



Trabalho de Conclusão de Curso

Desenvolvimento de um verificador de acessibilidade web de varredura completa

Jonathas Patrick H. de Azevedo
jpha@ic.ufal.br

Orientador:
Prof. Dr. Fábio José Coutinho da Silva

Maceió, Dezembro de 2024

Jonathas Patrick H. de Azevedo

Desenvolvimento de um verificador de acessibilidade web de varredura completa

Monografia apresentada como requisito parcial para
obtenção do grau de Bacharel em Ciência da Com-
putação do Instituto de Computação da Universidade
Federal de Alagoas.

Orientador:

Prof. Dr. Fábio José Coutinho da Silva

Maceió, Dezembro de 2024

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação do Instituto de Computação da Universidade Federal de Alagoas, aprovada pela comissão examinadora que abaixo assina.

Prof. Dr. Fábio José Coutinho da Silva - Orientador
Instituto de Computação
Universidade Federal de Alagoas

Prof.^a Dr.^a Maria Cristina Tenório Cavalcante Escarpini - Examinador
Instituto de Computação
Universidade Federal de Alagoas

Prof. Dr. Thiago Damasceno Cordeiro - Examinador
Instituto de Computação
Universidade Federal de Alagoas

Catálogo na fonte
Universidade Federal de Alagoas
Biblioteca Central

Divisão de Tratamento Técnico

Bibliotecária: Helena Cristina Pimentel do Vale – CRB4/661

- A994d Azevedo, Jonathas Patrick Hermenegildo de.
Desenvolvimento de um verificador de acessibilidade web de varredura completa / Jonathas Patrick Hermenegildo de Azevedo. – 2025.
77 f.: il.
- Orientador: Fábio José Coutinho da Silva
Monografia (Trabalho de Conclusão de Curso em Ciência da Computação) – Universidade Federal de Alagoas, Instituto de Computação. Maceió, 2024.
- Bibliografia: f. 46-49.
Apêndices: f. 50-77.
1. Acessibilidade web. 2. Verificadores de acessibilidade. 3. Wcag. 4. Web crawler. 5. Museus virtuais. I. Título.

CDU: 004.4'2

Agradecimentos

Gostaria de agradecer à minha família, que sempre me apoiou desde o começo, sempre esteve comigo nos momentos difíceis e nunca duvidou que um dia eu estaria aqui.

Agradeço a todos os amigos que fiz nessa jornada acadêmica, em especial Eduardo, Natália e Thalyssa. A amizade de vocês me fez ser quem eu sou hoje, sou muito grato por terem me mostrado o real significado de uma amizade sincera e verdadeira. Sem vocês essa jornada não seria possível, agradeço imensamente por terem feito parte da minha vida e por continuarem fazendo.

Por fim, e não menos importante, gostaria de agradecer ao meu orientador Fábio Coutinho por me guiar no processo de criação deste trabalho e me dar todo o suporte necessário.

Resumo

A acessibilidade digital é um requisito essencial para garantir a inclusão de pessoas com deficiência no ambiente online, mas ainda enfrenta desafios significativos devido à falta de conformidade de muitos *websites* com as diretrizes internacionais, como as *Web Content Accessibility Guidelines* (WCAG). Este trabalho tem como objetivo desenvolver o *FullScan Accessibility Checker*, uma aplicação gratuita e automatizada capaz de avaliar a acessibilidade de todas as páginas de um *website*, proporcionando uma visão abrangente sobre seu nível de conformidade.

Como estudo de caso, a ferramenta foi aplicada em *websites* de museus virtuais, revelando que a maioria dos sites apresenta um nível intermediário de acessibilidade, com falhas recorrentes em áreas essenciais como textos alternativos para imagens e navegação intuitiva. Esses resultados evidenciam a importância de ferramentas automatizadas para diagnosticar e corrigir problemas de acessibilidade, promovendo um ambiente digital mais inclusivo. O *FullScan Accessibility Checker* se destacou como uma solução eficaz, contribuindo para a detecção de problemas e auxiliando no processo de correção dos mesmos.

Palavras-chave: acessibilidade web; verificadores de acessibilidade; métodos de avaliação de acessibilidade; wcag; web crawler; museus virtuais.

Abstract

Digital accessibility is an essential requirement to ensure the inclusion of people with disabilities in the online environment, yet it still faces significant challenges due to the lack of compliance of many websites with international guidelines, such as the Web Content Accessibility Guidelines (WCAG). This study aims to develop the FullScan Accessibility Checker, a free and automated application capable of assessing the accessibility of all pages of a website, providing a comprehensive overview of its compliance level.

As a case study, the tool was applied to virtual museum websites, revealing that most of them exhibit an intermediate level of accessibility, with recurring issues in critical areas such as alternative text for images and intuitive navigation. These findings highlight the importance of automated tools in diagnosing and addressing accessibility issues, fostering a more inclusive digital environment. The FullScan Accessibility Checker proved to be an effective solution, contributing to the detection of accessibility barriers and supporting the process of their resolution.

Key-words: web accessibility; accessibility checkers; accessibility evaluation methods; wcag; web crawlers; virtual museums.

Conteúdo

1	Introdução	1
1.1	Motivação	1
1.2	Objetivos	2
1.2.1	Objetivo Geral	2
1.2.2	Objetivos Específicos	2
1.3	Metodologia	2
1.4	Trabalhos Relacionados	3
1.5	Organização do Documento	4
2	Fundamentação Teórica	5
2.1	Acessibilidade <i>web</i>	5
2.1.1	Diretrizes WCAG	6
2.2	Verificadores de Acessibilidade	22
2.2.1	Access Monitor	24
2.3	web crawlers	27
3	Desenvolvimento	28
3.1	Busca e varredura de páginas	30
3.2	Verificação de acessibilidade de páginas	31
3.3	Interface do usuário	31
4	Estudo de caso: Museus Virtuais	35
4.1	Levantamento e coleta de dados	35
4.2	Resultados	38
5	Conclusão	44
5.1	Trabalhos Futuros	44
	Referências bibliográficas	46
	Apêndices	50
A	Critérios de Sucesso para as diretrizes WCAG	50
A.1	Diretriz 1.2 Mídias com base em tempo	50
A.2	Diretriz 1.3 Adaptável	51
A.3	Diretriz 1.4 Discernível	53
A.4	Diretriz 2.1 Acessível por Teclado	58
A.5	Diretriz 2.2 Tempo Suficiente	59
A.6	Diretriz 2.3 Convulsões e Reações Físicas	62
A.7	Diretriz 2.4 Navegável	62

	A.8	Diretriz 2.5 Modalidades de Entrada	64
	A.9	Diretriz 3.1 Legível	66
	A.10	Diretriz 3.2 Previsível	67
	A.11	Diretriz 3.3 Assistência de Entrada	68
	A.12	Diretriz 4.1 Compatível	70
B		Códigos das principais funções utilizadas na Ferramenta	70
	B.1	Função queryString()	70
	B.2	Função getWebsiteScores()	71
	B.3	Função getPageScore()	73
	B.4	Função searchThroughWebsite()	75
C		Funções utilizadas no estudo de caso	76
	C.1	Função getMuseumsSite()	76

Lista de Figuras

2.1	Taxonomia dos métodos de avaliação de acessibilidade. Fonte: adaptada de Brajnik (2008)	22
2.2	Tipos de entrada do Access Monitor. Fonte: accessmonitor.acessibilidade.gov.pt	25
2.3	Resultados de análise do Access Monitor. Fonte: accessmonitor.acessibilidade.gov.pt	26
2.4	Práticas de acessibilidade encontradas pelo Access Monitor. Fonte: accessmonitor.acessibilidade.gov.pt	26
2.5	Detalhamento e sugestões de melhoria pelo Access Monitor. Fonte: accessmonitor.acessibilidade.gov.pt	26
2.6	Arquitetura de um <i>web crawler</i> . Fonte: adaptada de Singh and Varnica (2014)	27
3.1	Visão geral do funcionamento do FullScan Accessibility Checker. Fonte: Elaborado pelo autor, 2024	29
3.2	Tela inicial do FullScan Accessibility Checker. Fonte: Elaborado pelo autor, 2024	32
3.3	Tela de resultado de análise de páginas. Fonte: Elaborado pelo autor, 2024	33
3.4	Menu <i>dropdown</i> de seleção de páginas. Fonte: Elaborado pelo autor, 2024	34
3.5	Relatório de acessibilidade por página. Fonte: Elaborado pelo autor, 2024	34
4.1	Página inicial da plataforma MuseusBR. Fonte: https://cadastro.museus.gov.br/	36
4.2	Resultados de busca por tipo. Fonte: https://cadastro.museus.gov.br/	37
4.3	Página de um museu na plataforma MuseusBR. Fonte: https://cadastro.museus.gov.br/	37
4.4	Porcentagem dos resultados por nível de acessibilidade. Fonte: Elaborado pelo autor, 2024	41
4.5	Resultados por média de acessibilidade. Fonte: Elaborado pelo autor, 2024	41
4.6	Distribuição de falhas por critério de sucesso. Fonte: Elaborado pelo autor, 2024	42
4.7	Distribuição de elementos por falhas. Fonte: Elaborado pelo autor, 2024	43

Lista de Tabelas

2.1	Critérios de Sucesso para a diretriz 1.2	9
2.2	Critérios de Sucesso para a diretriz 1.3	10
2.3	Critérios de Sucesso para a diretriz 1.4	11
2.4	Critérios de Sucesso para a diretriz 2.1	12
2.5	Critérios de Sucesso para a diretriz 2.2	14
2.6	Critérios de Sucesso para a diretriz 2.3	15
2.7	Critérios de Sucesso para a diretriz 2.4	16
2.8	Critérios de Sucesso para a diretriz 2.5	17
2.9	Critérios de Sucesso para a diretriz 3.1	18
2.10	Critérios de Sucesso para a diretriz 3.2	19
2.11	Critérios de Sucesso para a diretriz 3.3	19
2.12	Critérios de Sucesso para a diretriz 4.1	21
4.1	Resultados da coleta dos links de museus após pré-processamento	38
4.2	Resultados da análise de acessibilidade de museus virtuais	40

1

Introdução

1.1 Motivação

Com o avanço da tecnologia e a crescente digitalização de serviços e conteúdos em áreas como educação, comércio, notícias, saúde e entretenimento, a acessibilidade digital tornou-se um elemento indispensável para promover a inclusão social ([Abuaddous et al. \(2016\)](#)). Pessoas com deficiência enfrentam desafios significativos no acesso a informações e recursos na internet devido à falta de conformidade de muitos *websites* com as diretrizes internacionais de acessibilidade, como as *Web Content Accessibility Guidelines* (WCAG) ([W3C \(2023\)](#)). Essa situação limita a plena participação de milhões de usuários na sociedade digital, comprometendo seus direitos básicos de acesso à informação e cultura previstos na declaração universal dos direitos humanos([ONU \(1948\)](#)).

Para ajudar a identificar os problemas de acessibilidade contidos nos *websites*, várias ferramentas de verificação de acessibilidade foram surgindo com o tempo, [Abu Doush et al. \(2023\)](#) catalogaram algumas, tais como *AChecker* ([Gay and Li \(2010\)](#)), *WAVE* ([WebAIM \(2021\)](#)), *TAW* ([CTIC \(2021\)](#)) etc. e notaram que muitas delas são limitadas em termos de funcionalidade ou estão restritas por altos custos, o que dificulta sua utilização por organizações menores ou sem fins lucrativos, como museus e instituições culturais. [Abu Doush et al. \(2023\)](#) explicam ainda que grande parte dessas ferramentas analisa apenas páginas individuais ou um número restrito de URLs, o que não fornece uma visão abrangente da acessibilidade de *websites* mais complexos e extensos.

Esse cenário evidencia a necessidade de uma solução computacional voltada para a verificação da acessibilidade web de forma gratuita, acessível e abrangente o suficiente para realizar análises de múltiplas páginas de forma automatizada, promovendo um diagnóstico completo do nível de acessibilidade de um *website*.

Diante disso, este trabalho descreve o desenvolvimento da ferramenta *FullScan Accessibility*

Checker, uma aplicação gratuita projetada para realizar análises abrangentes da acessibilidade de *websites*.

1.2 Objetivos

1.2.1 Objetivo Geral

Implementar a automatização do processo de análise de acessibilidade web a partir do desenvolvimento de uma ferramenta gratuita capaz de verificar a conformidade de todas as páginas disponíveis em um *website*, apresentando uma visão extensiva acerca de sua acessibilidade.

1.2.2 Objetivos Específicos

- Estudar os conceitos de acessibilidade web;
- Conhecer as características e compreender o funcionamento dos verificadores de acessibilidade *web* gratuitos;
- Desenvolver uma solução para automatizar o processo de verificação de acessibilidade em *websites*;
- Obter a validação da solução desenvolvida a partir de um estudo de caso sobre a análise da acessibilidade de *websites* de museus virtuais brasileiros.

1.3 Metodologia

Inicialmente, uma revisão bibliográfica foi feita visando compreender o estado da arte sobre as técnicas e tecnologias utilizadas para a avaliação de acessibilidade web. A busca foi realizada a partir da plataforma *Google Scholar* que reúne e agrega artigos, teses, livros e revistas na esfera acadêmica. Posteriormente, a busca foi ampliada utilizando a lista da W3C(*World Wide Web Consortium*)¹, que reúne uma quantidade significativa de verificadores e o buscador padrão do *Google* a fim de se verificar a existência de avaliadores de acessibilidade não mencionados em documentos acadêmicos. Para direcionar a busca, palavras-chave foram utilizadas tais como **Web Accessibility** (Acessibilidade web), **Accessibility checker** (Verificador de Acessibilidade), **Automatic evaluations accessibility tools** (Ferramentas de avaliação de acessibilidade automáticas), **webcrawling** e **WCAG**. A partir dos trabalhos encontrados na revisão, foram identificados os principais verificadores de acessibilidade disponíveis a fim de selecionar um deles para ser automatizado e servir de base para o desenvolvimento do *FullScan Accessibility Checker*. Os critérios utilizados para a escolha do verificador foram: ser gratuito;

¹<https://www.w3.org/WAI/test-evaluate/tools/list/>

basear-se nas diretrizes WCAG; fornecer um relatório detalhado acerca da conformidade ou não de uma página *web* com as diretrizes WCAG e atender a características técnicas para integrar-se à nossa aplicação.

Após a escolha do verificador, foi realizada uma análise detalhada em sua estrutura a fim de se entender as características de funcionamento do processo de verificação de conformidade, considerando quais tipos de entrada e saída o verificador aceita, bem como se seria possível coordenar uma automatização das análises de diferentes páginas *web*.

Com a análise do verificador feita, iniciou-se o processo de desenvolvimento da ferramenta *FullScan Accessibility Checker*. Tal processo foi dividido em três etapas: o desenvolvimento do mecanismo de busca e coleta das páginas de um *website* via *webcrawling*, o desenvolvimento do processo de interação da ferramenta com o verificador de acessibilidade escolhido para a realização das análises e coleta de resultados e, finalmente, o desenvolvimento da interface de interação com o usuário. A coordenação do mecanismo de busca e a interação com o verificador de acessibilidade foram realizadas através de um *webdriver* automatizado e o desenvolvimento da interface do usuário através de um *framework* voltado para visualização de dados.

Para o processo de busca e coleta de páginas, considerou-se a estrutura de um *website* como a estrutura de um grafo, desta forma, a página inicial representa o nó raiz da estrutura, assim como os links para as páginas representam as arestas que levam às demais páginas, que por sua vez, representam os nós filhos, também conhecidos como "vizinhos". Tendo em vista esta estrutura, a coleta das páginas foi realizada utilizando o método *Breadth-First Search* (Busca em Largura), que realiza as buscas priorizando os nós vizinhos do nó em questão, desta forma, obtém-se um maior número de *links* por página visitada.

A fim de validar o funcionamento do *FullScan Accessibility Checker* foi realizado um experimento com *websites* de museus virtuais do Brasil. Os museus virtuais foram selecionados a partir de uma pesquisa realizada na plataforma *MuseusBR*², que mantém um catálogo nacional com informações de museus brasileiros. A pesquisa por museus virtuais foi realizada através dos filtros de busca "tipo de museu" marcado como "virtual" e "status" marcado como "aberto".

1.4 Trabalhos Relacionados

Em SOARES (2023) é apresentada a ferramenta *Access Crawler*, um verificador de acessibilidade que analisa todas as páginas de um *website*. *Access Crawler* combina as funcionalidades de um *web crawler* e do verificador gratuito ASES³ que avalia a conformidade de uma página de um *website* com as diretrizes eMAG DGE (2014) criadas pelo governo brasileiro. Apesar de ter objetivo bastante similar a este trabalho, a ferramenta *Access Crawler* apresenta limitações importantes. Por exemplo, a ferramenta não disponibiliza nenhum tipo de interface de interação

²<https://cadastro.museus.gov.br/>

³<https://asesweb.governoeletronico.gov.br/>

com o usuário, seja na entrada de dados ou na exibição dos resultados. Tal fato dificulta a análise de *websites* diferentes visto que a entrada para a URL do *website* a ser analisado encontra-se como *hard-coded* definida para analisar o site da Universidade Federal de Ouro Preto. Os resultados de análise do *Access Crawler* são apresentados em um arquivo no formato .txt. Além disso, a ferramenta apenas analisa a conformidade com as diretrizes eMAG, que teve sua última atualização em 2014, não dispondo aos usuários a verificação de conformidade com as diretrizes WCAG. Tal fato pode ser considerado uma lacuna significativa visto que as diretrizes WCAG representam um padrão internacional de acessibilidade web amplamente reconhecido. Os autores informam que o processo de coleta das páginas é realizado em *threads*, paralelizando a busca e tornando-a mais eficiente. A partir do teste realizado no site da Universidade Federal de Ouro Preto, *Access Crawler* identificou 8808 páginas, das quais 7607 foram avaliadas com sucesso pela ferramenta ASES.

1.5 Organização do Documento

O restante do documento encontra-se organizado da seguinte forma: o Capítulo 2 apresenta os tópicos relacionados à acessibilidade web, necessários para o entendimento e o desenvolvimento deste trabalho; o Capítulo 3 descreve as etapas de desenvolvimento de *FullScan Accessibility Checker*, bem como o funcionamento de seus componentes essenciais; o Capítulo 4 apresenta um estudo de caso que analisa a acessibilidade de museus virtuais utilizando a ferramenta *FullScan Accessibility Checker*; por fim, o Capítulo 5 abrange as conclusões e trabalhos futuros.

2

Fundamentação Teórica

2.1 Acessibilidade *web*

Segundo o [W3C \(2005b\)](#) (*World Wide web Consortium*), a acessibilidade *web* implica que *websites*, ferramentas e tecnologias deverão ser pensadas e desenvolvidas de maneira que pessoas com algum tipo de deficiência possam utilizá-las. Mais especificamente, permite que essas pessoas possam entender, perceber, navegar e interagir na *web* além de contribuir para a mesma. [W3C \(2005b\)](#) ainda explica que a acessibilidade *web* deve contemplar todo e qualquer tipo de deficiência que afete o acesso a *web*, incluindo as deficiências auditivas, cognitivas, neurológicas, físicas, visuais e de fala. [Cusin et al. \(2009\)](#) definem acessibilidade *web* como um fator propulsor da inclusão digital, promovendo assim, igualdade de acesso a usuários com necessidades especiais, tal inclusão não se limita apenas ao acesso as informações mas abrange a aquisição de conhecimentos, competências e habilidades.

