pontos.	8
Figura 2.3 – Alfabeto romano em Braille.	10
Figura 2.4 – Célula braille invertida	16
Figura 2.5 – Reglete e punção.	17
Figura 2.6 – Máquina de escrever braille.	18
Figura 2.7 – Impressora braille.	18
Figura 2.8 – Linha Braille.	20
Figura 3.1 – Exemplos de árvore sintática	26
Figura 3.2 – Exemplo de máquina de estados.	28
Figura 3.3 – Estado inicial de um autômato	28
Figura 3.4 – Estado final de um autômato	28
Figura 3.5 – Representação de transições em um autômato	28
Figura 3.6 – Exemplo de Autômato Finito.	29
Figura 3.7 – Exemplo de Autômato Finito Determinístico.	29
Figura 3.8 – Exemplo de Autômato Finito Não Determinístico.	30
Figura 3.9 – Exemplo de Tabela de transição.	30
Figura 3.10–Exemplo de análise léxica	34
Figura 3.11–Exemplo de análise léxica	35
Figura 5.1 – Diagrama de Hierarquia dos Atores.	42
Figura 5.2 – Diagrama de Contexto	42
Figura 5.3 – Tela inicial da ferramenta MatBraille.	44
Figura 5.4 – Tela com exemplo de transcrição de um texto.	44
Figura 5.5 – Tela de inserção de fórmulas matemáticas.	45
Figura 5.6 – Captura de tela contendo MatBRaille transcrevendo um pangrama que contém todas as 26 letras do alfabeto.	47
Figura 5.7 – Captura de tela com MatBRaille transcrevendo um pangrama que contém todos os acentos da língua portuguesa.	48
Figura 5.8 – Captura de tela contendo MatBRaille transcrevendo um pangrama contendo todo alfabeto.	48
Figura 5.9 – Captura de tela contendo a janela de inserções de equações matemáticas.	49
Figura 5.10–Captura de tela contendo MatBRaille transcrevendo uma fórmula matemática.	49
Figura 5.11–Captura de tela contendo MatBRaille transcrevendo números naturais e decimais.	50
Figura 6.1 – Captura de parte da explicação de polinômios contida em (STEWART, 2008)	54
Figura 6.2 – Inserção de índice inferior	54
Figura 6.3 – Inserção de fração	55

Figura 6.4 – Transcrição de paragrafo 55

Lista de Tabelas

Tabela 2.1 – Representação em braille para vogais com acentos.	10
Tabela 2.2 – Alguns sinais exclusivos do sistema braille.	11
Tabela 2.3 – Alguns símbolos e pontuações em braille.	12
Tabela 2.4 – Representação dos números de 0 9.	13
Tabela 2.5 – Tabela contendo valores decimais	14
Tabela 2.6 – Tabela contendo números contendo mais de quatro algarismos	14
Tabela 2.7 – Tabela de simbologias matemáticas	15
Tabela 2.8 – Tabela de exemplos de índices inferior e superior	15
Tabela 2.9 – Tabela de Integrais	16
Tabela 3.1 – Tabela de definição de Tokens	25
Tabela 3.2 – Tabela de Integrais	33
Tabela 3.3 – Tokens de uma linguagem de programação	33
Tabela 4.1 – Transcritores on-line para Braille	40
Tabela 5.1 – Requisitos funcionais	43
Tabela 5.2 – Requisitos não funcionais	43
Tabela 6.1 – Respostas individuais formulário SUS	53

Lista de Abreviaturas e Siglas

ABNT	Associação Brasileira de Normas Técnicas
AFD	Autômato Finito Determinístico
AFND	Autômato Finito Não Determinístico
AsTeR	Audio System For Technical Readings
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CMU	Código Matemático Unificado
CSS	Cascading Style Sheets
DAISY	Digital Accessible Information System
DECOM	Departamento de Computação
EPUB	Electronic PUBlication
HTML	Hypertext Markup Language
IBGE	Instituto Brasileiro de Geografia e Estatística
ICEB	Instituto de Ciências Exatas e Biológicas
INEP	Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira
JS	JavaScript
JAWS	Job Access With Speech
LARAMARA	Associação Brasileira de Assistência ao Deficiente Visual
LaTeX	Lamport TeX
MathML	Mathematical Markup Language
MEC	Ministério da educação
MP3	MPEG Layer 3
NCE	Núcleo de Computação Eletrônica
NEI	Núcleo de Educação Inclusiva
NVDA	Non Visual Desktop Access

PDF	Portable Document Format
StArt	State of the Art through Systematic
UFOP	Universidade Federal de Ouro Preto
UFRJ	Universidade Federal do Rio de Janeiro
UFSCar	Universidade Federal de São Carlos
RFB	Recordings for the Blind
SUS	System Usability Scale
TIC	Tecnologia da informação e comunicação
XML	Extensible Markup Language
W3C	World Wide Web Consortiu

Sumário

1	Introdução	1
1.1	Justificativa e Motivação	2
1.2	Objetivos	3
1.3	Método de Trabalho	4
1.4	Organização do Trabalho	4
2	O Sistema Braille	6
2.1	Breve Histórico	6
2.2	Sistema de Escrita Braille	7
2.3	Grafia Braille Para Língua Portuguesa	9
2.3.1	Alfabeto	10
2.3.2	Números	13
2.3.3	Braille e a Matemática	14
2.4	Instrumentos para Escrita em Braille	15
2.4.1	Reglete e Punção	16
2.4.2	Máquina de Escrever Braille	17
2.4.3	Impressora Braille	18
2.5	Softwares de Auxílio ao Deficiente Visual	19
2.5.1	Leitores de Tela	19
2.5.2	Reconhecimento de Voz	20
2.5.3	Transcrição de Caracteres Alfanuméricos para Braille	20
2.5.4	Optical Braille Recognition	20
2.5.5	Livros digitais	21
2.5.6	Acessibilidade na web	21
2.6	Deficiência Visual no Brasil	22
3	Análise Léxica	24
3.1	Definições preliminares	24
3.2	Breve contextualização a respeito de Compiladores	25
3.3	Autômatos Finitos	27
3.4	Linguagem Regular	31
3.5	Expressões regulares	31
3.6	Gerador Léxico	34
4	Trabalhos Relacionados	36
5	Desenvolvimento	41
5.1	Modelagem da Aplicação	41
5.1.1	Atores	41
5.1.2	Diagrama de Contexto	42

5.1.3	Requisitos Funcionais e Não Funcionais	43
5.2	Protótipos de tela	43
5.3	Implementação	45
5.3.1	Tecnologias	45
5.3.2	Funcionalidades desenvolvidas	46
6	Resultados	51
6.1	SUS - <i>System Usability Scale</i>	51
6.2	Resultados Preliminares	52
6.3	Aplicação do MatBRaille	54
7	Considerações Finais	56
7.1	Trabalhos Futuros	56
	Referências	58

1 Introdução

Segundo [Bengala Legal \(2006\)](#), a palavra acessibilidade quando tratada como substantivo pode ser entendida como acessível, por outro lado, quando classificada como adjetivo remete ao alcance sem obstáculos. Na área da deficiência, a princípio, mencionava-se o termo acessibilidade como um sinônimo de “eliminação de barreiras”. Assim, promover a acessibilidade consiste em proporcionar oportunidades iguais de acesso a bens e serviços, não levando em consideração a classe social, escolaridade, idade, deficiência e/ou gênero. Logo, a ideia de acessibilidade não está ligada apenas ao conjunto de pessoas que possuem alguma deficiência, mas a necessidade de promover acesso a um item de qualquer natureza a todos indivíduos.

No contexto da pessoa com deficiência, de qualquer natureza ou mobilidade reduzida, a acessibilidade atua na viabilização da utilização segura e autônoma de espaços, equipamentos urbanos, edifícios, meios de transporte, meios de comunicação, tecnologias, serviços públicos ou privados de uso coletivo. Enquanto a inclusão consiste na ação de incluir, ou seja, diz respeito ao paradigma de tornar quaisquer ambientes acessíveis a todos e novamente não apenas a uma parcela ou grupo de pessoas. A inclusão visa combater a exclusão de qualquer pessoa a qualquer meio, bem ou espaço.

Com o intuito de promover a inclusão de deficientes visuais no Brasil, existem diversos órgãos governamentais e não governamentais. As duas maiores referências a respeito da deficiência visual no país são:

- a Fundação Dorina Nowill para Cegos que distribui, de forma gratuita, livros produzidos usando o Sistema Braille e em formato de áudio. Além de produzir variados tipos de materiais digitais acessíveis, inclusive, contribuindo para produção de livros de 3000 escolas, bibliotecas e organizações brasileiras;
- o Instituto Benjamin Constant (IBC) que é correlacionado ao Gabinete do Ministro de Estado da Educação, atende crianças e adolescentes cegos, surdos cegos, com baixa visão ou deficiência múltipla, além de capacitar e assessorar profissionais para atuarem nestas áreas. O IBC, conta também com um importante trabalho de reabilitação de pessoas que estão ou ficaram cegas, contando inclusive com um renomado centro de pesquisas na área oftalmologia. O Instituto, também produz e defende pesquisas na área da educação inclusiva, além de editar e imprimir revistas em braille.

No âmbito das tecnologias, o termo tecnologia assistiva consiste em um grupo de dispositivos, técnicas e processos que objetivam a assistência, melhoria e reinserção de deficientes. Neste aspecto, apontamos o Núcleo de Computação Eletrônica (NCE) da Universidade Federal

do Rio de Janeiro (UFRJ), que é responsável pelo desenvolvimento de diferentes tecnologias gratuitas voltadas para deficientes visuais.

No ensino regular, podemos citar: recursos auditivos, lupas e/ou materiais táteis, como auxiliares da aprendizagem de deficientes visuais. A transcrição de textos para o Sistema Braille, que consiste no processo de reescrita do material disponível em caracteres alfanuméricos para a uma versão em relevo com mesma significância, vem sendo realizada por softwares.

Dados divulgados pelo Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP) apontam que no ano de 2020, foram matriculados no ensino superior: 16.376 deficientes físicos, 13.906 alunos com baixa visão, 6.569 deficientes auditivos e 2.598 estudantes com cegueira. O número total de matrículas de alunos com deficiência, transtornos globais do desenvolvimento ou altas habilidades/superdotação de 2009 para 2019 saltou de 20.530 para 48.520, assim em 2019 o percentual em relação ao total de matrículas em cursos de Graduação era de 0,56% (INEP, 2020).

Nesta perspectiva, neste trabalho é proposto o MatBRaille, que consiste em um editor de texto voltado para a confecção de material usando o Sistema Braille e em concordância com a ortografia do português e o código matemático unificado. A ferramenta criada tem como funcionalidade, além da transcrição de caracteres alfanuméricos, a transcrição da simbologia matemática usada em disciplinas dos cursos superiores da área de exatas. Vale destacar que para o uso do editor não há a necessidade de um conhecimento prévio do usuário a respeito do Sistema Braille ou programação, o que viabiliza o seu uso para uma maior produção de livros didáticos e notas de aulas e, conseqüentemente, gerar uma maior inclusão de alunos com deficiência visual no meio acadêmico.

1.1 Justificativa e Motivação

A Lei n.º 7.853 de 1989¹ em seu artigo primeiro afirma que "ficam estabelecidas normas gerais que asseguram o pleno exercício dos direitos individuais e sociais das pessoas portadoras de deficiências, e sua efetiva integração social, nos termos desta Lei". Assim, em todo território brasileiro é assegurado por lei, que a pessoa deficiente deve ser amparada de modo que não lhe seja restringido saúde, lazer, trabalho ou educação. Entretanto, os dados do Censo 2010, divulgados pelo Instituto Brasileiro de Geografia e Estatística (IBGE), mostram uma realidade diferente: entre as pessoas deficientes no Brasil com mais de quinze anos apenas 14,15% possuem ensino fundamental completo ou médio incompleto, 17,67% têm ensino médio completo ou superior completo e 6,66% apresentam um diploma de graduação (IBGE, 2010).

Glat e Pletsch (2004) reforçam que um dos problemas no ensino brasileiro inclusivo é a falta de capacitação dos professores para receber e dar aulas de modo inclusivo. E agregado

¹ BRASIL. *Decreto LEI Nº 7.853, DE 24 DE OUTUBRO DE 1989*. 1989. Disponível em <https://www.planalto.gov.br/ccivil_03/leis/l7853.htm>. Acesso em: 17 nov. 2019.

a esse despreparo, a produção de materiais táteis tende a ser mais trabalhosa ou necessita de um alto valor de investimento, o que acarreta diretamente no menor número de materiais táteis de qualquer natureza. E como foi bem dito por Freire et al. (1982) a leitura é um meio de adquirir conhecimentos artísticos e científicos. E no caso dos deficientes visuais a leitura ocorre através do Sistema Braille. Porém, as tecnologias assistivas, que realizam a transcrição de caracteres alfanuméricos para o Sistema Braille, em muitos dos casos, não transcreve a parte matemática. Pois, em maioria, as fórmulas matemáticas no momento da transcrição são transformadas em imagens ou em códigos indecifráveis para softwares transcritores, sendo necessária a descrição de fórmulas e simbologias matemáticas voltados para conteúdos do ensino superior da área de exatas, como, por exemplo, a transcrição de materiais do Cálculo Diferencial e Integral.

Sendo assim, a criação de novas tecnologias assistivas, e que preferencialmente sejam distribuídas de forma gratuita, são de grande importância não apenas para a melhora da qualidade de ensino e inclusão dos alunos deficientes visuais, mas também para a permanência destes em cursos e qualificações. Voltado para esta temática, este trabalho busca identificar os *softwares* disponíveis no mercado que realizam a transcrição de textos matemáticos, além de fazer um levantamento sobre funcionalidades das ferramentas, verificando assim, se estas atendem as demandas do ensino superior. Em um segundo momento, foi desenvolvida uma nova alternativa de transcrição de textos para o Sistema Braille, o MatBRaille.

Assim, o presente trabalho se justifica pela relevância do tema proposto e também por contribuir com a inclusão dos alunos com deficiência visual dentro do ambiente universitário. O transcritor MatBRaille poderá ser usado para realizar a reprodução do material usando Sistema Braille, principalmente aqueles que envolvem simbologias e fórmulas matemáticas para alunos com deficiência visual da Universidade Federal de Ouro Preto (UFOP). Vale destacar, que uma das motivações para o desenvolvimento da ferramenta, foi a dificuldade vivenciada por estagiários do Núcleo de Educação Inclusiva (NEI) da UFOP em fazer a transcrição das notas de aulas dos professores que lecionam disciplinas com conteúdos matemáticos para Braille.

1.2 Objetivos

O objetivo principal deste trabalho é o desenvolvimento do MatBraille, que consiste em uma plataforma executável, disponível para os sistemas operacionais Windows e Mac, que realiza a transcrição dos textos e fórmulas matemáticas da escrita alfanumérica do português para o Sistema Braille.

São objetivos específicos deste trabalho:

- Propor um transcritor para o Sistema Braille gratuito e que não necessite de conhecimento prévio a respeito do Sistema Braille ou programação;

- Possibilitar a geração de sentenças de matemáticas visíveis em tela de forma comumente usada para videntes e sua transcrição correta de acordo com o Sistema Braille;
- Gerar um documento com fonte braille e formato que permita a impressão em braille;
- Aumentar a inclusão dos alunos com deficiência visual dentro do ambiente universitário.

1.3 Método de Trabalho

Para o desenvolvimento da ferramenta MatBRaille em um primeiro momento, foi realizada uma pesquisa bibliográfica, com o intuito de compreender os principais conceitos que envolvem a problemática deste trabalho e uma busca por trabalhos análogos, com o objetivo de fazer um levantamento das principais tecnologias disponíveis no mercado e de suas funcionalidades.

Em seguida, baseado no estado da arte e os objetivos propostos nesta monografia, foram definidos os principais requisitos, funcionalidades e linguagens para a criação da ferramenta MatBRaille, que será utilizada para atender uma demanda específica, que consiste em fazer a transcrição de textos e fórmulas matemáticas da escrita alfanumérica para o Sistema Braille. Quanto à ferramenta desenvolvida, optou-se por uma plataforma executável.

Definidas as funcionalidades, os protótipos das telas, requisitos funcionais e não funcionais do sistema são apresentadas, além das primeiras implementações realizadas para a conclusão deste trabalho. Por fim, uma primeira avaliação foi aplicada sob a versão preliminar da ferramenta, reforçando a receptividade com relação à usabilidade da plataforma proposta.

1.4 Organização do Trabalho

O restante deste trabalho apresenta a seguinte estrutura: o Capítulo 2 descreve o Sistema Braille, como é representada a grafia braille para a língua portuguesa, que além de seguir a gramática da língua portuguesa trás um conjunto de simbologias que visam representar efeitos visuais, como por exemplo, o efeito de sublinhado que é representado no Sistema Braille por ⠠ no início e fim da palavra sublinhada, além de tecnologias assistivas voltadas aos deficientes visuais.

No capítulo posterior, Capítulo 3 é abordada uma segunda fundamentação teórica, que tem como base introduzir conceitos de ciência da computação usados neste trabalho e que interferem no melhor entendimento de como se deu a implementação deste trabalho.

No Capítulo 4 são exibidos e descritos de forma breve as aplicações online e instaláveis que realizam de forma gratuita a transcrição de caracteres alfanuméricos para o Sistema Braille. Para fazer o levantamento dessas aplicações foi utilizado as primeiras etapas de uma revisão sistemática da literatura.

O Capítulo 5 descreve as metodologias de desenvolvimento empregadas nas duas etapas deste trabalho, apresentando a hierarquia de atores, o diagrama de contexto, os requisitos funcionais e não-funcionais, além dos protótipos das telas da aplicação proposta.

No Capítulo 6 são apresentados os resultados preliminares alcançados até o momento e uma avaliação da usabilidade do sistema para avaliar as primeiras implementações realizadas da ferramenta. Por fim, o Capítulo 7 apresenta as considerações finais a respeito do trabalho realizado e trabalhos futuros.

2 O Sistema Braille

(o ato de ler) não se esgota na decodificação pura da palavra escrita ou da linguagem escrita, mas que se antecipa e se alonga na inteligência do mundo (FREIRE et al., 2003).

A Associação de deficientes visuais e amigos (2019) afirma que o braille é “um código universal que permite às pessoas cegas beneficiar-se da escrita e da leitura, dando-lhes acesso ao conhecimento, favorecendo sua inclusão na sociedade e o pleno exercício da cidadania.”

O presente capítulo faz a introdução do Sistema Braille e se encontra organizado da seguinte maneira: na Seção 2.1 é apresentada uma breve contextualização de como se deu o processo da criação no Sistema Braille; na Seção 2.2 são dadas noções sobre o Sistema Braille, as principais adaptações da grafia braille para a língua portuguesa de acordo com as novas regras ortográficas estão presentes na Seção 2.3; na Seção 2.4 os principais instrumentos usados para a escrita em braille; na Seção 2.5 os *softwares* usados para auxílio de deficientes visuais e, por fim, na Seção 2.6 é apresentada uma contextualização do quadro de deficiência no Brasil.

2.1 Breve Histórico

Em 4 de janeiro de 1809, na pequena cidade Coupvray, pertencente ao distrito francês Seine-Marne, nascia Louis Braille. Três anos mais tarde, brincando na oficina de seu pai, ao tentar perfurar um pedaço de couro, Louis teve seu olho esquerdo ferido por um objeto pontiagudo. Após grave hemorragia, o ferimento infeccionou e devido à escassez de auxílio médico, meses depois a infecção atingiu o olho direito. Aos cinco anos Louis estava cego, pois a infecção generalizada arruinou ambas as córneas.

