

IncluiDev: Desenvolvimento e Avaliação de uma Plataforma Baseada em Evidências para Apoio ao Ensino de Programação para Pessoas com Deficiência Visual

Lucas R. Pedro¹, Esteic Janaina Santos Batista¹

¹Faculdade de Computação – Universidade Federal de Mato Grosso do Sul (UFMS)
Campo Grande – MS – Brazil

lucas.pedro@ufms.br, esteic.batista@ufms.br

Abstract. *The inclusion of blind and visually impaired (BVI) individuals in Computer Science presents significant challenges due to the historical dependence on visual metaphors in programming languages and environments [Stefik and Siebert 2013]. Although studies, tools, and guidelines for accessible programming education exist, these resources are highly fragmented across literature and the web. This study aims to develop and evaluate a centralized repository, named IncluiDev, which organizes and provides these tools, methods, and scientific papers. The repository was built as an accessible web application and features a “Guided Assistant”, a dynamic recommendation system based on user profiles (Students, Teachers, Developers, and Researchers). Accessibility compliance was validated through automated audits (Google Lighthouse, WAVE) and manual testing with screen readers and keyboard navigation. As a preliminary heuristic step, the recommendation logic was also assessed through LLM-simulated user personas applying the System Usability Scale (SUS) [Brooke 1996] and the Technology Acceptance Model (TAM) [Davis 1989]; empirical validation with real participants is planned as future work.*

Resumo. *A inclusão de Pessoas com Deficiência Visual (PcDV) no campo da Ciência da Computação apresenta desafios significativos devido à histórica dependência de metáforas visuais em linguagens e ambientes de programação [Stefik and Siebert 2013]. Apesar da existência de estudos, ferramentas e diretrizes voltadas para a acessibilidade no ensino de programação, esses recursos encontram-se dispersos na literatura e na web. O objetivo deste estudo é desenvolver e avaliar um repositório centralizado, denominado IncluiDev, que organiza e disponibiliza essas ferramentas, métodos e artigos científicos. O repositório foi construído como uma aplicação web acessível e incorpora um “Assistente Guiado”, um sistema de recomendação dinâmico baseado em perfis de usuário (Estudantes, Professores, Desenvolvedores e Pesquisadores). A conformidade de acessibilidade foi validada por meio de auditorias automatizadas (Google Lighthouse, WAVE) e testes manuais com leitores de tela e navegação por teclado. Como etapa preliminar e heurística, a lógica de recomendação também foi avaliada por meio de personas simuladas por agentes de IA, aplicando as escalas SUS [Brooke 1996] e TAM [Davis 1989]; a validação empírica com participantes reais está prevista como trabalho futuro.*

1. Introdução

A inclusão digital de Pessoas com Deficiência Visual (PcDV) nas áreas de Ciência da Computação e Tecnologia da Informação (TI) tem se mostrado cada vez mais essencial em uma sociedade fortemente impulsionada pela tecnologia. Segundo dados da Organização Mundial da Saúde (OMS) [World Health Organization 2023], estima-se que mais de 2,2 bilhões de pessoas no mundo possuam algum grau de deficiência visual. No cenário nacional, as estatísticas do Instituto Brasileiro de Geografia e Estatística (IBGE) [IBGE 2019] reiteram a escala desse contingente, registrando milhões de cidadãos brasileiros com cegueira ou baixa visão. Nesse sentido, garantir o direito ao aprendizado tecnológico e ao desenvolvimento profissional na área de computação é um imperativo ético e social, amparado pela legislação brasileira, especialmente por meio da Lei Brasileira de Inclusão da Pessoa com Deficiência (LBI - Lei nº 13.146/2015) [Brasil 2015], que estabelece a obrigatoriedade de sistemas educacionais inclusivos em todos os níveis.

Além do amparo legal à acessibilidade, as diretrizes pedagógicas contemporâneas, como a Base Nacional Comum Curricular (BNCC) [Brasil 2018], estabelecem o desenvolvimento do pensamento computacional e da cultura digital como competências fundamentais desde a Educação Básica. Contudo, a materialização dessas diretrizes enfrenta obstáculos estruturais significativos no ensino de programação tradicional, o qual historicamente exige a compreensão de metáforas visuais complexas, como diagramas de fluxo, código em blocos coloridos e a própria indentação espacial de código [Resnick et al. 2009]. Para estudantes cegos ou com baixa visão, a dependência intrínseca de recursos puramente visuais e a falta de adaptações adequadas em softwares e mídias didáticas criam significativas barreiras de aprendizagem, culminando na exclusão acadêmica e na sub-representação desse grupo no mercado de trabalho de tecnologia.

A comunidade acadêmica e a indústria têm respondido a essas barreiras por meio da concepção de ferramentas e abordagens assistivas alternativas, como linguagens de programação baseadas em evidências de usabilidade (como a Quorum Language) [Stefik et al. 2019], ambientes físicos e tangíveis (como o Code Jumper) [Morrison et al. 2019] e a adaptação ergonômica de leitores de tela em Ambientes de Desenvolvimento Integrado (IDEs) de mercado [Seo and Rogge 2023]. Contudo, a existência dessas soluções isoladas não assegura de forma automática o acesso democrático ao usuário final.

Na prática escolar e acadêmica, educadores da Educação Básica e do Ensino Superior raramente possuem formação especializada sobre tecnologias assistivas voltadas para a programação. Diante da dispersão massiva de ferramentas, artigos metodológicos e diretrizes de acessibilidade na internet, o professor enfrenta um exaustivo e muitas vezes frustrante processo de busca e validação por conta própria. A ausência de um canal centralizado e validado que intermedeie o encontro entre a necessidade pedagógica e a solução assistiva adequada inviabiliza intervenções didáticas tempestivas e eficientes, agravando o hiato educacional enfrentado por discentes cegos. Portanto, centralizar e recomendar recursos pedagógicos e técnicos acessíveis de forma estruturada é uma necessidade urgente para que o amparo legal da LBI e os objetivos de letramento digital da BNCC deixem de ser apenas aspirações abstratas e tornem-se práticas viáveis de inclusão educacional.

1.1. Problema de Pesquisa

Apesar do crescente número de estudos sobre o ensino de programação para PcDV, os recursos encontram-se dispersos, dificultando sua localização e utilização prática por diferentes públicos. Professores em busca de metodologias de ensino, desenvolvedores buscando diretrizes de acessibilidade e os próprios estudantes enfrentam a tarefa de buscar soluções fragmentadas em diversas fontes. Diante desse cenário, este trabalho busca responder à seguinte questão de pesquisa: como centralizar, classificar e recomendar de forma acessível metodologias, ferramentas e pesquisas sobre o ensino de programação para PcDV?

