

FRAMEWORK ACESSÍVEL PARA DESENVOLVIMENTO DE APLICAÇÕES WEB: RUMO À INCLUSÃO DIGITAL

Felipe Soares Milani¹, André Faria Ruaro²

Resumo: A acessibilidade digital é essencial para garantir que todas as pessoas possam utilizar serviços e conteúdos disponíveis na internet, independentemente de suas limitações físicas, sensoriais ou cognitivas. No entanto, muitos sistemas ainda são desenvolvidos sem considerar critérios mínimos de inclusão, o que reforça a necessidade de soluções práticas e acessíveis desde o início do projeto. Este trabalho apresenta o desenvolvimento de um framework com foco em acessibilidade, criado com tecnologias modernas de desenvolvimento web. A proposta oferece uma biblioteca de componentes reutilizáveis, já configurados para atender a critérios de acessibilidade e facilitar sua integração por desenvolvedores. Foram realizados testes com ferramentas especializadas e navegação por teclado e leitores de tela, garantindo a eficácia do framework na criação de interfaces inclusivas. O resultado é uma ferramenta prática que contribui para tornar a internet mais acessível e para incentivar a adoção de boas práticas no desenvolvimento web.

Palavras-chave: Acessibilidade digital; Desenvolvimento web; WCAG; Inclusão digital; Framework acessível.

¹Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense (Unesc), felipe.s.milani@gmail.com

²Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense (Unesc), andrefariaruaro@gmail.com

ABSTRACT: Digital accessibility is essential to ensure that all individuals can use online services and content, regardless of physical, sensory, or cognitive limitations. However, many systems are still built without considering basic inclusion criteria, which highlights the need for practical and accessible solutions from the start of development. This work presents the development of a framework focused on accessibility, using modern web development technologies. The proposal delivers a library of reusable components, already configured to meet accessibility requirements and easy to integrate by developers. Tests were conducted using specialized tools, as well as manual evaluations with keyboard navigation and screen readers, confirming the framework's effectiveness in building inclusive interfaces. The result is a practical tool that helps make the web more accessible and encourages the adoption of best practices in web development.

Keywords: Digital accessibility; Web development; WCAG; Digital inclusion; Accessible framework.

1 INTRODUÇÃO

A transformação digital tem mudado profundamente a forma como a sociedade interage com o mundo. Estar presente na internet, atualmente, é visto não apenas como um diferencial, mas como uma condição básica de participação em diversos setores. Serviços bancários, governamentais, educacionais, de saúde e lazer, entre muitos outros, migraram para plataformas digitais, promovendo agilidade e alcance geográfico. No entanto, esse avanço tecnológico não tem sido acompanhado por um compromisso proporcional com a acessibilidade digital, ou seja, com a garantia de que todas as pessoas, independentemente de suas capacidades físicas, sensoriais ou cognitivas, possam usufruir desses serviços com autonomia e equidade.

A OMS (Organização Mundial da Saúde, 2022) estima que mais de 1 bilhão de pessoas em todo o mundo convivem com algum tipo de deficiência, número que corresponde a cerca de 15% da população global. No Brasil, conforme dados do Censo 2023 do IBGE (Instituto Brasileiro de Geografia e Estatística, 2023), aproximadamente 7,5% da população brasileira declarou ter algum grau de dificuldade para enxergar, ouvir, caminhar ou compreender. Diante desses números, é inegável a urgência de iniciativas que promovam inclusão digital de forma sistemática e eficaz.

No campo da computação, diversos padrões e diretrizes foram desenvolvidos ao longo dos anos para orientar o desenvolvimento de con-

teúdos digitais acessíveis. O mais importante e reconhecido mundialmente é o conjunto de diretrizes do Web Content Accessibility Guidelines (WCAG), publicado pelo World Wide Web Consortium (W3C, 2024). No Brasil, o Modelo de Acessibilidade em Governo Eletrônico e-MAG (Departamento de Governo Eletrônico, 2014) representa a adaptação dessas diretrizes para a realidade dos portais públicos nacionais, sendo exigido para sites governamentais.

