

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE SÃO PAULO
THAMIRIS CRISTINA ANDRADE VICENTE

**Verificação da Acessibilidade em Websites: Um Estudo de Caso Sobre o Site
Armazém do Campo On-line**

MBA EM ENGENHARIA DE SOFTWARE

São Paulo
2024

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE SÃO PAULO

THAMIRIS CRISTINA ANDRADE VICENTE

**Verificação da Acessibilidade em Websites: Um Estudo de Caso Sobre o Site
Armazém do Campo On-line**

Trabalho de Conclusão de Curso apresentado à Banca Avaliadora da Pontifícia Universidade Católica de São Paulo, como exigência parcial para obtenção do título de especialista em Engenharia de Software, sob a orientação do Prof. Dr. Julio Arakaki.

São Paulo

2024

Autorizo exclusivamente para fins acadêmicos e científicos, a reprodução total ou parcial deste Trabalho de Conclusão de Curso por processos de fotocopiadoras ou eletrônicos.

Assinatura _____

Data _____

e-mail _____

Vicente, Thamiris Cristina Andrade

Verificação da Acessibilidade em Websites: Um Estudo de Caso Sobre o Site Armazém do Campo On-line. / Thamiris Cristina Andrade Vicente. -- São Paulo: [s.n.], 2024.

69p. il. ; 30 cm.

Orientador: Prof. Dr. Julio Arakaki.

Trabalho de Conclusão de Curso (Especialização)-- Pontifícia Universidade Católica de São Paulo, Especialização em Engenharia de Software, 2024.

1. Acessibilidade. 2. Usabilidade. 3. Casos de teste. I. Arakaki, Julio. II. Pontifícia Universidade Católica de São Paulo, Trabalho de Conclusão de Curso, Especialização em Engenharia de Software. III. Título.

À minha tia Landa, que nunca pôde mostrar todo seu talento para o mundo por puro preconceito da sociedade.

Agradecimentos

Gostaria de expressar minha gratidão ao meu marido Vinicius, cujo apoio e compreensão foram fundamentais durante todo o processo de elaboração desta monografia. Sua paciência e encorajamento me proporcionaram a força necessária para enfrentar os desafios e seguir em frente, mesmo nos momentos mais difíceis.

Agradeço também à minha família, que sempre incentivou minha busca pelo conhecimento e valorizou a importância do estudo em minha vida. Seu suporte emocional e motivação foram essenciais para a conclusão deste trabalho.

Não poderia deixar de mencionar meu orientador Julio Arakaki, cuja mentoria, sabedoria e conselhos foram imprescindíveis para a realização desta monografia.

Adicionalmente, agradeço à minha psicóloga Tauany, cuja acompanhamento foi fundamental para minha organização e para lidar com a ansiedade, ajudando a manter o equilíbrio emocional necessário durante todo o processo.

Por fim, reconheço meu próprio esforço e dedicação ao longo deste percurso. A perseverança e o compromisso foram cruciais para alcançar este objetivo, e cada momento de sacrifício refletiu o compromisso que assumi comigo mesma e com este trabalho.

Resumo

Esta monografia aborda a temática da acessibilidade na engenharia de *software*, fundamentando-se em um estudo de caso real. O trabalho passa por uma análise das documentações existentes sobre acessibilidade, com o objetivo de identificar boas práticas e padrões reconhecidos. Também é apresentada uma lista de etapas a serem seguidas para validar a acessibilidade de um *software*, oferecendo um guia prático e aplicável para desenvolvedores de aplicações *web*. Além disso, a pesquisa inclui sugestões de ferramentas de análise que podem ser utilizadas para a verificação de acessibilidade, e complementa com etapas de avaliação manual, pensando em cobrir a maior gama possível de barreiras de usabilidade por pessoas com deficiência.

Palavras-chave: Acessibilidade; usabilidade; estudo de caso; cenários de teste; casos de teste; testes manuais; qualidade de *software*; requisitos não funcionais.

Abstract

This monograph addresses the topic of accessibility in software engineering, based on a real case study. The work involves an analysis of existing documentation on accessibility, aiming to identify recognized best practices and standards. It also presents a list of steps to be followed to validate software accessibility, providing a practical and applicable guide for web application developers. Furthermore, the research includes suggestions for analysis tools that can be used for accessibility validation. It complements it with manual evaluation steps, intending to cover the widest possible range of usability barriers for people with disabilities.

Keywords: Accessibility; usability; case study; test scenarios; test cases; manual testing; software quality; non-functional requirements.

Lista de Ilustração

Figura 1 - Relação de impedimento X permanência do guia: <i>Inclusive</i> 101	17
Figura 2 - Imagem de teclado para uso de braile	19
Figura 3 - Captura de tela de trecho de código que utiliza atributos do tipo ARIA	23
Figura 4 - Comparativo de <i>HTML</i> não semântico e <i>HTML</i> semântico	26
Figura 5 - Comparativo entre código <i>HTML</i> , conteúdo <i>HTML</i> renderizado na <i>DOM</i> e conteúdo <i>HTML</i> renderizado na <i>AOM</i>	28
Figura 6 - Captura de tela do projeto clonado	31
Figura 7 - Captura de tela do validador Markup Validation Service antes de começar as análises e alterações no código	31
Figura 8 - Captura de tela do <i>plugin</i> WAVE antes de iniciar a análise	38
Figura 9 - Captura de tela da ferramenta de acessibilidade da Adobe.....	40
Figura 10 - Captura de tela do código do carrossel de imagens	48
Figura 11 - Captura de tela de item sinalizado como “em foco”	55
Figura 12 - Captura de tela do site Armazém do Campo <i>On-line</i> analisado através da ferramenta NVDA	56
Figura 13 - Captura de tela da mensagem de sucesso	58
Figura 14 - Captura de tela do controle de conteúdo do carrossel.....	60
Figura 15 - Captura de tela do <i>site</i> analisado utilizando o <i>zoom</i> em 200%.....	60
Figura 16 - Captura de tela de mensagem de erro.....	61

Lista de abreviaturas e siglas

AOM	Accessibility Object Model
CNPJ	Cadastro Nacional de Pessoa Jurídica
COVID-19	Corona Virus Disease 2019
CPF	Cadastro de Pessoa Física
CSS	Cascading Style Sheets
DOM	Document Object Model
EPP	Empresa de Pequeno Porte
eMag	Modelo de Acessibilidade em Governo Eletrônico
GAIA	Guidelines for Accessible Interfaces for people with Autism ou Guia de Acessibilidade de Interfaces Web com foco em aspectos do Autismo
GIF	Graphics Interchange Format
HTML	HyperText Markup Language
IBGE	Instituto Brasileiro de Geografia e Estatística
JUCESP	Junta Comercial do Estado de São Paulo
LIBRAS	Língua Brasileira de Sinais
MDN Web Docs	Mozilla Developer Network Web Documentation
NVDA	NonVisual Desktop Access
PNS	Pesquisa Nacional de Saúde
SVG	Scalable Vector Graphic
TEA	Transtorno do Espectro Autista
TIC	Tecnologias de Informação para Comunicação
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
WAI-ARIA Applications	Web Accessibility Initiative - Accessible Rich Internet Applications
WCAG	Web Content Accessibility Guidelines
WebAIM	Web Accessibility In Mind

Sumário

1. INTRODUÇÃO	13
1.1 Objetivo geral.....	14
1.2 Objetivos específicos	14
1.3 Organização do texto.....	14
1.4 Método de desenvolvimento	15
2. REVISÃO BIBLIOGRÁFICA	17
2.1 O que é deficiência	17
2.2 Tecnologia assistiva.....	19
2.3 Usabilidade <i>versus</i> acessibilidade	20
2.4 Documentações e literatura base	21
2.4.1 WCAG (<i>Web Content Accessibility Guidelines</i>).....	22
2.4.2 WAI-ARIA (<i>Web Accessibility Initiative – Accessible Rich Internet Applications</i>).....	23
2.4.3 GAIA (<i>Guidelines for Accessible Interfaces for People with Autism</i>)	24
2.4.4 eMAG (Modelo de Acessibilidade em Governo Eletrônico).....	24
2.4.5 MDN Web Docs (<i>Mozilla Developer Network Web Documentations</i>).....	25
2.4.6 Acessibilidade na Web: Boas práticas para construir <i>sítes</i> e aplicações acessíveis	25
2.4.7 Curso <i>Web Accessibility</i> por Google e Udacity	25
2.5 <i>HTML</i> Semântico	26
2.6 Árvore de acessibilidade	27
3. APLICAÇÃO PRÁTICA DOS CONCEITOS	29
3.1 Objeto de estudo e atributos escolhidos	29
3.2 Testes e análise de resultados	30
3.2.1 Análise automatizada	31
3.2.1.1 Markup Validation Service da W3C	31

3.2.1.2 <i>Plugin</i> WAVE e Adobe Color	38
3.2.2 Análise manual.....	50
3.2.2.1 Leitura clara e objetiva.....	51
3.2.2.2 Navegação por teclado.....	54
3.2.2.3 Navegação por leitor de tela (NVDA).....	55
3.2.2.4 Análise dos controles de conteúdo animado	59
3.2.2.5 Utilização de <i>zoom</i> em até 200%	60
3.2.2.6 Análise da informação mostrada em caso de erro.....	61
4. CONCLUSÃO E TRABALHOS FUTUROS	63
5. REFERÊNCIAS BIBLIOGRÁFICAS	65

1. INTRODUÇÃO

No Brasil de hoje, há um grande número de pessoas com deficiência. Segundo a Pesquisa Nacional de Saúde (PNS) realizada pelo Instituto Brasileiro de Geografia e Estatísticas (IBGE, 2019), cerca de 17,3 milhões de pessoas tem algum tipo de deficiência, (levando em conta a população com mais de dois anos e as deficiências consideradas na pesquisa) representando cerca de 8,4% da população. Dentre os idosos esse número também é grande, por volta de 8,5 milhões, ou 24,8% dos idosos. Apesar da relevância do assunto, visto o número de pessoas que possuem necessidades especiais, na prática, pouco tem se feito a favor da acessibilidade na *web*. De acordo com a organização Web Para Todos (2022), em sua análise “Número de *sites* brasileiros aprovados em todos os testes de acessibilidade tem queda em relação ao ano passado e é ainda menor que 1%”, como o título diz, menos de 1% dos *sites* brasileiros cumprem os requisitos para serem aceitos como acessíveis. Dos 21 milhões de *sites* analisados, apenas 0,46% foram considerados acessíveis, o que representou uma diminuição em relação ao ano anterior, quando a porcentagem chegou a 0,89%. A pesquisa ainda aponta que um fator de impacto nos resultados foi a quantidade de novos *sites* criados no período, sendo em sua maioria de empresas pequenas, que não veem a acessibilidade como fator importante na hora de desenvolver uma aplicação.

Outro tópico que pode ser levantado é a questão de o mundo estar cada vez mais conectado. De acordo com a Pesquisa sobre o uso das Tecnologias de Informação e Comunicação nos Domicílios Brasileiros (TIC Domicílios), realizado pelo Centro Regional de Estudos para Desenvolvimento da Sociedade da Informação (2020 apud Agência Brasil, 2021), houve um aumento de 12% na quantidade de domicílios com internet no Brasil, de 2019 para 2020, o que mostra que a procura pela digitalização não veio só por parte das empresas e instituições, mas sim da população em geral. Vale pontuar que esse dado é reforçado pela pandemia de COVID-19 que aconteceu justamente nesse período, trazendo a necessidade de distanciamento social, e, por consequência, fazendo crescer a procura por formas de contato digital.

Um estudo que salienta a ideia de que o mundo pós pandemia de COVID-19 é um mundo mais conectado foi feito pela agência Marco (2022), que questionou 14.200 consumidores sobre seus novos hábitos, pós 2020. Quando se fala sobre o uso da *internet* por brasileiros, é visto que 91% dos que responderam pesquisa alegou

comprar mais online depois da pandemia, liderando o *ranking* quando comparado os números dos outros países pesquisados.

A partir desses dados, pode-se observar uma necessidade de orientação quanto aos conceitos e aplicações da acessibilidade na *web*, já que há um crescente número de empresas e pessoas no universo digital, mas a acessibilidade digital no Brasil não tem avançado.

1.1 Objetivo geral

Tendo em vista a importância da acessibilidade para *websites* e pensando em evidenciar questões de acessibilidade de um caso real, foi escolhido para esse projeto fazer uma análise de um *site web*, evidenciando as funcionalidades que o tornam acessível e apontando sugestões de melhorias que poderiam ser implementadas com o propósito de torná-lo mais acessível para os usuários.

1.2 Objetivos específicos

A partir do objetivo geral, é esperado alcançar objetivos específicos ao projeto, sendo eles:

1. Listar recomendações básicas para uma aplicação *web* acessível
2. Elaborar uma lista de etapas para validação de acessibilidade de um *site na web*
3. Analisar formas de tornar acessível um *website* que já está pronto e disponível *online*

1.3 Organização do texto

O trabalho foi dividido em partes, buscando melhor compreensão do tema. Inicia-se com a introdução, apresentando o trabalho e o assunto a ser tratado. Em seguida, há a revisão bibliográfica, onde as bases do tema foram explicadas: deficiência, tecnologia assistiva, usabilidade, acessibilidade e documentações existentes. Após essa a exposição do tema, foi desenvolvida uma aplicação prática dos tópicos apresentados na forma de reformulação de um *site* escolhido seguindo os critérios descritos em tecnologia e aspectos de implementação. Por último, é tratada a conclusão, onde são descritos os conhecimentos adquiridos e propostas de trabalhos futuros.

1.4 Método de desenvolvimento

Para desenvolvimento do projeto, alguns métodos foram selecionados a fim de atingir os objetivos propostos. O procedimento do projeto se encaixa em “estudo de caso”, tendo o objeto de estudo escolhido a partir das características descritas na seção “3.1 Objeto de estudo e atributos escolhidos”. Para o desenvolvimento desse projeto, foram empregadas as seguintes etapas:

No capítulo “1. Introdução”, há as definições do projeto:

1. Escolha do tema
2. Seleção de objetivos gerais e específicos
3. Definição do desenvolvimento

No capítulo 2 são descritas as diretrizes desse trabalho, através da Revisão Bibliográfica:

4. Definições de conceitos importantes
5. Busca de documentações para referência
6. Seleção de ferramentas de análise

Então foi feita a execução de análises no capítulo 3, sobre Aplicação prática dos conceitos:

7. Seleção do objeto de estudo
8. Elaboração de lista de passo a passo
9. Análise automatizada utilizando as ferramentas escolhidas
10. Formulação de requisitos usando como base as documentações escolhidas
11. Execução da análise manual a partir dos requisitos selecionados

Por fim, é finalizado com o capítulo “4. Conclusão e trabalhos futuros”:

12. Síntese do conhecimento adquirido e formas de complemento do trabalho

2. REVISÃO BIBLIOGRÁFICA













2.1 O que é deficiência

Para iniciar-se o assunto sobre acessibilidade, precisamos definir o conceito de pessoa com deficiência. Para isso, foi considerada a definição do decreto conhecido como “Convenção sobre os direitos das pessoas com deficiência” (Brasil, 2009), que diz o seguinte sobre o assunto:

“Pessoas com deficiência são aquelas que têm impedimentos de longo prazo de natureza física, mental, intelectual ou sensorial, os quais, em interação com diversas barreiras, podem obstruir sua participação plena e efetiva na sociedade em igualdades de condições com as demais pessoas.”

Existe uma gama de características dentro do espectro da deficiência. FERRAZ (2020) lista os seguintes tipos: visual, auditiva, motora e cognitiva ou neurológica, e dentro dessas, ainda há divisões com aspectos únicos. Alguns exemplos de variações dentro dos tipos de deficiência são: cegueira e daltonismo (visual), surdez e baixa audição (auditiva), tetraplegia e amputação (motora) e dislexia e síndrome de Down (cognitiva). Outro aspecto que precisa ser evidenciado sobre o tema deficiência é a permanência. De acordo com o guia *Inclusive 101* da Microsoft (2016), existem 3 tipos de categorias de permanência: permanente, temporária e situacional. A deficiência permanente é o termo mais costumeiro e que foi tratado até agora no projeto: uma restrição a longo prazo. Mas há casos em que a dificuldade é temporária, como um braço quebrado ou infecção auricular, ou seja, em algumas semanas a pessoa não terá mais esse impedimento, mas no momento, ela precisa de auxílio para realizar suas tarefas. E por último, há as necessidades situacionais, como um pai com o filho no colo ou alguém em um lugar com muito barulho. Só por aquele momento a pessoa terá dificuldade em executar algumas funções.

