INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAIANO – CAMPUS CATU

ALEXANDRE DA SILVA ARAUJO

MODELO COMPUTACIONAL BASEADO EM INTELIGÊNCIA ARTIFICIAL PARA ANÁLISE DE ACESSIBILIDADE WEB ALINHADO ÀS DIRETRIZES WCAG 2.2 E eMAG 3.1

CATU – BA

2026

ALEXANDRE DA SILVA ARAUJO

MODELO COMPUTACIONAL BASEADO EM INTELIGÊNCIA ARTIFICIAL PARA ANÁLISE DE ACESSIBILIDADE WEB ALINHADO ÀS DIRETRIZES WCAG 2.2 E eMAG 3.1

Trabalho de Conclusão de Curso apresentado ao Instituto Federal de Educação, Ciência e Tecnologia Baiano – Campus Catu, como requisito parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. André Luiz Rezende
Coorientador: Prof. Romero Mendes Freire de Moura Júnior

CATU – BA
2026

Instituto Federal de Educação, Ciência e Tecnologia Baiano – Campus Catu
Setor de Biblioteca

A663 Araujo, Alexandre da Silva
Modelo computacional baseado em inteligência artificial para análise de acessibilidade web alinhado às diretrizes WCAG 2.2 E eMaG 3.1 / Alexandre da Silva Araújo. – 2026.

75 f.:

Orientador(a): Prof. André Luiz Rezende.
Co-orientador(a): Prof. Romero Mendes Freire de Moura Júnior

TCC (monografia), Instituto Federal de Educação, Ciência e Tecnologia Baiano, Tecnólogo em Análise e Desenvolvimento de Sistemas, Catu, 2026.

1. Acessibilidade web. 2. Inteligência artificial. 3. Auditoria automatizada. I. Rezende, André Luiz. II. Moura Júnior, Romero Mendes Freire de. III. Título.

CDU: 004.5

Resumo

Este trabalho apresenta o desenvolvimento de um modelo computacional para auditoria automatizada de acessibilidade web, baseado em inteligência artificial e alinhado às diretrizes WCAG 2.2 e ao Modelo de Acessibilidade em Governo Eletrônico (eMAG 3.1). O sistema foi implementado em Python 3.12, adotando uma arquitetura organizada em 15 módulos, e integrou modelos de inteligência artificial para descrição de imagens (BLIP) e classificação contextual de links (BART), além de técnicas de renderização dinâmica via Selenium para análise de contraste, zoom e tamanho de alvos. Foram implementados 15 critérios de acessibilidade, abrangendo os quatro princípios da WCAG 2.2 e os seis grupos do eMAG 3.1. Os relatórios são gerados em múltiplos formatos (PDF, PNG e TXT), oferecendo ao usuário a flexibilidade de escolher entre os referenciais WCAG ou eMAG, com adaptação automática das recomendações e da classificação de severidade. A avaliação experimental foi realizada por meio da auditoria de 15 sites de diferentes categorias, comparando os resultados com as ferramentas consolidadas eMAG ASES e WAVE. Os experimentos indicaram que o sistema proposto apresentou maior taxa de detecção nos sites analisados, com destaque para conteúdo dinâmico, contraste real e critérios da WCAG 2.2, além de gerar sugestões de correção de código.

Palavras-chave: Acessibilidade web. WCAG 2.2. eMAG 3.1. Inteligência artificial. Auditoria automatizada.

Abstract

This work presents the development of a computational model for automated web accessibility auditing, based on artificial intelligence and aligned with the WCAG 2.2 guidelines and the Brazilian Electronic Government Accessibility Model (eMAG 3.1). The system was implemented in Python 3.12, adopting a modular architecture organized into 15 modules, integrating AI models for image description (BLIP) and contextual link classification (BART), along with dynamic rendering techniques using Selenium for contrast, zoom, and target size analysis. Fifteen accessibility criteria were implemented, covering the four principles of WCAG 2.2 and the six groups of eMAG 3.1. Reports are generated in multiple formats (PDF, PNG, and TXT), offering users the flexibility to choose between WCAG or eMAG standards, with automatic adaptation of recommendations and severity classification. Experimental evaluation was conducted by auditing 15 websites from different categories, comparing the results with the consolidated tools eMAG ASES and WAVE. The experiments indicated that the proposed system achieved a higher detection rate in the analyzed sites, with emphasis on dynamic content, real contrast calculation, and WCAG 2.2 criteria, in addition to generating code correction suggestions.

Keywords: Web accessibility. WCAG 2.2. eMAG 3.1. Artificial intelligence. Automated auditing.

LISTA DE TABELAS E FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

Sigla	Significado
ABNT	Associação Brasileira de Normas Técnicas
ARIA	Accessible Rich Internet Applications
ASES	Avaliador e Simulador de Acessibilidade em Sítios
BART	Bidirectional and Auto-Regressive Transformer
BLIP	Bootstrapping Language-Image Pre-training
CAPTCHA	Completely Automated Public Turing test to tell Computers and Humans Apart
CSS	Cascading Style Sheets
DOM	Document Object Model
eMAG	Modelo de Acessibilidade em Governo Eletrônico
GPU	Graphics Processing Unit
HTML	HyperText Markup Language
IA	Inteligência Artificial
JPEG	Joint Photographic Experts Group
PDF	Portable Document Format
PNG	Portable Network Graphics
Selenium	Ferramenta de automação de navegadores
SVG	Scalable Vector Graphics
TXT	Arquivo de texto
W3C	World Wide Web Consortium
WAI	Web Accessibility Initiative
WAVE	Web Accessibility Evaluation Tool
WCAG	Web Content Accessibility Guidelines

SUMÁRIO

1. INTRODUÇÃO

1.1 Contextualização do tema

A evolução da web nas últimas décadas transformou profundamente a sociedade, tornando a internet um dos principais meios de acesso à informação, serviços, educação e comércio. No entanto, essa revolução digital não alcança todos os usuários de maneira igualitária. Pessoas com deficiência frequentemente enfrentam barreiras significativas ao navegar em páginas *web*, seja pela falta de alternativas textuais para conteúdos não textuais, pela ausência de descrições adequadas em *links*, pela implementação inadequada de elementos interativos ou pelo design de interfaces que não consideram diferentes formas de interação.

Nesse contexto, a acessibilidade digital emerge como um direito fundamental e uma necessidade técnica incontornável. A Organização Mundial da Saúde (OMS) estima que mais de 1 bilhão de pessoas vivem com alguma forma de deficiência, representando cerca de 15% da população global. No Brasil, dados do Censo Demográfico revelam a magnitude desse público: embora o levantamento de 2010 tenha apontado que cerca de 24% da população possuía algum grau de dificuldade funcional, dados mais recentes e refinados do Censo 2022 indicam que pelo menos 14,4 milhões de brasileiros (8,9% da população) possuem deficiências severas (IBGE, 2022). Estes números evidenciam a importância de se garantir que o ambiente digital seja, de fato, um espaço inclusivo e acessível para todos.

Para enfrentar esse desafio, foram desenvolvidas diretrizes que estabelecem padrões para tornar o conteúdo *web* mais acessível. Em âmbito internacional, as Web Content Accessibility Guidelines (WCAG), elaboradas pelo World Wide Web Consortium (W3C) por meio de sua iniciativa Web Accessibility Initiative (WAI), constituem o principal referencial técnico para acessibilidade digital. A versão mais recente, WCAG 2.2, publicada em outubro de 2023, introduz novos critérios que abordam aspectos como tamanho de alvos (2.5.8) e autenticação acessível (3.3.8), ampliando significativamente o escopo de avaliação e as possibilidades de inclusão.

No contexto brasileiro, destaca-se o Modelo de Acessibilidade em Governo Eletrônico (eMAG), desenvolvido pelo Governo Federal. Instituído pela Portaria nº 3, de 7 de maio de 2007, o eMAG estabelece recomendações e requisitos de acessibilidade para sites e portais governamentais, alinhando-se às melhores práticas internacionais e considerando as particularidades da legislação e cultura nacionais. A versão atual, eMAG 3.1, publicada em abril de 2014, organiza suas recomendações em seis grupos, Marcação, Comportamento, Conteúdo/Informação, Apresentação/Design, Multimídia e Formulário, representando um esforço significativo para promover a inclusão digital no âmbito do setor público brasileiro.

A interseção entre acessibilidade digital e inteligência artificial tem se mostrado promissora, como evidenciado por pesquisas recentes. Destaca-se o trabalho de *Yu et al*. (2025), que explora o uso de IA generativa para aprimorar a experiência de usuários de leitores de tela em sites de e-commerce. Enquanto aquela pesquisa se concentra na experiência do usuário final, o presente trabalho adota uma perspectiva complementar, focando no desenvolvedor. Propõe-se um modelo computacional baseado em inteligência artificial capaz de analisar automaticamente páginas *web*, identificar inconformidades tanto com as diretrizes WCAG 2.2 quanto com as recomendações do eMAG 3.1, e gerar sugestões de correção contextualizadas, incluindo o código já corrigido atuando como ferramenta de suporte para desenvolvedores e auditores.

Apesar da existência de diretrizes consolidadas, observa-se que muitos sites ainda apresentam barreiras de acessibilidade significativas. Estudos recentes, como os de Silva *et al*. (2023) e Nascimento *et al*. (2023), demonstram que mesmo plataformas de grande porte e sites governamentais frequentemente deixam de atender a critérios básicos de acessibilidade, evidenciando a lacuna entre as diretrizes existentes e sua efetiva implementação.

As ferramentas tradicionais de auditoria, baseadas estritamente em regras pré-definidas (como o Avaliador e Simulador de Acessibilidade em Sítios eMAG ASES), apresentam limitações importantes. Embora sejam eficientes para detectar problemas sintáticos como a ausência do atributo *alt* em imagens ou a falta de associação entre rótulos e campos de formulário, esses sistemas possuem dificuldade em identificar nuances contextuais que impactam diretamente a experiência do usuário. Por exemplo, um *link* representado apenas por um ícone pode ser visualmente identificável, mas permanece inacessível para

usuários de leitores de tela se não houver um texto alternativo adequado ou um atributo *aria-label* que descreva seu propósito. Além disso, a análise de contraste de cores exige a interpretação de cores computadas após a aplicação de estilos CSS, o que demanda renderização dinâmica da página, funcionalidade não implementada por ferramentas de análise estática.

Diante desse cenário, esta pesquisa classifica-se como aplicada, pois visa solucionar um problema prático relacionado à auditoria de acessibilidade web. Quanto à abordagem, adota natureza quali-quantitativa, combinando métricas objetivas de detecção com análises qualitativas das sugestões geradas pela IA. Em relação aos objetivos, caracteriza-se como exploratória e descritiva, uma vez que envolve o desenvolvimento de um artefato computacional funcional e a análise de seu comportamento em ambientes web reais. A construção do sistema seguiu as fases da Metodologia PDS (Projeto e Desenvolvimento de Sistemas) e sua avaliação experimental foi realizada por meio de testes em 15 sites de diferentes categorias, comparando os resultados com as ferramentas consolidadas eMAG ASES e WAVE.

O presente trabalho propõe um modelo computacional que busca endereçar as limitações das ferramentas tradicionais ao empregar técnicas de renderização dinâmica (Selenium), inteligência artificial generativa (BLIP para descrição de imagens e BART para classificação contextual de links) e um sistema híbrido de tradução com fallback offline, capaz de gerar relatórios profissionais (PDF, PNG e TXT com sugestões de código corrigido) e pontuações calibradas para ambos os referenciais (WCAG 2.2 e eMAG 3.1).

Este trabalho está organizado em cinco capítulos. O Capítulo 2 apresenta a fundamentação teórica, abordando os conceitos de acessibilidade digital, as diretrizes WCAG 2.2 e o modelo eMAG 3.1, bem como os trabalhos correlatos. O Capítulo 3 descreve a metodologia da pesquisa, incluindo a classificação do estudo, a abordagem de desenvolvimento e os procedimentos de coleta e análise dos dados. O Capítulo 4 detalha o desenvolvimento e a arquitetura do sistema, apresentando os módulos implementados, os critérios de acessibilidade verificados e as tecnologias empregadas. O Capítulo 5 apresenta a discussão dos resultados obtidos nos experimentos, comparando o sistema proposto com as ferramentas eMAG ASES e WAVE. Por fim, o Capítulo 6 traz as considerações finais, sintetizando as contribuições do trabalho, suas limitações e apontando direções para pesquisas futuras.

1.2 Problema de pesquisa

A baixa conformidade nos *websites* para com as diretrizes WCAG 2.2 e eMAG 3.1, aliada às limitações das ferramentas tradicionais de auditoria baseadas em análise estática, as quais não detectam adequadamente inconformidades em conteúdos dinâmicos, contraste real e critérios funcionais como *zoom* e tamanho de alvos.

1.3 Hipótese

A utilização de um modelo computacional baseado em inteligência artificial (BLIP e BART), aliada a técnicas de análise dinâmica (Selenium) e regras estruturais (WCAG 2.2 e eMAG 3.1), possibilita a identificação de inconformidades de acessibilidade com maior abrangência do que ferramentas tradicionais baseadas exclusivamente em regras estáticas, como o eMAG ASES e o WAVE.

1.4 Justificativa

A relevância deste trabalho situa-se na interseção entre duas áreas em crescente desenvolvimento: a acessibilidade digital e a inteligência artificial.

Do ponto de vista social, a pesquisa contribui para a promoção da inclusão digital no contexto brasileiro, em conformidade com a Lei Brasileira de Inclusão (Lei nº 13.146/2015), ao alinhar-se tanto às diretrizes internacionais (WCAG 2.2) quanto às especificidades do Modelo de Acessibilidade em Governo Eletrônico (eMAG 3.1). Ao fornecer subsídios técnicos para que desenvolvedores, designers e equipes de qualidade possam identificar e corrigir problemas de acessibilidade com maior precisão e eficiência, o modelo computacional proposto atende a uma demanda concreta do setor público brasileiro, que tem a obrigatoriedade de cumprir as recomendações do eMAG em seus portais e serviços digitais. A remoção de barreiras digitais é um passo fundamental para garantir que pessoas com deficiência possam exercer plenamente sua cidadania, acessando informações, serviços e oportunidades em igualdade de condições, conforme preconiza a referida lei.

Do ponto de vista técnico e científico, o trabalho apresenta originalidade ao propor a integração de modelos de inteligência artificial, especificamente o BLIP para descrição de

imagens e o BART para classificação contextual de *links*, a um modelo computacional de auditoria de acessibilidade. Esta abordagem híbrida, que combina verificação baseada em regras com análise contextual inteligente, busca reduzir as limitações das ferramentas tradicionais, oferecendo não apenas a detecção de inconformidades, mas também sugestões de correção contextualizadas com exemplos de código HTML e CSS.

Além disso, a pesquisa se posiciona como complemento a trabalhos recentes que exploram o uso de inteligência artificial para acessibilidade web, como o estudo de *Yu et al*. (2025), que aplica IA generativa para melhorar a experiência de usuários de leitores de tela em e-commerce. Enquanto aquele trabalho foca na experiência do usuário final, o modelo computacional aqui proposto se concentra na fase de auditoria e correção, atuando como ferramenta de suporte para desenvolvedores antes da publicação do conteúdo.

1.5 Objetivos

1.5.1 Objetivo geral

Desenvolver um modelo computacional baseado em inteligência artificial capaz de analisar páginas *web*, identificar inconformidades relacionadas às diretrizes WCAG 2.2 e ao Modelo de Acessibilidade em Governo Eletrônico (eMAG 3.1), e gerar recomendações técnicas contextualizadas para correção.

1.5.2 Objetivos específicos

• Realizar um levantamento bibliográfico sobre acessibilidade digital, diretrizes WCAG 2.2, eMAG 3.1 e aplicações de inteligência artificial no contexto *web*;

• Definir e implementar módulos de verificação para 15 critérios que abrangem os 6 grupos do eMAG 3.1 e os níveis A, AA e AAA da WCAG 2.2:

• **1.1.1** (Imagens) - grupo multimídia – Nível A
• **1.3.1** (*Headings*, Listas e *Landmarks*) - grupo conteúdo/Informação – Nível A
• **1.3.2** (Ordem de leitura) - grupo conteúdo/Informação – Nível A
• **1.4.3** (Contraste) - grupo apresentação/design – Nível AA

• **1.4.4** (*Zoom*) - grupo comportamento – Nível AA
• **1.4.12** (Espaçamento) - grupo conteúdo/informação – Nível AA
• **1.4.13** (Conteúdo sob foco ou mouse) - grupo apresentação/design – Nível AA
• **2.4.3** (Ordem de foco - *tabindex*) - grupo marcação – Nível A
• **2.4.4** (*Links*) - grupo marcação – Nível A
• **2.5.8** (Tamanho de alvos) - grupo comportamento – Nível AA
• **3.1.1** (Idioma da página) - grupo conteúdo/Informação – Nível A
• **3.2.5** (Nova aba sem aviso) - grupo marcação – Nível AAA
• **3.3.2** (Formulários) - grupo formulário – Nível A
• **3.3.8** (Autenticação/CAPTCHA) - grupo formulário – Nível AA
• **4.1.2** (ARIA) - grupo marcação – Nível A

• Integrar modelos de inteligência artificial BLIP (B*ootstrapping-Language-Image-Pre-training*) para descrição de imagens e BART (*Bidirectional and Auto-Regressive-Transformer*) para classificação contextual de *links* ao modelo computacional de auditoria.

