

CENTRO UNIVERSITÁRIO SERRA DOS ÓRGÃOS – UNIFESO
CENTRO DE CIÊNCIAS E TECNOLOGIA – CCT
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

TESTES E2E DE ACESSIBILIDADE EM APLICAÇÕES WEB

Lucas da Silva Cruz

Teresópolis/ RJ
Maio, 2021

CENTRO UNIVERSITÁRIO SERRA DOS ÓRGÃOS – UNIFESO
CENTRO DE CIÊNCIAS E TECNOLOGIA – CCT
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

TESTES E2E DE ACESSIBILIDADE EM APLICAÇÕES WEB

Lucas da Silva Cruz

Trabalho de Conclusão de Curso apresentado ao Centro
Universitário Serra dos Órgãos como requisito
obrigatório para obtenção do título de Bacharel em
Ciência da Computação.
Orientador(a): Tiago Resende

Teresópolis/ RJ
Maio, 2021

CENTRO UNIVERSITÁRIO SERRA DOS ÓRGÃOS – UNIFESO
CENTRO DE CIÊNCIAS E TECNOLOGIA – CCT
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

TESTES E2E DE ACESSIBILIDADE EM APLICAÇÕES WEB

Lucas da Silva Cruz

Trabalho de Conclusão de Curso aprovado como requisito parcial para obtenção do título de Bacharel em Ciência da Computação no Centro Universitário Serra dos Órgãos pela banca examinadora:

Nome do(a) Orientador(a) por extenso - titulação abreviada

Nome do(a) Convidado(a) por extenso - titulação abreviada

Nome do(a) Convidado(a) por extenso - titulação abreviada

Teresópolis/ RJ
Maio, 2021

“A criatividade pode ser uma contribuição social, mas apenas na medida em que a sociedade é livre para usar os resultados.”

Richard Stallman

*Aos meus pais, que me deram todas as oportunidades e
incentivos para estudar que eles próprios não tiveram.*

AGRADECIMENTOS

Antes de mais nada quero agradecer principalmente a minha família por todo apoio, participação e incentivo durante todo meu processo de formação. À minha companheira de longo prazo, Sarah de Almeida Ferreira, por sempre me ajudar e socorrer nos momentos mais difíceis da minha vida. Agradeço a todos os meus professores e professoras desde a escola à graduação, ao Centro Universitário Serra dos Órgãos (Unifeso) por todas as oportunidades, experiências e pelo ambiente acolhedor que se tornou a minha segunda casa. Um agradecimento especial ao meu orientador Tiago Resende pela sua impecável disponibilidade e paciência na execução de todo este trabalho.

A todos que, direta ou indiretamente, colaboraram para a realização deste trabalho.

RESUMO

O presente trabalho discorre sobre a marginalização informacional que a pessoa portadora de alguma deficiência enfrenta na internet e a importância dos testes manuais e automatizados de ponta à ponta para garantir requisitos fundamentais recomendados pelas Diretrizes de Acessibilidade para o Conteúdo da Web (WCAG). O estudo faz um levantamento das principais tecnologias e stacks envolvidas na automatização desses testes e argumenta sobre a importância de integrá-los à suíte de testes que normalmente é construída durante o processo de desenvolvimento de uma aplicação web, aplicando os procedimentos levantados na pesquisa ao estudo de caso do próprio Ambiente Virtual de Aprendizagem (AVA) do Centro Universitário Serra dos Órgãos (Unifeso), considerando que os métodos e resultados desenvolvidos e documentados nesta avaliação possam ser reutilizados em plataformas do mesmo tipo, que se popularizaram devido à mudança de rotina decorrente do confinamento – principal meio de controle do coronavírus.

Palavras-chave: Acessibilidade, Testes, WCAG, Exclusão Digital, TICs.

ABSTRACT

The present work discusses the informational marginalization that a person with a disability have to deal on internet and the importance of manual and automated end-to-end tests to guarantee fundamental requirements recommended by the Web Content Accessibility Guidelines (WCAG). This study researches the main technologies and stacks involved in automating tests and discuss about the importance of integrating them into the test suite that is normally built during the process of developing a web application, applying the researched procedures to a study of case of the Virtual Learning Environment (VLE) of the Serra dos Órgãos University Center (Unifeso), considering that the methods and results developed and documented in this acessibility check can be reused in platforms of the same type, which became popular due to change in routine resulting from confinement – the main way of controlling coronavirus.