Figura 1 - Relação de impedimento X permanência do guia: *Inclusive 101*

	Permanent	Temporary	Situational
Touch	 One arm	 Arm injury	 New parent
See	 Blind	 Cataract	 Distracted driver
Hear	 Deaf	 Ear infection	 Bartender
Speak	 Non-verbal	 Laryngitis	 Heavy accent

Fonte: Microsoft, 2016

A deficiência exerce ainda impacto na alfabetização, já que influencia sobre o aprendizado. Além de deficiências intelectuais, como a dislexia, que pode prejudicar a alfabetização, há outros tipos de deficiência, como cegueira e surdez, que pedem que o aprendizado ocorra de outras formas. Assim como os cegos possuem o *Braile*, de acordo com FLOR, VANZIN e ULBRICHT (2013), surdos podem aprender tendo como base a própria linguagem oral, ou utilizando línguas específicas como a Língua Brasileira de Sinais (LIBRAS). A forma como essa última é estruturada destoa em parte da língua portuguesa, o que pode influenciar a forma como um surdo lê e escreve. Um exemplo, trazido por SILVA e NOGUEIRA (2014), pode ser visto nas frases abaixo. Apesar de trazerem a mesma informação, pode-se ver que não há

conjugação verbal ou artigos e a intensidade é reforçada repetindo-se o sinal várias vezes.

- Frase em português: A mãe ama muito o filho
- Frase em LIBRAS: MÃE-AMAR-MUITO-FILHO (MUITO-MUITO-MUITO)

2.2 Tecnologia assistiva

Quando o assunto é o contexto de pessoas com deficiência dentro da *web*, é preciso falar sobre as formas de acesso dessas pessoas a esse tipo de conteúdo. Para definir o que seriam as tecnologias assistivas, pode-se consultar o Comitê de Ajudas Técnicas (2007, p. 3), que utiliza o seguinte conceito:

“[...]engloba produtos, recursos, metodologias, estratégias, práticas e serviços que objetivam promover a funcionalidade, relacionada à atividade e participação, de pessoas com deficiência, incapacidades ou mobilidade reduzida, visando sua autonomia, independência, qualidade de vida e inclusão social.”

Dois exemplos para esse tipo de tecnologia são leitores de tela, que fazem o papel de ler a tela do usuário e traduzir para a linguagem oral aquilo que está disposto em forma de texto, e a navegação por teclado, que pode ser feita pelo teclado de computador tradicional, ou por algum teclado adaptado para as necessidades do usuário, mas que consiste em transitar pelos elementos da página *web* e interagir com eles sem o uso de *mouse*. Existe ainda uma grande infinidade de outras ferramentas, como tradutores de LIBRAS, *mouses* adaptados e leitores de movimento ocular.

Figura 2 - Imagem de teclado para uso de braille



Fonte: Tecnovisão, 2024

A tecnologia assistiva pode estar presente tanto no computador do usuário, como em *softwares* leitores de tela, quanto dentro do próprio *site*, em forma de tradutores de LIBRAS, por exemplo. De acordo com o FLOR, VANZIN e ULBRICHT (2013), é importante pensar essas tecnologias dentro do próprio *site*, apresentando para o usuário diversas formas de utilização. Eles citam que não há apenas uma linguagem difundida na comunidade surda, sendo preciso pensar o surdo como um indivíduo de características únicas. Assim, somente dispor conteúdo textual e áudio legendado não supre a demanda surda, sendo preciso avaliar as linguagens que existem na comunidade surda, como visual e LIBRAS.

A documentação GAIA [s.d] (Guia de Acessibilidade de Interfaces *Web* com foco em aspectos do Autismo), reforça a ideia quando diz que tecnologias que fazem o *site* mais dinâmico, como ferramentas para aumentar a fonte do *site* ou adicionar alto contraste entre texto e fundo, são importantes para que o usuário consiga adaptar o conteúdo às suas necessidades.

2.3 Usabilidade *versus* acessibilidade

Existem vários requisitos que podem ser usados para avaliar a qualidade de um *software*, segundo a ISO/IEC 25010. Para essa monografia, que trata sobre

usabilidade, foi falado sobre a “Capacidade de Interação”, que é definida como: “Grau em que um produto ou sistema pode ser interagido por usuários específicos para trocar informações na interface do usuário, a fim de completar tarefas específicas em uma variedade de contextos de uso.” (ISO/IEC 25010, 2024, tradução nossa). Isso significa que, não se pode falar sobre a usabilidade de um *software* sem pensar em como os diferentes grupos de usuários irão interagir com a aplicação.

A usabilidade, quando tem seu foco voltado para pessoas com deficiência, chama-se “acessibilidade”, e para ter uma definição clara sobre esse tema, pode-se consultar a WCAG (*Web Content Accessibility Guidelines*) (vide tópico 2.4.1), que é o conjunto de diretrizes de acessibilidade para conteúdo *web* criadas pela W3C (*World Wide Web Consortium*). A documentação traz a seguinte definição:

“Acessibilidade na *web* significa que *websites*, ferramentas e tecnologias são desenhados e desenvolvidos para que as pessoas com deficiência possam usar. Mais especificamente, as pessoas podem: perceber, entender, navegar e interagir com a *web*; contribuir com a *web*”. (W3C, 2022, tradução nossa).

Mas é importante salientar que mais para frente a documentação diz:

“Acessibilidade na *web* também pode beneficiar pessoas sem deficiência, por exemplo: pessoas usando telefones celulares, relógios inteligentes, TVs inteligentes e outros dispositivos com telas pequenas, diferentes modos de entrada, etc; pessoas idosas com habilidades alteradas devido ao envelhecimento; pessoas com ‘deficiências temporárias’, como um braço quebrado ou que perderam os óculos; pessoas com ‘limitações situacionais’ como luz do sol intensa ou em um ambiente onde não podem ouvir áudio; pessoas com conexão de *internet* lenta, ou que possuem banda larga limitada ou cara”. (W3C, 2022, tradução nossa)

Isto é, pode-se dizer que a acessibilidade influencia diretamente como diversos grupos de pessoas utilizam as aplicações *web*, já que páginas acessíveis são mais fáceis de navegar e mais rápidas de carregar, ajudando assim, não só usuários com deficiência, mas pessoas com dificuldades temporárias, pessoas com *internet* instável, idosos com dificuldade de utilizar tecnologias e crianças em fase de aprendizado.

2.4 Documentações e literatura base

Para analisar o objeto de análise, foram selecionadas algumas documentações, são elas: WCAG, WAI-ARIA (*Web Accessibility Initiative – Accessible Rich Internet Applications*), GAIA e eMAG (Modelo de Acessibilidade em Governo Eletrônico).

Como ferramentas de suporte, também foi usado o livro “Acessibilidade na *web* – Boas práticas para construir *sites* e aplicações acessíveis” de Reinaldo Ferraz, o curso

ministrado pelo Google e Udacity, *Web Accessibility* e a documentação MDN Web Docs (*Mozilla Developer Network Web Documentations*).

2.4.1 WCAG (*Web Content Accessibility Guidelines*)

O *World Wide Web Consortium* (W3C), é um consórcio internacional responsável pela padronização da *World Wide Web*. Com membros no mundo todo, eles são responsáveis por formular as normas de segurança, privacidade, internacionalização e acessibilidade na *web*.

O conjunto de diretrizes de acessibilidade que são organizadas pelo grupo é chamado de WCAG e está atualmente na versão 2.1, lançada em setembro de 2023, mas recebe atualizações constantes de seus organizadores.

Essa documentação traz 4 princípios básicos que são as bases para a acessibilidade na *web*. São eles: perceptível, operável, compreensível e robusto.

O princípio “perceptível” diz respeito aos elementos estarem visíveis de alguma forma para os usuários, tendo em mente que ele deve estar disponível para todos os usuários, ou seja, caso seja inserido uma imagem relevante para o conteúdo, ela deve ter sua representação textual, fazendo com que pessoas com deficiência visual possam ter acesso a esse conteúdo.

Sobre a aplicação ser “operável”, é dito que o usuário deve conseguir operar a interface sem impedimentos. Um *website* deve, por exemplo, ser acessível utilizando-se qualquer tipo de navegação, como *mouse*, teclado, tela *touch*, entre outros.

O princípio da “compreensibilidade” diz respeito ao usuário compreender as informações e navegação do *website*. Deve-se ter em mente que diferentes usuários podem visualizar o *site* de formas diferentes, precisando de ajustes na *interface*, como aumentar o contraste entre o fundo e o conteúdo, ajustar o tamanho do conteúdo, entre outras formas de acesso.

Por último, quando se fala de uma aplicação robusta, tem-se que ela deve se comportar da mesma forma para uma gama grande de dispositivos, navegadores e tecnologias assistivas, como diferentes modelos de celular, tamanhos de tela de computadores e leitores de tela.

Ainda levando em conta as recomendações da WCAG, elas se dividem em 3 categorias: A, AA e AAA, que se organizam pela profundidade com relação a

atendimento de necessidades especiais, ou melhor, quanto maior a quantidade de “A” de um critério, mais específico ele é. A recomendação da WCAG é atender a maior quantidade possível dos critérios A e AA, mas deixar os critérios AAA para componentes e situações específicas.

2.4.2 WAI-ARIA (*Web Accessibility Initiative – Accessible Rich Internet Applications*)

O WAI-ARIA é uma documentação com séries de recomendações para tornar as aplicações *web* mais acessíveis, principalmente aquelas que são lidas com leitores de tela ou utilizadas por pessoas que não conseguem utilizar o *mouse*. Através do *framework* WAI-ARIA é possível identificar, relacionar e informar os estados dos elementos para o usuário da aplicação.

Os elementos principais do WAI-ARIA são os *roles*, que assimilam um elemento a uma função ou tipo; os *states*, que adicionam um estado ao atributo do elemento; e as *properties*, que enriquecem os elementos através da adição ou modificação de atributos. As mudanças dos estados e propriedades de um elemento podem ser notificadas ao usuário, dando retornos sobre as interações do usuário com a página, por exemplo. Tanto as propriedades padrão, quanto às alterações, podem ser lidas pelas tecnologias assistivas e notificadas ao usuário sobre as funcionalidades e informações contidas na página.

Figura 3 - Captura de tela de trecho de código que utiliza atributos do tipo ARIA

```
<form action="https://armazemdocampo.shop/search" method="get" class="site-
header__search small--hide" role="search">
  <input type="search" name="pesquisar" id="SiteNavSearch" placeholder="Pes
quisar" aria-label="Pesquisar" class="site-header__search-input">
  <button type="submit" class="text-link site-header__link site-header__sea
rch-submit">
    <svg aria-hidden="true" focusable="false" role="presentation" class="ic
on icon-search" viewBox="0 0 32 32">...</svg>
    <span class="icon__fallback-text">Procurar</span>
  </button>
</form>
```

Fonte: Armazém do Campo, 2024

Ainda sobre WAI-ARIA, é importante salientar que apesar dessa ferramenta poder transformar *tags HTML* simples em *tags* semânticas, assim como aconselha FERRAZ (2020), é preferível utilizar as *tags* próprias do *HTML* semântico (definido no tópico

2.5). No caso de um código legado¹ e que não haja possibilidade de reescrevê-lo no momento, uma *tag* `` pode receber o poder de ser lida como botão ao receber “*role=’button’*” e assim suprimir algumas funcionalidades do elemento que ele não tinha ao utilizar *tags* genéricas, como o reconhecimento pelas ferramentas de leitura de tela. Porém, FERRAZ aponta que é preciso prever muitos ajustes no código para que a *tag* `` realmente tenha comportamento de uma *tag* `<button>`, como adicionar funções que lidam com o comportamento do elemento para que as interações com o botão estejam corretas (como “*onfocus*” e “*onmouseover*”), ademais de inserir no CSS os atributos necessários para exibir a área de foco.

2.4.3 GAIA (*Guidelines for Accessible Interfaces for People with Autism*)

O Guia de Acessibilidade de *Interfaces Web* com foco em aspectos do Autismo, em português, é uma documentação específica que visa auxiliar desenvolvedores e educadores a pensarem aplicações *web* para crianças autistas, respeitando as singularidades que o TEA (Transtorno do Espectro Autista) possui.

São trazidas recomendações simples, tornando possível pensar formas de apresentar as informações levando em conta a usabilidade dos usuários e possíveis cenários que podem impactar de forma negativa a usabilidade do *site*.

2.4.4 eMAG (Modelo de Acessibilidade em Governo Eletrônico)

O governo brasileiro tem grande interesse no assunto de acessibilidade, sendo visto nos manuais de acessibilidade criados, e no suporte e manutenção da ferramenta de tradução de LIBRAS VLibras [s.d.], que é disponibilizada gratuitamente para todos que quiserem adicionar em suas aplicações *web*.

De acordo com a documentação do Modelo de Acessibilidade em Governo Eletrônico, o grande interesse por essa área é uma maneira de fomentar a inclusão social e diminuir a desigualdade, através do acesso à comunicação e informação via *internet*. A partir dessa ideia, foi criada uma série de regras que padronizasse o conteúdo dos *sites* do governo brasileiro, de forma a respeitar as normas internacionais, como a

¹ Código legado por ser definido como um código criado há algum tempo e até mesmo por pessoas diferentes da equipe atual.

WCAG, mas criando uma versão específica para o conteúdo nacional. E assim nasceu a eMAG, que hoje está na sua versão 3.1, publicada em abril de 2014.

2.4.5 MDN Web Docs (*Mozilla Developer Network Web Documentations*)

A documentação criada pela empresa de tecnologia Mozilla, chamada de MDN Web Docs, é uma rede de referências e cursos sobre os mais variados conceitos da programação, descrevendo desde conceitos independentes de linguagem, como *design patterns*, até conceitos mais específicos, como as descrições das *tags HTML* existentes.

Essa plataforma, apesar de ser mantida por uma empresa privada, tem o gerenciamento de conteúdo compartilhado, sendo possível que desenvolvedores independentes tragam sugestões e ajustes de conteúdo para a página, além de receber material de parceiros como as empresas Microsoft, Google e W3C.

2.4.6 Acessibilidade na Web: Boas práticas para construir *sites* e aplicações acessíveis

Trabalhando na área de desenvolvimento *web* desde 1998, Reinaldo Ferraz atua com acessibilidade na *web* desde 2004, contando com vários projetos sobre o tema no currículo, como traduções oficiais das diretrizes da W3C e compartilhamento de conteúdo gratuito sobre acessibilidade na *web*.

O livro usado nessa monografia é o terceiro escrito pelo autor. Ele passa sobre os principais tópicos necessários para validação da acessibilidade de uma aplicação *web*, trazendo sobre os conceitos bases, documentações fundamentais, ferramentas disponíveis no mercado e etapas de verificação, entre outros temas que cercam a acessibilidade na *web*.

2.4.7 Curso *Web Accessibility* por Google e Udacity

O curso disponível na plataforma *online* Udacity [s.d.] é ministrado por desenvolvedores da empresa de tecnologia Google e traz conceitos teóricos e práticos sobre o que é acessibilidade e como aplicá-la na prática, através dos módulos: foco, semântica, navegação, *aria* e estilo.

2.5 HTML Semântico

Apesar de hoje haver uma grande quantidade de dispositivos e tecnologias que melhoram a usabilidade e acessibilidade dos *sites*, sem um código que implemente as melhores práticas, as barreiras de comunicação continuam existindo.

O *HTML* é a linguagem de marcação de hipertexto (*HyperText Markup Language*) utilizada pelos navegadores para dispor os elementos na tela de *interface* do usuário, tendo uma versão “semântica” que conta com um conjunto de normas que dão significado ao código escrito, e que traz inúmeros benefícios à aplicação, já que permite o uso das *tags HTML* com propósito e funcionalidades nativos.

Entre a lista de benefícios, estão: a boa leitura por parte de mecanismos de busca e leitores de tela, além da melhor manutenção do projeto. Um exemplo que podemos citar é a hierarquização dos títulos e conteúdo do *website*. Caso isso não seja feito, mesmo com um leitor de tela, uma pessoa com deficiência visual encontrará grande dificuldade em localizar o conteúdo que procura, já que não conseguirá navegar pela estrutura do *site* de forma apropriada.

