

UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

**BrailleÁgil: Um Software Livre para Apoio
à Transcrição de Textos em Braille**

João Antonio Mazzer Mantovani

**MONOGRAFIA FINAL
MAC 499 — TRABALHO DE
FORMATURA SUPERVISIONADO**

Supervisor: Prof. Dr. Pedro Henrique Dias Valle

São Paulo
2025

*O conteúdo deste trabalho é publicado sob a licença CC BY 4.0
(Creative Commons Attribution 4.0 International License)*

Ficha catalográfica elaborada com dados inseridos pelo(a) autor(a)
Biblioteca Carlos Benjamin de Lyra
Instituto de Matemática e Estatística
Universidade de São Paulo

Mantovani, João Antonio Mazzer
BrailleÁgil: Um Software Livre para Apoio à Transcrição de
Textos em Braille / João Antonio Mazzer Mantovani; orientador,
Pedro Henrique Dias Valle. - São Paulo, 2025.
61 p.: il.

Trabalho de Conclusão de Curso (Graduação) - Ciência
da Computação / Instituto de Matemática e Estatística
/ Universidade de São Paulo.

Bibliografia

1. SISTEMAS COMPUTACIONAIS. 2. EDITORES DE TEXTO.
 3. INTERAÇÃO USUÁRIO-COMPUTADOR. I. Valle, Pedro
Henrique Dias. II. Título.
-

Bibliotecárias do Serviço de Informação e Biblioteca
Carlos Benjamin de Lyra do IME-USP, responsáveis pela
estrutura de catalogação da publicação de acordo com a AACR2:
Maria Lúcia Ribeiro CRB-8/2766; Stela do Nascimento Madruga CRB 8/7534.

Este trabalho é dedicado a todas as pessoas cegas do Brasil bem como aos transcritores Braille, que realizam um trabalho importantíssimo de acessibilidade.

Agradecimentos

Se eu vi mais longe, foi por estar sobre ombros de gigantes.

— Isaac Newton

Gostaria de agradecer a todos aqueles que contribuíram para a realização deste trabalho:

Ao meu orientador, Prof. Dr. Pedro Henrique Dias Valle, por confiar no projeto e por todo o auxílio prestado.

A todos os professores do Departamento de Computação do IME-USP, que me permitiram chegar até aqui. E a todos os professores de outros departamentos que fizeram parte dessa jornada.

À equipe do Núcleo de Computação Eletrônica da UFRJ que desenvolveu o Braille Fácil. Em especial: José Antonio Borges, Geraldo José Ferreira das Chagas Jr. e Júlio Tadeu Carvalho da Silveira.

Aos transcritores do IBC - Gregório, Jorge e Diogo - e especialmente ao Thiago Duarte, pela disponibilidade de participação na pesquisa e comentários valiosos.

Ao Luiz e ao Rodrigo, da LARAMARA, pelo auxílio e pela recepção, além da demonstração de diversas tecnologias Braille.

À minha namorada, Domênica Tezzei, por todo o apoio, dedicação, revisão e dicas.

Resumo

João Antonio Mazzer Mantovani. **BrailleÁgil: Um Software Livre para Apoio à Transcrição de Textos em Braille**. Monografia (Bacharelado). Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2025.

A transcrição de textos para o sistema Braille é uma atividade essencial para promover a acessibilidade educacional e informacional. No Brasil, esse processo tem sido historicamente apoiado pelo software Braille Fácil, que desempenhou papel relevante na difusão da escrita Braille em meio digital. Apesar de sua importância, o cenário atual da transcrição em Braille apresenta sinais de estagnação tecnológica, com forte dependência de uma ferramenta legada que impõe limitações operacionais significativas, como a restrição ao sistema operacional Windows e a ausência de visualização em tempo real. Essas limitações fragmentam o fluxo de trabalho e aumentam a carga cognitiva dos transcritores. Diante desse contexto, este trabalho tem como objetivo desenvolver e validar o BrailleÁgil, um software de código aberto projetado para apoiar a transcrição de textos em Braille de forma mais eficiente, portátil e alinhada às necessidades atuais dos profissionais da área. O BrailleÁgil foi implementado em Python, priorizando leveza, simplicidade e portabilidade multiplataforma. Sua principal inovação consiste em um mecanismo de visualização simultânea e dinâmica, que permite ao transcritor validar instantaneamente a diagramação do conteúdo. A validação do sistema foi conduzida por meio de uma avaliação com quatro transcritores do Instituto Benjamin Constant (IBC), utilizando as heurísticas de usabilidade de Nielsen e o Modelo de Aceitação de Tecnologia (TAM). Os resultados indicam uma boa aceitação do BrailleÁgil pelos participantes, evidenciando seu potencial para aprimorar o processo de transcrição em Braille. Além disso, a avaliação permitiu identificar pontos de melhoria, direcionar propostas de trabalhos futuros e destacar as principais contribuições do software para o avanço tecnológico na área de acessibilidade.

Palavras-chave: Transcrição. Braille. Acessibilidade.

Abstract

João Antonio Mazzer Mantovani. **BrailleÁgil: Open Software to Support the Transcription of Texts into Braille.** Capstone Project Report (Bachelor). Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2025.

Text transcription into the Braille system is an essential activity for promoting educational and informational accessibility. In Brazil, this process has historically been supported by the Braille Fácil software, which has played a relevant role in disseminating Braille writing in digital environments. Despite its importance, we observe that the current Braille transcription landscape shows signs of technological stagnation, with a firm reliance on a legacy tool that imposes significant operational limitations, such as restriction to the Windows operating system and the lack of real-time visualization. These limitations fragment the workflow and increase transcribers' cognitive load. In this context, we aim to develop and validate BrailleÁgil, an open-source software designed to support Braille text transcription in a more efficient, portable, and professionally aligned manner. We implemented BrailleÁgil in Python, prioritizing lightness, simplicity, and cross-platform portability. Its main innovation lies in a simultaneous, dynamic visualization mechanism that enables transcribers to validate the content layout instantly. We validated the system through an evaluation involving four transcribers from the Benjamin Constant Institute (IBC), using Nielsen's usability heuristics and the Technology Acceptance Model (TAM). The results indicate strong acceptance of BrailleÁgil among participants, demonstrating its potential to enhance the Braille transcription process. Furthermore, the evaluation enabled us to identify opportunities for improvement, guide future work proposals, and clearly articulate the software's main contributions to technological advancement in accessibility.

Keywords: Transcription. Braille. Accessibility.

Lista de abreviaturas

IBC	Instituto Benjamin Constant
IHC	Interação Humano-Computador
IME	Instituto de Matemática, Estatística e Ciência da Computação
NCE	Núcleo de Computação Eletrônica
PEOU	Perceived Ease-of-Use
PU	Perceived Usefulness
TA	Tecnologia Assistiva
TAM	Technology Acceptance Model
UFRJ	Universidade Federal do Rio de Janeiro
USP	Universidade de São Paulo

Lista de figuras

3.1	Fórmula estrutural do paracetamol e sua representação no Braille Fácil	9
3.2	Fluxograma de processamento de texto do BrailleÁgil	13
3.3	Representação Visual do Código do Sistema BrailleÁgil	14
3.4	Exemplo de quebra de página com <p>	15
3.5	Exemplo de recuo de dois espaços	16
3.6	Exemplo de paginação manual em algarismos romanos e indo-árabicos	17
3.7	Exemplo de paginação correspondente ao livro em tinta	17
3.8	Exemplo de formatação automática e sua desativação	18
3.9	Menu de configuração de página	18
3.10	Exemplo de destaque da linha em edição	19
3.11	Exemplo de separação silábica	19
4.1	Indicação de linha e coluna na área de texto do Braille Fácil	25
4.2	Indicação de página e linha atual na visualização do Braille Fácil	25

Lista de tabelas

4.1	Explicação das 10 heurísticas de Nielsen	23
4.2	Resultado da Avaliação Heurística do BrailleÁgil	24
4.3	Média da pontuação da categoria PU	28
4.4	Média da pontuação do construto Facilidade de Uso Percebida	29
C.1	Escala Likert de 7 pontos utilizada no questionário	41

C.2	Afirmções utilizadas para a avaliação do construto de Utilidade Percebida	41
C.3	Afirmções utilizadas para a avaliação do construto de Facilidade de Uso Percebida	41

Sumário

1	Introdução	1
2	Referencial Teórico e Trabalhos Relacionados	3
2.1	Cegueira	3
2.2	O Sistema Braille	3
2.3	Sistemas Acessíveis e Tecnologias Assistivas	4
2.4	Trabalhos Relacionados	5
2.5	Considerações Finais	7
3	O sistema BrailleÁgil	9
3.1	Público-alvo do BrailleÁgil	10
3.2	Requisitos Funcionais do BrailleÁgil	10
3.3	Especificações Técnicas	12
3.4	Arquitetura do BrailleÁgil	14
3.5	Recursos do BrailleÁgil	15
3.5.1	<i>Tags</i>	15
3.5.2	Configuração da página	18
3.5.3	Destaque e posicionamento de texto	18
3.5.4	Separação de sílabas	19
3.6	Testes Unitários	19
3.7	Considerações Finais	20
4	Avaliação do BrailleÁgil	21
4.1	Avaliação Heurística	22
4.1.1	Planejamento da Avaliação	22
4.1.2	Execução da Avaliação Heurística	23
4.1.3	Análise dos Resultados	24
4.2	Modelo de Aceitação de Tecnologia (TAM)	26
4.2.1	Planejamento da Avaliação TAM	26

4.2.2	Execução da Avaliação TAM	27
4.3	Análise dos Resultados	27
4.4	Discussão	29
4.5	Considerações Finais	30
5	Conclusão	33
5.1	Principais Contribuições	34
5.2	Trabalhos futuros	34
5.3	Considerações finais	35
Apêndices		
A	Roteiro de Avaliação do sistema	37
B	Tabela de avaliação Heurística	39
C	Formulário de avaliação do TAM	41
Bibliografia		
		43

Capítulo 1

Introdução

O sistema Braille permanece, desde sua criação no século XIX, como o principal alicerce da alfabetização e da inclusão socioeducacional de pessoas cegas ou com baixa visão (BATISTA, 2018). No contexto brasileiro, a garantia de acesso a materiais didáticos acessíveis constitui uma diretriz estabelecida na Lei Brasileira de Inclusão, que reforça a responsabilidade do Estado e das instituições de ensino na provisão de tecnologias assistivas que assegurem a autonomia do estudante (Lei n. 13.146, de 6 de julho de 2015). A produção desses materiais, entretanto, depende de um processo de transcrição rigoroso, que converte textos da norma culta da língua portuguesa para a grafia Braille, observando estritamente as normas técnicas de padronização vigentes (MEC, 2018).

Nas últimas décadas, a evolução das ferramentas de software voltadas à automação desse processo possibilitou avanços significativos na produção em larga escala de materiais acessíveis (LEMOS e CERQUEIRA, 2014). A ampliação do acesso à leitura tátil está diretamente associada à disponibilidade de recursos computacionais que permitam ao transcritor (frequentemente um pedagogo ou técnico especializado) realizar a conversão de documentos complexos de forma ágil e precisa. Conforme apontado por TOFANI, 2012, o desenvolvimento de ferramentas computacionais para o Braille deve ir além da mera tradução de caracteres, contemplando também a preservação da semântica e da estrutura lógica do documento original.

Apesar de sua relevância social, o ecossistema de software dedicado à transcrição em Braille no Brasil enfrenta um cenário de estagnação tecnológica. O Braille Fácil, desenvolvido pelo Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro (NCE-UFRJ), consolidou-se como o padrão de facto no país e permanece essencial para o funcionamento de grandes gráficas Braille, como o Instituto Benjamin Constant (IBC) (LEMOS e CERQUEIRA, 2014). Todavia, essa ferramenta apresenta um isolamento técnico significativo: desenvolvida em Delphi (INPI, 2018), uma linguagem cuja modernização se mostra limitada no contexto atual do projeto, o sistema recebeu apenas uma atualização entre 2017 e 2025, evidenciando sua defasagem tecnológica.¹

Essa obsolescência acarreta limitações operacionais críticas. A dependência exclusiva do sistema operacional Windows inviabiliza sua adoção em ambientes que utilizam distri-

¹ De acordo com notas de atualização disponíveis em <https://intervox.nce.ufrj.br/brfacil/>

buições Linux ou priorizam soluções baseadas em software livre. Além disso, a arquitetura do sistema não favorece os padrões contemporâneos de interatividade (BARBOSA e SILVA, 2010). A ausência de visualização em tempo real impõe ao transcritor um fluxo de trabalho fragmentado, no qual cada alteração textual exige a execução manual de comandos para atualizar a saída em Braille. Esse processo compromete a eficiência da atividade e aumenta substancialmente a carga cognitiva do usuário (BARBOSA e SILVA, 2010).

Diante desse cenário, justifica-se o desenvolvimento de uma alternativa moderna e de código aberto que contribua para a melhoria da experiência do usuário e para a evolução sustentável das tecnologias assistivas no país. A adoção de um modelo de desenvolvimento aberto possibilita a participação ativa da comunidade acadêmica e de desenvolvedores independentes, reduzindo a dependência de soluções proprietárias ou de grupos restritos de desenvolvimento. Além disso, a migração para linguagens de programação contemporâneas e multiplataforma, como o Python, amplia o alcance da ferramenta, permitindo sua adoção por instituições que priorizam o *open source* por razões econômicas, técnicas ou de segurança.

Nesse contexto, este trabalho propõe o desenvolvimento e a validação do BrailleÁgil, um editor de textos em Braille multiplataforma e de código aberto, concebido com foco na alta interatividade e na atualização dinâmica do conteúdo tático em tempo real. O sistema foi projetado para refletir instantaneamente as alterações textuais na visualização em Braille, reduzindo erros de transcrição e otimizando o tempo de produção de materiais acessíveis. Para tanto, foi desenvolvida uma interface gráfica intuitiva, baseada na biblioteca Tkinter, garantindo compatibilidade com os sistemas Linux e Windows, bem como um conjunto de comandos compatíveis com o Braille Fácil, de modo a facilitar o processo de aprendizagem e adaptação dos profissionais da área.