Keywords: Accessibility, Tests, WCAG, Digital Exclusion, ICTs.

LISTA DE ILUSTRAÇÕES

| | Página |
|--------------------------------------|--------|
| Figura 1 – [digite o nome aqui]..... | 14 |
| Figura 2 – [digite o nome aqui]..... | 15 |
| Figura 3 – [digite o nome aqui]..... | 15 |

LISTA DE ABREVIATURAS E SIGLAS

CAPTCHA: Completely Automated Public Turing Tests to tell Computers and Humans Apart

API: Application Programming Interface

OCR: Optical Character Recognition

SUMÁRIO

| | Página |
|--|-----------|
| 1 INTRODUÇÃO..... | 13 |
| 1.1 OBJETIVO GERAL..... | 13 |
| 1.2 HIPÓTESES..... | 13 |
| 1.3 OBJETIVOS ESPECÍFICOS..... | 13 |
| 2 FUNDAMENTAÇÃO TEÓRICA..... | 13 |
| 2.1 TESTES MANUAIS E AUTOMATIZADOS..... | 13 |
| 2.2 TESTES AUTOMATIZADOS DE ACESSIBILIDADE..... | 13 |
| 2.2.1 TESTES DE ACEITAÇÃO..... | 13 |
| 2.2.2 TESTES FUNCIONAIS..... | 13 |
| 2.2.3 TESTES UNITÁRIOS..... | 13 |
| 2.3 TESTES MANUAIS DE ACESSIBILIDADE..... | 13 |
| 2.3.1 FOCO E TAB ORDER..... | 13 |
| 2.3.2 LEITORES DE TELA..... | 13 |
| 2.4 FERRAMENTAS AUTOMATIZADAS PARA TESTES DE ACESSIBILIDADE..... | 13 |
| 2.4.1 aXe..... | 13 |
| 2.4.2 Pa1ly..... | 13 |
| 2.4.3 Lighthouse..... | 13 |
| 3 METODOLOGIA..... | 13 |
| 3.1 PESQUISA EXPLORATÓRIA..... | 13 |
| 3.2 ESCOPO E PÁGINAS DA AUDITORIA..... | 13 |
| 3.3 TESTES AUTOMATIZADOS..... | 13 |
| 3.4 TESTES MANUAIS..... | 13 |
| 4 RESULTADOS ESPERADOS..... | 13 |
| 4.1 ESTATÍSTICAS..... | 13 |
| 4.2 SUGESTÕES DE MELHORIA APURADAS ATÉ O MOMENTO..... | 13 |
| 5 CONCLUSÃO..... | 13 |
| 6 TRABALHOS FUTUROS..... | 13 |
| 7 REVISÃO BIBLIOGRÁFICA..... | 13 |
| 8 REFERÊNCIAS..... | 13 |

1. INTRODUÇÃO

Vivemos em um tempo de explosão informacional onde as Tecnologias de Informação e Comunicação (TICs) se tornaram elementos fundamentais e determinantes para a criação de práticas mais democráticas. Nesse contexto contemporâneo, o direito à comunicação e à informação, bem como a democratização das TICs, são fundamentais [MORIGI, V. J., et al. 2016].

Por outro lado, quando o acesso às tecnologias não se dá de forma uniforme é criado um fenômeno de "marginalização informacional", como afirma Mattelart (2002). Para João Pissarra Esteves (2011), essa exclusão social compreende uma nova versão do “digital divide”, que evidencia como as diferenças de acesso refletem as reais desigualdades sociais, políticas e econômicas entre a população incluída e excluída digitalmente.

Apesar do tema Inclusão Social e Digital vir sendo objeto de debates no meio acadêmico, governamental e empresarial, a questão da inclusão de pessoas portadoras de necessidades especiais, em todos os recursos da sociedade, ainda é muito incipiente no Brasil. [MACIEL, M. R. C., et al. 2010]. Uma das formas de exclusão digital está associada justamente à falta de acessibilidade nos serviços e informações da Web, embora ela tenha sido projetada para ser utilizada por qualquer pessoa. As Diretrizes de Acessibilidade para Conteúdo Web (WCAG) visam fornecer recomendações que tornem o conteúdo acessível a um maior número de pessoas com deficiência, que incluem:

“acomodações para cegueira e baixa visão, surdez e baixa audição, limitações de movimentos, incapacidade de fala, fotossensibilidade e combinações destas características, e alguma acomodação para dificuldades de aprendizagem e limitações cognitivas; mas não abordará todas as necessidades de usuários com essas deficiências. Seu conteúdo da Web também ficará mais acessível aos usuários em geral ao seguir estas diretrizes.”
(WCAG 2.1, 2018, Abstract)

Nos dias atuais, é uma prática intrínseca ao desenvolvimento de um Sistema Web garantir a sua usabilidade, que pode ser definida como o fator que assegura que um produto ou serviço seja fácil de usar, eficiente e agradável a partir do ponto de vista do usuário [PREECE; ROGERS; SHARP, 2007]. Porém, em relação a acessibilidade, que considera a diversidade de seus possíveis usuários e as peculiaridades da interação dessas pessoas com o produto [TORRES; MAZZONI, 2004], muitas das vezes os desenvolvedores e especialistas

em Garantia de Qualidade (QA) não consideram as diretrizes e padrões da WCAG no design e na test suite (conjunto de casos de teste) das aplicações, gerando uma série de problemas para o usuário com alguma deficiência.

Por causa disso surgiram as engines como Axe e Pa1ly, que implementam as regras da WCAG e rodam testes end-to-end (e2e) de acessibilidade nas páginas de uma aplicação. Elas foram arquitetadas e desenvolvidas para integrar com qualquer ambiente de teste já existente em um projeto, de forma que as organizações possam automatizar os testes de acessibilidade junto dos seus testes regulares.

Durante este estudo, foi realizado um levantamento das principais tecnologias e stacks utilizadas na construção de testes e2e de acessibilidade em aplicações web e como integrar essas stacks ao projeto, além das principais formas de se testar a acessibilidade manualmente e a resolução dos principais problemas de acessibilidade para resolver os asserts.

1.1. OBJETIVO GERAL

Os objetivos gerais desta monografia são conscientizar os desenvolvedores de aplicações web e especialistas em QAs da importância de se validar a acessibilidade do sistema através de testes automatizados e manuais; explicitar as principais ferramentas utilizadas para se fazer testes e2e de acessibilidade em aplicações web e como integrá-las a um sistema; além de fazer um estudo de caso da verificação de acessibilidade do próprio Ambiente Virtual de Aprendizagem (AVA) do Centro Universitário Serra dos Órgãos (AVA).

1.2. HIPÓTESES

1.2.1. Quando o acesso às tecnologias não se dá de forma uniforme é criado um fenômeno de marginalização informacional.

1.2.2. Testes de acessibilidade não são considerados pelas empresas que desenvolvem software com a mesma importância de outros tipos de testes automatizados, como de carga e estresse, integração ou de performance.

1.2.3. O confinamento como medida de contenção do coronavírus intensificou o problema de marginalização informacional da pessoa portadora de

deficiência uma vez que ferramentas digitais passaram a ser utilizadas com muito mais frequência e muitas das vezes de forma obrigatória.

1.3. OBJETIVOS ESPECÍFICOS

Os objetivos específicos do trabalho são apresentar os testes automatizados e diferenciar as principais abordagens de automatização; realizar uma análise e descrição dos diferentes tipos de testes automatizados; discorrer sobre a importância dos testes e para automação de testes de acessibilidade, fazer um levantamento das principais técnicas e ferramentas para fazer testes e de acessibilidade; avaliar comparativamente as principais tecnologias usadas no desenvolvimento desses testes e de acessibilidade; e, por fim, exemplificar alguns dos principais problemas de acessibilidade e mostrar como adaptar o sistema de acordo com as especificações da WCAG para resolver os asserts dos testes automatizados.

2. FUNDAMENTAÇÃO TEÓRICA

[escreva aqui]

3. METODOLOGIA

3.1. PESQUISA EXPLORATÓRIA

Por se tratar de um tema ainda pouco pesquisado, foi realizada uma pesquisa exploratória com o intuito de levantar procedimentos a serem adotados em estudos de caso como o deste trabalho. Além disso, também foram pesquisadas as principais diretrizes e organizações de padronização que desenvolvem os pilares de tecnologias para se testar acessibilidade de modo que antes de apurar as possíveis soluções sejam identificadas da melhor maneira possível as falhas de acessibilidade.