Apesar da existência de normas claras, diretrizes consolidadas e leis específicas, como a Lei Brasileira de Inclusão da Pessoa com Deficiência (BRASIL, 2015), a acessibilidade digital ainda está longe de ser uma realidade consolidada. Muitas aplicações são desenvolvidas sem considerar usuários com deficiência, o que contribui para um cenário de exclusão digital. Essa lacuna entre teoria e prática tem sido confirmada por diversos estudos acadêmicos e técnicos, que evidenciam uma resistência ou falta de preparo por parte de desenvolvedores e empresas para incorporar padrões de acessibilidade de maneira sistemática.

Nesse contexto, a pesquisa de JÚNIOR e PEREIRA (2019) é particularmente reveladora. Ao investigar a experiência de pessoas com deficiência visual e o grau de conhecimento dos desenvolvedores web sobre as WCAG, os autores identificaram que, embora a maioria dos desenvolvedores entrevistados tenha algum nível de familiaridade com as diretrizes, estas não são frequentemente aplicadas. Este achado ilustra uma das principais motivações deste trabalho: a necessidade de ferramentas que simplifiquem e padronizem o desenvolvimento acessível.

De maneira semelhante, o estudo de Bi et al. (2021) investigou a percepção de profissionais da indústria de software sobre o desenvolvimento e o design acessível, através de entrevistas e uma pesquisa com 365 participantes de 26 países. Os autores constataram que, embora a acessibilidade seja considerada relevante, ela é frequentemente tratada de forma superficial, com foco em metas de curto prazo. O estudo reforça a tese de que soluções técnicas precisam ser acompanhadas por mecanismos facilitadores, como frameworks e ferramentas, além de políticas organizacionais que incentivem a sua adoção sistemática.

Outro exemplo é o AccessHub, desenvolvido por SILVA (2023), que propôs um guia de acessibilidade prático para desenvolvedores. Embora tenha sido um avanço importante, o AccessHub é, essencialmente, uma documentação expandida, que ainda exige um alto grau de conhecimento técnico e esforço individual na implementação. A lacuna que per-

manece é a inexistência de um framework acessível funcional, com componentes prontos e validados, que reduzam o esforço e aumentem a adesão às boas práticas.

É nesse cenário que se insere esta pesquisa, que resultou no desenvolvimento de um framework acessível para aplicações web, utilizando tecnologias modernas como Angular e Node.js. A solução criada oferece ao desenvolvedor componentes reutilizáveis com recursos de acessibilidade integrados, facilitando sua aplicação prática em projetos reais. Com isso, busca-se simplificar a implementação de critérios técnicos de acessibilidade, estimular o uso de boas práticas inclusivas e tornar o desenvolvimento web mais acessível e alinhado com padrões de qualidade e responsabilidade social.

Além disso, a proposta se diferencia por incorporar componentes que já atendem a critérios de acessibilidade, como contraste adequado, navegação por teclado, estrutura semântica e compatibilidade com tecnologias assistivas. Isso permite que aplicações web sejam desenvolvidas com qualidade técnica e responsabilidade social, atendendo não apenas à legislação vigente, mas também aos princípios éticos da inclusão digital.

A acessibilidade digital é, simultaneamente, uma exigência legal, um imperativo ético e uma oportunidade técnica. Ignorá-la significa excluir milhões de pessoas do acesso a serviços essenciais e desrespeitar leis nacionais e internacionais. Por outro lado, promovê-la amplia o alcance das aplicações, melhora a usabilidade geral, fortalece a imagem institucional e atende aos princípios da responsabilidade social. O criador da World Wide Web, afirma: “O poder da Web está em sua universalidade. O acesso por todos, independentemente de deficiência, é um aspecto essencial.” (BERNERS-LEE, 1997).

Os trabalhos correlatos indicam um consenso, o problema da acessibilidade web não está na ausência de normas, mas sim na ausência de soluções práticas para implementá-las. Desta forma, esta pesquisa busca preencher essa lacuna ao desenvolver um framework acessível, técnico e funcional, viabilizando o desenvolvimento de aplicações verdadeiramente inclusivas. Ao propor um framework acessível, espera-se facilitar a vida de desenvolvedores, educar sobre boas práticas e impulsionar a construção de uma internet mais justa, diversa e acessível.