Aos dez anos, Louis ganhou uma bolsa de estudos no Instituto Real dos Cegos de Paris, a primeira escola de cegos do mundo. Na instituição Louis se mostrou um aluno brilhante e, inclusive, se tornou um excelente pianista. Além de aprender a ler um alfabeto ampliado e com várias polegadas de abertura e largura.

Após um amigo Louis ler um artigo publicado em jornal, Louis teve acesso à informação da existência de um modo alternativo para leitura para cegos, o código militar chamado “escrita noturna”. Desenvolvido por Charles Barbier, Capitão de Artilharia do Exército de Luís XIII, a "escrita noturna" que usava pontos e traços em alto-relevo que permitiam uma comunicação segura e no escuro entre soldados. Barbier, então, transformou-o em um sistema de escrita para cegos e a denominou como Grafia Sonora, pois o sistema era fonético. Qualquer frase poderia ser escrita, mas palavras não poderiam ser soletradas. Apesar deste sistema ser complicado, o sistema foi adotado como um método auxiliar de ensino na instituição frequentada por Braille.

Fazendo uso da Grafia Sonora, qualquer frase podia ser escrita, entretanto, Louis achou pequenas falhas no sistema, como por exemplo, palavras não podiam ser soletradas e vários sinais eram empregados na escrita de uma única palavra. Assim, vislumbrando um sistema de leitura sem falhas, aos 15 anos, Louis desenvolveu um alfabeto semelhante ao atual usando 63 combinações de uma cela de seis buracos, estes delimitados por um pequeno espaço. Sendo possível também representar acentuação, pontuação e sinais matemáticos.

Apesar das melhorias e aceitação dos alunos, o sistema desenvolvido por Braille não foi aceito de imediato, contudo, Louis nunca parou de aperfeiçoar seu trabalho chegando a publicar em 1838 a obra: “Pequena Sinopse de Aritmética para Principiantes”. No ano seguinte, Louis voltou a contribuir para a comunidade de deficientes visuais publicando o “Novo método para representação por sinais de formas de letras, mapas, figuras geométricas e símbolos musicais para uso de cegos”. Esta segunda invenção, denominada por Braille de “grafia pontilhada”, também foi bem aceita pelos cegos e contribuiu para a comunicação entre cegos e videntes.

Apenas 1943, o então diretor do Instituto Real dos Cegos de Paris, aceitou o ensino do sistema de Braille, fato que muito emocionou o criador do Sistema Braille. À partir desse marco, o sistema de Braille começou a ser ensinado por toda a Europa. Entretanto, aos 26 anos, Braille contraiu tuberculose e já com a saúde bem debilitada aos 41 anos, Braille se viu obrigado a demitir-se do cargo de professor que ocupava no Instituto Real para cegos, entretanto não deixou de lecionar aulas de piano. Em 1852 Braille faleceu devido a tuberculose.

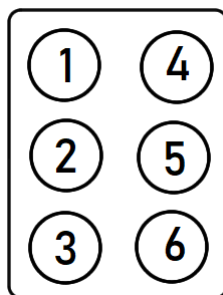
Em 1954, o sistema de Braille foi adaptado pela escola de cegos de Paris e disseminado pelo mundo. Em 17 de setembro deste mesmo ano, foi criado no Brasil, pelo Imperador D. Pedro II na cidade do Rio de Janeiro-RJ, o Instituto Benjamin Constant, sendo a primeira instituição de Educação Especial da América Latina e um ponto inicial nos direitos dos cegos no Brasil.

Esta seção foi escrita conforme dados que foram coletados e adaptados presentes em [Instituto Benjamin Constant \(2019\)](#), [CBO \(2019\)](#) e [Bezerra et al. \(2003, p.04\)](#), para maiores informações recomenda-se a leitura completas dos documentos referidos.

2.2 Sistema de Escrita Braille

O braille é um sistema de escrita e leitura tátil utilizado por deficientes visuais, onde são usadas a combinação de seis pontos distribuídos em uma cela braille ou célula braille, que é um espaço retangular composto por seis pontos dispostos em duas colunas verticais, tendo cada coluna três pontos ([BRASIL, 2018, p.17](#)). A [Figura 2.1](#) apresenta o exemplo de uma célula braille.

Figura 2.1 – Exemplo de uma célula braille.



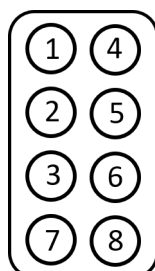
Na célula braille, a leitura dos seis pontos é realizada de cima para baixo, de modo que ao fim da contagem dos três pontos pertencentes a primeira coluna mais a esquerda, a contagem é retomada na coluna mais a direita à partir do primeiro ponto indicando, o ponto quatro.

Segundo **BRASIL (2018)** o código braille de seis pontos produz sessenta e três sinais diferentes ou sessenta e quatro para aqueles que consideram a célula vazia como um símbolo, e a combinação desses sinais permite representar alfabetos, números, códigos matemáticos, sinais de pontuação, simbologia química, musical e da informática. A sua escrita pode se dar por meio da:

- forma soletrada, ou seja, letra por letra (denominado braille de grau 1);
- ou fazendo uso de abreviaturas que variam de acordo com cada língua ou grupo linguístico podendo representar as conjunções, preposições, pronomes, prefixos, sufixos, grupos de letras (denominado grau 2). Este segundo método é um grande redutor de volumes de livros impressos, permitindo um melhor ritmo de leitura e escrita. Se enxergarmos as representações abreviadas como um grupo, podemos dizer que este grupo é composto pelo grau 2, já descrito e o denominado grau 3, que a apresenta a forma mais complexa, no sentido de uma exigência de conhecimento da língua e sensibilidade tátil.

Outra variação do Sistema Braille é o Braille de oito pontos, que tem como intuito principal ampliar a quantidade de representação de símbolos de modo a reduzir todos os símbolos a apenas uma célula braille. A ideia é inserir um sétimo ponto, abaixo do ponto três e o oitavo ponto abaixo do ponto seis como pode ser visualizado na Figura 2.2.

Figura 2.2 – Exemplo de uma célula braille 8 pontos.



É importante destacar que o braille não é um idioma e sim um alfabeto tátil, que pode ser usado para se escrever em alguns idiomas, como, por exemplo o português. Em outras palavras, o código braille é universal e o que muda são as combinações que esse código apresenta na hora de alfabetizar o deficiente visual em um determinado idioma.

A representação do Sistema Braille em português é normalizado por “Sistema Grafia Braille para a Língua Portuguesa”, “Normas Técnicas para a Produção de Textos em Braille”, “Grafia Química Braille para Uso no Brasil”, “Código Matemático Unificado para a Língua Portuguesa”, “Grafia Braille para Informática”, “Manual Internacional de Musicografia Braille”, “Etenografia Braille para a Língua Portuguesa - abreviações” e “A Construção do Conceito de Número e o Pré-Soroban”.

2.3 Grafia Braille Para Língua Portuguesa

Devido às particularidades do idioma português em relação à pontuação e acentuação o Ministério da Educação aprovou pela portaria no 2.678¹ uma grafia braille adaptada para a língua Portuguesa.

Nesta seção, será usado o documento (BRASIL, 2018) para descrever a grafia braille para a língua portuguesa. Este documento é uma normatização da grafia braille para a língua portuguesa, destinado principalmente para professores, transcritores, revisores e usuários do sistema de escrita braille. O objetivo do documento é permitir que às pessoas cegas brasileiras tenham acesso à leitura e à escrita de textos. Para informações adicionais, recomenda-se a consulta ao referido documento.

Além dos principais símbolos da escrita braille, o documento apresenta algumas alterações, novos símbolos e um conjunto de normas para a aplicação de toda a simbologia braille respeitando o novo acordo ortográfico da língua portuguesa, com o intuito de garantir a padronização dos textos em braille produzidos em todo o país. Segundo BRASIL (2018) “o principal objetivo deste documento é permitir que o Sistema Braille continue sendo o instrumento fundamental na educação, habilitação, reabilitação e profissionalização das pessoas cegas”.

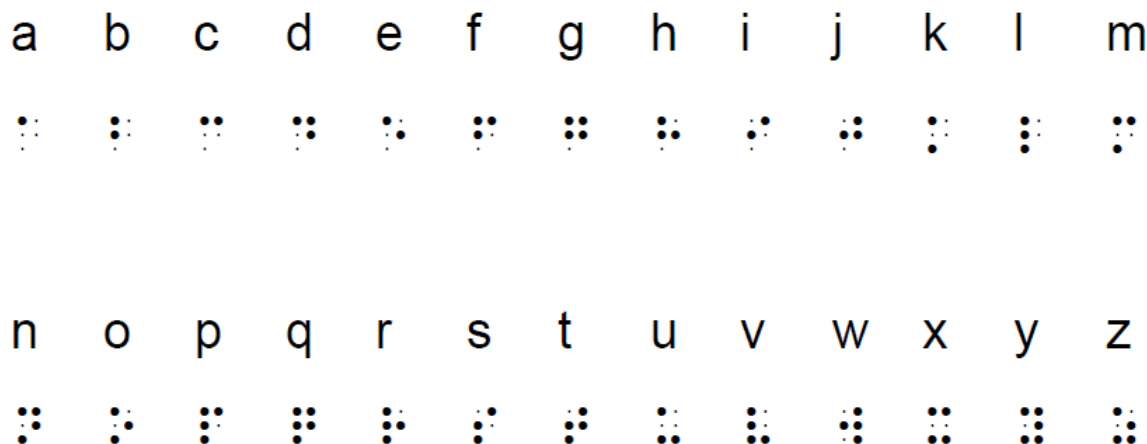
Neste contexto são apresentados na Subseção 2.3.1 a representação em braille do alfabeto romano e os principais símbolos de pontuação e acentuação e na Subseção 2.3.2 a explicação de como é realizado a representação dos números. Por fim, na Subseção 2.3.3 é realizada uma breve descrição de como são representados as principais operações e símbolos matemáticos em braille.

¹ Portaria nº 2.678 de 24/09/2002. Disponível em <<https://www.fnde.gov.br/index.php/aceso-a-informacao/institucional/legislacao/item/3494-portaria-mec-n%C2%BA-2678-de-24-de-setembro-de-2002>>. Data de acesso 08/12/2019.

2.3.1 Alfabeto

A Figura 2.3 apresenta o alfabeto braille e as letras respectivas no alfabeto romano usado na língua portuguesa.

Figura 2.3 – Alfabeto romano em braille. Os pontos mais escuros representam o que estará em relevo para percepção tátil.



Fonte: BRASIL (2018, p. 23)

Mesmo não fazendo parte do alfabeto usado na língua portuguesa, a letra “ç” compõem um grande número de palavras em português e sua transcrição na simbologia braille é representada pelos pontos ⠨ (pontos 12346).

As vogais podem ser acentuadas nas palavras escritas em português, o que resulta em uma grafia diferente de suas representações não acentuadas. A Tabela 2.1 apresenta as representações em braille das vogais com acentuação.

Tabela 2.1 – Representação em braille para vogais com acentos.

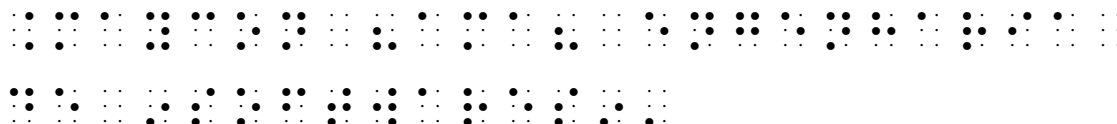
Vogais	a	e	i	o	u
Representação em braille	⠁	⠅	⠇	⠏	⠥
Representação braille com acento agudo	⠁́	⠅́	⠇́	⠏́	⠅́
Representação braille com acento grave	⠁̀	-	-	-	-
Representação braille com acento circunflexo	⠁̂	⠅̂	-	⠏̂	-
Representação braille com acento til	⠁̃	-	-	⠏̃	-

A Tabela 2.2 apresenta alguns sinais exclusivos do sistema braille, tais sinais foram incluídos no intuito de representar em braille alguns efeitos visuais ou regras, como por exemplo

a inicialização de nomes próprios com letras maiúsculas. Já Tabela 2.3 traz os sinais de pontuação internacional e alguns símbolos usados na língua portuguesa.

Tabela 2.2 – Alguns sinais exclusivos do sistema braille. Texto extraído de (BRASIL, 2018).

Sinais	Representação em Braille
Sinal de letra Maiúscula	⠠
Sinal de maiúscula em todas as letras da palavra (caixa alta)	⠠⠠⠠
sinal de série de palavras com todas as letras maiúsculas	⠠⠠⠠⠠
sinal de número	⠠
sinal índice superior	⠠
sinal de índice inferior	⠠
sinal de itálico, negrito ou sublinhado	⠠ ou ⠠⠠
variante tipográfica de números	⠠
sinal de minúscula latina; sinal especial de translineação de expressões matemáticas e sinal de translineação em contexto informático	⠠



2.3.2 Números

A Tabela 2.4 apresenta a representação em braille dos algarismos de zero a nove. Note que para representar os números são utilizados os seguintes caracteres específicos: ⠼ (pontos 3456). Esta célula é usada para diferenciar a representação numérica da representação de uma letra do alfabeto.

Tabela 2.4 – Representação dos números de 0 9.

Números	Representação
1	⠼⠠
2	⠼⠡
3	⠼⠢
4	⠼⠣
5	⠼⠤
6	⠼⠥
7	⠼⠦
8	⠼⠧
9	⠼⠨
0	⠼⠠

Note que na Figura 2.3 as escritas em braille da letra *a* é ⠁ e da letra *b* é ⠃, e, na Tabela 2.4, as escritas em Braille do número *um* é ⠼⠠ e do número *dois* é ⠼⠡. Para diferenciar os números 1 e 2 das letras a e b é adicionado o sinal de representação do número (⠼) a combinação dos pontos. Assim, a letra *a* será representada por ⠼⠠ e o número *um* por (⠼⠠).

Quando um número é formado por dois ou mais algarismos, só o primeiro número será precedido pelo sinal ⠼. E no caso da representação de um número decimal, para separar a parte inteira da parte decimal deve ser usado o sinal da vírgula ⠴. Alguns exemplos estão representados na Tabela 2.5.

Para separar as classes na parte inteira dos números com mais de quatro algarismos, será usado o sinal ⠼, alguns exemplos estão expostos na Tabela 2.6.

Tabela 2.5 – Tabela contendo valores decimais

Representação alfanumérica	Representação braille
0,75	⠠⠚⠛⠚⠠⠚⠚⠠⠚⠚
4,5	⠠⠚⠚⠠⠚⠠⠚⠚
7639,125	⠠⠚⠚⠠⠚⠚⠠⠚⠚⠠⠚⠚⠠⠚⠚⠠⠚⠚⠠⠚⠚

Tabela 2.6 – Tabela contendo números contendo mais de quatro algarismos

Representação alfanumérica	Representação braille
53 550 018	⠠⠚⠚⠠⠚⠚⠠⠚⠚⠠⠚⠚⠠⠚⠚⠠⠚⠚
4.000.000	⠠⠚⠚⠠⠚⠚⠠⠚⠚⠠⠚⠚⠠⠚⠚⠠⠚⠚
22.950,07	⠠⠚⠚⠠⠚⠚⠠⠚⠚⠠⠚⠚⠠⠚⠚⠠⠚⠚

2.3.3 Braille e a Matemática

Assim como os números as operações matemáticas também podem ser representadas fazendo uso do Sistema Braille, na Tabela 2.7 estão expostos alguns exemplos de operadores matemáticos e símbolos mais básicos. Vale lembrar que a transcrição da simbologia da matemática básica já se faz presente em alguns softwares, inclusive de forma simples e direta.

Segundo [Carvalho et al. \(2019\)](#) o braille é uma escrita linear, contudo, há representações para sobrescrito ou subscrito em braille, veja a Tabela 2.8.

Como a transcrição/escrita em braille ocupam um maior volume que a escrita a tinta. Assim, surgiram códigos como Código Nemeth para uso dos norte americanos, o Código Italiano, o Marburg para os que dominam o alemão, dentre outros, que visam a busca de uma representação de códigos matemáticos mais curta ([BERNAREGGI; ARCHAMBAULT, 2007](#)).

O Código Matemático Unificado (CMU) ([BRASIL, 2006](#)), para a Língua Portuguesa é uma tentativa de unificação do sistema braille utilizado na Europa e na América Latina. Este documento contém as regras de como representar através do alfabeto braille: “Prefixos alfabéticos e sinais unificadores”, “Índices e marca, Números (decimais, fracionários, ordinais, romanos, etc)”, “Operações aritméticas fundamentais e relações elementares”, “Frações, potências e raízes”, “Teoria de conjuntos e lógica”, “Aplicações (funções) e Geometria”. Sendo que no capítulo de Aplicações são demonstrados a codificação de funções de nível superior de ensino como Derivadas e Integrais (Tabela 2.9).

Tabela 2.7 – Tabela de simbologias matemáticas

Sinal matemático	Símbolo	Representação em braille
Sinal de adição: “mais”. Positivo	+	⠠⠨
Sinal de subtração: “menos”.Negativo	-	⠠⠤
“multiplicado por”	×	⠠⠢
“multiplicado por”	•	⠠⠨⠠⠨
sinal de divisão: “dividido por”; igual para todas as formas de representar a divisão	: ou / ou ÷	⠠⠠⠨
sinal de igualdade: “é igual a”	=	⠠⠨⠠⠨
“aproximadamente igual a”	≈	⠠⠨⠠⠨⠠⠨
“assim como”	::	⠠⠨⠠⠨⠠⠨
“menor que”	<	⠠⠨⠠⠨
“menor ou igual a”, para todas as variantes em tinta que tenham este mesmo significado.	≤	⠠⠨⠠⠨⠠⠨
“maior que”	>	⠠⠨⠠⠨
“maior ou igual a”, representação de todas as variantes que têm este mesmo significado.	≥	⠠⠨⠠⠨⠠⠨
“é diferente de”	≠	⠠⠨⠠⠨⠠⠨
“por cento”	%	⠠⠨⠠⠨⠠⠨





Tabela 2.8 – Tabela de exemplos de índices inferior e superior

Sinal matemático	Representação em braille
7^2	⠠⠠⠨⠠⠨⠠⠨
2^n	⠠⠠⠨⠠⠨⠠⠨
cm^3	⠠⠠⠨⠠⠨⠠⠨⠠⠨
4_2	⠠⠠⠨⠠⠨⠠⠨
a_1	⠠⠠⠨⠠⠨⠠⠨
x_n	⠠⠠⠨⠠⠨⠠⠨

2.4 Instrumentos para Escrita em Braille

Seja de cunho tecnológico ou não, os equipamentos de escrita braille promovem um grau a mais de independência e liberdade aos cegos, uma vez que promovem a possibilidade de registro e documentação de quaisquer tipos de declarações. Nos tópicos subsequentes conheça alguns aparelhos que permitem a escrita braille.

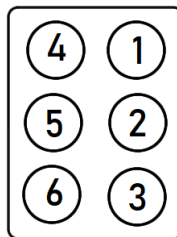
Tabela 2.9 – Tabela de Integrais

Sinal matemático	Símbolo	Representação em Braille
\int		integral indefinida
$\int \int$		integral dupla
$\int \int \int$		integral tripla
\int_a^b		integral definida

2.4.1 Reglete e Punção

Reglete consiste em duas placas, que fixadas entre uma folha de papel, permitem a escrita braille. A placa superior funciona como uma régua, possuindo várias células braille, já a parte inferior contém o relevo da célula braille, podendo ser confeccionada em metal, plástico ou madeira (BEZERRA et al., 2003, p.11). Não existe a padronização da confecção de regletes e a escrita fazendo uso de regletes é feita da direita para a esquerda, ou seja, usando a pontuação da célula braille invertida, como pode ser vista na Figura 2.4.