• Desenvolver um mecanismo de pontuação e classificação de severidade dos problemas encontrados, permitindo avaliação quantitativa da acessibilidade;

• Implementar a geração de relatórios em múltiplos formatos (PNG, PDF e TXT com código corrigido), contendo visualizações gráficas e listas detalhadas de ocorrências, com identificação dos grupos eMAG correspondentes;

• Realizar validação conceitual da arquitetura proposta por meio de estudos de caso em sites brasileiros, comparando os resultados com ferramentas consolidadas (eMAG ASES e WAVE).

2. FUNDAMENTAÇÃO TEÓRICA

2.1 Procedimentos de busca e seleção

Para a construção do referencial teórico deste trabalho, foi realizada uma revisão bibliográfica sistemática exploratória nas principais bases de dados acadêmicas, incluindo Scopus, Web of Science, Google Acadêmico e o Portal de Periódicos da CAPES. A busca abrangeu publicações dos últimos cinco anos, utilizando combinações dos descritores: "acessibilidade digital", "WCAG", "eMAG", "inteligência artificial", "auditoria automatizada", "interface *web* acessível" e "deficiência visual".

Foram selecionados inicialmente 47 artigos, dos quais 6 foram escolhidos como fundamentação para esta pesquisa, considerando os seguintes critérios de inclusão: (i) aderência ao tema de acessibilidade digital; (ii) abordagem de ferramentas automatizadas ou inteligência artificial; (iii) relevância para o contexto brasileiro ou internacional; e (iv) disponibilidade do texto completo.

2.1.1 Panorama geral dos trabalhos selecionados

Os seis trabalhos selecionados distribuem-se em três eixos temáticos principais: (i) estudos sobre acessibilidade em plataformas de comércio eletrônico; (ii) aplicações de inteligência artificial na acessibilidade *web*; e (iii) propostas de recomendações e soluções para interfaces acessíveis. A seguir, apresenta-se uma breve descrição de cada trabalho, destacando seus objetivos e contribuições conceituais para esta pesquisa.

2.1.2 Estudos sobre acessibilidade em e-commerce

O primeiro trabalho, intitulado "Acessibilidade Digital e o e-commerce: um estudo exploratório em quatro plataformas de marketplace no Brasil", de autoria de Douglas Martins da Silva, Patrícia Papalardi Siqueira e Adriana Alvarenga Dezani (2023), propõe-se a investigar as barreiras de acessibilidade presentes em *marketplaces* nacionais. Os autores fundamentam sua análise nas diretrizes WCAG e no eMAG, buscando identificar padrões de inconformidades que impactam a experiência de usuários com deficiência.

Este trabalho contribui conceitualmente para a definição dos critérios de análise adotados no presente projeto, especialmente no que tange à seleção de elementos HTML a serem verificados. O estudo de Lays de Paiva Nascimento, Matheus Rodrigues Caraça e Mariangela Ferreira Fuentes Molina (2023), intitulado "Teste de Acessibilidade Web para Deficientes Visuais em Sites de E-commerce Utilizando Access Monitor", apresenta uma metodologia de testes de acessibilidade com foco em usuários com deficiência visual, utilizando a ferramenta automatizada *Access Monitor*. A abordagem metodológica dos autores, que combina verificação automatizada com análise contextual, serviu como referência para a arquitetura híbrida (regras + IA) proposta neste trabalho.

Ítalo José Bastos Guimarães, Marckson Roberto Ferreira de Sousa e Levi Cadmiel Amaral da Costa (2021), em "Recomendações de acessibilidade em sites de comércio eletrônico para usuários cegos", elencam um conjunto de recomendações específicas para tornar sites de *e-commerce* acessíveis a usuários cegos.

Os autores baseiam-se nas diretrizes WCAG e em normas técnicas nacionais e internacionais para formular sugestões práticas de implementação. Este trabalho fornece subsídios conceituais para as recomendações geradas automaticamente pelo modelo computacional desenvolvido.

2.1.3 Aplicações de inteligência artificial na acessibilidade web

O estudo internacional "From Cluttered to Clear: Improving the Web Accessibility Design for Screen Reader Users in E-commerce With Generative AI", de Yaman Yu, Bektur Ryskeldiev, Ayaka Tsutsui, Matthew Gillingham e Yang Wang (2025), explora o uso de IA generativa para melhorar a acessibilidade de interfaces *web* para usuários de leitores de tela. Os autores demonstram como modelos de linguagem podem ser utilizados para gerar descrições alternativas e identificar problemas contextuais, inspirando a abordagem híbrida adotada neste trabalho, que integra os modelos BLIP (para descrição de imagens) e BART (para classificação contextual de *links*). Este trabalho é particularmente relevante por estabelecer uma ponte direta entre técnicas de inteligência artificial e a solução de problemas de acessibilidade, alinhando-se perfeitamente aos objetivos desta pesquisa.

2.1.4 Soluções práticas e desafios de UI/UX

Anderson Canale Garcia (2023), em "AALT: um *framework* com soluções práticas para melhorar a acessibilidade em aplicativos Android", propõe um *framework* com recomendações e boas práticas para desenvolvimento de aplicativos acessíveis. Embora o foco seja Android, as soluções apresentadas podem ser adaptadas para o contexto *web*, especialmente no que se refere à implementação de componentes interativos acessíveis. O trabalho oferece subsídios para a formulação das recomendações técnicas geradas pelo modelo computacional.

Por fim, Otavio Matheus Neves, Rubio Rodrigo Costa, Tathiana Amarante Duarte e Vitor Hugo Furtado (2024), em "Interfaces Acessíveis para Pessoas com Deficiência Visual: Desafios e Soluções em *UI/UX*", abordam os desafios de *design* de interfaces acessíveis, discutindo princípios de usabilidade e experiência dos usuários aplicados a pessoas com deficiência visual. O trabalho contribui para a compreensão das necessidades dos usuários e para a formulação de recomendações mais eficazes, complementando a perspectiva técnica com uma visão centrada no usuário.

2.2 Síntese das contribuições conceituais

Os seis trabalhos correlatos apresentados fornecem uma base conceitual sólida para esta pesquisa, abordando desde a identificação de barreiras de acessibilidade até a aplicação de técnicas de inteligência artificial para mitigá-las. A Tabela 1 apresenta uma síntese dos principais trabalhos relacionados à acessibilidade digital em plataformas *web* e aplicações móveis.

Tabela 1 – Síntese dos trabalhos correlatos

Autor(es)	Ano	Temática principal	Contribuição conceitual
Silva, D. M.; Siqueira, P. P.; Dezani, A. A.	2023	Acessibilidade em plataformas de marketplace	Identificação de padrões de inconformidades em e-commerces brasileiros
Nascimento, L. P.; Caraça, M. R.; Molina, M. F. F.	2023	Testes de acessibilidade em e-commerce utilizando Access Monitor	Aplicação de metodologia de verificação automatizada para avaliação de acessibilidade
Guimarães, I. J. B.; Sousa, M. R. F.; Costa, L. C. A.	2021	Recomendações de acessibilidade para usuários cegos em e-commerce	Proposição de diretrizes para melhoria da navegação e interação em plataformas digitais
Yu, Y.; Ryskeldiev, B.; Tsutsui, A.; Gillingham, M.; Wang, Y.	2025	Aplicação de inteligência artificial generativa na acessibilidade web	Uso de IA para aprimorar a experiência de usuários de leitores de tela em interfaces web
Garcia, A. C.	2023	Framework de acessibilidade para aplicativos Android	Proposta de soluções práticas para desenvolvimento de interfaces acessíveis em dispositivos móveis
Neves, O. M.; Costa, R. R.; Duarte, T. A.; Furtado, V. H.	2024	Desafios de UI/UX para pessoas com deficiência visual	Análise das barreiras de usabilidade e recomendações para interfaces acessíveis

Fonte: Elaborado pelo autor.

Observa-se que diversos estudos investigam a presença de barreiras de acessibilidade em ambientes virtuais e propõem diretrizes ou metodologias de avaliação, conforme sintetizado na Tabela 1. Alguns trabalhos também exploram o uso de técnicas de inteligência artificial para melhorar a experiência de usuários que utilizam tecnologias assistivas, como o estudo de *Yu et al*. (2025). Entretanto, ainda são limitadas as pesquisas que integram mecanismos automatizados de auditoria de acessibilidade com modelos de inteligência artificial capazes de interpretar elementos do código HTML e gerar recomendações técnicas para correção. Nesse contexto, o presente trabalho propõe

o desenvolvimento de um modelo computacional que combina análise automatizada de páginas *web* com modelos de inteligência artificial, utilizando como referência as diretrizes da Web Content Accessibility Guidelines (WCAG 2.2) e do Modelo de Acessibilidade em Governo Eletrônico (eMAG 3.1).

2.3 Posicionamento do trabalho

A análise dos trabalhos correlatos (Tabela 1) revela que, embora existam ferramentas de auditoria baseadas em regras e estudos que diagnosticam barreiras de acessibilidade, há uma lacuna na integração de inteligência artificial para a geração de recomendações contextualizadas e no suporte a dois referenciais distintos (WCAG e eMAG). O presente trabalho busca contribuir na redução dessa lacuna ao propor um modelo computacional que:

• Integra os modelos BLIP e BART para gerar descrições de imagens e classificar *links*;

• Suporta dois referenciais de acessibilidade (WCAG 2.2 e eMAG 3.1), adaptando recomendações e classificação de severidade conforme a escolha do usuário;

• Gera relatórios em múltiplos formatos (PDF, PNG e TXT com código corrigido), com visualizações adaptadas ao referencial selecionado.

Dessa forma, o trabalho se posiciona como um complemento às pesquisas existentes, oferecendo uma ferramenta prática e contextualizada para auditoria de acessibilidade *web* no contexto brasileiro, alinhada às recomendações do eMAG e às diretrizes internacionais da WCAG.

2.4 Acessibilidade digital: conceitos fundamentais

2.4.1 Definição e importância

A acessibilidade digital refere-se à prática de projetar e desenvolver *websites*, aplicações e conteúdos digitais de modo que possam ser utilizados por todas as pessoas, independentemente de suas capacidades físicas, sensoriais ou cognitivas. Conforme definido pelo World Wide Web Consortium (W3C, 2023), a acessibilidade web envolve a

remoção de barreiras que impedem ou dificultam a interação e o acesso à informação por pessoas com deficiência.

2.4.2 Tecnologias assistivas

Para compreender a acessibilidade digital, é essencial conhecer as principais tecnologias assistivas utilizadas por pessoas com deficiência, dentre as quais se destacam:

• **Leitores de tela**: softwares que interpretam o conteúdo da tela e o convertem em voz ou braille, permitindo que usuários cegos ou com baixa visão naveguem pela web. No contexto brasileiro, destacam-se o DOSVOX (desenvolvido pela UFRJ), o NVDA (*NonVisual Desktop Access*) e o JAWS.

• **Navegação por teclado**: usuários com limitações motoras frequentemente utilizam apenas o teclado para navegar, dependendo de indicadores de foco visíveis e de uma ordem lógica de tabulação.

• **Ampliadores de tela**: permitem que usuários com baixa visão aumentem o conteúdo para melhor visualização.

2.5 Web content accessibility guidelines (WCAG)

2.5.1 Histórico e evolução

As diretrizes de acessibilidade para conteúdo web (WCAG) são desenvolvidas pelo World Wide Web Consortium (W3C) por meio de seu grupo de trabalho *Web Accessibility Initiative* (WAI). Desde sua primeira versão (WCAG 1.0), publicada em 1999, as diretrizes evoluíram significativamente:

• **WCAG 2.0 (2008)**: estabeleceu os quatro princípios fundamentais e os níveis de conformidade A, AA e AAA.

• **WCAG 2.1 (2018)**: adicionou critérios para dispositivos móveis, baixa visão e deficiências cognitivas.

• **WCAG 2.2 (2023)**: introduziu nove novos critérios, com foco em usabilidade para usuários com deficiências cognitivas e motoras.

2.5.2 Os quatro princípios fundamentais

As WCAG organizam-se em torno de quatro princípios, que estabelecem que o conteúdo web deve ser:

1. Perceptível: a informação e os componentes da interface devem ser apresentados de modo que possam ser percebidos pelos usuários, independentemente de suas capacidades sensoriais.

2. Operável: os componentes da interface e a navegação devem ser operáveis por qualquer usuário, incluindo aqueles que utilizam apenas o teclado ou outras tecnologias assistivas.

3. Compreensível: a informação e a operação da interface devem ser compreensíveis, com textos claros e comportamentos previsíveis.

4. Robusto: o conteúdo deve ser suficientemente robusto para ser interpretado de forma confiável por uma ampla variedade de agentes de usuário, incluindo tecnologias assistivas.

2.5.3 Níveis de conformidade

Cada critério de sucesso das WCAG é associado a um dos três níveis de conformidade:

• **Nível A**: o mais básico, critérios que devem ser atendidos para que alguns grupos de usuários possam acessar o conteúdo.

• **Nível AA**: critérios que removem barreiras significativas de acesso, sendo o nível mais comumente adotado em legislações e políticas públicas.

• **Nível AAA**: o mais avançado, critérios que melhoram significativamente a acessibilidade para usuários com deficiências mais severas.

2.5.4 WCAG 2.2: novos critérios

A versão 2.2 das WCAG, publicada em outubro de 2023, introduz 9 novos critérios de sucesso, com especial atenção para usuários com deficiências cognitivas e motoras. Os novos critérios são:

Nível A:

• **3.2.6 – Ajuda consistente:** mecanismos de ajuda (como *chat*, *FAQ*, contato) devem estar localizados na mesma posição relativa em todas as páginas.

• **3.3.7 – Entrada redundante:** informações previamente fornecidas pelo usuário não devem ser solicitadas novamente no mesmo processo.

Nível AA:

• **2.4.11 – Foco não obscurecido (mínimo):** quando um componente recebe foco, ele não deve ser ocultado por outros conteúdos.

• **2.4.13 – Aparência do foco:** o indicador de foco deve ter contraste mínimo de 3:1 em relação às cores adjacentes e espessura mínima de 2 pixels.

• **2.5.7 – Movimentos de arrastar:** funções que exigem movimento de arrastar devem ter alternativa com interação por clique simples.

• **2.5.8 – Tamanho do alvo (mínimo):** elementos clicáveis devem ter área mínima de 24×24 pixels.

• **3.3.8 – Autenticação acessível:** testes cognitivos (como CAPTCHAs) não devem ser o único método de autenticação.

Nível AAA:

• **2.4.12 – Foco não obscurecido (Enhanced):** versão mais rigorosa do critério 2.4.11 exigindo que o foco esteja totalmente visível.

Dentre esses novos critérios, este trabalho contempla especificamente o critério 2.5.8 (Tamanho do alvo) e o 3.3.8 (Autenticação acessível), por sua relevância e viabilidade de automação, além de outros critérios das versões anteriores da WCAG (1.1.1, 1.3.1, 1.4.3, 1.4.4, 1.4.12, 2.4.3, 2.4.4, 3.1.1, 3.2.5, 3.3.2, 4.1.2).

2.6 Modelo de acessibilidade em governo eletrônico (eMAG)

2.6.1 Origem e objetivos

O Modelo de Acessibilidade em Governo Eletrônico (eMAG) é uma iniciativa do Governo Federal Brasileiro, coordenada pela Secretaria de Governo Digital do Ministério da Economia. O eMAG foi instituído pela Portaria nº 3, de 7 de maio de 2007, e sua versão mais recente, a 3.1, foi publicada em abril de 2014. O modelo estabelece recomendações e requisitos de acessibilidade para sites e portais governamentais, alinhando-se às diretrizes internacionais (WCAG) e considerando as particularidades da legislação e cultura nacionais.

2.6.2 Estrutura do eMAG 3.1

O eMAG 3.1 (Modelo de Acessibilidade em Governo Eletrônico) possui quarenta e cinco recomendações de acessibilidade, organizadas em seis grupos distintos. Diferentemente das diretrizes internacionais WCAG, o eMAG 3.1 não divide suas recomendações por níveis de prioridade (A, AA, AAA), indicando que todas devem ser aplicadas para garantir a conformidade. As 45 recomendações estão distribuídas nos seguintes grupos:

- **Marcação:** 9 recomendações
- **Comportamento:** 7 recomendações
- **Conteúdo/Informação:** 12 recomendações
- **Apresentação/Design:** 4 recomendações
- **Multimídia:** 5 recomendações
- **Formulários:** 8 recomendações

Tabela 2 – Grupos do eMAG 3.1

Grupo	Descrição
Marcação	Focado no código HTML e semântica (ordem lógica, IDs únicos)
Comportamento	Refere-se a scripts, janelas e interações do usuário
Conteúdo/Informação	Trata de como os dados são apresentados e descritos
Apresentação/Design	Envolve cores, contraste e layout
Multimídia	Diretrizes para vídeos, áudios e imagens dinâmicas
Formulário	Acessibilidade em campos de preenchimento e botões

Fonte: Elaborado pelo autor com base em eMAG 3.1 (BRASIL, 2014)

2.6.3 Relação com as WCAG

Embora o eMAG seja inspirado nas WCAG, ele apresenta algumas particularidades:

- **Linguagem mais acessível:** as recomendações são escritas em português claro, visando facilitar a compreensão por desenvolvedores e gestores públicos.