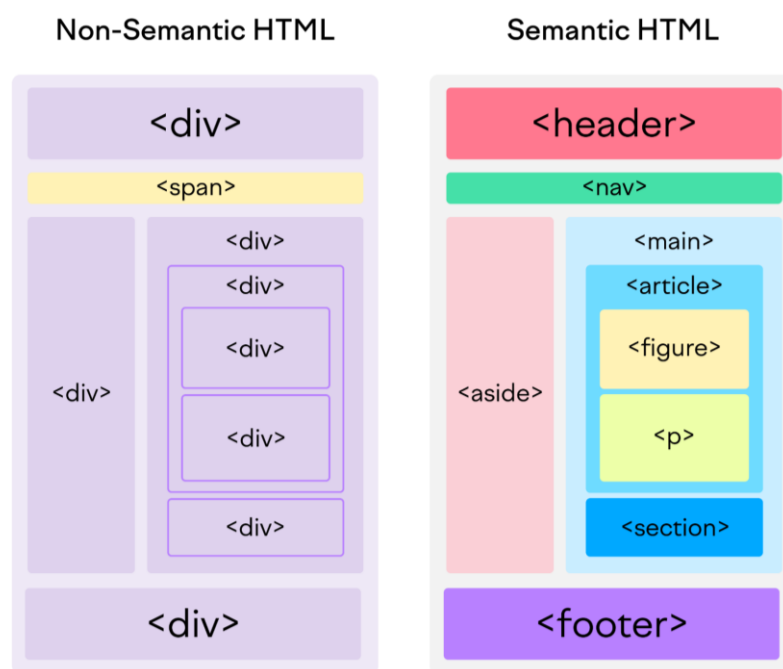
Existem vários assuntos que envolvem a semântica do *HTML*, como usar corretamente os *roles* (papéis) das *tags* (explicado no tópico 2.4.2 WAI-ARIA) e adicionar atributos para a identificação dos elementos. Apesar de haver várias *tags HTMLs*, cada uma tendo seu papel principal e facilitando a navegação dos usuários, nem sempre elas são utilizadas de forma correta, e não é incomum desenvolvedores ainda usarem a *tag* `<div>` para criar a maioria dos seus elementos, o que dificulta para as tecnologias assistivas identificarem os tipos dos conteúdos da página e suas funções de acordo com FERRAZ (2020).

Pode-se dizer, por fim, que o *HTML* semântico é a base para a acessibilidade na *web*, já que ele é responsável por informar ao navegador as informações fundamentais do conteúdo disponível, como ordem de leitura, tipos e estados dos elementos, formas de interação com a página, entre outras funções.

Na figura 4, temos uma imagem comparativa que mostra a diferença do *HTML* ao utilizarmos *tags* genéricas e *tags HTML* semânticas.

Figura 4 - Comparativo de *HTML* não semântico e *HTML* semântico

What Is Semantic HTML?



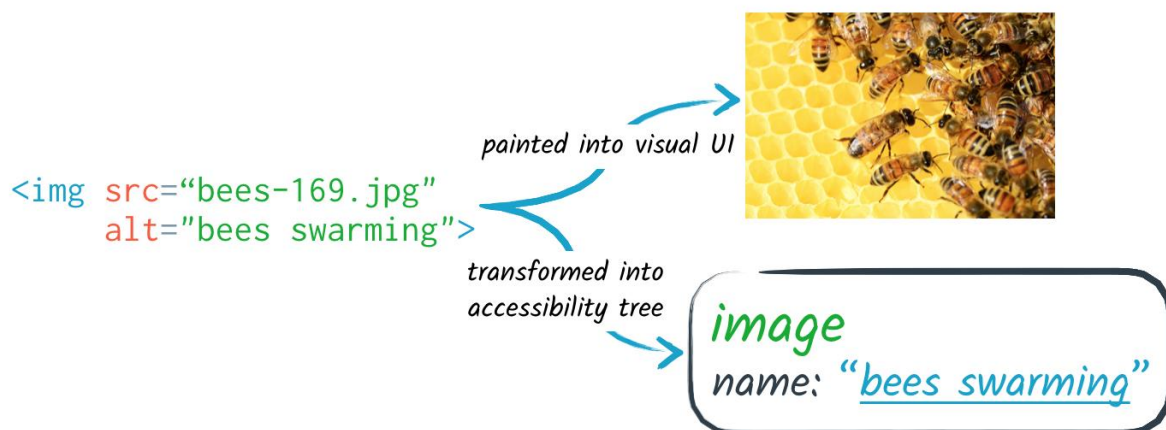
Fonte: Semrush, 2022

2.6 Árvore de acessibilidade

Quando é criada uma página *web*, utiliza-se recursos e linguagens que serão compilados e transformados em um arquivo *DOM* (*Document Object Model* ou Modelo de Objeto de Documentos) para que o navegador possa ler e reproduzir o conteúdo ao usuário. O *DOM*, como o nome sugere, é um modelo de documento constituído por uma árvore de nós, mostrando de forma organizada e hierárquica os elementos da página.

A partir desse modelo, o navegador pode criar uma árvore de acessibilidade, chamada de *AOM*, ou *Accessibility Object Model* (Modelo de Objeto de Acessibilidade, em tradução livre), como visto da figura 2. Esse modelo é uma versão dos dados específica para o uso de tecnologia assistiva, tendo quatro propriedades: “*role*”, “*state*” (que já foram explicados no tópico 2.4.2 WAI-ARIA), “*name*” (como o objeto é chamado) e “*description*” (uma explicação complementar do objeto, descrevendo as informações contidas nele, por exemplo).

Figura 5 - Comparativo entre código *HTML*, conteúdo *HTML* renderizado na *DOM* e conteúdo *HTML* renderizado na *AOM*



Fonte: Web Incubator CG, 2023

3. APLICAÇÃO PRÁTICA DOS CONCEITOS

3.1 Objeto de estudo e atributos escolhidos

A pandemia causada pela disseminação da *COVID-19* forçou as empresas a repensarem seus negócios, visto nos dados trazidos pelo jornal G1 (2020), que fala sobre um aumento de 40% na quantidade de *e-commerces* brasileiros. Além disso, é dito que desses *e-commerces*, quase metade possuem o faturamento de até R\$250.000, sendo assim empresas de pequeno porte. Por último, tem-se que 91,27% desses *sites* tem uma média mensal de visitas de até 500.000 por mês.

Para a seleção do objeto de estudo, foi procurado um *website* que se enquadrasse nos requisitos selecionados, ou seja, pertencente a uma empresa de pequeno porte com *e-commerce* criado durante a pandemia e que o *site* possui até 500.000 visitas por mês, sendo considerado de pequeno a intermediário.

Esses critérios foram escolhidos como uma forma de representar o tipo de *site* que mais cresceu durante a pandemia, e que, de acordo com o *site* Web para Todos, por serem *sites* pequenos em relação à quantidade de acessos, não têm tanta atenção com os critérios de acessibilidade.

A partir desses dados, houve uma primeira seleção com *sites* de empresas conhecidas da autora. Após análise de cada uma delas no *site* Semrush, para separar aquelas que tinham a quantidade de visitas necessárias, foi validado no *site* Who.is se a data que o *website* foi criado estava dentro do período desejado, e, por último, foi verificado no site da JUCESP (Junta Comercial do Estado de São Paulo) a data de inauguração da empresa, para ter certeza de que foi antes de 2019.

A empresa escolhida que passou por todos os critérios foi o Armazém do Campo, uma loja de produtos alimentícios orgânicos e agroecológicos de marca própria e de empresas terceiras, que também comercializa produtos de vestuário. Ela foi inaugurada em 2016 e está enquadrada como EPP (empresa de pequeno porte), de acordo com o registro na JUCESP (2024). O *e-commerce* da empresa foi criado em 31/03/2021 de acordo com o *site* Who.is [s.d.], ou seja, após a pandemia de *COVID-19* e possui até 500.000 visitas por mês, sendo considerado de pequeno a intermediário. Sobre esse último ponto, o número médio de visitas é de 19.000 por dia, totalizando a média 570.000 por mês, de acordo com o *site* Semrush (2024), mas por se tratar de uma média de acessos de apenas alguns dias, a margem de aproximação foi considerada aceitável.

Um ponto importante a levantar é que o *site* da empresa foi feito dentro da plataforma Shopify, que provém uma ferramenta para criar *sites* apenas arrastando e soltando componentes pré-prontos, que também podem ser customizados. Desse modo, apesar do *site* ser de uma empresa pequena, supõe-se que há por trás componentes desenvolvidos por especialistas na área de programação.

3.2 Testes e análise de resultados

No capítulo final de seu livro, Verificação de acessibilidade, FERRAZ (2020), nos traz uma descrição de processos para validação de uma aplicação. O primeiro ponto trazido é a análise automatizada, ou seja, utilizando algumas ferramentas externas ao *site* que levam em conta as diretrizes de acessibilidade já reconhecidas pelo mercado. Essa primeira etapa será complementada com a análise manual, avaliando questões que não conseguem ser validadas pelas ferramentas.

Enfim, chegamos a uma lista de etapas a serem seguidas e é esperado que teremos uma aplicação acessível ao final de todas essas análises, assim cumprindo parte dos objetivos do projeto:

1. Análise automatizada
 - a. Análise do uso das *tags* com o validador da W3C
 - b. Validação da interface no *plugin* WAVE
 - c. Verificação de contrastes de cores no *site* Adobe Color
2. Análise manual
 - a. Leitura clara e objetiva
 - b. Navegação por teclado
 - c. Navegação usando o leitor de tela NVDA
 - d. Exame dos controles de conteúdo animado
 - e. Validação do uso do zoom até 200%
 - f. Avaliação da informação mostrada em caso de erro
 - g. Análise de existência de representação textual para o conteúdo audiovisual (não foi validado, por não conter esse tipo de material no *site* analisado)

O *site* a ser analisado, conforme previsto nos objetivos do projeto (itens 1.1 e 1.2), foi baixado e salvo em um repositório do GitHub (que é uma plataforma para versionamento de código, assim como armazenagem de arquivos) e está disponível

no seguinte *link*: <https://github.com/thamiaviciente/monografia-armazem-do-campo-a11y>. Dessa forma tem-se certeza de que a versão usada durante todo o trabalho é a mesma, visto que não sofrerá possíveis mudanças dos desenvolvedores do *site* original. Ainda, o *site* será atualizado conforme os resultados das análises, para que a correção seja validada e a barreira de usabilidade seja removida.

Figura 6 - Captura de tela do projeto clonado



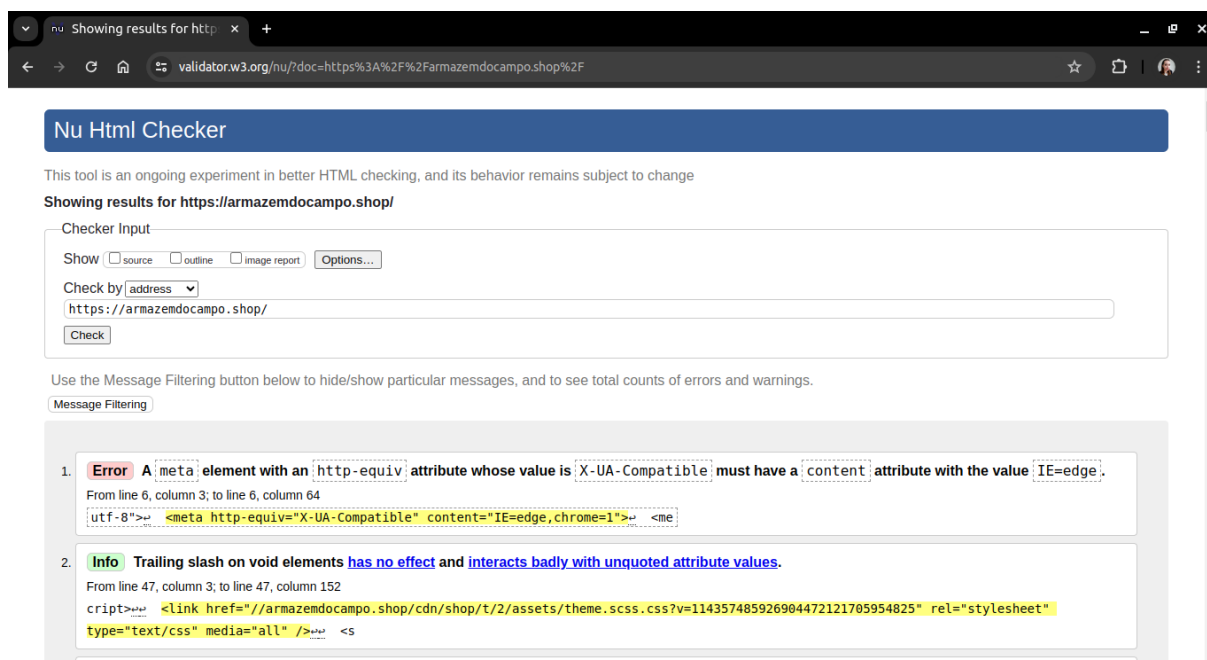
Fonte: Armazém do Campo, 2024

3.2.1 Análise automatizada

A análise foi iniciada com as ferramentas automatizadas, que fazem a varredura do *site* escolhido em busca de barreiras de acessibilidade, sinalizando-as e justificando o motivo de atrapalharem a navegação dos usuários, seguindo as recomendações da WCAG. Além disso, elas selecionam o código a ser alterado e sugerem ajustes.

3.2.1.1 Markup Validation Service da W3C

Figura 7 - Captura de tela do validador Markup Validation Service antes de começar as análises e alterações no código



Fonte: Validator W3, 2024

A primeira ferramenta a ser usada, segundo FERRAZ (2020), é o validador de *tags* da W3C, para verificar o uso dos elementos e atributos *HTML* na página. O passo a passo é simples: apenas inserir a *URL* da página que está sendo validada e utilizar a documentação de *tags* da W3C e outras fontes auxiliares para entender e corrigir os pontos levantados. Essa validação deve ser feita página a página para que o *site* inteiro seja analisado, mas para tornar esse trabalho mais enxuto, apenas a página inicial foi analisada.

Error: A meta element with an http-equiv attribute whose value is X-UA-Compatible must have a content attribute with the value IE=edge.

A primeira *tag* que o validador apontou erro foi a *tag meta* que define as especificações de compatibilidade do *site*. A explicação pede que se use o conteúdo “*IE=edge*” nos atributos da *tag*, sendo que o conteúdo atual é “*IE=edge, chrome=1*”. Isso se deve ao fato de o *Google Frame* não ter mais suporte por parte da Google, então essa especificação entrou em desuso. Removendo-se a parte do Chrome, o validador não acusa mais erro sobre isso.

Info: Trailing slash on void elements [has no effect](#) and [interacts badly with unquoted attribute values](#).

Os próximos 5 apontamentos são com relação a fechar as *tags void* (que não tem conteúdo dentro delas) com “/>” ao invés de “>”. Isso é apontado como informação, e não um erro, já que pode ter sido causado por algum *framework* que obriga esse uso e só causaria um erro se os atributos declarassem seus valores sem aspas. Sendo assim, isso pode ser removido, mas dependendo de como o *site* original foi feito, não pode ser modificado.

Error: Element [style](#) not allowed as child of element [div](#) in this context.
(Suppressing further errors from this subtree.)

O passo seguinte fala sobre utilização da *tag style* dentro de um elemento do tipo *div*. Isso é apontado como um erro e, portanto, o conteúdo dentro dessa *tag* foi migrado para o respectivo arquivo de CSS da página.

Warning: The [banner](#) role is unnecessary for element [header](#).

Warning: The [navigation](#) role is unnecessary for element [nav](#).

Warning: The [main](#) role is unnecessary for element [main](#).

A seguir temos 3 avisos com relação aos *roles* das *tags*. Em ambos os casos, foi adicionado um *role* desnecessário, já que a *tag* já tem por padrão o papel indicado. De acordo com a documentação MDN Web Docs (2024h), a *tag header* tem por padrão o *role* “*banner*” apenas em caso de estar fora das *tags aside*, *article*, *main*, *nav*, e *section*, já que, dentro delas, ele indica o cabeçalho do componente, e como é o caso, todos os *roles* desnecessários foram removidos.