1.2. Objetivos

O objetivo geral deste trabalho é construir e avaliar um portal web dinâmico e centralizado que funcione como um repositório interativo de práticas, ferramentas e estudos voltados ao ensino de programação para PcDV.

Para alcançar este propósito, foram definidos os seguintes objetivos específicos:

- Organizar os recursos identificados na literatura acadêmica e na indústria em uma taxonomia coerente;
- Desenvolver um repositório web que respeite as diretrizes de acessibilidade WCAG 2.1 nível AA [W3C 2018];
- Implementar um sistema de recomendações (“Assistente Guiado”) baseado em jornadas de perfis (Estudante, Professor, Pesquisador e Desenvolvedor);
- Avaliar a usabilidade, a aceitação e a acessibilidade do sistema utilizando métodos empíricos e escalas padronizadas [Brooke 1996, Davis 1989].

1.3. Organização do artigo

O restante deste artigo está organizado da seguinte forma: a Seção 2 apresenta o Referencial Teórico sobre o ensino acessível e os métodos de avaliação. A Seção 3 detalha os Métodos aplicados na construção do repositório e os procedimentos de validação. A Seção 4 reporta os Resultados alcançados, enquanto a Seção 5 discute as implicações desses achados. Finalmente, a Seção 6 traz a Conclusão e os trabalhos futuros.

2. Referencial Teórico

2.1. Ensino de Programação para Pessoas com Deficiência Visual

A compreensão de lógicas algorítmicas por indivíduos videntes frequentemente se apoia em recursos visuais como diagramas de bloco, pseudocódigos formatados e interfaces baseadas em interações de arrastar e soltar, a exemplo do Scratch [Resnick et al. 2009]. Para PcDV, a dependência quase exclusiva desses artefatos representa uma barreira significativa para o ensino e a prática de programação [Hadwen-Bennett et al. 2018].

Estudos na área apontam que a principal dificuldade reside na sobrecarga cognitiva imposta pela leitura sequencial dos leitores de tela [Baker et al. 2015]. Um programador vidente visualiza a estrutura de um laço de repetição ou de uma estrutura condicional de forma global por meio da indentação espacial. Em contrapartida, um usuário de leitor de tela necessita memorizar a estrutura lógica enquanto escuta o código linha por linha [Baker et al. 2019].

Para mitigar tais desafios, diferentes estratégias educacionais e tecnologias assistivas têm sido propostas. Abordagens baseadas em áudio e sonificação de código, em que blocos e erros emitem sons característicos em um ambiente acústico, e ambientes tangíveis, como a programação física utilizando hardwares modulares conectáveis [American Printing House for the Blind 2019], são exemplos de intervenções com eficácia demonstrada no ensino inclusivo [Stefik and Siebert 2013].

2.2. Recursos e Tecnologias para Inclusão

Duas estratégias distintas emergem na literatura: a adaptação de ferramentas já consolidadas e o desenvolvimento de novos ambientes e linguagens projetados desde a origem para acessibilidade. Ferramentas consolidadas no mercado, como o Visual Studio Code, implementaram modos de acessibilidade e atalhos otimizados, facilitando a navegação com leitores de tela [Seo and Rogge 2023, Microsoft 2024]. Paralelamente, linguagens de programação concebidas sob a perspectiva de evidências de usabilidade, como a linguagem Quorum [Stefik et al. 2019], contribuíram para reduzir a dependência de caracteres especiais complexos, otimizando a leitura fonética realizada por softwares de síntese de voz.

Apesar da crescente disponibilidade tecnológica, a aplicabilidade efetiva dessas ferramentas depende da curadoria da informação e do acesso a guias e diretrizes atualizadas, como as Diretrizes de Acessibilidade para Conteúdo Web (WCAG) [W3C 2018]. A organização sistemática desses materiais em repositórios qualificados é essencial para que os educadores não precisem criar adaptações pedagógicas a partir do zero, promovendo o reuso e a padronização das intervenções de acessibilidade digital.

2.3. Avaliação de Sistemas Educacionais

A validação de portais e repositórios educacionais exige o uso de instrumentos confiáveis de avaliação. A *System Usability Scale* (SUS) [Brooke 1996] é um questionário de dez itens, padronizado internacionalmente, amplamente adotado para medir de forma rápida e quantitativa a usabilidade global de uma interface tecnológica.

De maneira complementar, o *Technology Acceptance Model* (TAM) [Davis 1989] é utilizado para avaliar a utilidade percebida e a facilidade de uso percebida, fatores que determinam a intenção comportamental e a adoção do sistema pelo usuário.

No contexto da acessibilidade digital, esses modelos quantitativos, quando complementados por questões qualitativas abertas, fornecem um conjunto robusto de evidências sobre as barreiras enfrentadas durante a interação e o nível de satisfação do usuário.

3. Método

A presente pesquisa caracteriza-se como um estudo de natureza aplicada, conduzido sob uma abordagem de métodos mistos (quantitativa e qualitativa). O escopo do desenvolvimento e a seleção do catálogo de recursos basearam-se em um fluxo metodológico estruturado em etapas, partindo de uma revisão sistemática preliminar até a busca ativa na internet e curadoria final dos materiais, conforme detalhado a seguir.

3.1. Mapeamento Sistemático e Curadoria de Recursos

Para compor um acervo representativo e cientificamente validado para o portal, realizou-se uma Revisão de Literatura preliminar estruturada como uma revisão terciária, cujo propósito foi sintetizar revisões sistemáticas e mapeamentos existentes sobre o ensino de programação para pessoas com deficiência visual. Esse processo seguiu um protocolo formal de busca, seleção e extração para garantir a reprodutibilidade metodológica da pesquisa.

As buscas foram efetuadas no período de 2010 a 2025, consultando seis bases de dados indexadas de grande relevância científica: ACM Digital Library, Google Acadêmico, IEEE Xplore Digital Library, ISI Web of Science, Scopus e Springer Link. Utilizou-se a seguinte *string* de busca booleana para abranger os espectros de termos-chave do domínio:

(*“blind*” OR “visual impair*” OR “visually impaired”*) AND (*“programming” OR “algorithm*”*) AND (*“review” OR “mapping” OR “*review”*) AND (*“education” OR “teaching” OR “learning”*)

Os critérios formais aplicados para a seleção dos trabalhos foram:

- **CrITÉrios de Inclusão (CI):**

- CI1 - O trabalho é um estudo secundário (revisão sistemática, mapeamento, revisão integrativa ou terciária);
- CI2 - Aborda o ensino, aprendizagem ou formação em lógica e programação em qualquer nível educacional;
- CI3 - Discute aspectos de acessibilidade, inclusão ou tecnologias assistivas;
- CI4 - Inclui explicitamente pessoas com deficiência visual (cegueira ou baixa visão) em seu escopo;
- CI5 - O texto completo está em inglês ou português;
- CI6 - Publicado entre 2010 e 2025.