2 MATERIAIS E MÉTODOS

O desenvolvimento de soluções acessíveis requer não apenas o domínio de normas e diretrizes técnicas, mas também uma compreensão aprofundada dos desafios enfrentados por desenvolvedores e usuários no cenário real. Desta forma, esta seção descreve inicialmente a revisão da literatura, que forneceu o embasamento teórico e prático para a identificação das principais lacunas na aplicação da acessibilidade digital. Em seguida, são apresentados os recursos, ferramentas e critérios utilizados na construção da proposta deste trabalho.

2.1 REVISÃO DA LITERATURA

A revisão da literatura teve como objetivo levantar e analisar estudos prévios relacionados à acessibilidade digital, abrangendo tanto aspectos normativos quanto técnicos. Entre os principais referenciais internacionais, destacam-se as diretrizes WCAG, elaboradas pelo W3C. Essas diretrizes definem princípios técnicos para o desenvolvimento de conteúdos web acessíveis, baseando-se nos pilares de Perceptibilidade, Operabilidade, Compreensibilidade e Robustez (POUR).

Normativas como a Lei Brasileira de Inclusão da Pessoa com Deficiência (BRASIL, 2015) e o E-MAG (Departamento de Governo Eletrônico) estabelecem obrigações legais e diretrizes adaptadas à realidade nacional, sendo especialmente aplicáveis a portais públicos e serviços governamentais.

Além dos referenciais legais e técnicos, foram analisados trabalhos acadêmicos que abordam soluções práticas de acessibilidade digital, estudos exploratórios sobre o nível de conhecimento dos desenvolvedores em relação às WCAG e propostas de ferramentas de apoio ao desenvolvimento web acessível. Nesse sentido, a literatura revela uma lacuna entre o conhecimento teórico das diretrizes e sua efetiva aplicação nos projetos de software.

Autores como NORMAN (2013) reforçam a importância de considerar a diversidade de usuários desde as fases iniciais do design. Para ele, o design centrado no ser humano deve levar em conta limitações físicas, sensoriais e cognitivas, integrando acessibilidade como parte essencial da experiência do usuário, e não como um recurso adicional.

Os estudos apontam que, embora haja um corpo normativo bem estabelecido e diversas iniciativas voltadas à promoção da acessibilidade digital, ainda persistem dificuldades significativas na sua implementação

prática. Entre os principais desafios identificados estão a baixa familiaridade técnica de muitos desenvolvedores com as diretrizes internacionais, a escassez de ferramentas acessíveis que integrem boas práticas de forma automatizada, e a ausência de processos sistemáticos para garantir a conformidade ao longo do ciclo de desenvolvimento, evidenciando a necessidade de soluções mais integradas e orientadas à prática.

2.2 LACUNAS E DESAFIOS

A análise dos trabalhos existentes evidenciou algumas lacunas importantes encontradas pelos desenvolvedores:

- Falta de ferramentas práticas que integrem as WCAG de forma nativa no processo de desenvolvimento;
- Baixo nível de adoção de boas práticas por parte de desenvolvedores, mesmo com conhecimento prévio;
- Dificuldade em validar se uma aplicação atende aos critérios de acessibilidade durante seu desenvolvimento.

2.3 AMBIENTE DE DESENVOLVIMENTOS

O ambiente de desenvolvimento adotado para a criação do framework foi composto pelos seguintes recursos:

- Computador desktop com processador de 4.9GHz;
- Armazenamento SSD de 500 GB;
- Node.js v22: Suporte à publicação como pacote NPM;
- Angular 19: estruturação dos componentes interativos e dinâmicos;
- Visual Studio Code: ambiente de codificação;
- Git e GitHub: controle de versão e repositório do projeto;
- HTML e CSS: estrutura e estilização dos componentes com foco em legibilidade, contraste e navegação por teclado.