Figura 2.4 – Célula braille invertida



Já a punção é uma espécie de agulha grossa, que usada conjuntamente com o reglete, marca ponto a ponto, cela a cela, a o Sistema Braille. Veja na Figura 2.5 um exemplo de ambos objetos.

Figura 2.5 – Reglete e punção.



Foto: Felipe Lazzarotto/ EPTV (2014)

2.4.2 Máquina de Escrever Braille

Outro recurso para a escrita do sistema braille é máquina de escrever braille, que consiste em uma máquina de datilografia contendo sete teclas, sendo que seis das sete teclas representam cada uma um ponto braille e a restante é usada para demarcar o espaço em branco, ou cela vazia. O primeiro protótipo foi elaborado em 1939 pelo Prof. David Abraham. Através de um acordo de cooperação internacional entre a Perkins, a maior fabricante de máquinas de escrever braille, e a LARAMARA (Associação Brasileira de Assistência à Pessoa com Deficiência Visua), em abril de 1999, foi inaugurada no Brasil, a primeira fábrica de máquinas de escrever em braille da América Latina (BEZERRA et al., 2003, p.12).

Figura 2.6 – Máquina de escrever braille.



Fonte: Projeto acesso (2019)

2.4.3 Impressora Braille

As impressoras braille conjuntamente com softwares de transcritores são as grandes produtoras de materiais em relevo. Além da impressão de grandes volumes de materiais para leitura, as impressoras braille também viabilizam a impressão de desenhos, tabelas e gráficos táteis. Assim como impressoras comuns, as impressoras braille podem realizar impressão em ambos os lados de uma folha sem gerar erros de transcrições em um dos lados da folha (COLPES, 2014).

Figura 2.7 – Impressora braille.



2.5 Softwares de Auxílio ao Deficiente Visual

As tecnologias assistivas são tecnologias voltadas para o auxílio de deficientes em quaisquer atividades. A seguir são descritos os diferentes tipos de *softwares* ou tecnologias desenvolvidos para facilitar a realização de tarefas de deficientes visuais. Entretanto, devido ao interesse educacional deste trabalho, diversas tecnologias que auxiliam no dia-a-dia do deficiente visual não foram abordadas.

2.5.1 Leitores de Tela

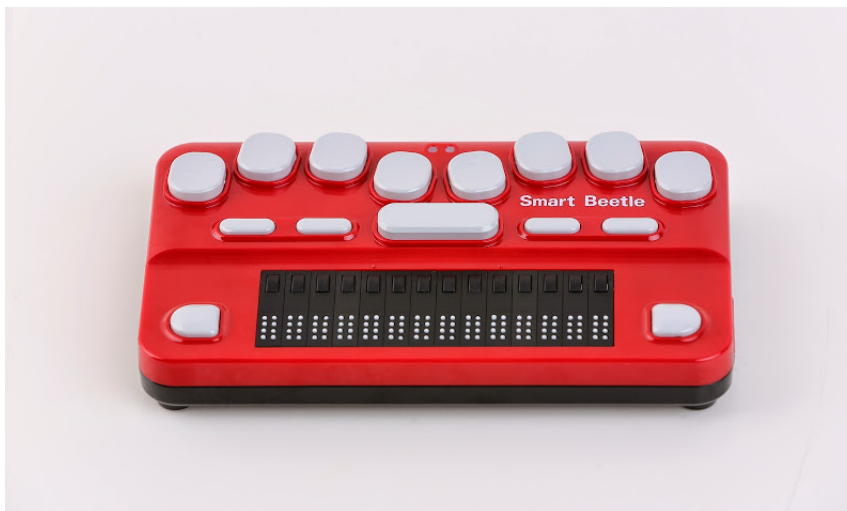
O leitor de tela é um artifício tecnológico que permite que textos em tela possam ser oralizados (BALANSIN, 2011, p.4), ou seja, convertem texto em voz, e essa conversão é denominada TSC(Tex-to-Speech Conversion) (CARVALHO et al., 2001, p.106).

Ainda segundo (BALANSIN, 2011, p.4-5), é essa tecnologia que integra deficientes visuais a mídias, computadores e a Internet. Entretanto, um dos maiores obstáculos dos leitores de telas são as imagens, pois, para que está sejam identificadas e “lidas” pelo *software* leitor de tela as imagens devem ser anexadas juntamente com sua descrição, recurso que é muito utilizado apenas em páginas web.

Outra característica dessa tecnologia é que ela deve promover a navegação pelo conteúdo exibido em tela de modo que o usuário cego tenha as mesmas possibilidades de iteração que um usuário vidente (BALANSIN, 2011, p.5).

Uma novidade que vem auxiliando a leitura de telas é equipamento eletrônico Linha Braille (Figura 2.8), que trabalha em sincronia com um *software* leitor de tela. Primeiramente o conteúdo da página é lido, depois ocorre a transcrição e por fim o teor da tela é representado em uma linha tátil repleta de células braille, cujos pinos se movimentam para cima e para baixo representando o sistema braille (BRAIL,CURSO EMAG, 2019).

Figura 2.8 – Linha Braille.



Fonte: Tecnovisão (2019)

2.5.2 Reconhecimento de Voz

Softwares de reconhecimento de voz permitem que deficientes visuais substituam comandos via teclado e/ou mouse por comandos de voz. Tais softwares são complexos, pois, reconhecem pequenas unidades de som chamadas fonemas, as analisa e tenta contextualiza-las montando frases coesas. As operações podem ser ajustadas de modo a realizarem uma grande variedade de comandos por voz. Tais tecnologias contam também com ajustes que se adéqua a comandos de múltiplos usuários, porém com menor segurança (CARVALHO et al., 2001, p.108).

2.5.3 Transcrição de Caracteres Alfanuméricos para Braille

Os *softwares* que realizam a transcrição de caracteres alfanuméricos para braille são os responsáveis pela produção de grande volume de material impresso tátil, pois são eles os responsáveis por viabilizar que um caractere alfanumérico se torne uma ou mais células braille. Logo, conjuntamente com impressoras braille eles dão a possibilidade gerar não só um documento, mas vários documentos com apenas uma transcrição, atingindo assim um maior número de deficientes visuais.

O *software* proposto neste trabalho, o MatBraille, é um transcritor de caracteres alfanuméricos para o Sistem Braille e trabalhos que desempenham uma função semelhante são descritos no Capítulo 4.

2.5.4 Optical Braille Recognition

Outro tipo de tecnologia que foi desenvolvida é *Optical Braille Recognition* que realiza a transcrição do texto braille para caracteres alfanuméricos. Tais *softwares* são integradas a alguns

tipos de *scanners* e pode transcrever até folhas impressas dos dois lados (BEZERRA et al., 2003, p.15).

2.5.5 Livros digitais

Outra importante fonte de propagação de conhecimento usado por deficientes visuais são os *e-books* ou livros digitais, que podem ser distribuídos de forma gratuita ou não. O formato que são distribuídos garantem a possibilidade de serem lidos por qualquer dispositivo, como é o caso dos formatos abertos DAISY² e ePUB (do inglês *Electronic Publication*), já os ditos como de formato proprietário, como por exemplo, MOBI da Mobipocket (2019) e AZW, Kindle File Format da Amazon Kindle (2019), são configurados para serem lidos por um leitor eletrônico específico.

2.5.6 Acessibilidade na web

O Art. 63 da Lei nº 13.146/2015 afirma:

É obrigatória a acessibilidade nos sítios da internet mantidos por empresas com sede ou representação comercial no País ou por órgãos de governo, para uso da pessoa com deficiência, garantindo-lhe acesso às informações disponíveis, conforme as melhores práticas e diretrizes de acessibilidade adotadas internacionalmente.

A linguagem de marcação HTML (*Hypertext Markup Language*), que é usualmente aplicada na construção de sites, uma vez navegados conjuntamente com leitores de telas como JAWS (do inglês *Job Access With Speech*) da Tecassistiva (2019), Window-Eyes da Microsoft (2019), ORCA da GNOME (2019) e o NVDA³ (do inglês *Non Visual Desktop Access*) proporcionam o acesso ao conteúdo codificado no site.

O MathML⁴ (*Mathematical Markup Language*) é uma linguagem de marcação fundamentada em XML (*Extensible Markup Language*) que é uma linguagem recomendada para codificação de marcações de documentos acessíveis pela W3C (*World Wide Web Consortium*), que é a principal organização de padronização de conteúdos *World Wide Web*. Voltada para a transcrição de notações matemáticas principalmente em páginas *World Wide Web*, o MathML foi desenvolvido em abril de 1998 e também é recomendada pela W3C. A partir de 2015 é parte do HTML5 e ISO ISO/IEC DIS 40314.

JavaScript, que é também conhecido como JS, se trata de uma linguagem de programação interpretada ou compilada *just-in-time*, dinâmica, *prototype-based* e multi-paradigma (MOZILLA, 2021). Responsável pela interatividade na Web conjuntamente com o HTML (*Hypertext Markup*

² DAISY Consortium disponível em: <www.daisy.org>

³ Disponível em: <<http://www.nvda.pt/pt-pt/downloads>>

⁴ Disponível em: <<https://www.w3.org/Math/>>

Language ou em português Linguagem de Marcação de Hipertexto) e o CSS (*Cascading Style Sheets* ou Folhas de Estilo em Cascata) formam a principal base das tecnologias da *World Wide Web*.

2.6 Deficiência Visual no Brasil

O sistema braille, criado a quase 200 anos, por Louis Braille, foi disseminado no Brasil através de José Álvares de Azevedo, que nasceu cego e aos dez anos de idade foi estudar em Paris. em sua vivência de seis anos em Paris, José Álvares aprendeu o sistema Braille. Também teve parte de seus estudos no então Imperial Instituto de Meninos Cegos, hoje conhecido como Instituto Benjamin Constant (IBC). E nos tempos atuais o Dia Nacional do Braille é comemorado dia 8 de abril, dia do nascimento de José Álvares de Azevedo, primeiro professor cego do Brasil.

Outro marco da história dos cegos no Brasil foio ano de 1946, pois além da inauguração do Curso Ginásial do Instituto Benjamin Constant, segundo [CBO \(2019\)](#):

Também em 1946, no dia 11 de março, a jovem cega Dorina de Gouvêa, juntamente com um grupo de amigas, deu início às atividades da Fundação para o Livro do Cego no Brasil, hoje Fundação Dorina Nowill para Cegos, com o objetivo de produzir livros em braille e atuar nas áreas de educação e reabilitação de pessoas com deficiência visual.

Segundo a [Fundação Dorina Novill \(2019\)](#), que considerada uma referência sobre o assunto, 23,9% (45,6 milhões de brasileiros) declararam ter algum tipo de deficiência. Entre as deficiências declaradas, a com maiores índices é a visual, atingindo 3,5% da população brasileira. Tendo como principais causas de cegueira no Brasil são: catarata, glaucoma, retinopatia diabética, cegueira infantil e degeneração macular.

Segundo dados do IBGE de 2010 ([IBGE, 2010](#)), dos mais de 6,5 milhões de brasileiros com alguma deficiência visual: 528.624 pessoas são incapazes de enxergar, os ditos como cegos; 6.056.654 pessoas possuem baixa visão ou visão subnormal, ou seja, grande e permanente dificuldade de enxergar, e outros 29 milhões de pessoas declararam possuir alguma dificuldade permanente de enxergar, ainda que façam uso de óculos ou lentes de contato.

A seguir são apresentadas algumas leis de acordo com ([BRASIL, 2019](#)) que amparam o deficiente visual no Brasil:

- LEI Nº 4.169, DE 4 DE DEZEMBRO DE 1962 - Oficializa as convenções Braille para uso na escrita e leitura dos cegos e o Código de Contrações e Abreviaturas Braille
- LEI Nº 10.753, DE 30 DE OUTUBRO DE 2003 - Institui a Política Nacional do Livro.
- LEI Nº 11.126, DE 27 DE JUNHO DE 2005 - Dispõe sobre o direito do portador de deficiência visual de ingressar e permanecer em ambientes de uso coletivo acompanhado de cão-guia.

- LEI N° 11.982, DE 16 DE JULHO DE 2009 - Acrescenta parágrafo único ao art. 4º da Lei N° 10.098, de 19 de dezembro de 2000, para determinar a adaptação de parte dos brinquedos e equipamentos dos parques de diversões às necessidades das pessoas com deficiência ou com mobilidade reduzida.

3 Análise Léxica

Um compilador é um programa que pode ler um programa em uma linguagem origem e traduzir para um programa semanticamente equivalente em outra linguagem (AHO et al., 2006). A necessidade de citarmos a definição de um compilador consiste no fato da Análise Léxica ser a primeira etapa do processo de compilação. Entretanto, para uma real compreensão do tema alvo, não basta apenas a definição acima, é necessário adentrar um pouco mais no mundo das Linguagens Formais e Autômatos.

Assim, todo esse Capítulo se faz necessário, para descrever e entender toda a computação por trás da aplicação desenvolvida. Iniciamos então, na Seção 3.1, trazendo definições mais gerais, mas que necessitam ser entendidas, para que se entenda as seções posteriores. Na seção sucessora, a Seção 3.2, é realizada uma contextualização a respeito do tema Compiladores. Já na Seção 3.3, são dadas noções a respeito de Autômatos Finitos. Em seguida, na seção Seção 3.4 é discutido a respeito das Linguagens Regulares, que é sucedida pela Seção 3.5 que apresenta a definição de Expressões Regulares. Por fim, mas não menos importante a Seção 3.6 que relata a respeito do Gerador Léxico.

3.1 Definições preliminares

Símbolo: é uma marca, um sinal ou palavra que é entendido como uma representação de uma ideia, objeto ou relacionamento. Exemplo: Símbolos: {1, 5, +, w}.

Alfabeto: Um alfabeto, Σ , é qualquer conjunto finito de símbolos distintos. (VIEIRA, 2006). Exemplo: $\Sigma = \{w, h, i, l, e, +, 1, 2, 3\}$.

String, palavra ou sentença: é a sequência ou cadeia, finita de símbolos pertencentes a um alfabeto Σ ligados pela operação de concatenação (BEZERRA et al., 2003).

Linguagem: qualquer conjunto de palavras.

União: assim como na teoria de conjuntos a união de linguagens é a linguagem que contém todas as palavras de ambas (APPEL; PALSBERG, 2002).

Concatenação: a concatenação de duas palavras ou símbolos é a junção destes para formar uma nova sequência de símbolos. A concatenação de duas linguagens é a linguagem que representa todas as possíveis concatenações entre suas palavras, respeitando a ordem aplicada à operação.

Fecho de Kleene: O fecho de Kleene de uma linguagem L é dado por $L^* = \bigcup_{n \in \mathbb{N}} L^n$, ou seja, zero ou mais concatenações da linguagem L consigo mesma.

Tokens: são uma classificação abstrata de unidades léxicas para auxiliar a análise sintática.

3.2 Breve contextualização a respeito de Compiladores

Antes de abordar o conceito de Análise Léxica se faz necessário falar de Compiladores, afinal a Análise Léxica é uma etapa do Sistema de Compilação. (GUIMARAES, 2007), define um compilador como “um programa que lê um programa escrito em uma linguagem L1 e o traduz para uma outra linguagem L2. Usualmente, L1 é uma linguagem de alto nível como C++ ou Prolog e L2 é assembler ou linguagem de máquina.”

A seguir são apresentados os cinco passos que compõem o processo de compilação de uma linguagem, contendo como primeiro passo a Análise Sintática, seguido pela Análise Léxica, Análise Semântica, Geração de código e Otimização do código.

Análise Léxica: também denominada como leitura ou scanning, se trata da primeira fase do processo de compilação. Sua principal função é identificar os símbolos que compõem o programa e verificar se o caractere lido pertence ao alfabeto da linguagem. A verificação de pertinência ou não é realizada com o auxílio de Autômatos Finitos Determinísticos ou por Autômatos Finitos Não Determinístico. Para tal, o analisador léxico divide o código fonte em símbolos léxicos, que serão usado na fase posterior da compilação, um exemplo da separa de tokens pode ser visualizada na Tabela 3.1. Tais símbolos léxicos possuem como formas mais comuns:

- identificadores: usados para nomear entidades como variáveis,, funções, métodos, classes, módulos;
- literais: que podem ser caracteres que indicam uma constante, ou número, um caractere, uma *string*, um booleano (verdadeiro ou falso).
- palavras chaves: palavras, normalmente reservadas e não utilizáveis que são adequadas para expressar estruturas da linguagem;
- sinais de pontuação: funcionam como auxiliares da estrutura do programa.

Tabela 3.1 – Tabela de definição de Tokens

Tipo	Eexemplo
ID	NF0012
NUM	73 000 264
REAL	65.3 1e67 5.3e-10
IF	if
NOTEQ	!=
LPAREN	(
RPAREN)

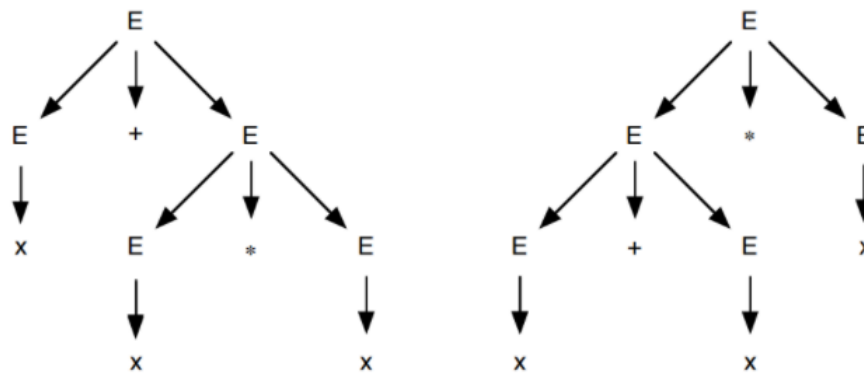
Nesta fase, também são eliminados todos os espaços em brancos, tabulações, símbolos de mudança de linha e comentários.

A etapa de análise léxica foi a única etapa de um processo de compilação que foi realmente usada na implementação da aplicação aqui proposta. Ou seja, após a entrada dos caracteres via teclado, ocorre a análise léxica de cada carácter identificando-o e verificando sua pertinência dentro do alfabeto da linguagem.

Análise Sintática: também conhecido como *parser*, identifica os símbolos lidos na etapa da Análise Léxica e os relacionam, validando-os em relação a gramática do código fonte. Por exemplo, suponha uma gramática G e uma sentença s , o analisador sintático recebe a sequência de *tokens* que fazem parte da sentença s e ou monta uma árvore de derivação consultando a gramática da linguagem, no caso da sentença s ser inválida e não correspondência na gramática um erro é sinalizado.

A árvore sintática, tem como objetivo representar a hierarquia do programa fonte. A estrutura da árvore é formada de modo que cada nó é composto por um registro que contém vários campos, e estes campos que podem ser usados nas fases posteriores do processo de compilação.