Error: Element [img](#) is missing required attribute [src](#).

O validador apontou 17 *tags img* sem o atributo “src” que é obrigatório, utilizando o atributo “data-src” no lugar. Quando o atributo “data-*” é utilizado, o navegador entende que isso não deve ser mostrado em tela, mas sim como uma informação adicional ao conteúdo, dessa forma, ele pode ser usado como identificador de conteúdo pelo *Javascript* para alterar o próprio *HTML*, o que parece ser o caso, uma vez que ao analisar o *site* após sua compilação, vemos que onde havia o atributo “data-src” foi-se alterado para “src”.

Antes da compilação:

```

```

Após a compilação:

```
<img class="lazyautosizes lazyloaded" data-widths="[100, 140, 180, 250, 305, 440, 610, 720, 930, 1080]" data-aspectratio="0.7493893502686859" data-sizes="auto" data-parent-fit="contain" data-image="" alt="ARROZ CATETO INTEGRAL TERRA LIVRE AGROECOLÓGICA 1 KG" data-srcset="//armazendocampo.shop/cdn/shop/products/IMG_20210212_153639048_100x.jpg?v=1613155515 100w,
//armazendocampo.shop/cdn/shop/products/IMG_20210212_153639048_140x.jpg?v=1613155515 140w,
//armazendocampo.shop/cdn/shop/products/IMG_20210212_153639048_180x.jpg?v=1613155515 180w,
//armazendocampo.shop/cdn/shop/products/IMG_20210212_153639048_250x.jpg?v=1613155515 250w,
//armazendocampo.shop/cdn/shop/products/IMG_20210212_153639048_305x.jpg?v=1613155515 305w,
//armazendocampo.shop/cdn/shop/products/IMG_20210212_153639048_440x.jpg?v=1613155515 440w,
//armazendocampo.shop/cdn/shop/products/IMG_20210212_153639048_610x.jpg?v=1613155515 610w,
//armazendocampo.shop/cdn/shop/products/IMG_20210212_153639048_720x.jpg?v=1613155515 720w,
//armazendocampo.shop/cdn/shop/products/IMG_20210212_153639048_930x.jpg?v=1613155515 930w,
//armazendocampo.shop/cdn/shop/products/IMG_20210212_153639048_1080x.jpg?v=1613155515 1080w"/>

```

```
?v=1613155515 440w,
//armazendocampo.shop/cdn/shop/products/IMG_20210212_153639048_610x.jpg
?v=1613155515 610w,
//armazendocampo.shop/cdn/shop/products/IMG_20210212_153639048_720x.jpg
?v=1613155515 720w,
//armazendocampo.shop/cdn/shop/products/IMG_20210212_153639048_930x.jpg
?v=1613155515 930w,
//armazendocampo.shop/cdn/shop/products/IMG_20210212_153639048_1080x.jp
g?v=1613155515 1080w" sizes="94px"
srcset="//armazendocampo.shop/cdn/shop/products/IMG_20210212_153639048_
100x.jpg?v=1613155515 100w,
//armazendocampo.shop/cdn/shop/products/IMG_20210212_153639048_140x.jpg
?v=1613155515 140w,
//armazendocampo.shop/cdn/shop/products/IMG_20210212_153639048_180x.jpg
?v=1613155515 180w,
//armazendocampo.shop/cdn/shop/products/IMG_20210212_153639048_250x.jpg
?v=1613155515 250w,
//armazendocampo.shop/cdn/shop/products/IMG_20210212_153639048_305x.jpg
?v=1613155515 305w,
//armazendocampo.shop/cdn/shop/products/IMG_20210212_153639048_440x.jpg
?v=1613155515 440w,
//armazendocampo.shop/cdn/shop/products/IMG_20210212_153639048_610x.jpg
?v=1613155515 610w,
//armazendocampo.shop/cdn/shop/products/IMG_20210212_153639048_720x.jpg
?v=1613155515 720w,
//armazendocampo.shop/cdn/shop/products/IMG_20210212_153639048_930x.jpg
?v=1613155515 930w,
//armazendocampo.shop/cdn/shop/products/IMG_20210212_153639048_1080x.jp
g?v=1613155515 1080w">
```

Error: Bad value for attribute `srcset` on element `img`: Ends with empty image-candidate string.

Os próximos dois erros têm relação com os anteriores, porque sinalizam que os atributos *srcset* estão vazios nas *tags img*. Ao observar o *HTML* após a compilação, podemos ver que o atributo *srcset* foi preenchido, indicando que houve uma modificação desse elemento após o carregamento do *Javascript*.

Antes do carregamento do *Javascript*:

```

```

Após o carregamento do *Javascript*:

```
<img class="lazyautosizes lazyloaded" data-widths="[100, 140, 180, 250, 305, 440,
610, 720, 930, 1080]" data-aspectratio="0.7208588957055214" data-sizes="auto"
data-parent-fit="contain" data-image="" alt="SUCO DE UVA INTEGRAL MONTE
VENETO 1,5L" data-
srcset="//armazemdocampo.shop/cdn/shop/products/sucodeuvaint1_5_100x.jpg?v=
1584668595 100w,
//armazemdocampo.shop/cdn/shop/products/sucodeuvaint1_5_140x.jpg?v=158466
8595 140w,
//armazemdocampo.shop/cdn/shop/products/sucodeuvaint1_5_180x.jpg?v=158466
8595 180w,
//armazemdocampo.shop/cdn/shop/products/sucodeuvaint1_5_250x.jpg?v=158466
8595 250w,
//armazemdocampo.shop/cdn/shop/products/sucodeuvaint1_5_305x.jpg?v=158466
8595 305w,
//armazemdocampo.shop/cdn/shop/products/sucodeuvaint1_5_440x.jpg?v=158466
8595 440w,
//armazemdocampo.shop/cdn/shop/products/sucodeuvaint1_5_610x.jpg?v=158466
8595 610w,
```

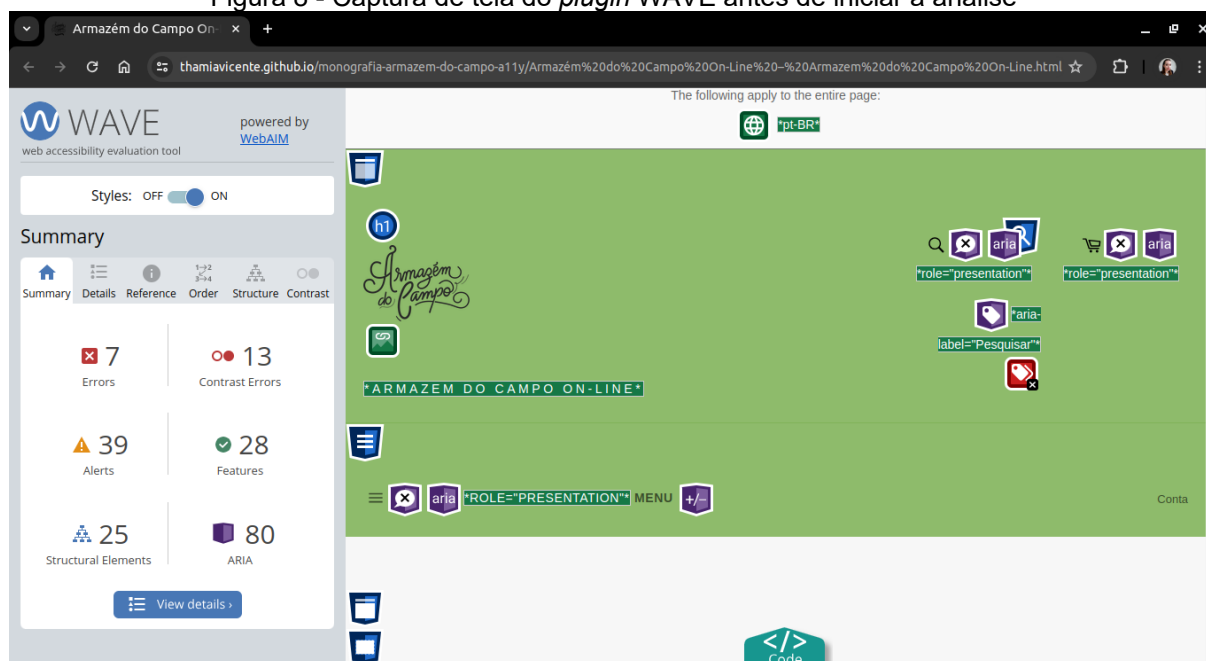
```
//armazendocampo.shop/cdn/shop/products/sucodeuvaint1_5_720x.jpg?v=158466
8595 720w,
//armazendocampo.shop/cdn/shop/products/sucodeuvaint1_5_930x.jpg?v=158466
8595 930w,
//armazendocampo.shop/cdn/shop/products/sucodeuvaint1_5_1080x.jpg?v=15846
68595 1080w" sizes="169px"
srcset="//armazendocampo.shop/cdn/shop/products/sucodeuvaint1_5_100x.jpg?v
=1584668595 100w,
//armazendocampo.shop/cdn/shop/products/sucodeuvaint1_5_140x.jpg?v=158466
8595 140w,
//armazendocampo.shop/cdn/shop/products/sucodeuvaint1_5_180x.jpg?v=158466
8595 180w,
//armazendocampo.shop/cdn/shop/products/sucodeuvaint1_5_250x.jpg?v=158466
8595 250w,
//armazendocampo.shop/cdn/shop/products/sucodeuvaint1_5_305x.jpg?v=158466
8595 305w,
//armazendocampo.shop/cdn/shop/products/sucodeuvaint1_5_440x.jpg?v=158466
8595 440w,
//armazendocampo.shop/cdn/shop/products/sucodeuvaint1_5_610x.jpg?v=158466
8595 610w,
//armazendocampo.shop/cdn/shop/products/sucodeuvaint1_5_720x.jpg?v=158466
8595 720w,
//armazendocampo.shop/cdn/shop/products/sucodeuvaint1_5_930x.jpg?v=158466
8595 930w,
//armazendocampo.shop/cdn/shop/products/sucodeuvaint1_5_1080x.jpg?v=15846
68595 1080w">
```

Warning: The `contentinfo` role is unnecessary for element `footer`.

O último aspecto apontado é o *role contentinfo* presente na *tag footer*. Esse comportamento já é padrão do elemento *footer*, portanto, pode ser removido.

3.2.1.2 Plugin WAVE e Adobe Color

Figura 8 - Captura de tela do *plugin* WAVE antes de iniciar a análise



Fonte: Armazém do Campo e WAVE, 2024

O *plugin* WAVE foi desenvolvido pela WebAIM, uma instituição sem fins lucrativos, que fomenta pesquisas, políticas e práticas sobre deficiência desde 1999. A ferramenta avalia o *HTML* do *website* a partir das diretrizes da WCAG. Seu uso consiste em instalar o *plugin* no navegador e ao iniciá-lo na página a ser analisada, é aberta uma aba com avisos e erros a serem corrigidos, todos com explicações e *links* para as diretrizes em que foram baseadas.

Ao iniciar o *plugin* na página *home* do *site*, foram apontados 6 erros, que serão descritos a seguir.

6 Errors

2 X Multiple form labels

4 X Broken ARIA reference

Existem algumas formas de rotular um elemento no *HTML*, como utilizando o atributo *aria-label* e a *tag label*. O primeiro passa um texto descritivo ao atributo do elemento

a ser descrito. Já no segundo caso, é utilizada a *tag label* que pode envolver o elemento, ou receber o identificador único do elemento a ser vinculado no atributo *for*. Quando olhamos para o campo de busca, temos que foram utilizadas essas duas formas de adicionar um rótulo ao campo de texto de pesquisa, sendo cada rótulo contado como um erro.

Como o ideal é que apenas uma dessas formas seja usada, em razão de a tecnologia assistiva poder não ler corretamente o rótulo, a sugestão é refatorar removendo a *tag label*, já que há um atributo do tipo “*role*” no campo que informa sua função e a *tag label* não está sendo usada para outro fim, estando escondida visualmente dos usuários através de atributos CSS.

O segundo erro apontado pelo *plugin* foi também sobre rótulos usando *aria*. Os *slides* que compõem o carrossel do *site* utilizam o *aria-describedby*, mas os elementos chamados no atributo não existem. Ou melhor, nenhum elemento estava descrevendo os *slides*. Para resolver, bastou adicionar identificadores nos títulos de cada *slide*, utilizando a mesma nomenclatura utilizada no *aria-describedby* de cada *slide*.

Como esse último erro acontecia em 4 *slides*, ao adicionar um identificador em cada título, resolvemos os 6 erros apontados pelo *plugin* na página *home*. Sendo assim, podemos passar pela próxima categoria mostrada: erros de contraste.

15 Contrast Errors

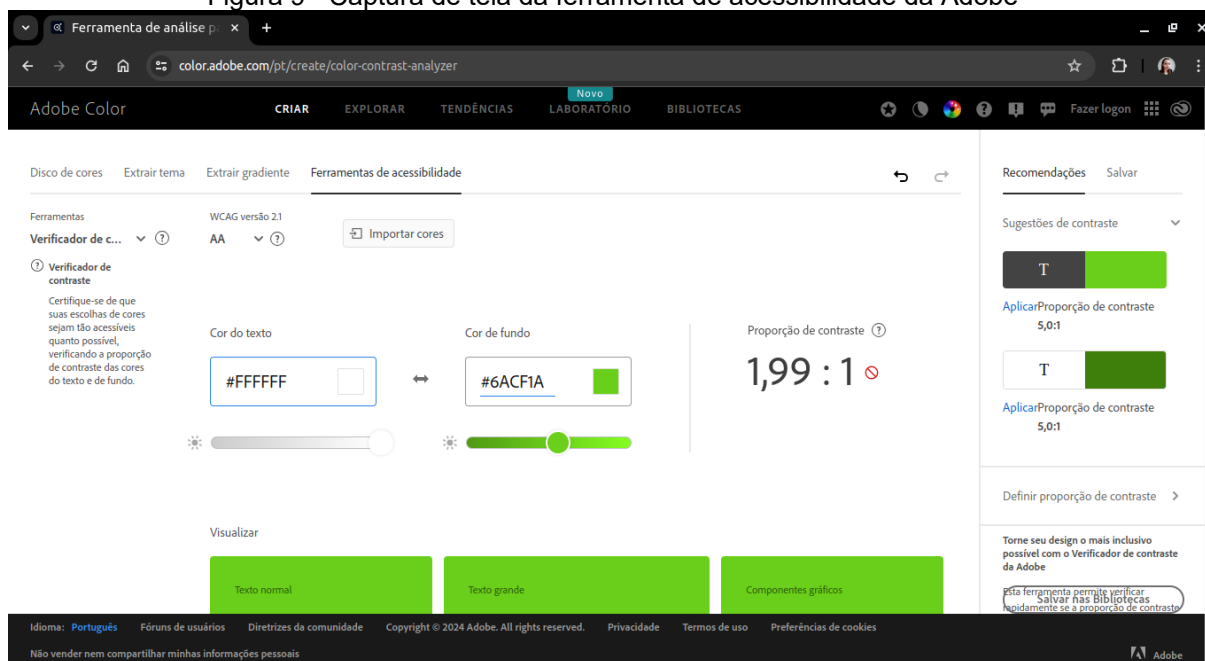
15 X Very low contrast

Começando pelo erro de contraste que diz respeito à âncora “Pular para o conteúdo”, tem-se que esse botão foi adicionado para ajudar os usuários de teclado a acessarem o conteúdo principal de forma mais rápida e não fica visível o tempo todo na tela, aparecendo apenas quando selecionado via teclado (esse assunto foi tratado no tópico 3.2.2.2 Navegação por teclado). O problema foi resolvido de forma simples, apenas foi adicionado o atributo “*style='color: #000000'*”, deixando o texto na cor preta com fundo branco.

Dois dos erros de contraste, eram referentes às *tags label* que foram removidas em uma etapa anterior. Já que as *tags* foram removidas, o problema foi solucionado.

Para ajudar a resolver os erros de contraste, foi utilizada também a ferramenta de acessibilidade da Adobe Color, que é uma plataforma *online* da Adobe para gestão de cores e paletas de cores, onde é possível testar os contrastes entre as cores, recebendo a proporção do contraste e sugestões, levando em conta das diretrizes da WCAG.