O fator social que justifica a acessibilidade *web* se embasa na crença de que todos têm o direito ao acesso, um bom design de acessibilidade é um bom design para todos ([Brophy and Craven \(2007\)](#)). Um experimento realizado por [Vollenwyder et al. \(2023\)](#) sustenta essa ideia, nele, um grupo de 131 pessoas (66 com alguma deficiência visual, 65 sem deficiência) realizaram uma série de tarefas em duas versões de um *website* de compras online, uma versão com baixa acessibilidade e outra versão com um bom nível de acessibilidade. O estudo conclui que, apesar de não se observar diferenças estatisticamente significativas relacionadas à usabilidade, os usuários com alguma deficiência visual tiveram experiências mais positivas em relação à versão acessível do *website* enquanto que os usuários sem deficiência tiveram quase nenhuma ou nenhuma experiência negativa.

Partindo para o fator econômico, há uma crença de que manter ou tornar um *website* acessível é mais custoso que manter um *website* sem estes cuidados. [Sloan \(2008\)](#) demonstra uma visão contrária a esta crença ao trazer um levantamento realizado por uma empresa britânica de

finanças. No ano de 2006, o *website* desta empresa foi reestruturado para atender aos padrões de acessibilidade recém estabelecidos para o país, após a reestruturação, a empresa apresentou uma redução de 200 mil euros anuais gastos em manutenção, um aumento de 30% no tráfego do *website*, um retorno de 100% no investimento da reestruturação em menos de 12 meses entre outros benefícios que contemplaram não somente a empresa como também os usuários.

Em termos de legislação, é notório que a maioria dos países do mundo tem interesse em garantir o acesso a informação de forma indiscriminada aos seus cidadãos, portanto, melhorar a experiência do usuário com deficiência é um fator chave na elaboração e ênfase de leis (Shinyong Jung and Lee (2024)). No Brasil, a convenção sobre os direitos das pessoas com deficiência e o estatuto da pessoa com deficiência são exemplos de instrumentos legislativos que asseguram o acesso igualitário e reforçam a ideia da acessibilidade *web* (Brasil (2009); Brasil (2015)), nos Estados Unidos destaca-se a lei de direitos civis *Americans with Disabilities act* (ADA) (USA (1990)) enquanto que na Austrália encontra-se o decreto *Disability Discrimination Act* (Australia (1992)). Cada país tem a sua legislação e diretrizes a serem seguidas relacionadas a acessibilidade, porém, existe um modelo internacional de diretrizes que serão discutidas na próxima subseção.

2.1.1 Diretrizes WCAG

Tim Berners-lee, criador da internet, pouco tempo após inventá-la fundou o W3C, que por sua vez criou a Iniciativa de Acessibilidade *web* (WAI - *Web Accessibility Initiative*). Tal iniciativa tem o objetivo de promover e alcançar a funcionalidade da *web* para pessoas com deficiências, para *tim é crucial que a web seja utilizável por qualquer pessoa, independente de suas capacidades individuais e deficiências* (W3C (1997)). Para alcançar tal objetivo, a WAI publicou as Diretrizes de Acessibilidade para o Conteúdo *web* (WCAG - *web Content Accessibility guidelines*) com o intuito de padronizar o conteúdo *web* de maneira a atender às necessidades de indivíduos, organizações e governos internacionalmente (W3C (2005a)).

Segundo a W3C (2005a) as diretrizes foram desenvolvidas como um padrão técnico, ou seja, são destinadas a desenvolvedores *web* (criadores de páginas *web*, designers de *sites*), desenvolvedores de ferramentas de avaliação de acessibilidade e qualquer um que necessite de um padrão de acessibilidade *web*, incluindo acessibilidade *mobile*.

As diretrizes de acessibilidade definem a forma de como tornar o conteúdo da web mais acessível para pessoas com deficiência. Estas diretrizes tornam também o conteúdo da web mais acessível por pessoas idosas, cujas habilidades estão em constante mudança devido ao envelhecimento, e muitas vezes melhoram a usabilidade para usuários em geral (W3C (2018c)).

A WCAG foi evoluindo ao longo do tempo para se adequar as novas tecnologias e aos novos padrões *web*, desde sua primeira publicação até os dias atuais tivemos várias versões dessas diretrizes como a WCAG 1.0 W3C (1999), WCAG 2.0 W3C (2008), WCAG 2.1 W3C (2018b), WCAG 2.2 W3C (2023) e atualmente está sendo desenvolvida a WCAG 3.0 W3C

(2024). Todas essas versões são "retrocompatíveis", ou seja, o conteúdo de uma versão atual engloba o conteúdo de suas versões anteriores, um conteúdo que está conforme a WCAG 2.2 também estará conforme a WCAG 2.1 e assim por diante W3C (2005a). Para este trabalho foi considerada a versão 2.1 da WCAG devido a escolha do verificador de acessibilidade que será discutido na seção 2.2.1.

As diretrizes WCAG se baseiam em 4 princípios: *perceptível*, *operável*, *compreensível* e *robusto*. Para cada diretriz são fornecidos critérios de sucesso que determinarão a conformidade do conteúdo *web* com o padrão estabelecido, tais critérios de sucesso estão divididos em 3 níveis de conformidade: A(o mais baixo), AA e AAA(o mais elevado). Por fim, para cada diretriz são fornecidas técnicas que podem ser necessárias ou sugeridas para que se satisfaçam os critérios de sucesso atrelados às diretrizes(W3C (2018c)).

Princípios

Nesta subseção serão discutidos os 4 princípios das diretrizes WCAG bem como os critérios de sucesso e os níveis de conformidade das mesmas.

1. Perceptível

O primeiro princípio determina que as informações e os componentes da interface do usuário devem ser apresentados em formas que possam ser percebidas pelo usuário. Este princípio engloba 4 diretrizes: Alternativas em texto, Mídias com base em tempo, Adaptável e Discernível.

Diretriz 1.1 Alternativas em Texto

Fornecer alternativas textuais para qualquer conteúdo não textual, para que possa ser transformado em outras formas de acordo com a necessidade dos usuários, tais como impressão com tamanho de fontes maiores, braille, fala, símbolos ou linguagem mais simples.

Esta diretriz possui apenas um critério de sucesso, **Critério de Sucesso 1.1.1 Conteúdo não textual**, nível A de conformidade e diz que todo o conteúdo não textual que é exibido ao usuário deverá possuir uma alternativa textual que serve um propósito equivalente exceto em algumas situações específicas indicadas a seguir:

- **Controles, Entrada:** Se o conteúdo não textual for um controle ou aceitar a entrada de dados por parte do usuário, então esse conteúdo não textual possui um nome que descreve a sua finalidade.
- **Mídias com base no tempo:** Se o conteúdo não textual consiste em mídia baseada em tempo, então as alternativas textuais fornecem, no mínimo, uma identificação descritiva do conteúdo não textual.

- **Teste:** Se o conteúdo não textual for um teste ou um exercício, que ficaria inválido se fosse apresentado em texto, então as alternativas textuais fornecem, no mínimo, uma identificação descritiva do conteúdo não textual.
- **Sensorial:** Se a finalidade do conteúdo não textual for, essencialmente, criar uma experiência sensorial específica, então as alternativas textuais fornecem, no mínimo, uma identificação descritiva do conteúdo não textual.
- **CAPTCHA:** e a finalidade do conteúdo não textual for confirmar que o conteúdo está sendo acessado por uma pessoa e não por um computador, então devem ser fornecidas alternativas textuais que identificam e descrevem a finalidade do conteúdo não textual. Formas alternativas de CAPTCHA, que utilizam modos de saída para diferentes tipos de percepção sensorial, devem ser apresentadas para atender diferentes deficiências.
- **Decoração, Formatação, Invisível:** Se o conteúdo não textual for meramente decorativo, se for utilizado apenas para formatação visual, ou se não for exibido aos usuários, então esse conteúdo não textual deve ser implementado de uma forma que possa ser ignorado pelas tecnologias assistivas.

Para fins de demonstração, serão apresentados 3 critérios de sucesso por diretriz (quando possível, um para cada nível de conformidade). As tabelas com todos os critérios de sucesso estão disponíveis na seção [A](#) dos apêndices.

Diretriz 1.2 Mídias com base em tempo

Fornecer alternativas para mídias baseadas em tempo. A tabela [2.1](#) a seguir apresenta os critérios de sucesso bem como o nível de conformidade dos mesmos para esta diretriz.

Tabela 2.1: Critérios de Sucesso para a diretriz 1.2

Critério de Sucesso	Descrição	Nível de conformidade
1.2.1 Apenas Áudio e Apenas Vídeo (Pré-gravado)	Apenas áudio pré-gravado: É fornecida uma alternativa para mídia com base em tempo que apresenta informação equivalente para o conteúdo composto por apenas áudio pré-gravado; Apenas vídeo pré-gravado: É fornecida uma alternativa em mídia com base em tempo ou uma faixa de áudio que apresenta informação equivalente para o conteúdo apenas de vídeo pré-gravado.	A
1.2.4 Legendas (Ao Vivo)	São fornecidas legendas para todo o conteúdo do áudio ao vivo existente em mídia sincronizada.	AA
1.2.6 Língua de sinais (Pré-gravada)	É fornecida interpretação em língua de sinais para todo o conteúdo de áudio pré-gravado existente em um conteúdo em mídia sincronizada.	AAA

Diretriz 1.3 Adaptável

Criar conteúdo que pode ser apresentado de diferentes maneiras (por exemplo um layout simplificado) sem perder informação ou estrutura. A tabela 2.2 a seguir apresenta os critérios de sucesso bem como o nível de conformidade dos mesmos para esta diretriz.

Tabela 2.2: Critérios de Sucesso para a diretriz 1.3

Critério de Sucesso	Descrição	Nível de conformidade
1.3.1 Informações e Relações	As informações, a estrutura, e os relacionamentos transmitidos através de apresentação podem ser determinados por meio de código de programação ou estão disponíveis no texto.	A
1.3.5 Identificar o Objetivo de Entrada	A finalidade de cada campo de entrada que coleta informações sobre o usuário pode ser determinada de forma programática quando: O campo de entrada serve à finalidade identificada na seção Finalidades de Entrada para Componentes de Interface de Usuário; O conteúdo é implementado por meio da utilização de tecnologias com suporte para identificar o significado esperado para os dados de entrada do formulário.	AA
1.3.6 Identificar o Objetivo	Em conteúdo implementado que utiliza linguagens de marcação, a finalidade dos Componentes de Interface de Usuário, ícones, e regiões podem ser determinados programaticamente.	AAA

Diretriz 1.4 Discernível

Facilitar a audição e a visualização de conteúdo aos usuários, incluindo a separação entre o primeiro plano e o plano de fundo. A tabela 2.3 a seguir apresenta os critérios de sucesso bem como o nível de conformidade dos mesmos para esta diretriz.

Tabela 2.3: Critérios de Sucesso para a diretriz 1.4

Critério de Sucesso	Descrição	Nível de conformidade
1.4.1 Utilização de Cores	A cor não é utilizada como o único meio visual de transmitir informações, indicar uma ação, pedir uma resposta ou distinguir um elemento visual.	A
1.4.3 Contraste Mínimo	A apresentação visual de texto e imagens de texto tem uma relação de contraste de, no mínimo, 4.5:1, exceto para o seguinte: Texto Ampliado-Texto em tamanho grande e as imagens compostas por texto em tamanho grande têm uma relação de contraste de, no mínimo, 3:1; Texto em plano Secundário-O texto ou imagens de texto que fazem parte de um componente de interface de usuário inativo, que são meramente decorativos, que não estão visíveis para ninguém, ou que são parte de uma imagem que inclui outro conteúdo visual significativo, não têm requisito de contraste; Logotipos - O texto que faz parte de um logotipo ou marca comercial não tem requisito mínimo de contraste.	AA
1.4.7 Áudio de fundo baixo ou sem Áudio de fundo	Para conteúdo composto por apenas áudio pré-gravado que (1) contenha essencialmente fala em primeiro plano, (2) não seja um CAPTCHA de áudio ou logotipo de áudio, e (3) não seja vocalização com o objetivo de ser, essencialmente, expressão musical, tal como cantar ou fazer batidas, no mínimo, uma das seguintes afirmações é verdadeira: Sem Fundo-O áudio não contém sons de fundo; Desligar - Os sons de fundo podem ser desligados; 20 dB - Os sons de fundo são, no mínimo, 20 decibéis mais baixos que o conteúdo da voz em primeiro plano, com a exceção de sons ocasionais que duram apenas um ou dois segundos.	AAA

2. Operável

O segundo princípio determina que todos os componentes de interface de usuário e a navegação devem ser operáveis, ou seja, o usuário deve ser capaz de navegar, inserir dados e interagir com o conteúdo *web*. Este princípio é composto por 5 diretrizes: Acessível por teclado, Tempo Suficiente, Convulsões e Reações Físicas, Navegável e Modalidades de Entrada.

Diretriz 2.1 Acessível por Teclado

Fazer com que toda funcionalidade fique disponível a partir de um teclado. A tabela 2.4 a seguir apresenta os critérios de sucesso, bem como o nível de conformidade dos mesmos para esta diretriz.

Tabela 2.4: Critérios de Sucesso para a diretriz 2.1

Critério de Sucesso	Descrição	Nível de conformidade
2.1.1 Teclado	Toda a funcionalidade do conteúdo é operável através de uma interface de teclado sem requerer temporizações específicas para digitação individual, exceto quando a função subjacente requer entrada de dados que dependa da cadeia de movimento do usuário e não apenas dos pontos finais.	A
2.1.2 Sem Bloqueio do Teclado	Se o foco do teclado puder ser movido para um componente da página utilizando uma interface de teclado, então o foco pode ser retirado desse componente utilizando apenas uma interface de teclado e, se for necessário mais do que as setas do cursor ou tabulação ou outros métodos de saída normalmente utilizados, o usuário deve ser informado sobre o método para retirar o foco.	A
2.1.3 Teclado Sem Exceção	Toda a funcionalidade do conteúdo é operável através de uma interface de teclado sem requerer temporizações específicas para digitação individual.	AAA

2.1.4 Atalhos de teclado por caractere	<p>Se um atalho de teclado é implementado no conteúdo utilizando apenas letras (incluindo letras maiúsculas e minúsculas), pontuação, números ou símbolos, então ao menos um dos itens é verdadeiro: Desativar - Um mecanismo está disponível para desativar o atalho; Remapear - Um mecanismo está disponível para remapear o atalho para utilizar um ou mais caracteres de teclado não imprimíveis (por exemplo, Ctrl, Alt etc); Ativo apenas quando recebe o foco - O atalho do teclado para um componente de interface de usuário está ativo apenas quando esse componente tem foco.</p>	A
--	--	---

Diretriz 2.2 Tempo Suficiente

Fornecer aos usuários tempo suficiente para ler e utilizar o conteúdo. A tabela 2.5 a seguir apresenta os critérios de sucesso, bem como o nível de conformidade dos mesmos para esta diretriz.

Tabela 2.5: Critérios de Sucesso para a diretriz 2.2

Critério de Sucesso	Descrição	Nível de conformidade
2.2.1 Ajustável por Temporização	Para cada limite de tempo definido pelo conteúdo, no mínimo, uma das seguintes afirmações é verdadeira: Desligar - O usuário pode desligar o limite de tempo antes de atingi-lo; ou Ajustar - O usuário pode ajustar o limite de tempo antes de atingi-lo, num intervalo de, no mínimo, dez vezes mais do que a configuração padrão; ou Prolongar - O usuário é avisado antes de o tempo expirar e tem, no mínimo, 20 segundos para prolongar o limite de tempo com uma simples ação (por exemplo, "pressionar a barra de espaços"), e o usuário pode prolongar o limite de tempo, no mínimo, dez vezes; ou Exceção em Tempo Real - O limite de tempo é uma parte necessária de um evento em tempo real (por exemplo, um leilão), e não é possível nenhuma alternativa ao limite de tempo; ou Exceção Essencial - O limite de tempo é essencial e prolongá-lo invalidaria a atividade; ou Exceção de 20 Horas - O limite de tempo é superior a 20 horas.	A
2.2.3 Sem Temporização	A temporização não é uma parte essencial do evento ou da atividade apresentada pelo conteúdo, exceto para mídia sincronizada não interativa e eventos em tempo real.	AAA

2.2.6 Limites de Tempo	Os usuários são avisados sobre a duração de qualquer inatividade do usuário que possa causar perda de dados, a menos que os dados sejam preservados quando o usuário não realizar nenhuma ação por mais de 20 horas.	AAA
------------------------	--	-----

Diretriz 2.3 Convulsões e Reações Físicas

Não criar conteúdo de uma forma conhecida por causar convulsões e reações físicas. A tabela 2.6 a seguir apresenta os critérios de sucesso, bem como o nível de conformidade dos mesmos para esta diretriz.

Tabela 2.6: Critérios de Sucesso para a diretriz 2.3

Critério de Sucesso	Descrição	Nível de conformidade
2.3.1 Três Flashes ou Abaixo do Limite	As páginas <i>web</i> não incluem nenhum conteúdo que pisque mais de três vezes no período de um segundo, ou o flash encontra-se abaixo dos limites de flash universal e flash vermelho.	A
2.3.2 Três Flashes	As páginas <i>web</i> não incluem qualquer conteúdo que pisca mais de três vezes no período de um segundo.	AAA
2.3.3 Animação de Interações	A animação de movimento acionada por interação pode ser desativada, a menos que a animação seja essencial para a funcionalidade ou para as informações transmitidas.	AAA

Diretriz 2.4 Navegável

Fornecer maneiras de ajudar os usuários a navegar, localizar conteúdos e determinar onde se encontram. A tabela 2.7 a seguir apresenta os critérios de sucesso, bem como o nível de conformidade dos mesmos para esta diretriz.

Tabela 2.7: Critérios de Sucesso para a diretriz 2.4

Critério de Sucesso	Descrição	Nível de conformidade
2.4.1 Ignorar Blocos	Um mecanismo está disponível para ignorar blocos de conteúdo que são repetidos em várias páginas <i>web</i> .	A
2.4.8 Localização	Informação sobre a localização do usuário está disponível em um conjunto de páginas <i>web</i> .	AAA
2.4.10 Cabeçalhos da Sessão	Os cabeçalhos da seção são utilizados para organizar o conteúdo.	AAA

Diretriz 2.5 Modalidades de Entrada

Torna mais fácil para os usuários operar a funcionalidade por meio de várias entradas além do teclado. A tabela 2.8 a seguir apresenta os critérios de sucesso, bem como o nível de conformidade dos mesmos para esta diretriz.

Tabela 2.8: Critérios de Sucesso para a diretriz 2.5

Critério de Sucesso	Descrição	Nível de conformidade
2.5.1 Gestos de Acionamento	Todas as funcionalidades que utilizam gestos multiponto ou baseados em caminhos para operação podem ser operadas com um único ponteiro sem um gesto baseado em caminho, a menos que um gesto multiponto ou baseado em caminho seja essencial.	A
2.5.5 Tamanho da Área Clicável	O tamanho do alvo para entradas de ponteiro é pelo menos 44 por 44 pixels CSS exceto quando: Equivalente - O alvo está disponível por meio de um link ou controle equivalente na mesma página, com pelo menos 44 por 44 pixels CSS; Inline - O alvo está em uma sentença ou bloco de texto; Controle de Agente de Usuário - O Tamanho da Área Clicável é determinado pelo agente de usuário e não é modificado pelo autor; Essencial - Uma apresentação específica do alvo é essencial à informação que está sendo transmitida.	AAA
2.5.6 Mecanismos de Entrada Simultâneos	O conteúdo da <i>web</i> não restringe o uso de modalidades de entrada disponíveis em uma plataforma, exceto quando a restrição é essencial, necessária para garantir a segurança do conteúdo ou obrigada a respeitar as configurações do usuário.	AAA

3. Compreensível

O terceiro princípio determina que a informação e a operação da interface de usuário devem ser compreensíveis. Este princípio é composto por 3 diretrizes: Legível, Previsível e Assistência de Entrada.