Figura 3.1 – Exemplos de árvore sintática



Fonte: (APPEL; PALSBERG, 2002)

Análise Semântica: uma vez que já detectamos a estrutura do programa a análise sintática tenta entender a essência do programa, ou seja, detecta se o programa está coerente com as regras semânticas específicas da linguagem de programação. Nesta fase também ocorre a eliminação de ambiguidades dos termos do programa. Em linguagem de programação de tipo estáticos, esta etapa também cuida da geração e manutenção da tabela de símbolos, que é usada como auxiliadora no acompanhamento da definição dos nomes em um programa, ou seja, possibilita a checagem de tipos em expressões e declarações. (FOLEISS et al., 2009) cita as seguintes verificações como mais usuais: "verificação de compatibilidade de parâmetros formais e atuais

(tipos e quantidade), compatibilidade de tipos em instruções de atribuição, determinação de expressões constantes, verificação de declaração antes do uso e regras de escopo."

Geração de código intermediário: produz a estrutura na representação intermediária similar ao código final, sendo algumas omitida das fases do processo de compilação.

Otimização do código intermediário: simplifica as estruturas intermediárias geradas, gerando uma abstração do código intermediário e bem como seu antecessor pode ser omitido das fases da compilação.

Geração de código objeto: produz a estrutura da linguagem objeto, ou seja, gera um código de baixo nível

Otimização de código objeto: melhora a estrutura produzida no passo anterior.

3.3 Autômatos Finitos

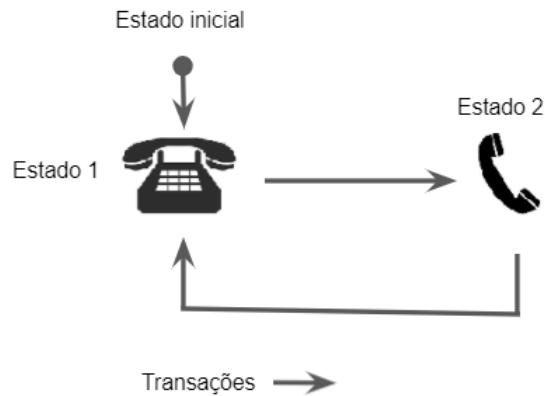
Autômato Finitos: consiste em um modelo matemático de um sistema com entradas e saídas discretas. Cada entrada ocasiona uma mudança de estado, também denominada como transição. Podendo o sistema estar em qualquer uma de suas configurações internas ou estados, tais estados guardam informações relevantes sobre o histórico do sistema.

De um modo mais formal podemos definir um autômato finito M como uma quintupla $M = (Q, \Sigma, \delta, s, F)$, onde:

- Q : conjunto de estados
- Σ : alfabeto de entrada
- δ : função de transição (define os possíveis estados do passo seguinte)
- $s \in Q$: estado inicial
- $F \subseteq Q$: conjunto e estados finais

Fazendo uma analogia a um telefone fixo podemos definir o seguinte: os dois estados possíveis seria o estado 1 em que o telefone está ocioso e encaixado no gancho, já o estado 2 seria o telefone em uso, assim definimos como estado inicial e final o estado 1 e por fim as transições seriam as ações de tirar e colocar o telefone no gancho. Tal esquema pode ser observado na Figura 3.6.

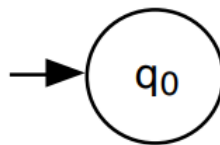
Figura 3.2 – Exemplo de máquina de estados.



Entretanto, uma real representação de um autômato pode ser vista na Figura 3.3. Detalhando cada parte do automato finito temos o seguinte padrão:

- estado inicial representado na Figura 3.3

Figura 3.3 – Estado inicial de um autômato



- estado final representado na Figura 3.4

Figura 3.4 – Estado final de um autômato



- transições paralelas representadas na Figura 3.5, onde a seta indica o estado destino, já os valores "0,1" indicam que caso o próximo *token* lido seja 0 ou 1 essa transição pode ser feita, esse "caminho" pode ser escolhido

Figura 3.5 – Representação de transições em um autômato

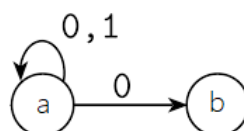
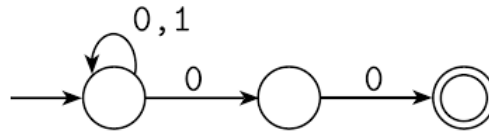


Figura 3.6 – Exemplo de Autômato Finito.

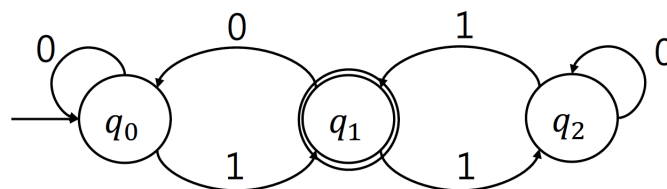


Vale destacar que as transições podem ter sentido único ou duplo então, a ponta da seta é usada para indicar o estado destino. Então, no caso da Figura 3.5, é possível sair do estado a e alcançar o estado b , mas não é permitido transitar do estado b para o estado a . Outra importante observação é que o que determina a chegada no estado final é a leitura de cada símbolo, assim, dizemos que uma palavra é aceita por um autônomo se todos os símbolos dessa palavra foram lidos e o estado atual é igual a um estado final, ou seja, a última transição de leitura de símbolo levou a um estado final.

Autômato Finito Determinístico (AFD): Em um autômato finito determinístico, duas arestas que saem do mesmo estado não são rotuladas com o mesmo símbolo, ou seja, suponha que o próximo caractere a ser lido é o caractere a , no caso de um AFD não podem existir dois "caminhos" que façam leitura do *token* a e leve a outro estado. Assim, dizemos que a linguagem reconhecida por um autômato é o conjunto de *strings* que ele aceita. Um AFD aceita ou rejeita uma sequência através dos seguintes passos:

1. Começando no estado inicial, para cada caractere na *string* de entrada, o autômato segue exatamente uma aresta para chegar ao próximo estado.
2. Depois de fazer n transições para uma *string* de n caracteres, se o autômato estiver em um estado final, ele aceita a *string*. Se não estiver em um estado final, ou se em algum ponto não houver nenhuma aresta devidamente rotulada a seguir, ela rejeita.

Figura 3.7 – Exemplo de Autômato Finito Determinístico.

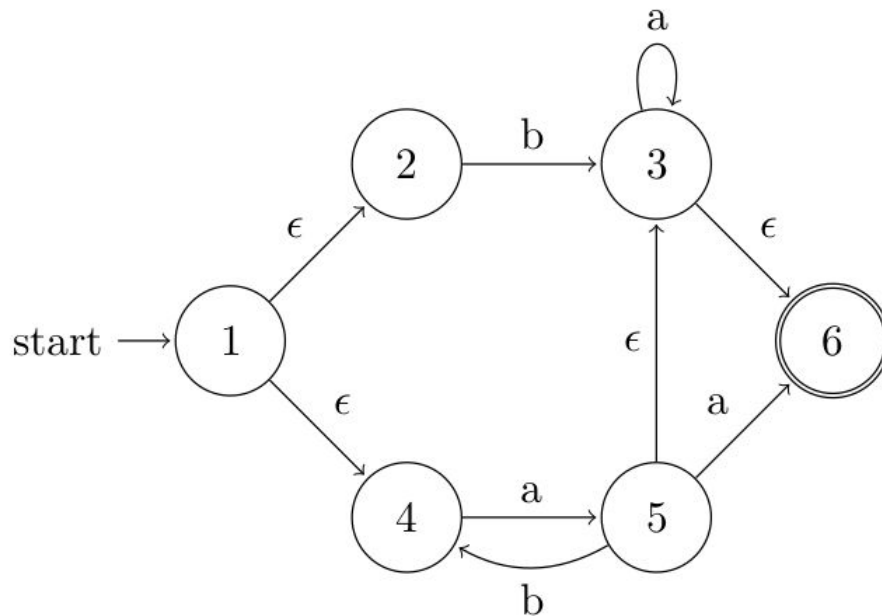


Fonte: Autômato... (a)

Atômato Finito Não Determinístico (AFND): Um autômato finito não determinístico é aquele que tem uma escolha de arestas - rotuladas com o mesmo símbolo - para sair de um estado.

Podendo também apresentar bordas especiais marcadas com (a letra grega épsilon) que podem ser seguidas sem ler nenhum símbolo da entrada. As Figuras 3.3 e Figura 3.8 são exemplos AFND's.

Figura 3.8 – Exemplo de Autômato Finito Não Determinístico.



Fonte: Autômato... (b)

Tabela de transição: é a representação tabular do diagrama de transição, podendo representar um AFD ou um AFND. Um exemplo de representação de uma tabela de transição pode ser visto na Figura 3.9

Figura 3.9 – Exemplo de Tabela de transição.

	0	1
→ e ₁	e ₂	e ₃
* e ₂	e ₂	e ₃
e ₃	e ₂	e ₃

Em maioria, as informações contidas nesta seção são baseadas no trabalho de (APPEL; PALSBERG, 2002).

3.4 Linguagem Regular

Podemos definir uma linguagem como um conjunto de palavras sobre um alfabeto Σ . Alguns exemplos de linguagens:

- o conjunto vazio \emptyset ;
- o conjunto unitário contendo apenas um caractere a ;
- o conjunto de números binários de 2 dígitos $\{00, 01, 10, 11\}$;
- o conjunto de todos os programas Apex sintaticamente corretos.

Uma vez que podemos aplicar operações sobre conjuntos, podemos também aplicar operações sobre linguagens, como por exemplo, a união entre uma linguagem M e uma segunda linguagem N , pode ser representada como $M \cup N$. Entretanto, quando o assunto é análise léxica as principais aplicações sobre o conjunto de linguagens são as operações de alternância, concatenação e Fecho Kleen, pois fazendo uso dessas operações conjuntamente com o uso de épsilon e símbolos podemos especificar o conjunto de caracteres correspondentes aos *tokens* da linguagem de programação. (APPEL; PALSBERG, 2002)

Uma **Linguagem Regular** é aquela que pode ser escrita fazendo uso de um dos seguintes itens: a Expressão Regular (ER), um Autômato Finito Determinístico (AFD), Autômato Finito Não Determinístico (AFND) ou Tabela de transição (DOGNINI; RAABE, 2003).

Podendo também ser expressa como um subconjunto de todas as *strings*. Sua empregabilidade é reconhecer padrões de dados e agrupar problemas computacionais, facilitando assim na abordagem de resolução destes problemas que foram agrupados.

3.5 Expressões regulares

Primeiramente considere as seguintes definições:

Repetição: Dada uma expressão regular M , seu fechamento Kleene é M^* . Uma *string* está em M^* se for a concatenação de zero ou mais strings, todas em M . Assim, $((a\ b) \cdot a)^*$ representa o conjunto infinito, "aa", "ba", "aaaa", "baaa", "aaba", "baba", "aaaaaa", (APPEL; PALSBERG, 2002).

Épsilon ou palavra: A expressão regular representa uma linguagem cuja única *string* é a *string* vazia. Assim, $(a \cdot b)$ representa o idioma $\{\epsilon, "ab"\}$ (APPEL; PALSBERG, 2002).

Agora considere uma linguagem contendo todos os números pares e uma linguagem contendo todas as letras do alfabeto, a respeito da primeira linguagem não é possível determinar sua finitude, ou total de strings que a compõem, já a linguagem formada por todas as letras

do alfabeto pode ser descrita como linguagem com descrição finita. E para esse segundo caso, usamos a notação de expressões, dado que cada expressão representa um conjunto de strings.

Com a junção de símbolos de alternância, concatenação, épsilon e fecho de Kleene é possível apontar o conjunto de caracteres ASCII correspondentes ao *tokens* lexicais de uma linguagem de programação. Observe a seguir alguns exemplos de expressões regulares:

- $(0|1)^*$ → Números binários que são múltiplos de dois. Exemplo de binários múltiplos de dois: 100,110.
- $\mathbf{b^*(abb^*)^*(a|\epsilon)}$ → Strings de a's e b's sem a's consecutivos. Exemplos de strings com o padrão descrito {babbba,bababb}.
- $\mathbf{(a|b)^*aa(a|b)^*}$ → Strings de a's e b's contendo a's consecutivos. Exemplos de strings com o padrão descrito {aaaabbabbaaa,baaabb}.

Observe nas expressões regulares acima que os símbolos de concatenação e o épsilon são omitidos, assim 100 tem o mesmo valor que $(1 \cdot 0) \cdot 0$, da mesma forma podemos ocultar épsilon assim $(a|)$ tem o mesmo valor que $(a|)$, ou seja, a ou vazio. Algumas outras abreviações são possíveis como por exemplo:

- $[abcd]$ é o mesmo que $(a | b | c | d)$;
- $[b - g]$ é o mesmo que $[bcdefg]$;
- $[b - gM - Qkr]$ é o mesmo que $[bcdfgMNO PQkr]$;
- $M?$ é o mesmo que $(M|\epsilon)$;
- $M+$ é o mesmo que $(M \cdot M^*)$;

Essas extensões são funcionais, mas nenhuma aumenta o poder descritivo das expressões regulares, de modo que qualquer conjunto de strings que pode ser descrito com essas abreviações acima, também pode ser escrito apenas pelo conjunto básico de operadores. Todos os operadores estão resumidos na Tabela 3.2.

Tabela 3.2 – Tabela de Integrais

Operação	Significado
a	Uma representação de si mesmo
ϵ	A string vazia
$M N$	Alternância, escolhendo entre M ou N.
$M \cdot N$	Concatenação, um M seguido por um N.
MN	Outra maneira de escrever a concatenação
M^*	Repetição (zero ou mais vezes)
M^+	Repetição, uma ou mais vezes.
[a-zA-Z]	Alternância de conjunto de caracteres
.	Um ponto representa qualquer caractere único, exceto nova linha
M?	Opcional, zero ou uma ocorrência de M
"a.+*"	Aspas, uma string entre aspas representa a si mesma literalmente.

Usando esta linguagem, podemos especificar os *tokens* lexicais de uma linguagem de programação, exemplo pode ser observado na Tabela 3.3

Tabela 3.3 – Tokens de uma linguagem de programação

If	If
[a-z][a-z0-9]	Id
[0-9]^+	Num
([0-9]^+ "." [0-9]^*) ([0-9]^* "." [0,9]^+)	Real
(-- "[a-z]^*" "\n") ("\n "\t")^+	não é um token, apenas espaço em branco
.	erro

Analisemos, agora a seguinte *string If8* como ela será classificada? Pelo *tokens If* e pelo *token* numérico ou como um identificador? Nessas situações ocorre o que chamamos de ambiguidade e para corrigir-la necessitamos empregar uma regra de desambiguação, abaixo são enunciadas as duas formas mais comuns de resolver o problema.

Correspondência mais longa: a substring inicial mais longa da entrada que pode corresponder a qualquer expressão regular é considerada a próxima símbolo.

Prioridade da regra: para uma substring inicial mais longa específica, a primeira expressão regular que pode corresponder determina seu tipo de token. Isso significa que a ordem de escrita das regras de expressão regular é significativa

Logo, a *string If8* será tratada como um identificador pela regra de correspondência mais longa e como o *tokens If* quando levamos em consideração a correspondência por prioridade de regra.

Toda essa seção é fortemente amparada pelo Capítulo de Expressões Regulares do livro (APPEL; PALSBERG, 2002).

Para realização desta etapa na implementação da aplicação, se fez necessário a consulta

as normas de escrita em braille nos documentos Sistema Grafia Braille para a Língua Portuguesa e Código Matemático Unificado para a Língua Portuguesa. De modo que são as expressões regulares as responsáveis, por exemplo, pela transcrição correta e diferente entre si de palavras iniciadas com letras maiúsculas e palavras contendo todas as letras em maiúsculo.

3.6 Gerador Léxico

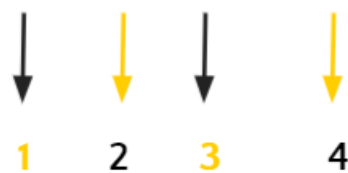
Tomando o alfabeto grego como exemplo de uma linguagem e não como um alfabeto, e também um símbolo qualquer, podemos classificar este símbolo como parte ou não deste idioma. Seguindo este raciocínio, a *string* não precisa ter significado algum, surgindo daí a ideia da realização do processo de análise léxica de forma automática. Desta forma, se faz necessário a especificação das expressões regulares e o restante da análise léxica pode ocorrer de forma automática por um gerador léxico (AHO et al., 2006).

O Lex ou sua versão mais recente o Flex, é usado para implementar um analisador léxico. A notação de entrada do Flex é conhecida como linguagem Lex e usa o compilador Lex, transformando os caracteres de entrada em *tokens*, assim, podemos dizer que ele lê caractere por caractere e retorna seu *token* correspondente e eliminando espaços vazios (AHO et al., 2006).

Uma forma amigável de visualizar como funciona um gerador léxico pode ser observado nas Figuras 3.10 e 3.11, onde se vê em ambos os casos a classificação correta do lexema, entretanto, sem se importar com a semântica da linguagem.

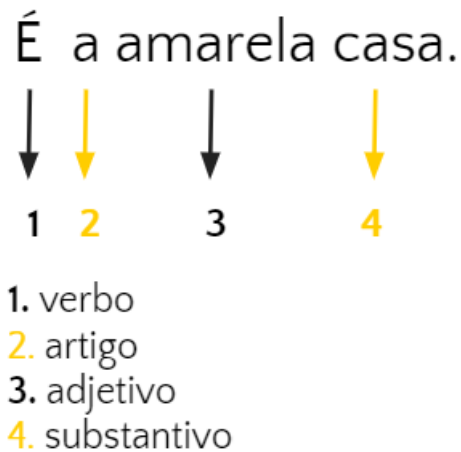
Figura 3.10 – Exemplo de análise léxica

A casa é amarela.



1. artigo
2. substantivo
3. verbo
4. adjetivo

Figura 3.11 – Exemplo de análise léxica



Assim, para a realização da análise léxica necessária para este trabalho foi usado o Flex¹, que é uma biblioteca gratuita para JavaScript. Esta biblioteca tem como base FLEX, desta forma a ideia, regras gerais e lógica por trás do Flex é a mesma do Flex, entretanto, algumas diferenças são notáveis, como por exemplo, FLEX é um gerador léxico, mas Flex.JS é uma classe lexer configurável.

¹ Flex.js disponível em: <<https://github.com/sormy/flex-js>>

4 Trabalhos Relacionados

Com o intuito de fazer um levantamento do estado da arte sobre os trabalhos relacionados a proposta desta monografia foi realizada as etapas iniciais de uma revisão sistemática da literatura, com o auxílio da *software StArt*¹. As pesquisas por *softwares* que proporcionavam a entrada, leitura e/ou transcrição de textos matemáticos de ensino superior foram realizadas nos bancos de dados *Google Academic* e CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior), onde foram encontrados 629 trabalhos relacionados. Foram incluídos também, porém, de forma manual, 8 artigos e excluídos da busca todos os trabalhos em que:

- os títulos que ressaltam trabalhos com gráficos, educação infantil, métodos pedagógicos, aplicativos móveis ou que se distanciavam do foco de pesquisa;
- trabalhos publicados em idiomas distintos de português ou inglês;
- *softwares* pagos;
- inclusão no ensino a distância;
- aplicações que fazem uso de hardwares táteis;
- recursos não tecnológicos de ensino;
- aplicações que auxiliam no dia-a-dia;
- trabalhos que não estão inteiramente disponíveis de forma gratuita na web;
- casos isolados de pesquisa (Exemplo: educação inclusiva no instituto X ou localidade Y);
- tecnologias reversas (Ex: braille para tinta, braille para português);
- trabalhos relacionados a alfabetização em braille;
- documentos não catalogados entre os anos de 2009 e 2021.