- **CrITÉrios de Exclusão (CE):**

- CE1 - O trabalho não se caracteriza como um estudo secundário;
- CE2 - Trata exclusivamente de outras deficiências sem abranger a deficiência visual;
- CE3 - Não aborda o ensino ou aprendizagem de programação (ex: foca apenas em usabilidade de sistemas gerais ou IHC corporativo);
- CE4 - O texto completo não está acessível para leitura;
- CE5 - Trata-se de um documento duplicado.

O fluxo que conecta a pesquisa bibliográfica à materialização do portal ocorreu de maneira híbrida, estruturado no caminho **busca** → **seleção** → **curadoria** → **portal**:

1. **Busca e Seleção Terciária:** Execução do protocolo de revisão terciária e triagem dos estudos secundários elegíveis;
2. **Técnica de Snowballing (Bola de Neve):** Através da leitura profunda das revisões selecionadas, foram extraídas as referências diretas de trabalhos primários, ferramentas computacionais de apoio, linguagens acessíveis, metodologias e diretrizes práticas recomendadas pela literatura acadêmica;

3. **Busca Ativa na Internet:** Realizaram-se buscas ativas complementares na web para rastrear as ferramentas mapeadas, identificando se os projetos continuavam ativos, se possuíam novas documentações ou repositórios oficiais e se existiam novas iniciativas comunitárias surgidas recentemente que não haviam sido capturadas pelos artigos;
4. **Curadoria de Recursos:** Consolidação de uma planilha taxonômica para filtrar materiais duplicados, obsoletos ou inacessíveis, catalogando os recursos em categorias estruturadas (diretrizes, ferramentas, artigos e jogos);
5. **Construção do Portal:** Integração dessa taxonomia de 34 recursos ao banco de dados do portal *IncluiDev* para alimentar o Assistente Guiado.

3.2. Desenvolvimento do Repositório

O repositório digital, nomeado *IncluiDev*, foi projetado segundo a arquitetura de *Single Page Application* (SPA), utilizando a biblioteca React [Meta Platforms, Inc. 2023] juntamente com o empacotador Vite [You 2023] e o ambiente Node.js. A versão de produção da plataforma pode ser acessada através do endereço <https://includev-portal.vercel.app/>.

A concepção arquitetural priorizou o desenvolvimento de componentes funcionais acessíveis, garantindo a incorporação nativa de atributos WAI-ARIA [W3C 2023], como as propriedades *aria-live* para atualizações dinâmicas e *aria-labelledby* para estruturação semântica.

O processo contínuo de catalogação do estado da arte gerou uma taxonomia bidimensional, na qual cada recurso levantado é mapeado e indexado em função do Nível de Ensino (Básico, Técnico, Superior) e do Público-Alvo primário.

As permissões, os papéis e a estrutura de interações dos diferentes perfis com as ferramentas disponíveis na aplicação encontram-se representados no Diagrama de Casos de Uso (Figura 1).

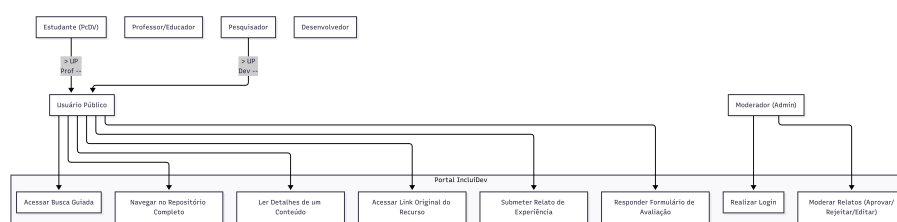


Figura 1. Diagrama de Casos de Uso do Portal IncluiDev

3.3. Construção do Assistente Guiado

Para superar as limitações dos mecanismos tradicionais de filtragem e pesquisa em tabela, foi proposto o “Assistente Guiado”: um sistema de recomendação guiado por perfil de usuário.

Diferentemente de uma listagem estática de filtros, o sistema orienta o usuário a uma autoidentificação com base em quatro perfis identificados na literatura: Estudante, Professor, Desenvolvedor e Pesquisador.

A partir do perfil selecionado, a aplicação apresenta perguntas de múltipla escolha focadas no objetivo pedagógico ou técnico principal (por exemplo, exploração de ferramentas baseadas em áudio ou obtenção de referências de revisões sistemáticas). O sistema compara as respostas do usuário com as *tags* dos conteúdos cadastrados, apresentando recomendações relevantes sem sobrecarregar o usuário de tecnologias assistivas.

O fluxo percorrido pelo usuário, desde a seleção de perfil até a exibição dos resultados pelo sistema de recomendação, está ilustrado na Figura 2.

3.4. Avaliação do Sistema

O protocolo de validação do repositório foi estruturado de forma incremental. Como etapa preliminar para homologação e pré-teste da arquitetura lógica do portal, empregou-se a simulação por Agentes de IA (baseados em Large Language Models) atuando como usuários sintéticos para testes de IHC e usabilidade. Essa abordagem tem se consolidado recentemente na literatura científica de computação [Calvano et al. 2025, Hsueh et al. 2024, Ni et al. 2026], demonstrando-se útil na geração de simulações conversacionais realistas que reproduzem fluxos de navegação e reações de personas com diferentes níveis de instrução e objetivos de pesquisa [Lubos et al. 2025].

Contudo, é fundamental demarcar os limites epistemológicos e práticos dessa técnica. Como alertado por Seshadri et al. (2026) [Seshadri et al. 2026], agentes baseados em LLMs constituem *proxies* imperfeitos e não confiáveis para simular o comportamento humano real em avaliações de sistemas. Tais modelos sofrem de severas limitações:

- **Descalibração de julgamento:** Apresentam uma tendência sistemática à condescendência e otimismo exacerbado (viés de bajulação), raramente expressando a frustração orgânica decorrente de quebras reais de interface;
- **Ausência de barreiras sensoriais orgânicas:** Modelos de linguagem processam texto de maneira abstrata e não experimentam, de forma corpórea, as dificuldades físicas de acessibilidade;
- **Ausência de fadiga sensorial:** Agentes não vivenciam a sobrecarga cognitiva e auditiva decorrente do uso prolongado de leitores de tela como NVDA ou JAWS, o que mascara gargalos de linearidade textual.