2.4 REQUISITOS

A definição dos requisitos do framework foi conduzida com base nas diretrizes da WCAG 2.1 e 2.2, do e-MAG (Departamento de Governo

Eletrônico, 2014) e nas legislações brasileiras vigentes, como a Lei Brasileira de Inclusão da Pessoa com Deficiência (BRASIL, 2015). O processo teve como objetivo transformar os princípios teóricos em critérios práticos para a construção dos componentes do framework, garantindo que estes fossem compatíveis com tecnologias assistivas e estivessem preparados para cenários reais de uso.

Os requisitos foram classificados em três categorias principais: requisitos funcionais, requisitos não funcionais e requisitos técnicos de acessibilidade.

2.4.1 Requisitos Funcionais

Esses requisitos definem as funcionalidades essenciais que o framework deve oferecer aos desenvolvedores:

- Disponibilizar componentes interativos acessíveis, como botões, campos de formulário, menus, modais, tooltips e alertas;
- Permitir personalização visual (tamanho da fonte, cores, contraste) sem comprometer a acessibilidade;
- Garantir que todos os componentes sejam navegáveis exclusivamente via teclado;
- Oferecer mensagens de feedback acessíveis para ações como envio de formulários e notificações em tempo real;
- Implementar recursos que possibilitem foco visível durante a navegação com tab.

2.4.2 Requisitos Não Funcionais

Estes requisitos garantem que o framework mantenha um padrão de qualidade que vá além da funcionalidade, abrangendo desempenho, manutenibilidade e conformidade com normas:

- Conformidade com as WCAG 2.1/2.2 nível AA, que inclui critérios como contraste de cores, navegação por teclado, descrições alternativas e uso de semântica HTML apropriada;
- Compatibilidade com tecnologias assistivas, como leitores de tela;
- Responsividade, para que os componentes funcionem corretamente em diferentes tamanhos de tela, inclusive em dispositivos móveis;

- Documentação acessível e detalhada, com exemplos de uso e instruções claras sobre como integrar os componentes no projeto.

2.4.3 Requisitos Técnicos de Acessibilidade

Esta categoria traduz diretamente os critérios de sucesso das WCAG (World Wide Web Consortium, 2024) em orientações práticas de implementação:

- Utilização de atributos semânticos HTML como `role`, `aria-label`, `aria-describedby`, `aria-live`, `aria-hidden`, entre outros, para descrever corretamente a funcionalidade dos elementos para usuários com deficiência;
- Estrutura hierárquica adequada com uso correto de tags como `<h1>` a `<h6>`, `<main>`, `<nav>`, `<section>`, entre outras, promovendo melhor interpretação por leitores de tela;
- Implementação de alternativas textuais para imagens, ícones e conteúdos visuais relevantes;
- Garantia de contraste mínimo de 4.5:1 entre texto e fundo, conforme critério 1.4.3 das WCAG;
- Foco visível com destaque visual suficiente para que o usuário identifique onde está a navegação via teclado.

2.5 DESENVOLVIMENTO

O desenvolvimento foi realizado de forma incremental, priorizando entregas curtas e testáveis. O processo seguiu princípios da engenharia de software orientada a componentes, com foco em reutilização, padronização e acessibilidade desde a concepção.

A estrutura foi organizada em um monorepositório Angular com suporte a bibliotecas independentes e uso de componentes *standalone*, facilitando a integração modular. Cada componente passou pelas seguintes etapas:

- Análise dos critérios da WCAG 2.1 e 2.2 e das diretrizes do e-MAG;
- Prototipação com HTML semântico e responsivo;
- Implementação com Typescript e Angular, utilizando diretivas acessíveis;

- Testes com teclado e Narrador(Leitor de tela nativo do windows);
- Testes com ferramentas como Axe e Lighthouse;
- Refatorações contínuas para garantir clareza de código, encapsulamento e responsividade.

As ferramentas Axe e Lighthouse são utilitários amplamente utilizados para a avaliação da acessibilidade em aplicações web. O Axe é uma biblioteca de código aberto que realiza varreduras automáticas baseadas nas diretrizes WCAG, identificando barreiras de acessibilidade diretamente no código da página. Já o Lighthouse, desenvolvido pelo Google, é uma ferramenta integrada ao navegador Chrome que realiza auditorias automatizadas em diversos aspectos da qualidade de uma página.