A pesquisa adota uma abordagem incremental, utilizando o Python como base tecnológica, em razão de sua robustez no tratamento de cadeias de caracteres e da facilidade de integração com módulos de expressões regulares para a conversão de textos. A interface gráfica foi construída com ênfase na leveza do sistema e na portabilidade entre diferentes ambientes computacionais. A principal contribuição deste trabalho consiste na disponibilização do BrailleÁgil como uma alternativa viável, moderna e alinhada às demandas contemporâneas da transcrição Braille no Brasil. Diferentemente das soluções existentes, o BrailleÁgil introduz o conceito de edição dinâmica, no qual a cela Braille é tratada como um elemento de interface responsivo. Adicionalmente, o projeto contribui para o fortalecimento do ecossistema de software livre ao disponibilizar integralmente seu código-fonte em um repositório público no GitHub.²

O documento está organizado da seguinte forma: o Capítulo 2 discute o referencial teórico, abordando o sistema Braille, a legislação relativa à acessibilidade e o estado da arte dos produtos de software de transcrição. O Capítulo 3 descreve o projeto e o desenvolvimento do BrailleÁgil, detalhando as tecnologias empregadas. O Capítulo 4 apresenta a avaliação do sistema, incluindo os instrumentos adotados, o perfil dos participantes e a análise dos resultados. Por fim, o Capítulo 5 reúne as considerações finais, discutindo as contribuições do trabalho, suas limitações e as perspectivas para trabalhos futuros.

² Disponível em <https://github.com/Joaomantovani32/BrailleAgil>

Capítulo 2

Referencial Teórico e Trabalhos Relacionados

Este capítulo apresenta o panorama conceitual e tecnológico relacionado à transcrição em Braille, fundamentando-se na análise de produções científicas sobre deficiência visual, tecnologias assistivas e produtos de software acessíveis. Além disso, são discutidos trabalhos relacionados que abordam tanto ferramentas de apoio à leitura e à escrita em Braille quanto soluções computacionais voltadas à adaptação de conteúdos educacionais para pessoas cegas.

2.1 Cegueira

A deficiência visual abrange um amplo espectro de comprometimentos funcionais, que varia desde a baixa visão até a cegueira total (UMBELINO, 2023). Segundo a Organização Mundial da Saúde (OMS) (OMS, 2019), a cegueira é caracterizada por critérios específicos de acuidade visual e de restrição do campo visual, configurando uma condição em que o indivíduo passa a depender predominantemente de sentidos remanescentes, como o tato e a audição, para a percepção do ambiente e o acesso à informação.

No contexto brasileiro, não há dados oficiais consolidados sobre a população cega. No entanto, estimativas do Conselho Brasileiro de Oftalmologia indicam que há aproximadamente 267.000 pessoas cegas no país, o que evidencia a relevância social de políticas públicas e tecnologias voltadas à acessibilidade e à inclusão educacional (UMBELINO, 2023).

2.2 O Sistema Braille

O sistema Braille, criado em 1825 por Louis Braille, representa um marco histórico na educação de pessoas cegas ao substituir métodos rudimentares de leitura em relevo por uma cela padronizada composta por seis pontos salientes. Conforme destacado por LEMOS e CERQUEIRA, 2014, essa inovação possibilitou não apenas a alfabetização, mas também a representação de símbolos matemáticos, científicos e musicais, ampliando significativamente o acesso ao conhecimento formal.

O Brasil possui uma trajetória pioneira na adoção do Braille. Em 1854, com a fundação do Imperial Instituto dos Meninos Cegos, atual Instituto Benjamin Constant (IBC), o país tornou-se o primeiro da América Latina a oficializar o uso do sistema. Desde então, o IBC atua como centro de referência nacional, sendo responsável pela normatização da grafia em Braille para a Língua Portuguesa e pela liderança na produção de materiais didáticos acessíveis (LEMOS e CERQUEIRA, 2014).

2.3 Sistemas Acessíveis e Tecnologias Assistivas

A acessibilidade digital consolidou-se como elemento central da cidadania contemporânea, recebendo respaldo jurídico no Brasil por meio do Decreto Federal nº 5.296/2004. Esse dispositivo define a acessibilidade como a condição essencial para o uso autônomo e seguro de ambientes, produtos e, especialmente, de sistemas de comunicação e informação (CAT, 2009). No âmbito da Computação, a concretização desses direitos ocorre por meio das tecnologias assistivas (TA) (CAT, 2009).

De acordo com o Comitê de Ajudas Técnicas (CAT, 2009), a TA constitui uma área interdisciplinar que abrange recursos, estratégias, práticas e serviços destinados a promover a funcionalidade, a autonomia e a inclusão social de pessoas com deficiência ou mobilidade reduzida. Para usuários cegos, a interação com sistemas computacionais é mediada por um ecossistema de software que opera em diferentes camadas informacionais, tendo como base os leitores de tela, responsáveis por converter conteúdo textual em síntese de voz ou em sinais táteis por meio de linhas Braille.

Historicamente, o cenário brasileiro foi marcado pelo desenvolvimento do Dosvox, criado pelo NCE/UFRJ, que ofereceu uma solução integrada de ambiente computacional com síntese de voz em português (LEMOS e CERQUEIRA, 2014). Tal sistema foi pioneiro na inclusão digital de pessoas cegas, antecedendo a ampla disseminação de leitores de tela nativos em *smartphones* e em sistemas operacionais modernos (LEMOS e CERQUEIRA, 2014).

Para além da navegação, a produção de conteúdo tátil requer ferramentas especializadas de tradução e edição em Braille. No cenário internacional, destaca-se o *Duxbury Braille Translator* (DBT), uma solução comercial amplamente adotada, compatível com diversos idiomas e integrada a fluxos de trabalho complexos no *Microsoft Word*. No Brasil, entretanto, o Braille Fácil permanece como a principal ferramenta gratuita utilizada por instituições como o IBC (LEMOS e CERQUEIRA, 2014). Ainda assim, a adaptação de conteúdos acessíveis vai além do texto linear, abrangendo a criação de diagramas e ilustrações táteis. Nesse contexto, o software Monet desempenha papel relevante ao permitir a conversão de elementos visuais em representações táteis, viabilizando o acesso a conteúdos de áreas como geografia, biologia e matemática (VISSOSSI e LIMA, 2024).

A especialização em TAs alcança níveis ainda mais complexos no domínio da musicografia em Braille, em que a precisão sistêmica é fundamental (TOFANI, 2012). Ferramentas como o Musibraille buscam automatizar esse processo, enfrentando desafios arquiteturais significativos. Conforme discutido por TOFANI, 2012, a navegação em estruturas hierárquicas complexas exige o uso de APIs de acessibilidade robustas, capazes de garantir o processamento lógico da informação pelos dispositivos de saída.

Apesar dos avanços tecnológicos, a alfabetização em Braille permanece indispensável. Conforme alertado por [BATISTA, 2018](#), leitores de tela e de livros digitais não devem substituir o ensino do Braille, uma vez que a ausência da leitura tátil compromete a compreensão ortográfica e gramatical da linguagem escrita. Assim, a tecnologia assistiva deve atuar de forma complementar à educação em Braille, assegurando autonomia sem comprometer a formação linguística plena.

2.4 Trabalhos Relacionados

O Braille Fácil

A criação do software Braille Fácil, em 1999, constitui um marco na produção de documentos táteis no Brasil. Desenvolvido pelo NCE/UFRJ, sob a coordenação de José Antonio Borges, em parceria com o Instituto Benjamin Constant, o sistema surgiu com o objetivo de democratizar o processo de transcrição em Braille ([CARVALHO et al., 2018](#)).

Antes de sua introdução, a transcrição exigia conhecimentos técnicos avançados e equipamentos de alto custo ([CARVALHO et al., 2018](#)). O Braille Fácil simplificou esse cenário ao permitir a conversão automática de textos digitais convencionais em Braille, viabilizando sua impressão em impressoras específicas. Conforme apontam [ANDRIGHETTO et al., 2020](#), o software tornou-se fundamental para o Programa Nacional do Livro Didático (PNLD), assegurando que estudantes cegos recebessem seus livros simultaneamente aos demais alunos.

Entre suas principais inovações destacam-se a distribuição gratuita e a simplicidade de uso, fatores que contribuíram para sua ampla adoção em um contexto historicamente dominado por soluções proprietárias ([CARVALHO et al., 2018](#)). Ainda assim, as limitações estruturais e tecnológicas do sistema impõem desafios à sua evolução contínua.

O livro didático em Braille e o processo de adaptação

O trabalho de [VISSOSSI e LIMA, 2024](#) discute a relevância do livro didático em Braille na educação de pessoas cegas, enfatizando o processo de adaptação gráfica dos materiais. As autoras descrevem detalhadamente as etapas de diagramação e transcrição, destacando sua complexidade e a necessidade de planejamento criterioso da disposição textual e gráfica no material impresso.

O estudo evidencia, ainda, desafios estruturais da área, como a ausência de regulamentação formal da profissão de transcritor em Braille, o que pode resultar em materiais fora dos padrões normativos. Ademais, observa-se que, em algumas instituições, a simples impressão dos textos ocorre sem a validação por usuários proficientes em Braille, comprometendo a qualidade do material acessível. Tais aspectos reforçam a necessidade de ferramentas que auxiliem o processo de transcrição e adaptação de forma mais eficiente e padronizada.

Braillearning: software para simular a máquina de escrever em Braille

Reforçando a importância do ensino do Braille, [SANTANA et al., 2019](#) apresentam o Braillearning, um software voltado à simulação da máquina de escrever Perkins. O sistema

oferece diferentes modos de uso, incluindo tutoriais, exercícios guiados e digitação livre, com áudio como principal meio de interação.

A validação do sistema foi realizada com 20 usuários cegos alfabetizados em Braille, dos quais 95% possuíam experiência prévia com máquinas Perkins. Os resultados indicaram elevada aceitação da ferramenta, embora alguns participantes tenham ressaltado a ausência de *feedback* tático, o que evidencia que o sistema não substitui integralmente o contato com o Braille impresso.

Uma avaliação de programas de tradução Braille (1987)

Na década de 1980, **GOLDBERG et al., 1987** demonstraram preocupação com a qualidade dos sistemas de tradução em Braille ao comparar ferramentas como *Braille Talk*, *PC-Braille* e *Duxbury Translator*. O estudo avaliou aspectos como documentação, facilidade de uso, formatação e suporte ao usuário, considerando os padrões estabelecidos pela *National Library Service* dos Estados Unidos.

Os resultados apontaram inconsistências nos três sistemas, incluindo erros de pontuação e de contração, além de diferenças significativas em usabilidade e eficiência. Ressalta-se que todas as ferramentas avaliadas possuíam licenças pagas, com custos elevados à época. A conclusão indica que, embora funcionais, os sistemas exigiam persistência por parte dos usuários, evidenciando limitações de usabilidade ainda relevantes nos dias atuais.

Tradução computadorizada de Texto e Braille

O trabalho de **KING, 2001** apresenta uma análise aprofundada de sistemas computacionais de tradução bidirecional entre texto e Braille, com ênfase em abordagens baseadas em regras e em máquinas de estados finitos. A pesquisa propõe uma arquitetura modular e extensível, capaz de suportar múltiplos idiomas e diferentes graus de contração do Braille.

Do ponto de vista da engenharia de software, a adoção de interfaces abstratas favorece a reutilização e integração em sistemas maiores. Os resultados evidenciam um equilíbrio entre flexibilidade e desempenho, destacando desafios inerentes à adoção de padrões Unicode em sistemas de tradução em Braille.

eBraille: um programa baseado na web para tradução de texto em Japonês para Braille

O sistema eBraille (**SUGANO et al., 2010**), desenvolvido no Japão, introduziu o conceito de tradução em Braille via web, utilizando uma arquitetura cliente-servidor. Apesar de inovador, o sistema limitava-se à submissão e ao recebimento de arquivos, sem permitir edição interativa do conteúdo. Ainda assim, os resultados indicaram níveis de precisão comparáveis ou superiores a outros sistemas existentes, especialmente após a inclusão de um vocabulário médico especializado.

Uma ferramenta para notação musical em Braille

A dissertação defendida por **TOFANI, 2012** aborda os desafios da musicografia em Braille e descreve o desenvolvimento do software livre Delius, projetado para a edição

e a transcrição de partituras. A ferramenta utiliza um emulador de teclado Perkins, segue o padrão arquitetural MVC e permite exportação no formato *MusicXML*, garantindo interoperabilidade com editores musicais convencionais.

Os resultados demonstram que o sistema promove a autonomia do usuário cego e reforça a importância do código aberto para a evolução colaborativa das tecnologias assistivas. O Delius destaca-se como referência para o desenvolvimento de editores Braille que integrem a escrita tátil à visualização gráfica.

2.5 Considerações Finais

Apesar da reconhecida importância do livro em Braille para a educação de pessoas cegas, observa-se uma escassez de pesquisas, especialmente no que diz respeito ao desenvolvimento de soluções de software. O Braille Fácil, embora amplamente adotado e distribuído gratuitamente, apresenta limitações estruturais decorrentes de sua arquitetura legada e de seu código proprietário, o que dificulta sua evolução tecnológica.

Nesse contexto, o BrailleÁgil surge com o propósito de enfrentar dois desafios centrais da área: a necessidade de inovação tecnológica e a promoção da liberdade de software. A proposta de visualização em tempo real oferece *feedback* imediato ao transcritor, otimizando o fluxo de trabalho, enquanto a compatibilidade com os comandos do Braille Fácil contribui para reduzir a curva de aprendizado e aumentar a aceitação do sistema.