3.2. ESCOPO E PÁGINAS DA AUDITORIA

Foram analisadas 89 páginas do AVA (Ambiente Virtual de Aprendizagem) do Unifeso (Centro Universitário Serra dos Órgãos) a partir de um login com perfil de estudante. 178 testes automatizados foram executados – 1 por página – e encontradas, em um primeiro momento, um total de 600 falhas de acessibilidade. Ainda é preciso analisar as páginas com privilégio administrativo, disponíveis apenas no login de professores.

Em segunda instância, verificou-se que algumas páginas estavam sendo afetadas pelos mesmos componentes, o que causou duplicidades na análise. A pesquisa seguiu identificando em cada resultado por página, manualmente, os problemas de acessibilidade que já haviam sido acusados pelos testes em páginas anteriores. Foram consideradas duplicatas as análises sobre os mesmos componentes em duas páginas diferentes, que acusavam as mesmas vulnerabilidades. O mesmo tipo de falha de acessibilidade, encontrado em elementos distintos, foi analisado separadamente considerando que apesar de ter a mesma solução ela deve ser aplicada em contextos diferentes, dependendo do elemento. Nessa fase, o total de falhas de acessibilidade passou a ser de 148 vulnerabilidades.

Durante a análise de duplicidade, os resultados também mostraram que a estrutura das páginas e seções é que continham os problemas de acessibilidade e não os dados ou valores taggeados. Com isso, foi revelado também que o mesmo tipo de página ou

seção apresentavam os mesmos problemas de falha de acessibilidade ainda que fizesse parte de uma disciplina diferente. Por exemplo, a disciplina “2021/1 – DESENVOLVIMENTO DE APLICAÇÕES MÓVEIS” possuía as mesmas falhas de acessibilidade na seção Conteúdo das Aulas que a disciplina “2021/1 – COMPUTAÇÃO GRÁFICA E PROCESSAMENTO DE IMAGENS” na mesma seção. Ou seja, apesar dos testes automatizados terem sido executados em cima das disciplinas ofertadas para o 6º/7º período flex. das turmas semestrais de Ciência da Computação em 2021, os resultados cobrem todas as disciplinas do curso que apresentam as mesmas seções.

3.3. TESTES AUTOMATIZADOS

Os testes automatizados foram realizados usando o Google Lighthouse, uma ferramenta automatizada de código aberto que utiliza a biblioteca axe-core para fornecer um conjunto de testes de acessibilidade. Ela roda todas as regras marcadas com os tipos wcag2a e wcag2aa, apesar de desabilitar alguns itens específicos.

```
22 function runAllyChecks() {
23   // @ts-ignore axe defined by axeLibSource
24   return window.axe.run(document, {
25     elementRef: true,
26     runOnly: {
27       type: 'tag',
28       values: [
29         'wcag2a',
30         'wcag2aa',
31       ],
32     },
33     resultTypes: ['violations', 'inapplicable'],
34     rules: {
35       'tabindex': {enabled: true},
36       'accesskeys': {enabled: true},
37       'table-fake-caption': {enabled: false},
38       'td-has-header': {enabled: false},
39       'marquee': {enabled: false},
40       'area-alt': {enabled: false},
41       'blink': {enabled: false},
42       'server-side-image-map': {enabled: false},
43     },
44     // @ts-ignore
45   }).then(axeResult => {
46     // Augment the node objects with outerHTML snippet & custom path string
47     // @ts-ignore
48     axeResult.violations.forEach(v => v.nodes.forEach(node => {
```

Figura 1: Código retirado do repositório do Lighthouse no Github. [Disponível aqui](#).

O Lighthouse visa oferecer uma auditoria abrangente de todos os aspectos de Qualidade de um app da Web. Ela foi adicionada ao Chrome DevTools a partir da versão 60 do Google Chrome e também pode ser usada como uma ferramenta de linha de comando. Atualmente, o Lighthouse cobre, além dos testes de acessibilidade, testes

de desempenho, progressividade, boas práticas e SEO. A ferramenta também pode fazer testes para desktop ou mobile, no último caso usando emuladores de celular durante a auditoria.

3.4. TESTES MANUAIS

Ainda é preciso analisar o ambiente com testes manuais para verificar os aspectos que os automatizados não conseguem identificar, como [discorra sobre o exemplo1 aqui] e [discorra sobre o exemplo2 aqui].