É necessário, ainda, distinguir a forma de aplicação adotada para os instrumentos SUS e TAM. Em seu uso convencional, esses questionários são respondidos por usuários humanos, que atribuem a cada item uma pontuação em escala Likert, posteriormente convertida em escores numéricos. Neste trabalho, os itens das escalas não foram executados como questionário pontuado, mas serviram como roteiro de inspeção: os agentes produziram respostas dissertativas orientadas por esses itens, abordando tanto a usabilidade da interface, dimensão central do SUS, quanto a utilidade e a facilidade de uso percebidas, dimensões do TAM.

Por esses motivos, os testes simulatórios baseados nas escalas *System Usability Scale* (SUS) [Brooke 1996] e *Technology Acceptance Model* (TAM) [Davis 1989] não são tratados aqui como substitutos psicométricos de avaliações humanas. Trata-se, estritamente, de uma etapa preliminar de natureza heurística, conduzida em ambiente de *sandbox* para homologar a consistência interna das árvores de decisão do Assistente Guiado e verificar se há erros de roteamento de fluxos antes de submeter participantes reais a testes

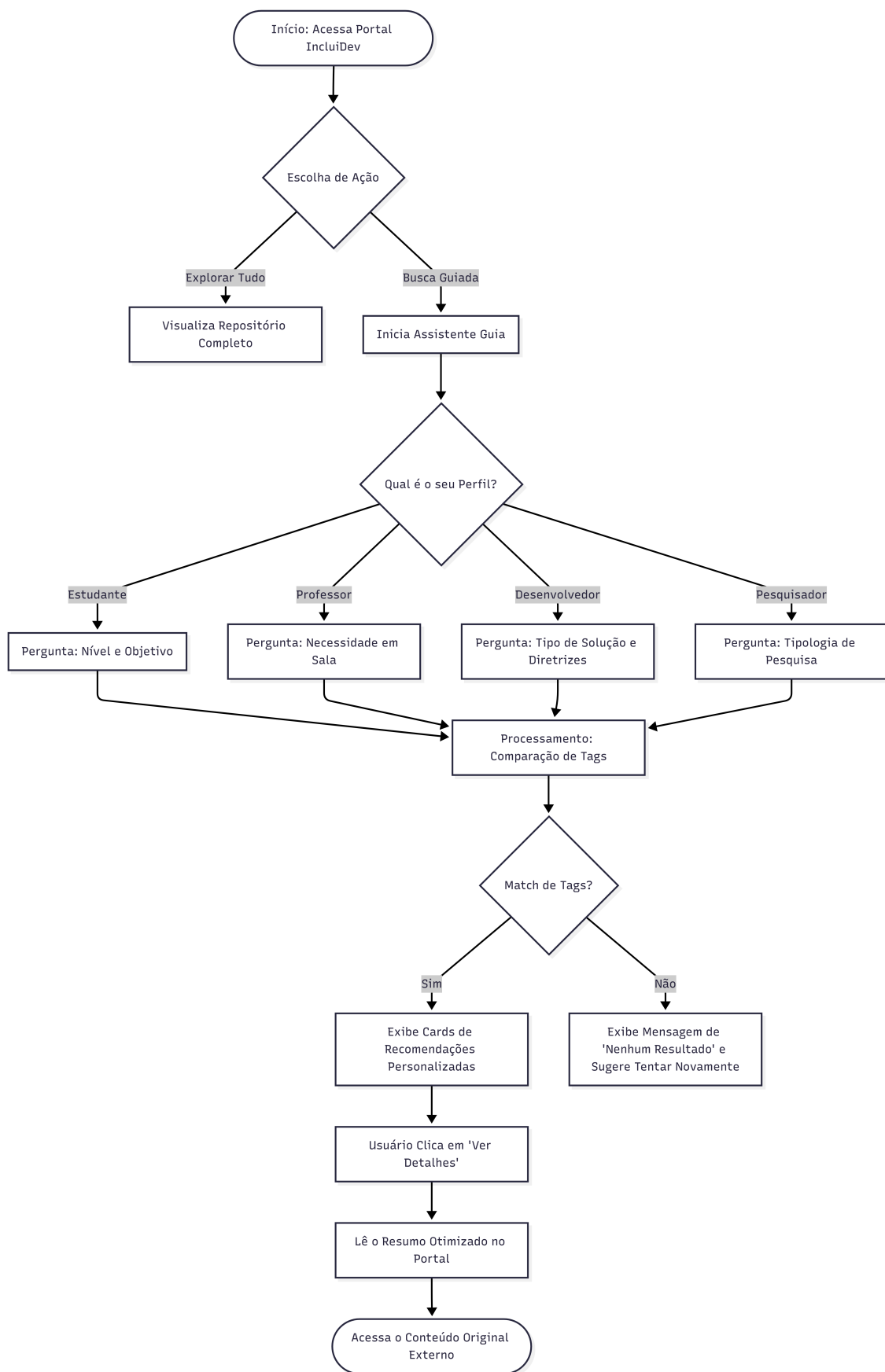


Figura 2. Fluxograma de Navegação do Assistente Guiado

exaustivos. Os agentes foram instruídos a interpretar os quatro perfis de usuário mapeados (Estudante, Professor, Pesquisador e Desenvolvedor), interagindo com a plataforma e respondendo, em formato dissertativo, aos itens das escalas SUS e TAM, funcionando como uma validação heurística preparatória. A validação empírica presencial com discentes e docentes cegos ou com baixa visão permanece como o padrão-ouro de validação do projeto, planejada como a etapa subsequente e definitiva de homologação da plataforma.

4. Resultados

4.1. Caracterização do Repositório

A etapa de curadoria e extração consolidou um catálogo contendo 34 recursos fundamentais, classificados segundo a taxonomia proposta. A categorização buscou ser abrangente, incluindo tanto *frameworks* de alcance global (como as diretrizes de acessibilidade W3C [W3C 2018] e a documentação MDN [Mozilla Developer Network 2024]) quanto ambientes lúdicos e tangíveis focados na iniciação do pensamento computacional sem monitor, como o Code Jumper [Morrison et al. 2019] e linguagens baseadas em evidências como a Quorum [Stefik et al. 2019].

O portal oferece interfaces que reúnem resumos detalhados com acesso direcionado às fontes científicas originais, para que professores e desenvolvedores encontrem referências técnicas e legais em um único lugar.

O código-fonte completo do sistema, bem como os artefatos de validação descritos nas seções seguintes, estão disponíveis no repositório público do projeto: <https://github.com/lucasmantovany/includev-portal>.

Para ilustrar a interface da aplicação, a Figura 3 apresenta a tela inicial do portal, destacando o acesso rápido por perfis e níveis de ensino. A Figura 4 demonstra a tela do repositório com filtros e a listagem de conteúdos.