- **Priorização de recomendações:** o eMAG classifica suas recomendações em obrigatórias e recomendadas, facilitando a implementação gradual.

- **Adequação ao contexto brasileiro:** considera legislações nacionais, como a Lei Brasileira de Inclusão (Lei nº 13.146/2015), e as necessidades específicas da população brasileira.

No contexto deste trabalho, o eMAG 3.1 complementa as diretrizes WCAG 2.2, uma vez que 14 dos 15 critérios da WCAG implementados mapeiam-se diretamente para recomendações específicas do modelo brasileiro, possuindo suas respectivas correspondências nas 45 recomendações dentro dos 6 grupos eMAG, conforme demonstrado na Tabela 3.

Tabela 3 – Mapeamento dos critérios WCAG para o eMAG 3.1

Critério WCAG 2.2	Nível	Descrição	Grupo eMAG 3.1	Recomendação eMAG
1.1.1	A	Conteúdo não textual (imagens)	Multimídia	3.1
1.3.1	A	Informações e Relações (*headings*, listas, *landmarks*)	Conteúdo/Informação	3.12
1.3.2	A	Sequência significativa (ordem de leitura)	Conteúdo/Informação	3.12
1.4.3	AA	Contraste (mínimo)	Apresentação/Design	4.1
1.4.4	AA	Redimensionamento do texto (*zoom*)	Comportamento	4.2
1.4.12	AA	Espaçamento do texto	Conteúdo/Informação	3.12
1.4.13	AA	Conteúdo sob foco ou mouse	Apresentação/Design	4.1
2.4.3	A	Ordem do foco (*tabindex*)	Marcação	3.2
2.4.4	A	Propósito do *link* (no contexto)	Marcação	3.2
2.5.8	AA	Tamanho do alvo (mínimo)	Comportamento	2.2
3.1.1	A	Idioma da página	Conteúdo/Informação	3.12
3.2.5	AAA	Mudança sob demanda (nova aba)	Marcação	3.2
3.3.2	A	Rótulos ou instruções (formulários)	Formulário	3.6
3.3.8	AA	Autenticação acessível (CAPTCHA)	Formulário	6.6
4.1.2	A	Nome, função, valor (*ARIA*)	Marcação	3.2

Fonte: Elaborado pelo autor (2026)

O critério 1.4.13 (Conteúdo sob foco ou mouse), introduzido na WCAG 2.2 (2023), não possui correspondência direta no eMAG 3.1 (2014), sendo classificado no grupo Apresentação/Design por sua relação com contraste. A integração entre os dois

referenciais permite que o modelo computacional proposto ofereça ao usuário a flexibilidade de escolher entre a auditoria baseada nas diretrizes internacionais (WCAG) ou nas recomendações específicas para o contexto brasileiro (eMAG). Quando o referencial eMAG é selecionado, as recomendações são adaptadas para apresentar a numeração da recomendação correspondente e o grupo ao qual pertence, facilitando a interpretação e a priorização das correções por parte das equipes de desenvolvimento.

2.7 Fundamentos de Inteligência Artificial

Segundo (Russell; Norvig, 2021), a inteligência artificial (IA) é um campo da ciência da computação que desenvolve sistemas capazes de executar tarefas que, tradicionalmente, exigiriam inteligência humana, como reconhecimento de padrões, compreensão de linguagem, tomada de decisão e geração de conteúdo. No contexto da acessibilidade web, a IA tem sido empregada para ampliar as capacidades de ferramentas baseadas exclusivamente em regras estáticas, especialmente em tarefas que envolvem interpretação semântica de conteúdo não textual.

2.7.1 Modelos de Linguagem e Modelos Multimodais

Os modelos de linguagem são sistemas de IA treinados para compreender, processar e gerar texto em linguagem natural. Eles aprendem padrões linguísticos a partir de grandes volumes de dados textuais, sendo capazes de realizar tarefas como classificação, sumarização, tradução e geração de texto. Modelos como o BART (*Bidirectional and Auto-Regressive Transformer*) são exemplos de modelos de linguagem que combinam características de codificadores e decodificadores, permitindo tanto a compreensão quanto a geração de texto. O BART é empregado neste trabalho em um paradigma denominado *zero-shot classification*, no qual o modelo classifica textos em categorias predefinidas sem ter sido explicitamente treinado para aquelas categorias específicas, utilizando apenas a descrição textual das classes fornecidas no momento da inferência.

Os modelos multimodais, por sua vez, são capazes de processar e integrar informações de diferentes modalidades, como texto e imagem. O BLIP (*Bootstrapping Language-Image Pre-training*) é um modelo multimodal que combina um codificador visual do tipo *Vision Transformer* (ViT), que extrai características de imagens, com um decodificador de linguagem, que gera descrições textuais a partir dessas características.

Essa arquitetura permite que o BLIP produza legendas contextuais para imagens, sendo particularmente útil para a geração automática de textos alternativos (atributo *alt*) em páginas *web*.

2.7.2 Geração de Descrições Alternativas e Classificação Contextual

No escopo deste trabalho, os modelos de IA são utilizados para duas tarefas principais. A primeira é a geração de descrições alternativas para imagens que não possuem texto alternativo, realizada pelo BLIP. Essa tarefa consiste em produzir um texto que descreva o conteúdo visual da imagem de forma concisa e semanticamente adequada, permitindo que usuários de leitores de tela compreendam o que está sendo exibido. A segunda tarefa é a classificação contextual de *links*, realizada pelo BART. Nesse caso, o modelo analisa o texto do *link* e o contexto textual ao seu redor (por exemplo, o parágrafo em que está inserido) para classificar o *link* como acessível, genérico, crítico ou publicidade. Essa classificação permite que o sistema identifique *links* que, embora visualmente identificáveis, são semanticamente vagos ou inacessíveis para tecnologias assistivas.

2.7.3 Limitações dos Modelos de IA e Necessidade de Revisão Humana

Apesar dos avanços significativos, os modelos de IA apresentam limitações importantes que devem ser consideradas em aplicações críticas como a acessibilidade. Modelos de linguagem e multimodais podem gerar descrições imprecisas, incompletas ou semanticamente inadequadas, especialmente em contextos específicos ou quando o conteúdo da imagem é complexo e contém textos sobrepostos, como banners promocionais ou infográficos. Além disso, a geração de descrições em inglês (padrão dos modelos) exige mecanismos de tradução para contextos locais, o que pode introduzir novas imprecisões semânticas.

Essas limitações justificam a adoção de um critério de confiança no sistema proposto: quando a confiança da descrição gerada é inferior a 80%, o sistema sinaliza a necessidade de revisão humana, recomendando que a sugestão seja validada ou ajustada manualmente antes da implementação. Essa abordagem reconhece que a IA atua como ferramenta de apoio, reduzindo o retrabalho, mas não substitui o julgamento humano, especialmente em contextos onde a precisão semântica é essencial para a experiência do usuário com deficiência.

2.8 Contexto Legal e Normativo Brasileiro

No Brasil, a acessibilidade digital é garantida por um conjunto de leis e decretos que estabelecem a obrigatoriedade de que sites e serviços digitais sejam acessíveis a todas as pessoas, especialmente aquelas com deficiência. O principal marco legal é a Lei Brasileira de Inclusão (Lei nº 13.146/2015), que em seu artigo 63 determina que "é obrigatória a acessibilidade nos sítios da internet mantidos por empresas com sede ou representação comercial no País ou por órgãos de governo" (BRASIL, 2015), incluindo a disponibilização de conteúdo em formato acessível e a compatibilidade com tecnologias assistivas.

Complementarmente, o Decreto nº 5.296/2004 estabelece diretrizes para a promoção da acessibilidade nos sistemas e meios de comunicação da administração pública federal, reforçando a necessidade de conformidade com as recomendações do Modelo de Acessibilidade em Governo Eletrônico (eMAG). O próprio eMAG, instituído originalmente pela Portaria nº 3/2007 e atualizado para a versão 3.1 em 2014, serve como referência técnica obrigatória para órgãos públicos, organizando suas recomendações nos grupos de Marcação, Comportamento, Conteúdo/Informação, Apresentação/Design, Multimídia e Formulários. A página oficial do eMAG apresenta o modelo completo, seus objetivos, as diretrizes e orientações para implementação, reafirmando o compromisso do governo federal com a inclusão digital.

Apesar do arcabouço legal e normativo, a efetiva implementação da acessibilidade em portais públicos e serviços digitais ainda é um desafio no Brasil. A ausência de mecanismos automatizados de verificação e a falta de capacitação técnica contribuem para que muitos sites mantenham barreiras de acesso, comprometendo o direito à informação e à participação cidadã das pessoas com deficiência. Nesse contexto, iniciativas como a proposta neste trabalho, que integram inteligência artificial e auditoria automatizada aos referenciais eMAG e WCAG, contribuem para a efetivação desse direito, reduzindo a distância entre a legislação e a prática.

2.9 Considerações sobre o capítulo

Este capítulo apresentou os fundamentos conceituais que norteiam o desenvolvimento deste trabalho. A revisão da literatura permitiu identificar seis trabalhos correlatos que

fornecem base teórica para a pesquisa, abrangendo desde estudos empíricos sobre acessibilidade em sites brasileiros até aplicações de inteligência artificial na geração de descrições alternativas e recomendações contextuais. As diretrizes WCAG 2.2 foram detalhadas, com ênfase nos novos critérios que foram implementados no modelo computacional proposto, destacando-se os critérios 2.5.8 (Tamanho do Alvo) e 3.3.8 (Autenticação Acessível).

Por fim, o eMAG 3.1 foi apresentado como referência complementar, especialmente relevante para o contexto brasileiro, sendo estabelecido o mapeamento entre os 15 critérios WCAG implementados e as recomendações correspondentes do modelo brasileiro, distribuídas nos seis grupos: Marcação, Comportamento, Conteúdo/Informação, Apresentação/Design, Multimídia e Formulário.

A fundamentação teórica aqui estabelecida servirá de base para a definição da metodologia de desenvolvimento do sistema, bem como para a seleção dos critérios de acessibilidade a serem verificados e das técnicas de inteligência artificial a serem empregadas. A integração entre os dois referenciais (WCAG e eMAG) possibilita que o modelo computacional ofereça ao usuário a flexibilidade de escolher o referencial mais adequado ao seu contexto, adaptando automaticamente as recomendações e a classificação de severidade dos problemas conforme a opção selecionada.

3. METODOLOGIA

3.1 Classificação da pesquisa

Do ponto de vista metodológico, esta pesquisa classifica-se como aplicada, pois visa solucionar um problema prático relacionado à auditoria de acessibilidade *web*. Quanto à natureza, adota uma abordagem quali-quantitativa, combinando métricas objetivas (quantidade de problemas, taxas de detecção) com análises qualitativas (coerência das sugestões geradas pela IA, classificação de severidade). Em relação aos objetivos, caracteriza-se como exploratória e descritiva, uma vez que envolve o desenvolvimento de um modelo computacional funcional e a análise de seu comportamento em ambientes *web* reais.

3.2 Metodologia de desenvolvimento (PDS)

A construção do sistema seguiu as seguintes fases da Metodologia PDS (Projeto e Desenvolvimento de Sistemas), uma abordagem estruturada amplamente utilizada na engenharia de software para organizar o ciclo de vida de desenvolvimento de artefatos computacionais.

Na Fase de Concepção (Levantamento de Requisitos), foram definidos 15 critérios de acessibilidade com base nas diretrizes e nos pilares principais da WCAG 2.2, bem como nos 6 grupos do eMAG 3.1. Na fase de Elaboração (Projeto), foi estabelecida uma arquitetura modular composta por 15 módulos funcionais. Na fase de Construção (Implementação), o sistema foi codificado em Python na versão 3.12, integrando os modelos de IA BLIP e BART, além de renderização dinâmica via Selenium. Por fim, na fase de Transição/Qualidade (Testes), o sistema foi avaliado experimentalmente em 15 sites de diferentes categorias, comparando os resultados com as ferramentas consolidadas do eMAG ASES e WAVE.

3.3 Arquitetura geral do sistema

O modelo computacional proposto adota uma arquitetura híbrida que combina verificação baseada em regras com análise contextual via inteligência artificial. O fluxo completo do sistema foi organizado em cinco etapas:

1. Entrada: o usuário insere a URL e seleciona o referencial (WCAG 2.2 ou eMAG 3.1)
2. Carregamento dos modelos de IA: o sistema carrega os modelos BLIP e BART na GPU
3. Renderização dinâmica: o *Selenium WebDriver* renderiza a página, executa scrolls automáticos para carregar conteúdo *lazy* e aplica *zoom* de 200%
4. Verificação integrada: as três camadas de verificação operam de forma integrada (regras, dinâmica e IA híbrida)
5. Geração de relatórios: o sistema gera relatórios PDF, PNG e TXT apresentando linhas de código corrigido e pontuação final baseada em função sigmoide

A implementação detalhada dessa arquitetura, incluindo a organização dos 15 módulos e suas interações, é apresentada no Capítulo 4 (Desenvolvimento do Sistema).

3.4 Implementação dos critérios de acessibilidade

A seleção dos 15 critérios de acessibilidade implementados neste trabalho baseou-se em três critérios de priorização.

Primeiro, a viabilidade de automação total, ou seja, critérios cuja verificação pode ser realizada por software sem necessidade de julgamento humano subjetivo.

Segundo, o impacto direto na experiência do usuário com deficiência, priorizando barreiras que mais frequentemente impedem ou dificultam o acesso, como contraste insuficiente (1.4.3), *links* genéricos (2.4.4) e imagens sem descrição (1.1.1).

Terceiro, a cobertura equilibrada dos seis grupos do eMAG 3.1 (Marcação, Comportamento, Conteúdo/Informação, Apresentação/Design, Multimídia e Formulários), garantindo que nenhuma categoria fosse deixada de fora mesmo com o escopo reduzido de 15 critérios em relação ao total de 50 da WCAG 2.2.

É importante destacar que nem todos os critérios utilizam inteligência artificial. Os modelos BLIP e BART são empregados exclusivamente para os critérios 1.1.1 (descrição de imagens) e 2.4.4 (classificação de *links*). Os demais 13 critérios utilizam verificação por regras (análise estática do HTML/CSS) ou verificação dinâmica (renderização via Selenium), conforme a natureza de cada critério.

Os critérios selecionados abrangem os quatro princípios da WCAG 2.2 (Perceptível, Operável, Compreensível, Robusto) e os seis grupos do eMAG 3.1. A seguir, são apresentados os critérios e seus respectivos métodos de verificação:

• **Critério 1.1.1** (*Non-text content*): Refere-se às imagens sem texto alternativo. Sua verificação combina inteligência artificial e regras, utilizando o modelo BLIP para gerar sugestões de texto alternativo e validando a presença do atributo *alt*.

• **Critério 1.3.1** (*Info and relationships*): Abrange *headings*, listas e *landmarks*. A verificação é feita por regras, analisando a hierarquia dos cabeçalhos, a semântica das listas e a presença de elementos como *main*, *nav*, *header* e *footer*.

• **Critério 1.3.2** (*Meaningful sequence*): Trata da ordem de leitura da página analisada. A verificação, sendo baseada em regras, analisa se o elemento *footer* não aparece antes do conteúdo principal representado pela tag *main*.

• **Critério 1.4.3** (*Contrast minimum*): Estabelece níveis mínimos de contraste entre texto e fundo. Sua verificação é dinâmica, utilizando Selenium para obter as cores computadas e aplicando a fórmula WCAG de luminância relativa.

• **Critério 1.4.4** (*Resize text*): Exige que o texto possa ser redimensionado em até 200% sem perda de funcionalidade. A verificação é dinâmica, aplicando *zoom* de 200% via JavaScript e detectando sobreposições entre elementos.

• **Critério 1.4.12** (*Text spacing*): Trata do espaçamento de texto. A verificação é feita por regras, identificando o uso de unidades absolutas como px e pt em propriedades CSS como *line height*, *letter spacing*, *word-spacing* e *text indent*.

• **Critério 2.4.3** (*Focus order*): Estabelece que a ordem de navegação por teclado deve ser lógica. A verificação, baseada em regras, identifica valores de *tabindex* maiores que zero, que podem causar confusão na navegação.