Figura 9 - Captura de tela da ferramenta de acessibilidade da Adobe



Fonte: Adobe, 2024

Quando adicionado o tom de verde e branco utilizado no título “Página inicial”, que é o próximo item destacado pelo WAVE, a ferramenta informa que a proporção do contraste é de 1,99:1, abaixo do recomendado pela WCAG (2023) que é de 3:1 para textos com fonte de tamanho maior que 18pt ou 14pt em *bold*, visto na diretriz 1.4.3. Seguindo sugestões da ferramenta da Adobe, o tom de verde #6acf1a (cor em hexadecimal) foi trocado pelo verde #305E0C (cor em hexadecimal), que tem uma proporção de contraste com o branco #FFFFFF de 7,69:1. Ao substituir o código da cor em todos os arquivos CSS, foram resolvidos onze erros de contrastes, sobrando apenas um a ser verificado.

O último erro de contraste apontado pelo *plugin* WAVE era o subtítulo “Visualizar tudo”. Para resolver o problema, foram utilizados os mesmos tons de branco e verde validados no título “Página Inicial”, assim mantendo a identidade de cores, mas ao mesmo tempo resultando no contraste necessário. Apenas foi invertido para fundo

branco com texto em verde, ao contrário do título “Página Inicial” logo acima dele, que usa texto em branco em fundo verde.

Para finalizar a etapa de contraste, foi utilizada a ferramenta de acessibilidade da Adobe para validar o contraste de cores do *header* e *footer*, que usam as mesmas cores. O contraste entre o fundo e os textos era de 3,13:1, abaixo do valor indicado pela diretriz 1.4.3 da WCAG (2023) para textos menores que 14pt em *bold*, que é de 4.5:1. O tom de verde do texto apenas foi escurecido um pouco, chegando na cor #202B1A, elevando o contraste para 4,87:1. Como toque final, foi adicionado ao *hover* dos textos o efeito de “*text-decoration: underline*”, de forma a evidenciar ainda mais o local onde o usuário está posicionado, já que antes era apenas utilizado uma leve mudança de cores, que poderia não ser notada por usuários com daltonismo ou outras deficiências visuais.

39 Alerts

1 X Skipped heading level
 9 X Possible heading
 1 X Redundant link
 27 X Noscript element
 1 X Underlined text

Terminados os erros, o próximo exame é nos alertas. O primeiro alerta apontado pela ferramenta é com relação a quebra hierarquia de títulos, já que dentro do componente de rodapé da página, foram adicionados apenas títulos *h4*, pulando a ordem de *h2* e *h3*. Sendo assim, apenas a troca de *h4* para *h2* já fez com que a ferramenta não apontasse mais esse alerta.

Os 9 apontamentos seguintes tratam de textos que poderiam ser *headings*, mas que não foram colocados como tais. Conforme a MDN Web Docs (2024i), é uma boa prática ter somente um título do nível *h1* na página e como já existe um elemento desse tipo, o título de “página inicial”, foi escolhido para ser *h2*. Depois, há 6 títulos que são *links* para páginas externas e trazem os nomes das categorias de produtos, como sucos e bebidas, doces e geleias e acessórios. Como o título com nome da

página atual foi tratado como *h2*, o padrão foi seguido e cada título receberá uma *tag h2* também.

Os dois títulos finais que foram apontados tratam de informações sobre os benefícios de produtos orgânicos e como funcionam as entregas. Como o foco da página é apresentar produtos, e há páginas específicas para falar sobre as entregas, foi definido a *tag h3* para eles.

A seguir, é sinalizado o alerta sobre um *link* duplicado para a página “carrinho”. Há na página dois componentes de menu de navegação no cabeçalho, mas que são usados de forma alternada para tamanhos de tela *mobile* e *desktop* e por isso, ambos os *links* não ficam acessíveis para o usuário ao mesmo tempo e nada precisou ser feito.

Os próximos 27 alertas dizem respeito à *tag noscript*. Ela é utilizada para que a página rode *scripts* quando o *Javascript* estiver desabilitado. A recomendação da ferramenta é garantir que o *Javascript* esteja acessível e ter em mente que essa *tag* não afeta a acessibilidade, já que poucos usuários desabilitam o *Javascript* da página. Como é apenas um aviso, e não afeta o funcionamento do *site*, as *tags* foram mantidas.

O último alerta diz respeito a um texto com *underline*. A própria ferramenta diz que já é uma prática utilizar *underline* para *links*, e por isso sugere remover esse estilo de textos simples. Ao analisar o código, foi visto que o elemento sinalizado possui o texto “Visualizar carrinho e finalizar compra” e pode ser clicado, levando o usuário ao carrinho de compras dele. Sendo assim, ele foi considerado um *link* e não precisou de ajuste.

28 Features

7 X Null or empty alternative text

20 X Linked image with alternative text

1 X Language

O próximo assunto é sobre as *features*, que são as funcionalidades que devem estar presentes no projeto para que ele seja considerado acessível.

Primeiramente foram sinalizados 7 apontamentos que dizem respeito a atributos *alt* vazios. O *plugin* sinaliza que esse atributo é importante para que pessoas que utilizam

leitores de tela possam entender do que se trata a imagem disposta na tela. Portanto, foram colocados textos descritivos para cada imagem apontada.

Em seguida, há 20 casos em que a ferramenta sugere que sejam revisados os atributos *alt* já existentes em imagens que tem *links*. A sugestão é que o texto alternativo seja de acordo com a imagem ou o *link*. Ao validar os textos desses atributos, foi notado que eles estavam todos em caixa alta, o que pode dificultar a leitura por tecnologias assistivas, podendo as palavras serem confundidas com siglas, assim como sinaliza o *Bureau of Internet Accessibility* (2024). Desse modo, os textos foram trocados por versões mais sucintas, assim ficando mais rápido de ser lidos. Também foi usada a formatação capitular, utilizando caixa alta e baixa.

Finalmente, é pedido para revisar a *tag* de linguagem. Como há uma *tag* informando que o conteúdo está em português brasileiro e não há conteúdos em outras línguas na página, não precisou ser ajustado.

25 Structural Elements

1 X Heading level 1

4 X Heading level 2

2 X Heading level 3

6 X Unordered list

3 X Inline frame

1 X Header

1 X Navigation

4 X Search

1 X Main content

1 X Footer

1 X Generic region

Agora é levantado sobre as validações de elementos estruturais, começando pela validação do conteúdo de *h1*. O elemento escolhido pelos desenvolvedores do *site* é o logotipo no *header*. Como ele contém um texto alternativo com o nome do *site*,

Armazém do Campo - Online, e trata-se da página inicial, foi mantido da forma como está.

A seguir, há que se analisar sobre os títulos *h2* e *h3* da página, sendo pedido pela ferramenta que seja validado se as posições hierárquicas desses conteúdos estão corretas. Alguns dos títulos apontados foram ajustados nas etapas anteriores e os demais tratam dos títulos presentes no carrossel de imagens e nas seções da página (rodapé e redes sociais), portanto foi mantido da forma que está.

A seguir foi feita a análise das listas não ordenadas, sendo 6 listas, 3 tratam de menus de *links* de navegação com as páginas do *site* e nenhum deles está contido dentro de uma *tag nav*, assim, a lista não ordenada foi mantida, mas foi envolta da *tag* de navegação.

A próxima lista foi utilizada para criar a funcionalidade de troca de *slides* no carrossel, sendo seus itens os botões de próximo *slide* e *slide* anterior que também não precisou de ajustes.

Há ainda uma última lista de *links*, mas dessa vez são os *links* externos, para as redes sociais da empresa. Por não ser um item de navegação do *site*, mas apenas uma lista de *links*, não foi adicionada a *tag nav* e nada foi alterado.

A última lista traz mensagens de instruções para o usuário sobre a utilização do *site*, como “Pressione a tecla de espaço e as teclas de seta para fazer uma seleção.”, o que são mensagens apenas para usuários de leitores de tela, já que são chamadas por atributos *aria* por outros elementos de maneira a descrever as regiões do *site* ou passos que o usuário deve fazer. Por não terem um intuito visual, mas apenas isolar as mensagens que o usuário pode receber, não precisou de alteração.

Continuando para a etapa de validar as *tags iframe* da página, o *plugin* pede que seja validado se o conteúdo é acessível ou pelo menos possui uma descrição, mas examinando-se o conteúdo, ele chama alguns arquivos *HTML* que parecem ser de uso interno da ferramenta de *e-commerce* Shopify. Já que os elementos estão escondidos visualmente e dos acessos via teclado e tecnologias assistivas através de atributos *HTML* e *CSS*, eles não foram alterados.

A seguir foi feita a validação de conteúdo do *header* da página, verificando se o que está contido nele é realmente conteúdo de navegação e a introdução da página. Dentro dessa *tag*, no código, foram colocados os menus superiores de navegação, assim como os botões de carrinho, acesso à conta e pesquisa. Como se trata, tanto

visualmente quanto hierarquicamente, da primeira parte do *site*, julgou-se que não é necessário fazer alterações.

É chegado então na verificação dos campos de pesquisa. A plataforma pede para que todos as áreas de pesquisa sejam marcadas com `“role='search’”` a fim de avisar as tecnologias assistivas a função daquele campo. Apenas foi confirmado que todos os campos de busca do *site* possuem esse atributo.

O próximo item pede que seja validado o elemento *main*. A ferramenta pede que seja confirmado se o conteúdo dentro dessa *tag* é o conteúdo principal da página. Ao analisar o código, verificou-se que os elementos fora da *tag main* são o *header* e o *footer*, enquanto a apresentação dos produtos e os textos de apoio sobre o *site* estão contidos na *tag main*. Assim, pôde-se considerar que o conteúdo principal da página está estruturado na *tag* correta.

Partindo para a análise do *footer* da página, a ferramenta sugere que seja analisado se o conteúdo dentro da *tag footer* realmente corresponde ao *footer* da página, que geralmente contém informações de *copyright* e *links* relacionados. A região contida nessa *tag* realmente é o final da página, que está visualmente separado do conteúdo principal por um fundo verde e contém *links* de navegação para onde o usuário pode ir ao finalizar a leitura da página atual. Então pôde-se considerar que o uso dessa *tag* está correto.

O último ponto sobre estrutura que foi analisado é sobre o `“role='region’”`, presente no *slider* da página. De acordo com a documentação do MDN Web Docs (2024c) essa *tag* diz respeito a uma região genérica, mas que seja importante para a navegação do usuário. Ela ainda traz que alguns critérios de acessibilidade devem ser atendidos quando se utiliza esse elemento, como conter o atributo *aria-describedby* ou *aria-label*. Mas na parte de “Melhores práticas” da documentação é dito que o ideal é utilizar o *HTML* semântico através da *tag section* que contém nativamente o `“role='region’”`. Por isso, apesar do código respeitar as regras ditas na documentação da MDN foi alterado para usar a *tag section*.

84 ARIA

27 X ARIA

9 X ARIA label
5 X ARIA description
7 X ARIA tabindex
2 X ARIA alert or live region
32 X ARIA hidden
2 X ARIA expanded

A última parte da análise que foi feita é com relação aos atributos *ARIA* presentes no código. O primeiro deles é a validação de todos os *role*, *state* e *property* presentes, totalizando 27 elementos. A recomendação da ferramenta é validar seu uso e trocar o atributo *ARIA* por uma *tag* de *HTML* semântico sempre que for possível.

Do total, 18 são ícones ilustrativos, com propósito apenas visual, como ícones de carrinho de compras e de lupa. Esses ícones estavam marcados como “*role=’presentation’*”, o que, de acordo com a MDN (2024b), faz o elemento não aparecer na árvore de acessibilidade e ser ignorado por tecnologias assistivas. Por esse motivo nenhum ícone precisou ser alterado.

Depois disso, foi visto que dois itens possuem o atributo “*current=’page’*”. A MDN Web Docs (2024e) diz que esse atributo serve para identificar um item como o elemento atual dentro de um contexto. No caso do *site* analisado, ele identifica que dentro de uma lista de *links*, o *link* marcado traz a *URL* da página onde o usuário está navegando e por isso nada foi alterado.

O próximo passo é a análise do atributo “*aria-atomic=’true’*”. A MDN Web Docs (2024d) traz que ele identifica uma área do *site* que sofrerá alguma modificação e isso será sinalizado ao usuário quando acontecer. No caso do *site* analisado, foi marcada a área de notificações, onde o usuário recebe mensagens como “sucesso” e “erro” das ações que ele executou. Por isso, nada precisou de alteração.

O próximo atributo que a ferramenta aponta é “*role=’listbox’*”. A MDN (2024a) define esse atributo como uma lista onde o usuário pode selecionar um item, mas que cada elemento dentro do *listbox* deve ter um atributo “*role=’option’*”. A partir disso, podemos determinar que o uso está correto, já que esse atributo foi usado para marcar o *slider* de imagens, onde é possível selecionar cada item, ou seja, cada *slide*, e sendo assim essas imagens estão marcadas com o atributo necessário.

Por último, é apontado sobre o atributo “*aria-pressed='false'*”. De acordo com a MDN (2024g), o atributo *aria-pressed* serve para apontar um botão como “botão de alternância”, ou seja, dar um estado ao botão que pode ser *true*, *false*, *mixed* ou *undefined*. No caso do objeto de estudo, foi usado para marcar o botão de “*start/pause*” do *slider* de imagens, permitindo que o usuário entenda se o *slider* de imagens está pausado ou rodando.

Terminadas as descrições dos *roles*, foi analisado sobre a *tag aria-label*. Das 9 *tags* encontradas, 4 são referentes a *inputs* marcados com “*aria-label='pesquisar'*”. O primeiro refere-se ao campo pesquisar do menu lateral, o segundo ao campo pesquisar presente no menu superior, e o terceiro e quarto aos campos pesquisar presentes no menu fixo que só aparece ao rolar a página para baixo, nas versões *mobile* e *desktop*. Como todos os campos são de pesquisa, pôde-se concluir que o uso do *label* está correto.

Os outros 5 atributos do tipo *aria-label* apontados pela ferramenta são referentes ao *slider* de imagem. O primeiro contém o texto “*slideshow*” mostrando que se trata de uma sequência de imagens; o segundo diz “*Slide 1, atual*”, mostrando qual é a etapa que está sendo mostrada no *slider*; o terceiro sinaliza o botão de pausa do *slider*, através do texto “Pausar apresentação de *slides*”; e os dois últimos são referentes aos botões de avançar e voltar do *slider* e apresentam os textos “*Próximo slide*” e “*Slide anterior*”. O único texto ajustado foi o “*slideshow*”, sendo alterado para “carrossel de imagens”, que seria uma tradução livre do termo. Apesar de “*Slide 1, atual*” não ser descritivo quanto ao que está apresentado na imagem do *slide*, a imagem contida no *slide* possui descrição no atributo *alt*, mostrando que o *aria-label* está em uso apenas para informar o estado do *slider*. As demais descrições são claras quanto ao significado de cada elemento que representam.

A etapa a seguir sugerida pela plataforma é validar os elementos que possuem *aria-describedby*. Esse atributo recebe um ou mais identificadores de elementos terceiros com o intuito de descrever o elemento que contém o atributo, parecido com a *tag label* que já foi tratada nesse texto, mas pode carregar descrições maiores. O primeiro elemento apontado, que é o carrossel de imagens, é descrito por um texto de instruções de como passar os *slides* do carrossel de imagens. Há mais 4 elementos com esse atributo, sendo eles os *slides* dentro do carrossel que têm como descrições

os respectivos títulos *h2* dentro deles, assim direcionando ao usuário a mensagem principal de cada *slide*.

O próximo objeto de análise é o atributo *tabindex*. A ferramenta WAVE pede para que seja validado o uso desse atributo recebendo valores diferentes de 0, já que isso muda a ordem de navegação do teclado. No caso do nosso objeto de estudo, temos sete elementos recebendo o atributo *tabindex* com o valor -1, sendo assim retirados da navegação de teclado. Desse total, seis se referem ao *slider* de imagens: o carrossel geral, seus quatro slides e os clones dos *slides* 1 e 4 do carrossel (usado para criar efeito de *slide* infinito, conforme visto na figura 7). Acredita-se que esse elemento recebeu a propriedade a fim de remover seu acesso via tecla *tab*, mas permitindo que o usuário acesse as informações contidas nele, porque, apesar da necessidade de acesso aos textos e imagens dentro do *slider*, permitir acessar os *slides* em si via teclado deixaria o usuário preso nesse elemento, já que quando atingisse o último *slide*, seria direcionado ao primeiro, e vice-versa.