Diretriz 3.1 Legível

Tornar o conteúdo do texto legível e compreensível. A tabela 2.9 a seguir apresenta os critérios de sucesso, bem como o nível de conformidade dos mesmos para esta diretriz.

Tabela 2.9: Critérios de Sucesso para a diretriz 3.1

Critério de Sucesso	Descrição	Nível de conformidade
3.1.1 Idioma da Página	O idioma humano pré-definido de cada página <i>web</i> pode ser determinado por meio de código de programação.	A
3.1.2 Idioma das Partes	O idioma de cada passagem ou frase no conteúdo pode ser determinado por meio de código de programação exceto para nomes próprios, termos técnicos, palavras de idioma indeterminado e palavras ou frases que se tornaram parte do vernáculo do texto que as envolve.	AA
3.1.5 Nível de Leitura	Quando o texto exigir uma capacidade de leitura mais avançada do que o nível de educação secundário inferior (equivalente no Brasil aos últimos anos do ensino fundamental), após a remoção dos nomes próprios e títulos adequados, um conteúdo suplementar, ou uma versão que não exija uma capacidade de leitura mais avançada do que o nível de educação secundário inferior (equivalente no Brasil aos últimos anos do ensino fundamental) está disponível.	AAA

Diretriz 3.2 Previsível

Fazer com que as páginas web apareçam e funcionem de modo previsível. A tabela 2.10 a seguir apresenta os critérios de sucesso, bem como o nível de conformidade dos mesmos para esta diretriz.

Tabela 2.10: Critérios de Sucesso para a diretriz 3.2

Critério de Sucesso	Descrição	Nível de conformidade
3.2.1 Em Foco	Quando qualquer componente de interface do usuário recebe o foco, não inicia uma alteração de contexto.	A
3.2.4 Identificação Consistente	Os componentes que têm a mesma funcionalidade em um conjunto de páginas <i>web</i> são identificados de forma consistente.	AA
3.2.5 Alteração Mediante Solicitação	As alterações de contexto são iniciadas apenas a pedido do usuário, ou um mecanismo para desativar essas alterações está disponível.	AAA

Diretriz 3.3 Assistência de Entrada

Ajudar os usuários a evitar e corrigir erros. A tabela 2.11 a seguir apresenta os critérios de sucesso bem como o nível de conformidade dos mesmos para esta diretriz.

Tabela 2.11: Critérios de Sucesso para a diretriz 3.3

Critério de Sucesso	Descrição	Nível de conformidade
3.3.1 Identificação do Erro	Se um erro de entrada for automaticamente detectado, o item que apresenta erro é identificado e o erro é descrito para o usuário em texto.	A

3.3.3 Sugestão de Erro	Se um erro de entrada for automaticamente detectado e forem conhecidas sugestões de correção, então as sugestões são fornecidas ao usuário, a menos que coloque em risco a segurança ou o propósito do conteúdo.	AA
3.3.6 Prevenção de Erros (Todos)	Para páginas <i>web</i> que exijam que o usuário envie informações, no mínimo, uma das seguintes afirmações é verdadeira: Reversível - As submissões são reversíveis. Verificado - Os dados introduzidos pelo usuário são verificados quanto à existência de erros de entrada e é oferecida ao usuário uma oportunidade de corrigi-los. Confirmado - Um mecanismo está disponível para rever, confirmar e corrigir as informações antes de finalizar o envio.	AAA

4. Robusto

O quarto e último princípio determina que o conteúdo deve ser robusto o suficiente para ser interpretado de forma confiável por uma ampla variedade de agentes de usuário, incluindo tecnologias assistivas. Este princípio é composto por uma única diretriz: Compatível.

Diretriz 4.1 Compatível

Maximizar a compatibilidade entre os atuais e futuros agentes de usuário, incluindo tecnologias assistivas. A tabela 2.12 a seguir apresenta os critérios de sucesso, bem como o nível de conformidade dos mesmos para esta diretriz.

Tabela 2.12: Critérios de Sucesso para a diretriz 4.1

Critério de Sucesso	Descrição	Nível de conformidade
4.1.1 Análise	No conteúdo implementado utilizando linguagens de marcação, os elementos dispõem de tags completas de início e de fim, os elementos são aninhados de acordo com as respectivas especificações, os elementos não contêm atributos duplicados, e quaisquer IDs são exclusivos, exceto quando as especificações permitem estas características.	A
4.1.2 Nome, Função, Valor	Para todos os componentes de interface de usuário (incluindo, mas não se limitando a: elementos de formulário, links e componentes gerados por scripts), o nome e a função podem ser determinados por meio de código de programação; os estados, as propriedades e os valores, que possam ser definidos pelo usuário, podem ser definidos por meio de código de programação; e a notificação sobre alterações destes itens está disponível para os agentes de usuário, incluindo as tecnologias assistivas.	A
4.1.3 Mensagens de Status	Em conteúdo implementado que utiliza linguagens de marcação, as mensagens de status podem ser determinado por meio de código de programação por meio da função ou de propriedades, de modo que possam ser apresentadas ao usuário por tecnologias assistivas sem receber foco.	AA

2.2 Verificadores de Acessibilidade

A análise de acessibilidade *web* pode ser definida como uma avaliação do quão bem um *website* pode ser utilizado por usuários com deficiências e é um processo essencial para verificar se os padrões de acessibilidade adotados foram cumpridos. Tal processo envolve uma gama de aspectos relacionados a tecnologias, diretrizes, padrões e ferramentas de avaliação, por isso, diversos Métodos de Avaliação de Acessibilidade (MAAs) são utilizados e estão divididos em 3 categorias: Testes automatizados, Inspeção manual e Testes de usuário (Abascal et al. (2019)). A taxonomia dos Métodos de Avaliação de Acessibilidade é descrita na figura 2.1 abaixo.

Figura 2.1: Taxonomia dos métodos de avaliação de acessibilidade. Fonte: adaptada de [Brajnik \(2008\)](#)

corretamente, fornecem grande suporte aos desenvolvedores *web* e reduzem significativamente o tempo e o esforço requerido para se realizar uma avaliação.

Seguindo a taxonomia de [Brajnik \(2008\)](#), os verificadores produzem resultados de saída que podem ser qualitativos e quantitativos e que podem consistir de modos de falha, defeitos, recomendações de design ou qualquer combinação das três opções. Com relação ao propósito, podem ser tanto formativos (explorativos e diagnósticos) quanto sumativos, focando na parte avaliativa e validativa. A natureza será sempre analítica e a entrada baseada em medidas de performance.

Devido a essa variedade de formas que um verificador de acessibilidade pode apresentar, [Manca et al. \(2023\)](#) definiram critérios de transparência que um verificador deve possuir para que fique mais claro aos usuários quais ferramentas se encaixam melhor com suas necessidades. Tais critérios são:

- **Quais padrões, critérios de sucesso e técnicas são seguidos?**

Este é um ponto muito importante pois evidencia o motivo pelo qual diferentes ferramentas exibem diferentes resultados, além disso, quanto mais técnicas uma ferramenta dispor, mais completos são os resultados, diferentes técnicas revelam diferentes problemas de acessibilidade. As diretrizes WCAG 2.1 são compostas por mais de 70 critérios de sucesso que foram aumentando com o tempo, alguns deles não podem ser validados automaticamente, portanto, os usuários precisam entender qual a cobertura dessas diretrizes para as diferentes ferramentas.

- **Como os problemas de acessibilidade são categorizados?**

A classificação da validação dos resultados de acessibilidade pelo padrão de avaliação e relatório da W3C (EARL - *Evaluation and Report Language*) [W3C \(2018a\)](#) recomenda utilizar uma das seguintes categorias para indicar os resultados dos testes: passou, falhou, indefinido, inaplicável e não testado. Quanto mais uma ferramenta utiliza esse padrão para a classificação dos problemas de acessibilidade, mais compreensível serão os resultados para os usuários. Se uma ferramenta utiliza termos diferentes para categorizar seus resultados, sua explicação deverá ser clara e facilmente disponível para os usuários juntamente de como a ferramenta se refere à categorização padrão.

- **Como os resultados da validação são fornecidos pela ferramenta?**

Para este propósito é importante que tal informação seja fornecida com níveis de granularidade variados e utilizando diferentes tipos de apresentação.

- **Granularidade:** A ferramenta deve ser capaz de fornecer indicadores de acessibilidade para elementos específicos mas também medidas de acessibilidade gerais para páginas ou sites inteiros. Além de retornar uma lista detalhada de problemas, o uso de métricas gerais podem ajudar a identificar o nível geral de acessibilidade dos

websites considerados. Estas métricas podem ser definidas baseadas em critérios de sucesso e as correspondentes técnicas suficientes, consultivas e de falha. Alguns usuários não técnicos podem se interessar por resultados que tenham um alto nível de abstração, portanto, métricas e tabelas podem ajudar tais usuários a obterem uma compreensão melhor dos resultados de acessibilidade.

- **Tipo de Apresentação:** Devem existir diferentes maneiras de se reportar os resultados da validação de tal maneira a satisfazer as necessidades de diferentes tipos de usuários com diferentes níveis de habilidades e expertise. Por exemplo, um código com anotações se adequa mais a desenvolvedores *web*, enquanto que um relatório com gráficos e estatísticas sumarizando os problemas detectados podem ser mais intuitivos para usuários não técnicos.

- **A ferramenta fornece indicações práticas de como resolver os problemas identificados?**

Algumas ferramentas são apenas capazes de avaliar páginas *web* e não incluem a funcionalidade de ajudar os usuários a corrigirem as violações de acessibilidade identificadas. Certamente, uma adição útil a essas ferramentas seriam funcionalidades "de reparo" que auxiliariam o usuário no processo de correção de problemas ou simplesmente funcionalidades que forneceriam recomendações de soluções para cada problema.

- **A ferramenta é capaz de fornecer informações acerca de suas limitações?**

Este é um ponto crítico que permite os usuários interpretar os resultados corretamente. Um exemplo que representa bem este ponto é a capacidade da ferramenta de analisar páginas dinâmicas ou não. Inúmeras ferramentas de avaliação de acessibilidade se baseiam no HTML estático, porém, os *web* sites atuais em sua maioria evoluíram para um conteúdo mais dinâmico. Nestes casos, a falta de erros não necessariamente indica que o site é totalmente acessível, mas sim que a ferramenta é incapaz de acessar a verdadeira versão que o usuário interage. Assim, não só a ferramenta é incapaz de acessar o *web* site por inteiro, como também não fornece nenhum indicativo acerca de sua limitação com o potencial risco de se gerar uma falsa sensação de confiança entre os usuários. Para ser transparente, as ferramentas de avaliação de acessibilidade devem fornecer aos seus usuários informações claras acerca de suas funcionalidades, incluindo possíveis limitações.

2.2.1 Access Monitor

O *Access Monitor* é um verificador de acessibilidade criado pela AMA (Agência para a Modernização Administrativa) em parceria com o governo de Portugal (AMA (2020)). Este foi o verificador escolhido para servir de base para a criação do *Full Scan Accessibility Checker*.

A escolha deste verificador para este trabalho se deu pelos seguintes motivos:

- É um verificador gratuito;
- Tem uma interface intuitiva;
- Realiza suas análises com base nas diretrizes WCAG 2.1;
- Possui um sistema de notas de 0 a 10 em seus relatórios de acordo com o nível de conformidade da páginas com as diretrizes, tornando mais compreensível seus resultados;
- Apresenta de forma detalhada os problemas de acessibilidade contidos em uma página assim como indica sugestões de melhorias e correções para os problemas apresentados.

O verificador fornece três opções de entrada de dados: inserção de URL, inserção de código HTML e inserção de arquivo HTML como pode ser visto na figura 2.2

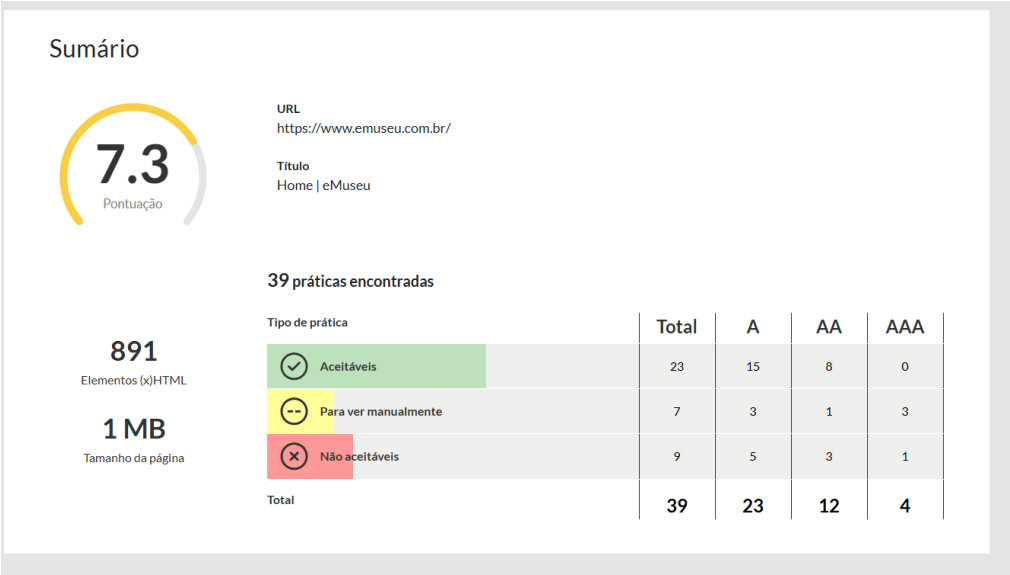
Figura 2.2: Tipos de entrada do Access Monitor. Fonte: accessmonitor.acessibilidade.gov.pt



The screenshot shows the 'Access Monitor' interface on the website 'Acessibilidade.gov.pt'. The main heading is 'A sua página Web é acessível?' (Is your website accessible?). Below it, the instruction is 'Obtenha um relatório de acessibilidade através de um dos seguintes métodos:' (Get an accessibility report through one of the following methods:). There are three tabs: 'Inserir URL' (selected), 'Inserir código HTML', and 'Carregar ficheiro HTML'. Under the 'Inserir URL' tab, the text 'Introduza o endereço de uma página' (Enter the address of a page) is followed by a text input field containing 'http(s)'. Below the input field is a 'Validar' button with a right-pointing arrow. To the right of the input field is a large green checkmark icon inside a circle, with a document icon above it.

O resultado da análise de uma página apresenta informações como pontuação da página de acordo com o nível de conformidade, quantidade de elementos HTML analisados, tamanho(em *MegaBytes*) da página analisada e quantidade de práticas de acessibilidade encontradas assim como seus tipos. A figura 2.3 demonstra essa estrutura.

Figura 2.3: Resultados de análise do Access Monitor. Fonte: accessmonitor.acessibilidade.gov.pt



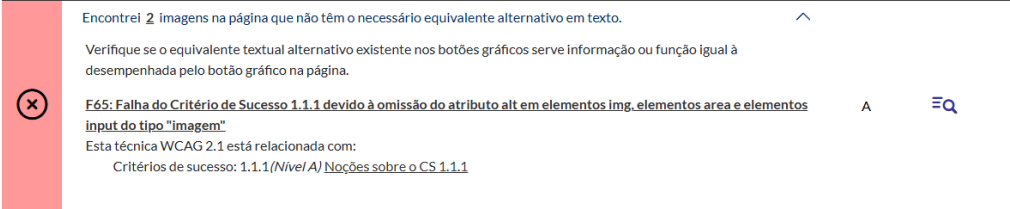
Logo abaixo do resultado de análise temos uma seção onde são detalhadas as práticas de acessibilidade encontradas bem como as sugestões de melhoria como podemos ver nas figuras 2.4 e 2.5.

Figura 2.4: Práticas de acessibilidade encontradas pelo Access Monitor. Fonte: accessmonitor.acessibilidade.gov.pt

Avaliação

Prática encontrada	Nível	Ver detalhe
<div>✗</div> Encontrei 2 imagens na página que não têm o necessário equivalente alternativo em texto.	✓ A	≡Q
<div>⚠</div> Encontrei 3 imagens na página com alt="" (alt vazio).	✓ A	≡Q
<div>✗</div> Encontrei 4 imagens na página com alt que não serve de equivalente alternativo.	✓ A	≡Q
<div>⚠</div> Encontrei 2 grupos de links com o mesmo texto mas cujo destino é diferente.	✓ AAA	≡Q
<div>✗</div> Constatei que a primeira hiperligação da página não permite saltar diretamente para a área do conteúdo principal.	✓ A	≡Q
<div>⚠</div> Encontrei 11 cabeçalhos na página.	✓ AAA	≡Q
<div>✓</div> Constatei que todos os controlos de formulário têm um nome acessível.	✓ A	

Figura 2.5: Detalhamento e sugestões de melhoria pelo Access Monitor. Fonte: accessmonitor.acessibilidade.gov.pt



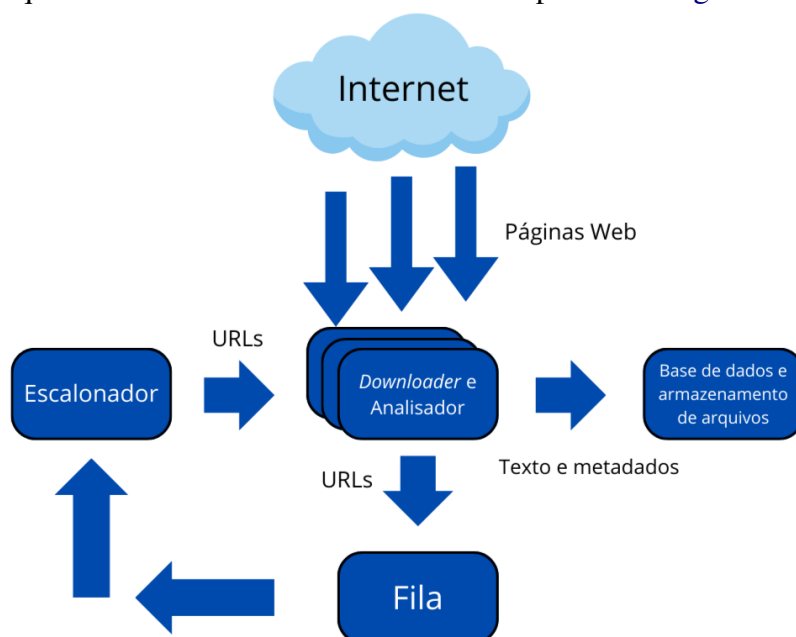
2.3 web crawlers

De acordo com [Kausar et al. \(2013\)](#) *web crawlers* são programas/software ou scripts que navegam pela internet de maneira automatizada e sistemática, também são conhecidos como robôs, *spiders* e *worms*. [Kausar et al. \(2013\)](#) definem a internet como um grafo direcionado onde cada página *web* é um nó e cada *hyperlink* uma aresta, desta forma, o processo de busca realizado por um *web crawler* pode ser descrito como o mesmo de uma travessia por um grafo.

web crawlers foram criados com o propósito de recuperar e armazenar páginas *web* para salvá-las em um repositório local, posteriormente criando réplicas das páginas visitadas que serão processadas por mecanismos de busca e indexadas para uma rápida recuperação da informação. Apesar disso, *web crawlers* podem ser usados em diversas aplicações como validação de páginas, análise e visualização estrutural, notificação de atualizações, agentes *web* pessoais etc([Kausar et al. \(2013\)](#) e [Dhenakaran and Sambanthan \(2011\)](#)).