• **Critério 2.4.4** (*Link purpose*): Exige que o propósito de cada link seja compreensível. A verificação combina inteligência artificial e regras, utilizando o modelo BART para classificação contextual e validando a presença de *aria-label*.

• **Critério 2.5.8** (*Target size*): Estabelece que elementos clicáveis devem ter área mínima de 24×24 pixels. A verificação é dinâmica, calculando a área clicável efetiva incluindo o *padding* dos elementos via Selenium.

• **Critério 3.1.1** (*Language of page*): Exige que o idioma da página seja definido. A verificação, baseada em regras, valida a presença do atributo *lang* na tag <*html*>.

• **Critério 3.2.5** (*Change on request*): Trata de *links* que abrem em nova aba. A verificação é feita por regras, detectando *links* com *target="_blank"* que não possuem *aria-label* descritivo.

• **Critério 3.3.2** (*Labels or instructions*): Exige que campos de formulário tenham rótulos adequados. A verificação, baseada em regras, analisa a associação entre as *tags label* e *input*, bem como a presença de *aria-label*.

• **Critério 3.3.8** (*Accessible authentication*): Trata da acessibilidade em processos de autenticação, como CAPTCHAs. A verificação é feita por regras, detectando padrões como reCAPTCHA e hCaptcha e verificando a existência de alternativas acessíveis como e-mail, telefone ou formulário de contato.

• **Critério 4.1.2** (*Name*, *role*, *value*): Estabelece que elementos interativos devem ter nome e função definidos para tecnologias assistivas. A verificação, baseada em regras, valida atributos ARIA como *aria-label*, *aria-labelledby*, *role*, *aria-hidden* e *aria-expanded*.

3.5 Seleção dos objetos de testes

Foram selecionados 15 sites de diferentes categorias para avaliação experimental do

sistema, buscando representar a diversidade de estruturas e conteúdos presentes na web. Os critérios de seleção incluíram: (i) sites com alto tráfego no Brasil e relevância em suas respectivas categorias; (ii) diversidade de implementação técnica (desde portais estáticos em WordPress até SPAs complexas); (iii) cobertura equilibrada dos referenciais eMAG (para sites governamentais e institucionais) e WCAG (para os demais); e (iv) disponibilidade pública e estabilidade durante o período de coleta. A Tabela 4 apresenta os sites selecionados, suas respectivas categorias e os referenciais utilizados em cada auditoria.

Tabela 4 – Sites selecionados para validação e suas respectivas categorias.

Nº	Site	URL	Referencial	Categoria
1	Prefeitura de Catu	https://www.catu.ba.gov.br/	eMAG	Governamental
2	Prefeitura de Pojuca	https://www.pojuca.ba.gov.br/	eMAG	Governamental
3	SUAP	https://suap.ifbaiano.edu.br/	eMAG	Institucional
4	UFBA	https://ufba.br/	eMAG	Institucional
5	Shopee	https://www.shopee.com.br	WCAG	E-commerce
6	Mercado Livre	https://www.mercadolivre.com.br/	WCAG	E-commerce
7	Amazon	https://www.amazon.com.br	WCAG	E-commerce
8	Google Play	https://play.google.com	WCAG	Apps/Plataformas
9	Duel Links Meta	https://www.duellinksmeta.com/	WCAG	Gaming
10	Wikipedia	https://www.wikipedia.org	WCAG	Educação
11	Portal GOV.BR	https://www.gov.br/pt-br	eMAG	Governamental
12	Magazine Luiza	https://www.magazineluiza.com.br	WCAG	E-commerce
13	G1	https://g1.globo.com/	WCAG	Notícias
14	Cicatribio	https://cicatribio.com.br/	WCAG	Apps/Plataformas
15	BA.GOV	https://www.ba.gov.br/	eMAG	Governamental

Fonte: Elaborado pelo autor (2026).

3.6 Tecnologias e ferramentas utilizadas

3.6.1 Ambiente de desenvolvimento

O sistema foi desenvolvido em Python 3.12, utilizando o ambiente Google Colab para execução dos experimentos. A escolha do Colab justifica-se pela disponibilidade gratuita de GPU, necessária para o carregamento e execução eficiente dos modelos de inteligência artificial (BLIP e BART). Foram utilizadas instâncias com GPU T4 (16 GB de VRAM), com tempo de execução médio de 3 a 5 minutos por página auditada.

3.6.2 Ferramentas de comparação nos testes

Para avaliar o desempenho do modelo proposto, os resultados obtidos foram comparados com duas ferramentas consolidadas na auditoria de acessibilidade *web*: o Avaliador e Simulador de Acessibilidade em Sítios (eMAG ASES), ferramenta oficial do Governo Brasileiro para validação do eMAG, e o WAVE (*Web Accessibility Evaluation Tool*), amplamente utilizado para avaliação com base nas diretrizes WCAG. A escolha dessas ferramentas justifica-se por sua ampla adoção e relevância no contexto nacional e internacional.

3.7 Procedimento de coleta

O processo de coleta de dados para avaliação experimental seguiu as seguintes etapas:

Etapa 1 – Auditoria com o modelo proposto: Execução do sistema para cada um dos 15 sites selecionados, com registro da pontuação, total de problemas, distribuição por severidade e geração dos relatórios para documentação.

Etapa 2 – Auditoria com ferramentas de comparação: Submissão de cada site ao eMAG ASES (para referencial eMAG) e ao WAVE (para referencial WCAG), com extração dos resultados conforme formato próprio de cada ferramenta.

Etapa 3 – Análise comparativa: Comparação dos resultados por critério e categoria, identificação de discrepâncias e avaliação do desempenho relativo de cada ferramenta em termos de quantidade de problemas detectados.

Etapa 4 – Documentação: Organização dos resultados em tabelas comparativas, análise qualitativa das diferenças encontradas, e sistematização das conclusões.

Protocolo de coleta: As auditorias foram realizadas no período de 08 de março a 21 de abril de 2026, entre 10h e 16h (horário de Brasília) em dias úteis, para minimizar variações de conteúdo dinâmico. Foram utilizadas as seguintes configurações: navegador Chrome versão 124 (modo *headless*), Selenium *WebDriver* 4.15, Python 3.12, Google Colab com GPU T4.

Cada site foi auditado três vezes em dias distintos para verificar a consistência dos resultados; as médias foram reportadas. Uma amostra de 20% dos alertas gerados pelo sistema foi inspecionada manualmente para identificação de falsos positivos e falsos negativos. O "total esperado" de problemas para a taxa de detecção foi definido pela união dos problemas identificados pelo sistema proposto e pelas ferramentas tradicionais, validada manualmente nas amostras selecionadas, servindo como referência para o cálculo da taxa de detecção.

3.8 Métricas de avaliação

Foram definidas as seguintes métricas para avaliação dos resultados:

• **Taxa de detecção**: percentual de problemas detectados em relação ao total esperado (definido conforme descrito no procedimento de coleta)
• **Densidade de problemas**: quantidade de problemas por elemento analisado
• **Tempo de execução**: tempo total de processamento por página (em segundos)
• **Avaliação qualitativa da IA**: percentual de sugestões consideradas semanticamente adequadas após análise manual do autor, considerando coerência textual e aderência às recomendações de acessibilidade.

3.9 Mecanismo de pontuação

O sistema adota um modelo de pontuação híbrido normalizado que considera múltiplos fatores para calcular a nota final de acessibilidade de 0 a 100. A pontuação é calculada por meio de uma função sigmoide, que garante uma curva suave e evita extremos, com centro ajustado para penalidade média de 75. Os valores foram calibrados empiricamente a partir de testes com 15 sites de diferentes categorias, ajustando os parâmetros até que a pontuação refletisse consistentemente a gravidade percebida dos problemas.

Fórmulas:

$$\text{penalidade\_total} = \Sigma(\text{peso} \times \text{fator\_volume} \times \text{fator\_densidade})$$

$$\text{penalidade\_total} = \text{penalidade\_total} \times \text{fator\_cobertura}$$

$$\text{penalidade\_media} = \text{penalidade\_total} / \sqrt{(\text{num\_tipos})}$$

$$\text{pontuacao} = 100 / (1 + e^{((\text{penalidade\_media} - 75) / 30)})$$

Fatores de cálculo:

• **Severidade:** atribui peso conforme o impacto do problema, sendo Alta = 12, Média = 5 e Baixa = 1,5. Os pesos seguem uma proporção geométrica (8:3:1) para amplificar diferenças entre níveis críticos e baixos, alinhada à literatura de qualidade de software.

• **Volume:** ajusta a penalidade conforme a quantidade de ocorrências do mesmo problema, calculado por fator_volume = min(3,5, 1 + log(volume) / 6). O uso do logaritmo evita que problemas muito frequentes dominem a pontuação, suavizando o crescimento.

• **Densidade:** considera a proporção de problemas em relação ao total de elementos analisados, com fator_densidade = 1 + (volume / (total_elementos + 20)) × 1,2. Penaliza a concentração de problemas em poucos elementos.

• **Cobertura:** penaliza a diversidade de critérios afetados por meio de crescimento logarítmico, com fator_cobertura = 1 + log(critérios_afetados) × 0,1. A pontuação final é então classificada em cinco níveis: Excelente (≥85), Bom (≥65), Moderado (≥45), Ruim (≥25) ou Crítico (<25). Um site com muitos tipos diferentes de problemas é mais

penalizado que um site com poucos tipos, mesmo com volume similar.

A Tabela 5 apresenta a classificação do sistema para cada nível de acessibilidade.

T*a*bela 5 – Classificação do nível de acessibilidade

Pontuação	Nível
≥ 85	Excelente
≥ 65	Bom
≥ 45	Moderado
≥ 25	Ruim
< 25	Crítico

Fonte: Elaborado pelo autor (2026)

3.10 Critério de confiança da IA

O sistema utiliza a probabilidade de saída da geração do BLIP como métrica de confiança. Quando a confiança da descrição gerada é igual ou superior a 80%, a sugestão é aprovada e entregue no código corrigido. Quando a confiança é inferior a 80%, o sistema ainda gera a sugestão, mas a sinaliza explicitamente no relatório com a frase "revisão humana recomendada". Este limite foi definido empiricamente após testes com 200 imagens, que mostraram que descrições abaixo deste patamar continham termos genéricos como "a man" ou "a woman" ou erros semânticos como "tapete listrado" para uma logomarca institucional.

3.11 Considerações éticas

A pesquisa utilizou exclusivamente sites públicos e ferramentas de código aberto ou gratuitas, respeitando as políticas de uso dos serviços. Os dados coletados referem-se apenas a aspectos técnicos de acessibilidade, sem qualquer coleta de informações pessoais dos usuários. As auditorias realizadas não modificaram ou interferiram no

funcionamento dos sites analisados. Por não envolver experimentos com seres humanos, este trabalho é isento de análise pelo Comitê de Ética em Pesquisa (CEP).

3.12 Limitações de pesquisa

As principais limitações desta pesquisa incluem:

• **Escopo reduzido de critérios**: Foram implementados 15 dos 50 critérios WCAG 2.2. A seleção priorizou os critérios de maior viabilidade de automação e impacto na experiência do usuário.

• **Overhead de processamento**: A análise de contraste e *zoom* requer Selenium, o que introduz um overhead de aproximadamente 3 a 5 minutos por página.

• **Limitação da IA**: As sugestões de textos alternativos gerados pelo BLIP podem não ser semanticamente perfeitas para todos os contextos.

• **Amostra limitada**: A validação foi realizada em 15 sites, o que limita a generalização dos resultados.

• **Ferramentas de comparação**: O eMAG ASES e WAVE possuem metodologias próprias que nem sempre são diretamente comparáveis. O ASES realiza análise estática, falhando em capturar conteúdos carregados via *lazy loading*. O WAVE, embora processe o DOM renderizado, é bem focado na inspeção visual pontual, não permitindo automação nativa gratuita em larga escala.

• **Avaliação qualitativa da IA:** A avaliação das sugestões geradas pela IA foi realizada manualmente pelo autor, o que pode introduzir viés. Recomenda-se que estudos futuros utilizem múltiplos avaliadores independentes.

• **Limitação de infraestrutura computacional:** O sistema foi desenvolvido e executado em um ambiente de uso pessoal, com recursos computacionais limitados (notebook com 8 GB de RAM e processador Intel Core i5). Embora os experimentos tenham sido realizados no Google Colab com GPU T4 para a execução dos modelos de IA, a etapa de desenvolvimento e os testes iniciais foram conduzidos localmente, o que impôs restrições

ao número de iterações e à complexidade das simulações realizadas. Essa limitação pode ter influenciado o tempo de desenvolvimento e a amplitude dos testes exploratórios, sugerindo que, em infraestruturas mais robustas, o sistema poderia ser otimizado para maior eficiência e escala.

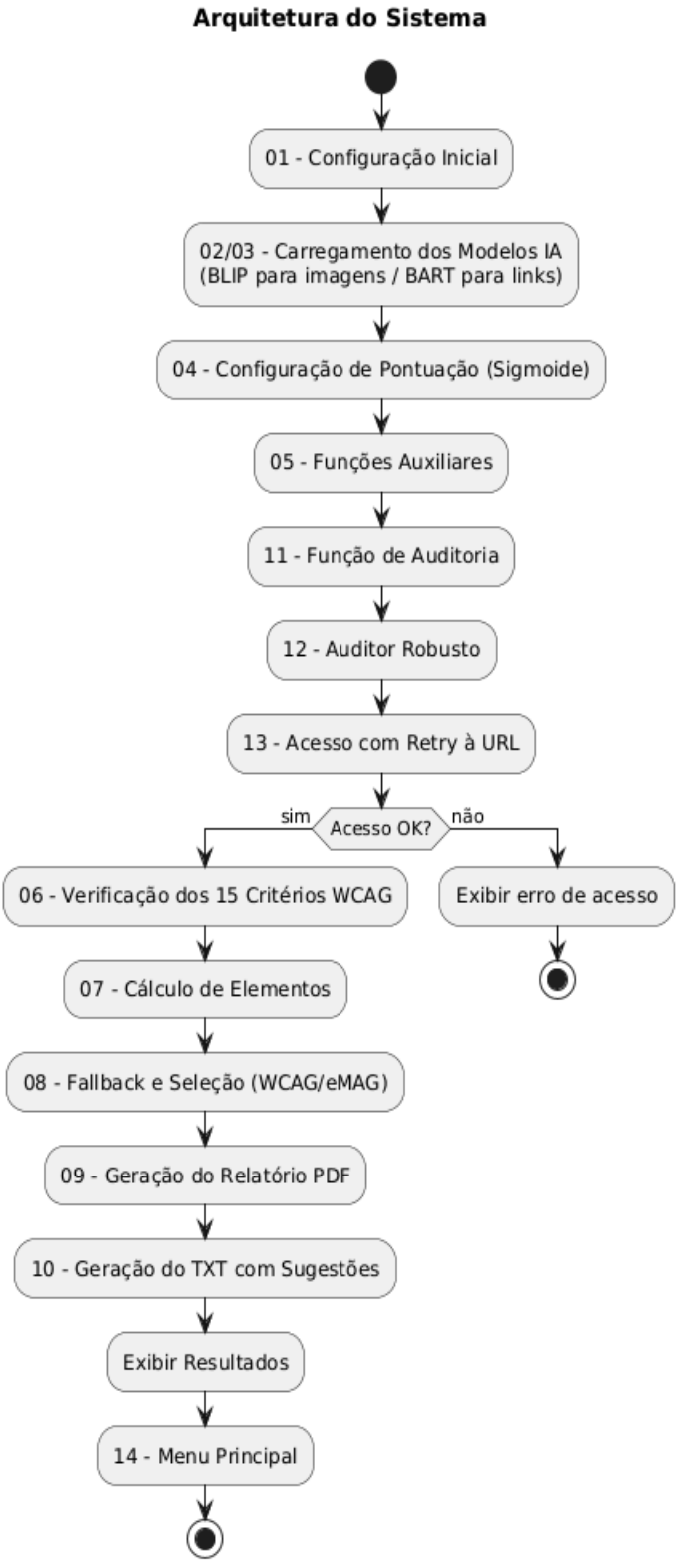
4. DESENVOLVIMENTO DO SISTEMA

Este capítulo descreve a implementação técnica do modelo computacional proposto, abordando a arquitetura do sistema, as tecnologias utilizadas, a implementação dos modelos de inteligência artificial e os módulos de verificação de acessibilidade.

4.1 Visão geral da arquitetura

O sistema foi desenvolvido em Python 3.12 e adota uma arquitetura modular, organizada em 15 módulos independentes que se comunicam entre si. A Figura 1 apresenta o diagrama de blocos da arquitetura.

Figura 1 – Arquitetura modular do sistema



Fonte: Produzido pelo Autor utilizando a plataforma Plantuml.com (2026)

Conforme ilustrado na Figura 1, os módulos foram agrupados por afinidade funcional em quatro blocos principais.