Figura 10 - Captura de tela do código do carrossel de imagens

```
<div class="slick-track" style="opacity: 1; width: 8106px; transform: translate3d(-2702px, 0px, 0px);" role="listbox">
  ::before
  ><div class="hero_slide hero_slide--5f8e5eee-468d-4963-960c-d9e6e7fc8ef7 slick-slide slick-cloned" data-slick-index="-1" aria-hidden="
    true" style="width: 1351px;" tabindex="-1"> @</div>
  ><div class="hero_slide hero_slide--1601912975222 slick-slide" data-slick-index="0" aria-hidden="true" style="width: 1351px;"
    tabindex="-1" role="option" aria-describedby="slick-slide00"> @</div>
  ><div class="hero_slide hero_slide--0b21101f-491d-44e4-8175-cf2d8c090796 slick-slide slick-current slick-active" data-slick-index="1"
    aria-hidden="false" style="width: 1351px;" tabindex="-1" role="option" aria-describedby="slick-slide01"> @</div>
  ><div class="hero_slide hero_slide--47bd3d53-e529-4bb4-b3fc-0a2833a3293b slick-slide" data-slick-index="2" aria-hidden="true" style="w
    idth: 1351px;" tabindex="-1" role="option" aria-describedby="slick-slide02"> @</div>
  ><div class="hero_slide hero_slide--5f8e5eee-468d-4963-960c-d9e6e7fc8ef7 slick-slide" data-slick-index="3" aria-hidden="true" style="w
    idth: 1351px;" tabindex="-1" role="option" aria-describedby="slick-slide03"> @</div>
  ><div class="hero_slide hero_slide--1601912975222 slick-slide slick-cloned" data-slick-index="4" aria-hidden="true" style="width: 1351
    px;" tabindex="-1"> @</div>
  ::after
</div>
```

Fonte: Armazém do Campo, 2024

O último elemento com alteração do *tabindex* valendo 0 trata-se de um *iframe* que está totalmente fora de alcance do usuário tanto visualmente, quanto via teclado, e que foi usado para importar recursos usados pelo *Javascript*, e por isso não precisou de ajuste.

O próximo ponto diz respeito ao atributo *aria-live* que, segundo explicação da MDN Web Docs (2024f), serve para indicar ao usuário que uma determinada área sofreu alteração. No nosso objeto de estudo, temos 2 elementos marcados com esse atributo: A mensagem de erro e o botão de mover o *slide* do carrossel. O atributo de mensagem de erro está inacessível ao usuário até que um erro seja disparado, e por se tratar de algo que necessita atenção imediata, foi marcado como "*aria-*

live='assertive'” o que vai imediatamente informar ao usuário que o elemento foi alterado e ler seu conteúdo. Já o botão do *slider* foi marcado como *“aria-live='polite'”*, já que não é algo que necessite de pronta ação, pode esperar que a tecnologia assistiva termine as ações requeridas anteriormente para então ler que a área foi alterada. Assim, foi considerado que não há necessidade de ajustes.

O seguinte passo para validação são os elementos que contém o atributo *“aria-hidden='true'”*. A plataforma pede para que seja analisado se esses elementos realmente devem estar escondidos do acesso do usuário de tecnologia assistiva, já que, esse atributo somente remove o elemento da árvore de acessibilidade, mas não o remove visualmente da página.

Foram contados dezessete elementos do tipo SVG, que são ícones usados apenas para decoração, como lupa em botão de busca, logos de redes sociais para os respectivos *links* externos e “xis” para botões de fechar. Como esses elementos têm finalidade apenas visual, e há algum elemento relacionado a ele com uma descrição verbal para usuários de tecnologia assistiva, nada precisou ser feito. Juntamente com esses elementos do tipo SVG, podem ser inclusos dois elementos do tipo *span* que o conteúdo é apenas uma seta acompanhando um texto de *link*, ou seja, também tem o intuito de ser apenas visual.

Em seguida, há o atributo *aria-hidden* nos elementos de notificação de erro ou sucesso. A mensagem deve aparecer somente se o usuário realizar alguma ação que necessite o disparo dessas mensagens, como adicionar um item no carrinho de compras, por isso faz sentido esconder o elemento de ser acessado até que seja necessário.

Há cinco elementos escondidos visualmente que estão relacionados ao *slider* de imagens. São eles: o *container* de *slider*, os quatro *slides*, o clone do *slider* (para fim de efeito de *slider* infinito de imagens e pode ser visto na figura 7) e o contador do número de *slides*. O intuito é o mesmo do *tabindex*: o atributo *“aria-hidden='true'”* foi adicionado de forma a não prender o usuário no *slider* e deixar disponível para acesso somente os botões de navegação e o *slide* atual. Assim, o usuário consegue navegar entre os *slides* e não fica infinitamente indo do primeiro ao último ou do último ao primeiro *slide*.

Depois, foram analisados dois elementos *div* com o texto “promoção” que estão escondidos da árvore de acessibilidade. Esses elementos estão posicionados em

cima da imagem do produto, mostrando visualmente que ele está com preço reduzido. Porém, no preço do produto logo abaixo está escrito que o item está em promoção e o valor do item. Para que o usuário de tecnologia assistiva não precise ler duas vezes que o produto está em promoção, foi deixado acessível para ele somente o preço completo do item.

E por fim, há um elemento do tipo *script* que foi usado para importar recursos para o *Javascript*, então não precisa estar disponível para acesso do usuário.

Os últimos elementos que foram analisados são os que contém o atributo *aria-expanded*. Há dois itens que possuem essa informação, ambos são botões do menu lateral do tipo “hambúrguer”, e sinalizam que o menu está aberto ou fechado. A própria ferramenta sinaliza que é comum ter botões que controlam conteúdo expansivo utilizando esse atributo, então esses elementos foram considerados validados.

Finalizada a parte de análise utilizando ferramentas automatizadas, e com os ajustes feitos, acredita-se que temos o *site* funcionando com os requisitos mínimos solicitados pela WCAG (2023). Além disso, parte dos objetivos foi cumprida, já que foi elaborada uma lista de recomendações e ajustes a partir do *site* analisado.

A fim de avançar no tema e complementar a lista de melhorias, é preciso aprofundar a inspeção e considerar a navegação de alguns grupos de usuários, e para isso, foi feita a análise dos atributos do *site* de forma manual, simulando situações específicas.

3.2.2 Análise manual

Para a análise manual, foi criada uma lista de requisitos e casos de testes para validar as funcionalidades, examinando individualmente cada item para avaliar se o *site* está em conformidade com os padrões esperados, assim concluindo mais uma parte dos objetivos desse projeto. Os requisitos foram definidos com base no livro de FERRAZ (2020), assim como as documentações citadas no capítulo “2. Revisão bibliográfica”. Em seu capítulo, “11 Verificação de Acessibilidade”, o livro de FERRAZ (2020 p.205-207) fala sobre os processos de validação de acessibilidade, dentre eles, há o texto “11.5 Verificação Manual de Acessibilidade”, que contém os seguintes itens:

1. “Verifique se o *site* está acessível via teclado”
2. “Verifique o texto alternativo das imagens”
3. “Verifique o contraste de cores e tamanho de texto”
4. “Verifique se os vídeos e áudio tem alternativas em texto”

5. “Verifique se há uma forma de parar o conteúdo em movimento”
6. “Verifique a hierarquia de cabeçalhos”
7. “Teste sua aplicação com tecnologia assistiva”
8. “Peça ajuda para pessoas com deficiência”

Alguns desses tópicos já foram tratados no texto anterior sobre análise automatizada, e por isso foram considerados validados, assim como outros foram removidos devido o *site* não precisar da validação, como não haver vídeos para verificar se há conteúdo textual. Outro ponto é a análise por pessoas com deficiência que não pôde ser executada. Em paralelo a isso, alguns assuntos foram adicionados na lista de validação utilizando o mesmo livro e estão listados a seguir.

Do capítulo 6 do livro de FERRAZ (2020), dentro do texto “6.6 Formulários”, há um tópico chamado “Instruções e mensagens de erro”, onde o autor comenta sobre boas práticas com relação a orientações ao usuário e mensagens de erro.

Já no capítulo “7 Eliminando barreiras – *Design*”, temos o tópico “7.4 Redimensionar texto”, onde o autor cita que a recomendação da W3C é de que o usuário seja capaz de acessar o *site* utilizando o *zoom* em até 200% sem que a usabilidade seja alterada. Outro ponto que foi analisado foi retirado da documentação GAIA [s.d.], com foco em pessoas com deficiência cognitiva. No item “G02 - Textos”, da documentação, é dito que os textos devem ser “simples, evitando jargões, erros ortográficos, metáforas, abreviações e acrônimos, fazendo uso de termos, expressões, nomes e símbolos familiares ao contexto de seus usuários.”, e isso gerou o primeiro tópico da análise manual.

3.2.2.1 Leitura clara e objetiva

Requisitos: Textos simples e curtos são a forma de o usuário não ter dificuldade em entender o conteúdo ou se perder rapidamente ao deparar com blocos muito extensos de texto. Isso auxilia tanto pessoas neuro divergentes quanto pessoas idosas, que podem ter dificuldade para ler quantidades muito grandes de texto ou textos com diagramação muito larga.

Utilizar componentes que auxiliam no entendimento do usuário e na localização também são peças-chaves para a usabilidade, como o título da página localizado na aba do navegador, que indica para o usuário onde ele está navegando.

Casos de testes:

1. O usuário deve entender como navegar no *site* de forma clara e sem necessidade de ajuda externa.
2. O usuário deve entender de forma clara em que parte do *site* está.

Execução dos testes:

1. Começamos pelo título da página, que fica na aba do navegador. FERRAZ (2020) sugere que ao invés de colocar o nome do *site* e depois o título da página, que se coloque o nome da página e depois o nome do *site*, ou seja, “Início | Armazém do Campo *On-line*”, já que dependendo do tamanho da tela do usuário, o conteúdo pode não aparecer por completo, então a informação mais importante (onde o usuário está) deve ir primeiro.
2. Agora temos a validação pelos botões com representação em forma de ícones, que possuem as devidas descrições, menos o “carrinho de compras” e “pesquisa”, que só deixam claro seus papéis quando recebem interação, já que ao clicar no botão de busca, aparece o texto “pesquisar” e o carrinho de compras abre a página “Carrinho de compras”. Mas como são ícones e funções comuns em outros *sites*, entende-se que está clara a funcionalidade deles.
3. O próximo elemento no *site* é o carrossel. Cada *slide* possui um título, uma imagem e um *link*. No primeiro *slide*, com o título “Visite nossa loja”, o texto do *link* é “Armazém do Campo São Paulo”, mas poderia ser trocado para “Nosso endereço”, deixando claro que a página a seguir será a localização da loja. O *slide* 2, fala sobre compra de orgânicos e tem o seu *link* e textos condizentes com essa ação, então entende-se que a mensagem está clara. Assim também o *slide* 3 que não possui *link* e apenas tem uma imagem de produtos com o texto “Agroecologia é o caminho”. Por fim, no *slide* 4, o *link* com o texto “Boné MST vermelho, compre aqui” vai direto para a página de compra do boné vermelho, mas poderia redirecionar para uma página de acessórios onde o usuário pode escolher qual o modelo do boné que deseja, já que a imagem do *slide* possui várias opções de bonés.
4. Entrando na parte de produtos da página, temos um *link* “Página inicial”, que redireciona o usuário para uma página com apenas produtos. Esse texto e o título da página para qual ele redireciona poderiam ser trocados para “Página de produtos”, já que não pode haver duas páginas iniciais do *site*.

5. Ainda na parte de produtos, as descrições de cada mercadoria estão em caixa alta, o que pode dificultar a leitura, segundo o manual *Digital Accessibility* da Harvard University [s.d.], além de prejudicar a varredura por *sites* de busca.
6. Na seção de coleções, onde há as categorias de produtos disponíveis no *site*, foram encontrados dois problemas a serem resolvidos. Os blocos que separam cada item da coleção parecem não seguir a diretriz 1.3.1 da documentação WCAG (2023), que diz que a organização da estrutura da página deve ser compreendida pelo usuário independente da forma de acesso. Ainda podemos citar a diretriz G11 da documentação GAIA [s.d.] que fala sobre a organização visual do texto, citando que o conteúdo relacionado precisa estar visualmente próximo, e separado dos demais. Porém, a forma como as informações de cada categoria está disposta pode deixar o usuário confuso sobre qual título pertence a qual conjunto. O ideal é que seja adicionada uma borda em volta de cada unidade ou que o espaçamento seja alterado para separar visualmente cada item da coleção. O outro ponto de ajuste é a remoção da coleção vazia, que possui o texto “Título da coleção” e uma imagem de marcação.
7. O bloco a seguir é composto por três colunas, sendo a primeira um *GIF* de uma imagem de café com o texto “Produtos livres de agrotóxicos são a melhor opção”, e as outras duas são textos sobre os benefícios dos orgânicos e como funcionam as entregas. O *GIF* poderia ser trocado por um texto fixo em *HTML*, afinal, a imagem não pode ser parada pelo usuário e o fundo com textura prejudica a leitura do texto. E os textos ao lado poderiam ser alinhados à esquerda, como sugerido pela GAIA [s.d.], pois orientar os textos no centro ou à direita prejudicam a ordem de leitura, criando bordas irregulares nas linhas do bloco.
8. O *site* possui alguns textos e *links* sobre dúvidas de entrega, pagamento e alteração de pedido que poderiam ser consolidados em uma página de “Perguntas frequentes”.
9. Para finalizar, o conteúdo no geral precisa ser revisto, dado que há produtos em categorias que não os representam, como a presença de produtos alimentícios na categoria de acessórios, ou até mesmo mercadorias que não possuem categoria alguma, visto que na página de produtos, ao tentar filtrar os

itens, só existem as opções “todos”, “mel” e “terra conquistada”, apesar de haver diversos artigos de comidas, bebidas e acessórios.

3.2.2.2 Navegação por teclado

Requisitos: O usuário deve ser capaz de navegar pelo *site* usando somente o teclado, conforme traz a documentação WCAG (2023) em sua diretriz 2.1.1. A navegação por esse meio deve ser fácil e circular, voltando o usuário para o início quando ele alcançar o final de uma página.

Dar a opção de pular para determinadas partes do *site* através de um botão, que pode permanecer visualmente escondido, é uma maneira de agilizar o fluxo de navegação dentro do *site*, conforme citado na diretriz 2.4.1 da WCAG (2023).

A navegação deve ainda seguir o mesmo fluxo tanto via teclado, quanto visualmente, ou seja, se um título é mostrado no topo da página, qualquer usuário deve acessá-lo no começo de sua visita ao *site*.

Quanto ao visual da página, é preciso deixar claro para o usuário o elemento em foco, para que ele saiba qual parte do *site* ele selecionou para que a navegação e interação com os itens de possam ser eficientes.