Espera-se que este trabalho contribua de forma relevante para a área de transcrição em Braille, resultando em uma ferramenta útil, acessível e alinhada às demandas contemporâneas, capaz de agilizar a produção de materiais táteis e de fortalecer a inclusão educacional de pessoas cegas.

Capítulo 3

O sistema BrailleÁgil

Como discutido anteriormente, o cenário de transcrição em Braille no Brasil apresenta-se estagnado, marcado pela predominância de um *software* desenvolvido em 1999 que, além de possuir código proprietário, foi implementado em Delphi (INPI, 2018). Tal contexto dificulta a participação de desenvolvedores independentes, bem como a evolução, modificação e adaptação da ferramenta a diferentes contextos de uso e a tecnologias mais recentes.

Diante desse cenário, a solução proposta neste trabalho consiste no desenvolvimento de um novo *software* de transcrição em Braille. O principal foco da solução é a visualização simultânea do texto em formato convencional e em Braille. No *software* Braille Fácil, por exemplo, é necessário atualizar manualmente a visualização de impressão sempre que se deseja verificar o resultado de uma modificação no texto. Em casos mais complexos, como a transcrição de conteúdos da área de Química ou a adaptação de esquemas visuais, torna-se necessário ajustar manualmente formas e linhas. Considerando que a alteração de um único caractere pode impactar toda a estrutura do conteúdo transcrita, é fundamental dispor de uma visualização fiel e imediata do Braille gerado, a fim de garantir maior agilidade e precisão no processo de transcrição.

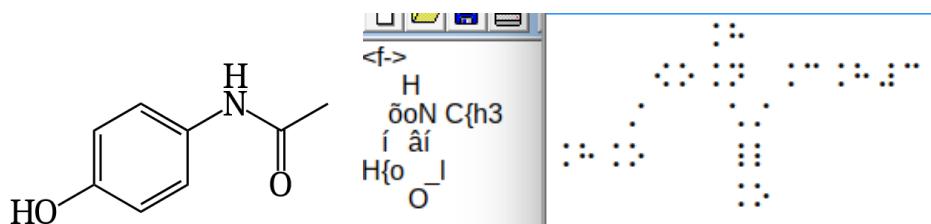


Figura 3.1: Fórmula estrutural do paracetamol e sua representação no Braille Fácil

Nesse contexto, o **BrailleÁgil** surge como uma resposta às limitações identificadas, ao propor um ambiente de edição moderno, desenvolvido em código aberto e orientado à agilidade. Seu principal diferencial reside na visualização simultânea e em tempo real, permitindo que o transcritor valide instantaneamente a diagramação do conteúdo, reduzindo a ocorrência de erros e otimizando o fluxo de trabalho.

3.1 P blico-alvo do Braille gil

O p blico-alvo principal do Braille gil   composto por profissionais videntes que atuam na transcri o e na produ o de materiais em Braille. Esse grupo abrange transcritores, revisores, funcion rios de institutos para pessoas cegas, bibliotecas especializadas, gr ficas em Braille e editoras que necessitam converter grandes volumes de texto para o sistema t til. O sistema foi concebido para otimizar e agilizar o fluxo de trabalho desses especialistas, reduzindo significativamente o tempo e o esfor o necess rios   produ o de livros, apostilas e outros documentos acess veis. Nesse contexto, o foco da ferramenta est  na redu o de etapas manuais e repetitivas, promovendo o aumento da produtividade e da escala de produ o, sem comprometer a qualidade e a prec o t cnica exigidas na transcri o em Braille.

Embora o Braille gil tamb m possa ser utilizado pontualmente por pessoas cegas ou por seus familiares para tarefas simples de impress o, sua arquitetura e conjunto de funcionalidades foram planejados prioritariamente para atender  s demandas t cnicas e operacionais do ambiente profissional. A proposta central do sistema   constituir-se como um instrumento de trabalho robusto, capaz de aprimorar e complementar solu es j  existentes, como o Braille F cil, desenvolvido pelo IBC. Dessa forma, embora o impacto final da ferramenta seja a amplia o do acervo de materiais acess veis, beneficiando toda a comunidade de leitores com defici ncia visual, o usu rio direto e priorit rio do *software*   o especialista que domina as normas de transcri o e busca maior efici ncia operacional em processos sistem ticos e de larga escala.

3.2 Requisitos Funcionais do Braille gil

Antes do in cio do desenvolvimento, tornou-se necess ria a defini o dos requisitos funcionais do *software*. Esses requisitos correspondem  s funcionalidades que o sistema deve oferecer e est o diretamente relacionados  s a o es que o usu rio pode executar. Em outras palavras, cada requisito funcional descreve uma capacidade espec fica que o sistema deve oferecer ao usu rio.

No contexto do *Braille gil*, os requisitos funcionais foram definidos para expandir e aprimorar as solu es existentes. Contudo, considerando o escopo deste trabalho, foram priorizados os requisitos considerados imprescind veis para o desenvolvimento de um *software* capaz de apoiar efetivamente a transcri o em Braille, ainda que em est gios iniciais de pesquisa e valid o . Adicionalmente, foram inclu dos requisitos de implementa o relativamente simples, que agregam valor significativo ao produto final. Os requisitos funcionais definidos s o apresentados a seguir:

- RF1 O Braille gil deve suportar s mbolos da l ngua portuguesa (acentos, cedilha e h fen). Os acentos e sinais gr ficos da l ngua portuguesa s o imprescind veis, pois, al m de garantir a corre o gramatical, podem ser utilizados na composi o  de desenhos, esquemas e represent es espec ficas.
- RF2 O Braille gil deve representar corretamente letras mai sculas na visualiza o em Braille.

No sistema Braille, há símbolos específicos para indicar letras maiúsculas. Quando uma palavra é escrita integralmente em caixa alta, dois sinais de maiúsculas devem preceder a palavra. Esses sinais ocupam espaço na diagramação e, portanto, seu uso correto é essencial no processo de transcrição.

RF3 O BrailleÁgil deve permitir a quebra automática de linha, respeitando a separação silábica da língua portuguesa.

A quebra automática de linha com separação silábica é uma funcionalidade essencial tanto para a economia de espaço nos livros em Braille quanto para a agilidade do trabalho do transcritor. No entanto, determinados materiais não permitem a separação silábica, sendo necessário que o sistema permita o controle da ativação dessa funcionalidade.

RF4 O BrailleÁgil deve disponibilizar um menu completo de configuração de página, incluindo opções como número de células por linha, número de linhas por página e definição de título em todas as páginas ou apenas em páginas ímpares.

Esses recursos de configuração são fundamentais para a diagramação correta do conteúdo, uma vez que diferentes tipos de material exigem parâmetros de página distintos. Além disso, impressoras Braille podem apresentar variações no tamanho físico das páginas.

RF5 O BrailleÁgil deve realizar quebras automáticas de página conforme as configurações definidas pelo usuário.

A quebra de página é um elemento essencial da diagramação em Braille e segue regras específicas de paginação. Por exemplo, o título deve estar posicionado na mesma página que, no mínimo, duas linhas de texto subsequentes (MEC, 2018). Dessa forma, o sistema deve ser capaz de gerar quebras automáticas corretamente, além de permitir a inserção de quebras manuais quando necessário.

RF6 O BrailleÁgil deve suportar a abertura e o fechamento de *tags* para a execução de ações específicas, como o recuo de dois espaços e a desativação da separação silábica. Esse requisito visa garantir a compatibilidade com o *software* Braille Fácil, adotando os mesmos padrões de marcação. As *tags*, delimitadas pelos caracteres < >, devem ser interpretadas pelo sistema como comandos, não sendo exibidas na visualização em Braille.

Além dos requisitos funcionais, foram definidos requisitos não funcionais, que não estão diretamente relacionados às funcionalidades do sistema, mas sim às suas características globais de qualidade, como desempenho, usabilidade e interface. Esses requisitos foram estabelecidos com o objetivo de assegurar o adequado funcionamento do sistema mesmo em computadores com recursos limitados, fator relevante para sua adoção em diferentes contextos institucionais. Os requisitos não funcionais definidos são apresentados a seguir:

RNF1 O BrailleÁgil deve apresentar bom desempenho em computadores com configurações básicas, considerando que o sistema lida exclusivamente com processamento de texto e, portanto, demanda baixo uso de recursos computacionais.

RNF2 O BrailleÁgil deve possuir uma interface simples, intuitiva e de fácil utilização.

RNF3 O BrailleÁgil deve possibilitar um fluxo de trabalho mais ágil do que as soluções

disponíveis atualmente.

3.3 Especificações Técnicas

Inicialmente, a abordagem adotada consistiu na modificação do *software BrailleZephyr*. Esse programa é um simulador de máquina Perkins, um equipamento físico utilizado para a marcação de pontos Braille em papel. O *software* oferece visualização simultânea de texto convencional e em Braille, permitindo tanto a conversão de texto para Braille quanto a digitação direta no estilo Perkins. Em um primeiro momento, essa estratégia mostrou-se promissora, uma vez que funcionalidades relevantes já estavam implementadas. No entanto, a ausência de documentação adequada sobre o funcionamento interno do *software*, aliada à baixa familiaridade com a linguagem Java, na qual o programa foi desenvolvido, resultou em atrasos significativos no andamento do trabalho.

Diante dessas dificuldades, a partir do estudo da biblioteca Tkinter em Python, constatou-se que a alternativa mais adequada seria o desenvolvimento de um *software* inteiramente novo, utilizando uma linguagem mais familiar e de fácil compreensão. Assim, o sistema foi implementado em Python, versão 3.12.3, por meio da biblioteca Tkinter. Essa biblioteca permite a criação de interfaces gráficas de forma simplificada, utilizando recursos nativos do sistema operacional do usuário e tratando cada elemento da interface como um item posicionado em uma matriz. Dessa forma, é possível criar botões e áreas de texto, sendo que o Tkinter já disponibiliza, de maneira direta, componentes de edição de texto, menus e outros elementos de interface, definindo sua renderização por meio do método `.grid()`, que especifica a linha e a coluna em que cada elemento deve ser exibido.

A escolha da biblioteca Tkinter justifica-se por sua natureza leve e nativa, o que garante que o *software* não exija recursos de hardware excessivos. Além disso, ao utilizar *widgets* nativos dos sistemas operacionais Ubuntu, Windows e macOS, a interface mantém uma aparência familiar ao usuário, o que contribui para a usabilidade da ferramenta. Como bibliotecas auxiliares ao desenvolvimento, foram utilizadas: *Pyphen*, responsável pela separação silábica; *re*, utilizada para o processamento de expressões regulares e para a identificação das *tags* de marcação do programa; e *roman*, que auxilia na geração da numeração romana empregada na paginação do preâmbulo do livro.

Em sua implementação inicial, o *software* adotava uma arquitetura imperativa, composta por uma função principal e funções auxiliares que, em conjunto com os comandos da biblioteca de interface gráfica, permitiam a execução do programa. Contudo, devido à natureza de um editor de textos, um tipo de aplicação que demanda diversas variáveis de controle relacionadas à formatação, essa abordagem implicava o uso extenso de variáveis globais ou a passagem de uma grande quantidade de parâmetros entre funções. Diante disso, optou-se pela adoção de uma arquitetura orientada a objetos.

Na implementação final, o editor de texto foi modelado como uma classe instanciada apenas uma vez. Em substituição às variáveis globais, passaram a ser utilizados os atributos do objeto. Além disso, dado que não há múltiplas instâncias do editor, as funções utilitárias foram definidas como métodos da própria classe. Dessa forma, evita-se a necessidade de lidar com múltiplos parâmetros em chamadas de função, sendo suficiente acessar ou modificar os atributos do objeto por meio da palavra-chave *self*.

O programa, então, organiza a interface em duas áreas principais de texto. À esquerda, localiza-se o campo de edição, enquanto, à direita, encontra-se a área de visualização. A cada vez que uma tecla do teclado do usuário é liberada, o sistema lê todo o conteúdo textual e o processa conforme os parâmetros definidos pelo usuário, como o tamanho da página, o tamanho da linha, a ativação ou não da separação silábica e a presença de cabeçalho. Adicionalmente, o sistema processa as *tags* inseridas pelo usuário no texto, de forma semelhante a uma linguagem de marcação, permitindo a definição de quebras de página, numeração de páginas e ajustes de formatação.

Durante o processamento de cada palavra, o sistema aplica as regras necessárias, como a inserção de símbolos de letras maiúsculas e a paginação, e insere o resultado na área de visualização em Braille. Essa visualização é realizada por meio de uma fonte Braille.¹ Essa abordagem facilita a alternância entre a visualização em Braille e em texto convencional, bastando alterar a fonte exibida na interface, preservando a fidelidade do conteúdo. Tal recurso é especialmente relevante para transcritores com menor familiaridade com a leitura direta em Braille.

Devido à forma como o programa reprocessa o texto a cada interação, houve preocupação com o desempenho do sistema em textos grandes. Dessa forma, foi implementada a possibilidade de atualizar manualmente ou automaticamente a visualização em Braille. Caso seja selecionada a opção manual, o botão de atualização é adicionado à barra de ferramentas, onde o usuário atualiza o Braille gerado. Posteriormente, o usuário pode retornar à atualização automática pelo mesmo botão.

Além disso, destaca-se novamente a utilização das bibliotecas auxiliares *re* para o processamento do texto e das *tags* por meio de expressões regulares e de *Pyphen*, que utiliza dicionários do LibreOffice para a separação silábica. Essa última mostrou-se essencial para viabilizar a separação silábica no contexto do *software* desenvolvido.

O diagrama a seguir apresenta o fluxo de processamento do BrailleÁgil.