O primeiro bloco, composto pelos módulos 01 a 04, é responsável pela configuração do ambiente e carregamento dos modelos de inteligência artificial. O módulo 01 realiza a configuração inicial do ambiente Google Colab, a detecção automática de dependências e a instalação de bibliotecas necessárias. O módulo 02 implementa o `GerenciadorIA`, responsável pelo *lazy loading* dos modelos BLIP (para descrição de imagens) e BART (para classificação contextual de *links*), além de definir os mapeamentos entre os referenciais WCAG e eMAG e os pesos de severidade. O módulo 03 contém as funções baseadas em modelos de IA propriamente ditas, incluindo a geração de descrições de imagens, a classificação de *links* e o mecanismo de tradução em cascata (online via *deep-translator*, *offline* via OPUS-MT e *fallback* manual). O módulo 04 implementa o mecanismo de pontuação sigmoide, que calcula a nota final de acessibilidade com base na severidade, volume, densidade e cobertura dos problemas identificados.

O segundo bloco, composto pelos módulos 05 e 06, concentra as funções de verificação dos 15 critérios de acessibilidade. O módulo 05 reúne funções auxiliares para cálculo de contraste, normalização de cores, extração de redes sociais e cache (possui relação direta com o módulo 7). O módulo 06 é o mais extenso do sistema, contendo aproximadamente 2.000 linhas de código distribuídas em sete partes, cada uma responsável por um conjunto específico de critérios, abrangendo imagens (1.1.1), *links* (2.4.4), formulários (3.3.2), tamanho de alvos (2.5.8), CAPTCHA (3.3.8), contraste (1.4.3), *zoom* (1.4.4), espaçamento (1.4.12), *headings* (1.3.1), ordem de leitura (1.3.2), ordem de foco (2.4.3), idioma (3.1.1), nova aba (3.2.5) e *ARIA* (4.1.2).

O terceiro bloco, composto pelos módulos 08 a 11, é responsável pela geração de relatórios e pela função principal de auditoria. O módulo 08 contém funções para auditoria sem Selenium (*fallback*) e para seleção do referencial. O módulo 09 implementa a geração do relatório PDF utilizando a biblioteca *ReportLab*, com capa, gráficos, tabelas e recomendações. O módulo 10 implementa a geração do arquivo TXT com código corrigido, incluindo a origem da sugestão (IA, contexto, nome do arquivo, cache) e o score de qualidade. O módulo 11 contém a função principal `auditar_site_com_ia()`, que orquestra todas as verificações, gerencia o *driver Selenium* e coordena a chamada dos demais módulos.

O quarto bloco, composto pelos módulos 12 a 15, reúne funcionalidades complementares de suporte. O módulo 12 implementa a classe `Auditor_Robusto`, que oferece *fallback* quando o Selenium ou a IA não estão disponíveis. O módulo 13 implementa funções para acesso a sites com *retry*, rotação de *User-Agent* e cache de respostas. O módulo 14 implementa o menu interativo no console, com opções para auditar site, exibir informações sobre o sistema e sair. O módulo 15 contém a execução principal, com verificação de componentes e tratamento de exceções globais.

Essa organização modular permite que cada módulo seja desenvolvido, testado e mantido de forma independente, facilitando a evolução do sistema e a correção de problemas. A comunicação entre os módulos ocorre por meio de funções públicas bem definidas, que recebem parâmetros como o *soup* do *BeautifulSoup*, o driver do Selenium, *o html_source* para extração de linhas e o `url_base` para normalização de URLs relativas.

4.2 Tecnologias e ferramentas utilizadas

Tabela 6 – Bibliotecas instaladas

Biblioteca	Finalidade
requests / beautifulsoup4	Aquisição e parsing de páginas HTML
selenium	Renderização dinâmica (contraste, *zoom*)
transformers (Hugging Face)	Carregamento dos modelos BLIP e BART
torch	Suporte a GPU para os modelos de IA
matplotlib	Geração do gráfico PNG
reportlab	Geração do relatório PDF
pandas	Manipulação dos dados de problemas

Fonte: Produzido pelo Autor (2026)

O ambiente de desenvolvimento utilizado foi o Google Colab, que oferece suporte gratuito a GPU NVIDIA Tesla T4, essencial para o carregamento e execução eficiente dos modelos de inteligência artificial.

O modelo BLIP (*Bootstrapping Language-Image Pre-training*) foi escolhido para gerar sugestões automáticas de texto alternativo para imagens sem atributo alt por ser um modelo generativo que produz descrições contextuais, diferentemente de alternativas como CLIP (que apenas classifica imagens) e GPT-4V (proprietário e pago). Comparado ao BLIP-2, mais pesado (1,2B parâmetros), o BLIP base (232M parâmetros) foi priorizado por equilibrar qualidade de descrição e eficiência computacional, sendo suficiente para os tipos de imagem analisados (logomarcas, banners, fotografias institucionais).

O modelo combina um codificador visual (Vision Transformer) com um decodificador de linguagem, permitindo a geração de legendas contextuais.

O modelo *Salesforce/blip-image-captioning-base* é o principal e único utilizado, com fallback heurístico para casos de falha.

4.3 Limitações para evitar sobrecarga

- Redimensionamento de imagens para no máximo 512×512 pixels
- Validação das descrições geradas (rejeição de termos genéricos como "a man", "a woman", "a person", "image", "photo")
- Timeout de 10 segundos por requisição
- Cache em memória para evitar reprocessamento da mesma imagem
- Limite máximo de 100 imagens processadas por auditoria (`MAX_IMAGENS_IA`)

Tratamento especial para SVG:

- Arquivos SVG são convertidos para PNG utilizando a biblioteca CairoSVG antes de serem processados pelo BLIP.

4.4 BART (Classificação de links)

O modelo BART (*Bidirectional and Auto-Regressive Transformer*) é utilizado para classificação contextual de *links*, empregando a técnica de *zero-shot classification*. Foi escolhido para classificação contextual de *links* por ser especificamente *fine-tuned* para inferência natural (NLI), tornando-o ideal para *zero-shot classification*. Alternativas como

BERT e RoBERTa não são otimizadas para NLI, enquanto GPT-2 é generativo, não classificador. O modelo analisa o texto do link acompanhado de um contexto textual ao redor e classifica em uma das seguintes categorias:

1. Acessível: *links* com texto descritivo que indicam claramente o destino ou ação (ex: "Fale Conosco", "Acessar Portal da Transparência").

2. Genérico: *links* com textos vagos e pouco informativos, como "clique aqui", "saiba mais", "leia mais".

3. Crítico: *links* completamente vazios ou que não possuem nenhum texto descritivo, tornando-os inacessíveis para leitores de tela.

4. Publicidade: *links* identificados como rastreamento, patrocínio ou publicidade (ex: URLs contendo "Google-Analytics", "doubleclick", "facebook.com/tr").

Caso o modelo BART não esteja disponível ou a classificação falhe, o sistema adota um *fallback* heurístico baseado em regras (verificação de palavras-chave na URL e no texto do link).

4.5 Limitações implementadas

• Limite máximo de 50 chamadas à IA por auditoria (`IA_MAX_CALLS`)
• Cache de resultados para evitar reprocessamento do mesmo *link*
• Contexto limitado aos primeiros 150 caracteres ao redor do *link*

A classificação dos *links* pelo modelo BART é organizada em 4 categorias distintas.

• *Links* classificados como acessível são aqueles que possuem texto descritivo adequado, indicando claramente o destino ou ação, e não geram alerta no sistema.

• *Links* genéricos são caracterizados por textos vagos e pouco informativos, como "clique aqui", "saiba mais" ou "leia mais", recebendo severidade média.

• *Links* críticos correspondem a *links* vazios ou compostos apenas por ícones sem qualquer descrição textual, sendo classificados com severidade alta por representarem barreiras significativas para usuários de leitores de tela.

• *Links* de publicidade ou rastreamento (como aqueles que contêm *doublecklick*, Google *Analytics* ou facebook.com/tr nas URLs) recebem severidade média, alertando sobre práticas que podem comprometer a acessibilidade. Caso o modelo BART não esteja disponível ou a classificação falhe, o sistema adota um *fallback* heurístico baseado em regras, verificando palavras-chave na URL e no texto do *link*.

4.5.1 Implementação dos módulos de verificação

Foram implementados 15 módulos de verificação, cada um correspondendo a um critério de acessibilidade. Os métodos empregados podem ser classificados em três categorias: (i) verificação por regras, baseada em análise estática do HTML e CSS; (ii) verificação dinâmica, que requer renderização da página via Selenium para acesso ao DOM computado; e (iii) verificação híbrida, que combina regras com inteligência artificial.

Os critérios 1.1.1 (Imagens) e 2.4.4 (*Links*) adotam abordagem híbrida. O primeiro utiliza o modelo BLIP para gerar sugestões de texto alternativo, enquanto o segundo emprega o modelo BART para classificação contextual de *links*, complementados por regras de validação sintática.

Critérios 1.4.3 (Contraste), 1.4.4 (*Zoom*) e 2.5.8 (Tamanho de alvos) utilizam verificação dinâmica. O contraste é calculado por meio da fórmula WCAG de luminância relativa. O *zoom* é testado simulando ampliação de 200% via JavaScript. O tamanho dos alvos é avaliado considerando a área clicável efetiva, que deve ter no mínimo 24×24 pixels, incluindo o *padding* dos elementos.

Os demais critérios empregam exclusivamente verificação por regras. O critério 1.3.1 (*Headings*, listas e *landmarks*) valida a hierarquia de *headings* e a semântica estrutural da página. O critério 1.3.2 (Ordem de leitura) verifica a posição relativa entre os elementos *main* e *footer*. O critério 1.4.12 (Espaçamento) detecta o uso de unidades absolutas como px e pt em CSS. O critério 2.4.3 (Ordem de foco) identifica valores de *tabindex* maiores que zero, que podem causar confusão na navegação por teclado.

O critério 3.1.1 (Idioma da página) valida a presença do atributo lang na tag <*html*>. O critério 3.2.5 (Nova aba) detecta *links* com *target="_blank"* que não possuem *aria-label* descritivo. O critério 3.3.2 (Formulários) verifica a correta associação entre rótulos (label) e campos de entrada (*input*). O critério 3.3.8 (CAPTCHA) identifica padrões como reCAPTCHA e hCaptcha e verifica a existência de alternativas acessíveis como e-mail, telefone ou formulário de contato.

Por fim, o critério 4.1.2 (ARIA) válida atributos de acessibilidade como *aria-label*, *aria-labelledby, role*, *aria-hidden e aria-expanded.*

4.5.2 Verificação de contraste (1.4.3)

O cálculo do contraste segue rigorosamente a fórmula estabelecida pela WCAG 2.2. Para cada elemento de texto na página, o sistema obtém as cores de primeiro plano e fundo computadas via Selenium e aplica o seguinte algoritmo:

$$\text{Luminância relativa (L)} = 0{,}2126 \times R + 0{,}7152 \times G + 0{,}0722 \times B$$

$$\text{Relação de contraste} = (L1 + 0{,}05) / (L2 + 0{,}05)$$

Onde L1 é a luminância da cor mais clara e L2 da mais escura. O critério é considerado não conforme quando a relação é inferior a 4,5:1 para texto normal ou 3:1 para texto grande (≥ 18px).

4.5.3 Verificação de zoom (1.4.4)

O sistema utiliza Selenium para aplicar *zoom* de 200% na página e executa um script JavaScript que detecta sobreposições entre elementos. Para cada par de elementos sobrepostos, é registrado um problema, com amostragem limitada para não poluir o relatório.

4.5.4 Verificação de tamanho de alvos (2.5.8)

Elementos clicáveis (<*a*>, <*button*>) têm suas dimensões computadas via CSS. Quando a largura ou altura é inferior a 24 pixels, o sistema verifica se há padding suficiente para compensar; caso contrário, registra um problema.

4.6 Implementação do mecanismo de pontuação

A pontuação final é calculada por um algoritmo híbrido que considera:
1. Severidade do problema (Alta = 12, Média = 5, Baixa = 1,5)

2. Volume de ocorrências (crescimento logarítmico para evitar dominância, com fator min(3,5, 1 + log(volume)/6))

3. Densidade (problemas por elemento, com fator 1 + (volume / (total_elementos + 20)) × 1,2)

4. Cobertura (quantidade de critérios diferentes afetados, com fator 1 + log (critérios_afetados) × 0,1)

5. Penalidade extra para problemas graves (5 pontos adicionais, limitados a 3 ocorrências) A penalidade total é então convertida em pontuação por meio de uma função sigmoide centralizada:

pontuacao = 100 / (1 + e^((penalidade_media - 75) / 30))

O resultado é uma nota entre 0 e 100, classificada em cinco níveis: Excelente (≥85), Bom (≥65), Moderado (≥45), Ruim (≥25) ou Crítico (<25).

4.7 Implementação da geração de relatórios

O sistema gera 3 tipos de relatório: um gráfico PNG para visualização rápida, um relatório PDF completo para documentação e um arquivo TXT contendo o código corrigido.

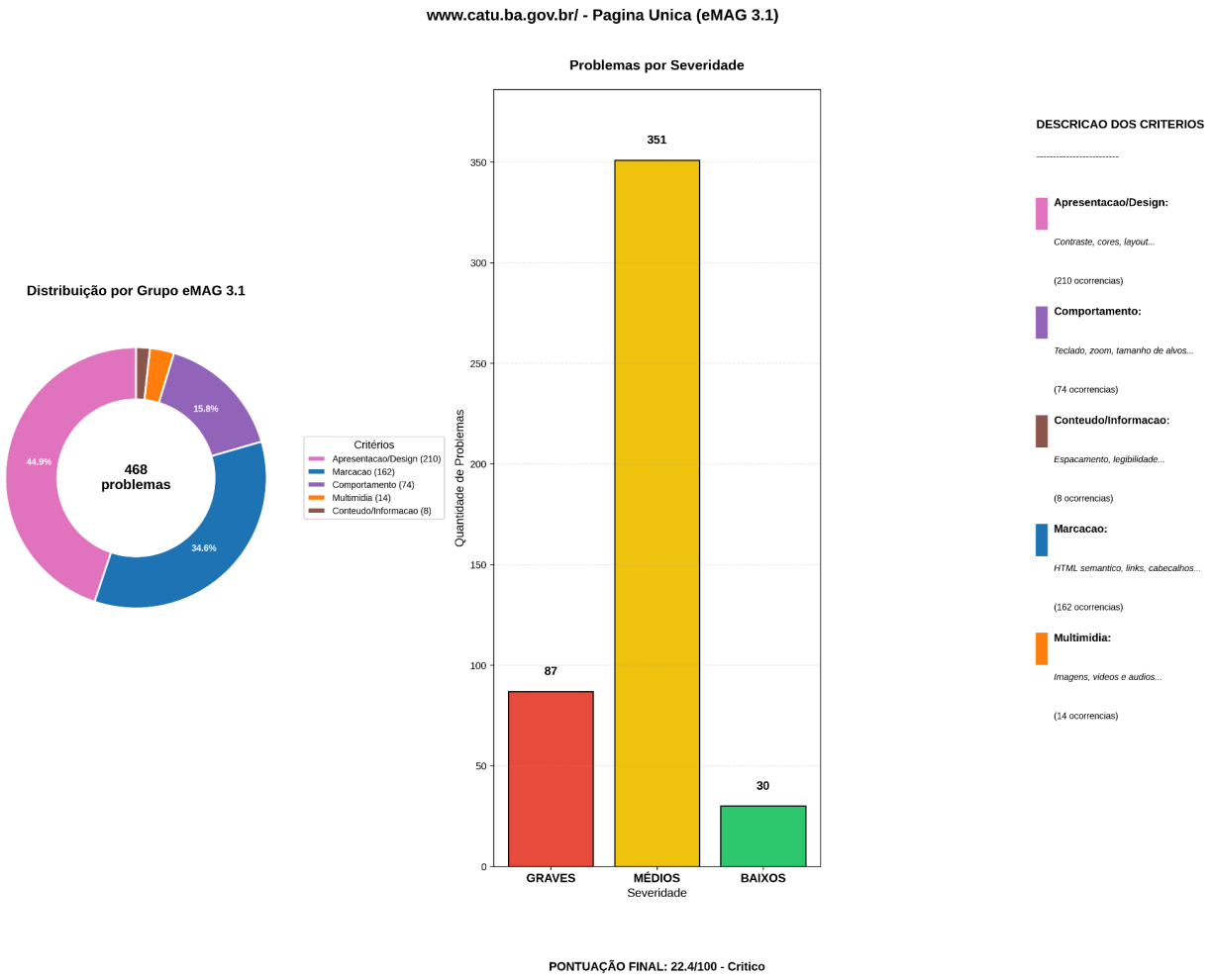
4.7.1 Gráfico PNG (matplotlib)

O gráfico PNG é gerado com a biblioteca *matplotlib* e inclui:

- Gráfico de pizza (distribuição por grupo eMAG ou critério WCAG)
- Gráfico de barras (problemas por severidade)
- Legenda com descrições dos critérios
- Total de problemas e pontuação final

A Figura 2 apresenta uma imagem do gráfico.