Casos de testes:

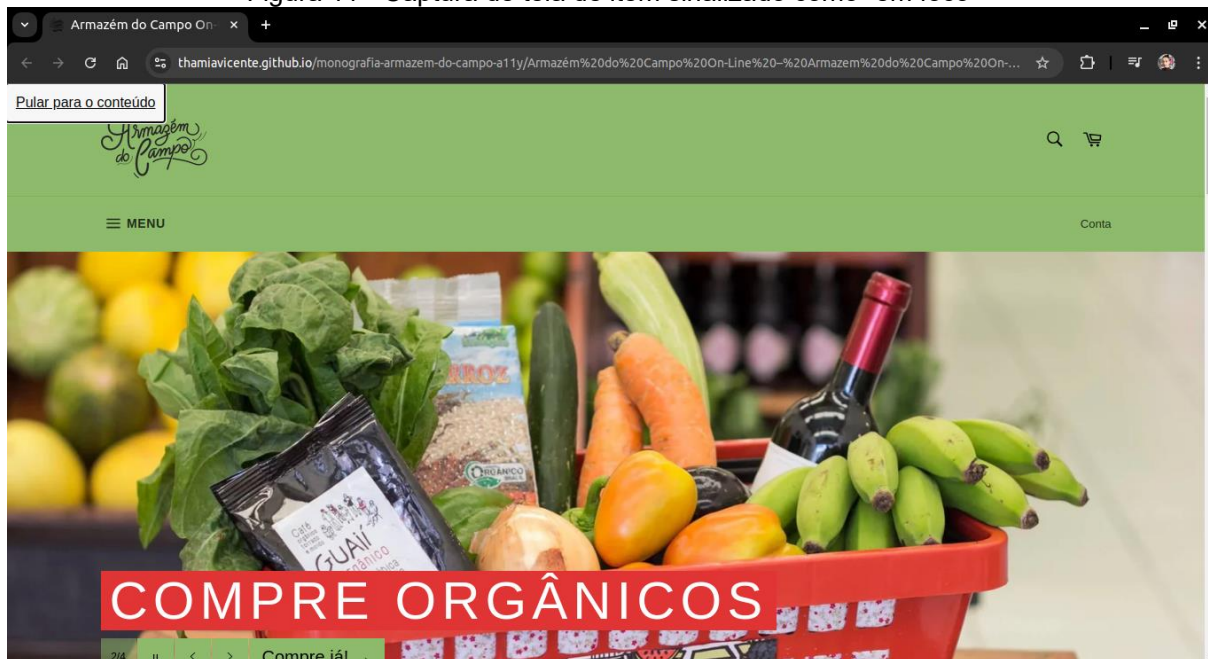
1. O usuário deve conseguir entrar nas páginas utilizando os menus e submenus, assim como as funcionalidades acionadas via botões e âncoras somente pelo teclado.
2. O usuário deve conseguir pular para o conteúdo principal sem precisar passar por todos os itens dos menus usando somente o teclado.
3. O usuário de teclado deve sempre saber onde está o foco da navegação.

Execução dos testes:

1. É possível acessar os itens dos menus tanto no *header* quanto no *footer*, entrar nas páginas de cada produto, ver as páginas de coleções, acessar as redes sociais, assim como campos de busca e realizar pesquisas somente utilizando o teclado.
2. Há um botão de “Pular para o conteúdo” que ajuda o usuário a ir diretamente para a parte principal do *site* sem ter que passar por todos os itens do *header*.
3. O elemento em foco sempre recebe uma borda preta, sinalizando que ele está selecionado. Em alguns casos, o foco é enfatizado por outros elementos

visuais, como nos itens do menu *footer* onde o texto também é sublinhado quando o item está em foco.

Figura 11 - Captura de tela de item sinalizado como “em foco”



Fonte: Armazém do Campo, 2024

3.2.2.3 Navegação por leitor de tela (NVDA)

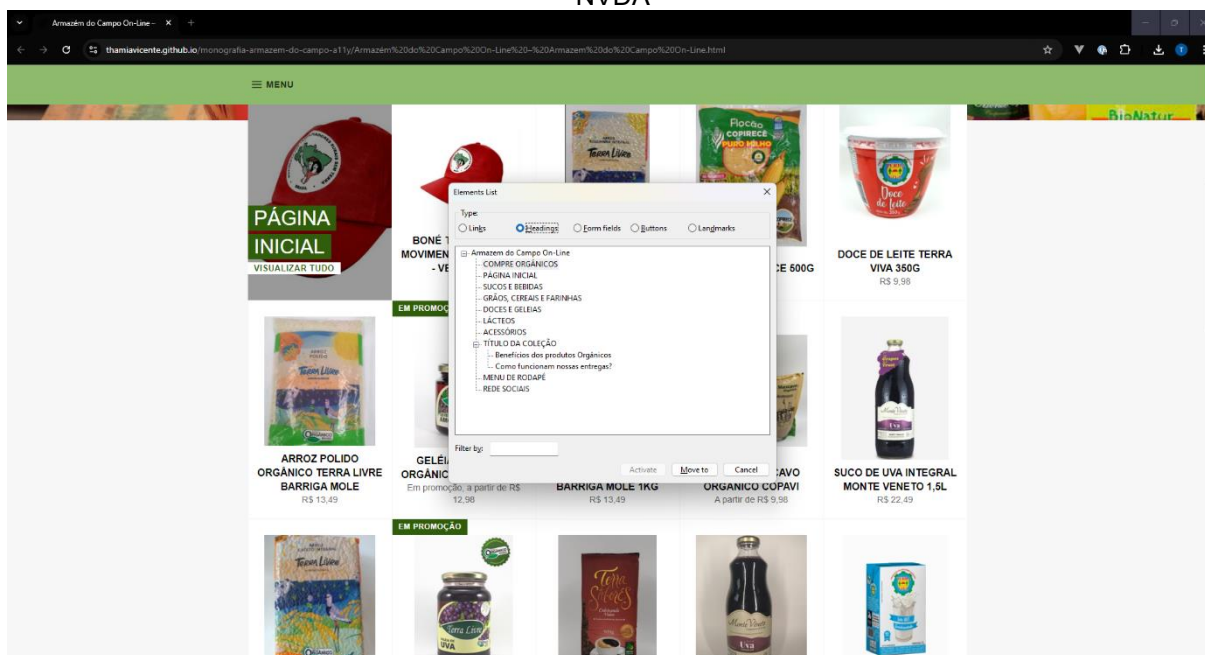
Requisitos: O usuário deve ser capaz de navegar pelo *site* sem precisar de recursos visuais, tendo o apoio de um leitor de tela. Para isso, as informações devem ter textos claros e objetivos, mostrando para o usuário o que ele está acessando e para onde ele será direcionado, evitando, por exemplo, a utilização de *links* com texto “clique aqui”, e empregando um conteúdo mais descritivo.

Toda representação visual deve conter seu correspondente em texto, sendo imagens, ícones, vídeos ou qualquer outra informação necessária à navegação. Da mesma forma, o usuário não deve ter seu fluxo atrapalhado por conteúdos que não passam informações importantes, como por exemplo, um botão com ícone de lupa, que deve ser sinalizado como “campo de busca” (ou similar), ao invés de “ícone de lupa”. Caso um conteúdo precise de descrição adicional, pode ser utilizado um documento ou *link* externo onde o usuário pode acessar essa informação.

Para a navegação via leitores de tela, é importante que o *site* utilize as *tags* de forma correta e hierárquica, ordenando os conteúdos por ordem de prioridade, para que seja lido da forma correta. Essa parte será importante para que usuários possam, por

exemplo, navegar em uma lista de títulos para encontrar de forma rápida o conteúdo que desejam.

Figura 12 - Captura de tela do site Armazém do Campo *On-line* analisado através da ferramenta NVDA



Fonte: Armazém do Campo e NVDA, 2024

A ferramenta selecionada para a validação do acesso sem recursos visuais é a NVDA, que foi indicada pelo livro de FERRAZ (2020). Trata-se de um leitor de tela criado pela empresa NV Access, fundada por dois amigos que se conheceram em um acampamento para cegos, e tem o lema “empoderar vidas através de acesso não visual à tecnologia” (NVAccess, 2024, tradução nossa). O *software* possui tem a seguinte usabilidade: através do teclado, o usuário navega pela tela do computador, e durante esse processo o leitor usa recursos de áudio para sinalizar ao usuário as respostas das ações solicitadas e a leitura dos itens selecionados. Esse *software* em específico só possui compatibilidade com o sistema operacional Windows, mas há outros que podem ser usados por usuários de diferentes sistemas.

Para realização dos testes, a parte visual do *site* não foi utilizada, mantendo a tela do computador desligada. Para simular a navegação de uma pessoa com cegueira, utilizou-se apenas o teclado e o áudio provido pelo leitor de tela.

Casos de testes:

1. O usuário com deficiência visual, deve entender uma imagem que seja importante, sem ver seu conteúdo visual, utilizando apenas o texto alternativo ou alguma descrição adicional.
2. O usuário com deficiência visual, deve conseguir entender o que os botões fazem quando sua representação visual for um ícone, mas sem ter o fluxo atrapalhado por textos alternativos de imagens não relevantes, como ícones decorativos.
3. O usuário com deficiência visual, deve ser capaz de entender para onde os *links* o levarão, através de um texto claro, em conformidade com a diretriz 2.4.4 da WCAG (2023).
4. O usuário com deficiência visual, deve ser informado quando algum conteúdo da página for modificado, com ou sem sua interação.
5. O usuário com deficiência visual, deve conseguir navegar pelos títulos de forma hierárquica usando o teclado.

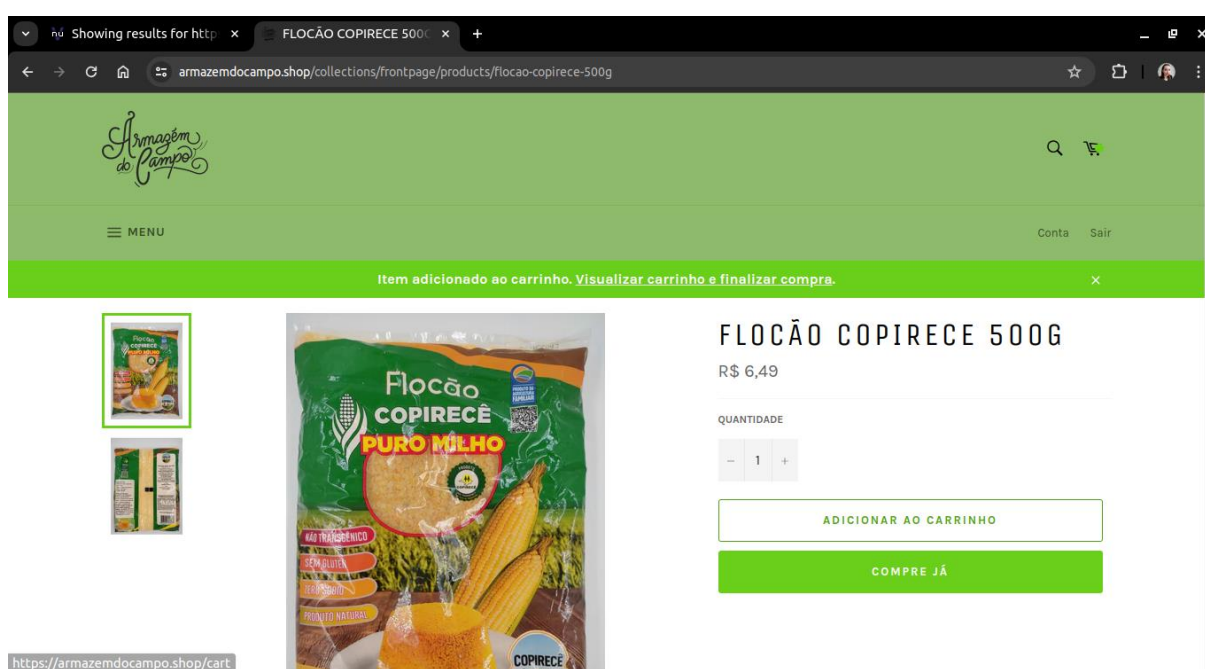
Execução dos testes:

1. As imagens de produtos e itens do carrossel de imagem possuem textos alternativos que descrevem o que é retratado. Em adição a isso, os botões e *links* que possuem representação visual de ícone, tem sua imagem ignorada, descrevendo apenas sua função.
2. Os *links* presentes na página passam a informação do que fazem de forma clara, como por exemplo, a mensagem de sucesso que aparece ao adicionar produtos no carrinho de compras, que utiliza o texto “Visualizar carrinho e finalizar compra” ao invés de “Clique aqui”.
3. As partes do *site* que podem ser alteradas pelo usuário, como adicionar itens no carrinho de compras, ou mudar os *slides* do carrossel, sinalizam ao usuário que elas sofreram alterações, assim como as mensagens de erro e sucesso.
4. Com o leitor de tela ligado, ao clicar na tecla “h” é possível navegar pelos títulos da página seguindo o fluxo: logo do *header*, títulos do carrossel, “visualizar tudo” na seção de produtos, títulos das coleções, título rodapé da página e título de redes sociais.

Para completar a análise com o leitor de tela, foi feito um fluxo de compra dentro do *site*, de modo a identificar possíveis falhas no processo. A seguir temos o passo a passo seguido, juntamente com as conclusões:

1. Após acessar a página inicial, foi navegado até a seção que contém produtos e selecionado o produto “Flocão Copierece 500g”. Foi possível entender qual produto se tratava e o preço, além de acessar a página do produto.
2. Através dos botões “Adicionar produto” e “Remover produto”, pôde-se selecionar uma quantidade de produtos, mas também editar a quantidade de produtos necessária. E quando selecionado o campo “Quantidade” foi falado a quantidade final.
3. Ao clicar no botão “Adicionar ao carrinho”, uma mensagem indicou que a mercadoria foi adicionada ao carrinho de compras e um *link* para acessar a página de compras foi sinalizado.

Figura 13 - Captura de tela da mensagem de sucesso



Fonte: Armazém do Campo, 2024

4. Na página de carrinho de compras foi possível ouvir um resumo da compra, passando pelos itens adicionados e suas devidas quantidades, além de alterar a quantidade de produtos na lista.
5. O próximo passo é um formulário de cadastro, onde todos os campos possuem informações acessíveis para orientar ao usuário os dados necessários.
6. Passando para a página de seleção de frete, ao selecionar as opções de entrega, pôde-se ouvir o valor do pedido atualizado, além da quantidade de dias úteis necessários e a forma de entrega.

7. A última página dentro do próprio *site* do Armazém do Campo é a seleção de formas de pagamento, onde foi encontrado um resumo da compra e informações adicionais de faturamento, como pedido de CPF ou CNPJ. Essa foi a única etapa onde foi encontrado um problema de acessibilidade. Visualmente havia a informação de que as formas de pagamento apareceriam ao finalizar a compra, ao redirecionar o usuário para a página de um *site* terceiro, mas ao navegar sem utilizar a visão, o leitor de tela passa direto por essa informação e o usuário não é avisado de como vai escolher a forma de pagamento.
8. A etapa de pagamento fica dentro do *site* “Pagar.me” onde alguns ajustes de acessibilidade podem ser feitos. Foi pedido ao usuário que novamente digitasse todos os seus dados, aumentando a quantidade de passos necessários para realizar a compra. Ainda foi visto que o leitor de tela passa direto pelas opções de pagamento e resumo da compra, ou seja, se o usuário não souber visualmente que as opções estão ali, ele não é avisado sobre elas. Essas informações também não possuem foco visível, trazendo dificuldade para usuários que dispõem da visão, mas navegam pelo teclado.

3.2.2.4 Análise dos controles de conteúdo animado

Requisitos: De acordo com a diretriz 2.2.2 da documentação WCAG, deve-se dispor ao usuário alguma forma de pausar conteúdos animados, como áudio, vídeo ou carrossel ajudando o usuário a lidar com elementos que podem causar distrações.

Vale pontuar que os elementos não podem piscar mais de 3 vezes por segundo, o que pode causar convulsões, tontura, enxaqueca ou outros problemas a pessoas fotossensíveis, segundo o item 2.3.2 da WCAG.

Casos de testes:

1. O usuário, deve ser capaz de pausar conteúdos animados e não essenciais utilizando *mouse* ou teclado.

Execução dos testes:

1. É possível controlar o carrossel de imagens, utilizando o botão de pausar/reproduzir e os botões de avançar/voltar o item. Além disso, foi possível entrar nas páginas relacionadas em cada item do carrossel.

2. O único elemento que não pode ser pausado foi o *GIF* contendo uma frase e uma imagem de grãos de café. Essa imagem poderia ser substituída por um texto feito no *HTML*, conforme comentado no item 3.2.2.1.

Figura 14 - Captura de tela do controle de conteúdo do carrossel



Fonte: Armazém do Campo, 2024

3.2.2.5 Utilização de *zoom* em até 200%

Requisitos: O usuário deve poder visualizar o conteúdo utilizando o *zoom* em até 200%, sem que a página quebre ou fique ilegível. O *layout* pode ser trocado para mostrar a versão *mobile*.