A abordagem utilizada no desenvolvimento garantiu alta coesão e baixo acoplamento, permitindo o uso dos componentes de forma independente ou combinada em diferentes interfaces. A documentação foi elaborada junto ao código, com exemplos de uso, instruções técnicas e foco na acessibilidade.

2.5.1 Arquitetura

A organização do projeto foi concebida de forma modular e escalável, utilizando as boas práticas do Angular CLI para criação de bibliotecas reutilizáveis. O framework foi estruturado dentro de um monorepositório Angular, permitindo a separação entre a biblioteca principal de componentes e aplicação (Figura 1). Cada componente foi isolado em sua própria pasta, contendo arquivos separados para lógica (.ts), template (.html) e estilos (.css). Essa abordagem favorece a reutilização, os testes e a padronização da codificação.

Figura 1 - Estrutura do framework.



Fonte: Elaborado pelo autor.

2.5.2 Desenvolvimento dos Componentes

O desenvolvimento inicial contemplou três componentes acessíveis: botão, campo de entrada e modal. Ao longo do projeto, novos componentes foram adicionados com base nas mesmas diretrizes de acessibilidade. A seguir, os componentes implementados no framework:

- **Botão acessível:** `AccessibleButtonComponent`, com semântica adequada, foco visível e suporte a ativação via teclado;
- **Campo de entrada acessível:** `AccessibleInputComponent`, com rótulo associado, foco visível e mensagem de erro;
- **Modal acessível:** `AccessibleModalComponent`, com foco preso, leitura por leitores de tela e fechamento via tecla Esc;
- **Tabela acessível:** `AccessibleTableComponent`, com elementos semânticos `<table>`, `<caption>`, `<th>`, e navegação por teclado;
- **Menu de seleção (*select*) acessível:** `AccessibleSelectComponent`, com foco visível, navegação por teclado e atributos ARIA;
- **Lista de alertas acessíveis:** `AccessibleAlertListComponent`, com geração dinâmica via lista, suporte a `aria-live` e botão de fechamento por item;
- **Componente colapsável:** `AccessibleCollapseComponent`, com botão toggle, foco gerenciado, `aria-expanded` e transição de conteúdo;
- **Dropdown acessível:** `AccessibleDropdownComponent`, navegação com setas, foco visível e suporte a leitores de tela.

Todos os componentes foram projetados com base nos princípios **POUR** (Perceptível, Operável, Compreensível e Robusto), garantindo que a biblioteca esteja em conformidade com os requisitos das diretrizes de acessibilidade vigentes.

2.5.3 Documentação

A documentação técnica do framework foi tratada como uma etapa essencial do processo de desenvolvimento, com o objetivo de garantir sua utilização eficiente, manutenção a longo prazo e suporte para desenvolvedores.

Cada componente acessível criado foi acompanhado de instruções claras sobre sua finalidade, modo de uso, parâmetros de entrada (@Input), eventos de saída (@Output) e requisitos de acessibilidade contemplados, incluindo exemplos práticos de integração.

A estrutura da documentação inclui:

- **Descrição funcional** de cada componente, explicando sua utilidade e comportamentos esperados;
- **Exemplos de uso em HTML e Typescript**, demonstrando como integrar os componentes em projetos Angular;
- **Explicação de acessibilidade**, indicando os atributos semânticos utilizados, como role e elementos nativos;
- **Lista de propriedades configuráveis** e como alterá-las no contexto de uso;
- **Boas práticas** de utilização, acessibilidade e estilo visual;
- **Organização padronizada por componente**, com visual limpo e instruções concisas.

Além disso, um resumo introdutório foi elaborado para ser exibido na página inicial do framework, explicando o propósito do projeto, os princípios de acessibilidade adotados, e os componentes atualmente disponíveis. Essa seção serve como ponto de partida para novos usuários e desenvolvedores.

A documentação também contribui para o aprendizado de acessibilidade no desenvolvimento front-end, promovendo uma cultura de inclusão digital desde a base da construção de interfaces.