4.2. Funcionamento do Assistente Guiado

Durante os testes técnicos, a implementação do Assistente Guiado confirmou que as árvores de decisão e o cruzamento de *tags* retornam ao menos uma recomendação em todos os cenários cobertos pelo acervo atual. O fluxo respondeu de forma consistente às variações de contexto para cada um dos quatro perfis e manteve a compatibilidade com leitores de tela nos ambientes de teste. O arranjo visual limpo, sem *pop-ups* inesperados, favorece a navegação independente de usuários que dependem exclusivamente do teclado. A Figura 5 ilustra a iteração inicial do Assistente, iniciando na seleção do perfil.

4.3. Avaliação de Acessibilidade

Para avaliar o cumprimento das diretrizes de acessibilidade WCAG 2.1 (Nível AA), foram realizadas auditorias mistas. A avaliação automatizada com o Google Lighthouse apontou alta conformidade, e a ferramenta WAVE confirmou a ausência de erros estruturais, indicando apenas alertas pontuais de contraste de cores em elementos secundários. A síntese dos escores obtidos é apresentada na Tabela 1.

Esses resultados, contudo, têm alcance limitado. Estudos de benchmarking de ferramentas de avaliação automatizada indicam que elas identificam apenas uma fração dos problemas de acessibilidade de uma interface, com cobertura estimada entre 14% e

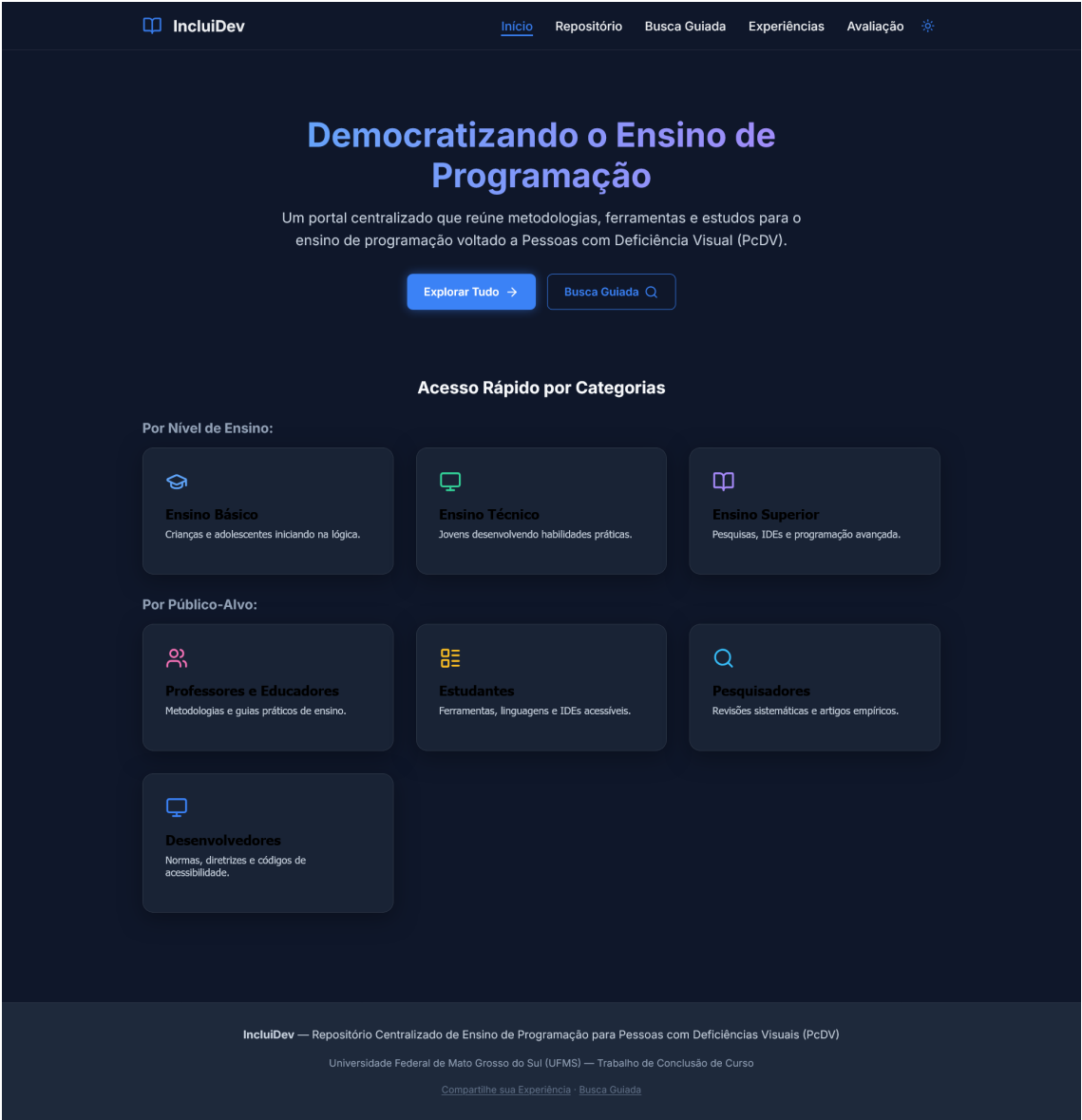


Figura 3. Tela Inicial do Portal InlucDev

Tabela 1. Resultados das auditorias automatizadas de acessibilidade		
Ferramenta / Métrica	Desktop	Móvel
Google Lighthouse (escore de Acessibilidade)	96/100	95/100
WAVE – Erros estruturais	0	0
WAVE – Alertas de contraste (elementos secundários)	Pontuais	Pontuais