[Singh and Varnica \(2014\)](#) descrevem o funcionamento de um *web crawler* da seguinte maneira: a partir de uma URL denominada de *seed*(semente) são identificados diferentes *hyperlinks* na página que são adicionados a uma lista, utilizando uma série de regras e políticas, as URLs na lista são visitadas individualmente, diferentes páginas da *web* são baixadas pelo analisador e armazenadas na base do sistema do mecanismo de busca. As URLs são então colocadas na fila e posteriormente escalonadas pelo escalonador e podem ser acessadas uma por uma pelo mecanismo de busca sempre que requeridas, os *links* e arquivos relacionados ficam disponíveis a qualquer momento requerido. A figura 2.6 ilustra a arquitetura de um *web crawler*.

Figura 2.6: Arquitetura de um *web crawler*. Fonte: adaptada de [Singh and Varnica \(2014\)](#)



3

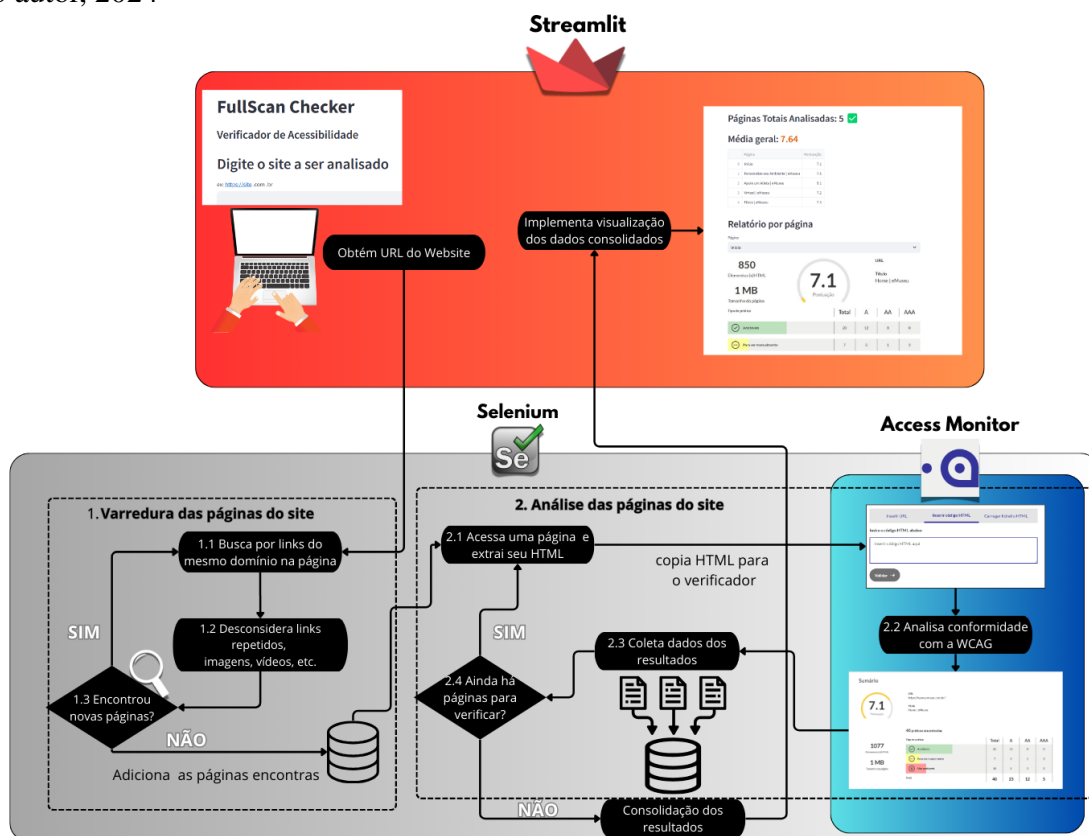
Desenvolvimento

Este capítulo detalha o desenvolvimento da ferramenta *FullScan Accessibility Checker*¹, um verificador de acessibilidade de varredura completa, ou seja, um verificador capaz de realizar a verificação de acessibilidade em todas as páginas de um *website*. A ferramenta consiste de três componentes principais: o *web crawler* que realiza a busca pelas páginas de um *website*, o avaliador que realiza a verificação de acessibilidade de cada página e a interface do usuário. O processo de *webcrawling* e a coordenação do processo de análise são implementados através da ferramenta *Selenium*, que provê um *webdriver* para realizar o acesso e a interação com os *websites*, principalmente aqueles com conteúdo dinâmico. A interface é implementada de maneira simples através do *framework Streamlit*, que facilita a visualização e interação com dados. A linguagem *Python* foi utilizada para o desenvolvimento e integração desses componentes.

A figura 3.1 apresenta a visão geral do funcionamento da ferramenta *FullScan Accessibility Checker*.

¹Disponível em <https://github.com/PatrickStyle66/FullScanAccessibilityChecker>

Figura 3.1: Visão geral do funcionamento do FullScan Accessibility Checker. Fonte: Elaborado pelo autor, 2024



Após obter a URL do *website* a ser analisado, o processo de varredura de páginas é iniciado. A ferramenta então acessa a página fornecida e realiza a busca por *links* que estejam sob o mesmo domínio dessa página (etapa 1.1). Durante esse processo de busca, *links* indesejados como *links* repetidos e arquivos de mídia são descartados (etapa 1.2), em seguida, a ferramenta verificará se novas páginas foram encontradas (etapa 1.3), caso essa afirmação seja verdadeira, a ferramenta retorna para a etapa 1.1 e o processo se repetirá até que nenhuma página nova seja encontrada. Com todas as páginas do *website* visitadas, a ferramenta irá adicionar os *links* de cada página em uma lista e seguirá para o processo de análise dessas páginas.

Durante o processo de avaliação de páginas, a ferramenta acessa uma página contida na lista e extrai seu HTML (etapa 2.1), em seguida, o HTML é copiado e enviado para o verificador *Access Monitor*, que realizará a análise de conformidade com a WCAG (etapa 2.2). As informações fornecidas pelo relatório de avaliação do *Access Monitor* são coletadas e armazenadas em um *dataframe* (etapa 2.3), em seguida, a ferramenta verificará se ainda há páginas na lista para serem avaliadas (etapa 2.4), caso a afirmação for verdadeira, a ferramenta retorna para a etapa 2.1 e o processo se repetirá até que todas as páginas presentes na lista tenham sido verificadas. Finalmente, os dados coletados são consolidados para serem exibidos na interface do usuário.

As próximas seções entrarão em detalhes a respeito do funcionamento de cada componente da ferramenta.

3.1 Busca e varredura de páginas

Antes de se inicializar a busca completa por páginas no domínio do *website*, uma verificação de acessibilidade é realizada na primeira página do mesmo. Tal verificação tem o intuito de testar se o verificador *Access monitor* será capaz de realizar as demais verificações de páginas, levando em consideração que, em determinados casos, por problemas internos da ferramenta, o verificador pode falhar em gerar o relatório de acessibilidade. Portanto, caso a verificação inicial falhe, o processo de busca pelas demais páginas não será iniciado, evitando-se assim, a alocação desnecessária de recursos.

Após a realização da verificação de teste na primeira página do *website*, a ferramenta irá iniciar a busca por mais páginas que estão sob o mesmo domínio. Para o processo de localização e interação com os elementos das páginas web foram utilizadas consultas XPATH, que selecionam nós de um documento HTML através de filtros definidos nas consultas. Tais filtros, além de servirem para a localização dos elementos, servem para o descarte de *links* indesejáveis no processo de busca por páginas.

A consulta recebe a variável que contém o domínio do *website* a ser analisado, a partir daí, a mesma será guiada através do operador *contains* e expressões booleanas que buscarão elementos que contenham em seus *links* o domínio do *website* ou aqueles que se iniciam com "/" ou "#", indicando *links* relativos à raiz do domínio. A consulta também será responsável por desconsiderar *links* desnecessários para a análise através do operador *not* antecedendo uma série de expressões booleanas, aqui foram definidos *links* de redes sociais notáveis com mais probabilidade de estarem contidos nos *websites* (*instagram*, *facebook*, *tiktok* etc.) e arquivos de mídia com os formatos .png, .jpg/.jpeg, .gif, .mp3, .pdf além de *links* para *e-mails*.

Após realizada a consulta, serão extraídos os *links* de cada elemento encontrado, serão descartados os *links* repetidos através da estrutura de conjunto, que admite apenas uma instância de cada elemento e serão salvos os *links* restantes em uma lista. Com a lista de *links* coletada da primeira página do *website*, a ferramenta irá navegar nas demais páginas e continuar o processo de coleta.

Em seguida é iniciado o *loop* que irá iterar por cada link contido na lista. Dentro desse *loop* será feita uma requisição para a página que está sendo iterada, caso o link analisado seja diferente da página inicial e caso o código de requisição retornado seja 200, ou seja, se a requisição for bem-sucedida, o *webdriver* acessará a página. Logo após, uma verificação é feita para testar se a página acessada continua estando no domínio do *website*, tal verificação é necessária pois, em alguns casos, *links* que estão sob o domínio de um *website* podem redirecionar o usuário para um domínio externo alheio ao atual. Caso a página acessada esteja em um domínio externo, a mesma é adicionada em uma lista para remoção e o *loop* continua na próxima página da lista.

Caso a página esteja sob o domínio do *website*, o processo de busca de páginas é feito novamente. Após a busca dos elementos, obtém-se uma nova lista de *links* encontrados, dessa

nova lista, apenas *links* novos são adicionados na lista original. O processo se repetirá até que se chegue no final da lista original, isto é, até que não haja mais nenhum link novo a ser adicionado no final da lista, significando que todas as páginas do *website* foram acessadas.

3.2 Verificação de acessibilidade de páginas

Com o processo de busca e varredura de páginas finalizado, a lista estará completa com todas as páginas do *website* prontas para serem analisadas. Um *loop* é iniciado e para cada link presente na lista, a ferramenta irá acessar a página e armazenar seu HTML em uma variável, em seguida, o *webdriver* irá acessar o verificador *Access Monitor*, localizar o elemento que habilita a caixa de *input*, clicá-lo e localizar a caixa para então deixá-la ativa. O código HTML da página é copiado e então colado na caixa de *input*. O *webdriver* localiza o botão de envio e o clica para que seja realizado o processo de análise da página.

Logo após, o *webdriver* irá aguardar até que seja gerado o relatório de acessibilidade pelo *Access Monitor* com um tempo limite de 1 minuto. Caso a ferramenta não consiga gerar o relatório neste tempo, a página será desconsiderada para a análise. Com o relatório gerado, o *webdriver* irá localizar e capturar a imagem do elemento que contém a nota de acessibilidade gerada pela ferramenta, bem como o valor numérico da mesma. Em seguida, a ferramenta localizará a tabela contendo as práticas de acessibilidade e irá capturar as imagens elemento por elemento desta tabela. Por fim, a tabela que resume a quantidade de práticas e as separa por critérios de sucesso também é localizada e tem sua imagem capturada.

Finalizando a captura das imagens, informações de título e link da página são armazenadas em um dicionário e as imagens contendo as demais informações são armazenadas em suas respectivas listas. Há casos em que mais de uma página possui o mesmo nome, para isso, a ferramenta renomeia as páginas adicionando no final um número equivalente a ordem em que esta repetição de nomes ocorreu (Ex.: Acervo - 1, Acervo - 2).

Saindo do *loop* principal, ou seja, após se verificarem todas as páginas do *website*, a média geral é calculada, assim como o dataframe contendo os resultados da análise é gerado para ser exibido.

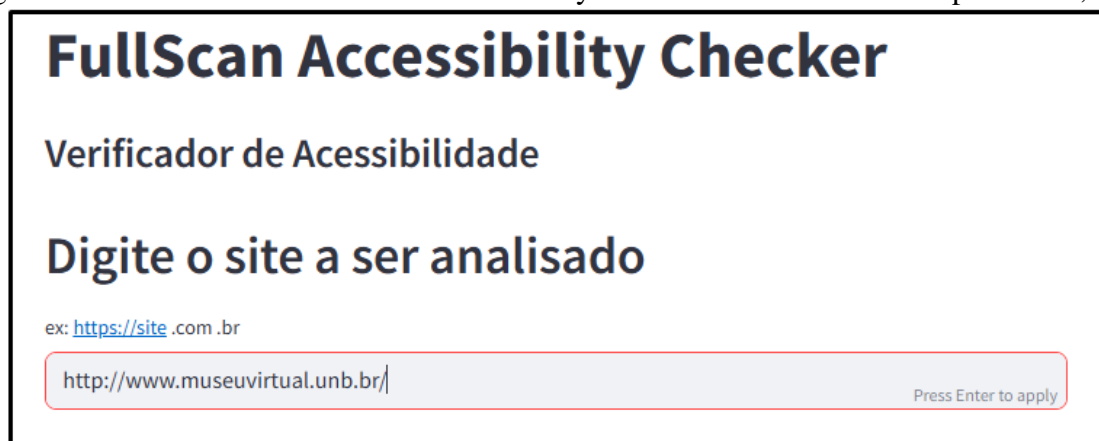
As funções responsáveis pela busca e varredura, bem como as responsáveis pela verificação de acessibilidade das páginas podem ser conferidas a partir da seção [B](#) dos apêndices.

3.3 Interface do usuário

Ao iniciar a execução da aplicação através do comando `streamlit run FullScanChecker.py`, o usuário se depara com uma tela simples: um título indicando o nome da ferramenta, um subtítulo indicando do que se trata a ferramenta e uma instrução seguida de uma caixa de texto indicando o local onde o usuário irá inserir a URL do web site a ser analisado como indica a

figura 3.2.

Figura 3.2: Tela inicial do FullScan Accessibility Checker. Fonte: Elaborado pelo autor, 2024

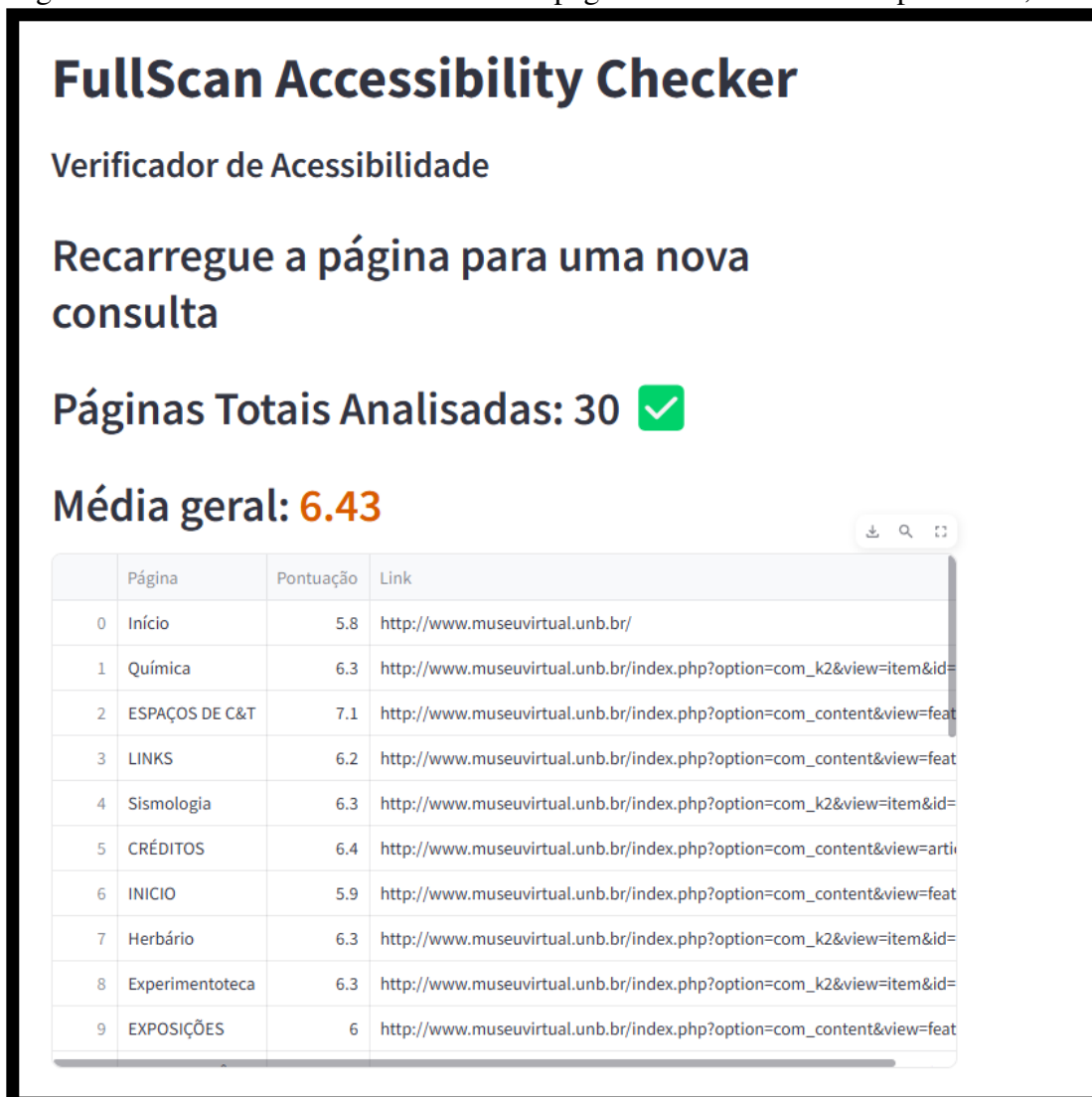


The image shows the initial screen of the FullScan Accessibility Checker. It features a white background with a black border. At the top, the title "FullScan Accessibility Checker" is displayed in a large, bold, dark blue font. Below it, the subtitle "Verificador de Acessibilidade" is in a smaller, bold, dark blue font. The main instruction "Digite o site a ser analisado" is in a large, bold, dark blue font. Below this, there is a small example text "ex: <https://site.com.br>". A text input field contains the URL "http://www.museuvirtual.unb.br/". To the right of the input field, there is a small text "Press Enter to apply".

Após inserida a URL, o processo de análise será iniciado começando pela verificação da primeira página. Logo em seguida, ocorre o processo de busca por páginas, onde a interface indicará em qual página a ferramenta está realizando a busca e quantas páginas já foram encontradas até o momento.

Finalizado o processo de verificação de acessibilidade, a ferramenta irá apresentar os resultados. Começando pelo número total de páginas verificadas pela ferramenta, vale ressaltar que este número pode diferir do número de páginas encontradas para casos em que uma ou mais páginas falhem na geração de relatório pelo *Access Monitor*. Em seguida, é apresentada a média geral de acessibilidade do site juntamente com um *dataframe* interativo que permite ser baixado e convertido em .csv contendo as páginas verificadas, as notas individuais de acessibilidade e os *links* das páginas como é demonstrado na figura 3.3.