4. RESULTADOS ESPERADOS

A pesquisa deverá levantar, a partir dos dados gerados pelos testes automatizados, estatísticas que ajudem a identificar os problemas chaves de acessibilidade, mais frequentes e como estão distribuídos, que se relacionem com os fatores de pontuação e classificação de acessibilidade provida pelo Google Lighthouse. Com as informações dessas estatísticas, até o momento estima-se 13 propostas de soluções para resolver as falhas de acessibilidade, que devem ser aplicadas contextualmente a cada elemento acusado pelo Google Lighthouse. As soluções quando implementadas impactariam consideravelmente a pontuação de acessibilidade do AVA, é esperada uma mudança na classificação da plataforma de laranja para verde – considerada ideal pela ferramenta do Google.

4.1. ESTATÍSTICAS

Estatísticas até o momento.

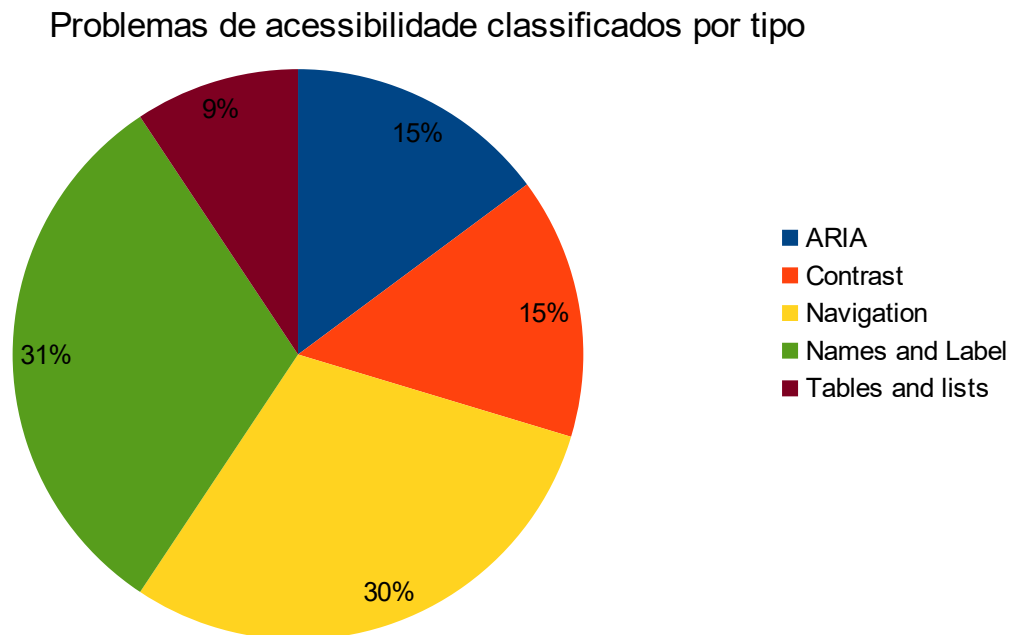


Figura 2: Flacando acul

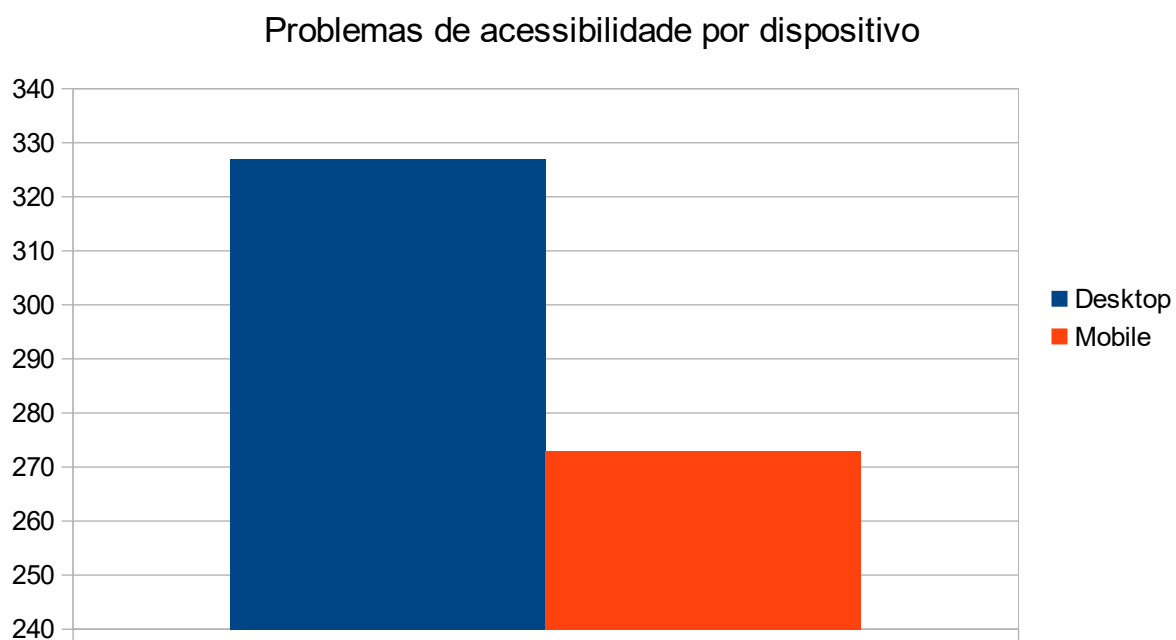


Figura 2: Ilustração geral

Demais gráficos ainda em apuração.

4.2. SUGESTÕES DE MELHORIA APURADAS ATÉ O MOMENTO

4.2.1. ARIA input fields do not have accessible names

A falha de acessibilidade número 1 diz respeito aos elementos que não têm um valor de ARIA role apropriado e por isso não podem ser anunciados adequadamente aos usuários que utilizam leitores de tela. O Lighthouse tem algumas auditorias que cobrem um conjunto diferente de funções ARIA, dentre elas o conjunto conhecido como aria-input-field-name que cuida das roles combobox, listbox, searchbox, slider, spinbutton e textbox. Essa auditoria que, por exemplo, faz com que a div abaixo falhe no teste de acessibilidade.

```
<div class="carousel-inner" role="listbox">
```

O problema pode ser resolvido, conforme listado na seção 2.1 da Accessible Rich Internet Applications (WAI-ARIA) 1.1, subtópico listbox (role), adicionando o atributo aria-label ao elemento, que permite os leitores de tela e outras tecnologias assistivas anunciar seu valor para o usuário. Dessa forma, o problema acima poderia ser resolvido refatorando o exemplo da seguinte maneira:

```
<div class="carousel-inner" role="listbox" aria-label="Texto descritivo aqui">
```

4.2.2. Elements with an ARIA [role] that require children to contain a specific [role] are missing some or all of those required children.