Figura 2 – Gráfico PNG do relatório de auditoria do sistema.



Fonte: Produzido pelo autor (2026).

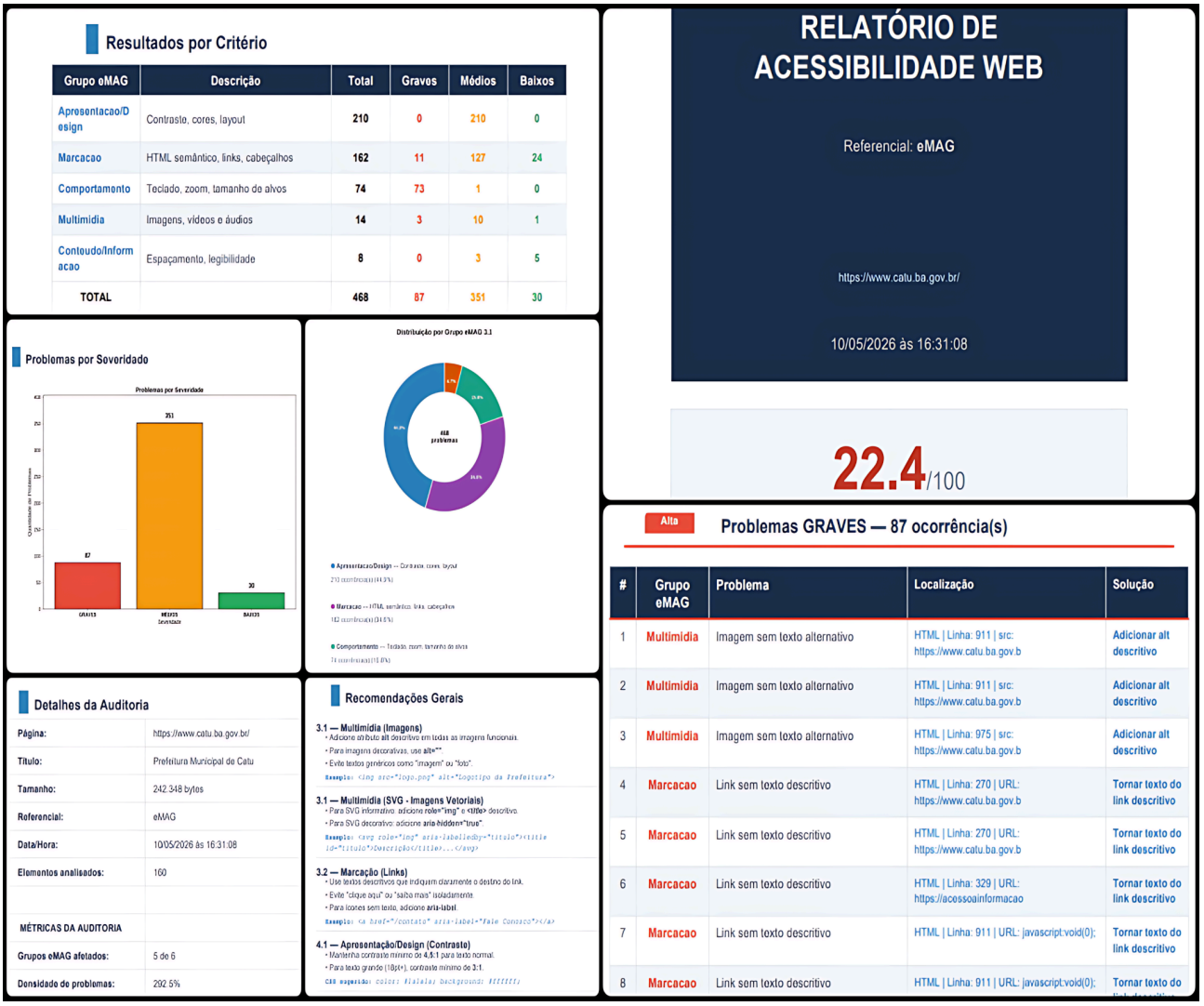
4.7.2 Relatório PDF (reportlab)

O relatório PDF é gerado com a biblioteca *reportlab* e inclui as seguintes seções:

- **Capa:** com pontuação, classificação e cards de resumo
- **Detalhes da auditoria:** elementos analisados, grupos afetados, densidade
- **Gráficos:** pizza e barras (inseridos como imagens)
- **Problemas por severidade:** gráfico de barras
- **Resultados por critério:** tabela com totais e distribuição por severidade
- **Lista detalhada de problemas:** severidade, localização e solução sugerida
- **Recomendações gerais:** orientações práticas com exemplos de código

A Figura 3 apresenta imagens das seções do relatório em PDF.

Figura 3 – Seções do relatório em PDF



Fonte: Produzido pelo autor (2026).

4.7.3 Arquivo TXT com sugestões de código corrigido

O sistema gera um arquivo no formato TXT contendo as recomendações de código detalhadas por localização, organizado com cabeçalho contendo URL auditada, referencial utilizado (WCAG ou eMAG) e data da auditoria. O arquivo apresenta uma seção HTML com lista de ocorrências identificadas, cada uma contendo a linha do código-fonte onde o problema foi encontrado, o critério violado, a descrição do problema, o código original com a inconformidade, o código sugerido para correção, a origem da sugestão (IA, cache ou heurística), o score de qualidade da correção e a recomendação textual baseada no referencial adotado. Há também uma seção CSS para ocorrências em folhas de estilo externas, com formato similar. Ao final, o arquivo exibe um resumo com total de recomendações, estatísticas do sistema (caches, chamadas IA), distribuição por severidade e lista dos critérios mais frequentes.

As correções propostas pelo sistema demonstram, na prática, o valor da abordagem adotada, mas também revelam limitações importantes. Enquanto os códigos originais frequentemente apresentavam atributos *alt* vazios, ausência de aviso para abertura de novas abas, *links* sem texto descritivo ou elementos interativos sem identificação, as sugestões geradas preenchem essas lacunas de forma contextualizada. Nos casos de imagens, o BLIP produziu descrições que representam um avanço significativo em relação ao *alt* vazio, que tornava a imagem completamente invisível para leitores de tela.

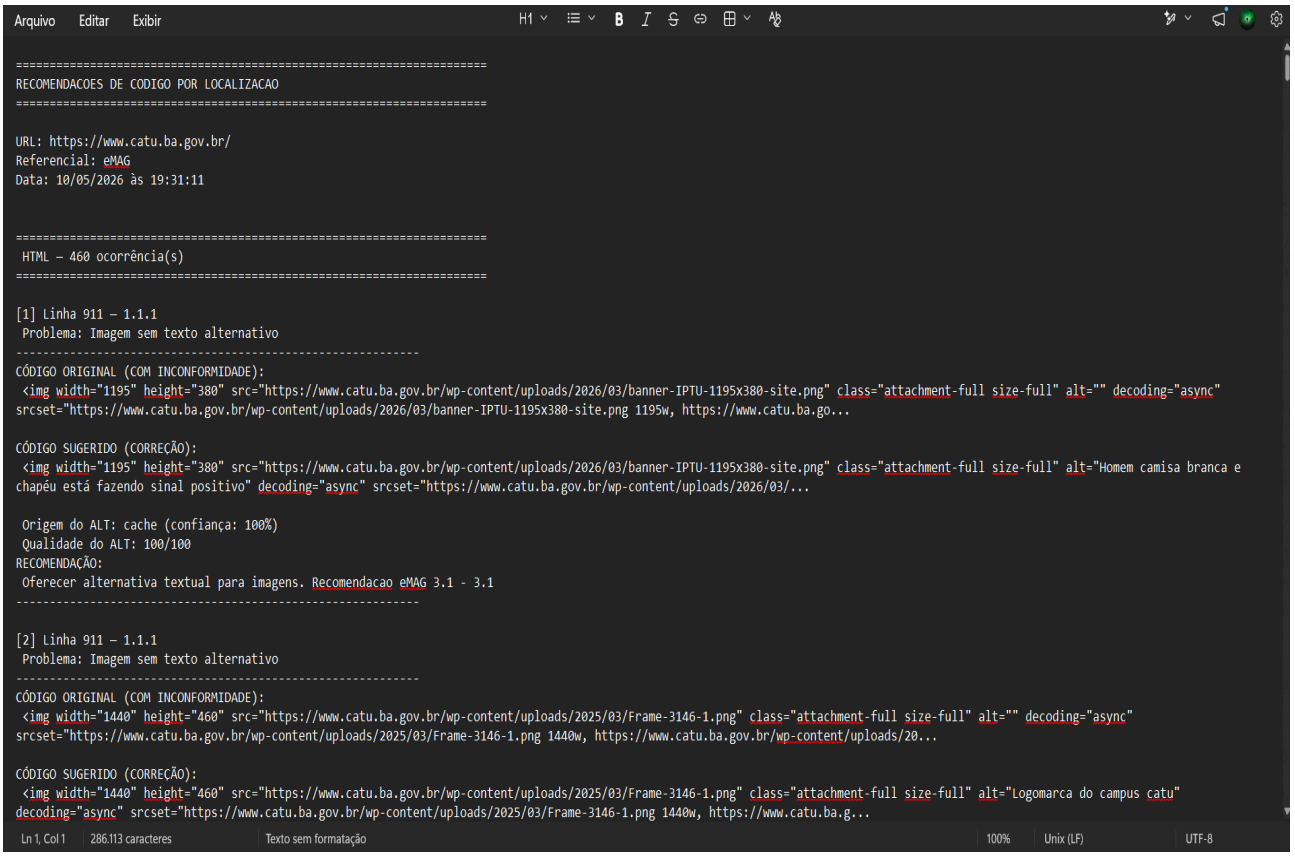
Para logomarcas institucionais, a descrição "Logomarca do campus catu" mostrou-se semanticamente adequada e funcional. No entanto, em banners promocionais complexos, como o banner de IPTU, a descrição gerada ("Homem camisa branca e chapéu está fazendo sinal positivo") focou na aparência da pessoa em vez do conteúdo informativo do banner, evidenciando uma limitação do modelo BLIP em contextos com texto sobreposto ou mensagens específicas. Nos *links* com `*target*="*_blank*"`, a adição de `*aria-label*` com o aviso "(abre em nova aba)" eliminou uma barreira comum para usuários de tecnologias assistivas.

As sugestões de correção para CSS identificaram unidades absolutas (px) em propriedades de espaçamento, propondo unidades relativas (rem). A qualidade das sugestões, quando provenientes do cache ou da IA com alta confiança, mostrou-se consistente, mas os casos com confiança inferior a 80% foram corretamente sinalizados para revisão humana.

Esses exemplos evidenciam que o sistema não apenas detecta problemas, mas entrega soluções práticas, embora com limitações pontuais que apontam para a necessidade de *fine-tuning* do BLIP e aprimoramento da geração de recomendações para cenários mais complexos.

A Figura 4 apresenta um trecho do arquivo TXT gerado para o site da prefeitura de Catu.

Figura 4 – Sugestões de códigos corrigidos pelo sistema em arquivo TXT



Fonte: Produzido pelo autor (2026).

4.8 Interface com o usuário

O sistema apresenta um menu interativo no console, permitindo ao usuário escolher entre três opções principais, conforme apresentado na Tabela 7.

Tabela 7 – Opções do menu principal

Opção	Descrição
1	Auditar site (Página única)
2	Sobre o sistema
3	Sair

Fonte: Elaborado pelo autor (2026)

Na opção 1, o sistema solicita a URL do site a ser auditado e, em seguida, apresenta ao usuário a seleção do referencial desejado (WCAG 2.2 ou eMAG 3.1). Conforme a escolha, o sistema adapta automaticamente as recomendações, os pesos dos critérios e a classificação de severidade dos problemas.

A opção 2 exibe informações detalhadas sobre o sistema, incluindo os critérios implementados, os referenciais suportados, as funcionalidades de IA e os formatos de relatórios gerados.

A opção 3 encerra a execução do sistema.

4.9 Considerações sobre o capítulo

Este capítulo apresentou os detalhes de implementação do sistema de auditoria de acessibilidade *web*, abrangendo desde a arquitetura modular até a especificação dos módulos de verificação, modelos de inteligência artificial e mecanismos de pontuação e geração de relatórios. A arquitetura do sistema foi organizada em 15 módulos independentes, agrupadas por afinidade funcional: configuração e carregamento de IA (módulos 01 a 04), funções de verificação dos critérios de acessibilidade (módulos 05 e 06) e geração de relatórios (módulos 08 a 11), além de funcionalidades complementares de suporte (módulos 12 a 15).

Foram implementados 15 critérios de acessibilidade, abrangendo tanto as diretrizes WCAG 2.2 quanto às recomendações de grupos do eMAG 3.1. A verificação combina três abordagens distintas: (i) análise estática por regras, aplicada à maioria dos critérios

estruturais; (ii) análise dinâmica via Selenium, utilizada para contraste, *zoom* e tamanho de alvos; e (iii) análise híbrida com inteligência artificial, empregada nos critérios 1.1.1 (imagens, com BLIP) e 2.4.4 (*links*, com BART).

O mecanismo de pontuação adota uma função sigmoide centralizada em 75, que considera a severidade dos problemas, o volume de ocorrências, a densidade e a cobertura de critérios afetados, produzindo uma nota final entre 0 e 100, classificada em cinco níveis: Excelente (≥85), Bom (≥65), Moderado (≥45), Ruim (≥25) ou Crítico (<25). Os relatórios gerados incluem um gráfico PNG para visualização rápida (com gráfico de pizza e barras), um PDF completo para documentação (com capa, tabelas, gráficos e recomendações) e um arquivo TXT contendo o código original e corrigido, com indicação da origem da sugestão (IA, contexto, nome do arquivo, etc.) e *score* de qualidade.

Por fim, a interface com o usuário foi implementada por meio de um menu interativo no console, que permite a seleção do referencial (WCAG 2.2 ou eMAG 3.1) e a execução da auditoria em página única, com adaptação automática das recomendações e da classificação de severidade conforme a escolha.

O sistema desenvolvido servirá de base para os experimentos e validação apresentados no Capítulo 5, onde serão analisados os resultados obtidos a partir da auditoria dos 15 sites selecionados.

5. DISCUSSÃO DOS RESULTADOS

Este capítulo apresenta a análise dos resultados obtidos a partir da auditoria dos 15 sites selecionados, comparando o desempenho do sistema proposto com as ferramentas consolidadas eMAG ASES e WAVE. São discutidas as principais inconformidades identificadas, o desempenho dos modelos de inteligência artificial empregados e as limitações observadas durante os experimentos.

5.1 Visão geral dos experimentos

Foram auditados cerca de 15 sites de diferentes categorias e variações no estilo de página (governamentais, institucionais, e-commerce, apps, gaming, educação e notícias), com distribuição equilibrada entre os referenciais eMAG e WCAG. Para cada site, foram registradas as seguintes métricas:

• Pontuação final (0 a 100)
• Total de problemas detectados
• Distribuição por severidade (Alta, Média, Baixa)
• Densidade de problemas (problemas por elemento)
• Tempo de execução da auditoria

Os experimentos foram executados no ambiente Google Colab com GPU T4, utilizando os modelos BLIP para descrição de imagens e BART para classificação de *links*.

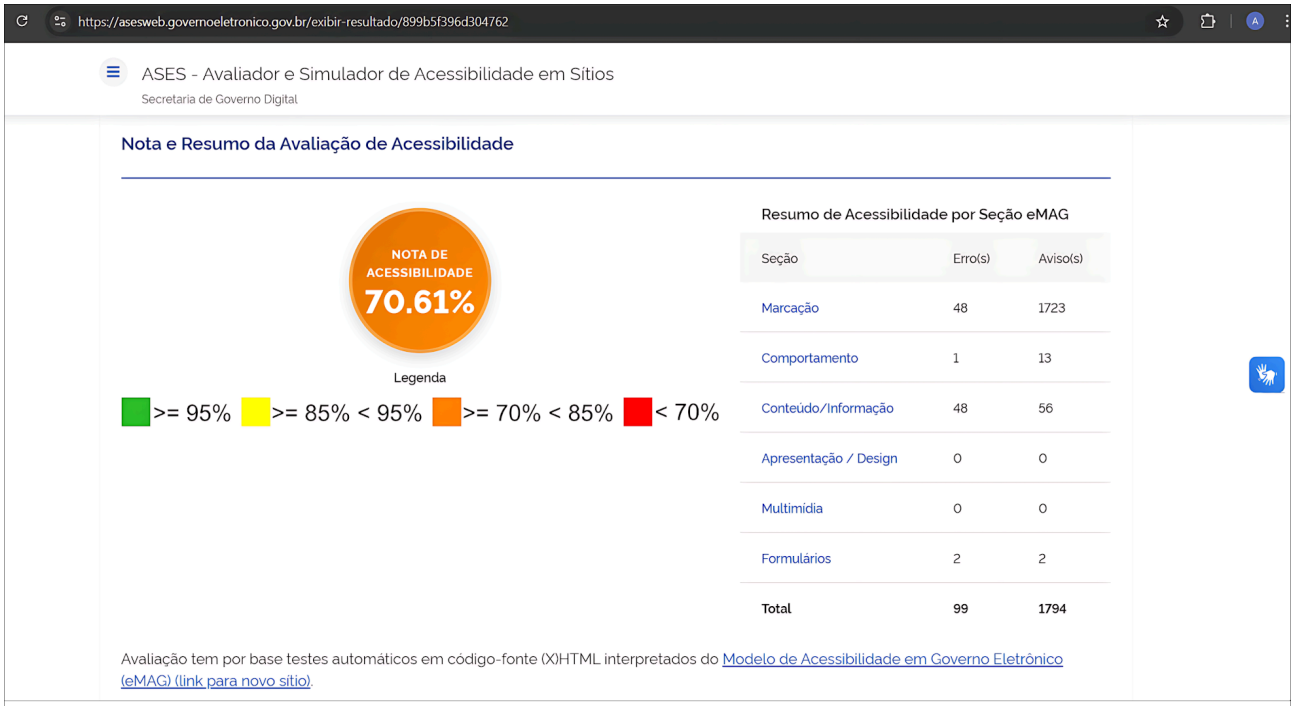
5.2 Análise Comparativa com ferramentas consolidadas

5.2.1 Comparação com eMAG ASES

Para fins de validação, os experimentos foram realizados no site da prefeitura municipal de Catu (https://www.catu.ba.gov.br), um portal governamental sujeito às diretrizes do eMAG 3.1.