Após as etapas de exclusão por título e leitura dos resumos foram refinados 38 trabalhos para leitura. A seguir são apresentadas as tecnologias assistivas capturadas nos artigos aceitos, bem como uma apresentação de suas funcionalidades:

¹ A ferramenta StArt (*State of the Art through Systematic*) foi desenvolvida pela Universidade Federal de São Carlos, tem como objetivo auxiliar o pesquisador em todas as fases de uma revisão sistemática. A ferramenta StArt é um software gratuito e está disponível em [UFSCar \(2019\)](#)

TalkMaths é uma ferramenta que transforma matemática falada em fórmulas matemáticas (Exemplo: fala “dois mais 1” sendo visualizado no editor “ $2+1$ ”), sendo possível a exportação do documento final em formato LaTeX². Se trata de uma ferramenta gratuita, entretanto com suporte apenas ao idioma inglês (TALKMATHS, 2015).

ChattyInfty é uma ferramenta que produz conteúdo Daisy e/ou ePUB acessível. Sendo possível a entrada e conversão de conteúdos em inglês, japonês, grego, italiano, turco, língua tcheca e língua vietnamita (SCIENCE ACCESSIBILITY NET, 2019)

Lambda é um código matemático derivado do *MathML* desenvolvido com o intuito de ser usado conjuntamente com periféricos braille e de síntese de voz. O editor permite uma conversão em tempo real, porém como pode ser visualizado em sua página inicial na web, são necessários alguns conhecimentos de ciência da computação. Outro contraponto é o fato de usar um sistema braille de 8 pontos (usualmente são usados seis pontos). Seu suporte são para braille italiano, espanhol e inglês (LAMBDA, 2019).

Emacspeak é uma ferramenta de fala que permite um deficiente visual desempenhe tarefas no computador de forma autônoma. Disponível gratuitamente na internet Emacspeak é compatível com os sistemas operacionais Linux³ (EMACSPEAK, 2019).

AsTeR (do inglês *Audio System For Technical Readings*, Sistema de áudio para leituras técnicas em português) como o próprio nome indica é um leitor de documentos técnicos, atuando como um livro falante. É cedido pela RFB (do inglês *Recordings for the Blind*⁴, traduzido como Gravações para cegos) e faz leitura de algumas expressões matemáticas escritas inicialmente na forma de LaTeX (ASTER, 1994).

MathPlayer *Design Science MathPlayer*TM realiza leitura de matemática de forma que a matemática seja verbalizada em produtos de tecnologia assistiva. Com suporte para MathML o MathPlayer é fornecido gratuitamente e é compatível com sistemas Windows e Mac⁵ (DESIGN SCIENCE - HOW SCIENCE COMMUNICATES, 2018a).

ReadHear PC é um leitor de textos no formato DAISY, dentre outros formatos acessíveis. Este software pode ser baixado de forma gratuita para dispositivos com sistemas operacionais Windows 98/Me/NT/2000/XP/2003/Vista/7 (DESIGN SCIENCE - HOW SCIENCE COMMUNICATES, 2018b).

MecDaisy desenvolvido em 2009 pela Universidade Federal do Rio de Janeiro, o MecDaisy compreende em um conjunto de softwares que proporciona o desenvolvimento de livros em formato digital acessível no formato Daisy, além de reproduzir áudios neste formato.

² Disponível em: <<https://www.latex-project.org/>>

³ <<https://www.linux.org/>>

⁴ Disponível em: <<http://www.cs.cornell.edu/Info/People/raman/rfb-math-workshop/summary.html>>

⁵ Disponível em: <https://www.apple.com/br/mac/?afid=p238%7CsOEUd0tGq-dc_mtid_1870765e38482_pcrd_362275086478_pgrid_42688348305_&cid=aos-br-kwgo-mac--slid---mac-e-product->

Para o manuseio e confecção de documentos não é necessário noções de programação e se trata de uma aplicação gratuita (NCE/UFRJ - PROJETO DOSVOX, 2002d).

Openbook é um conversor que transforma textos escaneados em áudio MP3, além de permitir ampliação, espaçamento especial entre caracteres ou aplicação de alto contraste (FREEDOM SCIENTIFIC, 2019).

DOSVOX é um sistema compatível com o sistema operacional Microsoft Windows⁶, que além de exibir o conteúdo normal na tela do computador, emprega a síntese de voz em português, permitindo também a configuração para outros idiomas, permitindo assim uma interação entre deficiente visual e as aplicações presentes no computador. Algumas das vantagens do sistema são: grande parte das mensagens emitidas faz uso de voz humana gravada, a gratuidade do software e o fato que o software não só lê o que está escrito na tela, mas estabelece um diálogo com o usuário. O DOSVOX foi desenvolvido em 1993 pela Universidade Federal do Rio de Janeiro (NCE/UFRJ - PROJETO DOSVOX, 2002b).

MATVOX é uma calculadora programável gratuita que funciona através de comandos de voz, auxiliando na implementação de algoritmos e na execução de cálculos matemáticos. Algumas das funcionalidades são: resolução de expressões matemáticas, conversões de unidades, programação estrutural com declaração de variáveis, operadores aritméticos, instruções de condição e repetição, importação e exportação de dados, etc. Uma importante observação é que para sua instalação é necessário a instalação prévia do DOSVOX (NCE/UFRJ - PROJETO DOSVOX, 2002c).

FINANVOX⁷ a Calculadora Financeira FINANVOX surge como um aparato na execução e no desenrolar de cálculos financeiros e estatísticos. Onde a entrada de dados acontece através do teclado do computador e síntese de voz para obtenção de resultados. Assim como ocorre no MATVOX, o FINANVOX depende do DOSVOX pré-instalado. Vale salientar que se trata de um software gratuito.

Biblos se trata de um software que permite a produção de materiais para cegos nos formatos de áudio livro (MP3 e DAISY) e braille impresso, contando também com o recurso de gráficos táteis. Um aspecto relevante sobre esta aplicação gratuita, é a produção que pode ser realizada por videntes e/ou por cegos oferecendo suporte para inglês norte americano, italiano, francs e espanhol (BIBLOS, 2019).

Odt2braille é uma extensão do Braille para os editores de texto gratuitos OpenOffice.orgTMWriter⁸ ou Libre Office⁹. Contando com suporte para o português brasileiro, essa extensão permite exportação e impressão de documentos personalizáveis em braille (OPENOFFICE, 2010). O odt2braille compõem parte de um projeto da AEGIS que tem como

⁶ Disponível em: <<https://www.microsoft.com/pt-br/windows/>>

⁸ Disponível em: <<http://www.openoffice.org/>>

⁹ Disponível em: <<https://www.libreoffice.org/discover/libreoffice/>>

objetivo fornecer acessibilidades a qualquer Tecnologia da informação e comunicação (TIC) de código aberto. Sendo financiado European Commission's Seventh Framework Program (em português: Sétimo Programa-Quadro da Comissão Europeia).

Latex2nemeth se trata de um pacote para LaTeX que converte expressões matemática escritas usando códigos LaTeX em código Nemeth Braille. Quanto ao texto são transcritos materiais escritos em grego ou em inglês ([COMPREHENSIVE TEX ARCHIVE NETWORK, 2019](#)).

MathJax MathJax é uma biblioteca de JavaScript que proporciona a exibição de notações matemáticas durante a navegação de páginas web. O MathJax é gratuito e faz uso de comandos MathML, LaTeX e ASCIIMathML (linguagem que permite a escrita matemática)([MATHJAX, 2019](#)).

Math Speak & Write é um software que possibilita a entrada de expressões matemáticas via voz e via escrita. Está disponível de forma gratuita e foi desenvolvido para ser usado conjuntamente com o Microsoft Word. Suportado apenas para o idioma inglês ([UNIVERSITY OF CALIFORNIA AT BERKELEY, 2019](#)).

InftyEditor é um editor que fornece a possibilidade de inserção de expressões matemáticas via código LaTeX realizando a conversão e exibição em tela do resultado em braille. O documento final pode ser salvo LaTeX , MathML , PDF , Microsoft Word 2007 e Human Readable TeX e tem suporte para braille em inglês e braille japonês ([INFITYPROJECT, 2019](#)).

NatBraille é um software francês gratuito para conversão de texto para braille francês. Se trata de uma aplicação gratuita e que não necessita de conhecimento prévio de braille para a geração de documentos nos formatos braille.

Braille Fácil é um editor de texto gratuito onde a entrada do texto em português é convertida para braille, sem a necessidade de qualquer conhecimento prévio de braille, a conversão ocorre de forma totalmente automática. Permite inclusive, a impressão de fórmulas matemáticas e visualização do texto em português e da saída em braille. ([NCE/UFRJ - PROJETO DOSVOX, 2002a](#)).

Além das tecnologias de síntese de voz e aplicações que devem instaladas nos dispositivos existem transcritores on-line. Foi realizada uma pesquisa em navegadores pelos termos “conversor braille”, “conversor web braille”, “text to braille converter”, “web braille converter”. A Tabela 4.1 apresenta o resultado da busca, descrevendo a plataforma e algumas características.

Tabela 4.1 – Transcritores on-line para Braille

Transcritor on-line	Suporta braille português?	Permite importação de documento?	Transcreve fórmulas matemáticas complexas?
Unicode Braille Maker (UNICODE TOYS, 2016)	Não	Não	Não
Traductor braille (TRADUCTOR BRAILLE, 2019)	Não	Não	Não
Braille converter (BRAILLE CONVERTER, 2019)	Não	Não	Não
Braille Alphabet Decoder and Alphabet (BOXENTRIQ, 2019)	Não	Não	Não
Font Meme (FONT MEME, 2019)	Não	Não	Não
Online Braille Generator (BYRON KNOLL, 2019)	Não	Não	Não
BrailleTranslator.org (BRAILLETRANSLATOR.ORG, 2019)	Não	Não	Não
ATRATOR (ATRATOR, 2019)	Sim	Sim	Não
RoboBraille (ROBOBRAILLE.COM, 2019)	Não	Não	Não
Braille Translator (LINGOJAM, 2019)	Não	Não	Não
Braille Translator (BRANAH.COM, 2019)	Não	Não	Não
Braille Translator (MATHSISFUN.COM, 2016)	Não	Não	Não
Braille Translator (BRAILLE TRANSLATOR, 2019)	Não	Não	Não

A partir da pesquisa realizada, notou-se a ausência da funcionalidade de transcrição de materiais externos pelas ferramentas, em outras palavras, fazer a transcrição de textos escritos em outras extensões. Outras deficiências encontradas foram a escassez de *softwares*/plataformas que fazem a transcrição de fórmulas matemáticas, principalmente as fórmulas que envolvem conteúdos do ensino superior, e a falta de documentação para a realização da tarefa de transcrição em algumas ferramentas.

5 Desenvolvimento

Este trabalho propõem um transcritor de texto tinta em português para Sistema Braille, chamado de MatBRraille, que tem por objetivo fazer a inserção e transcrição de expressões matemáticas de diferentes níveis. A seguir estão demonstrados o passo a passo do desenvolvimento e documentação da ferramenta proposta. Na Seção 5.1 é descrita modelagem da aplicação, na Seção 5.2 são expostos os protótipos das telas da aplicação e por fim na Seção 5.3 há um pouco sobre como se passou a etapa de desenvolvimento da aplicação.

5.1 Modelagem da Aplicação

No momento de desenvolvimento de qualquer aplicação deve-se questionar: Quais as necessidades do público alvo? Como será oferecido tais recursos aos usuários? O modo como se pretende oferecer os recursos atenderão as necessidades dos usuários? Dentre outros questionamentos que a Engenharia de *Softwares* ajuda a compor e documentar de modo que ao fim do desenvolvimento a aplicação esteja de acordo com o esperado e tudo esteja registrado conforme foi construído.

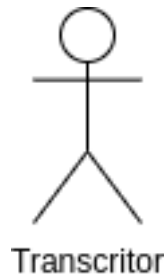
Nesta Seção são expostos alguns recursos de Engenharia de *Softwares* de modo que sejam expostas todas as características da aplicação web proposta, em que parte do desenvolvimento se encontra a aplicação proposta, como serão oferecidos os recursos da aplicação, quais os recursos usados no desenvolvimento da aplicação e qual o público aplicação pretende atender.

5.1.1 Atores

A definição de quem são os atores de uma aplicação consiste em definir qual o papel de cada usuário dentro da aplicação, ou seja, quem fará o quê dentre as funcionalidades oferecidas na aplicação.

Neste trabalho, há apenas um ator, que é a pessoa que digita o texto/documento no transcritor, pois este é o único que faz uso direto da ferramenta proposta. O usuário está representado na Figura 5.1. Ou seja, o deficiente visual não terá contato com aplicação desenvolvida, mas sim quem pretende oferecer algum material tátil a ele.

Figura 5.1 – Diagrama de Hierarquia dos Atores.

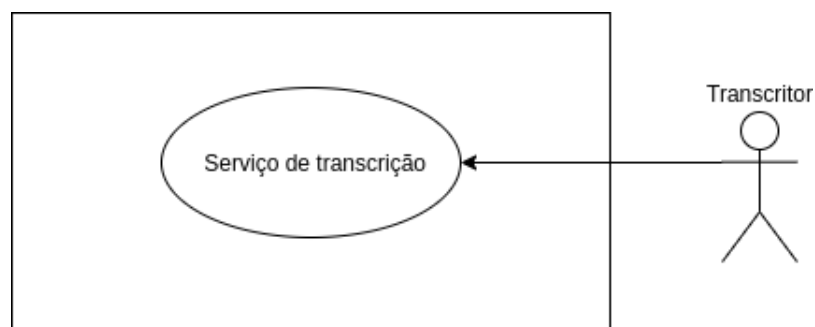


É importante destacar que o ator da aplicação, que está em fase de desenvolvimento, foi nomeado como Transcritor, toda via, vale ressaltar que para transcrição de texto braille existe o cargo de Transcritor de Sistema Braille, que tem, em suma maioria, formação e conhecimentos prévios para exercer tal cargo. Contudo, o objetivo do *software* é possibilitar a conversão correta e de forma facilitada para quaisquer usuários videntes.

5.1.2 Diagrama de Contexto

O Diagrama de Contexto oferece quais as tarefas ou funções serão oferecidas a cada ator do sistema. O serviço oferecido pela aplicação proposta é o Serviço de transcrição, ou seja, é proposta a tarefa de transcrever um conteúdo visual que será ou foi desenvolvido pelo ator usando representação alfanumérica e lhe será oferecido a transcrição do material em braille. A relação do ator e o sistema proposto é representada pelo Diagrama de Contexto ilustrado na Figura 5.2.

Figura 5.2 – Diagrama de Contexto



Dentro da função proposta ao ator, ele poderá realizar algumas tarefas específicas. As principais estão descritas abaixo:

1. Serviço de transcrição
 - a) Digitar documento e conversão semi-automática
 - b) Inserir formulas matemáticas
 - c) Configurar exibição/geração final do documento (Formatações do documento)
 - d) Abrir documento

- e) Salvar documento
- f) Imprimir material tátil produzido

Uma primeira e importante observação é que a plataforma proposta faz a transcrição do material digitado ou importado para a plataforma e inicialmente dá a possibilidade de salvar o documento apenas no formato .txt em versões futuras será disponibilizado o salvamento em outros formatos. O segundo ponto a ser evidenciado é que a impressão em relevo só será possível com uso conjunto de uma impressora braille.

5.1.3 Requisitos Funcionais e Não Funcionais

Os Requisitos Funcionais são requisitos que o *software* deverá realizar ou satisfazer, ou seja, servem para delimitar aquilo que será possível fazer dentro do *software*. Os requisitos funcionais da aplicação proposta são apresentados na Tabela 5.1.

Tabela 5.1 – Requisitos funcionais

RF01	Transcrição de texto escrito em português para braille português, ou seja, aceitando acentuação e letras do alfabeto.
RF02	Escrita de fórmulas matemáticas e transcrição destas de acordo com o Código Matemático Unificado. Português
RF03	Transcrição que diferencie letras maiúsculas de minúsculas.
RF04	Possibilidade de salvar e gerar documento para impressão braille.

Já os requisitos não funcionais são todos os requisitos que devem ser cumpridos para que o *software* funcione, para que o *software* atinga seu propósito. Na Tabela 5.2 estão indicados os Requisitos Não Funcionais da aplicação web proposta.

Tabela 5.2 – Requisitos não funcionais

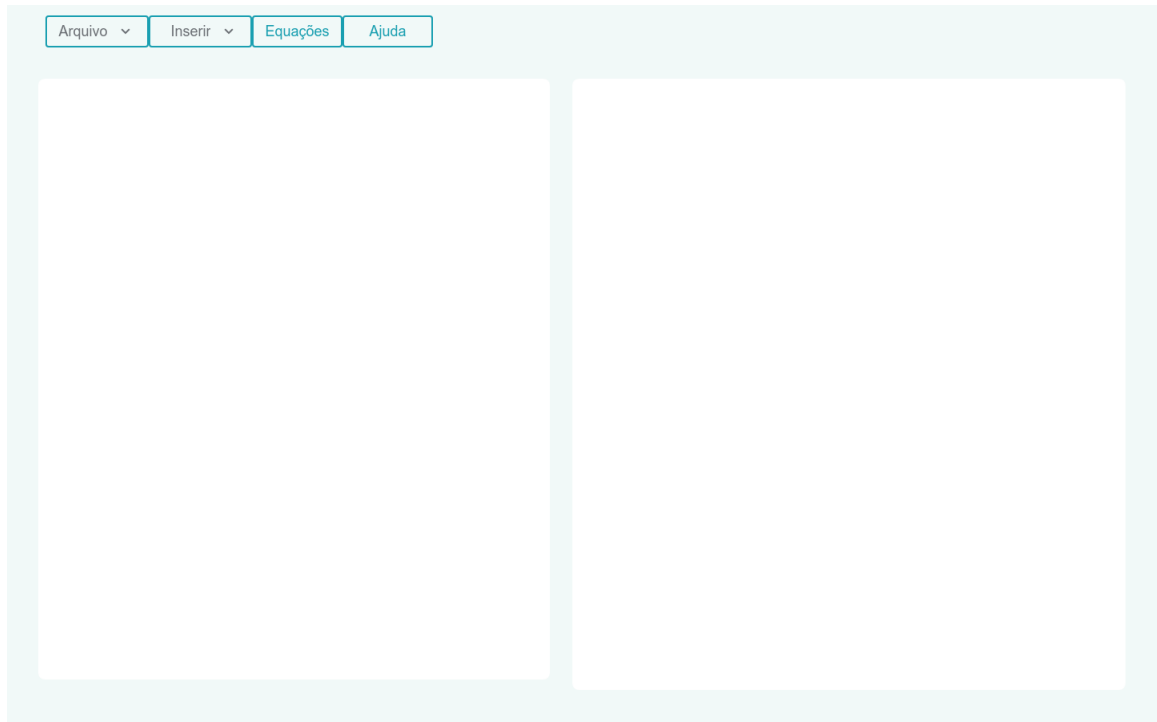
RNF01	O transcritor de braille MatBRaille deve estar hospedado em endereço web fixo.
RNF02	O transcritor MatBRaille deve sempre ser acessado de forma gratuita, assim como os serviços oferecidos pela aplicação.
RNF03	Quaisquer alterações na grafia braille e ou no Código Matemático Unificado devem acarretar em alterações no código fonte do transcritor, de modo que ele continue a transcrever de forma correta.