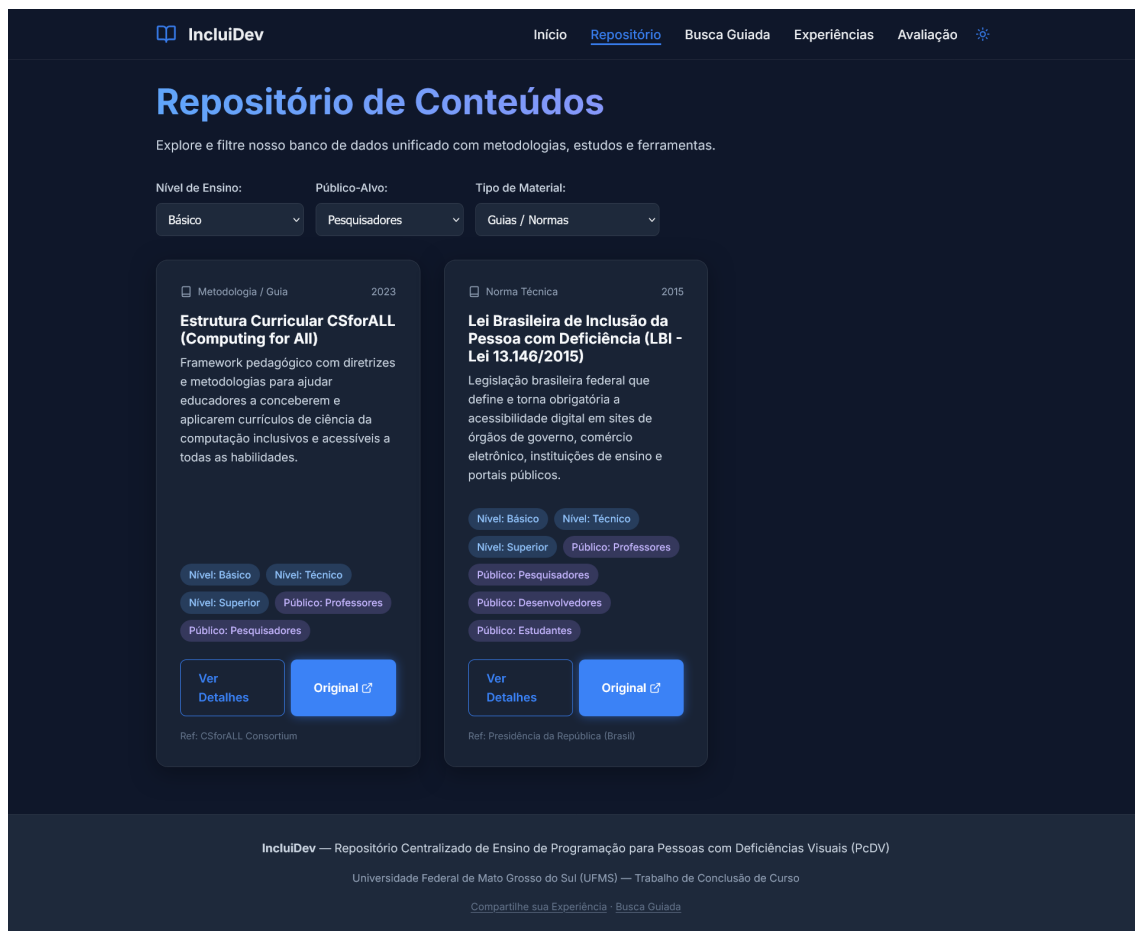


Figura 4. Tela do Repositório e Filtros do Assistente

IncluiDev

[Início](#) [Repositório](#) [Busca Guiada](#) [Experiências](#) [Avaliação](#) 

Busca Guiada — Etapa 1 de 4

Visualização Clássica

Qual é o seu perfil ou papel principal na área de ensino?

☐ **Estudante**
Desejo encontrar linguagens, IDEs, ferramentas táteis e tutoriais práticos.

☐ **Professor(a)**
Procuro metodologias, planos de aula, currículos e ferramentas pedagógicas.

☐ **Desenvolvedor(a)**
Busco diretrizes, normas, APIs e exemplos de design inclusivo.

☐ **Pesquisador(a)**
Busco revisões sistemáticas, mapeamentos de literatura e estudos empíricos.

< Voltar

Avançar >

IncluiDev — Repositório Centralizado de Ensino de Programação para Pessoas com Deficiências Visuais (PcDV)

Universidade Federal de Mato Grosso do Sul (UFMS) — Trabalho de Conclusão de Curso

[Compartilhe sua Experiência](#) · [Busca Guiada](#)

Figura 5. Iteração inicial do Assistente Guiado para seleção do perfil do usuário

38% dos critérios [Vigo et al. 2013]. Por isso, a auditoria manual, conduzida pelo próprio pesquisador com leitores de tela e navegação estrita por teclado, teve papel central na avaliação.

Na auditoria manual, verificou-se o funcionamento adequado da navegação por teclado, sem armadilhas de foco (*keyboard traps*). A aplicação emprega regiões dinâmicas (*aria-live="assertive"*) de modo a garantir que as mudanças de tela do Assistente Guiado sejam anunciadas de imediato por leitores de tela como o NVDA, o que evita a desorientação do usuário.

4.4. Validação Simulatória por Agentes de IA

Como etapa preliminar de validação estrutural e lógica da interface, a avaliação de usabilidade foi conduzida utilizando a simulação por Agentes de IA (*LLM-Simulated Users*), uma abordagem apoiada por estudos recentes na área de interação humano-computador. Esta etapa inicial visa assegurar a coesão do fluxo de navegação antes de submeter a plataforma a testes empíricos com usuários reais. Para isso, foram elaborados *prompts* de sistema simulando as quatro personas do projeto: Estudante, Professor, Pesquisador e Desenvolvedor. O registro completo desse processo os *prompts* utilizados e as respostas geradas pelos quatro agentes está disponível para consulta no repositório público do projeto, garantindo a transparência metodológica desta etapa.

A interação simulada das quatro personas com o Assistente Guiado resultou em respostas consistentemente favoráveis aos itens do SUS e do TAM, sem que nenhuma das personas reportasse obstáculos estruturais de navegação ou de compreensão do fluxo. Quanto à usabilidade aferida pelos itens do SUS, as justificativas apontaram baixa carga percebida de leitura sequencial; quanto às dimensões do TAM, registrou-se alta utilidade percebida na recomendação direcionada por perfil e facilidade de uso na condução do fluxo. A persona de Estudante, por exemplo, registrou: “*A interface fez perguntas objetivas. Não tive que ler centenas de links de pesquisa, o que aliviou minha fadiga com o NVDA.*” A persona de Pesquisador comparou favoravelmente a busca guiada a consultas booleanas tradicionais.

Uma análise temática das quatro respostas revela um padrão recorrente entre os perfis de Estudante, Professor e Pesquisador: todos destacaram a redução do esforço de busca e leitura sequencial como o principal benefício percebido, contrastando a experiência guiada com a navegação tradicional por mecanismos de busca genéricos ou bases bibliográficas extensas. Já a persona de Desenvolvedor enfatizou um eixo distinto, voltado à completude técnica e à ausência de ruído informacional na apresentação dos recursos. Essa diferença sugere que, embora o Assistente Guiado atenda de forma consistente à necessidade central de reduzir a sobrecarga de navegação, futuras iterações do sistema poderiam explorar critérios de relevância específicos por perfil — por exemplo, priorizando profundidade técnica para Desenvolvedores e amplitude de fontes para Pesquisadores. Tais padrões, contudo, derivam de uma simulação textual e devem ser interpretados como hipóteses a confirmar com usuários reais, e não como conclusões definitivas sobre o comportamento do público-alvo.