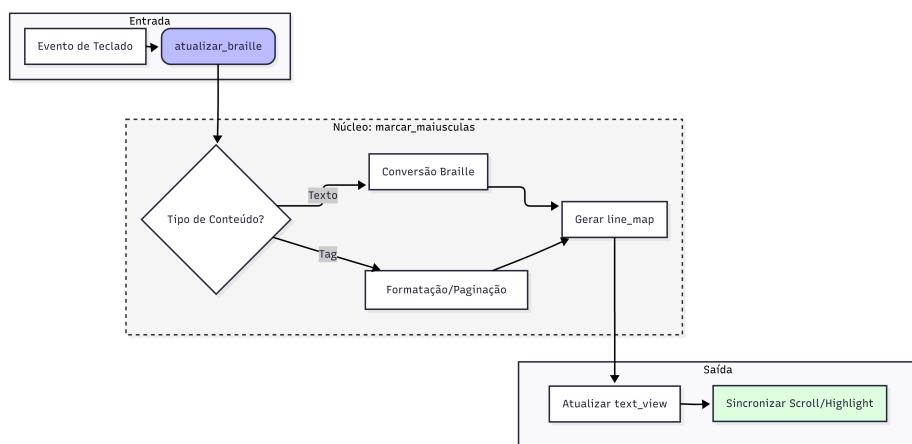


Figura 3.2: Fluxograma de processamento de texto do BrailleÁgil

¹ A fonte utilizada é a BraillePT, desenvolvida em Portugal sob a licença *SIL Open Font*, versão 1.1, com algumas alterações realizadas pelo autor.

3.4 Arquitetura do BrailleÁgil

O algoritmo do BrailleÁgil foi concebido com base em uma arquitetura monolítica orientada a eventos, característica comum em aplicações de interface gráfica interativa, conforme apresentado na Figura 3.3. Essa abordagem foi adotada com o objetivo de favorecer a simplicidade estrutural do sistema, bem como facilitar sua compreensão, manutenção e evolução futura.

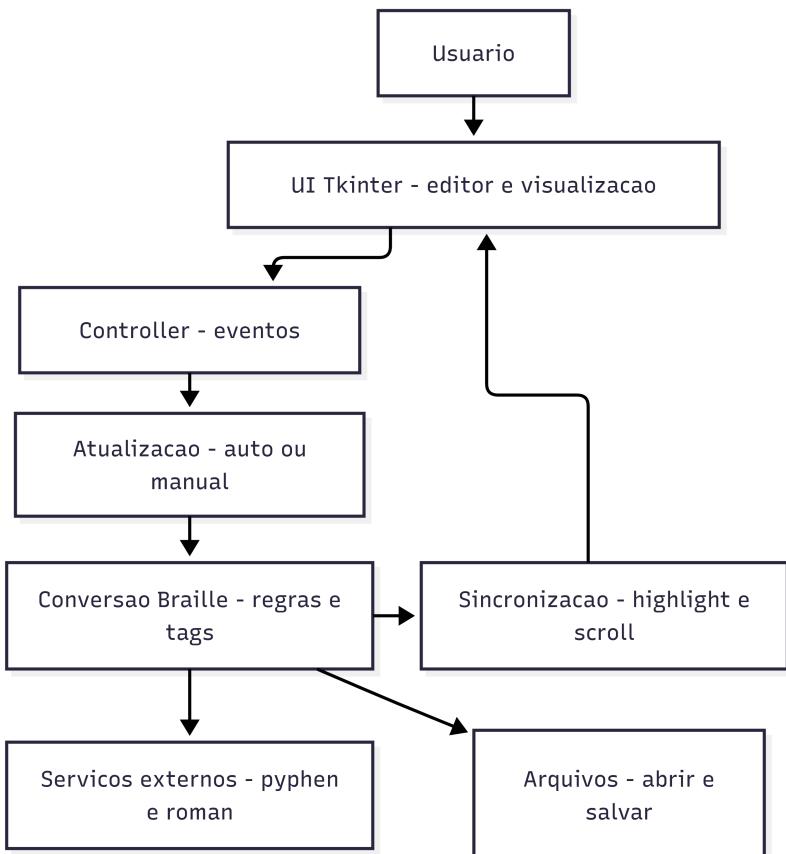


Figura 3.3: Representação Visual do Código do Sistema BrailleÁgil

A arquitetura do sistema pode ser compreendida a partir de uma divisão conceitual em camadas funcionais, ainda que estas coexistam em um único artefato executável. A camada de apresentação é responsável pela interface gráfica do usuário, abrangendo campos de edição de texto, a visualização do conteúdo em Braille, além de botões, menus e demais elementos visuais de apoio. Essa camada foi implementada por meio dos *widgets* da biblioteca Tkinter e constitui o principal ponto de interação entre o usuário e o sistema.

A camada de controle atua no tratamento dos eventos gerados pela interface gráfica, como entradas via teclado, cliques do mouse e alternância entre os modos de atualização automática e manual da transcrição. Essa camada desempenha o papel de intermediária entre a interface e o núcleo de processamento, sendo responsável por coordenar o fluxo de execução e determinar quando e como a conversão do texto em Braille deve ser realizada.

O núcleo de processamento concentra a lógica central do sistema. Nele estão imple-

mentadas as regras de transcrição Braille, incluindo o tratamento de letras maiúsculas, números, símbolos especiais e a interpretação de marcações específicas inseridas no texto pelo usuário. Adicionalmente, esse núcleo é responsável pela diagramação do conteúdo, realizando a quebra de linhas, a organização em páginas e a geração da representação Braille final apresentada ao usuário.

De forma complementar, o sistema incorpora um mecanismo de sincronização entre o texto original e sua respectiva visualização em Braille, possibilitando o destaque da linha correspondente e a navegação bidirecional entre ambas as representações. Há ainda um módulo dedicado à persistência de dados, que viabiliza a abertura e o salvamento de arquivos de texto, bem como a integração com bibliotecas externas que oferecem funcionalidades adicionais, como hifenização e numeração de páginas.

3.5 Recursos do BrailleÁgil

Para garantir compatibilidade com o Braille Fácil, algumas funcionalidades do BrailleÁgil foram projetadas de forma semelhante às do *software* original. Essa estratégia contribui para reduzir a curva de aprendizado dos transcritores, ao mesmo tempo em que aproveita conceitos já consolidados no cenário da transcrição em Braille.

3.5.1 Tags

O BrailleÁgil processa, por meio de expressões regulares, as *tags* inseridas pelo usuário, definidas pelos caracteres < >. Essas *tags* não são exibidas na visualização do texto, o que torna os símbolos < > adequados também para a inserção de comentários no documento, de forma semelhante ao que ocorre no *software* Braille Fácil.

Quebra de página

A tag <P> permite a realização de quebras de página manualmente. O uso dessa *tag* evita a necessidade de inserir múltiplas quebras de linha até o final da página corrente. Em textos corridos, por exemplo, o título não deve ser separado do corpo do texto. Caso não haja espaço suficiente para acomodar ao menos duas linhas após o título, este deve ser deslocado para a página seguinte. Assim, para evitar a inserção manual de diversas quebras de linha, utiliza-se a tag <P> (ou <p>).

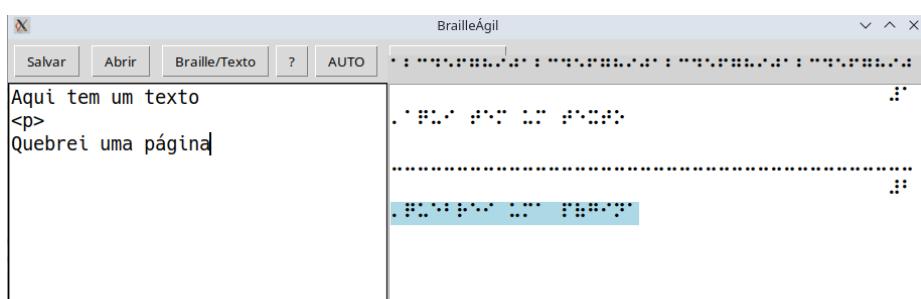


Figura 3.4: Exemplo de quebra de página com <p>

Recuo de dois espaços

A tag `<r+>` inicia um trecho de texto com recuo de dois espaços em relação à margem. Dessa forma, todas as linhas de um parágrafo, com exceção da primeira, permanecem deslocadas dois espaços em relação à margem esquerda. Em livros Braille, essa formatação é obrigatória em citações, títulos e, no caso de livros didáticos, em perguntas e exercícios.

Como o texto é processado de forma linear, as tags `<r+>` e `<r->` alteram uma variável interna que controla o recuo. Assim, basta inserir a tag correspondente no ponto desejado, sem a necessidade de controlar explicitamente a abertura ou o fechamento de um bloco anterior. Ao identificar o comando `<r+>`, o programa inicia um trecho de texto recuado; de forma análoga, ao identificar o comando `<r->`, o texto passa a ser alinhado à margem.

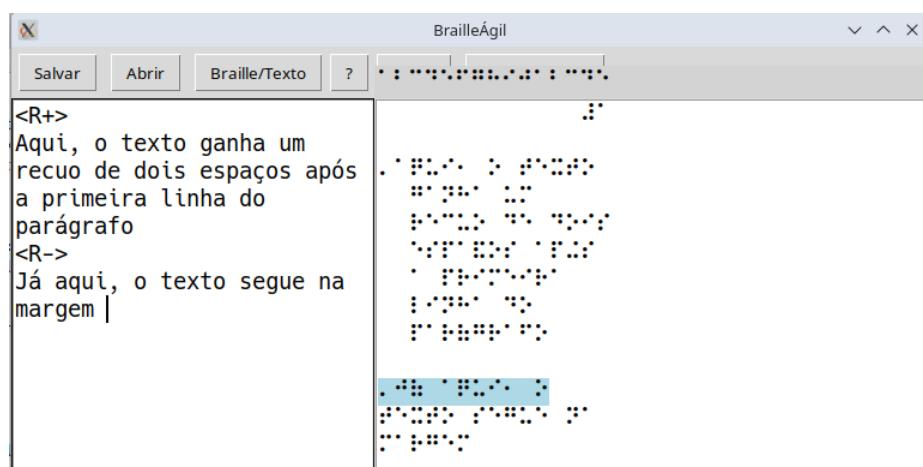


Figura 3.5: Exemplo de recuo de dois espaços

Paginação Braille

Um livro em Braille possui um limite prático de páginas, de modo a evitar volumes excessivamente grandes e difíceis de transportar. Por esse motivo, é comum que uma obra em Braille seja dividida em múltiplos volumes. Nesses casos, é necessário definir manualmente a numeração da página a ser exibida no cabeçalho. Para esse fim, o sistema disponibiliza as tags `<t*XX>` e `<t+XX>`, em que XX representa o número inicial da paginação desejada.

A tag `<t*>` inicia a paginação em algarismos romanos, tradicionalmente utilizada nas páginas iniciais do documento. Em cada volume Braille, é obrigatório incluir uma ficha técnica e um sumário do volume, sendo que o primeiro volume deve conter, adicionalmente, um sumário geral de toda a obra. Essa seção inicial do livro, que utiliza algarismos romanos, é processada com o auxílio da biblioteca *roman* da linguagem Python.

Já a tag `<t+>` inicia a paginação em algarismos indo-árabicos, sendo empregada na numeração do corpo principal do texto. Ressalta-se que a numeração gerada a partir das tags `<t*>` e `<t+>` é incrementada automaticamente a cada quebra de página, seja ela automática ou manual.

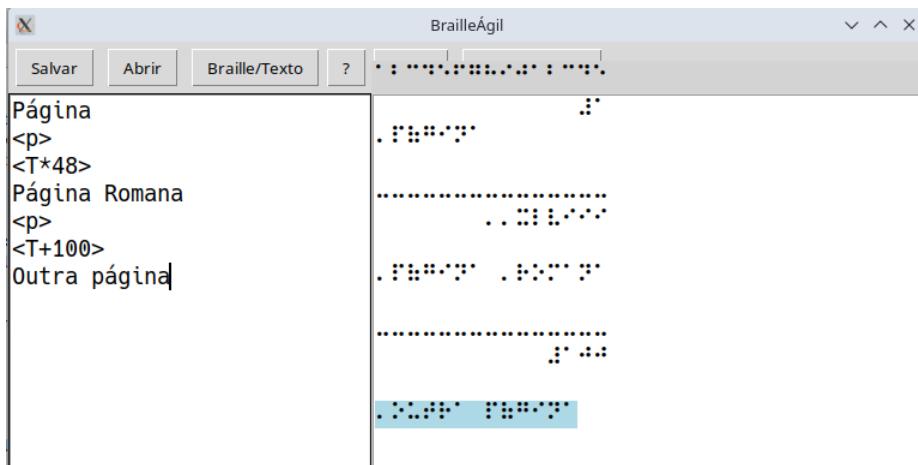


Figura 3.6: Exemplo de paginação manual em algarismos romanos e indo-arábicos

Paginação original

Quando o usuário insere uma *tag* com apenas um número, o valor é exibido no cabeçalho à esquerda da página. Esse recurso é especialmente relevante em livros didáticos, nos quais a referência utilizada em sala de aula corresponde à numeração das páginas do livro impresso em tinta. Dessa forma, o estudante com deficiência visual pode identificar com precisão a página correspondente do livro convencional em sua versão em Braille.