2.6 FERRAMENTAS

A escolha das tecnologias utilizadas no desenvolvimento do framework acessível teve como base critérios de escalabilidade, modularidade, suporte à comunidade e compatibilidade com boas práticas de acessibilidade. Entre essas ferramentas, destacam-se o Angular e o Node.js, ambas amplamente consolidadas no ecossistema do desenvolvimento web moderno.

O Angular foi escolhido como base para a criação da biblioteca de componentes acessíveis devido à sua estrutura robusta, baseada em componentes reutilizáveis, e por oferecer suporte nativo à criação de bibliotecas modulares. Além disso, o framework proporciona uma separação clara entre lógica, apresentação e estilo, o que facilita a implementação de práticas como uso correto de semântica HTML, navegação por teclado e aplicação de atributos ARIA. A sua compatibilidade com ferramentas como Angular Material e a possibilidade de integração com práticas acessíveis, como foco gerenciado e rotas declarativas, também foram fatores determinantes.

Já o Node.js foi utilizado para estruturar o ambiente de suporte do framework, sendo responsável pelo empacotamento e publicação no NPM. Sua arquitetura orientada a eventos e seu ecossistema rico em pacotes NPM permitiram a integração com ferramentas de análise de acessibilidade, como o Pa11y e o Axe CLI, além de garantir agilidade no processo de build e deploy da biblioteca.

A combinação de Angular e Node.js mostrou-se eficiente não apenas do ponto de vista técnico, mas também por promover uma curva de aprendizado unificada para os desenvolvedores, dado que ambas as tecnologias utilizam JavaScript/TypeScript como linguagem base. Isso contribuiu para a consistência do código, facilidade de manutenção e possibilidade de adoção em projetos reais com equipes multidisciplinares.

3 DISCUSSÃO E RESULTADOS

3.1 Testes e Validação

Para garantir a conformidade do framework com os princípios de acessibilidade descritos na WCAG 2.1 e 2.2 (World Wide Web Consortium, 2024) e no modelo e-MAG (Departamento de Governo Eletrônico, 2014), foram realizados testes automatizados e manuais com foco na usabilidade por pessoas com deficiência e no suporte a tecnologias assistivas.

3.1.1 Testes Automatizados com Axe DevTools

Utilizou-se a ferramenta Axe DevTools, uma extensão para navegadores baseada na biblioteca *axe-core*, para validar automaticamente aspectos técnicos de acessibilidade. Os testes foram realizados diretamente na aplicação em execução, analisando as diferentes páginas com diferentes combinações de componentes acessíveis. Entre os critérios avaliados estavam:

- Existência de rótulos e nomes acessíveis;
- Contraste mínimo de cores (nível AA e AAA);
- Uso adequado de atributos ARIA (*aria-label*, *aria-expanded*);
- Presença de elementos com foco programado e foco visível;
- Adequação aos padrões de acessibilidade estabelecidos pelas WCAG 2.2.

O resultado evidencia a conformidade total do framework com a WCAG 2.1 nível AA, os resultados obtidos são apresentados na Tabela 1.

Tabela 1 – Resultados dos testes automatizados com a ferramenta Axe DevTools.

Categoria	Descrição
Problemas críticos	0
Problemas sérios	0
Problemas moderados	0
Problemas menores	0
Conformidade	WCAG 2.1 AA

Fonte: Dados obtidos por meio de teste automatizado com Axe DevTools.

3.1.2 Testes Automatizados com Lighthouse

Para validar a acessibilidade do framework, foi realizado um teste automatizado com a ferramenta Lighthouse, amplamente reconhecida por sua eficiência na verificação de conformidade com as diretrizes da WCAG 2.2 (World Wide Web Consortium, 2024).