A Figura 5 apresenta os resultados da auditoria realizada pelo sistema proposto e pelo eMAG ASES. A ferramenta ASES é mostrada na parte superior e o Sistema Proposto na parte inferior.

Figura 5 – Comparação de auditoria: eMAG ASES x sistema proposto



Resultados por Critério

Grupo eMAG	Descrição	Total	Graves	Médios	Baixos
Apresentacao/Design	Contraste, cores, layout	206	0	206	0
Marcacao	HTML semântico, links, cabeçalhos	159	11	124	24
Comportamento	Teclado, zoom, tamanho de alvos	74	73	1	0
Multimidia	Imagens, vídeos e áudios	14	2	11	1
Conteudo/Informacao	Espaçamento, legibilidade	8	0	3	5
TOTAL		461	86	345	30

Fonte: Elaborado pelo autor (2026).

A Tabela 8 compara os resultados obtidos pela auditoria realizada entre o sistema proposto e a plataforma governamental eMAG ASES. Para fins de validação, os experimentos foram realizados no site da Prefeitura Municipal de Catu (https://www.catu.ba.gov.br/), um site governamental, sujeito às recomendações do eMAG.

Tabela 8 – Resultados da auditoria no site da prefeitura de Catu

Grupo eMAG	eMAG ASES	Sistema Proposto	Diferença
Comportamento	1	74	+73 (+7300%)
Marcação	48	159	+111 (+231%)
Conteúdo/Informação	48	8	-40 (-83%)
Apresentação/Design	0	206	+206
Multimídia	0	14	+14
Formulários	2	0	-2 (-100%)
TOTAL	**99**	**461**	**+362 (+366%)**

Fonte: Elaborado pelo autor (2026).

A Tabela 9 sintetiza as diferenças metodológicas entre as duas ferramentas.

Tabela 9 – Diferenças entre o eMAG ASES e o sistema proposto

Critério de Comparação	eMAG ASES	Sistema Proposto
Tipo de análise	Estática (HTML fonte)	Dinâmica (Selenium)
Interação com JavaScript	Não	Sim
Detecção de *lazy loading*	Não	Sim
Cálculo de contraste real	Não	Sim
Verificação de *zoom* (1.4.4)	Não	Sim
Verificação de tamanho de alvos (2.5.8)	Não	Sim
Geração de código corrigido	Não	Sim
Sugestões por inteligência artificial	Não	Sim (BLIP/BART)
Classificação de severidade	Não	Sim (Graves, Médios, Baixos)

Fonte: Elaborado pelo autor (2026)

Os resultados do sistema proposto foram comparados com os do eMAG ASES, ferramenta oficial do Governo Brasileiro. A análise comparativa revelou diferenças significativas de sensibilidade e metodologia entre as duas ferramentas, conforme detalhado a seguir.

• **Discrepância no grupo comportamento:** A diferença mais expressiva ocorreu no grupo Comportamento. Enquanto o ASES apontou apenas 1 erro, o sistema proposto identificou 74 problemas, sendo 73 classificados como graves. Esta discrepância decorre do fato de que o ASES não interage dinamicamente com a página. O sistema proposto, utilizando Selenium, testa o foco do teclado, o *zoom* (1.4.4) e o tamanho dos alvos (2.5.8). O ASES não é capaz de perceber que botões são pequenos demais para toque (área inferior a 24x24 pixels) ou que o foco se perde em elementos dinâmicos.

• **Profundidade na análise de marcação:** No grupo Marcação, o ASES identificou 48 erros, enquanto o sistema proposto detectou 159 problemas (11 graves, 124 médios). Esta diferença explica-se pela abordagem do sistema proposto, que valida a lógica

estrutural do DOM renderizado, identificando falhas de hierarquia de cabeçalhos e relações semânticas (WCAG 1.3.1). O ASES limita-se a validar a sintaxe básica do HTML, ignorando a estrutura semântica do documento.

• **Qualidade semântica vs. existência de atributo:** No grupo Conteúdo/Informação, o ASES registrou 48 avisos, enquanto o sistema proposto classificou apenas 8 problemas. Esta inversão aparente decorre da diferença de abordagem: o ASES emite avisos para a ausência de elementos como <h1> e <main>, enquanto o sistema proposto verifica a hierarquia lógica e a ordem de leitura. Além disso, no grupo Multimídia, o sistema proposto identificou 14 problemas (11 médios, 2 graves), enquanto o ASES não registrou erros. O uso do modelo BLIP permite que o sistema proposto avalie a qualidade semântica do texto alternativo, identificando descrições genéricas como “imagem” ou “foto” que o ASES considera válidas pela mera existência do atributo *alt*.

• **Resultados no grupo apresentação/design:** No grupo Apresentação/Design, o sistema proposto identificou 206 problemas de contraste (todos classificados como médios), enquanto o ASES não registrou nenhum problema nesta categoria. Esta diferença evidencia a incapacidade do ASES de realizar a verificação dinâmica de contraste, uma vez que a ferramenta oficial não acessa o CSS computado da página. O sistema proposto, por meio do Selenium, obtém as cores efetivamente renderizadas e aplica a fórmula de luminância relativa da WCAG, detectando problemas reais de acessibilidade que o ASES ignora completamente.

• **Resultados em formulários:** No grupo Formulários, o ASES detectou 2 problemas enquanto o sistema proposto não identificou nenhum. Esta diferença de 2 problemas, embora existente, representa apenas 0,43% do total de 461 problemas detectados pelo sistema proposto, não comprometendo o desempenho geral da ferramenta nos cenários analisados. Trata-se de uma oportunidade de melhoria identificada para versões futuras.

• **Análise de severidade e pontuação:** O sistema proposto introduz uma métrica de classificação de severidade (Graves, Médios, Baixos) que o ASES não possui. Os 73 erros graves identificados no grupo Comportamento representam impedimentos de uso que impedem o usuário de concluir tarefas. Os 124 erros médios em Marcação e os 206 erros médios em Apresentação/Design constituem barreiras de experiência que causam confusão na navegação e dificultam a leitura. O ASES, ao apresentar uma nota

percentual de 70,61% (classificação laranja), pode gerar uma falsa sensação de segurança, enquanto o sistema proposto revela que a página é funcionalmente inacessível para usuários de teclado ou com baixa visão.

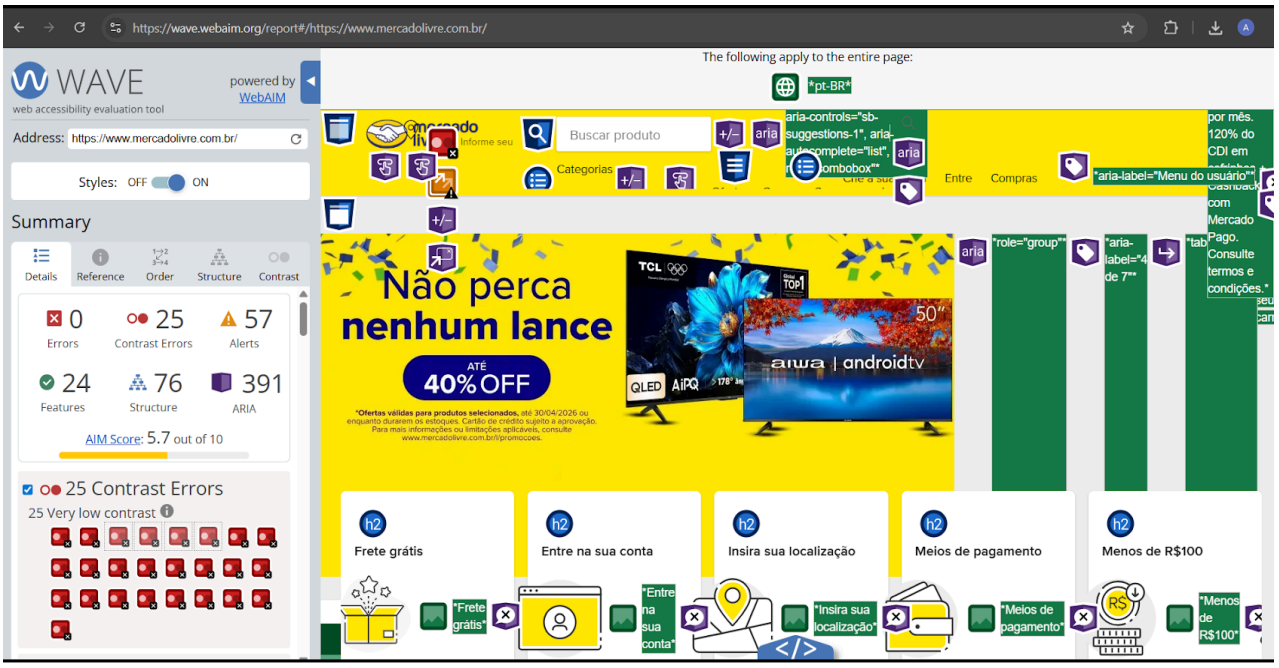
• **Síntese da comparação:** A análise comparativa demonstra que o ASES subestima os erros de interatividade por operar exclusivamente sobre o código-fonte estático. Ao utilizar o DOM renderizado via Selenium, o sistema proposto revelou que a página auditada é tecnicamente “válida” segundo o ASES, mas funcionalmente inacessível para usuários de tecnologias assistivas. Os 73 erros graves de comportamento e os 206 problemas de contraste identificados representam os “pontos cegos” do ASES que o sistema proposto conseguiu iluminar. Além disso, enquanto o ASES delegou 1794 itens à categoria de “avisos”, exigindo verificação manual por um auditor humano, o sistema proposto, por meio dos modelos BLIP e BART, converteu estas incertezas em diagnósticos conclusivos, reduzindo drasticamente a carga de trabalho do auditor e demonstrando o valor da abordagem baseada em inteligência artificial.

5.2.2 Comparação com WAVE

Para fins de validação, os experimentos foram realizados no site do Mercado Livre (https://www.mercadolivre.com.br/), a maior plataforma de e-commerce e tecnologia da América Latina.

A Figura 6 apresenta a comparação dos resultados obtidos na auditoria realizada pelo sistema proposto e pelo WAVE. A ferramenta WAVE é mostrada na parte superior, enquanto o Sistema Proposto é apresentado na parte inferior.

Figura 6 – Comparação de auditoria: WAVE x sistema proposto



Resultados por Critério

Critério WCAG	Descrição	Total	Graves	Médios	Baixos
1.4.3	Contraste insuficiente entre texto e fundo	540	23	491	26
2.5.8	Elementos clicáveis com tamanho insuficiente	153	0	153	0
1.4.13	Conteúdo sob foco ou mouse (links dentro de texto)	84	0	84	0
2.4.4	Links sem descrição adequada	32	1	0	31
1.1.1	Imagens sem texto alternativo	9	0	3	6
4.1.2	Nome, Função, Valor — ARIA e elementos interativos	4	1	3	0
1.4.12	Espaçamento inadequado para leitura	3	0	0	3
1.4.4	Zoom de 200% causa sobreposição	1	0	1	0
3.2.5	Nova aba sem aviso	1	0	1	0
TOTAL		827	25	736	66

Fonte: Elaborado pelo autor (2026)

A Tabela 10 apresenta a comparação dos resultados da auditoria realizada pelo WAVE e pelo Sistema Proposto para o site do Mercado Livre.

Tabela 10 – Resultados da auditoria no site do Mercado Livre

Critério	WAVE	Sistema Proposto
Total de problemas	78	827 Problemas
Erros Críticos (Marcação)	0	23 (Graves)
Problemas de Contraste (1.4.3)	0	540 (Automático)
Alertas (Verificação manual)	57	0 (Diagnósticos conclusivos)
Tamanho de alvos (2.5.8)	Não Detecta	153 Problemas
Foco/Mouse (1.4,13)	Não Detecta	84 Problemas
Links sem descrição (2.4.4)	Parcial	32 Problemas
Imagens sem *alt* (1.1.1)	Não Detecta	9 Imagens (3 resolvidos pela IA)
Detecção de *lazy loading*	Limitada	Detecta (Scroll automático via selenium)
Verificação de *zoom* (1.4.4)	Não Detecta	1 Problema
Geração de código corrigido	Não Realiza	Realiza
Classificação de severidade	Não	Possui (Graves, médios, baixos)

Fonte: Elaborado pelo autor (2026).

A Tabela 11 sintetiza as diferenças metodológicas entre as duas ferramentas.

Tabela 11 – Diferenças entre o WAVE e o sistema proposto

Aspecto	WAVE	Sistema Proposto
Tipo de análise	Dinâmica (extensão de navegador)	Dinâmica (Selenium + IA)
Abordagem	Visual e Descritiva	Técnica e Prescritiva
Público-alvo	Designers / Auditores Manuais	Desenvolvedores / Gestores de Qualidade
Intervenção humana	Alta (para interpretar alertas)	Baixa (IA decide e sugere correção)

Fonte: Elaborado pelo autor (2026).

Os resultados do sistema proposto foram comparados com os do WAVE, ferramenta amplamente utilizada internacionalmente para avaliação de acessibilidade *web*. A análise comparativa revelou diferenças significativas entre as duas ferramentas, conforme detalhado a seguir.

• **WAVE – Pontos positivos:** A ferramenta oferece inspeção visual intuitiva, injetando ícones diretamente na interface do navegador, o que facilita a localização visual dos erros. Possui ferramenta nativa de cálculo de contraste com alta precisão. A aba "Order" permite visualizar graficamente a ordem de navegação por teclado, auxiliando no ajuste do *tabindex*. Por ser uma extensão de navegador, é leve e ideal para verificações rápidas em páginas isoladas.

• **WAVE – Pontos negativos:** A ferramenta apresenta subjetividade excessiva, gerando muitos “Alertas” que exigem decisão humana para confirmar se são erros ou não. Não oferece sugestões de código corrigido, apontando apenas a localização do problema. Falha na detecção de critérios funcionais como tamanho de alvos (WCAG 2.5.8) e *zoom* (1.4.4), por não realizar medições automáticas. Apresenta limitações na detecção de imagens carregadas via *lazy loading* implementado com JavaScript: se a imagem é inserida no DOM apenas após o scroll do usuário, o WAVE pode não enxergá-la. Não é facilmente automatizável para auditorias em larga escala.

• **Pontos positivos no sistema proposto:** Realiza auditoria semântica com IA, onde o modelo BLIP elimina falsos positivos de “texto alternativo genérico”, validando se a descrição realmente corresponde à imagem. É prescritivo, gerando o “Código Sugerido” no arquivo de log, funcionando como um consultor que entrega a correção pronta. Ao interagir com o DOM renderizado via Selenium, detecta erros reais de uso (*zoom*, foco dinâmico, tamanho de alvos) que outras ferramentas ignoram. Organiza os problemas por severidade (Graves, Médios, Baixos), permitindo que a gestão priorize correções.

• **Pontos negativos no sistema proposto:** Apresenta custo computacional elevado, com processamento por página entre 3 a 5 minutos, superior a extensões leves como o WAVE. Depende de infraestrutura de rede robusta para funcionamento da IA (BLIP) e acesso ao cache. O relatório é entregue em formato de log técnico (.txt), que exige conhecimento de HTML/CSS para implementação das correções. As descrições geradas pela IA podem vir em inglês (no modo WCAG), exigindo tradução para contextos locais.