É importante destacar que esses resultados não substituem uma medição psicométrica real: por se tratar de uma simulação textual, os valores eventualmente atribuídos pelos agentes às escalas SUS e TAM não possuem o mesmo lastro de vali-

dade que escores coletados com usuários humanos. Eles são reportados aqui apenas como sinalização heurística de que a arquitetura lógica do sistema não apresenta inconsistências graves nos quatro fluxos avaliados, servindo de base para a etapa de validação empírica planejada como trabalho futuro (Seção 6).

5. Discussão

Os achados desta primeira fase podem ser interpretados à luz da literatura sobre a carga cognitiva imposta pelos leitores de tela. Estudos sobre a experiência de programadores cegos documentam o esforço considerável despendido para reconstruir mentalmente a estrutura do código durante a leitura sequencial [Baker et al. 2015, Baker et al. 2019]. O Assistente Guiado não atua sobre o código em si, mas transpõe esse princípio para a tarefa de busca: ao substituir a varredura de listas extensas por um diálogo curto orientado a perfil, reduz o volume de conteúdo que o usuário precisa percorrer e manter em memória. A baixa carga de leitura sequencial relatada nas simulações é coerente com essa expectativa, embora sua confirmação dependa de medição com usuários reais.

No plano do problema de pesquisa, os resultados sustentam a hipótese de que a centralização e a curadoria respondem à dispersão de recursos apontada na literatura [Hadwen-Bennett et al. 2018]. Ao reunir em um único ponto ferramentas, linguagens baseadas em evidências e diretrizes de acessibilidade [Stefik et al. 2019, W3C 2018], o *IncluiDev* reduz o custo de localização que recai sobre professores e desenvolvedores. A taxonomia bidimensional e a recomendação por perfil constituem, nesse sentido, menos uma inovação técnica e mais uma estratégia de organização de conhecimento já existente, alinhada à necessidade de que materiais de acessibilidade sejam reusáveis e padronizados em repositórios qualificados.

A principal cautela interpretativa recai sobre a natureza da avaliação. Os indícios de usabilidade e aceitação derivam de personas simuladas, e não de usuários humanos, o que restringe seu alcance a uma verificação de coerência lógica. Essa limitação é reconhecida pela própria literatura que fundamenta o método, pois agentes de IA tendem a julgamentos descalibrados e otimistas e não reproduzem a experiência sensorial do público-alvo [Seshadri et al. 2026]. Por esse motivo, os sinais favoráveis quanto às dimensões do SUS e do TAM devem ser entendidos como hipóteses a testar, não como evidência de aceitação. Soma-se a isso o fato de que a conformidade técnica medida pelas auditorias, ainda que combinada à inspeção manual, cobre apenas parte das barreiras reais de interação [Vigo et al. 2013]. Esses fatores delimitam a validade externa do estudo e reforçam que a etapa empírica com pessoas com deficiência visual é condição necessária para confirmar os benefícios aqui sinalizados.

6. Conclusão

A inclusão de Pessoas com Deficiência Visual no ensino de programação é um desafio complexo, agravado pela dispersão de ferramentas e metodologias acessíveis. Este estudo apresentou o *IncluiDev*, um repositório centralizado e dinâmico, projetado para mitigar essa fragmentação por meio de uma taxonomia estruturada e de uma navegação orientada a perfis de usuário.

As auditorias técnicas indicaram que a ferramenta atinge altos padrões de acessibilidade e cumpre as diretrizes da WCAG 2.1. A etapa preliminar de validação lógica,

conduzida por simulação com agentes de IA, sugeriu que o Assistente Guiado oferece boa usabilidade e recomenda recursos pertinentes aos quatro perfis de usuário, com potencial para reduzir a carga cognitiva da navegação por leitores de tela.

Como discutido na Seção 5, as principais limitações deste estudo decorrem da natureza preliminar da validação por agentes de IA, que não substitui a experiência de usuários reais, e da ausência de testes empíricos com participantes humanos, motivada pelo prazo hábil da pesquisa para submissão. Soma-se a isso o fato de que ferramentas automatizadas de acessibilidade, como discutido na Seção 4.3, capturam apenas uma fração dos problemas reais de interação da interface. Por isso, como trabalhos futuros, torna-se essencial a execução de testes empíricos com grupos representativos de participantes, em especial pessoas com deficiência visual, tanto discentes quanto docentes desse público, que poderão utilizar o sistema com frequência, bem como a comparação dos resultados simulados de SUS/TAM com avaliações conduzidas com usuários reais.

Espera-se que o portal *IncluiDev* gere um impacto positivo imediato ao facilitar o encontro de ferramentas e metodologias de programação inclusiva, reduzindo o tempo de pesquisa de educadores e estudantes. Com isso, o projeto busca contribuir para a pesquisa tecnológica educacional e para o fomento de um ensino de computação mais inclusivo e estruturado. Adicionalmente, a expansão contínua da base de dados catalogada e o aprimoramento das heurísticas de recomendação constituem os próximos passos para consolidar o IncluiDev como ferramenta de referência na área.

Referências

- American Printing House for the Blind (2019). Code jumper: A physical programming language for students who are blind or visually impaired. <https://www.aph.org/product/code-jumper/>. Originalmente desenvolvido como Project Torino pela Microsoft Research.
- Baker, C. M., Bennett, C. L., and Ladner, R. E. (2019). Educational experiences of blind programmers. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19)*, pages 759–765. ACM.
- Baker, C. M., Milne, L. R., and Ladner, R. E. (2015). Structjumper: A tool to help blind programmers navigate and understand the structure of code. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*, pages 3043–3052. ACM.
- Brasil (2015). Lei nº 13.146, de 6 de julho de 2015. institui a lei brasileira de inclusão da pessoa com deficiência (estatuto da pessoa com deficiência). Diário Oficial da União. Disponível em: http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2015/lei/l13146.htm.
- Brasil (2018). Base nacional comum curricular (bncc). http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518_versaofinal_site.pdf. Acessado em: 2025.
- Brooke, J. (1996). Sus: A ‘quick and dirty’ usability scale. In Jordan, P. W., Thomas, B., Weerdmeester, B. A., and McClelland, I. L., editors, *Usability Evaluation in Industry*, pages 189–194. Taylor & Francis, London.