O resultado obtido foi altamente satisfatório, com uma pontuação máxima de **100/100**, indicando a ausência de falhas automatizáveis. Além disso, a ferramenta indicou diversas boas práticas implementadas corretamente:

1. Touch targets possuem tamanho e espaçamento suficientes;
2. Células de tabela referem-se corretamente com o atributo `headers`;
3. Elementos de título aparecem em ordem sequencial descendente;
4. Uso de ARIA roles apenas em elementos compatíveis;
5. Roles ARIA obsoletas não foram utilizadas;
6. Imagens não possuem textos alternativos redundantes;
7. Tabelas possuem conteúdos diferentes no atributo `summary` e na `<caption>`;
8. Somente atributos ARIA permitidos foram utilizados;
9. Valores de `role` são válidos;
10. Cores de fundo e texto possuem contraste suficiente;
11. Documento contém o elemento `<title>`;
12. Formulários possuem rótulos associados;
13. Listas contêm apenas elementos válidos e suportados;
14. Itens de lista estão contidos em elementos de lista apropriados;
15. Elementos `<select>` possuem rótulos associados;
16. Nenhum elemento possui valor `tabindex` superior a 0;
17. Atributos ARIA estão adequadamente especificados conforme o papel do elemento;
18. Botões possuem nomes acessíveis;
19. Imagens possuem atributos `alt`.

Este resultado confirma que a aplicação está plenamente alinhada às boas práticas de acessibilidade, assegurando uma experiência inclusiva para todos os usuários.

3.1.3 Testes Manuais com Teclado e Leitores de Tela

Além dos testes automatizados, foram realizados testes manuais com:

- Navegação por teclado (Tab, Shift+Tab, Enter, Esc, setas);
- Leitor de tela Narrador do Windows;
- Simulação de navegação em dispositivos móveis com foco tátil.

Os resultados confirmaram que todos os componentes implementados atendem aos requisitos básicos de acessibilidade para navegação não visual, com destaque para:

- Foco gerenciado corretamente no modal e dropdown;
- Uso correto de `role="alert"` e `aria-live` nos balões de mensagem;
- Leitura adequada das opções em `select`, `dropdown` e `collapse`;
- Foco visual claramente perceptível em todos os elementos interativos.

3.2 Comparativo com outras Soluções

Ao comparar com soluções como o AccessHub e bibliotecas como Material, o framework proposto apresentou diferenças significativas (Tabela 2).

Tabela 2 – Comparação entre o AccessHub, Material UI e o framework desenvolvido quanto a aspectos de acessibilidade.

Critério	AccessHub	Material UI	Framework
Componentes prontos	Não	Sim	Sim
Validação de acessibilidade	Sim	Parcial	Sim
Foco em acessibilidade	Alto	Médio	Alto
Suporte a leitores de tela	Manual	Parcial	Completo

Fonte: Elaborado pelo autor.

3.3 Limitações Identificadas

Apesar dos resultados positivos, algumas limitações foram observadas:

- Necessidade de conhecimento básico de Angular para integração;
- Falta de suporte inicial a internacionalização (i18n);
- Compatibilidade reduzida com navegadores legados como o Internet Explorer.

Essas limitações serão abordadas nas próximas versões do framework com o intuito de aumentar a área de atuação da ferramenta e alcançar mais usuários.

3.4 Análise Comparativa

Os resultados obtidos com o framework proposto demonstram avanços importantes em relação aos trabalhos correlatos analisados na revisão da literatura. Enquanto o AccessHub (SILVA, 2023) se limita a fornecer um guia textual com recomendações de acessibilidade, exigindo interpretação e implementação manual por parte do desenvolvedor, a proposta deste trabalho entrega componentes prontos, testados e documentados, com acessibilidade integrada.

Essa diferença prática contribui diretamente para superar a lacuna identificada por JÚNIOR e PEREIRA (2019), que apontaram a baixa aplicação das diretrizes WCAG mesmo entre profissionais que as conhecem. Ao contrário do cenário descrito por Bi et al. (2021), em que a acessibilidade é frequentemente tratada de forma superficial e relegada às etapas finais, o framework apresentado neste trabalho incorpora critérios acessíveis desde a concepção dos componentes, favorecendo sua adoção sistemática em projetos reais. Assim, os achados desta pesquisa não apenas confirmam os desafios relatados por esses autores, como também propõem uma solução concreta e validada para enfrentá-los.