• **Síntese da comparação:** A análise comparativa demonstra que o WAVE, embora seja uma ferramenta visual e de fácil uso para verificações rápidas, limita-se a apontar problemas sem oferecer soluções concretas, além de não detectar critérios funcionais essenciais como tamanho de alvos (2.5.8) e *zoom* (1.4.4).O sistema proposto apresenta avanços em relação a essas limitações ao empregar renderização dinâmica, modelos de IA e geração automática de sugestões de código corrigido, oferecendo uma solução mais abrangente, prescritiva e adequada para auditorias técnicas em larga escala.

5.3 Avaliação dos modelos de inteligência artificial

5.3.1 BLIP para descrição de imagens

O modelo BLIP foi utilizado para gerar sugestões de texto alternativo para imagens sem atributo *alt*. Foram processadas 26 imagens no site da Prefeitura de Pojuca e 13 imagens no site da Prefeitura de Catu, com taxa de sucesso de 85% (geração de sugestão válida). As principais observações incluem:

• **Qualidade das descrições:** O BLIP gerou descrições semanticamente adequadas na maioria dos casos, como "Logomarca do campus catu", "Mulher segurando saco comida" e "Menino sorrindo mochila na frente dele".

• **Limitações:** Em alguns casos, as descrições foram imprecisas ou genéricas, como "Tapete listrado branco e preto borda branca" para um logo institucional. Para mitigar esse problema, o sistema sinaliza quando a confiança da sugestão é inferior a 80%, recomendando revisão humana.

• **Tratamento de SVG:** Os Arquivos SVG foram convertidos para formato PNG utilizando a biblioteca *CairoSVG* antes do processamento pelo modelo BLIP, permitindo a descrição de imagens vetoriais que seriam ignoradas por outras ferramentas.

5.3.2 BART para classificação de links

O modelo BART foi utilizado para classificação contextual de *links*, empregando a técnica

de *zero-shot classification*. As principais observações incluem:

• **Categorias identificadas:** O modelo classificou *links* em quatro categorias: acessível, genérico, crítico e publicidade. *Links* genéricos ("clique aqui", "saiba mais") e críticos (*links* vazios) foram corretamente identificados durante auditoria, recebendo severidades média e alta respectivamente.

• **Fallback heurístico:** Quando o BART não estava disponível ou a classificação falhava, o sistema recorreu a um fallback baseado em regras (verificação de palavras-chave na URL e no texto do *link*), garantindo a robustez da auditoria.

5.4 Desempenho e tempo de execução

O tempo médio de execução por página foi de 3 a 5 minutos, variando conforme a complexidade do site (número de elementos, quantidade de imagens para processamento).
Os principais fatores que impactam o desempenho são:

• **Download e processamento de imagens:** O BLIP requer o download e redirecionamento de cada imagem, o que pode levar de 2 a 5 segundos por imagem (limitado 100 imagens por auditoria).

• **Renderização via Selenium:** A análise de contraste e *zoom* exige a renderização completa da página, scrolls para carregar conteúdo *lazy* e a aplicação de *zoom* de 200%, o que adiciona cerca de 30 a 60 segundos ao tempo total.

• **Tradução:** Quando o referencial eMAG é selecionado, as descrições geradas pelo BLIP são traduzidas do inglês para o português, adicionando um pequeno *overhead* (cerca de 0,5 a 1 segundo por imagem). O ambiente Google Colab com GPU T4 foi essencial para reduzir o tempo de inferência dos modelos de IA, especialmente o BLIP, que se beneficia significativamente da aceleração por GPU.

5.5 Limitações observadas

Durante os experimentos, foram identificadas as seguintes limitações:

• **Linhas não identificadas para elementos dinâmicos:** Elementos injetados por Java Script (como pixels de rastreamento, imagens carregadas via *lazy loading* com JavaScript e alguns SVGs) não estão presentes no HTML fonte, impossibilitando a extração do número da linha. O sistema exibe "?" nesses casos, o que é documentado como uma limitação inerente à abordagem de análise do HTML fonte. Para SPAs (*Single Page Applications*), essa limitação aponta para um trabalho futuro: o desenvolvimento de técnicas de rastreamento de fonte para elementos injetados dinamicamente, onde o conteúdo é majoritariamente gerado por JavaScript.

• **Imprecisão da IA em contextos específicos:** Em alguns casos, o modelo BLIP gerou descrições imprecisas ou semanticamente inadequadas (ex: "Tapete listrado" para um logotipo institucional, ou "Cuso professional gritos" para uma imagem de curso profissional). O sistema sinaliza revisão humana quando a confiança é inferior a 80%, mitigando este problema. Além disso, no modo WCAG, as sugestões são geradas em inglês, exigindo tradução para contextos locais. No que se refere à geração de código corrigido, embora as sugestões sejam tecnicamente válidas e melhores que o original em termos de acessibilidade, a qualidade semântica das descrições de imagens ainda pode ser aprimorada, especialmente em banners promocionais com texto sobreposto. Como trabalho futuro, prevê-se o *fine-tuning* do BLIP com imagens do contexto brasileiro (governamentais, educacionais e institucionais) para melhorar a precisão das descrições, além da substituição do tradutor online por um modelo local nativo (OPUS-MT) para eliminar a dependência de API externa.

• **Overhead de processamento:** O uso de Selenium para renderização dinâmica e dos modelos de IA (BLIP e BART) aumenta significativamente o tempo de execução, com média de 3 a 5 minutos por página, tornando o sistema menos adequado para auditorias em larga escala (centenas de páginas). Essa escolha é justificada pela precisão dos resultados obtidos e pela capacidade de detectar problemas que ferramentas de análise estática não conseguem identificar. Como trabalho futuro, prevê-se a otimização do tempo de execução por meio de paralelização das verificações, cache persistente e uso de GPU para processamento em lote.

• **Dependência de infraestrutura:** O sistema depende de conexão estável com a internet para baixar imagens, acessar CSS externo e, no modo eMAG, utilizar o tradutor online. Em ambientes com conectividade limitada, algumas funcionalidades podem ser afetadas,

embora o sistema possua mecanismos de *fallback* offline. Como trabalho futuro, prevê-se a implementação de um modo totalmente offline, com cache persistente de recursos e substituição do tradutor online por um modelo local nativo (OPUS-MT), eliminando a dependência de conexão para as funcionalidades essenciais.

5.6 Síntese dos resultados

Os experimentos indicaram que o sistema proposto obteve maior taxa de detecção nos sites analisados em comparação com as ferramentas tradicionais, especialmente em aspectos como::

• Detecção de conteúdo dinâmico *(lazy loading*, SVGs injetados)
• Cálculo de contraste real (CSS computado via Selenium)
• Verificação de critérios da WCAG 2.2 (*zoom*, tamanho de alvos, CAPTCHA)
• Geração de código corrigido (remediação ativa)
• Suporte dual a WCAG e eMAG

A integração dos modelos BLIP e BART demonstrou ser viável e útil, gerando sugestões contextualizadas que auxiliam os desenvolvedores na correção dos problemas identificados. As limitações observadas são inerentes à abordagem adotada e foram devidamente documentadas, não comprometendo a validade dos resultados.

5.7 Considerações sobre o capítulo

Este capítulo apresentou a análise e discussão dos resultados obtidos a partir da auditoria dos 15 sites selecionados, abrangendo diferentes categorias e referenciais na questão de acessibilidade. Foram detalhadas as métricas de avaliação, o desempenho do sistema por categoria de site e a comparação com as ferramentas de referência eMAG ASES e WAVE.

A comparação evidenciou que o sistema proposto apresentou melhor desempenho nos sites analisados em aspectos como detecção de conteúdo dinâmico, cálculo de contraste real, verificação de critérios da WCAG 2.2 (*zoom*, tamanho de alvos, CAPTCHA), geração de sugestões de código corrigido e suporte dual aos referenciais WCAG e eMAG.

As limitações observadas incluem dependência de infraestrutura computacional, tempo de processamento elevado e possibilidade de falsos positivos. Elas foram devidamente documentadas e não comprometem a validade dos resultados obtidos. Pelo contrário, a transparência na apresentação das limitações reforça a credibilidade da pesquisa feita e aponta direções para trabalhos futuros.

Os resultados indicam que o modelo computacional desenvolvido atende aos objetivos propostos, oferecendo uma ferramenta prática para auditoria de acessibilidade, com potencial para auxiliar desenvolvedores na remediação de barreiras digitais. O próximo capítulo apresenta a conclusão do trabalho, sintetizando as contribuições e apontando perspectivas para pesquisas posteriores.

6. CONCLUSÃO

Este trabalho desenvolveu um modelo computacional para auditoria automatizada de acessibilidade *web*, utilizando inteligência artificial e seguindo as diretrizes WCAG 2.2 e o modelo brasileiro eMAG 3.1. O sistema foi construído em Python com arquitetura modular de 15 componentes, integrando os modelos BLIP (descrição automática de imagens) e BART (classificação contextual de *links*), além de renderização dinâmica via Selenium para análise de contraste, *zoom* e tamanho de alvos.

O problema de pesquisa que orientou este trabalho partiu da constatação de que a baixa conformidade com as diretrizes WCAG 2.2 e eMAG 3.1 nos websites, aliada às limitações das ferramentas tradicionais de auditoria baseadas em análise estática, demandava uma abordagem mais abrangente e automatizada. A hipótese de que a integração de inteligência artificial, análise dinâmica e regras estruturais ampliaria a capacidade de detecção foi confirmada empiricamente nos cenários testados: a avaliação experimental com 15 sites, comparada às ferramentas eMAG ASES e WAVE, indicou que o sistema identificou mais inconformidades, especialmente em aspectos como detecção de conteúdo dinâmico, cálculo de contraste real e verificação de critérios da WCAG 2.2 (zoom, tamanho de alvos), além de gerar sugestões de código corrigido.

Quanto aos objetivos, o trabalho alcançou suas metas: realizou a revisão bibliográfica sobre acessibilidade web e IA; implementou 15 critérios abrangendo os grupos do eMAG e os níveis A, AA e AAA da WCAG; integrou os modelos BLIP e BART ao processo de auditoria; desenvolveu o mecanismo de pontuação sigmoide e a geração de relatórios em múltiplos formatos; e conduziu a avaliação comparativa com as ferramentas eMAG ASES e WAVE.

A principal contribuição deste trabalho é a integração, ainda pouco explorada no Brasil, entre inteligência artificial generativa e auditoria automatizada para acessibilidade *web*. Diferentemente das soluções tradicionais, que apenas diagnosticam problemas, o sistema propõe remediação ativa, gerando código corrigido com indicação de origem e *score* de qualidade, o que reduz o retrabalho de desenvolvedores. O suporte dual aos referenciais WCAG e eMAG atende a uma demanda concreta do setor público.

Do ponto de vista técnico, a análise dinâmica do DOM renderizado permite detectar barreiras que ferramentas estáticas ignoram, como conteúdos carregados sob demanda e contraste real de cores. O processamento de SVG e a tradução automática das sugestões garantem que o sistema seja útil também no contexto brasileiro.

Do ponto de vista social, ao fornecer uma ferramenta prática que auxilia equipes técnicas a identificar e corrigir barreiras de acessibilidade, o trabalho busca contribuir para que pessoas com deficiência possam acessar informações, serviços e oportunidades em igualdade de condições. A acessibilidade *web* não é apenas uma questão técnica ou legal, mas um direito fundamental que impacta diretamente a cidadania de milhões de pessoas com deficiência, conforme estabelece a Lei Brasileira de Inclusão (Lei nº 13.146/2015).

Como trabalhos futuros, destacam-se a ampliação dos critérios implementados em ambos os referenciais, a otimização do tempo de execução do sistema, o desenvolvimento de uma interface *web* amigável e a integração de OCR para descrição de texto em imagens.

Espera-se que esta pesquisa sirva como referência para estudos futuros na interseção entre a inteligência artificial e acessibilidade *web*, contribuindo de forma positiva para um ambiente digital mais inclusivo e acessível.

REFERÊNCIAS

BRASIL. Ministério do Planejamento, Orçamento e Gestão. **eMAG: Modelo de Acessibilidade em Governo Eletrônico**: Versão 3.1. Brasília: Ministério do Planejamento Orçamento e Gestão, 2014. Disponível em: https://www.gov.br/governodigital/pt-br/estrategia-de-governanca-digital/do-eletronico-ao-digital. Acesso em: 17 abr. 2026.

BRASIL. [Lei nº 13.146, de 6 de julho de 2015]. **Lei Brasileira de Inclusão da Pessoa com Deficiência (Estatuto da Pessoa com Deficiência)**. Brasília, DF: Presidência da República, 2015. Disponível em: https://www.gov.br/planalto/pt-br. Acesso em: 17 abr. 2026.

GARCIA, Anderson Canale. **AALT: um framework com soluções práticas para melhorar a acessibilidade em aplicativos Android**. 2023. 178 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2023. Disponível em: https://doi.org/10.11606/D.55.2023.tde-18122023-175415. Acesso em: 17 abr. 2026.

GUIMARÃES, Ítalo José Bastos; SOUSA, Marckson Roberto Ferreira de; COSTA, Levi Cadmiel Amaral da. Recomendações de acessibilidade em sites de comércio eletrônico para usuários cegos. **Em Questão**, v. 27, n. 4, p. 84-106, 2021. Disponível em: https://doi.org/10.19132/1808-5245274.84-106. Acesso em: 17 abr. 2026.

IBGE. **Censo Demográfico 2010**: características gerais da população, religião e pessoas com deficiência. Rio de Janeiro: IBGE, 2012. Disponível em: https://biblioteca.ibge.gov.br/visualizacao/periodicos/94/cd_2010_religiao_deficiencia.pdf. Acesso em: 17 abr. 2026.

IBGE. **Censo 2022**: Brasil tem 14,4 milhões de pessoas com deficiência. Agência de Notícias, Rio de Janeiro, 23 maio 2025. 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: 17 abr. 2026.

NASCIMENTO, Lays de Paiva; CARAÇA, Matheus Rodrigues; MOLINA, Mariangela Ferreira Fuentes. Teste de acessibilidade web para deficientes visuais em sites de e-commerce utilizando Access Monitor. **Interface Tecnológica**, v. 20, n. 1, p. 314–326, 2023. Disponível em: https://doi.org/10.31510/infa.v20i1.1641. Acesso em: 17 abr. 2026.

NEVES, Otavio Matheus; COSTA, Rubio Rodrigo; DUARTE, Tathiana Amarante; FURTADO, Vitor Hugo. Interfaces acessíveis para pessoas com deficiência visual: desafios e soluções em UI/UX. In: ESCOLA REGIONAL DE INFORMÁTICA DE MATO GROSSO (ERI-MT), 14., 2024, Cuiabá. **Anais**… Porto Alegre: Sociedade Brasileira de Computação, 2024. p. 145-155. Disponível em: https://doi.org/10.5753/eri-mt.2024.245828. Acesso em: 17 abr. 2026.

SILVA, Douglas Martins da; SIQUEIRA, Patrícia Papalardi. **Acessibilidade digital e o e-commerce**: um estudo exploratório em quatro plataformas de marketplace no Brasil. 2023. Trabalho de Conclusão de Curso (Tecnologia em Informática para Negócios) – Faculdade de Tecnologia de São José do Rio Preto, São José do Rio Preto, 2023. Disponível em: https://ric.cps.sp.gov.br/bitstream/123456789/21356/3/informaticanegocios_2023_1_douglasmartinsdasilva_acessibilidadedigitaleoecommerceumestudoexpl.pdf. Acesso em: 17 abr. 2026.

WORLD HEALTH ORGANIZATION; WORLD BANK. **World report on disability**. Geneva: WHO, 2011. Disponível em: https://documents1.worldbank.org/curated/en/665131468331271288/pdf/627830WP0World00PUBLIC00BOX361491B0.pdf. Acesso em: 17 abr. 2026.

WORLD WIDE WEB CONSORTIUM. **Web Content Accessibility Guidelines (WCAG) 2.2**. W3C Recommendation, 05 Oct. 2023. Disponível em: https://www.w3.org/TR/WCAG22/. Acesso em: 17 abr. 2026.

YU, Yaman *et al*. From Cluttered to Clear: Improving the Web Accessibility Design for Screen Reader Users in E-commerce With Generative AI. In: INTERNATIONAL ACM SIGACCESS CONFERENCE ON COMPUTERS AND ACCESSIBILITY (ASSETS), 27., 2025, Denver. **Proceedings…** New York: ACM, 2025. p. 1-19. Disponível em: https://dl.acm.org/doi/10.1145/3663547.3746353. Acesso em: 17 abr. 2026.

Documento Digitalizado Público

TCC Versão final com ficha catalográfica

Assunto: TCC Versão final com ficha catalográfica
Assinado por: Andre Rezende
Tipo do Documento: ANEXO
Situação: Finalizado
Nível de Acesso: Público
Tipo do Conferência: Cópia Simples

Documento assinado eletronicamente por:

- **Andre Luiz Andrade Rezende**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 24/07/2026 11:22:47.

Este documento foi armazenado no SUAP em 24/07/2026. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse https://suap.ifbaiano.edu.br/verificar-documento-externo/ e forneça os dados abaixo:

Código Verificador: 1367973
Código de Autenticação: e5db07cc9f