- Calvano, M., Curci, A., Lanzilotti, R., Piccinno, A., and Ragone, A. (2025). Leveraging large language models for usability testing: A preliminary study. In *Companion Proceedings of the 30th International Conference on Intelligent User Interfaces (IUI '25 Companion)*, pages 78–81.
- Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13(3):319–340.
- Hadwen-Bennett, A., Sentance, S., and Morrison, C. (2018). Making programming accessible to learners with visual impairments: A literature review. *International Journal of Computer Science Education in Schools*, 2(2):3–13.
- Hsueh, N.-L., Lin, H.-J., and Lai, L.-C. (2024). Applying large language model to user experience testing. *Electronics*, 13(23):4633.
- IBGE (2019). Pesquisa nacional de saúde: Ciclos de vida. <https://biblioteca.ibge.gov.br/visualizacao/livros/liv101748.pdf>. Acessado em: 2025.
- Lubos, S., Felfernig, A., Garber, D., Le, V.-M., and Tran, T. N. T. (2025). Towards LLM-based usability analysis for recommender user interfaces. In *12th Joint Workshop on Interfaces and Human Decision Making for Recommender Systems (IntRS'25) at RecSys 2025*. arXiv:2511.14359.
- Meta Platforms, Inc. (2023). React: A javascript library for building user interfaces. <https://react.dev/>. Acessado em: 2025.
- Microsoft (2024). Accessibility in Visual Studio Code. <https://code.visualstudio.com/docs/editor/accessibility>. Acessado em: 2025.
- Morrison, C., Cutrell, E., Dhareashwar, A., Villar, N., Thieme, A., Taylor, A., et al. (2019). Physical programming for blind and low vision children at scale. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–13.
- Mozilla Developer Network (2024). MDN web docs: Resources for developers, by developers. <https://developer.mozilla.org/>. Acessado em: 2025.
- Ni, B., Wang, Y., Wang, L., Kveton, B., Deroncourt, F., Xia, Y., Chen, H., Luera, R., Basu, S., Mukherjee, S., et al. (2026). A survey on LLM-based conversational user simulation. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2026)*.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., and Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(11):60–67.
- Seo, J. and Rogge, M. (2023). Coding non-visually in visual studio code: Collaboration towards accessible development environment for blind programmers. *Journal on Technology and Persons with Disabilities*, 11:120–136.
- Seshadri, P., Cahyawijaya, S., Odumakinde, A., Singh, S., and Goldfarb-Tarrant, S. (2026). Lost in simulation: Llm-simulated users are unreliable proxies for human users in agentic evaluations. *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (ACL 2026)*.

- Stefik, A., Ladner, R. E., Groce, A., and Hollings, S. (2019). The quorum programming language. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19)*. ACM.
- Stefik, A. and Siebert, S. (2013). An empirical investigation into programming language syntax. *ACM Transactions on Computing Education (TOCE)*, 13(4):1–40.
- Vigo, M., Brown, J., and Conway, V. (2013). Benchmarking web accessibility evaluation tools: Measuring the harm of sole reliance on automated tests. In *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility (W4A '13)*. ACM.
- W3C (2018). Web content accessibility guidelines (WCAG) 2.1. W3c recommendation, World Wide Web Consortium (W3C). Disponível em: <https://www.w3.org/TR/WCAG21/>.
- W3C (2023). Accessible rich internet applications (WAI-ARIA) 1.2. W3c recommendation, World Wide Web Consortium (W3C). Disponível em: <https://www.w3.org/TR/wai-aria-1.2/>.
- World Health Organization (2023). Blindness and vision impairment. <https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment>. Acessado em: 2025.
- You, E. (2023). Vite: Next generation frontend tooling. <https://vitejs.dev/>. Acessado em: 2025.



**ATA DA COMISSÃO AVALIADORA DA COMPONENTE CURRICULAR NÃO
DISCIPLINAR DE TRABALHO DE CONCLUSÃO DE CURSO DO CURSO DE
SISTEMAS DE INFORMAÇÃO DA FACULDADE DE COMPUTAÇÃO DA FUNDAÇÃO
UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL.**

Aos vinte e cinco dias do mês de junho do ano de dois mil e vinte e seis, às quatorze horas, por videoconferência por meio da ferramenta Google Meet, reuniu-se a Comissão Avaliadora composta pelos membros: Esteic Janaina Santos Batista, Anderson Correa de Lima e Thiago Gonçalves de Almeida, sob a presidência do primeiro para avaliar a CCND de Trabalho de Conclusão de Curso do estudante LUCAS RODRIGUES PEDRO do Curso de Sistemas de Informação - Bacharelado, apresentado sob o título "*IncluiDev: Desenvolvimento e Avaliação de uma Plataforma Baseada em Evidências para Apoio ao Ensino de Programação para Pessoas com Deficiência Visual*", e orientação de Esteic Janaina Santos Batista. A presidente da Comissão declarou abertos os trabalhos e agradeceu a presença de todos os Membros. A seguir, concedeu a palavra ao aluno que expôs seu Trabalho. Terminada a exposição, os senhores membros da Banca Examinadora iniciaram as arguições. Terminadas as arguições, a presidente da Banca Examinadora fez suas considerações. A seguir, a Banca Examinadora reuniu-se para avaliação, e após, emitiu Parecer expresso conforme segue:

EXAMINADOR

Ma. Esteic Janaina Santos Batista (Presidente)

Dr. Anderson Correa de Lima

Me. Thiago Gonçalves de Almeida.

RESULTADO FINAL:

(X) Aprovado () Reprovado

Para registro e finalidades legais, lavrou-se a presente Ata, que vai assinada pela Presidente e os Membros da Comissão Avaliadora. Nada mais havendo a ser tratado, a Presidente declarou a sessão encerrada e agradeceu a todos pela presença.

Campo Grande, 25 de junho de 2026.

NOTA
MÁXIMA
NO MEC

UFMS
É 10!!!



Documento assinado eletronicamente por **Thiago Gonçalves de Almeida, Usuário Externo**, em 25/06/2026, às 15:57, conforme horário oficial de Mato Grosso do Sul, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).

NOTA
MÁXIMA
NO MEC

UFMS
É 10!!!



Documento assinado eletronicamente por **Anderson Correa de Lima, Professor do Magisterio Superior**, em 25/06/2026, às 15:58, conforme horário oficial de Mato Grosso do Sul, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).

NOTA
MÁXIMA
NO MEC

UFMS
É 10!!!



Documento assinado eletronicamente por **Esteic Janaina Santos Batista, Usuário Externo**, em 25/06/2026, às 15:58, conforme horário oficial de Mato Grosso do Sul, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no site https://sei.ufms.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **6493088** e o código CRC **3DCD1231**.

FACULDADE DE COMPUTAÇÃO

Av Costa e Silva, s/nº - Cidade Universitária

Fone:

CEP 79070-900 - Campo Grande - MS

Referência: Processo nº 23104.019438/2020-51

SEI nº 6493088