5.2 Protótipos de tela

Nesta seção serão apresentadas protótipos das telas, ou seja, esboços das telas da aplicação.

Ao abrir a ferramenta, o usuário se depara com uma tela semelhante a Figura 5.3. O campo a esquerda é dedicado a escrita do documento a ser transcrito, por sua vez, no campo a direita é exibido o texto transcrito em braille.

Figura 5.3 – Tela inicial da ferramenta MatBraille.



Na Figura 5.4, há uma exemplificação de como ocorre a transcrição. No campo a esquerda visualizamos o texto "Olá, seja bem vindo!" e no campo a direita a sua real transcrição para braille português.

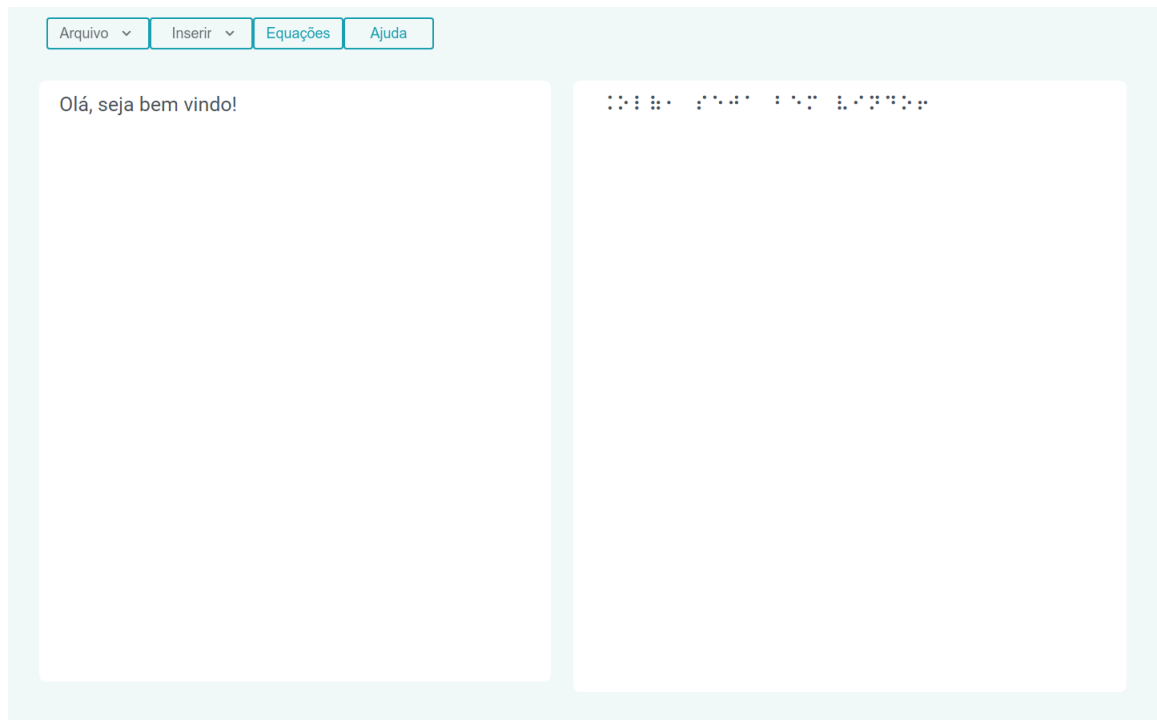


Figura 5.4 – Tela com exemplo de transcrição de um texto.

Uma segunda janela é apresentada na Figura 5.5. Esta janela é aberta após o evento de clique sobre o botão "Equações". O principal motivo da separação de janelas é a intenção de orientar o usuário a inserir de forma correta algumas simbologias matemáticas.

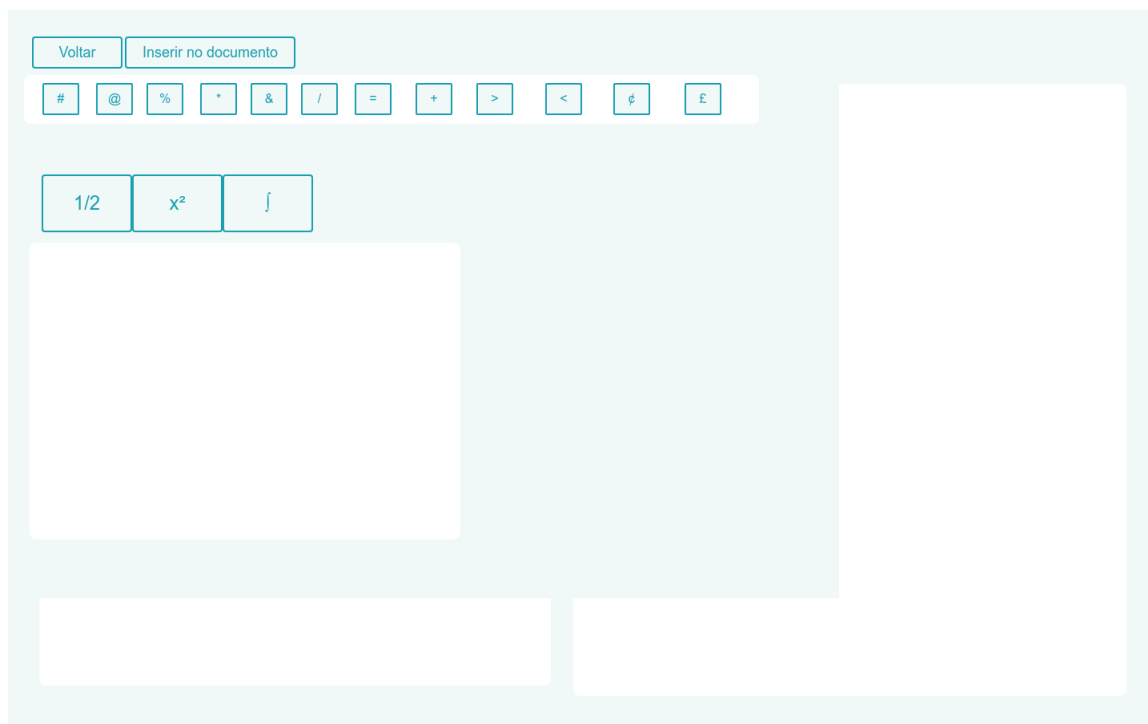


Figura 5.5 – Tela de inserção de fórmulas matemáticas.

Nesta abordagem foram priorizadas a representação apenas das funcionalidades realmente implementadas, dando assim um aspecto mais realístico. Entretanto, em versões futuras novas funcionalidades serão inseridas como, por exemplo, opções de edições e configurações do documento e da aplicação.

5.3 Implementação

Primeiramente, na Subseção 5.3.1 são detalhadas quais foram as tecnologias que permitiram a criação da aplicação proposta. Em sua Subseção posterior (Subseção 5.3.2), são apresentadas as funcionalidades desenvolvidas conjuntamente com uma execução real da aplicação desempenhando a funcionalidade.

5.3.1 Tecnologias

A implementação da ferramenta MatBRaille tem a base no seu desenvolvimento no *framework* código aberto Electron, que é mantido pelo GitHub. Este *framework* se baseia na ideia da construção de aplicações para desktop GUI usando tecnologias originalmente usadas na criação de aplicações web: Node.js para o *back end* e Chromium para o *front end*.

Por ter sua base de desenvolvimento em um *framework* voltado para aplicações web, a aplicação proposta usa linguagem de marcação HTML5, a estilização via CSS e Bootstrap. Para um maior esclarecimento, o *framework* para programação *front-end*, Bootstrap é uma ferramenta de estilização que auxilia de forma gratuita no desenvolvimento de aplicações web conjuntamente com o HTML, CSS e JavaScript.

Outra tecnologia empregada na construção da aplicação web proposta é a programação em JavaScript, que se faz presente em funcionalidade como salvar documento, por exemplo. Para a realização da escrita de exemplificação e símbolos matemáticas foi adicionado ao projeto a biblioteca MathJax, citada no início dos Trabalhos Relacionados (Capítulo 4. Para realização da análise de forma automática foi incluída a biblioteca Flex, que também foi devidamente citada anteriormente no Capítulo 3, Seção 3.6. Já a funcionalidade de expansão do menu em submenus é garantida pelo *framework* Bootstrap trabalhando conjuntamente com a biblioteca de funções do JavaScript, jQuery.

5.3.2 Funcionalidades desenvolvidas

Nesta primeira versão da aplicação proposta estão disponíveis as seguintes funcionalidades:

- digitação e transcrição para braille de textos acentuados, tal funcionalidade por ser observada nas Figuras 5.6 e 5.7;
- tratamento de letras maiúsculas, tal funcionalidade por ser observada nas Figuras 5.6, 5.7 e 5.8;
- tratamento de números inteiros e reais, tais funcionalidades podem ser contatadas nas Figuras 5.10 e 5.11;
- pontuações ortográficas, tal funcionalidade por ser observada nas Figuras 5.6, 5.7 e 5.7;
- X símbolos, tal funcionalidade por ser observada de forma parcial (apenas sua representação alfanumérica na janela de equações) na Figura 5.9;
- inserção e transcrição de frações, tal funcionalidade por ser observada de forma parcial (apenas sua representação alfanumérica na janela de equações) na Figura 5.9
- inserção e transcrição do alfabeto grego minúsculo, tal funcionalidade por ser observada de forma parcial (apenas sua representação alfanumérica na janela de equações) na Figura 5.9;
- inserção e transcrição de limite diferencial, tal funcionalidade por ser observada de forma parcial (apenas sua representação alfanumérica na janela de equações) na Figura 5.9

- inserção e transcrição de integrais indefinidas, tal funcionalidade por ser observada de forma parcial (apenas sua representação alfanumérica na janela de equações) na Figura 5.9;
- inserção e transcrição de índice inferior direito, índice superior direito e índice superior e inferior direito;
- função salvar texto comum (não transcrito para o Sistema Braille);
- função salvar texto transcrito para o Sistema Braille;

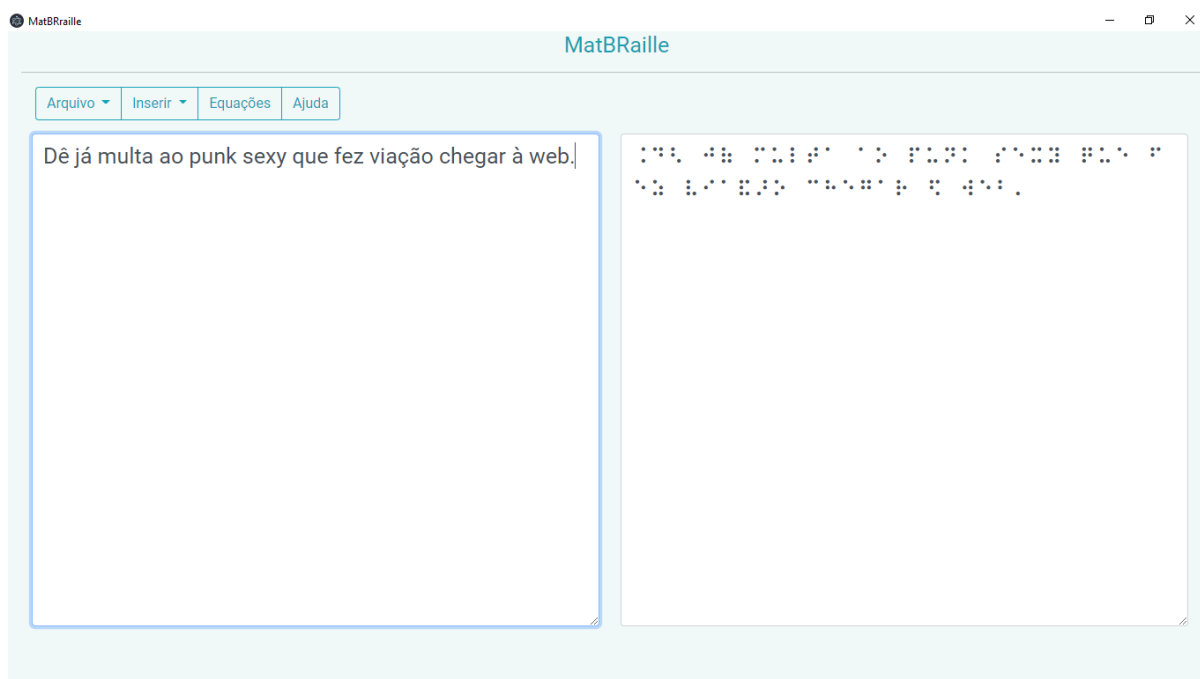


Figura 5.6 – Captura de tela contendo MatBRaille transcrevendo um pangrama que contem todas as 26 letras do alfabeto.

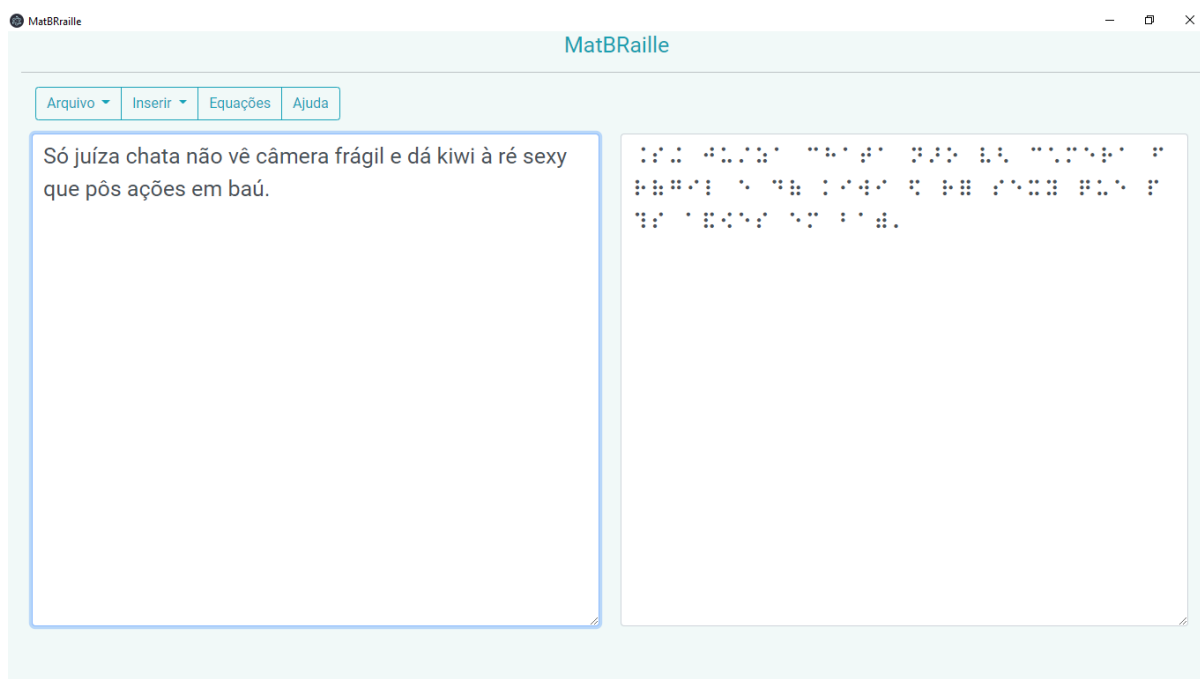


Figura 5.7 – Captura de tela com MatBRaille transcrevendo um pangrama que contem todos os acentos da língua portuguesa.

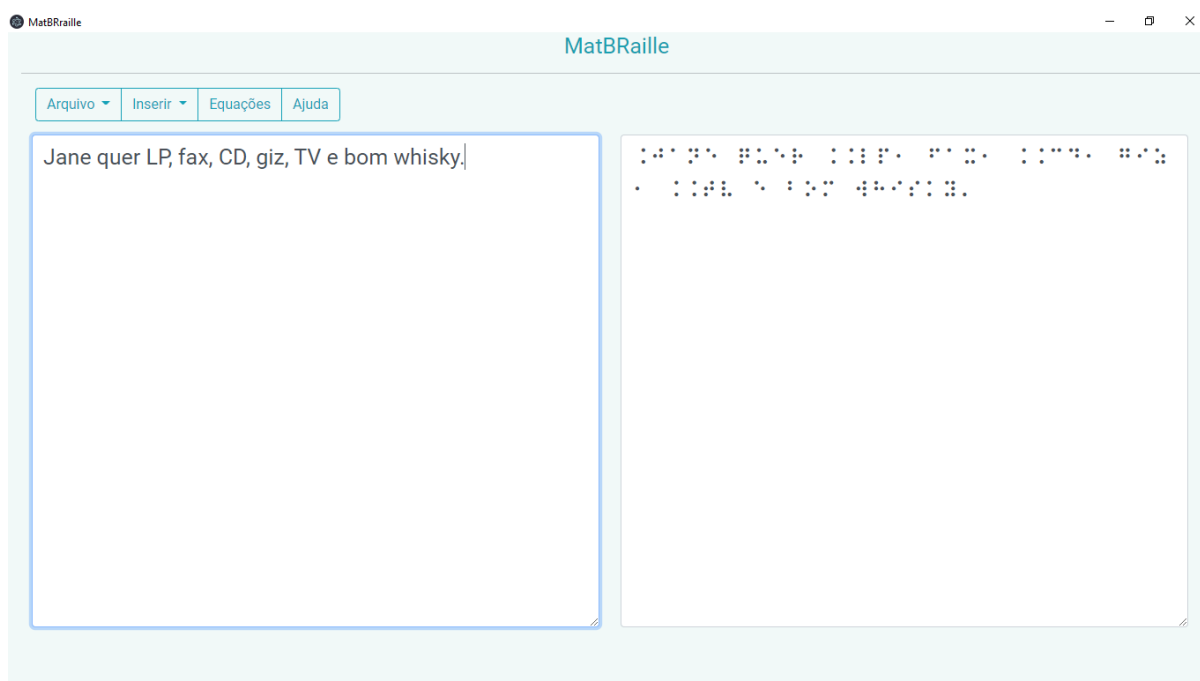


Figura 5.8 – Captura de tela contendo MatBRaille transcrevendo um pangrama contendo todo alfabeto.

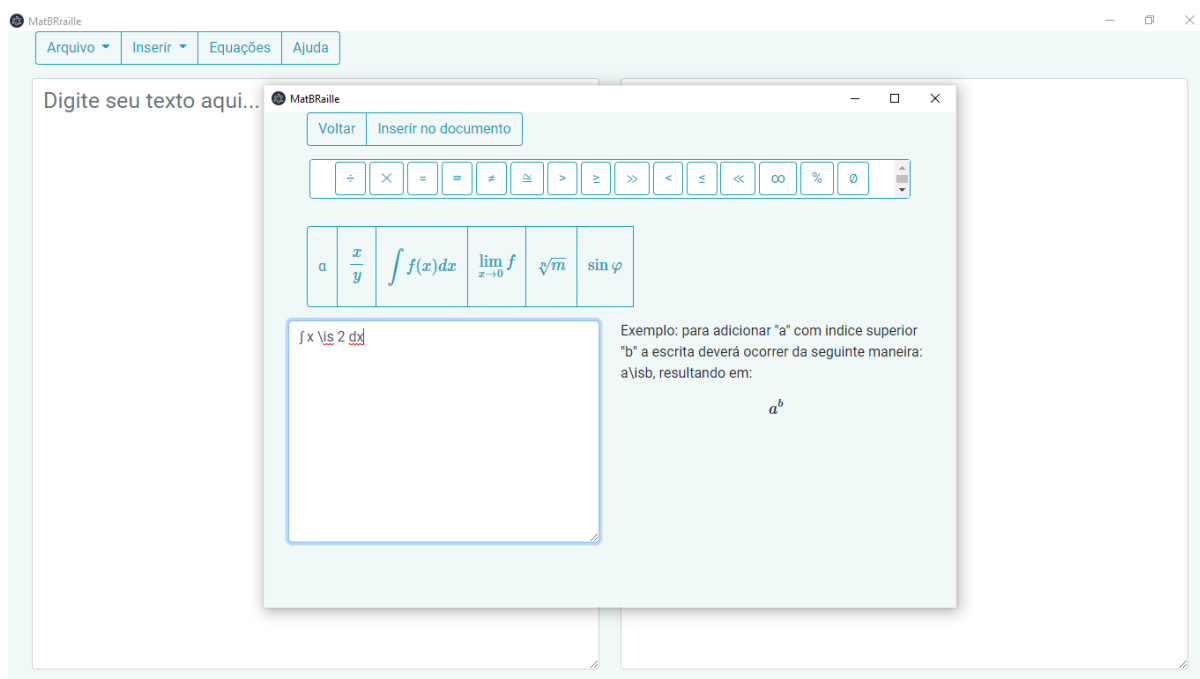


Figura 5.9 – Captura de tela contendo a janela de inserções de equações matemáticas.