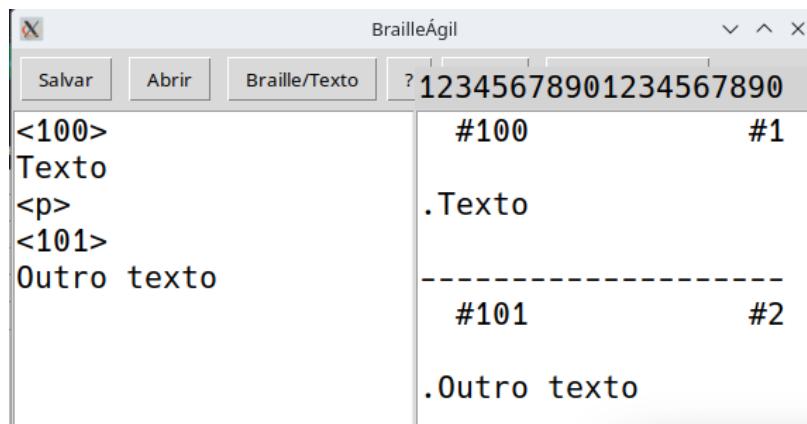


Figura 3.7: Exemplo de paginação correspondente ao livro em tinta

Formatação

O processamento padrão do texto ignora espaços excedentes entre palavras. Entretanto, em determinadas situações, como na transcrição de tabelas, fórmulas químicas ou esquemas visuais, pode ser necessário desativar essa formatação automática. Para esse propósito, as *tags* `<f->` e `<f+>` funcionam de maneira análoga às *tags* de recuo, alterando uma variável interna que controla o modo de formatação do texto.

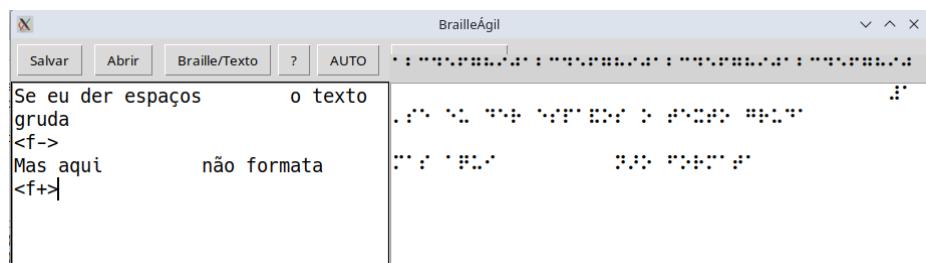


Figura 3.8: Exemplo de formatação automática e sua desativação

3.5.2 Configuração da página

O sistema disponibiliza opções básicas de configuração de página, permitindo definir o tamanho da linha, o número de linhas por página e a inserção de cabeçalho em todas as páginas ou apenas nas páginas pares. O cabeçalho pode conter um texto definido pelo usuário e, obrigatoriamente, a numeração das páginas. Adicionalmente, é possível controlar a ativação ou não da separação silábica.

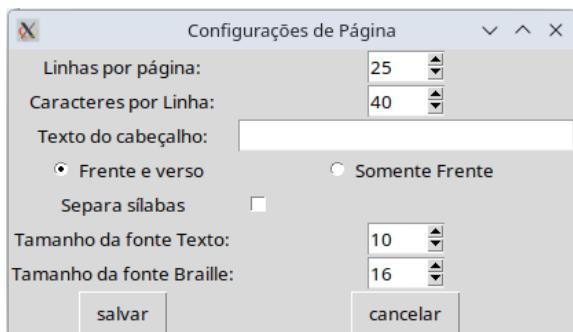


Figura 3.9: Menu de configuração de página

Além das configurações de página, o programa permite a personalização da interface, possibilitando o ajuste do tamanho das fontes tanto na área de edição quanto na área de visualização.

3.5.3 Destaque e posicionamento de texto

No BrailleÁgil, a área de visualização opera de forma integrada à área de edição. Por esse motivo, a linha atualmente em edição recebe destaque em azul na visualização, permitindo que o transcritor identifique com precisão o ponto exato do texto que está sendo modificado.

De forma complementar, caso o transcritor identifique, na visualização, um trecho que necessite de edição, é possível posicionar o cursor diretamente na linha correspondente da área de edição por meio de um duplo clique. Esse recurso, inspirado no ambiente L^AT_EX, é particularmente útil durante o processo de revisão. Em soluções tradicionais, como o Braille Fácil, a revisão realizada a partir da visualização exige a busca manual do trecho correspondente no editor de texto, uma vez que não há um mecanismo equivalente de sincronização.

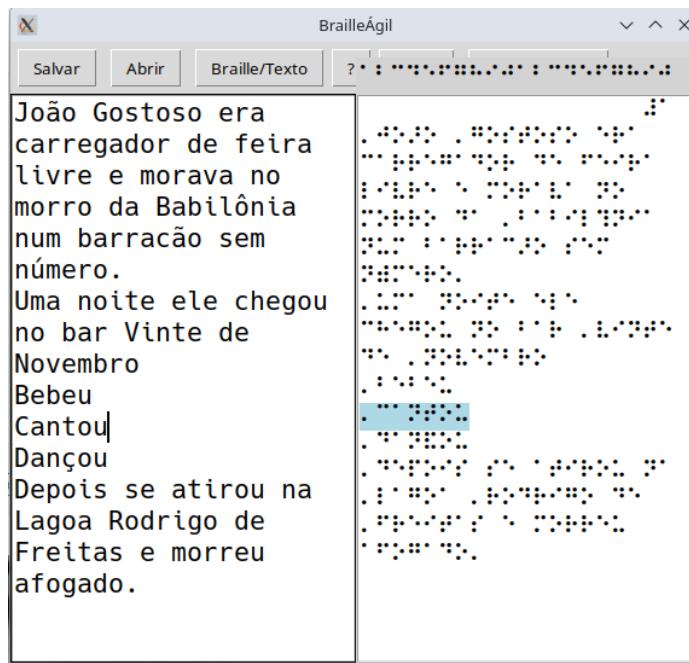


Figura 3.10: Exemplo de destaque da linha em edição

3.5.4 Separação de sílabas

Para a separação silábica, foi empregada a biblioteca *Pyphen*, que utiliza arquivos de dicionário do LibreOffice para dividir palavras em diferentes idiomas. O sistema utiliza a função *wrap*, que recebe como parâmetros a palavra a ser processada e um valor inteiro que representa o tamanho máximo da linha. Caso haja possibilidade de hifenização com comprimento igual ou inferior ao limite informado, a função retorna a palavra já separada em sílabas, pronta para ser inserida na visualização.

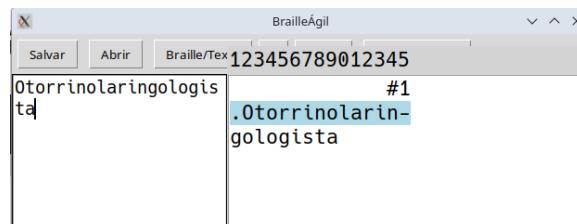


Figura 3.11: Exemplo de separação silábica

3.6 Testes Unitários

Com o objetivo de garantir a confiabilidade do processo de transcrição, foram implementados testes unitários voltados à lógica de conversão e de formatação do sistema. Utilizando a biblioteca *unittest* da linguagem Python, o processo de validação consistiu na submissão de diferentes cadeias de caracteres (*strings*) às funções de processamento, verificando-se se as saídas correspondiam aos resultados esperados.

Os principais casos de teste contemplaram:

- **Conversão de caracteres especiais:** validação de que letras acentuadas e a cedilha são convertidas corretamente para seus respectivos pontos Braille.
- **Lógica de maiúsculas:** verificação da inserção automática do sinal de maiúscula simples e do sinal de caixa alta.
- **Separação silábica:** teste da quebra de linha em palavras complexas, com integração à biblioteca *Pyphen*.
- **Processamento de tags:** garantia de que as marcações, como <P> e <r+>, alteram corretamente o comportamento da diagramação sem serem exibidas na visualização final.

3.7 Considerações Finais

O desenvolvimento do BrailleÁgil partiu de uma premissa prática: a construção de um protótipo funcional mínimo que incorporasse os recursos considerados essenciais pelos transcritores profissionais. A definição dos requisitos funcionais e não funcionais foi orientada por esse propósito, priorizando funcionalidades críticas para o fluxo de trabalho real, como a visualização simultânea, o processamento de *tags* e a separação silábica, em detrimento de funcionalidades mais complexas, previstas para iterações futuras. Essa abordagem permitiu concentrar os esforços na validação da proposta central do sistema: um editor capaz de fornecer *feedback* imediato e preciso durante o processo de transcrição.

O processo de construção foi marcado por escolhas tecnológicas deliberadas, pautadas pela simplicidade e pela acessibilidade, com a adoção da linguagem Python e da biblioteca Tkinter. Paralelamente, houve a evolução da arquitetura do sistema, de um modelo imperativo para uma abordagem orientada a objetos, visando a um gerenciamento mais adequado da complexidade associada ao estado e à formatação do texto. A implementação buscou emular e, simultaneamente, aprimorar a lógica de marcação do Braille Fácil, assegurando compatibilidade com práticas consolidadas, ao mesmo tempo em que introduziu inovações na interface, como a interação bidirecional entre o texto em edição e sua pré-visualização em Braille. Adicionalmente, uma suíte de testes unitários foi desenvolvida em paralelo, com o objetivo de garantir a corretude das conversões e do processamento das regras do sistema Braille, aspecto fundamental para uma ferramenta destinada ao uso em ambientes de produção.

Dessa forma, este capítulo apresentou a construção do *BrailleÁgil*. O sistema, em seu estado atual, configura-se como uma prova de conceito operacional, reunindo os mecanismos fundamentais identificados como necessários para a criação de um ambiente de transcrição mais ágil e eficiente. A existência desse protótipo funcional representa o resultado concreto de um processo de desenvolvimento iterativo, orientado pelas necessidades do usuário profissional, estabelecendo uma base sólida para avaliações práticas e para a evolução futura do *software* a partir do *feedback* direto de seus potenciais usuários.

Capítulo 4

Avaliação do BrailleÁgil

De acordo com **MALHOTRA, 2012**, a pesquisa exploratória tem como objetivo aprofundar a compreensão do problema investigado, contribuindo para uma delimitação mais precisa. Esse tipo de pesquisa caracteriza-se pelo uso de amostras reduzidas e não representativas, sendo recorrente a adoção de técnicas como entrevistas com especialistas da área. Sob essa perspectiva, a avaliação do BrailleÁgil apresenta claramente um caráter exploratório, uma vez que busca compreender percepções iniciais e levantar subsídios para investigações futuras.

Para a condução do estudo, foram realizados contatos por *e-mail* e por telefone com gráficas Braille de diferentes regiões do Brasil, identificadas por meio de buscas no Google. Contudo, essa estratégia mostrou-se pouco efetiva, com exceção do Instituto Benjamin Constant,¹ que demonstrou prontamente interesse em colaborar com a pesquisa. A instituição disponibilizou quatro transcritores de seu quadro funcional, que utilizam diariamente o *software* Braille Fácil há mais de oito anos, constituindo um grupo com elevada experiência prática no domínio avaliado.

A avaliação do BrailleÁgil foi conduzida de forma livre e esclarecida, sem qualquer incentivo financeiro, garantindo aos participantes o direito de interromper sua participação a qualquer momento. Inicialmente, realizou-se uma reunião *on-line* na qual se apresentou o funcionamento do sistema proposto. Na mesma ocasião, foram introduzidas as dez Heurísticas de Nielsen (**NIELSEN e MOLICH, 1990**), adotadas como base para a identificação de problemas de usabilidade, esclarecendo-se que os transcritores deveriam seguir o roteiro proposto e, durante a execução das tarefas, realizar a avaliação heurística do sistema. Em seguida, os participantes instalaram o BrailleÁgil e puderam interagir livremente com o sistema. Após um período inicial de exploração, foi disponibilizado um texto de exemplo, juntamente com um roteiro de tarefas.

Após essa etapa inicial, cada participante executou as tarefas descritas no roteiro a seguir.

Roteiro:

- Abrir a linearização

¹Disponível em: <https://www.gov.br/ibc/pt-br/>

- Configurar a página: 28 linhas por 31 colunas, separa sílabas, inserir um cabeçalho
- Definir a página Braille inicial como 151
- Diagramar as 10 primeiras páginas Braille (verificar títulos, parágrafos etc)
- Salvar
- Fechar o programa

Concluída a execução do roteiro, foi conduzida uma segunda etapa de avaliação por meio do instrumento do Modelo de Aceitação de Tecnologia (TAM), com o objetivo de analisar a percepção dos participantes quanto à utilidade e à facilidade de uso do sistema. Na sequência, apresenta-se o planejamento da avaliação realizada.

4.1 Avaliação Heurística

Essa seção apresenta o planejamento, a execução e os resultados da avaliação heurística realizada para verificar a usabilidade do sistema BrailleÁgil.

4.1.1 Planejamento da Avaliação

A usabilidade é definida pela ISO 9241-11 como a extensão em que um produto pode ser usado por usuários específicos para alcançar objetivos específicos com eficácia, eficiência e satisfação. No âmbito da Interação Humano-Computador (IHC), a Avaliação Heurística, proposta por [NIELSEN e MOLICH, 1990](#), destaca-se como um método de “engenharia de usabilidade de baixo custo”. Ao contrário de testes de desempenho puramente quantitativos, as heurísticas funcionam como princípios de design que orientam a criação de interfaces que respeitam as limitações cognitivas humanas.

De acordo com [BARBOSA e SILVA, 2010](#), a avaliação heurística pode ser aplicada mesmo em estágios iniciais do desenvolvimento, desde que haja uma representação da interface proposta, ainda que seja um protótipo do trabalho final. Essa flexibilidade do artefato avaliado torna a avaliação heurística especialmente útil no escopo deste trabalho, no qual se realiza uma pesquisa exploratória, com o objetivo de identificar cursos relevantes de ação ([MALHOTRA, 2012](#)).

A seguir, são descritos os artefatos resultantes das decisões tomadas para a condução da avaliação heurística.

- **Objetivo:** Identificar problemas de usabilidade no sistema BrailleÁgil, a fim de subsidiar a proposição de melhorias futuras para o projeto.
- **Seleção dos participantes:** Após o contato com diferentes instituições e a apresentação do planejamento da avaliação do sistema, foram indicados quatro transcritores do IBC para participar do estudo. Todos os participantes utilizam o *software* Braille Fácil diariamente em suas atividades profissionais.
- **Seleção dos instrumentos de avaliação:** Com o objetivo de assegurar uma avaliação alinhada ao contexto real de uso, foi definido um processo de diagramação em Braille baseado em experiências prévias e em consultas a transcritores. O texto

de exemplo utilizado corresponde a um livro-texto previamente transcreto. A tabela de avaliação individual foi elaborada com o intuito de minimizar vieses entre os participantes, contemplando as 10 Heurísticas de Nielsen, apresentadas de forma estruturada, acompanhadas de uma breve descrição de cada uma, bem como de um campo destinado à indicação da gravidade dos problemas identificados.