Figura 3.3: Tela de resultado de análise de páginas. Fonte: Elaborado pelo autor, 2024



Logo abaixo do resultado geral, é apresentada a seção de relatório individual por página, onde o usuário tem uma visão detalhada da análise de acessibilidade de uma determinada página. Após o usuário selecionar a página desejada através de um menu *dropdown* indicado na figura 3.4, são exibidas as informações de pontuação de acessibilidade, quantidade de elementos HTML presentes na página, tamanho da página, tabela de resumo de práticas de acessibilidade separadas por critério de sucesso e tipo (Aceitáveis, Para ver manualmente e Não aceitáveis) e finalmente, cada prática de acessibilidade individualmente como é demonstrado na figura 3.5

Figura 3.4: Menu *dropdown* de seleção de páginas. Fonte: Elaborado pelo autor, 2024

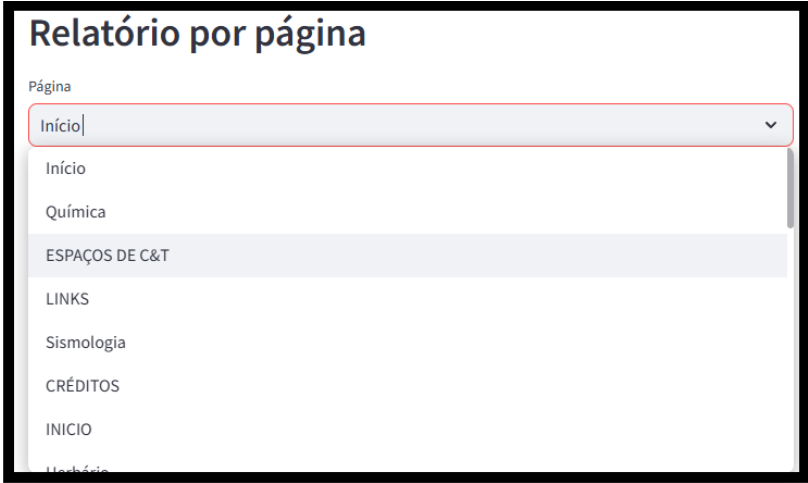
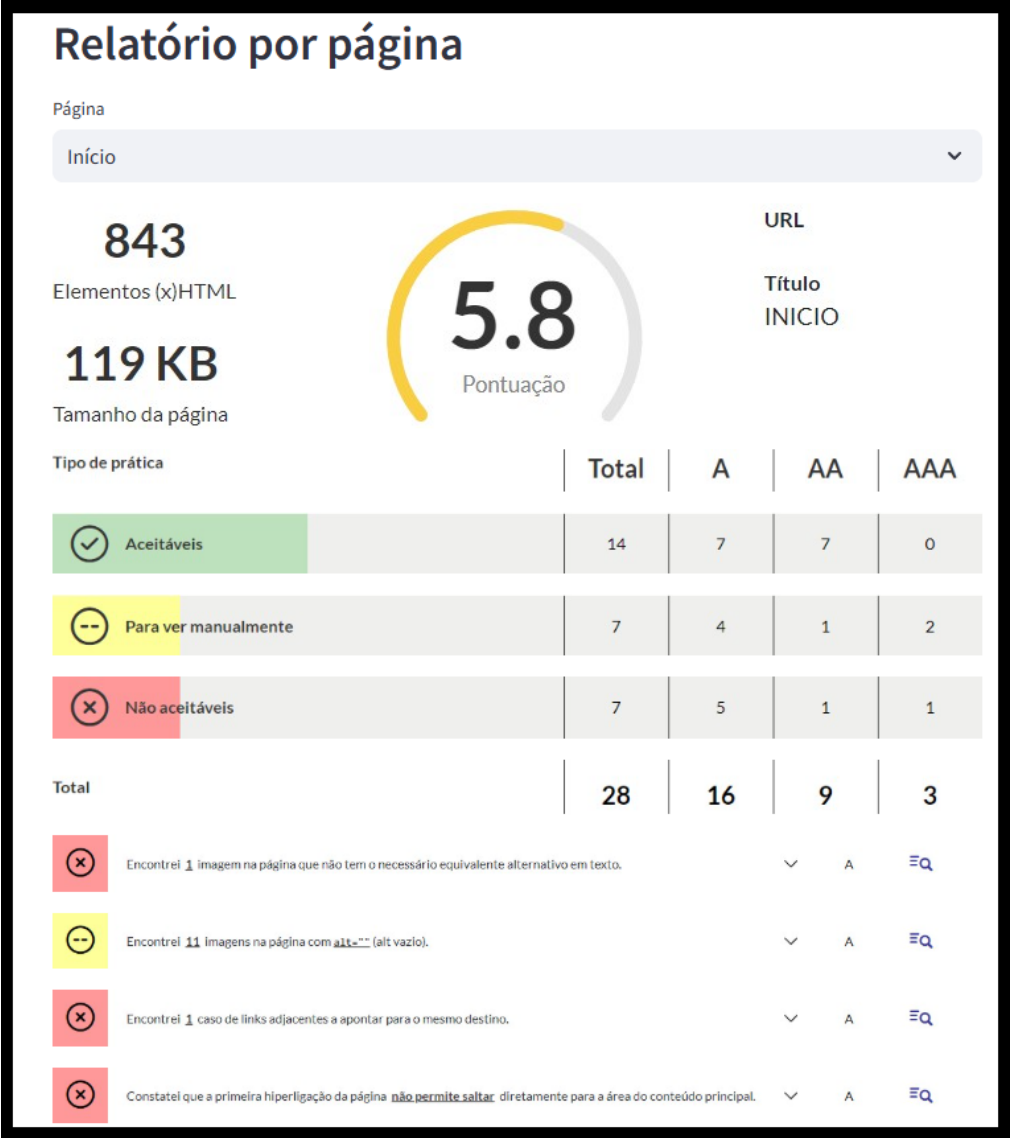


Figura 3.5: Relatório de acessibilidade por página. Fonte: Elaborado pelo autor, 2024





Estudo de caso: Museus Virtuais

Espaços de cultura e lazer, como museus, atraem diversos tipos de público, incluindo visitantes com algum tipo de deficiência. A legislação brasileira busca assegurar as condições para o acesso de pessoas com deficiência a esses espaços, garantindo que sejam acessíveis a qualquer indivíduo. Em se tratando de museus virtuais, a expectativa não é diferente. Segundo [Schweibenz \(1998\)](#) *O museu virtual não compete com o tradicional visto que não pode oferecer a seus visitantes objetos reais, entretanto, ele pode disseminar ideias e conceitos em acervos no espaço digital, alcançando visitantes virtuais que talvez nunca tiveram a possibilidade de visitar um museu pessoalmente.*

Para a validação da ferramenta *FullScan Accessibility Checker* foi realizado um estudo de caso com o propósito de verificar a acessibilidade de web sites de museus virtuais brasileiros, provendo uma análise completa acerca da acessibilidade desses espaços com base nas diretrizes WCAG. As seções seguintes descrevem as etapas implementadas no processo de levantamento de dados e análises deste estudo de caso.

4.1 Levantamento e coleta de dados

Inicialmente, foi necessário encontrar uma fonte confiável que reunisse um catálogo abrangente de museus virtuais, permitindo o acesso remoto a exposições, acervos e até mesmo informações institucionais. Após uma pesquisa criteriosa, optou-se pela utilização da plataforma MuseusBR¹, vinculada ao IBRAM² (Instituto Brasileiro de Museus) que mantém o cadastro de mais de 3.000 museus espalhados pelo território brasileiro, como mostra a figura 4.1.

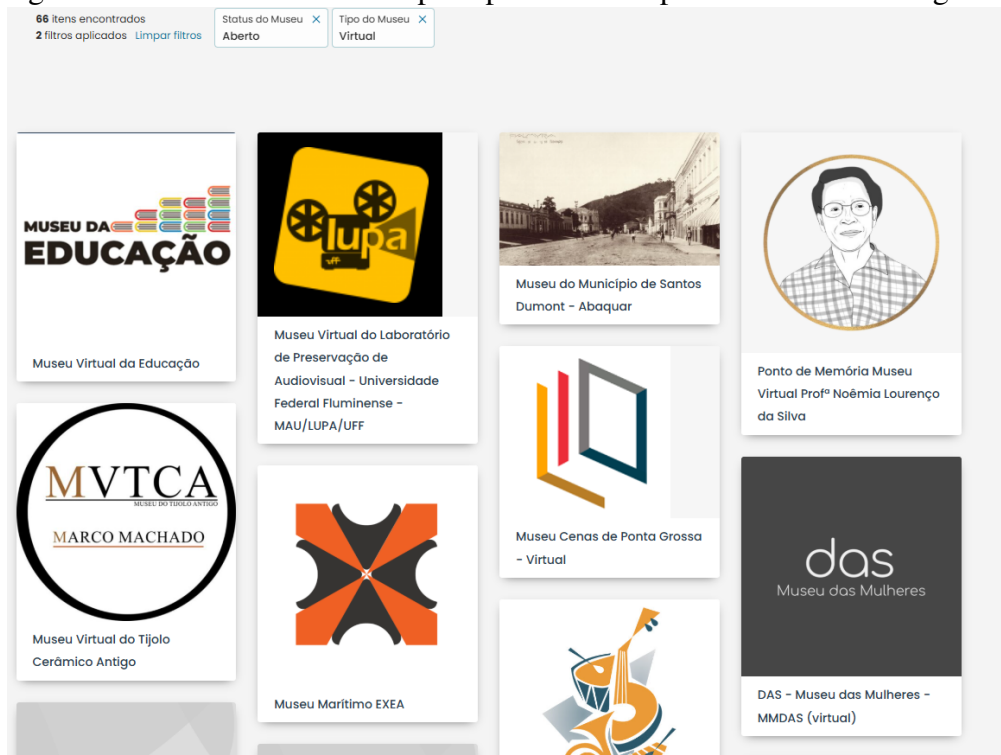
¹<https://cadastro.museus.gov.br/>

²<https://www.gov.br/museus/pt-br>

Figura 4.1: Página inicial da plataforma MuseusBR. Fonte: <https://cadastro.museus.gov.br/>

A plataforma MuseusBR permite realizar buscas por museus brasileiros através de vários filtros como *status* do museu (aberto ou fechado), esfera administrativa (particular ou pública), tipo do museu, sua temática e o Estado a que pertence. A busca para o estudo de caso foi realizada em 26/07/2024³ e utilizou o filtro Tipo do Museu, selecionando a opção *Virtual* e o filtro Status com a opção *Aberto*, o que permitiu recuperar todos os museus virtuais cadastrados na plataforma. A plataforma apresentou como resultado uma série de museus virtuais como mostra a figura 4.2. Ao selecionar um museu, a plataforma direciona o usuário para uma página contendo várias informações acerca do museu selecionado, incluindo o *link* de acesso conforme apresentado na figura 4.3

³para alguns museus, uma nova verificação foi realizada posteriormente devido a problemas com links coletados inicialmente.

Figura 4.2: Resultados de busca por tipo. Fonte: <https://cadastro.museus.gov.br/>

Para coletar os links dos museus virtuais apresentados na plataforma foi desenvolvido um script em *python* com o auxílio da ferramenta *Selenium* e da biblioteca *BeautifulSoup*. A combinação dessas ferramentas foi essencial para a automatização do processo de extração dos dados, a ferramenta *Selenium* permitiu a navegação automatizada e interativa pela plataforma enquanto que a biblioteca *BeautifulSoup* foi empregada para realizar o processamento e análise do código HTML das páginas acessadas extraindo as informações de links e nomes dos museus.

Figura 4.3: Página de um museu na plataforma MuseusBR. Fonte: <https://cadastro.museus.gov.br/>

O script inicia com a função `getMuseumsSite()`, que invoca o *webdriver* acessando a plataforma MuseusBR através do método `get` do driver e a variável `MuseusBR` que contém a página

de busca já com os filtros configurados. Em seguida, o *webdriver* localizará o elemento que contém a informação de quantas páginas a busca retornou e salvará na variável *pages*, tal informação é necessária para determinar quantas vezes o processo será repetido. Posteriormente, o *webdriver* localizará todos os elementos que contêm os museus listados e salvará na variável *museums*. O loop principal é iniciado, para cada elemento contido na lista será extraído seu *link* e em seguida será feita uma requisição HTML para esse *link*, caso a requisição seja bem sucedida, o conteúdo HTML será salvo na variável *content* para ser processado pela biblioteca *BeautifulSoup*. De posse do conteúdo HTML, a biblioteca localizará os elementos que contêm as informações de nome e *link* para os museus virtuais salvando-os em um dicionário para posteriormente ser convertido num arquivo .csv denominado *MuseumsScores*. Após a extração das informações, o *webdriver* localizará o elemento contendo o botão que leva à próxima página e clicará nele, o processo se repetirá até que se tenha obtido os links na última página da busca. A função descrita é apresentada na seção C.1 dos apêndices.

4.2 Resultados

No momento da realização do estudo, o script retornou 60 museus virtuais. A partir da geração da lista de museus, um pré-processamento foi realizado para garantir a qualidade e relevância dos dados. Nesse processo, os links foram verificados quanto à sua disponibilidade, sendo removidos aqueles que estavam fora do ar ou que retornavam erros de conexão e quanto a resposta dada pelo verificador *Access Monitor*, links que não puderam ser analisados pela ferramenta, devido a incompatibilidades ou falhas técnicas, também foram descartados. Esse pré-processamento foi essencial para assegurar que apenas os sites ativos e compatíveis fossem considerados nas etapas subsequentes da análise. A partir do pré-processamento, 17 museus foram removidos por estarem fora do ar e 7 removidos por erro no *Access Monitor*, restando 36 museus os quais podem ser observados na tabela 4.1.

Tabela 4.1: Resultados da coleta dos links de museus após pré-processamento

Museu	Website
Museu Virtual Prof. Noêmia L. da Silva	https://www.museunoemia.com.br
Museu Cenas de Ponta Grossa – Virtual	https://museucenas.com.br/#/
DAS – Museu das Mulheres – MMDAS	https://www.museudasmulheres.com.br/
Museu Marítimo EXEA	https://www.museuexea.org/
Museu Virtual de Instrumentos Musicais	http://www.mvim.com.br/
Museu do Mun. Santos Dumont – Abaquar	https://museu.abaquarcult.com.br/
Museu Afrodigital do Maranhão – UFMA	https://www.museuafro.ufma.br/
Museu da Lava Jato	https://museudalavajato.com.br/
Museu das Coisas Banais – UFPel	https://museudascoisasbanais.com.br/
Museu Digital Unila – MUD UNILA	https://museudigital.unila.edu.br/
Museu Afro-brasil Sul – MABSul – UFPel	https://acervosvirtuais.ufpel.edu.br/museuafrobrasilsul/

Museu Digital Tramas Daqui	http://museutramasdaqui.art.br/
Museu Virtual de Rochas, Solos e Minerais – UFPel	https://wp.ufpel.edu.br/museudesolosrochaseminerais/
Museu de Memes – UFF	http://www.museudememes.com.br/
Museu de Artes e Ofícios Bodoque	http://maob.artebodoque.com/
Museu Virtual do Esporte – UERJ	http://www.emuseu.com.br
Museu da Pelada Social Clube	http://www.museudapelada.com
Portal de Arquitetura Alagoana – UFAL	http://arquiteturaalagoana.al.org.br/
Museu Itamar Assumpção	https://www.itamarassumpcao.com/
Museu do Cerrado – UnB	https://museucerrado.com.br/
Museu dos Meninos – Comp. do Alemão	https://www.museudosmeninos.com.br/
Centro de Memória do Esporte (CEMESP)	http://www.cemesp.unimontes.br/
Centro História e Cultura David Pimenta	http://centro davidpimenta.webnode.com
Museu das Memórias (In)Possíveis	http://www.apoa.com.br/museu
Memorial da Televisão Brasileira	https://memoria.abert.org.br/exposicao-virtual
Museu Virtual Vale do Mucuri	https://museuvirtualvaledomucuri.com.br/
Cinema de Terreiro – Museu Digital	https://cinemadeterreiro.com/
Museu Virtual de Adamantina	https://www.muvia.com.br
Museu Bajubá	https://museubajuba.org/
Museu da Fotografia Documental	http://www.mfd.mus.br
Museu Virtual do Rio Grande	http://www.riograndeemfotos.fot.br/museuvirtual/
Museu Afro-digital da Memória Africana e Afro-brasileira da UFBA	http://www.museuafrodigital.ufba.br
Museu Virtual de São José do Norte	http://www.riograndeemfotos.fot.br/museusaojosedonorte
Museu Virtual de Uberlândia	http://www.museuvirtualdeuberlandia.com.br/
Museu Virtual de Ciência e Tec. da UnB	http://www.museuvirtual.unb.br
Museu da Mantiqueira	https://soundcloud.com/museudamantiqueira
Museu Virtual Rio Memórias	https://riomemorias.com.br/

Após a realização do pré-processamento, os links de museus foram submetidos à análise do *FullScan Accessibility checker* um por um e seus resultados são exibidos na tabela 4.2

Tabela 4.2: Resultados da análise de acessibilidade de museus virtuais

Museu	Pontuação Média	Páginas Verificadas
Museu Cenas de Ponta Grossa – Virtual	8.84	31
DAS – Museu das Mulheres – MMDAS	8.82	51
Museu Virtual Prof. Noêmia L. da Silva	8.77	9
Museu da Pelada Social Clube	8.67	234
Museu Virtual de Rochas, Solos e Minerais – UFPel	8.48	18
Museu Marítimo EXEA	8.32	186
Museu Virtual Rio Memórias	8.24	25
Museu Virtual de Uberlândia	8.18	90
Museu dos Meninos – Virtual – Complexo do Alemão	7.83	8
Museu da Mantiqueira	7.8	39
Museu da Lava Jato	7.72	39
Museu Virtual do Esporte – Universidade Estadual do Rio de Janeiro	7.64	5
Museu do Cerrado – Universidade de Brasília – UnB	7.56	50
Museu do Mun. de Santos Dumont – Abaquare	7.44	36
Museu Virtual de Adamantina	7.44	103
Museu Digital Unila – MUD UNILA	7.38	20
Museu das Memórias (In)Possíveis – Virtual	7.28	146
Museu de Artes e Ofícios Bodoque – Virtual	7.12	86
Portal de Arquitetura Alagoana – Universidade Federal de Alagoas	7.02	255
Museu Virtual Vale do Mucuri	6.95	16
Centro de Memória do Esporte (CEMESP)	6.94	5
Museu da Fotografia Documental	6.89	28
Centro Virtual de História e Cultura David Pimenta	6.85	25
Museu das Coisas Banais – UFPel	6.64	27
Museu Virtual de Ciência e Tecnologia da Universidade de Brasília	6.53	29
Cinema de Terreiro – Museu Digital	6.5	25
Museu Afro-brasil Sul – MABSul – UFPel	6.44	148
Museu de Memes – Universidade Federal Fluminense	6.4	40
Museu Bajubá	6.33	28
Museu Digital Tramas Daqui	6.28	74
Museu Afro-digital da Memória Africana e Afro-brasileira da Universidade Federal da Bahia	6.24	206
Museu Virtual de Instrumentos Musicais	6.13	1497
Museu Itamar Assumpção	5.93	63
Museu Afrodigital do Maranhão – UFMA	5.48	81
Museu Virtual do Rio Grande	3.95	11
Museu Virtual de São José do Norte	3.77	3

Dos 36 museus avaliados, 8 apresentaram um alto nível de acessibilidade (nota de 8 a 10), 24 apresentaram um nível intermediário de acessibilidade (nota de 6 a 7) e 4 apresentaram um baixo nível de acessibilidade (nota de 0 a 5) como é descrito nas figuras 4.4 e 4.5. O museu

com o nível mais baixo de acessibilidade por média foi o Museu Virtual de São José do Norte, apresentando uma média de 3.77, enquanto que o museu com maior nível de acessibilidade por média foi o Museu Cenas de Ponta Grossa com média 8.84. Considerando o número de páginas verificadas, o Museu Virtual de Instrumentos Musicais aparece no topo com 1497 páginas, contrastando com o Museu Virtual de São José do Norte com apenas 3 páginas.