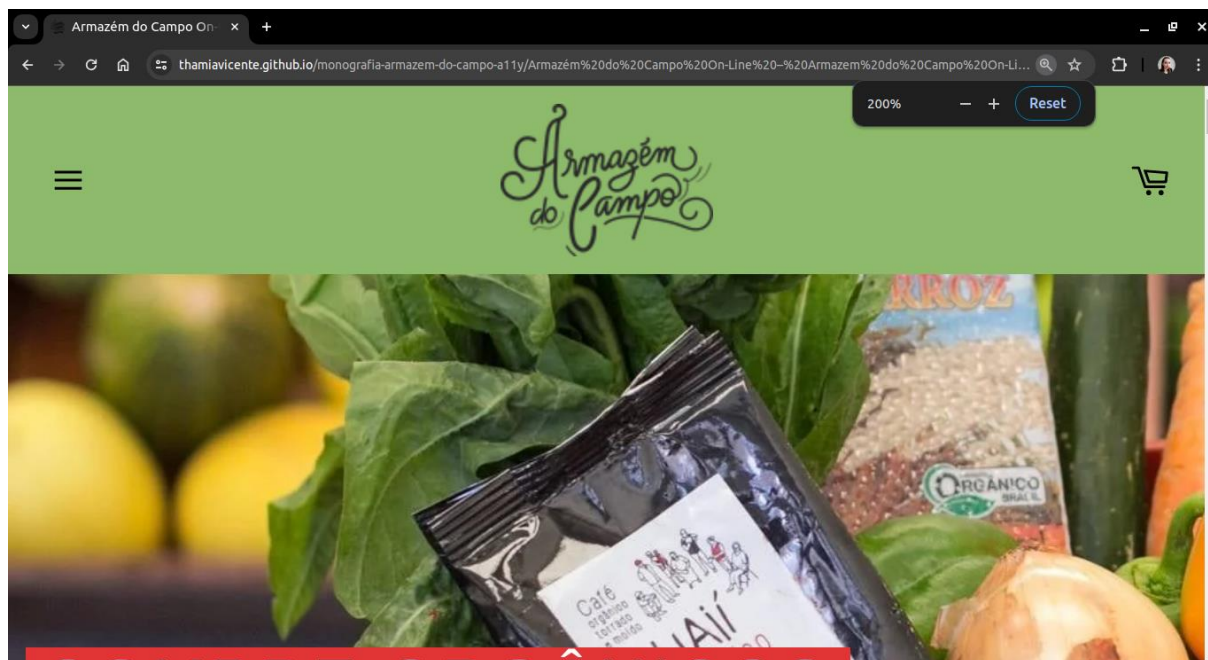
Casos de testes:

1. O usuário com deficiência visual, deve conseguir aumentar o conteúdo em até 200% para melhor visualização.

Execução dos testes:

1. Com o *zoom* em 200%, o *site* mostra o *layout* versão *mobile*, o que faz com que o conteúdo fique maior sem perder a legibilidade ou cortar o que está contido na página.

Figura 15 - Captura de tela do *site* analisado utilizando o *zoom* em 200%



Fonte: Armazém do Campo, 2024

3.2.2.6 Análise da informação mostrada em caso de erro

Requisitos: O usuário deve receber uma mensagem em caso de erro que seja clara e não passe despercebida. É importante não utilizar somente cor para sinalizar um erro ao usuário, tendo um apoio textual para usuários com deficiência visual, entrando em conformidade com a diretriz 1.4.1 da WCAG (2023).

Casos de teste:

1. O usuário deve ser sinalizado em caso de erro de sistema ou inserção de dados errados de forma clara e perceptível, por meio de informação verbal eficiente e não apenas utilizando mudança de cores.

Execução dos testes:

1. Ao tentar autenticar o usuário no *site* com *e-mail* e/ou senha incorretos, aparece uma mensagem de erro textual em cima do campo do formulário de acesso sobre a falha de autenticação.
2. Quando adicionado 25 unidades de um determinado produto (adoçante calda de agave), foi exibida uma mensagem de que não é possível adicionar mais produtos desse mesmo no carrinho de compras, logo abaixo do *header* da página.

Figura 16 - Captura de tela de mensagem de erro

Activities Google Chrome ago 8 18:37

ADOÇANTE CALDA DE A

armazemdocampo.shop/products/adocante-calda-de-agave-azul-sitio-do-moinho-330g?_pos=1&_sid=4426faec8&_ss=r

Armazém do Campo

MENU Conta

Você não pode adicionar mais ADOÇANTE CALDA DE AGAVE AZUL SÍTIO DO MOINHO 330G ao carrinho.

ADOÇANTE CALDA DE AGAVE AZUL SÍTIO DO MOINHO 330G

R\$ 38,49

QUANTIDADE

- 25 +

ADICIONAR AO CARRINHO

COMPRE JÁ

Fonte: Armazém do Campo, 2024

4. CONCLUSÃO E TRABALHOS FUTUROS

A acessibilidade é fundamental para que os usuários tenham uma melhor experiência ao utilizar um *site*, seja uma pessoa com deficiência ou não. Por isso, é indispensável pensar sobre critérios de acessibilidade ao pensar uma aplicação para *internet*.

Levando isso em conta e conforme previsto nos objetivos do projeto, foi realizada a validação das funcionalidades de um *site* já existente na *web*, identificando barreiras de acessibilidade que comprometem o uso por pessoas com deficiência. A partir desse levantamento, no item 3.2, foi estabelecida uma lista de etapas a serem seguidas, fundamentada nas principais diretrizes de acessibilidade. Essa lista foi detalhada em requisitos e casos de teste para avaliar a acessibilidade do *site*, com recomendações de ajustes que, posteriormente, foram aplicadas através de um repositório no GitHub. Com a implementação dessas melhorias, o *site* passou a oferecer uma experiência mais inclusiva e acessível, facilitando a navegação para pessoas com deficiência.

Após a conclusão do desenvolvimento, pôde-se ver que muitas partes do código precisaram ser reavaliadas e alteradas de forma a atender os critérios propostos pelas documentações utilizadas, o que poderia ser evitado caso a acessibilidade fosse pensada como uma regra logo no início do projeto. Além de ser custosa, a alteração com o projeto pronto pode abrir brechas para *bugs*, já que o código precisa de modificação. Desse modo, recomenda-se que os ajustes sejam feitos por partes, de preferência isolando o código alterado. Pode-se começar pelas partes críticas, ou por componentes que já estão previstos para serem mudados, seja para consertar problemas ou ganhar novas funcionalidades.

O tipo de acessibilidade necessária pode variar de aplicação para aplicação, por isso é importante listar os requisitos do seu sistema, sempre validando as regras de negócio com as necessidades dos seus usuários. Um *site* para pessoas com diabetes, por exemplo, pode demandar mais recursos para pessoas com deficiência visual do que para outros tipos de deficiência, já que a doença pode levar a esse tipo de deficiência.

Há diversas formas que esse trabalho pode ser complementado, sendo uma delas, a adição de material sobre editores de página, que tornem o conteúdo mais flexível para cada usuário, como aumentar os contrastes entre os elementos, ou até mesmo tecnologias assistivas dentro do próprio *site*, como tradutores de LIBRAS.

Outro ponto é expandir as análises feitas para as outras páginas do *site*, como produtos e carrinho de compras, já que esse projeto cobre primordialmente a página inicial do *site*.

Ainda pode-se trazer pessoas com deficiências diversas para complementar o trabalho com sugestões e comentários, já que são o público-alvo das melhorias apresentadas por esse trabalho.

5. REFERÊNCIAS BIBLIOGRÁFICAS

ADOBE COLOR. **Ferramenta de análise para verificação de contraste de cores.** 2024. Disponível em: <https://color.adobe.com/pt/create/color-contrast-analyzer>. Acesso em: 28 fev. 2024.

ARMAZÉM DO CAMPO. Armazém Do Campo Shop. São Paulo, 2024. Disponível em: <https://armazemdocampo.shop/>. Acesso em: 5 fev. 2024.

BUREAU OF INTERNET ACCESSIBILITY. **All-Caps Headings: Are They Bad for Accessibility?**. 2024. Disponível em: <https://www.boia.org/blog/all-caps-headings-are-they-bad-for-accessibility>. Acesso em: 30 jun. 2024

BRASIL. **Convenção Sobre Os Direitos Das Pessoas. Com Deficiência.** Brasília, DF: Presidência da República, [2009]. Disponível em: https://www.planalto.gov.br/ccivil_03/_ato2007-2010/2009/decreto/d6949.htm. Acesso em: 31 jan. 2023.

CHOU, Joyce; DIETRICH-MULLER, Derek; DVORKINA, Elena; HOLMES, Kat; KILE, Nathan; KIM, Doug; MALEKZADEH, Sogol; MORRIS, Sarah; PRICE, Margaret; SHUM, Albert; WOOLERY, Kris. Inclusive 101 Guidebook. Microsoft, 2016. *E-book*. Disponível em: <https://inclusive.microsoft.design/tools-and-activities/Inclusive101Guidebook.pdf>. Acesso em: 2 jan. 2024.

COORDENADORIA NACIONAL PARA INTEGRAÇÃO DA PESSOA PORTADORA DE DEFICIÊNCIA. Ata da VII Reunião do Comitê de Ajudas Técnicas realizada em 13 e 14 de dez. de 2007, Brasília. Disponível em: https://www.assistiva.com.br/Ata_VII_Reuni%C3%A3o_do_Comite_de_Ajudas_T%C3%A9cnicas.pdf. Acesso em: 2 jan. 2024.

DIGITAL ACCESSIBILITY. **Design for readability | Digital Accessibility.** [s.d]. Disponível em: <https://accessibility.huit.harvard.edu/design-readability>. Acesso em: 4 jul. 2024.

ESTATÍSTICAS SOCIAIS. PNS 2019: País tem 17,3 milhões de pessoas com algum tipo de deficiência. **Agência De Notícias – IBGE**, 26 ago. 2021. Disponível em: <https://agenciadenoticias.ibge.gov.br/agencia-sala-de-imprensa/2013-agencia-de-noticias/releases/31445-pns-2019-pais-tem-17-3-milhoes-de-pessoas-com-algum-tipo-de-deficiencia>. Acesso em: 17 out. 2022.

FERRAZ, Reinaldo. **Acessibilidade na Web**: Boas práticas para construir sites e aplicações acessíveis. 1. ed. São Paulo: Casa do Código, 2020. p. 1-246.

FLOR, Carla da Silva; ULBRICHT, Vânia; VANZIN, Tarcisio. Recomendações da WCAG 2.0 (2008) e a acessibilidade de surdos em conteúdos da Web. **Revista Brasileira de Educação Especial**, Web, n. 19, p. 1, jun. 2013. DOI: <https://doi.org/10.1590/S1413-65382013000200002>. Disponível em: <https://www.scielo.br/j/rbee/a/Z97pcT9qP5nTWbRQ56NCbgt/?lang=pt>. Acesso em: 7 fev. 2024.

GAIA. **GAIA - Guia de Acessibilidade de Interfaces para Autismo**. [s.d.]. Disponível em: <https://gaia.wiki.br/>. Acesso em: 14 jan. 2024.

GOV. **V-Libras – Governo Digital**. [s.d.]. Disponível em: <https://www.gov.br/governodigital/pt-br/vlibras>. Acesso em: 17 out. 2022.

ISO 25000. **ISO/IEC 25010**. 2024. Disponível em: <https://iso25000.com/index.php/en/iso-25000-standards/iso-25010> Acesso em: 3 nov. 2024

JUCESP. **Pesquisa de empresa: Comercio de Alimentos da Terra LTDA**. 2024. Disponível em: https://www.jucesponline.sp.gov.br/Pre_Visualiza.aspx?nire=35233719538&idproduto=. Acesso em: 5 fev. 2024.

JÚNIOR, Alessandro Feitosa. Número de lojas virtuais cresce 40% em 2020 com empurrão da pandemia. **G1 – GLOBO**, 2020. Disponível em: <https://g1.globo.com/economia/tecnologia/noticia/2020/08/26/numero-de-lojas->

virtuais-cresce-40-por-cento-em-2020-com-empurrao-da-pandemia.ghhtml. Acesso em: 8 nov. 2023.

MARCO. **MARCO RESEARCH: Post Covid Consumer Behaviour II.** 2022. Disponível em: https://www.themarcosurvey.com/_files/ugd/13bc5f_5676093452cd452c8d7e9175c9fa2ba2.pdf. Acesso em: 5 fev. 2024.

MDN WEB DOCS. **ARIA: listbox role.** 2024a. Disponível em: https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Roles/listbox_role. Acesso em: 27 mai. 2024.

MDN WEB DOCS. **ARIA: presentation role.** 2024b. Disponível em: https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Roles/presentation_role. Acesso em: 14 mai. 2024.

MDN WEB DOCS. **ARIA: region role.** 2024c. Disponível em: https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Roles/region_role. Acesso em: 13 mai. 2024.

MDN WEB DOCS. **Aria-atomic.** 2024d. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Attributes/aria-atomic>. Acesso em: 14 mai. 2024.

MDN WEB DOCS. **Aria-current.** 2024e. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Attributes/aria-current>. Acesso em: 14 mai. 2024.

MDN WEB DOCS. **Aria-live.** 2024f. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Attributes/aria-live>. Acesso em: 3 jun. 2024.

MDN WEB DOCS. **Aria-pressed.** 2024g. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Attributes/aria-pressed>. Acesso em: 27 mai. 2024.

MDN WEB DOCS. **Header: The Header element**. 2024h. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/header>. Acesso em: 20 jun. 2024.

MDN WEB DOCS. **H1–h6: The HTML Section Heading elements**. 2024i. Disponível em: https://developer.mozilla.org/en-US/docs/Web/HTML/Element/Heading_Elements. Acesso em: 30 jun. 2024.

NV ACCESS. NV Access | About NV Access. 2024. Disponível em: <https://www.nvaccess.org/>. Acesso em: 25 jun. 2024

NITAHARA, Akemi. Estudo mostra que pandemia intensificou uso das tecnologias digitais. **Agência Brasil**, Rio de Janeiro, 25 nov. 2021. Disponível em: <https://agenciabrasil.ebc.com.br/geral/noticia/2021-11/estudo-mostra-que-pandemia-intensificou-uso-das-tecnologias-digitais>. Acesso em: 27 jun. 2023.

SEMRUSH. **Domain Overview - armazenadocampo.shop**. 2024. Disponível em: <https://www.semrush.com/analytics/overview/?q=https%3A%2F%2Farmazenadocampo.shop%2F&protocol=https&searchType=domain>. Acesso em: 5 fev. 2024.

SEMRUSH. **Semantic HTML: What It Is and How to Use It Correctly**. 2022. Disponível em: <https://www.semrush.com/blog/semantic-html5-guide/>. Acesso em: 29 abr. 2024.

SILVA, Odenilza Gama da; NOGUEIRA, Antônia Fernanda de Souza. Comparando aspectos gramaticais de português e de LIBRAS. **ANAIS - I Colóquio de Letras da FALE/CUMB**, Pará, p. 1-9, fev. 2014. Disponível em: https://www.coloquiodeletras.ufpa.br/downloads/i-coloquio/anais/33_ODENILZA.pdf. Acesso em: 13 mar. 2024.

TECNOVISÃO. **QBraille XL**. 2024. Disponível em: <https://tecnovisao.net/produtos/qbraille-xl/>. Acesso em: 3 nov. 2024.

UDACITY. **Web Accessibility**. [s.d.]. Disponível em: <https://learn.udacity.com/courses/ud891>. Acesso em: 28 fev. 2023.

W3C. **Web Content Accessibility Guidelines (WCAG) 2.1**. 2023. Disponível em: <https://www.w3.org/TR/WCAG21/>. Acesso em: 31 jan. 2023.

W3C. **Introduction to Web Accessibility**. 2022. Disponível em: <https://www.w3.org/WAI/fundamentals/accessibility-intro/#what>. Acesso em: 17 out. 2022.

WEB PARA TODOS. **Número de sites brasileiros aprovados em todos os testes de acessibilidade tem queda em relação ao ano passado e é ainda menor que 1%**. 13 jun. 2022. Disponível em: <https://mwpt.com.br/numero-de-sites-brasileiros-aprovados-em-todos-os-testes-de-acessibilidade-tem-queda-em-relacao-ao-ano-passado-e-e-ainda-menor-que-1/>. Acesso em: 17 out. 2022.

WHO IS. **Armazendocampo.shop - Who is information**. [s.d.]. Disponível em: <https://who.is/whois/armazendocampo.shop>. Acesso em: 5 fev. 2024.