- **Instrumentação:** Roteiro de tarefas (A), texto de exemplo² e tabela de avaliação com as 10 heurísticas de Nielsen(B).

A tabela 4.1 apresenta uma breve explicação das 10 Heurísticas de Nielsen, consideradas na avaliação do BrailleÁgil.

Heurística	Aplicação Prática no Projeto
1. Visibilidade do status do sistema	Barra de status indicando modo de digitação, posição do cursor e se o arquivo está salvo.
2. Correspondência entre o sistema e o mundo real	Uso de termos como “Cela”, “Tabela Braille” e “Pontos”, além da disposição visual que remete à folha de papel.
3. Controle e liberdade do usuário	Presença de comandos de desfazer (Ctrl+Z) e cancelamento de operações de exportação.
4. Consistência e padrões	Manutenção dos atalhos universais (Ctrl+S, Ctrl+C) e atalhos específicos do Braille Fácil (ex: comandos de formatação).
5. Prevenção de erros	Diálogos de confirmação antes de fechar sem salvar ou ao sobreescriver tabelas de tradução.
6. Reconhecimento em vez de recordação	Menus estruturados e listas de arquivos recentes, evitando que o usuário precise decorar caminhos de pastas.
7. Flexibilidade e eficiência de uso	Criação de “atalhos de teclado” para funções frequentes, permitindo que usuários experientes trabalhem mais rápido.
8. Estética e design minimalista	Interface limpa, focada na área de texto, sem ícones desnecessários que poluam a tela.
9. Ajuda os usuários a reconhecerem, diagnosticarem e recuperarem-se de erros	Mensagens de erro claras (ex: “Arquivo não suportado”) em vez de códigos de erro técnicos.
10. Ajuda e documentação	Seção de “Ajuda” ou guia rápido de comandos integrado ao software.

Tabela 4.1: *Explicação das 10 heurísticas de Nielsen*

4.1.2 Execução da Avaliação Heurística

De acordo com **NIELSEN e MOLICH, 1990**, o objetivo central da avaliação heurística é reduzir a carga cognitiva do usuário. Por exemplo, a heurística de “Visibilidade do status do sistema” garante que o usuário nunca se sinta perdido, enquanto a “Correspondência entre o sistema e o mundo real” assegura que a terminologia utilizada (no caso, termos da transcrição em Braille) seja familiar ao especialista. Ao aplicar esse método, busca-se não apenas identificar erros, mas também garantir que a curva de aprendizado seja minimizada, promovendo uma Experiência do Usuário (UX) fluida, na qual a interface se torna “invisível” e o foco recaia totalmente na tarefa produtiva.

A partir disso, realizou-se uma avaliação com quatro especialistas na área de transcrição em Braille. Foi realizada uma reunião *on-line* para esclarecer dúvidas, no dia 24 de novembro de 2025, com a participação do autor e dos quatro avaliadores. Na reunião, explicaram-se o objetivo do estudo e o tipo de avaliação. Também foram explicadas as 10 heurísticas de Nielsen e como se aplicam ao sistema. Em seguida, os instrumentos de avaliação (roteiro,

² Foi utilizado o primeiro capítulo do livro Segue a trilha - História - 7º ano, ed. Palavras: Educação. Mais informações em: <https://palavraseducacao.com.br/catalogo/pnld/segue-a-trilha-historia-7o-ano/>

livro e tabela) foram enviados por *e-mail* aos participantes. Os avaliadores tiveram um dia para interagir com o programa e seguir o roteiro, anotando suas considerações na tabela, que, posteriormente, foi devolvida devidamente preenchida.

4.1.3 Análise dos Resultados

Dentre os quatro avaliadores, observou-se que um deles apresentou dificuldades na condução da avaliação, tendo registrado descrições das heurísticas no campo destinado à identificação dos problemas de usabilidade. Em razão desse equívoco metodológico, os dados dessa avaliação foram desconsiderados. Por sua vez, outro participante não compreendeu adequadamente o formato de preenchimento referente às dez Heurísticas de Nielsen, registrando suas observações em formato de texto contínuo. Nesse caso, os apontamentos foram posteriormente reorganizados e mapeados de forma sistemática às heurísticas correspondentes, de modo a viabilizar sua inclusão na análise. Ao final do processo, foram identificadas violações em nove das dez heurísticas avaliadas. A Tabela 4.2 apresenta a síntese das falhas de usabilidade apontadas pelos participantes.

Heurística	Violação	Gravidade	Participantes
1. Visibilidade do status do sistema	Melhorar <i>feedback</i> de posição no documento (linhas, paginação)	Cosmética	P3
2. Correspondência entre o sistema e o mundo real	Necessidade de mais opções de formatação de texto	Cosmética	P1
	Garantir que caracteres especiais sejam traduzidos corretamente para braille.	Grave	P3
	Grafia e normas Braille aplicadas de forma inadequada	Grave	P1
4. Consistência e padrões	Comandos para maiúsculas e separação silábica estão incorretos	Cosmética	P3
5. Prevenção de erros	Não existe corretor ortográfico	Grave	P1
6. Reconhecimento em vez de recordação	Não há um menu com diversos comandos ou mais usados	Grave	P2
	Há poucas opções de comandos	Cosmética	P1
7. Flexibilidade e eficiência de uso	Faltam muitos recursos relacionados ao Braille	Grave	P1, P3
	Não há barra de ferramentas nem suporte a símbolos técnicos.	Grave	P3
8. Estética e design minimalista	Deveria se priorizar os ícones verdadeiramente importantes	Cosmética	P2
9. Ajuda os usuários a reconhecerem, diagnosticarem e recuperarem-se de erros	Não há indicação de erros	Grave	P2
10. Ajuda e documentação	Ícone de ajuda apresenta poucas opções	Grave	P2

Tabela 4.2: Resultado da Avaliação Heurística do BrailleÁgil

Ao todo, foram identificadas 13 violações de usabilidade por três avaliadores. Embora esse quantitativo seja expressivo, cabe destacar que a avaliação do BrailleÁgil tem caráter exploratório. Nesse tipo de estudo, espera-se um número elevado de apontamentos, uma vez que o objetivo principal é identificar problemas iniciais e fornecer subsídios para orientar as próximas etapas do desenvolvimento do sistema.

No que se refere à Heurística 1 (Visibilidade do status do sistema), foi identificada uma violação de natureza cosmética, relacionada ao *feedback* sobre a posição do usuário no documento. Essa heurística já havia sido parcialmente aprimorada com base em um *feedback* informal de um profissional da área, o que resultou na inclusão de um destaque visual para a linha atualmente em edição. No entanto, sistemas consolidados, como o

Braille Fácil, oferecem um retorno visual mais completo, incluindo a indicação explícita da linha e da coluna ativas no editor de texto, bem como da página correspondente na visualização do documento, conforme ilustrado nas figuras a seguir.:

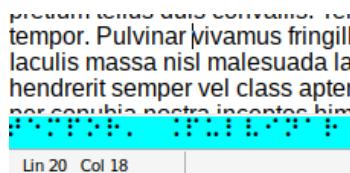


Figura 4.1: Indicação de linha e coluna na área de texto do Braille Fácil



Figura 4.2: Indicação de página e linha atual na visualização do Braille Fácil

Na Heurística 2 (Correspondência entre o sistema e o mundo real), foram identificadas três violações, sendo uma classificada como cosmética e as outras duas como graves. Todas estão relacionadas à formatação do texto em Braille, evidenciando a necessidade de maior rigor na aplicação das normas específicas desse sistema de escrita. Os sinais de letras maiúsculas, por exemplo, foram aplicados de forma inadequada, o que também configurou uma violação da Heurística 4 (Consistência e padronização), uma vez que o comportamento das letras maiúsculas não se mostrou uniforme tanto dentro quanto fora dos comandos de formatação.

A Heurística 5 (Prevenção de erros) apresentou uma violação classificada como grave, associada à ausência de um corretor ortográfico. Tal funcionalidade é considerada essencial no contexto da produção de textos em Braille, pois contribui diretamente para a agilidade do processo e para a redução da carga cognitiva do revisor.

Em relação à Heurística 6 (Reconhecimento em vez de recordação), foram identificadas duas violações. A limitação na oferta de opções de comandos foi considerada uma violação cosmética, enquanto a ausência de um menu estruturado com os comandos mais frequentemente utilizados foi classificada como grave. Durante o desenvolvimento, priorizou-se a inclusão de diversas funcionalidades diretamente na barra de ferramentas, sem uma organização orientada à experiência do usuário. Como consequência, o excesso de botões tornou a interface confusa, dificultando o reconhecimento imediato das ações disponíveis. Essa questão também se refletiu na Heurística 8 (Estética e design minimalistas),

na qual uma violação cosmética indicou a necessidade de priorizar visualmente apenas os ícones essenciais.

A Heurística 7 (Flexibilidade e eficiência de uso) apresentou duas violações, ambas classificadas como graves e relacionadas à ausência de recursos considerados fundamentais. Destacam-se, nesse contexto, a falta de uma barra de ferramentas mais estruturada e a ausência de suporte a símbolos técnicos. Esses apontamentos reforçam os problemas identificados nas heurísticas anteriores e merecem atenção especial, uma vez que o BrailleÁgil foi concebido com o propósito explícito de aumentar a eficiência do trabalho de transcrição em Braille. Assim, tais limitações comprometem diretamente o objetivo central do sistema.

No que concerne à Heurística 9 (Ajuda no reconhecimento, diagnóstico e recuperação de erros), foi identificada uma violação grave, decorrente da ausência de mecanismos de indicação de erros. Além disso, o sistema não oferece recursos que auxiliem o usuário na recuperação após falhas, o que pode prejudicar a confiança e a continuidade do uso.

Por fim, a Heurística 10 (Ajuda e documentação) apresentou uma violação grave, decorrente da ausência de documentação ou de mecanismos de ajuda que orientem o usuário sobre o funcionamento do sistema e suas principais funcionalidades.

A seguir, discute-se o planejamento, a execução e a avaliação dos resultados obtidos com o uso do TAM para avaliar o BrailleÁgil.

4.2 Modelo de Aceitação de Tecnologia (TAM)

O TAM, proposto em 1989, fundamenta-se na Teoria da Ação Racional e tem como objetivo explicar os fatores que influenciam a aceitação e o uso de sistemas computacionais. O principal diferencial do TAM em relação a outros modelos reside em sua capacidade de explicar o comportamento do usuário a partir de dois construtos centrais de natureza perceptiva: a Utilidade Percebida (*Perceived Usefulness* – PU) e a Facilidade de Uso Percebida (*Perceived Ease of Use* – PEOU).

De acordo com [DAVIS, 1989](#), a Utilidade Percebida apresenta a maior correlação com o uso efetivo do sistema, uma vez que usuários tendem a adotar tecnologias que proporcionem ganhos concretos de desempenho, produtividade ou eficiência. Entretanto, a Facilidade de Uso Percebida desempenha um papel mediador fundamental nesse processo, pois interfaces de difícil operação podem gerar resistência à adoção e comprometer a intenção de uso, mesmo quando o sistema oferece benefícios funcionais relevantes.

4.2.1 Planejamento da Avaliação TAM

No contexto de transição entre produtos de *software* consolidados, como a migração do Braille Fácil para o BrailleÁgil, o TAM permite analisar se a facilidade percebida, decorrente da compatibilidade entre os comandos, se traduz em utilidade percebida suficiente para justificar a substituição do sistema legado pelos transcritores. Com base nesse pressuposto, foi definido o seguinte planejamento de avaliação:

- **Objetivo:** Avaliar o nível de aceitação do novo *software* entre profissionais da área de transcrição em Braille.

- **Seleção dos participantes:** Realização de contatos por *e-mail* e pelo WhatsApp com diversas gráficas Braille, convidando profissionais para uma sessão de teste do sistema, seguida da aplicação do instrumento de avaliação.
- **Seleção dos instrumentos de avaliação:** Inspirada no trabalho original de **DAVIS, 1989**, foram selecionadas as 12 afirmações propostas no modelo original do TAM, traduzidas e adaptadas ao contexto do BrailleÁgil. Para viabilizar a coleta de dados, foi elaborado um formulário na plataforma *Google Forms*. O roteiro de tarefas e o texto de exemplo utilizados foram os mesmos empregados na avaliação heurística, garantindo consistência metodológica entre as avaliações.
- **Instrumentação:** Formulário on-line, com seis afirmações referentes ao construto Utilidade Percebida (PU) e seis afirmações referentes ao construto Facilidade de Uso Percebida (PEOU) (C).

4.2.2 Execução da Avaliação TAM

Para a aplicação do TAM no contexto do BrailleÁgil, foram adaptadas as questões originais propostas por **DAVIS, 1989** em sua avaliação do sistema *CHART-MASTER*, mantendo-se a mesma escala de mensuração. Utilizou-se uma escala de Likert de 7 pontos, variando de 1 (*Muito improvável*) a 7 (*Muito provável*). As afirmações foram traduzidas e contextualizadas de modo a refletir as características específicas do sistema avaliado.

A utilização da escala de Likert com número ímpar de pontos permite ao respondente expressar uma avaliação neutra. Além disso, esse tipo de escala apresenta facilidade de construção e aplicação, bem como boa compreensibilidade por parte dos participantes, ainda que exija maior tempo de resposta devido à necessidade de leitura e interpretação das afirmações (**MALHOTRA, 2012**).