4 CONCLUSÃO

Ao longo do trabalho, foi possível identificar, por meio da revisão bibliográfica e da análise de soluções existentes, uma lacuna significativa entre a teoria da acessibilidade digital e sua aplicação prática no desenvolvimento de sistemas. A implementação do framework buscou preencher essa lacuna ao disponibilizar componentes reutilizáveis, com acessibilidade

embutida, capazes de facilitar a adoção de boas práticas por desenvolvedores, inclusive aqueles com pouca familiaridade com as normas técnicas.

Além de atender aos critérios técnicos e legais, a proposta contribui também para a democratização do acesso à informação e à tecnologia, ampliando a participação digital de pessoas com deficiência. A acessibilidade deixou de ser apenas uma obrigação normativa, tornando-se uma característica essencial de qualidade e responsabilidade social.

Como trabalhos futuros, propõe-se a ampliação da biblioteca com mais componentes acessíveis, como carrosséis, abas, tooltips, sliders e elementos interativos frequentemente utilizados em aplicações web modernas. A inclusão desses novos elementos permitirá que o framework atenda a uma variedade maior de cenários de interface, mantendo sempre a conformidade com os critérios técnicos das WCAG.

Pretende-se também incorporar suporte à internacionalização (i18n), viabilizando a tradução de textos, mensagens de erro e rótulos acessíveis para múltiplos idiomas. Tal recurso tornará o framework mais adequado a projetos multilíngues, aplicações públicas e iniciativas com abrangência internacional.

Outra linha de continuidade promissora consiste na reimplimentação do framework utilizando o ecossistema do React, uma das bibliotecas mais populares para desenvolvimento de interfaces web. Dessa forma, expandir o framework representa não apenas uma questão de interoperabilidade, mas também de ampliação do seu impacto e aplicabilidade no mercado.

Por fim, espera-se que este trabalho sirva de base para outras iniciativas voltadas à acessibilidade digital, fomentando uma cultura de desenvolvimento centrada na inclusão e no respeito à diversidade de usuários.

REFERÊNCIAS

BERNERS-LEE, T. **W3C Launches Web Accessibility Initiative**. [S.l.]: W3C, 1997.

BI, T. et al. **Accessibility in Software Practice: A Practitioner's Perspective**. ACM Transactions on Software Engineering and Methodology, 2021, ACM, p. 1–27.

BRASIL. **Lei nº 13.146, de 6 de julho de 2015. Estatuto da Pessoa com Deficiência**. 2015. Disponível em: <https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2015/lei/l13146.htm>. Acesso em: 15 abr. 2025.

Departamento de Governo Eletrônico. **eMAG – Modelo de Acessibilidade em Governo Eletrônico**. [S.l.]: Ministério do Planejamento, Orçamento e Gestão, 2014. Disponível em: <<https://emag.governoeletronico.gov.br/>>. Acesso em: 27 abr. 2025.

Instituto Brasileiro de Geografia e Estatística. **Censo 2022: Brasil tem 14,4 milhões de pessoas com deficiência**. [S.l.]: Agência de Notícias IBGE, 2023. Disponível em: <<https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/43463-censo-2022-brasil-tem-14-4-milhoes-de-pessoas-com-deficiencia>>. Acesso em: 1 jun. 2025.

JÚNIOR, J. F.; PEREIRA, A. A. **Análise e soluções de acessibilidade Web para deficientes visuais utilizando os padrões W3C**. Revista Científica Fagoc Multidisciplinar, 2019, v. 4.

NORMAN, D. A. **The Design of Everyday Things**. Revised and expanded ed. Cambridge: MIT Press, 2013.

Organização Mundial da Saúde. **Mundo tem 1 bilhão de pessoas com deficiência excluídas de tecnologias de apoio**. [S.l.]: ONU News, 2022. Disponível em: <<https://news.un.org/pt/story/2022/05/1789172>>. Acesso em: 1 jun. 2025.

SILVA, T. **AccessHub: Guia de acessibilidade para desenvolvedores**. 2023. Acesso em: 15 abr. 2025.

World Wide Web Consortium. **Web Content Accessibility Guidelines (WCAG) 2.2**. 2024. Disponível em: <<https://www.w3.org/TR/WCAG22/>>. Acesso em: 20 mar. 2025.