A falha de acessibilidade número 2 acontece quando uma ARIA role é atribuída a um elemento com propósito de dizer aos leitores de tela e outras tecnologias assistivas qual o comportamento e os controles customizados que um componente da aplicação tem. Algumas dessas roles exigem que os filhos do elemento também tenham roles específicas que trabalham em conjunto com a do pai. Por exemplo, a role tablist exige que os filhos tenham a role tab.

No exemplo abaixo, podemos ver que a div acusada com falha de acessibilidade possui uma role listbox, que por sua vez está associada as roles options.

```
▼<main id="moodle-page" class="clearfix">
  ▼<div id="page-header" class="clearfix ">
    ▼<div id="snap-site-carousel" class="carousel slide" data-ride="carousel">
      ▼<div class="carousel-inner" role="listbox"> == $0
        ▶<div class="carousel-item carousel-slide_one active">...</div>
        ▶<div class="carousel-item carousel-slide_two ">...</div>
        ▶<div class="carousel-item carousel-slide_three ">...</div>
        ::after
      </div>
      ▶<div class="carousel-controls">...</div>
      ▶<ol class="carousel-indicators js-only">...
    </ol> flex
  </div>
```

Figura 4: [Legenda aqui]

A solução para este tipo de falha de acessibilidade também é encontrada na seção WAI-ARIA 1.1, no subtópico option (role), que afirma a necessidade de adicionar aos elementos filhos de uma div com a role listbox o atributo role com valor option. Caso contrário, a especificação adverte que os elementos não serão corretamente mapeados pela API de acessibilidade. Sendo assim, o problema poderia ser resolvido refatorando o *case* acima da seguinte maneira:

```
▼<main id="moodle-page" class="clearfix">
  ▼<div id="page-header" class="clearfix ">
    ▼<div id="snap-site-carousel" class="carousel slide" data-ride="carousel">
      ▼<div class="carousel-inner" role="listbox"> == $0
        ▶<div class="carousel-item carousel-slide_one active" role="option">...</div>
        ▶<div class="carousel-item carousel-slide_two " role="option">...</div>
        ▶<div class="carousel-item carousel-slide_three " role="option">...</div>
        ::after
      </div>
```