A aplicação do instrumento consistiu na experimentação do sistema a partir de um roteiro de tarefas previamente definido (A). Para o TAM, foi adotado o mesmo roteiro utilizado na avaliação heurística, assegurando uniformidade no processo de uso do sistema antes da avaliação perceptiva. O modelo do TAM foi aplicado em duas partes: a primeira, voltada à avaliação da Utilidade Percebida, e a segunda, à avaliação da Facilidade de Uso Percebida. Ao todo, participaram da avaliação quatro transcritores com mais de oito anos de experiência profissional.

A coleta de dados foi realizada por meio de um formulário on-line, no qual os participantes atribuíram valores a cada afirmação utilizando a escala de Likert, variando de 1 (*Muito improvável*) a 7 (*Muito provável*), conforme apresentado a seguir.

4.3 Análise dos Resultados

Essa seção apresenta os resultados da avaliação utilizando o instrumento TAM. Os resultados são apresentados por categoria do instrumento.

Utilidade Percebida (PU)

O primeiro construto analisado foi a Utilidade Percebida (PU), definida como o grau em que o usuário acredita que o uso do sistema avaliado pode melhorar seu desempenho no trabalho (DAVIS, 1989). No contexto da avaliação do BrailleÁgil, as afirmações relacionadas a esse construto, bem como suas respectivas médias, são apresentadas na tabela a seguir:

Descrição	Média
Utilizar o BrailleÁgil no meu trabalho me permitiria cumprir minhas tarefas mais rapidamente	5,25
Utilizar o BrailleÁgil melhoraria minha performance no trabalho	5,25
Utilizar o BrailleÁgil no meu trabalho melhoraria minha produtividade	5,5
Utilizar o BrailleÁgil aumentaria minha efetividade no trabalho	5,25
Utilizar o BrailleÁgil tornaria meu trabalho mais fácil	5,5
Eu acho que o BrailleÁgil seria útil em meu trabalho	6,25

Tabela 4.3: Média da pontuação da categoria PU

Todas as afirmações relacionadas ao construto PU apresentaram médias superiores a 5. De modo geral, as avaliações situaram-se entre valores neutros e positivos (acima de 4), indicando uma percepção favorável quanto ao impacto do BrailleÁgil no desempenho profissional. A única exceção ocorreu na afirmação 5, “Utilizar o BrailleÁgil tornaria meu trabalho mais fácil”, que recebeu as notas 3, 6, 6 e 7. Nesse caso, observa-se uma avaliação negativa pontual, na qual um dos participantes manifestou a percepção de que o sistema poderia não facilitar seu trabalho.

Na primeira afirmação, “Utilizar o BrailleÁgil no meu trabalho me permite cumprir minhas tarefas mais rapidamente”, atribuíram-se as notas 4, 5, 6 e 6, todas classificadas como neutras ou positivas. As afirmações 2, 3 e 4 apresentaram avaliações igualmente favoráveis. Na segunda afirmação, “Utilizar o BrailleÁgil melhoraria minha performance no trabalho”, as notas atribuídas foram 4, 5, 5 e 7. Na terceira, “Utilizar o BrailleÁgil no meu trabalho melhoraria minha produtividade”, as avaliações foram 4, 5, 6 e 7. Já a quarta afirmação, “Utilizar o BrailleÁgil aumentaria minha efetividade no trabalho”, obteve as notas 4, 5, 5 e 7. A última afirmação de PU, “Eu acho que o BrailleÁgil seria útil em meu trabalho”, foi a mais bem avaliada, obtendo duas notas máximas (5, 6, 7, e 7).

A partir desses resultados, infere-se um bom nível de utilidade percebida do sistema, refletido em uma média geral de 5,5 pontos em uma escala de 1 a 7. A menor pontuação individual registrada foi 3, associada à afirmação sobre a facilidade percebida na execução das tarefas. Ao relacionar esse resultado às respostas obtidas na questão aberta “Qual(is) funcionalidade(s) você gostaria de ver em um novo *software* Braille?”, observa-se que três participantes destacaram a importância da visualização simultânea em Braille e em tinta em tempo real. Considerando que essa funcionalidade constitui a principal inovação proposta pelo BrailleÁgil, é plausível associar os elevados valores de PU à implementação desse recurso, que atende diretamente a uma demanda expressa pelos usuários.

Facilidade de Uso Percebida (PEOU)

De forma análoga, foram avaliadas seis afirmações relacionadas ao construto PEOU, utilizando a mesma escala de Likert, que variava de 1 (Muito improvável) a 7 (Muito provável). As afirmações e seus respectivos resultados são apresentados na Tabela 4.4.

Descrição	Média
Aprender a operar o BrailleÁgil seria fácil para mim	6,0
Eu acharia fácil fazer o que eu quiser fazer com o BrailleÁgil	5,75
Minha interação com o BrailleÁgil seria clara e compreensível	6,25
Eu acharia o BrailleÁgil flexível de se interagir	6,0
Seria fácil me tornar habilidoso no uso do BrailleÁgil	5,75
Eu acharia o BrailleÁgil fácil de se utilizar	5,5

Tabela 4.4: Média da pontuação do construto Facilidade de Uso Percebida

A média das avaliações atribuídas às afirmações relativas ao construto PEOU foi positiva. A primeira afirmação, “Aprender a operar o BrailleÁgil seria fácil para mim”, recebeu as notas 3, 7, 7 e 7, evidenciando uma avaliação negativa pontual contrastando com três avaliações máximas. A segunda afirmação, “Eu acharia fácil fazer o que quisesse com o BrailleÁgil”, foi avaliada com as notas 4, 6, 6 e 7. A terceira, “Minha interação com o BrailleÁgil seria clara e compreensível”, obteve avaliações igualmente favoráveis, com notas 4, 7, 7 e 7.

A quarta afirmação, “Eu acharia o BrailleÁgil flexível de se interagir”, apresentou novamente uma distribuição semelhante à primeira, com uma avaliação negativa (nota 3) e três avaliações máximas. A quinta afirmação, “Seria fácil me tornar habilidoso no uso do BrailleÁgil”, recebeu as notas 4, 5, 7 e 7. Por fim, a sexta afirmação, “Eu acharia o BrailleÁgil fácil de se utilizar”, foi avaliada com as notas 3, 5, 7 e 7.

De modo geral, todas as afirmações de PEOU apresentaram médias superiores a 5. Entretanto, observa-se a presença de discrepâncias nas avaliações de um dos participantes, que atribuiu três notas negativas ao longo do instrumento. Ainda assim, a média geral do construto PEOU foi superior à observada para o construto PU, atingindo 5,875 pontos em uma escala de 1 a 7. Esse resultado se explica pela elevada proporção de avaliações máximas: das 24 respostas possíveis, 14 corresponderam à nota máxima, o que representa aproximadamente 58% do total.

4.4 Discussão

Após a análise dos resultados, observou-se uma avaliação predominantemente positiva do sistema. Ressalta-se que todos os participantes possuem mais de oito anos de experiência profissional na transcrição em Braille e utilizam o *software* Braille Fácil em suas atividades diárias. Nesse contexto, os resultados favoráveis podem ser atribuídos, em grande parte, à compatibilidade do BrailleÁgil com os comandos do Braille Fácil, um sistema consolidado no cenário brasileiro. Essa compatibilidade contribuiu para uma curva de aprendizado mais curta, ao mesmo tempo em que se beneficiou das características de simplicidade e facilidade de uso já reconhecidas no Braille Fácil, inclusive para novos usuários.

Por outro lado, as quatro avaliações negativas identificadas merecem atenção, uma vez que todas se relacionam a aspectos de facilidade de uso. Tais resultados podem estar associados, em primeiro lugar, ao caráter preliminar da versão avaliada, que ainda carece de recursos considerados essenciais em ambientes profissionais. Adicionalmente, a organização da interface, baseada exclusivamente em botões dispostos na tela e sem a presença de menus ou abas estruturadas, foi apontada como um fator que comprometeu a

usabilidade. Essa limitação apresenta relação direta com a violação da heurística de “Ajuda e Documentação”, visto que, embora exista um ícone de ajuda com comandos de marcação, o sistema não dispõe de documentação formal que oriente o usuário de maneira sistemática.

As avaliações também evidenciaram oportunidades relevantes de aprimoramento do sistema. Entre elas, destaca-se a necessidade de compatibilidade direta com impressoras Braille. A abordagem inicialmente concebida, na qual o texto seria preparado para posterior impressão por meio de outro *software*, mostrou-se ineficiente e excessivamente morosa. Nesse sentido, os avaliadores indicaram que a possibilidade de impressão direta a partir do BrailleÁgil representaria um avanço significativo na eficiência operacional.

Com base nos resultados obtidos, destacam-se como melhorias prioritárias a serem implementadas:

- **Ajuda e documentação:** Desenvolvimento de uma documentação clara e acessível, com orientações detalhadas ao usuário, bem como a reorganização dos menus de ajuda, para torná-los mais completos e funcionais.
- **Interface e organização dos comandos:** Reestruturação da interface gráfica, com a introdução de menus e abas organizadas, priorizando os comandos mais utilizados e reduzindo a sobrecarga visual.
- **Funcionalidades profissionais:** Inclusão de recursos essenciais ao fluxo de trabalho dos transcritores, como suporte direto à impressão em Braille.

Ainda assim, os resultados das avaliações quantitativas e qualitativas indicam que o sistema apresenta um elevado nível de aceitação entre os usuários, mesmo atualmente dispondo de um conjunto limitado de funcionalidades. Destaca-se a recorrência de pontuações máximas associadas ao construto Facilidade de Uso Percebida (PEOU). Esse resultado, aliado às respostas obtidas nas questões abertas, que evidenciam a demanda por maior agilidade na produção de textos em Braille por meio de um sistema com visualização em tempo real, sugere não apenas uma percepção positiva quanto à facilidade de uso, mas também uma clara predisposição do público-alvo à adoção da solução proposta.

Por fim, é relevante destacar que, diferentemente do Braille Fácil, o BrailleÁgil é concebido como um *software* de código aberto, o que favorece sua evolução contínua e a criação de versões adaptadas às necessidades específicas de diferentes públicos. Essa característica tende a facilitar tanto a manutenção quanto a incorporação de melhorias de forma mais ágil. Além disso, conforme relatado pelos avaliadores, o Braille Fácil foi desenvolvido por meio de Delphi e é mantido por um único profissional, que também acumula outras atribuições institucionais no IBC. Tal cenário reforça a importância de uma alternativa aberta que ofereça funcionalidades equivalentes, garantindo maior sustentabilidade e escalabilidade ao longo do tempo.

4.5 Considerações Finais

Após a realização das duas etapas de avaliação, foi possível constatar uma boa aceitação do sistema, mesmo em fase inicial de desenvolvimento. As principais inovações propostas pelo BrailleÁgil mostraram-se adequadas, o que se refletiu em um nível satisfatório de

utilidade percebida. Além disso, a estratégia de manter compatibilidade com os comandos de um *software* já consolidado revelou-se positiva, contribuindo para o construto PEOU e resultando em escores mais elevados nesse aspecto.

Por outro lado, a avaliação heurística evidenciou fragilidades associadas à imaturidade do sistema, típicas de versões iniciais de desenvolvimento. Embora tais problemas não tenham comprometido significativamente o uso da ferramenta durante a avaliação, sua correção é considerada prioritária para uma eventual disponibilização do sistema em larga escala.

Capítulo 5

Conclusão

A realidade do mercado de transcrição em Braille no Brasil é marcada por uma notável concentração e estagnação tecnológica (LEMOS e CERQUEIRA, 2014). O Braille Fácil, software amplamente adotado pelas grandes gráficas por ser o único programa nacional para essa finalidade, tornou-se um padrão de fato, mas sofre com um ciclo de atualizações lento: oito anos separaram as duas últimas versões principais, com dependência exclusiva do IBC para evolução e restrição ao sistema Windows, excluindo usuários de outras plataformas.

Além disso, o fluxo de trabalho imposto pelo Braille Fácil implica a atualização manual da visualização de impressão, o que gera ineficiência, especialmente na transcrição de obras extensas ou com elementos complexos, como fórmulas e diagramas, em que qualquer ajuste demanda retrabalho significativo. Esse cenário cria uma clara lacuna para a novas pesquisas no tópico.

Neste contexto, o produto de software proposto neste trabalho, denominado BrailleÁgil, constitui uma contribuição relevante para o cenário. Seu desenvolvimento foi realizado em Python, com a biblioteca Tkinter, e teve como objetivo central superar a principal deficiência operacional identificada: a falta de *feedback* visual imediato para o usuário. A premissa da visualização em tempo real do Braille gerado constitui a inovação central da ferramenta, visando agilizar substancialmente os processos de transcrição e, principalmente, de revisão. A estratégia de adoção foi cuidadosamente planejada para garantir total compatibilidade com as tags de marcação do Braille Fácil. Desse modo, o sistema reduz a curva de aprendizado e a resistência à mudança entre os usuários especializados, que já dominam essa linguagem.

O BrailleÁgil foi desenvolvido por meio de uma metodologia iterativa e, para sua validação, foram realizados testes unitários. Além disso, realizou-se uma avaliação com especialistas para aferir a usabilidade do sistema. Inicialmente, a ferramenta foi apresentada a um especialista da Fundação Laramara e, posteriormente, conduzida uma avaliação com quatro especialistas do IBC, utilizando métodos de avaliação heurística e a aplicação do Modelo de Aceitação de Tecnologia (TAM), os quais forneceram evidências relevantes sobre a qualidade do sistema. Os resultados do TAM indicaram elevados níveis tanto de Utilidade Percebida (PU), com média de 5,5 em uma escala de 7 pontos, quanto de Facilidade de Uso Percebida (PEOU), com média de 5,785 em 7 pontos. Esses resultados corroboram não

apenas a aceitabilidade do BrailleÁgil, mas também a percepção clara de seu valor prático por parte dos usuários finais. Destaca-se, ainda, a avaliação positiva da funcionalidade de visualização simultânea, apontada pelos avaliadores como uma necessidade há muito tempo no contexto de suas atividades.