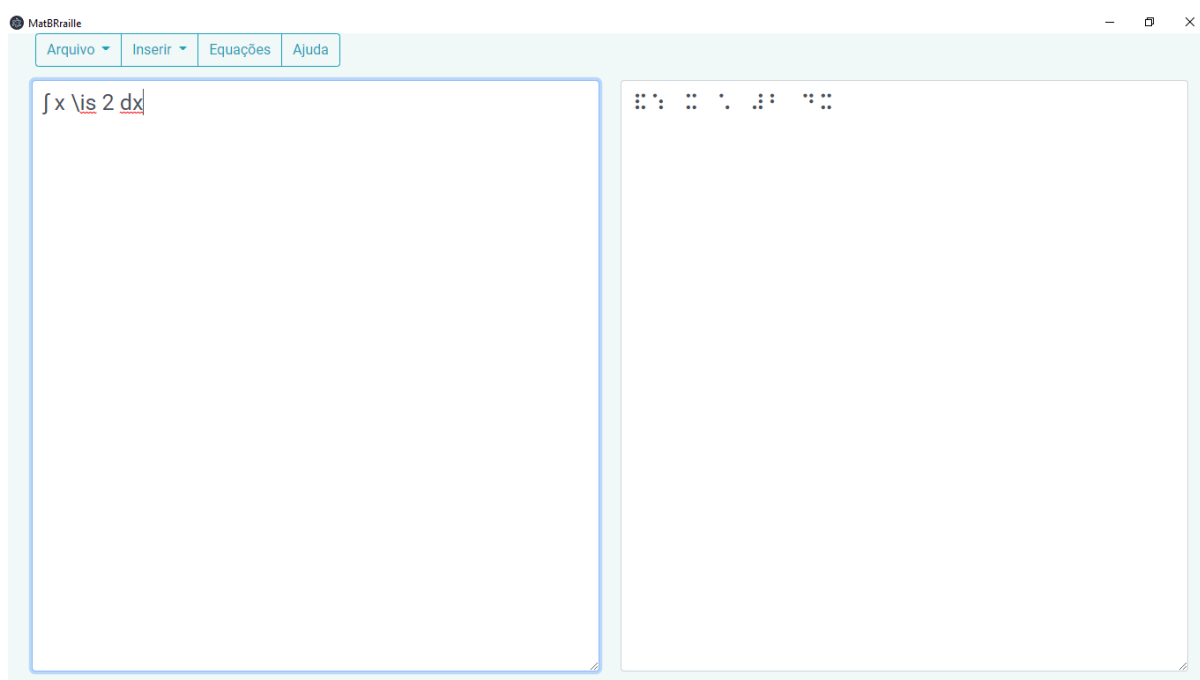


Figura 5.10 – Captura de tela contendo MatBRaille transcrevendo uma fórmula matemática.

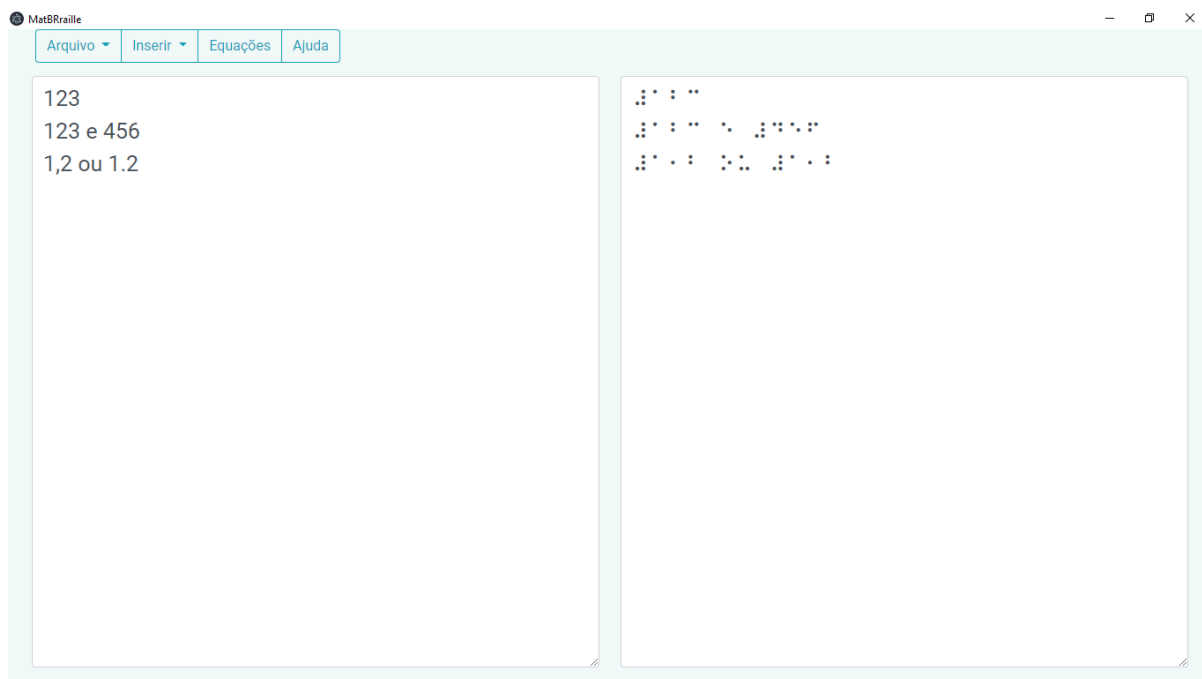


Figura 5.11 – Captura de tela contendo MatBraille transcrevendo números naturais e decimais.

6 Resultados

Toda aplicação após parte do desenvolvimento ou lançamento de uma versão deve ser testada, de modo que se tenha um *feedback* daquilo que foi concebido, permitindo assim sugestões, críticas e acima de tudo permita melhorias no desenvolvimento e correções de erros. Após o desenvolvimento de uma versão simplificada do MatBRaille foi realizado um teste inicial, mas por se tratar apenas da primeira versão este e outros testes serão aplicados sob a versão final do *software* que será desenvolvida na Monografia II.

Nesta Seção é mostrado um primeiro teste de usabilidade aplicado assim como suas métricas para avaliar a usabilidade do sistema (Seção 6.1). Em seguida na Seção 6.2 trás a computação de resultados preliminares e o que estes resultados representam.

6.1 SUS - *System Usability Scale*

Para avaliar a ferramenta proposta neste trabalho são necessários testes de usabilidade e de validação de conversão. Nesta etapa, um dos testes usados é o método SUS (*System Usability Scale*), que de acordo com Brooke et al. (1996), consiste na aplicação de um questionário que pretende avaliar aspectos de efetividade (os usuários conseguem completar seus objetivos?), eficiência (quanto esforço e recursos são necessários para isso?) e satisfação (a experiência foi satisfatória?). As respostas do formulário devem variar de 1 a 5, sendo que 1 significa discordo completamente e 5 concordo completamente. Após a aplicação do formulário e coleta de dados é calculada uma pontuação de acordo com o retorno dado pelos usuários, essa pontuação pode variar de 0 a 100.

A seguir estão as questões padrões do SUS:

1. Eu acho que gostaria de usar esse sistema com frequência.
2. Eu acho o sistema desnecessariamente complexo.
3. Eu achei o sistema fácil de usar.
4. Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.
5. Eu acho que as várias funções do sistema estão muito bem integradas.
6. Eu acho que o sistema apresenta muita inconsistência.
7. Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.

8. Eu achei o sistema atrapalhado de usar.
9. Eu me senti confiante ao usar o sistema.
10. Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.

Todos os enunciados acima são classificados de 1 a 5 por cada um dos usuários e a pontuação de cada usuário é calculada da seguinte maneira:

- para as respostas das questões ímpares deve-se subtrair uma unidade da quantia dada por cada usuário. Exemplo: se o *usuário X* deu nota 4 na questão 1, desse valor 4 deve ser subtraído uma unidade, pois, se trata de uma questão de número ímpar, resultando na pontuação 3. Este mesmo raciocínio deve ser empregado nas outras questões ímpares (3, 5, 7, 9) de todos os n usuários (considerando n como o número total de usuários do sistema) ;
- em respostas pares, ou seja, nas questões 2, 4, 6 e 8, a nota será o resultado do valor 5 subtraído pela nota atribuída pelo usuário. Exemplo: a resposta do *usuário Y* na questão 2 foi 3. Assim, o valor final do *usuário Y* na questão 2 será: $5 - 3 = 2$;
- por fim multiplica-se por 2.5 a soma de todos os pesos obtidos.

A nota final é a média dos valores obtidos anteriormente, ou seja, a soma da pontuação que cada usuário fez, dividido pelo número de usuários que testaram o *software*.

6.2 Resultados Preliminares

Para a aplicação do formulário SUS, foi proposta a realização de uma tarefa enviada via Google formulários, que pode ser acessada pelo seguinte link:

<https://drive.google.com/drive/folders/1IqnP6c6rMllgApOxrPRsruTWrvVWmOxOH?usp=sharing>

1. Fazer a instalação da ferramenta. No link se encontram duas opções de download e um manual de instalação.
2. Após executar a aplicação, faça:
 - digite: seu nome, idade e altura
 - no menu principal clique no botão: Equações e uma nova janela de visualização será aberta
 - tente então inserir a operação: $\int x^2 dx$
 - insira a equação na tela anterior e salve a sua versão.
3. Por fim, foi acrescentado um campo pra sugestões e críticas.

Após a avaliação de usabilidade utilizando o SUS na primeira versão da ferramenta, foram obtidos os resultados dispostos na Tabela 6.2, onde está exposta cada pontuação dada por cada um dos 12 usuários nas dez questões do SUS. No último campo, de cada linha da tabela, é mostrado o resultado do cálculo da pontuação obtida através da resposta daquele usuário.

Tabela 6.1 – Respostas individuais formulário SUS

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Total
P1	5	1	5	2	5	1	5	1	5	1	97.5
P2	5	1	4	1	4	3	5	2	3	1	82.5
P3	3	2	3	5	4	2	5	2	3	2	70
P4	5	1	5	1	5	1	5	1	4	1	91.5
P5	5	1	4	2	5	1	5	1	5	1	95
P6	5	1	5	1	5	1	5	1	4	1	97.5
P7	3	1	5	1	5	1	5	5	4	1	90
P8	4	1	4	1	5	1	5	1	5	1	95
P9	5	1	5	1	5	1	5	1	5	1	100
P10	3	1	5	2	5	1	5	1	2	1	87.5
P11	5	1	5	1	5	1	5	1	4	1	97.5
P12	4	2	4	1	4	1	5	1	4	1	87.5

A seguir, são exibidas as sugestões e críticas dos usuários registradas no formulário SUS aplicado (em alguns itens há também breves comentários sobre)

- P2: Tive dificuldades para encontrar como fazer a operação de elevar ao quadrado. Achei que não teria como e continuei testando quais outras operações eu iria conseguir fazer. Ao testar essas outras funções foi quando eu encontrei um "submenu". Por outro lado, as funções são bem resumidas e acredito que em pouco tempo um usuário consegue entender bem como funciona. Acredito também que é um software que tem bastante impacto e um bom propósito.
- P5: Gostaria de saber o que deveria fazer com arquivo após salvá-lo.
- P6: Eu só sugiro deixar uma tecla legível para inserir sobrescritos nas equações. No mais, está perfeito!
- P7: A maioria das funções é intuitiva e fácil de aprender. Apenas demorei um pouco para encontrar a opção de inserir o sobrescrito. Acredito que seria mais facilmente encontrada se ela aparecesse direto na barra de opções sem ser no menu suspenso, ou se o título do menu suspenso fosse diferente, indicando que ali existem outras opções além da raiz quadrada.
- P11: Excelente trabalho, parabéns!

A média das 12 pontuações dadas pelos usuários é de 83.04166 o que configura uma boa receptividade com relação à usabilidade da primeira versão da aplicação proposta neste trabalho.

6.3 Aplicação do MatBraille

Como parte dos resultados apresentamos a transcrição de um parágrafo do livro (STEWART, 2008), página 26, 7ª Edição. O referente parágrafo poder ser visualizado na Figura 6.1. A Figura 6.2, mostra o processo de inserção a índice inferior n (a_n); a Figura 6.3 um exemplo de inserção de fração ($\frac{2}{5}$) e por fim na Figura 6.4 é exposta toda parte visível em tela da transcrição completa do parágrafo proposto.

Polinômios

Uma função P é denominada **polinômio** se

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_2 x^2 + a_1 x + a_0$$

onde n é um inteiro não negativo e os números $a_0, a_1, a_2, \dots, a_n$ são constantes chamadas **coeficientes** do polinômio. O domínio de qualquer polinômio é $\mathbb{R} = (-\infty, \infty)$. Se o coeficiente dominante $a_n \neq 0$, então o **grau** do polinômio é n . Por exemplo, a função

$$P(x) = 2x^6 - x^4 + \frac{2}{5}x^3 + \sqrt{2}$$

é um polinômio de grau 6.

Figura 6.1 – Captura de parte da explicação de polinômios contida em (STEWART, 2008)