Figura 5: [Legenda aqui]

4.2.3. Background and foreground colors do not have a sufficient contrast ratio.

Textos que não têm contraste o suficiente, além de afetarem principalmente usuários com baixa visão, também dificultam a leitura de todos os tipos de usuários. Isso pode ser notado, por exemplo, ao tentar ler algo no celular a partir de um ambiente externo e iluminado. Para resolver isso os critérios de sucesso mínimos (Nível AA) da WCAG 2.1 exigem uma taxa de contraste de pelo menos 3:1 para textos grandes – também definidos pelas diretrizes como textos maiores que 18pt se não estiverem em negrito e 14pt se estiverem. Para os demais tamanhos de texto a taxa de contraste é de 4.5:1.

Na falha de acessibilidade número 3 o teste automatizado identificou, nos elementos apontados por ele, que a taxa de contraste é menor do que as taxas exigidas pela WCAG. Para resolver o problema, é recomendada a utilização de alguma ferramenta que mensure e redefina o contraste de um texto. Atualmente, existem várias ferramentas criadas com esse propósito, como a Chrome DevTools' Color Picker do próprio Google, a WCAG Color Contrast Checker e a Contrast Grid.

A ferramenta utilizada na pesquisa foi a Chrome DevTools' Color Picker pela integração com o Google Chrome, fazendo um link entre as análises automatizadas do Lighthouse e facilitando a identificação e apuração do elemento com falha na acessibilidade. Ela pode ser habilitada nas configurações do DevTools, na seção Experiments, como ilustrado mais abaixo na figura X.

Para utilização da ferramenta, depois de habilitada, basta inspecionar o elemento que terá a taxa de contraste apurada, identificar o valor de sua cor na guia Styles do DevTools e clicar na *thumbnail* à esquerda do valor. À direita do agrupamento Contrast ratio, poderá ser visualizada a taxa de contraste do elemento. Abaixo, o contraste sugerido pela WCAG no nível AA e no nível AAA. Este procedimento está ilustrado adiante na figura Y.

Settings

Preferences

Workspace

Experiments

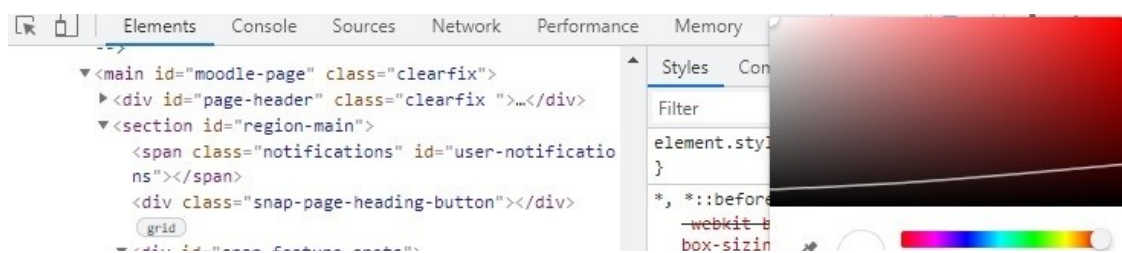
Ignore List

Devices

Throttling

Experiments

- ☐ Show option to take heap snapshot where globals are not treated as root
- ☐ Source diff
- ☐ Source order viewer
- ☐ Recorder
- ☐ Timeline: event initiators



4.2.4. [id] attributes on active, focusable elements are not unique

O atributo id no HTML, além de ser utilizado para identificar o elemento em scripts e no CSS, também responde a leitores de tela e outras tecnologias assistivas. Essas tecnologias, por esperar que o id seja único por todo o documento, acabam anunciando somente o primeiro elemento que compartilha o id em casos onde o mesmo id é definido em mais de um elemento, como informado na Técnicas para os Critérios de Sucesso 4.1.1 da WCAG 2.0. Dessa forma, os ids duplicados tornam apenas o primeiro elemento focalizável nas navegações por tab e shift + tab, fazendo com que o usuário não consiga acessar todas as funcionalidades da página.