5.1 Principais Contribuições

Este trabalho apresenta contribuições relevantes tanto do ponto de vista técnico quanto do prático, ao propor e avaliar uma alternativa moderna para apoiar o processo de transcrição em Braille. As principais contribuições podem ser sintetizadas a seguir:

- **Inovação de interface e fluxo de trabalho:** introdução de um mecanismo de visualização em tempo real, que atua diretamente sobre um ponto crítico de ineficiência identificado nas ferramentas atualmente utilizadas, reduzindo retrabalho e melhorando a experiência do usuário.
- **Arquitetura aberta e multiplataforma:** desenvolvimento do BrailleÁgil como uma solução aberta, licenciada sob a GNU GPL 3.0, permitindo auditoria, adaptação e evolução colaborativa pela comunidade. O suporte nativo ao sistema operacional Linux amplia o acesso e a liberdade de escolha tecnológica.
- **Compatibilidade com padrões existentes:** adoção de uma estratégia de compatibilidade com o padrão já utilizado no contexto da transcrição em Braille, o que facilita a transição, reduz a curva de aprendizado e favorece a adoção da ferramenta por profissionais e instituições.
- **Impacto prático no contexto de uso:** demonstração do potencial do BrailleÁgil para tornar o fluxo de trabalho do transcritor mais ágil e eficiente, com perspectivas de redução de tempo e custos na produção de materiais em Braille e de diminuição da barreira de entrada para novos profissionais e instituições de menor porte.

5.2 Trabalhos futuros

Algumas funcionalidades relevantes ficaram fora do escopo deste trabalho, mas representam oportunidades promissoras para futuras evoluções do BrailleÁgil. Entre elas, destaca-se a necessidade de um módulo de ajuda e documentação mais robusto, incluindo um manual claro, completo e descriptivo tanto para os usuários finais quanto para os desenvolvedores, por meio de documentação técnica do código. Outra possibilidade de aprimoramento refere-se à incorporação de um corretor ortográfico, que pode facilitar a digitação de textos em Braille e tornar o processo de revisão mais ágil e menos suscetível a erros.

Além disso, a implementação das demais *tags* e das funcionalidades existentes no Braille Fácil ampliaria a cobertura funcional da ferramenta e aumentaria sua compatibilidade com práticas já consolidadas. O suporte direto a impressoras Braille também se apresenta como um avanço significativo, pois permitiria a realização do fluxo de transcrição de forma integral, do texto à impressão. Por fim, a introdução de mecanismos de automação de

tarefas, como a geração automática de sumários e a divisão automática de volumes em documentos extensos, tem potencial para reduzir significativamente o esforço manual e otimizar o trabalho dos transcritores.

5.3 Considerações finais

O presente trabalho buscou, ao longo de todo seu desenvolvimento, criar uma nova ferramenta que fosse útil aos profissionais de produção Braille. Durante o período de avaliação, os profissionais contatados demonstraram grande interesse no desenvolvimento da ferramenta, contribuindo com *feedbacks* valiosos e pedindo pela continuidade do desenvolvimento. Como descrito, o cenário de *softwares* de transcrição Braille é marcado por uma escassez de ferramentas. Espera-se que esse trabalho inspire novas publicações no futuro, sejam sistemas específicos para cegueira, sejam sistemas acessíveis em geral.

Apêndice A

Roteiro de Avaliação do sistema

Roteiro:

- Abrir a linearização
- Configurar a página: 28 linhas por 31 colunas, separa sílabas, inserir um cabeçalho
- Definir a página Braille inicial como 151
- Diagramar as 10 primeiras páginas Braille (verificar títulos, parágrafos etc)
- Salvar
- Fechar o programa

Apêndice B

Tabela de avaliação Heurística

Avaliação Heurística BrailleÁgil					
1: Visibilidade do status do sistema					
O sistema deve sempre manter o usuário informado sobre o que está acontecendo.					
Problema:					
Gravidade:	Catastrófica	Grave	Cosmética	Não há	N/A
2: Correspondência entre o sistema e o mundo real					
A interface deve usar uma linguagem familiar ao usuário, com termos e conceitos do dia a dia.					
Problema:					
Gravidade:	Catastrófica	Grave	Cosmética	Não há	N/A
3: Controle e liberdade do usuário					
Os usuários precisam de opções para desfazer ações ou sair de processos facilmente.					
Problema:					
Gravidade:	Catastrófica	Grave	Cosmética	Não há	N/A
4: Consistência e padronização					
O design deve manter padrões visuais e comportamentais ao longo da interface.					
Problema:					
Gravidade:	Catastrófica	Grave	Cosmética	Não há	N/A
5: Prevenção de erros					
É melhor prevenir do que corrigir erros depois.					
Problema:					
Gravidade:	Catastrófica	Grave	Cosmética	Não há	N/A

6: Reconhecimento ao invés de memória					
O sistema deve minimizar a carga de memória do usuário, tornando opções visíveis.					
Problema:					
Gravidade:	Catastrófica	Grave	Cosmética	Não há	N/A
7: Eficiência e flexibilidade de uso					
A interface deve funcionar bem para iniciantes e também permitir atalhos para usuários experientes.					
Problema:					
Gravidade:	Catastrófica	Grave	Cosmética	Não há	N/A
8: Estética e design minimalista					
Menos é mais. Interfaces limpas ajudam o usuário a focar no que realmente importa.					
Problema:					
Gravidade:	Catastrófica	Grave	Cosmética	Não há	N/A
9: Recuperação diante de erros					
Mensagens de erro devem ser claras, indicar o problema e sugerir uma solução.					
Problema:					
Gravidade:	Catastrófica	Grave	Cosmética	Não há	N/A
10: Ajuda e documentação					
Embora o ideal seja que o sistema seja fácil de usar sem instruções, é bom ter ajuda disponível.					
Problema:					
Gravidade:	Catastrófica	Grave	Cosmética	Não há	N/A

Apêndice C

Formulário de avaliação do TAM

Valor	Descrição
1	Discordo totalmente
2–6	Pontos intermediários sem rótulo explícito
7	Concordo totalmente

Tabela C.1: Escala Likert de 7 pontos utilizada no questionário

Afirmiação	Avaliação (1–7)
Utilizar o BrailleÁgil no meu trabalho me permitiria cumprir minhas tarefas mais rapidamente	()
Utilizar o BrailleÁgil melhoraria minha performance no trabalho	()
Utilizar o BrailleÁgil no meu trabalho melhoraria minha produtividade	()
Utilizar o BrailleÁgil aumentaria minha efetividade no trabalho	()
Utilizar o BrailleÁgil tornaria meu trabalho mais fácil	()
Eu acho que o BrailleÁgil seria útil em meu trabalho	()

Tabela C.2: Afirmações utilizadas para a avaliação do construto de Utilidade Percebida

Afirmiação	Avaliação (1–7)
Aprender a operar o BrailleÁgil seria fácil para mim	()
Eu acharia fácil fazer o que eu quiser fazer com o BrailleÁgil	()
Minha interação com o BrailleÁgil seria clara e compreensível	()
Eu acharia o BrailleÁgil flexível de se interagir	()
Seria fácil me tornar habilidoso no uso do BrailleÁgil	()
Eu acharia o BrailleÁgil fácil de se utilizar	()

Tabela C.3: Afirmações utilizadas para a avaliação do construto de Facilidade de Uso Percebida

Bibliografia

- [ANDRIGHETTO *et al.* 2020] Rosalia ANDRIGHETTO, Elizandra Mayer Leite PREICHARDT e Daiane Hoffmann MUMBACH. “Softwares livres brasileiros de acessibilidade ao braille: funcionalidades e potencialidades do brailendo e braille fácil”. *Scientia Generalis* 1.3 (maio de 2020), pp. 1–19. URL: <https://www.scientiageneralis.com.br/index.php/SG/article/view/v1n3a1> (citado na pg. 5).
- [BARBOSA e SILVA 2010] Simone Diniz Junqueira BARBOSA e Bruno Santana da SILVA. *INTERAÇÃO HUMANO-COMPUTADOR*. Elsevier, 2010 (citado nas pgs. 2, 22).
- [BATISTA 2018] Rosana Davanzo BATISTA. “O processo de alfabetização de alunos cegos e o movimento da Desbrailização”. Diss. de mestr. Piracicaba, Brasil: Faculdade de Ciências Humanas, Universidade Metodista de Piracicaba, 2018 (citado nas pgs. 1, 5).
- [CAT 2009] BRASIL. COMITÊ DE AJUDAS TÉCNICAS. *Tecnologia Assistiva*. Subsecretaria Nacional de Promoção dos Direitos da Pessoa com Deficiência. Brasília: CORDE/SEDH/PR, 2009 (citado na pg. 4).
- [MEC 2018] BRASIL. MINISTÉRIO DA EDUCAÇÃO. SECRETARIA DE EDUCAÇÃO CONTINUADA, ALFABETIZAÇÃO, DIVERSIDADE E INCLUSÃO. *Grafia Braille para a Língua Portuguesa*. 3ª ed. Brasília-DF, 2018 (citado na pg. 1).
- [CARVALHO *et al.* 2018] Marcos Fialho de CARVALHO, Angélica Fonseca da SILVA, José Antonio dos Santos BORGES e Júlio Tadeu Carvalho da SILVEIRA. “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: MEMÓRIAS DO V SIMPÓSIO DE HISTÓRIA DA INFORMÁTICA NA AMÉRICA LATINA E CARIBE* (2018), pp. 119–131 (citado na pg. 5).
- [DAVIS 1989] Fred D. DAVIS. “Perceived usefulness, perceived ease of use, and user acceptance of information technology”. *MIS Quarterly* 13.3 (set. de 1989), pp. 319–340. DOI: [10.2307/249008](https://doi.org/10.2307/249008) (citado nas pgs. 26–28).

- [GOLDBERG *et al.* 1987] A.M. GOLDBERG, E. SCHREIER, J.D. LEVENTHAL e J.C. De WITT. “An evaluation of braille translation programs”. *Journal of Visual Impairment & Blindness* 81.10 (1987), pp. 487–492. doi: [10.1177/0145482X8708101010](https://doi.org/10.1177/0145482X8708101010). URL: <https://doi.org/10.1177/0145482X8708101010> (citado na pg. 6).
- [INPI 2018] INSTITUTO NACIONAL DA PROPRIEDADE INDUSTRIAL. *Revista da Propriedade Industrial* nº 2456. Programas de Computador, Seção VII, p. 28. Rio de Janeiro, 30 de jan. de 2018. URL: <https://revistas.inpi.gov.br/rpi/> (citado nas pgs. 1, 9).
- [KING 2001] Alasdair KING. “Text and Braille Computer Translation”. Diss. de mestr. University of Manchester Institute of Science e Technology, 2001 (citado na pg. 6).
- [LEMOS e CERQUEIRA 2014] Edison Ribeiro LEMOS e Jonir Bechara CERQUEIRA. “O sistema braille no brasil”. *Revista Benjamin Constant* Edição Especial (nov. de 2014), pp. 23–28 (citado nas pgs. 1, 3, 4, 33).
- [MALHOTRA 2012] Naresh K. MALHOTRA. *Pesquisa de marketing: uma orientação aplicada*. 6^a ed. Bookman, 2012 (citado nas pgs. 21, 22, 27).
- [NIELSEN e MOLICH 1990] Jakob NIELSEN e Rolf MOLICH. “Heuristic evaluation of user interfaces”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’90. Seattle, Washington, USA: Association for Computing Machinery, 1990, pp. 249–256. ISBN: 0201509326. doi: [10.1145/97243.97281](https://doi.org/10.1145/97243.97281). URL: <https://doi.org/10.1145/97243.97281> (citado nas pgs. 21–23).
- [OMS 2019] World Health ORGANIZATION. *World report on vision*. Rel. técn. United Nations, 2019. URL: <https://iris.who.int/server/api/core/bitstreams/a1406549-82eb-468b-a5a7-1c30a233eb2f/content> (citado na pg. 3).
- [SANTANA *et al.* 2019] Kayo C. SANTANA, Claudia P. PEREIRA e Beatriz S. de SANTANA. “Braillelearning: software para simular a máquina de escrever em braille”. *Anais do XXX Simpósio Brasileiro de Informática na Educação* (2019), pp. 1101–1110 (citado na pg. 5).
- [SUGANO *et al.* 2010] Aki SUGANO *et al.* “Ebraille: a web-based translation program for japanese text to braille”. *Kobe University Repository : Kernel* (2010). doi: [10.1108/10662241011084121](https://doi.org/10.1108/10662241011084121) (citado na pg. 6).
- [TOFANI 2012] Arthur Piza Mosterio TOFANI. “Uma ferramenta para notação musical em braille”. Diss. de mestr. São Paulo, Brasil: Instituto de Matemática e Estatística, Universidade de São Paulo, 2012 (citado nas pgs. 1, 4, 6).
- [UMBELINO 2023] Cristiano Caixeta UMBELINO. *As condições de saúde ocular no Brasil* 2023. Rel. técn. Conselho Brasileiro de Oftalmologia, 2023 (citado na pg. 3).

BIBLIOGRAFIA

[VISSOSSI e LIMA 2024] Alessandra Aparecida VISSOSSI e Hylea de Camargo Vale Fernandes LIMA. “O livro didático em braille e o processo de adaptação: uma tecnologia assistiva para a sala de aula”. *Revista Diálogos e Perspectivas em Educação Especial* 11.1 (fev. de 2024), e0240002. DOI: [10.36311/2358-8845.2024.v11n1.e0240002](https://doi.org/10.36311/2358-8845.2024.v11n1.e0240002). URL: <https://revistas.marilia.unesp.br/index.php/dialogoseperspectivas/article/view/14239> (citado nas pgs. 4, 5).