Figura 4.4: Porcentagem dos resultados por nível de acessibilidade. Fonte: Elaborado pelo autor, 2024

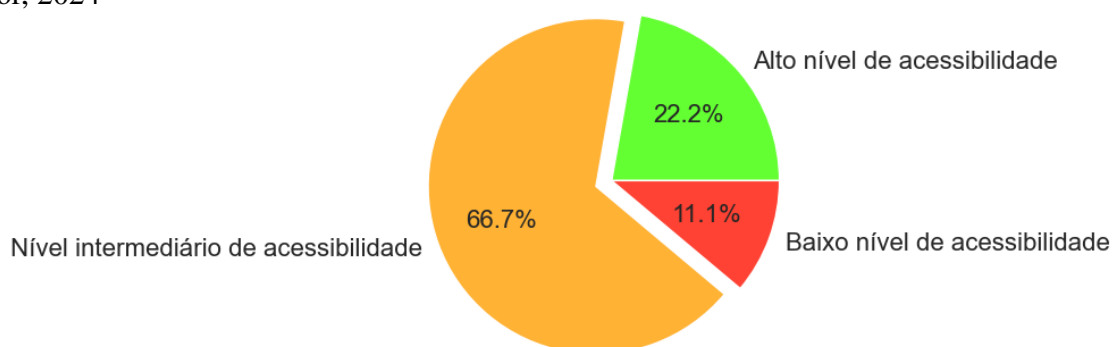
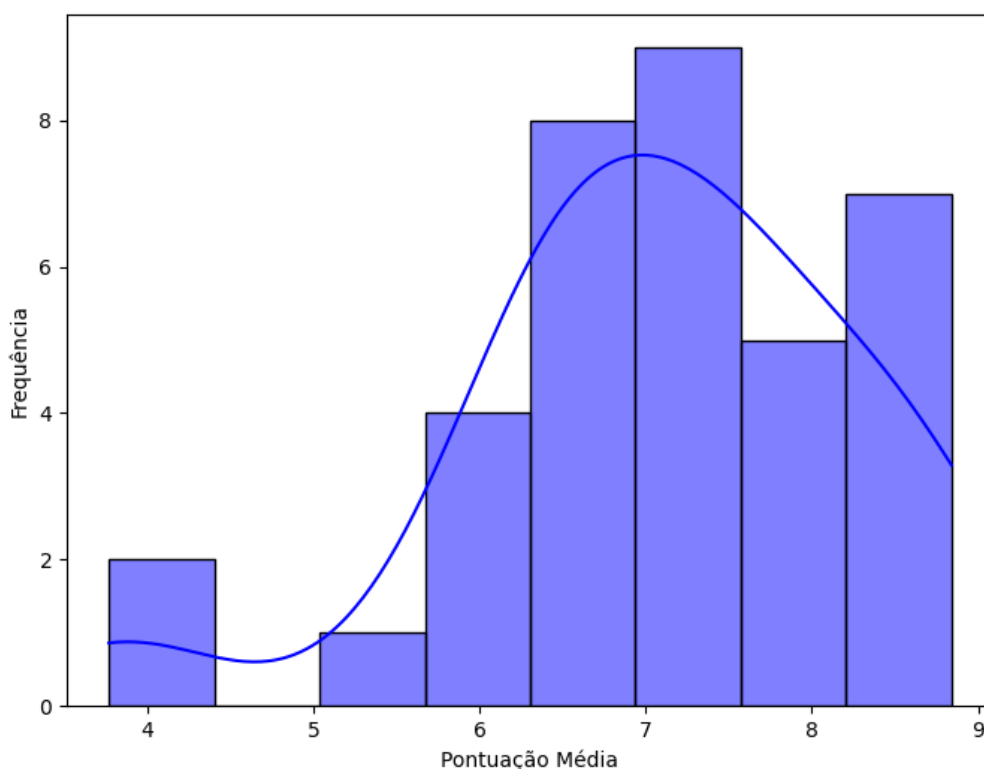


Figura 4.5: Resultados por média de acessibilidade. Fonte: Elaborado pelo autor, 2024

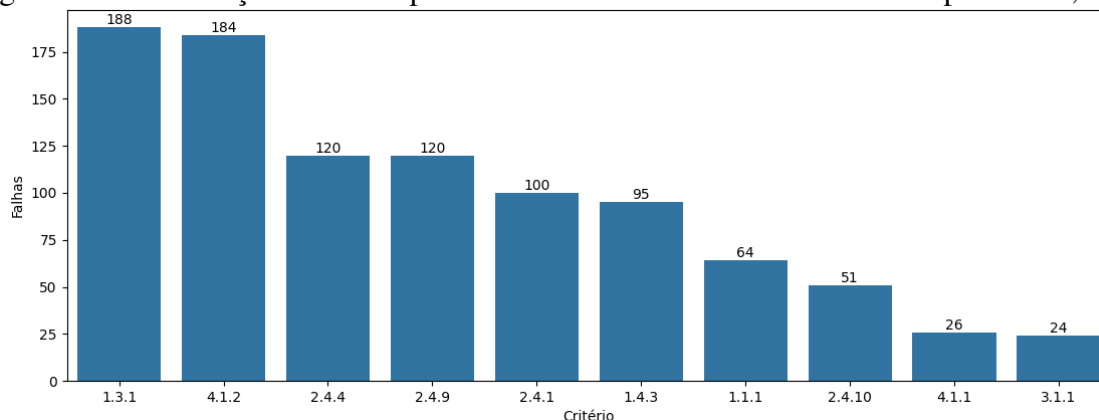


Analisando os resultados considerando a amplitude de avaliação, ou seja, levando em conta a influência que o número de páginas verificadas tem para a média final, observa-se que o Museu da Pelada Social Clube demonstrou o resultado mais satisfatório. Em um cenário onde um número maior de variáveis torna o resultado mais robusto, o museu apresentou uma média de

8.67 com 234 páginas verificadas mostrando-se o mais consistente em relação ao seu alto nível de conformidade com as diretrizes WCAG ao longo de suas páginas. Em contrapartida, o Museu Virtual de Instrumentos musicais, com o maior número de páginas verificadas, apresentou uma média de 6.13 demonstrando um resultado intermediário para uma amostra mais extensa. Para o nível mais baixo de acessibilidade, é identificado o Museu Virtual do Rio Grande que, com 11 páginas verificadas, apresenta uma média de 3.95, indicando que o problema de acessibilidade persiste por todo o website.

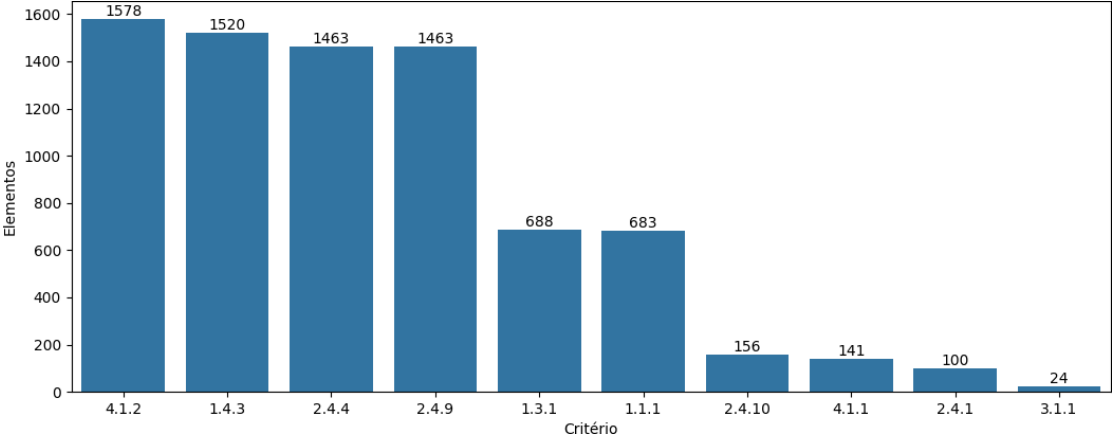
Coletando uma amostra de 130 páginas selecionadas aleatoriamente do grupo de museus com um nível intermediário de acessibilidade e analisando as falhas nos critérios de sucesso, obteve-se o seguinte resultado: 817 falhas foram encontradas, sendo 188 delas falhas no critério de sucesso 1.3.1 (Informações e Relações), 184 no critério de sucesso 4.1.2 (Nome, Função, Valor), 120 nos critérios de sucesso 2.4.4 (Finalidade do *link* em contexto) e 2.4.9 (Finalidade do *link*(apenas o *link*)), 100 no critério 2.4.1 (Ignorar Blocos) e as demais falhas espalhadas pelos critérios de sucesso restantes. Os 10 critérios de sucesso que mais apresentaram falhas são demonstrados na figura 4.6.

Figura 4.6: Distribuição de falhas por critério de sucesso. Fonte: Elaborado pelo autor, 2024



Considerando o número de elementos que apresentaram falhas, o critério de sucesso 4.1.2 (Nome, Função, Valor) aparece em primeiro lugar com 1578 elementos, em segundo lugar temos o critério de sucesso 1.4.3 (Contraste Mínimo) com 1520 elementos seguido dos critérios 2.4.4 (Finalidade do *link* em contexto) e 2.4.9 (Finalidade do *link*(apenas o *link*)) com ambos 1463 elementos. Os critérios de sucesso com maior número de elementos por falha são demonstrados na figura 4.7.

Figura 4.7: Distribuição de elementos por falhas. Fonte: Elaborado pelo autor, 2024



5

Conclusão

Neste trabalho foi desenvolvido um verificador de acessibilidade com a capacidade de realizar uma análise completa em um *website*, tendo como estudo de caso uma análise geral acerca da acessibilidade em museus virtuais brasileiros. Os objetivos traçados foram claros e cumpridos de maneira satisfatória; a estratégia de combinação de um modelo de *Web Crawler* com um avaliador de acessibilidade foi crucial para a criação de uma ferramenta que atendesse aos requisitos do trabalho. A interface apresentada de maneira simples e intuitiva facilita a interação com o usuário, focando a atenção para a exibição dos dados.

Os resultados apresentados pelo estudo caso revelam que a grande maioria dos museus virtuais possui um nível de acessibilidade intermediário, evidenciando um cenário onde há muito a ser melhorado para que se garanta uma experiência inclusiva a todo tipo de usuário. Embora a pontuação média obtida por esses *websites* indique alguma conformidade com os padrões, é importante destacar que os museus virtuais apresentam falhas em aspectos essenciais como a determinação de uma finalidade clara para os elementos e navegação intuitiva, além da falta de disponibilização de texto alternativo para imagens. Este tipo de resultado enfatiza a importância de ferramentas como a desenvolvida neste trabalho no auxílio da identificação de problemas e, posteriormente, na implementação de melhorias.

5.1 Trabalhos Futuros

Apesar de cumprir sua função, o verificador *FullScan Accessibility Checker* tem espaço para melhorias e otimizações. Em termos de eficiência, para um *website* com um número extenso de páginas, o verificador leva um tempo considerável para realizar a busca e a verificação de páginas; portanto, uma forma de otimizar a busca por links do mesmo domínio de um *website*, bem como a verificação da análise de páginas, seria por meio da implementação de *threads*, paralelizando o processo e diminuindo o tempo de resposta. Em termos de cobertura de página,

alguns *websites* possuem mecanismos à prova de automações como a verificação de *bot*, o que impede que páginas sejam acessadas e analisadas. Uma melhoria para o verificador seria implementar uma forma de superar esses mecanismos, permitindo assim a avaliação completa do *website*.

Referências bibliográficas

Julio Abascal, Myriam Arrue, and Xabier Valencia. *Tools for Web Accessibility Evaluation*, pages 479–503. Springer London, London, 2019. ISBN 978-1-4471-7440-0.

DOI [10.1007/978-1-4471-7440-0_26](https://doi.org/10.1007/978-1-4471-7440-0_26). URL

https://doi.org/10.1007/978-1-4471-7440-0_26.

Iyad Abu Doush, Khalid Sultan, Mohammed Azmi Al-Betar, Zainab Almeraj, Zaid Abdi Alkareem Alyasseri, and Mohammed A. Awadallah. Web accessibility automatic evaluation tools: to what extent can they be automated? *CCF Transactions on Pervasive Computing and Interaction*, 5(3):288–320, Sep 2023. ISSN 2524-5228.

DOI [10.1007/s42486-023-00127-8](https://doi.org/10.1007/s42486-023-00127-8). URL

<https://doi.org/10.1007/s42486-023-00127-8>.

Hayfa Y Abuaddous, Mohd Zalisham Jali, and Nurlida Basir. Web accessibility challenges. *International Journal of Advanced Computer Science and Applications*, 7(10), 2016.

Abdullah Alsaeedi. Comparing web accessibility evaluation tools and evaluating the accessibility of webpages: Proposed frameworks. *Information*, 11(1), 2020. ISSN 2078-2489. **DOI** [10.3390/info11010040](https://doi.org/10.3390/info11010040). URL

<https://www.mdpi.com/2078-2489/11/1/40>.

AMA. Access monitor, 2020. URL <https://accessmonitor.acessibilidade.gov.pt/>. Acessado em novembro de 2024.

Australia. Disability discrimination act, 1992. URL

<https://www.legislation.gov.au/C2004A04426/2018-04-12/text>. Acessado em outubro de 2024.

Giorgio Brajnik. Beyond conformance: the role of accessibility evaluation methods. In *International Conference on Web Information Systems Engineering*, pages 63–80. Springer, 2008.

Brasil. Decreto nº 6.949, de 25 de agosto de 2009., 2009. URL [http:](http://www.planalto.gov.br/ccivil_03/_ato2007-2010/2009/decreto/d6949.htm)

[//www.planalto.gov.br/ccivil_03/_ato2007-2010/2009/decreto/d6949.htm](http://www.planalto.gov.br/ccivil_03/_ato2007-2010/2009/decreto/d6949.htm).

Acessado em outubro de 2024.

- Brasil. Lei nº 13.146, de 6 de julho de 2015. lei brasileira de inclusão da pessoa com deficiência (estatuto da pessoa com deficiência), 2015. URL https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2015/lei/113146.htm. Acessado em outubro de 2024.
- Peter Brophy and Jenny Craven. Web accessibility. *Library trends*, 55(4):950–972, 2007.
- CTIC. Taw, 2021. URL <https://www.tawdis.net/>. Acessado em novembro de 2024.
- Cesar Augusto Cusin, Silvana Aparecida Borsetti Gregorio Vidotti, et al. Inclusão digital via acessibilidade webl digital inclusion via web accessibility. *Liinc em Revista*, 5(1), 2009.
- DGE. eMAG - Modelo de Acessibilidade em Governo Eletrônico, 2014. URL <https://emag.governoeletronico.gov.br>. Acessado em novembro de 2024.
- SS Dhenakaran and K Thirugnana Sambanthan. Web crawler-an overview. *International Journal of Computer Science and Communication*, 2(1):265–267, 2011.
- Francisco Filipe, Ivan Miguel Pires, and António Jorge Gouveia. Why web accessibility is important for your institution. *Procedia Computer Science*, 219:20–27, 2023. ISSN 1877-0509. DOI <https://doi.org/10.1016/j.procs.2023.01.259>. URL <https://www.sciencedirect.com/science/article/pii/S1877050923002673>.
- CENTERIS – International Conference on ENTERprise Information Systems / ProjMAN – International Conference on Project MANagement / HCist – International Conference on Health and Social Care Information Systems and Technologies 2022.
- Greg Gay and Cindy Qi Li. Achecker: open, interactive, customizable, web accessibility checking. In *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A)*, W4A '10, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781450300452. DOI [10.1145/1805986.1806019](https://doi.org/10.1145/1805986.1806019). URL <https://doi.org/10.1145/1805986.1806019>.
- Md Abu Kausar, VS Dhaka, and Sanjeev Kumar Singh. Web crawler: a review. *International Journal of Computer Applications*, 63(2):31–36, 2013.
- Miikka Kuutila, Mika Mäntylä, and Päivi Raulamo-Jurvanen. Benchmarking web-testing - selenium versus watir and the choice of programming language and browser, 2016. URL <https://arxiv.org/abs/1611.00578>.
- Marco Manca, Vanessa Palumbo, Fabio Paternò, and Carmen Santoro. The transparency of automatic web accessibility evaluation tools: Design criteria, state of the art, and user perception. *ACM Trans. Access. Comput.*, 16(1), March 2023. ISSN 1936-7228. DOI [10.1145/3556979](https://doi.org/10.1145/3556979). URL <https://doi.org/10.1145/3556979>.

- ONU. Declaração universal dos direitos humanos, 1948. URL <https://brasil.un.org/sites/default/files/2020-09/por.pdf>. Acessado em novembro de 2024.
- Paruchuri Ramya, Vemuri Sindhura, and P. Vidya Sagar. Testing using selenium web driver. In *2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pages 1–7, 2017. DOI [10.1109/ICECCT.2017.8117878](https://doi.org/10.1109/ICECCT.2017.8117878).
- Werner Schweibenz. The "virtual museum": New perspectives for museums to present objects and information using the internet as a knowledge base and communication system. volume 34 of *Schriften zur Informationswissenschaft*, pages 185–200, 1998. DOI [10.5281/ZENODO.4136806](https://doi.org/10.5281/ZENODO.4136806). URL <https://doi.org/10.5281/zenodo.4136806>.
- Kwangsoo Park Shinyong Jung and Yoon Joo Lee. Exploring global trade show website content accessibility and policies worldwide for social inclusion of persons with disabilities. *Journal of Travel & Tourism Marketing*, 41(1):128–141, 2024. DOI [10.1080/10548408.2023.2239845](https://doi.org/10.1080/10548408.2023.2239845). URL <https://doi.org/10.1080/10548408.2023.2239845>.
- MSADJ Singh and Bal Varnica. Web crawler: Extracting the web data. *International Journal of Computer Trends and Technology*, 13(3):132–137, 2014.
- David Sloan. The importance of web accessibility. *Web accessibility: practical advice for the library and information professional*, page 161, 2008.
- Filipe Nunes SOARES. Accesscrawler : Web crawler para avaliar a acessibilidade web, 2023. URL <http://www.monografias.ufop.br/handle/35400000/6051>.
- USA. Americans with disabilities act, 1990. URL <https://www.ada.gov/law-and-regs/ada/>. Acessado em outubro de 2024.
- Beat Vollenwyder, Serge Petralito, Glena H. Iten, Florian Brühlmann, Klaus Opwis, and Elisa D. Mekler. How compliance with web accessibility standards shapes the experiences of users with and without disabilities. *International Journal of Human-Computer Studies*, 170:102956, 2023. ISSN 1071-5819. DOI <https://doi.org/10.1016/j.ijhcs.2022.102956>. URL <https://www.sciencedirect.com/science/article/pii/S1071581922001756>.
- W3C. World wide web consortium (w3c) launches international web accessibility initiative, 1997. URL <https://www.w3.org/press-releases/1997/wai-launch/>. Acessado em outubro de 2024.
- W3C. Web content accessibility guidelines 1.0, 1999. URL <https://www.w3.org/TR/WAI-WEBCONTENT/>. Acessado em novembro de 2024.