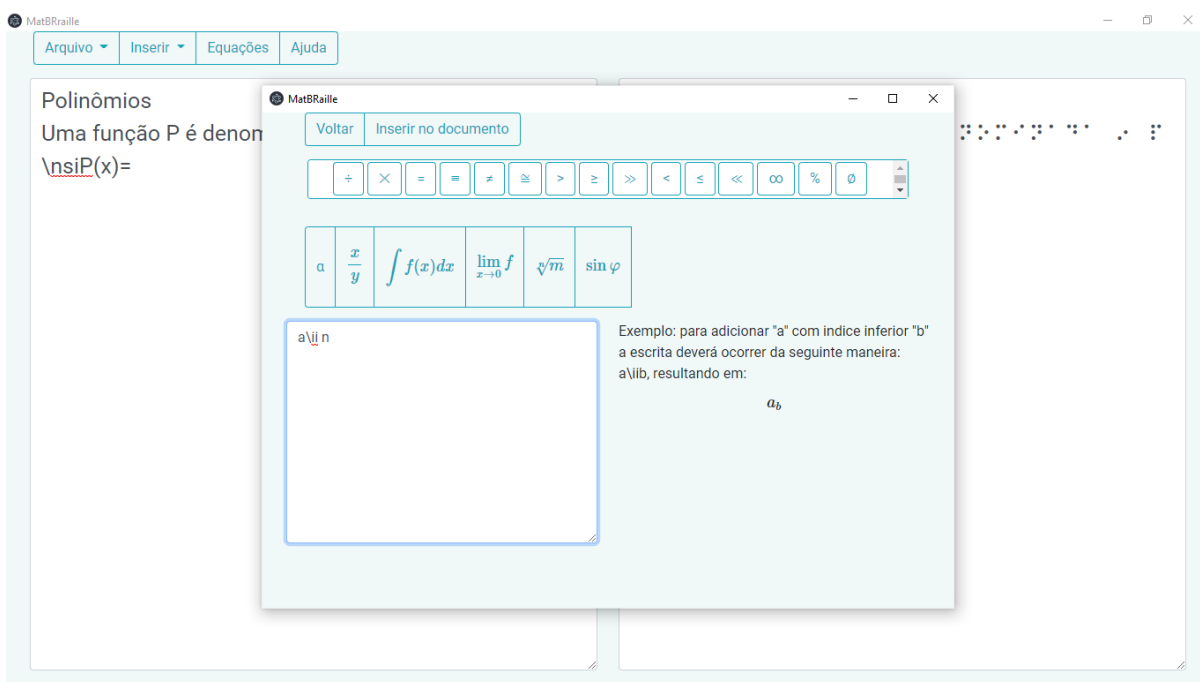


Figura 6.2 – Inserção de índice inferior

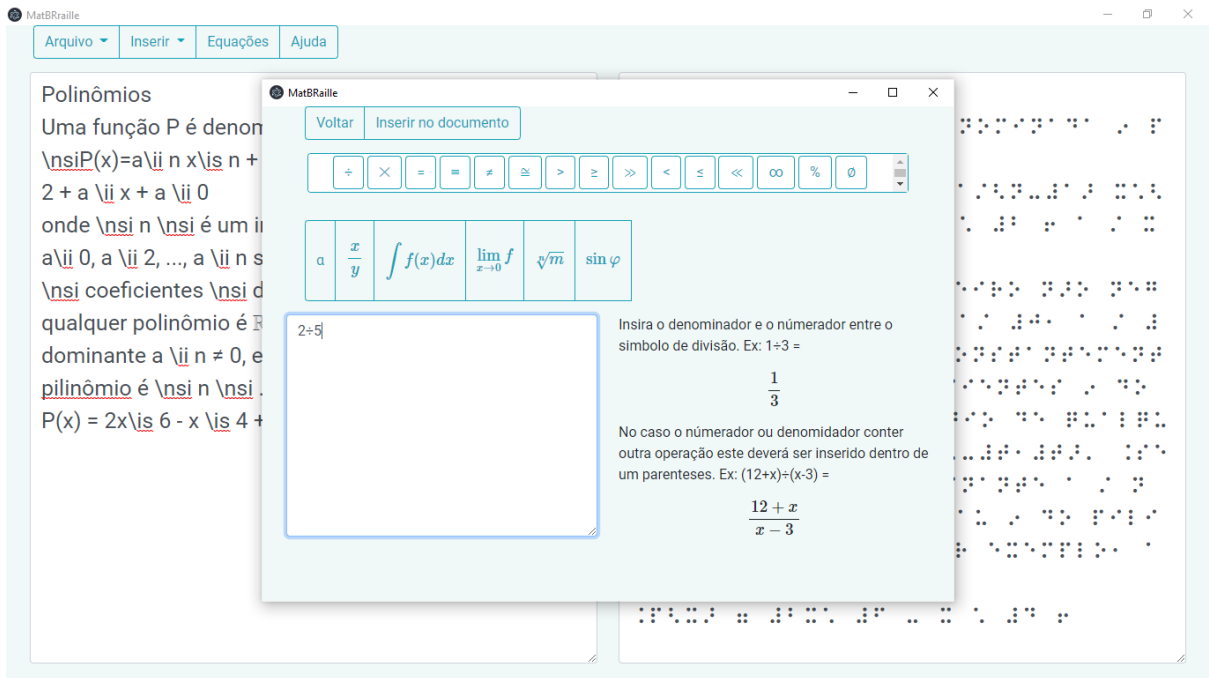


Figura 6.3 – Inserção de fração

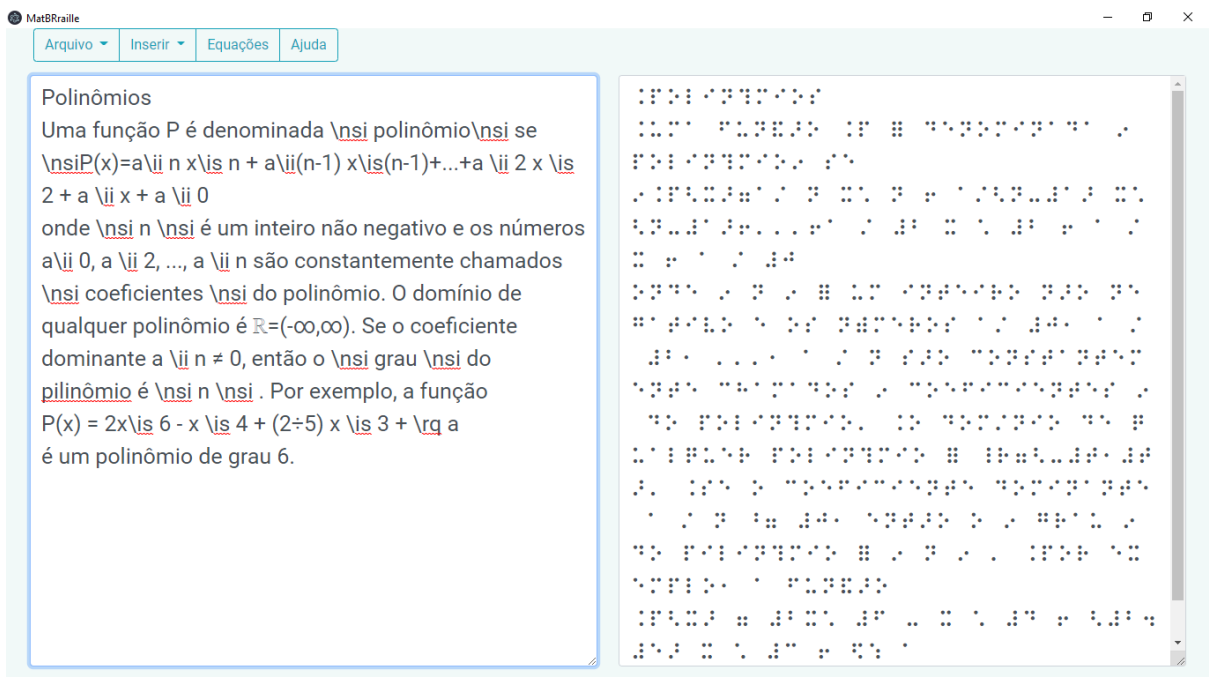


Figura 6.4 – Transcrição de paragrafo

7 Considerações Finais

Nesta monografia é proposta a ferramenta MatBraille para fazer a transcrição de textos escritos em português e digitados de forma alfanumérica para braille seguindo as regras do documento BRASIL (2018). Tendo como diferencial a possibilidade de inserção direta e simples de fórmulas matemáticas usadas no ensino superior, caracteres e símbolos especiais nas quais seguirão as regras de transcrição do documento Brasil (2006).

Para tanto, foi realizada uma revisão bibliográfica seguindo os passos de uma revisão sistemática da literatura para obter um levantamento das ferramentas relacionadas ao trabalho proposto no estado da arte e obter as definições importantes para o entendimento do trabalho. Seguindo a metodologia proposta foram expostas uma documentação inicial do transcritor on-line, que ainda está em fase de desenvolvimento. A documentação que se faz necessária e configura como uma boa prática de Engenharia de Software.

Por fim, foi apresentado o início da implementação do transcritor de texto em português com suas impressões iniciais sobre usabilidade, averiguadas através do SUS.

Ao fim desta primeira etapa foi obtida a primeira versão da ferramenta MatBRaille. Entretanto, esta primeira versão apresenta opção salvar documento de texto comum e transcrição de textos não acentuados, com ausência de letras maiúsculas, ç, pontuações e simbologias. Ressaltando que esta aplicação web que tem como objetivo transcrever textos e fórmulas matemáticas para braille português.

Outro importante ponto é que já foi executado o primeiro teste de usabilidade SUS, onde foi obtida a pontuação final de 83.04166 que caracteriza um bom resultado com relação à usabilidade da primeira versão desenvolvida no trabalho de monografia.

7.1 Trabalhos Futuros

Como trabalhos futuros pretende-se realizar as seguintes tarefas:

1. Implementação de mais funcionalidades matemáticas, das quais se destacam:
 - integrais definidas;
 - alfabeto grego maiúsculo;
 - símbolos de provas matemáticas;
2. implementação do menu Editar, contendo as funcionalidades:
 - copiar e colar;

- desfazer e refazer;
 - localizar e localizar e substituir
3. implementação do menu Inserir, contendo as funcionalidades:
- inserção de margem;
 - inserção de símbolo de fim de página;
4. Reavaliação da usabilidade acatando sugestões dadas neste trabalho e realização de uma nova validação do sistema.

Referências

- AHO, A. V.; LAM, M. S.; SETHI, R.; ULLMAN, J. D. *Compilers - Principles, Techniques, and Tools*. 2. ed. Pearson/Addison Wesley, 2006. ISBN 0321486811. Disponível em: <<http://gen.lib.rus.ec/book/index.php?md5=90DB32D070CFB70CA617E655D5C35529>>.
- AMAZON KINDLE. AZW. 2019. Disponível em: <https://kdp.amazon.com/pt_BR/help/topic/G200634390>. Acesso em: 14 nov. 2019.
- APPEL, A.; PALSBERG, J. *Modern compiler implementation in Java*. 2002. [S.l.]: Cambridge University Press, 2002.
- ASSOCIAÇÃO DE DEFICIÊNTES VISUAIS E AMIGOS. *Sistema Braille*. 2019. Disponível em: <<https://www.adeva.org.br/braille.php>>. Acesso em: 10 dez. 2019.
- ASTER. *AsTeR*. 1994. Disponível em: <<http://www.cs.cornell.edu/home/raman/aster/aster-toplevel.html>>. Acesso em: 19 nov. 2019.
- ATTRACTOR. *Braille translator*. 2019. Disponível em: <<https://www.atractor.pt/mat/matbr/matbraille.html>>. Acesso em: 19 nov. 2019.
- AUTÔMATO Finito Determinístico. Disponível em: <<http://brkmnd.com/pages/projects/p/nfa2dfa/billeder/nfareexam16.jpg>>. Acesso em: 20 ago. 2021.
- AUTÔMATO Finito Não Determinístico. Disponível em: <https://miro.medium.com/max/1838/1*Ip9dV7y7kXk5lseU9Dlwuw.png>. Acesso em: 20 ago. 2021.
- BALANSIN, C. F. Especificação e implementação de um leitor de tela. *Unioeste-Universidade Estadual do Oeste do Paraná. Monografia*. <http://www.inf.unioeste.br/~tcc/2011/TCC-Cleiton.pdf>. Maio, 2011.
- BENGALA LEGAL. *Acessibilidade, Inclusão Social e Desenho Universal: Tudo a Ver*. 2006. Disponível em: <<http://www.bengalalegal.com/martagil>>. Acesso em: 19 dez. 2019.
- BERNAREGGI, C.; ARCHAMBAULT, D. Mathematics on the web: emerging opportunities for visually impaired people. In: ACM. *Proceedings of the 2007 international cross-disciplinary conference on Web accessibility (W4A)*. [S.l.], 2007. p. 108–111.
- BEZERRA, C. M. C. et al. Br braille: programa tradutor de textos braille digitalizados para caracteres alfanumericos em portugues. [sn], 2003.
- BIBLOS. *Biblos*. 2019. Disponível em: <<http://www.digrande.it/biblos/Default.asp>>. Acesso em: 19 nov. 2019.
- BOXENTRIQ. *Braille Alphabet Decoder and Alphabet*. 2019. Disponível em: <<https://www.boxentriq.com/code-breaking/braille-alphabet>>. Acesso em: 19 nov. 2019.
- BRAIL,CURSO EMAG. *Linha Braille*. 2019. Disponível em: <<http://emag.governoeletronico.gov.br/cursodesenvolvedor/introducao/tecnologia-assistiva-linha-braille.html>>. Acesso em: 6 dez. 2019.

- BRaille CONVERTER. *Braille Converter*. 2019. Disponível em: <<http://braille.ru/>>. Acesso em: 19 nov. 2019.
- BRaille TRANSLATOR. *Braille translator*. 2019. Disponível em: <<https://brailletranslator.net/>>. Acesso em: 19 nov. 2019.
- BRailleTRANSLATOR.ORG. *BrailleTranslator.org*. 2019. Disponível em: <<https://www.brailletranslator.org/>>. Acesso em: 19 nov. 2019.
- BRANAH.COM. *Braille translator*. 2019. Disponível em: <<https://www.branah.com/braille-translato>>. Acesso em: 19 nov. 2019.
- BRASIL. *Decreto LEI Nº 7.853, DE 24 DE OUTUBRO DE 1989*. 1989. Disponível em <https://www.planalto.gov.br/ccivil_03/leis/l7853.htm>. Acesso em: 17 nov. 2019.
- BRASIL. *Código matemático unificado para a língua portuguesa*. [S.l.]: Brasília: Secretaria de Educação Especial, 2006.
- BRASIL. *Grafia Braille para Língua Portuguesa*. 3. ed. Brasília - DF: Ministério da Educação do Brasil: Secretaria de Educação Continuada, Alfabetização, Diversidade e Inclusão, 2018. Organização: Fernanda Christina dos Santos e Regina de Fátima Caldeira de Oliveira.
- BRASIL. *Algumas leis que protegem os deficientes visuais brasileiros*. 2019. Disponível em: <<http://www4.planalto.gov.br/ipcd/assuntos/legislacao>>. Acesso em: 6 dez. 2019.
- BROOKE, J. et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, London-, v. 189, n. 194, p. 4–7, 1996.
- BYRON KNOLL. *Online Braille Generator*. 2019. Disponível em: <<http://www.byronknoll.com/braille.html>>. Acesso em: 19 nov. 2019.
- CARVALHO, J. O. F. d. et al. Soluções tecnológicas para viabilizar o acesso do deficiente visual à educação a distância no ensino superior. [sn], 2001.
- CARVALHO, M. F. de; SILVA, A. F. da; SANTOS, J. A. dos; BORGES, J. T. C. d. S. Da reglete ao braille fácil—uma breve história do desenvolvimento tecnológico do braille no brasil. *TRAJETÓRIAS DA INFORMÁTICA NA AMÉRICA LATINA E CARIBE: AUTONOMIAS, (IN)DEPENDÊNCIAS E MUITAS OUTRAS HISTÓRIAS*, NCE/UFRJ, p. 119, 2019.
- CBO. *Série Deficiência Visual- vol 5*. 2019. Disponível em: <http://visaosubnormal.org.br/downloads/serie_deficiencia_visual_vol5_cbo_bq.pdf>. Acesso em: 17 ago. 2021.
- COLPES, K. M. Impressora de gráficos em alto-relevo para cegos: um facilitador no ensino da física e da matemática. 2014.
- COMPREHENSIVE TEX ARCHIVE NETWORK. *Latex2nemeth*. 2019. Disponível em: <<https://www.ctan.org/pkg/latex2nemeth>>. Acesso em: 19 nov. 2019.
- DESIGN SCIENCE - HOW SCIENCE COMMUNICATES. *MathPlayer*. 2018. Disponível em: <<https://www.dessci.com/en/products/mathplayer/>>. Acesso em: 19 nov. 2019.
- DESIGN SCIENCE - HOW SCIENCE COMMUNICATES. *ReaderHear*. 2018. Disponível em: <https://download.cnet.com/ReadHear/3000-2051_4-75557585.html#full-specs>. Acesso em: 19 nov. 2019.

DOGNINI, M. J.; RAABE, A. L. A. Eduling-software educacional para linguagens regulares. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. [S.l.: s.n.], 2003. v. 1, n. 1, p. 216–225.

EMACSPEAK. *Emacspeak*. 2019. Disponível em: <<http://emacspeak.sourceforge.net/>>. Acesso em: 19 nov. 2019.

FOLEISS, J. H.; ASSUNÇÃO, G. P.; CRUZ, E.; GONÇALVES, R.; FELTRIN, V. Scc: Um compilador c como ferramenta de ensino de compiladores. In: *WEAC2009-Workshop Educação em Arquitetura de Computadores*. [S.l.: s.n.], 2009. p. 15–22.

FONT MEME. *Braille Alphabet Decoder and Alphabet*. 2019. Disponível em: <<https://fontmeme.com/braille/>>. Acesso em: 19 nov. 2019.

FOTO: FELIPE LAZZAROTTO/ EPTV. *Reglete e punção*. 2014. Disponível em: <<https://images.app.goo.gl/9bQgB7mj9xMuW2WQ9>>. Acesso em: 06 dez. 2019.

FREEDOM SCIENTIFIC. *OpenBook*. 2019. Disponível em: <<https://support.freedomscientific.com/Downloads/OpenBook>>. Acesso em: 19 nov. 2019.

FREIRE, P. et al. *A importância do ato de ler*. [S.l.]: São Paulo: Cortez, 1982. v. 98.

FREIRE, P. et al. *A importância do ato de ler*. [S.l.]: Moderna, 2003.

FUNDAÇÃO DORINA NOVILL. *Dados de deficientes visuais no Brasil*. 2019. Disponível em: <<https://www.fundacaodorina.org.br/a-fundacao/deficiencia-visual/estatisticas-da-deficiencia-visual/>>. Acesso em: 6 dez. 2019.

GLAT, R.; PLETSCHE, M. D. O papel da universidade frente às políticas públicas para educação inclusiva. *Revista Benjamin Constant*, v. 29, n. 1, p. 3–8, 2004.

GNOME. *Orca*. 2019. Disponível em: <<https://wiki.gnome.org/action/show/Projects/Orca?action=show&redirect=Orca>>. Acesso em: 14 nov. 2019.

GUIMARAES, J. de O. *Construção de compiladores*. 2007.

IBGE. *Censo 2010*. 2010. Disponível em: <<https://biblioteca.ibge.gov.br/index.php/biblioteca-catalogo?view=detalhes&id=794>>. Acesso em: 18 nov. 2019.

INEP. *Censo 2020*. 2020. Disponível em: <https://download.inep.gov.br/educacao_superior/censo_superior/documentos/2020/Apresentacao_Censo_da_Educacao_Superior_2019.pdf>. Acesso em: 17 ago. 2021.

INFTYPROJECT. *InftyEditor*. 2019. Disponível em: <<http://www.inftyproject.org/en/software.html>>. Acesso em: 19 nov. 2019.

INSTITUTO BENJAMIN CONSTANT. *Quem foi Louis Braille*. 2019. Disponível em: <<http://www.ibc.gov.br/fique-por-dentro/339-quem-foi-louis-braille>>. Acesso em: 9 dez. 2019.

LAMBDA. *Lambda*. 2019. Disponível em: <<http://www.lambdaproject.org/>>. Acesso em: 19 nov. 2019.

LINGOJAM. *Braille translator*. 2019. Disponível em: <<https://lingojam.com/BrailleTranslator>>. Acesso em: 19 nov. 2019.

- MATHJAX. *MathJax*. 2019. Disponível em: <mathjax.org>. Acesso em: 19 nov. 2019.
- MATHSISFUN.COM. *Braille translator*. 2016. Disponível em: <<https://www.mathsisfun.com/braille-translation.html>>. Acesso em: 19 nov. 2019.
- MICROSOFT. *Window-Eyes*. 2019. Disponível em: <<https://www.gwmicro.com/Window-Eyes/>>. Acesso em: 14 nov. 2019.
- MOBIPOCKET. *Mobipocket*. 2019. Disponível em: <<http://www.mobifilereader.com/>>. Acesso em: 14 nov. 2019.
- MOZILLA. *JavaScript|MDN*. 2021. Disponível em: <<https://developer.mozilla.org/en-US/docs/Web/JavaScript>>. Acesso em: 20 ago. 2021.
- NCE/UFRJ - PROJETO DOSVOX. *Braille Fácil*. 2002. Disponível em: <<http://intervox.nce.ufrj.br/brfacil/>>. Acesso em: 19 nov. 2019.
- NCE/UFRJ - PROJETO DOSVOX. *Dosvoz*. 2002. Disponível em: <<http://intervox.nce.ufrj.br/dosvoz/>>. Acesso em: 19 nov. 2019.
- NCE/UFRJ - PROJETO DOSVOX. *Matvoz*. 2002. Disponível em: <<http://www.decom.fee.unicamp.br/~martini/matvoz02.zip>>. Acesso em: 19 nov. 2019.
- NCE/UFRJ - PROJETO DOSVOX. *MecDayse*. 2002. Disponível em: <<http://intervox.nce.ufrj.br/mecdaisy/>>. Acesso em: 19 nov. 2019.
- OPENOFFICE. *Odt2braille*. 2010. Disponível em: <<http://odt2braille.sourceforge.net/>>. Acesso em: 19 nov. 2019.
- PROJETO ACESSO. *Máquina de escrever braille*. 2019. Disponível em: <<https://images.app.goo.gl/dHWUz9sVcCZyfAn98>>. Acesso em: 06 dez. 2019.
- ROBOBRAILLE.COM. *Robo Braille*. 2019. Disponível em: <<https://www.robobraille.org/pt-pt>>. Acesso em: 19 nov. 2019.
- SCIENCE ACCESSIBILITY NET. *ChattyInfty*. 2019. Disponível em: <<https://www.sciaccess.net/en/ChattyInfty/index.html#Download>>. Acesso em: 19 nov. 2019.
- STEWART, J. *Cálculo: volume 1*. [S.l.]: Thomson, 2008.
- TALKMATHS. *TalkMats*. 2015. Disponível em: <<http://talkmaths.sourceforge.net/download.php>>. Acesso em: 19 nov. 2019.
- TECASSISTIVA. *JAWS*. 2019. Disponível em: <<https://www.tecassistiva.com.br/catalogo/jaws/>>. Acesso em: 14 nov. 2019.
- TECNOVISÃO. *Linha Braille*. 2019. Disponível em: <<https://images.app.goo.gl/Eaq6K9VhFFnpj0367>>. Acesso em: 6 dez. 2019.
- TRADUCTOR BRAILLE. *Traductor Braille*. 2019. Disponível em: <<https://traductorbraille.online/en>>. Acesso em: 19 nov. 2019.
- UFSCAR. *Start*. 2019. Disponível em: <<http://lapes.dc.ufscar.br/resources-and-downloads/tools>>. Acesso em: 14 nov. 2019.

UNICODE TOYS. *Unicode Braille Maker*[http](http://qaz.wtf/u/braille.cgi?text=hi+1). 2016. Disponível em: <<http://qaz.wtf/u/braille.cgi?text=hi+1>>. Acesso em: 19 nov. 2019.

UNIVERSITY OF CALIFORNIA AT BERKELEY. *Math Speak & Write*. 2019. Disponível em: <<https://people.eecs.berkeley.edu/~fateman/msw/msw.html>>. Acesso em: 19 nov. 2019.

VIEIRA, N. J. *Introdução aos fundamentos da computação: linguagens e máquinas*. [S.l.]: Pioneira Thomson Learning, 2006.