W3C. Wcag 2 overview, 2005a. URL

<https://www.w3.org/WAI/standards-guidelines/wcag/>. Acessado em outubro de 2024.

W3C. Introduction to web accessibility, 2005b. URL

<https://www.w3.org/WAI/fundamentals/accessibility-intro/#what>. Acessado em outubro de 2024.

W3C. Web content accessibility guidelines (wcag) 2.0, 2008. URL

<https://www.w3.org/TR/WCAG20/>. Acessado em novembro de 2024.

W3C. Evaluation and report language (earl) overview, 2018a. URL

<https://www.w3.org/WAI/standards-guidelines/earl/>. Acessado em novembro de 2024.

W3C. Web content accessibility guidelines (wcag) 2.1, 2018b. URL

<https://www.w3.org/TR/WCAG21/>. Acessado em novembro de 2024.

W3C. Diretrizes de acessibilidade para conteúdo web (wcag) 2.1, 2018c. URL

<https://www.w3c.br/traducoes/wcag/wcag21-pt-BR/>. Acessado em novembro de 2024.

W3C. Web content accessibility guidelines (wcag) 2.2, 2023. URL

<https://www.w3.org/TR/WCAG22/>. Acessado em novembro de 2024.

W3C. Wcag 3 introduction, 2024. URL

<https://www.w3.org/WAI/standards-guidelines/wcag/wcag3-intro/>. Acessado em novembro de 2024.

WebAIM. Wave, 2021. URL <https://wave.webaim.org/>. Acessado em novembro de 2024.

Apêndices

A Critérios de Sucesso para as diretrizes WCAG

A.1 Diretriz 1.2 Mídias com base em tempo

Critério de Sucesso	Descrição	Nível de conformidade
1.2.1 Apenas Áudio e Apenas Vídeo (Pré-gravado)	Apenas áudio pré-gravado: É fornecida uma alternativa para mídia com base em tempo que apresenta informação equivalente para o conteúdo composto por apenas áudio pré-gravado; Apenas vídeo pré-gravado: É fornecida uma alternativa em mídia com base em tempo ou uma faixa de áudio que apresenta informação equivalente para o conteúdo apenas de vídeo pré-gravado.	A
1.2.2 Legendas (Pré-gravadas)	São fornecidas legendas para todo conteúdo de áudio pré-gravado em mídia sincronizada, exceto quando a mídia for uma alternativa para texto e for claramente identificada como tal.	A
1.2.3 Audiodescrição ou Mídia Alternativa (Pré-gravada)	Uma alternativa para mídia com base ou uma audiodescrição do conteúdo em vídeo pré-gravado é fornecida para mídia sincronizada, exceto quando a mídia é uma alternativa ao texto e for claramente identificada como tal.	A

1.2.4 Legendas (Ao Vivo)	São fornecidas legendas para todo o conteúdo do áudio ao vivo existente em mídia sincronizada.	AA
1.2.5 Audiodescrição (Pré-gravada)	É fornecida audiodescrição para todo o conteúdo de vídeo pré-gravado existente em mídia sincronizada.	AA
1.2.6 Língua de sinais (Pré-gravada)	É fornecida interpretação em língua de sinais para todo o conteúdo de áudio pré-gravado existente em um conteúdo em mídia sincronizada.	AAA
1.2.7 Audiodescrição Estendida (Pré-gravada)	Quando as pausas no áudio de primeiro plano forem insuficientes para permitir que as audiodescrições transmitam o sentido do vídeo, é fornecida uma audiodescrição estendida para todo o vídeo pré-gravado existente no conteúdo em mídia sincronizada.	AAA
1.2.8 Mídia Alternativa (Pré-gravada)	É fornecida uma alternativa para mídia com base em tempo para todo o conteúdo existente em mídia sincronizada pré-gravada e para a todo o conteúdo multimídia composto por apenas vídeo pré-gravado.	AAA
1.2.9 Apenas Áudio (Ao Vivo)	É fornecida uma alternativa para mídia com base em tempo que apresenta informações equivalentes para conteúdo apenas áudio ao vivo .	AAA

A.2 Diretriz 1.3 Adaptável

Critério de Sucesso	Descrição	Nível de conformidade
----------------------------	------------------	------------------------------

1.3.1 Informações e Relações	As informações, a estrutura, e os relacionamentos transmitidos através de apresentação podem ser determinados por meio de código de programação ou estão disponíveis no texto.	A
1.3.2 Sequência com Significado	Quando a sequência na qual o conteúdo é apresentado afeta o seu significado, uma sequência de leitura correta pode ser determinada por meio de código de programação.	A
1.3.3 Características Sensoriais	As instruções fornecidas para compreender e utilizar o conteúdo não dependem somente das características sensoriais dos componentes, tais como forma, cor, tamanho, localização visual, orientação ou som.	A
1.3.4 Orientação	O conteúdo não restringe sua visualização e operação a uma única orientação de exibição, como um retrato ou uma paisagem, a menos que uma orientação de exibição específica seja essencial.	AA
1.3.5 Identificar o Objetivo de Entrada	A finalidade de cada campo de entrada que coleta informações sobre o usuário pode ser determinada de forma programática quando: O campo de entrada serve à finalidade identificada na seção Finalidades de Entrada para Componentes de Interface de Usuário; O conteúdo é implementado por meio da utilização de tecnologias com suporte para identificar o significado esperado para os dados de entrada do formulário.	AA

1.3.6 Identificar o Objetivo	Em conteúdo implementado que utiliza linguagens de marcação, a finalidade dos Componentes de Interface de Usuário, ícones, e regiões podem ser determinados programaticamente.	AAA
------------------------------	--	-----

A.3 Diretriz 1.4 Discernível

Critério de Sucesso	Descrição	Nível de conformidade
1.4.1 Utilização de Cores	A cor não é utilizada como o único meio visual de transmitir informações, indicar uma ação, pedir uma resposta ou distinguir um elemento visual.	A
1.4.2 Controle de Áudio	Se qualquer áudio em uma página web tocar automaticamente durante mais de 3 segundos, deve estar disponível um mecanismo para fazer uma pausa ou parar o áudio, ou um mecanismo para controlar o volume do áudio, independentemente do nível global de volume do sistema deve disponibilizar.	A

1.4.3 Contraste Mínimo	A apresentação visual de texto e imagens de texto tem uma relação de contraste de, no mínimo, 4.5:1, exceto para o seguinte: Texto Ampliado-Texto em tamanho grande e as imagens compostas por texto em tamanho grande têm uma relação de contraste de, no mínimo, 3:1; Texto em plano Secundário-O texto ou imagens de texto que fazem parte de um componente de interface de usuário inativo, que são meramente decorativos, que não estão visíveis para ninguém, ou que são parte de uma imagem que inclui outro conteúdo visual significativo, não têm requisito de contraste; Logotipos - O texto que faz parte de um logotipo ou marca comercial não tem requisito mínimo de contraste.	AA
1.4.4 Redimensionar Texto	Exceto para legendas e imagens de texto, o texto pode ser redimensionado sem tecnologia assistiva até 200 por cento sem perder conteúdo ou funcionalidade.	AA
1.4.5 Imagens de Texto	Se as tecnologias que estiverem sendo utilizadas puderem proporcionar a apresentação visual, é utilizado texto para transmitir informações em vez de imagens de texto exceto para o seguinte: Personalizável - A imagem de texto pode ser visualmente personalizada de acordo com os requisitos do usuário; Essencial-Uma determinada apresentação de texto é essencial para as informações que serão transmitidas.	AA

1.4.6 Contraste (Melhorado)	<p>A apresentação visual do texto e imagens de texto tem uma relação de contraste de, no mínimo, 7:1, exceto para as seguintes situações: Texto Ampliado-Texto em tamanho grande e as imagens compostas por texto em tamanho grande têm uma relação de contraste de, no mínimo, 4.5:1; Texto em plano Secundário - O texto ou as imagens de texto que fazem parte de um componente de interface de usuário inativo, que são meramente decorativos, que não estão visíveis para ninguém, ou que fazem parte de uma imagem que inclui outro conteúdo visual significativo, não têm requisito de contraste; Logotipos - O texto que faz parte de um logotipo ou marca comercial não tem requisito de contraste.</p>	AAA
1.4.7 Áudio de fundo baixo ou sem Áudio de fundo	<p>Para conteúdo composto por apenas áudio pré-gravado que (1) contenha essencialmente fala em primeiro plano, (2) não seja um CAPTCHA de áudio ou logotipo de áudio, e (3) não seja vocalização com o objetivo de ser, essencialmente, expressão musical, tal como cantar ou fazer batidas, no mínimo, uma das seguintes afirmações é verdadeira: Sem Fundo-O áudio não contém sons de fundo; Desligar - Os sons de fundo podem ser desligados; 20 dB - Os sons de fundo são, no mínimo, 20 decibéis mais baixos que o conteúdo da voz em primeiro plano, com a exceção de sons ocasionais que duram apenas um ou dois segundos.</p>	AAA

1.4.8 Apresentação Visual	<p>1.4.8 Apresentação Visual: Para a apresentação visual de blocos de texto, um mecanismo está disponível para se obter o seguinte: As cores do primeiro plano e do plano de fundo podem ser selecionadas pelo usuário; A largura não tem mais do que 80 caracteres ou glifos (40 se CJK); O texto não é justificado (alinhado a ambas as margens esquerda e direita); O espaçamento entre linhas (principal) tem, no mínimo, um espaço e meio nos parágrafos, e o espaçamento entre parágrafos é, no mínimo, 1,5 vezes maior do que o espaçamento entre linhas; O texto pode ser redimensionado sem tecnologia assistiva até 200 por cento, de um modo que o usuário não necessite rolar horizontalmente para ler uma linha de texto em uma janela em tela cheia.</p>	AAA
1.4.9 Imagens de Texto sem exceção	<p>As imagens de texto só são utilizadas por questões meramente decorativas ou quando uma determinada apresentação de texto é essencial para a informação que está sendo transmitida.</p>	AAA
1.4.10 Realinhar	<p>O conteúdo pode ser apresentado sem perda de informação ou funcionalidade e sem exigir rolagem em duas dimensões para: Conteúdo de rolagem vertical com largura equivalente a 320 pixels CSS; Conteúdo de rolagem horizontal com altura equivalente a 256 pixels CSS. Exceto por partes do conteúdo que requerem layout bidimensional para uso ou significado.</p>	AA

1.4.11 Contraste Não textual	A apresentação visual a seguir tem um relação de contraste de pelo menos 3:1 contra cor(es) adjacente(s): Componentes de Interface de Usuário -Informações visuais necessárias para identificar componentes de interface de usuário e estados,exceto para componentes inativos ou quando a aparência do componente é determinada pelo agente de usuário e não é modificada pelo autor; Objetos Gráficos - Partes de gráficos necessários para entender o conteúdo, exceto quando uma apresentação específica de gráficos é essencial para as informações que estão sendo transmitidas.	AA
1.4.12 Espaçamento de Texto	No conteúdo implementado utilizando linguagens de marcação que suportam as seguintes propriedades de estilo de texto, nenhuma perda de conteúdo ou funcionalidade ocorre pela configuração de todos os itens a seguir e pela alteração de nenhuma outra propriedade de estilo: Altura da linha (espaçamento entre linhas) de pelo menos 1,5 vezes o tamanho da fonte; Espaçamento dos parágrafos seguintes de pelo menos 2 vezes o tamanho da fonte; Espaçamento de letras (rastreamento) de pelo menos 0,12 vezes o tamanho da fonte; Espaçamento de palavras de pelo menos 0,16 vezes o tamanho da fonte. Exceção: linguagens humanas e scripts que não fazem uso de uma ou mais dessas propriedades de estilo de texto em textos escritos podem conformar-se utilizando somente propriedades existentes para essa combinação de linguagem e script.	AA

1.4.13 Conteúdo em foco por mouse ou teclado	<p>Ao receber e então remover o ponteiro do mouse ou do foco do teclado aciona o conteúdo adicional o tornando visível e então oculto, o itens seguintes são verdadeiros: Descartável - Um mecanismo está disponível para descartar conteúdo adicional sem mover o ponteiro do mouse ou o foco do teclado, a menos que o conteúdo adicional comunique um erro de entrada ou não obscureça ou substitua outro conteúdo; Flutuante (passar o cursor sobre) - Se o ponteiro do mouse puder acionar o conteúdo adicional ao passar sobre ele, então o ponteiro poderá ser movido sobre o conteúdo adicional sem que o conteúdo adicional desapareça; Persistente - O conteúdo adicional permanece visível até que o acionador do mouse ou do foco seja removido, o usuário o descarte ou sua informação não seja mais válida. Exceção: a apresentação visual de conteúdo adicional é controlada pelo agente de usuário e não é modificada pelo autor.</p>	AA
--	---	----

A.4 Diretriz 2.1 Acessível por Teclado

Critério de Sucesso	Descrição	Nível de conformidade
2.1.1 Teclado	<p>Toda a funcionalidade do conteúdo é operável através de uma interface de teclado sem requerer temporizações específicas para digitação individual, exceto quando a função subjacente requer entrada de dados que dependa da cadeia de movimento do usuário e não apenas dos pontos finais.</p>	A

2.1.2 Sem Bloqueio do Teclado	Se o foco do teclado puder ser movido para um componente da página utilizando uma interface de teclado, então o foco pode ser retirado desse componente utilizando apenas uma interface de teclado e, se for necessário mais do que as setas do cursor ou tabulação ou outros métodos de saída normalmente utilizados, o usuário deve ser informado sobre o método para retirar o foco.	A
2.1.3 Teclado Sem Exceção	Toda a funcionalidade do conteúdo é operável através de uma interface de teclado sem requerer temporizações específicas para digitação individual.	AAA
2.1.4 Atalhos de teclado por caractere	Se um atalho de teclado é implementado no conteúdo utilizando apenas letras (incluindo letras maiúsculas e minúsculas), pontuação, números ou símbolos, então ao menos um dos itens é verdadeiro: Desativar - Um mecanismo está disponível para desativar o atalho; Remapear - Um mecanismo está disponível para remapear o atalho para utilizar um ou mais caracteres de teclado não imprimíveis (por exemplo, Ctrl, Alt etc); Ativo apenas quando recebe o foco - O atalho do teclado para um componente de interface de usuário está ativo apenas quando esse componente tem foco.	A

A.5 Diretriz 2.2 Tempo Suficiente

Critério de Sucesso	Descrição	Nível de conformidade
----------------------------	------------------	------------------------------

<p>2.2.1 Ajustável por Temporização</p>	<p>Para cada limite de tempo definido pelo conteúdo, no mínimo, uma das seguintes afirmações é verdadeira: Desligar - O usuário pode desligar o limite de tempo antes de atingi-lo; ou Ajustar - O usuário pode ajustar o limite de tempo antes de atingi-lo, num intervalo de, no mínimo, dez vezes mais do que a configuração padrão; ou Prolongar - O usuário é avisado antes de o tempo expirar e tem, no mínimo, 20 segundos para prolongar o limite de tempo com uma simples ação (por exemplo, "pressionar a barra de espaços"), e o usuário pode prolongar o limite de tempo, no mínimo, dez vezes; ou Exceção em Tempo Real - O limite de tempo é uma parte necessária de um evento em tempo real (por exemplo, um leilão), e não é possível nenhuma alternativa ao limite de tempo; ou Exceção Essencial - O limite de tempo é essencial e prolongá-lo invalidaria a atividade; ou Exceção de 20 Horas - O limite de tempo é superior a 20 horas.</p>	<p>A</p>
---	---	----------

2.2.2 Colocar em Pausa, Parar, Ocultar	<p>Para informações em movimento, em modo intermitente, em deslocamento ou em atualização automática, todas as seguintes afirmações são verdadeiras: Em movimento, em modo intermitente, em deslocamento - Para quaisquer informações em movimento, em modo intermitente ou em deslocamento, que (1) sejam iniciadas automaticamente, (2) durem mais de cinco segundos, e (3) sejam apresentadas em paralelo com outro conteúdo, existe um mecanismo para o usuário colocar em pausa, parar, ou ocultar as mesmas, a menos que o movimento, o modo intermitente ou o deslocamento façam parte de uma atividade, na qual sejam essenciais; e Em atualização automática - Para quaisquer informações em atualização automática, que (1) sejam iniciadas automaticamente e (2) sejam apresentadas em paralelo com outro conteúdo, existe um mecanismo para o usuário colocar em pausa, parar ou ocultar as mesmas, ou controlar a frequência da atualização, a menos que a atualização automática faça parte de uma atividade, onde é essencial.</p>	A
2.2.3 Sem Temporização	<p>A temporização não é uma parte essencial do evento ou da atividade apresentada pelo conteúdo, exceto para mídia sincronizada não interativa e eventos em tempo real.</p>	AAA
2.2.4 Interrupções	<p>As interrupções podem ser adiadas ou suprimidas pelo usuário, exceto interrupções que envolvam uma emergência.</p>	AAA
2.2.5 Nova Autenticação	<p>Quando uma seção autenticada expira, o usuário pode continuar a atividade sem perder dados após a nova autenticação.</p>	AAA

2.2.6 Limites de Tempo	Os usuários são avisados sobre a duração de qualquer inatividade do usuário que possa causar perda de dados, a menos que os dados sejam preservados quando o usuário não realizar nenhuma ação por mais de 20 horas.	AAA
------------------------	--	-----

A.6 Diretriz 2.3 Convulsões e Reações Físicas

Critério de Sucesso	Descrição	Nível de conformidade
2.3.1 Três Flashes ou Abaixo do Limite	As páginas web não incluem nenhum conteúdo que pisque mais de três vezes no período de um segundo, ou o flash encontra-se abaixo dos limites de flash universal e flash vermelho.	A
2.3.2 Três Flashes	As páginas web não incluem qualquer conteúdo que pisca mais de três vezes no período de um segundo.	AAA
2.3.3 Animação de Interações	A animação de movimento acionada por interação pode ser desativada, a menos que a animação seja essencial para a funcionalidade ou para as informações transmitidas.	AAA

A.7 Diretriz 2.4 Navegável

Critério de Sucesso	Descrição	Nível de conformidade
----------------------------	------------------	------------------------------

2.4.1 Ignorar Blocos	Um mecanismo está disponível para ignorar blocos de conteúdo que são repetidos em várias páginas web.	A
2.4.2 Página com Título	As páginas web têm títulos que descrevem o tópico ou a finalidade.	A
2.4.3 Ordem de Foco	Se uma página web puder ser navegada de forma sequencial e as sequências de navegação afetarem o significado ou a operação, os componentes que podem ser focados recebem o foco em uma ordem que preserva o significado e a operabilidade.	A
2.4.4 Finalidade do Link Em contexto	A finalidade de cada link pode ser determinada a partir do link sozinho ou a partir do texto do link em conjunto com seu respectivo contexto do link determinado por meio de código de programação, exceto quando a finalidade do link for ambígua para os usuários em geral.	A
2.4.5 Várias Formas	Está disponível mais de uma forma para localizar uma página web em um conjunto de páginas web exceto quando a Página Web for o resultado, ou uma etapa, de um processo.	AA
2.4.6 Cabeçalhos e Rótulos	Os cabeçalhos e os rótulos descrevem o tópico ou a finalidade.	AA
2.4.7 Foco Visível	Qualquer interface de usuário operável por teclado dispõe de um modo de operação onde o indicador de foco do teclado está visível.	AA

2.4.8 Localização	Informação sobre a localização do usuário está disponível em um conjunto de páginas web.	AAA
2.4.9 Finalidade do Link (Apenas o Link)	Um mecanismo está disponível para permitir que a finalidade de cada link seja identificada a partir apenas do texto do link, exceto quando a sua finalidade for ambígua para os usuários em geral.	AAA
2.4.10 Cabeçalhos da Sessão	Os cabeçalhos da seção são utilizados para organizar o conteúdo.	AAA

A.8 Diretriz 2.5 Modalidades de Entrada

Critério de Sucesso	Descrição	Nível de conformidade
2.5.1 Gestos de Acionamento	Todas as funcionalidades que utilizam gestos multiponto ou baseados em caminhos para operação podem ser operadas com um único ponteiro sem um gesto baseado em caminho, a menos que um gesto multiponto ou baseado em caminho seja essencial.	A

2.5.2 Cancelamento de Acionamento	Para funcionalidade que pode ser operada usando um único ponteiro, pelo menos um dos seguintes itens é verdadeiro: Não Down-Event - O down-event do ponteiro não é utilizado para executar nenhuma parte da função; Interromper ou Desfazer - A conclusão da função está no up-event, e um mecanismo está disponível para anular a função antes da conclusão ou para desfazer a função após a conclusão; Ativação Reversa - O up-event anula qualquer resultado do down-event precedente. Essencial - É essencial completar a função sobre o down-event.	A
2.5.3 Rótulo em Nome Acessível	Para componentes de interface de usuário com rótulos que incluem texto ou imagens de texto, o nome contém o texto presente visualmente.	A
2.5.4 Atuação em Movimento	A funcionalidade que pode ser operada pelo movimento do dispositivo ou movimento do usuário pode também ser operada por componentes de interface de usuário e a resposta ao movimento pode ser desativada para evitar atuação acidental, exceto quando: Interface Suportada - O movimento é utilizado para operar a funcionalidade por meio de uma interface com suporte a acessibilidade; Essencial - O movimento é essencial para a função e fazendo isso invalidaria a atividade.	A

2.5.5 Tamanho da Área Clicável	O tamanho do alvo para entradas de ponteiro é pelo menos 44 por 44 pixels CSS exceto quando: Equivalente - O alvo está disponível por meio de um link ou controle equivalente na mesma página, com pelo menos 44 por 44 pixels CSS; Inline - O alvo está em uma sentença ou bloco de texto; Controle de Agente de Usuário - O Tamanho da Área Clicável é determinado pelo agente de usuário e não é modificado pelo autor; Essencial - Uma apresentação específica do alvo é essencial à informação que está sendo transmitida.	AAA
2.5.6 Mecanismos de Entrada Simultâneos	O conteúdo da Web não restringe o uso de modalidades de entrada disponíveis em uma plataforma, exceto quando a restrição é essencial, necessária para garantir a segurança do conteúdo ou obrigada a respeitar as configurações do usuário.	AAA

A.9 Diretriz 3.1 Legível

Critério de Sucesso	Descrição	Nível de conformidade
3.1.1 Idioma da Página	O idioma humano pré-definido de cada página web pode ser determinado por meio de código de programação.	A
3.1.2 Idioma das Partes	O idioma de cada passagem ou frase no conteúdo pode ser determinado por meio de código de programação exceto para nomes próprios, termos técnicos, palavras de idioma indeterminado e palavras ou frases que se tornaram parte do vernáculo do texto que as envolve.	AA

3.1.3 Palavras Incomuns	Um mecanismo para identificar definições específicas de palavras ou expressões utilizadas de uma forma restrita e incomum está disponível, incluindo expressões idiomáticas e jargões.	AAA
3.1.4 Abreviaturas	Está disponível um mecanismo para identificar a forma expandida ou o significado das abreviaturas.	AAA
3.1.5 Nível de Leitura	Quando o texto exigir uma capacidade de leitura mais avançada do que o nível de educação secundário inferior (equivalente no Brasil aos últimos anos do ensino fundamental), após a remoção dos nomes próprios e títulos adequados, um conteúdo suplementar, ou uma versão que não exija uma capacidade de leitura mais avançada do que o nível de educação secundário inferior (equivalente no Brasil aos últimos anos do ensino fundamental) está disponível.	AAA
3.1.6 Pronúncia	Um mecanismo está disponível para identificar a pronúncia específica de palavras, onde o significado das mesmas, no contexto, é ambíguo se a pronúncia não for conhecida.	AAA

A.10 Diretriz 3.2 Previsível

Critério de Sucesso	Descrição	Nível de conformidade
3.2.1 Em Foco	Quando qualquer componente de interface do usuário recebe o foco, não inicia uma alteração de contexto.	A

3.2.2 Em Entrada	Alterar a definição de um componente de interface de usuário não provoca, automaticamente uma alteração de contexto a menos que o usuário tenha sido avisado sobre esse comportamento antes de utilizar o componente.	A
3.2.3 Navegação Consistente	Os mecanismos de navegação que são repetidos em múltiplas páginas web dentro de um conjunto de páginas web ocorrem na mesma ordem relativa a cada vez que são repetidos, a menos que seja iniciada uma alteração pelo usuário.	AAA
3.2.4 Identificação Consistente	Os componentes que têm a mesma funcionalidade em um conjunto de páginas web são identificados de forma consistente.	AA
3.2.5 Alteração Mediante Solicitação	As alterações de contexto são iniciadas apenas a pedido do usuário, ou um mecanismo para desativar essas alterações está disponível.	AAA

A.11 Diretriz 3.3 Assistência de Entrada

Critério de Sucesso	Descrição	Nível de conformidade
3.3.1 Identificação do Erro	Se um erro de entrada for automaticamente detectado, o item que apresenta erro é identificado e o erro é descrito para o usuário em texto.	A

3.3.2 Rótulos ou Instruções	Rótulos ou instruções são fornecidos quando o conteúdo exigir a entrada de dados por parte do usuário.	A
3.3.3 Sugestão de Erro	Se um erro de entrada for automaticamente detectado e forem conhecidas sugestões de correção, então as sugestões são fornecidas ao usuário, a menos que coloque em risco a segurança ou o propósito do conteúdo.	AA
3.3.4 Prevenção de Erros (Legal, Financeiro, Dados)	Para páginas web que façam com que ocorram responsabilidades jurídicas ou transações financeiras para o usuário, que modificam ou eliminam dados controláveis pelo usuário em sistemas de armazenamento de dados, ou que enviem respostas de teste do usuário, no mínimo, uma das seguintes afirmações é verdadeira: Reversível - Os envios de informações são reversíveis. Verificado - Os dados introduzidos pelo usuário são verificados quanto à existência de erros de entrada e é oferecida ao usuário uma oportunidade de corrigi-los. Confirmado - Um mecanismo está disponível para rever, confirmar e corrigir as informações antes de finalizar o envio.	AA
3.3.5 Ajuda	Está disponível ajuda contextual.	AAA
3.3.6 Prevenção de Erros (Todos)	Para páginas web que exijam que o usuário envie informações, no mínimo, uma das seguintes afirmações é verdadeira: Reversível - As submissões são reversíveis. Verificado - Os dados introduzidos pelo usuário são verificados quanto à existência de erros de entrada e é oferecida ao usuário uma oportunidade de corrigi-los. Confirmado - Um mecanismo está disponível para rever, confirmar e corrigir as informações antes de finalizar o envio.	AAA

A.12 Diretriz 4.1 Compatível

Critério de Sucesso	Descrição	Nível de conformidade
4.1.1 Análise	No conteúdo implementado utilizando linguagens de marcação, os elementos dispõem de tags completas de início e de fim, os elementos são aninhados de acordo com as respectivas especificações, os elementos não contêm atributos duplicados, e quaisquer IDs são exclusivos, exceto quando as especificações permitem estas características.	A
4.1.2 Nome, Função, Valor	Para todos os componentes de interface de usuário (incluindo, mas não se limitando a: elementos de formulário, links e componentes gerados por scripts), o nome e a função podem ser determinados por meio de código de programação; os estados, as propriedades e os valores, que possam ser definidos pelo usuário, podem ser definidos por meio de código de programação; e a notificação sobre alterações destes itens está disponível para os agentes de usuário, incluindo as tecnologias assistivas.	A
4.1.3 Mensagens de Status	Em conteúdo implementado que utiliza linguagens de marcação, as mensagens de status podem ser determinado por meio de código de programação por meio da função ou de propriedades, de modo que possam ser apresentadas ao usuário por tecnologias assistivas sem receber foco.	AA

B Códigos das principais funções utilizadas na Ferramenta

B.1 Função queryString()

Função responsável por retornar a string XPATH de consulta.

```
1 def queryString(site):
2     return f'//a[(contains(@href, "{site}") or starts-with(@href, "/") or
        starts-with(@href, "#")) and not(contains(@href,"jpg") or
        contains(@href,"youtube") or contains(@href,"youtu.be") or
        contains(@href,"instagram") or contains(@href,"facebook") or
        contains(@href,"linkedin") or contains(@href,"tiktok") or
        contains(@href,"mailto") or contains(@href,"jpeg") or
        contains(@href,"png") or contains(@href,"mp3") or
        contains(@href,"twitter") or contains(@href,"x.com") or
        contains(@href,"google") or contains(@href,"wikipedia") or
        contains(@href,"pdf") or contains(@href,"JPEG") or
        contains(@href,"PNG")or contains(@href,"JPG") or contains(@href,"PDF"))]'
```

B.2 Função getWebsiteScores()

Função responsável por iniciar a busca por mais páginas e consolidar os resultados.

```
1     global count, finalScore, placeholder, AnalyzedSite, driver
2     print("Iniciando Anlise...")
3
4     try:
5         driver.switch_to.window(driver.window_handles[0])
6         driver.get(site)
7         site = driver.current_url
8         print(f'domnio real: {site}')
9     except:
10        print("Site no encontrado.")
11        AnalyzedSite.markdown("### Site no encontrado, Recarregue a pgina e tente
            novamente")
12        driver.quit()
13        return
14
15    points = 0
16    AnalyzedSite.markdown(f"### Analisando pgina inicial {site}")
17    result = getPageScore(site)
18    if result == -1:
19        return result
20    driver.switch_to.window(driver.window_handles[0])
21    print(result)
22    if result != 0:
```

```
23         ScoresTable[' Pontuao '].append(result)
24         points += result
25         count += 1
26     WebDriverWait(driver, 0.1)
27     elementList = driver.find_elements(By.XPATH, queryString(site))
28     print(elementList)
29     linkList = list(set(map(getLinkFromElement, elementList)))
30     driver.switch_to.window(driver.window_handles[2])
31     linkList = searchThroughWebsite(linkList, site)
32     driver.switch_to.window(driver.window_handles[0])
33     for item in linkList:
34         placeholder.markdown(f"### :blue-background[Pginas encontradas:
35                               {len(linkList) + 1} Pginas analisadas: {count}
36                               :hourglass_flowing_sand: ]")
37     try:
38         global firstPage
39         firstPage = False
40         print(item)
41         if item == site:
42             continue
43         driver.switch_to.window(driver.window_handles[2])
44         driver.get(item)
45         html = driver.page_source
46         AnalyzedSite.markdown(f"### Analisando {item}")
47         result = getPageScore(html, site=item)
48         driver.switch_to.window(driver.window_handles[0])
49         if result != 0:
50             ScoresTable[' Pontuao '].append(result)
51             print(result)
52             points += result
53             count += 1
54     except Exception as error:
55         print(error)
56         continue
57 if count == 0:
58     finalScore = 0
59 else:
60     finalScore = points / count
61 print(f"Mdia: {finalScore}")
62 print(f"Pginas Verificadas: {count}")
```

```
62     print (ScoresTable)
63     df = pd.DataFrame(data=ScoresTable)
64     df2 = df.dropna()
65     df2.to_csv(f"{site.split('.') [1]}.csv")
66     driver.quit()
67     return df2
```

B.3 Função getPageScore()

Função responsável por retornar os resultados de análise de acessibilidade de cada página.

```
1     def getPageScore(html, site = ''):
2         global driver, actions
3         practicesList = []
4         tableList = []
5         info = []
6         driver.switch_to.window(driver.window_handles[1])
7         driver.get('https://accessmonitor.acessibilidade.gov.pt/')
8         HtmlMode = WebDriverWait(driver,
9             10).until(EC.presence_of_element_located((By.XPATH,
10                 '//button[contains(@data-rr-ui-event-key, "tab2")]'))))
11         HtmlMode.click()
12         search = WebDriverWait(driver,
13             10).until(EC.presence_of_element_located((By.ID, "html")))
14         search.clear()
15         if firstPage:
16             driver.switch_to.window(driver.window_handles[0])
17             pyperclip.copy(driver.page_source)
18             site = html
19         else:
20             pyperclip.copy(html)
21             driver.switch_to.window(driver.window_handles[1])
22             search.send_keys(Keys.CONTROL, 'v')
23             sendButton = WebDriverWait(driver,
24                 10).until(EC.presence_of_element_located((By.XPATH,
25                     '//button[contains(@id, "btn-html")]'))))
26             sendButton.click()
27         try:
28             score = WebDriverWait(driver, 60).until(
29                 EC.presence_of_element_located((By.TAG_NAME, "svg")))
30             scoreImage = WebDriverWait(driver,
```

```

10).until(EC.presence_of_element_located((By.XPATH,
    '//div[contains(@class,"d-flex flex-row mt-5 mb-5
    justify-content-between container_uri_chart")]'))))
26 actions.move_to_element(scoreImage).perform()
27 scoreImage = scoreImage.screenshot_as_png
28 score = str(score.text).split('\n')[1]
29 results = WebDriverWait(driver, 10).until(
30     EC.presence_of_all_elements_located((By.XPATH,
        '//table[contains(@class,"table table_primary ")]//tbody//tr'))))
31 infoElements = WebDriverWait(driver,
    10).until(EC.presence_of_all_elements_located((By.XPATH,
        '//div[contains(@class,"size_container d-flex flex-column
        gap-4")]//div[contains(@class,"d-flex flex-column")]'))))
32 for element in infoElements:
33     actions.move_to_element(element).perform()
34     info.append(element.screenshot_as_png)
35 title = WebDriverWait(driver,
    10).until(EC.presence_of_element_located((By.XPATH,
        '//tr[contains(@class,"mobile_table")]'))))
36 actions.move_to_element(title).perform()
37 title = title.screenshot_as_png
38 tableList.append(title)
39 tableElements = WebDriverWait(driver, 10).until(
40     EC.presence_of_all_elements_located((By.XPATH,
        '//table[contains(@class,"table table-bordered table-alternative
        ")]//tbody//tr'))))
41 for element in tableElements:
42     actions.move_to_element(element).perform()
43     tableList.append(element.screenshot_as_png)
44 for element in results:
45     actions.move_to_element(element).perform()
46     practicesList.append(element.screenshot_as_png)
47 if firstPage:
48     ScoresTable['Pgina'].append('Inicio')
49     ScoresTable['Link'].append(site)
50     imagesList['Inicio'] = practicesList
51     scoreList['Inicio'] = scoreImage
52     overviewList['Inicio'] = tableList
53     infoList['Inicio'] = info
54 else:
55     driver.switch_to.window(driver.window_handles[2])

```

```
56         if driver.title in imagesList.keys():
57             repeatList.append(driver.title)
58             repeat = repeatList.count(driver.title) + 1
59             repeatTitle = f'{driver.title}-{repeat}'
60             ScoresTable['Pgina'].append(repeatTitle)
61             ScoresTable['Link'].append(site)
62             imagesList[repeatTitle] = practicesList
63             scoreList[repeatTitle] = scoreImage
64             overviewList[repeatTitle] = tableList
65             infoList[repeatTitle] = info
66         else:
67             ScoresTable['Pgina'].append(driver.title)
68             ScoresTable['Link'].append(site)
69             imagesList[driver.title] = practicesList
70             scoreList[driver.title] = scoreImage
71             overviewList[driver.title] = tableList
72             infoList[driver.title] = info
73
74         print(f'score da pgina: {score}')
75         return float(score)
76     except Exception as error:
77         if firstPage:
78             return -1
79         print(error)
80         return 0
```

B.4 Função searchThroughWebsite()

Função responsável por buscar mais páginas através da lista de páginas encontrada na primeira página do website.

```
1     def searchThroughWebsite(linkList,site):
2         global placeholder,pageCount,AnalyzedSite, driver
3         removeList = []
4         for link in linkList:
5             try:
6                 req = requests.get(link)
7                 print(req.status_code)
8             except:
9                 continue
10            if req.status_code == 200 and link != site:
```

```
11         driver.get(link)
12         current_url = driver.current_url
13         if site not in current_url:
14             removeList.append(link)
15             continue
16         try:
17             AnalyzedSite.markdown(f"### procurando mais pginas em {link}")
18             elementList = WebDriverWait(driver, 0.5).until(
19                 EC.presence_of_all_elements_located((By.XPATH, queryString(site))))
20         except:
21             continue
22         FoundLinks = set(map(getLinkFromElement, elementList))
23         linkSet = set(linkList)
24         FoundLinks = list(FoundLinks - linkSet)
25         linkList.extend(FoundLinks)
26         print(f'links novos encontrados:{len(FoundLinks)} links totais
27               assimilados: {len(linkList)}')
28         pageCount = len(linkList) - len(removeList)
29         placeholder.markdown(f"### :blue-background[Procurando Pginas:
30                               {pageCount + 1} :mag_right: ]")
31     for item in removeList:
32         try:
33             linkList.remove(item)
34         except:
35             continue
36     return linkList
```

C Funções utilizadas no estudo de caso

C.1 Função getMuseumsSite()

```
1 def getMuseumsSite():
2     print("Iniciando busca...")
3     driver.get(MuseusBR)
4     pages = WebDriverWait(driver, 10).until(EC.presence_of_all_elements_located(
5         (By.XPATH, "//a[@role = 'button'] and (@class = 'pagination-link')]))
6     pages = pages[-1].text
7     for _ in range(int(pages)):
8         time.sleep(3)
9         museums = driver.find_elements(By.CLASS_NAME, "tainacan-masonry-item")
```

```
10     for museum in museums:
11         req = requests.get(museum.get_attribute('href'))
12         if req.status_code == 200:
13             content = req.content
14             soup = BeautifulSoup(content, 'html.parser')
15             try:
16                 link = soup.find(name='blockquote').find(name='a')['href']
17                 name = soup.find(name='h1', attrs={'class':
18                     'page-title'}).get_text()
19                 table['Museum'].append(name)
20                 table['Website'].append(link)
21             except:
22                 link = soup.find(name='a', attrs={'target': '_blank'})['href']
23                 name = soup.find(name='h1', attrs={'class':
24                     'page-title'}).get_text()
25                 print(f'{name}\n{link}')
26                 table['Museum'].append(name)
27                 table['Website'].append(link)
28         try:
29             next = WebDriverWait(driver, 10).until(
30                 EC.presence_of_element_located((By.XPATH, "//a[(@role = 'button')
31                     and (@class = 'pagination-link pagination-next
32                     pagination-next')]")))
33             driver.execute_script("window.scrollTo(0, 2800);")
34             time.sleep(0.5)
35             next.click()
36         except:
37             pass
38     df = pd.DataFrame(data=table)
39     df.to_csv('MuseumsWebsites.csv')
```