A falha de acessibilidade número 4 indica que há mais de um elemento compartilhando o mesmo id, como na acusação de id redundante no exemplo abaixo.

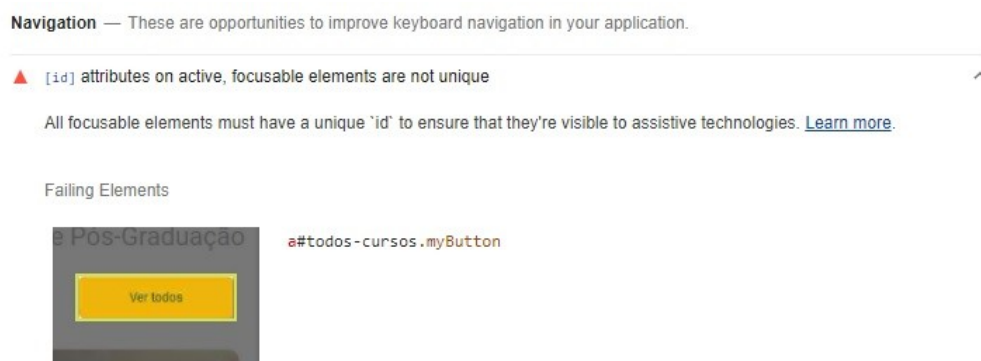


Figura 8: [Legenda aqui]

Para resolver esse problema é preciso identificar quais elementos têm o mesmo id e modificá-los para que cada um tenha seu próprio id. Uma alternativa é cogitar retirar o id dos elementos e manipulá-los por suas classes.

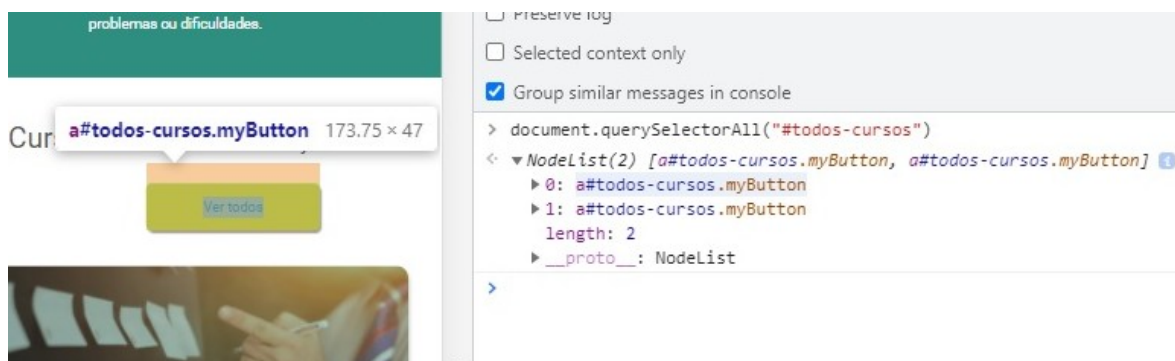


Figura 9: [Legenda aqui]

4.2.5. Heading elements are not in a sequentially-descending order

Os chamados headings no HTML são definidos com as tags de <h1> até <h6> e representam os seis níveis de título de seção, assim especificado na seção 4.3.9 das recomendações do W3C para o HTML 5.2. Um erro muito comum é utilizar esses elementos para marcar slogans, títulos alternativos e subtítulos que não pretendem representar semanticamente o título de uma nova seção, além da utilização dos seus níveis para diminuir ou aumentar o tamanho da fonte do cabeçalho. Esses enganos costumam levar a ordens não-sequências e descendentes – ideal para que leitores de tela consigam navegar de heading a heading.



Figura 10: Análise da acessibilidade dos headings da página inicial pelo Web Developer (Próprio Autor)

Na figura acima, através da extensão Web Developer do Google Chrome, podemos visualizar que os elementos acusados pelo teste automatizado estão todos no nível h4, apesar dos headings anteriores estarem no nível h2. Para solucionar o problema, os headings devem ficar ordenados decrescente e sequencialmente, mudando os h4s da div com id “redius” para h3s e removendo o gap na sequência dos headings.

4.2.6. Form elements do not have associated labels

A falha de acessibilidade número 6 [...]

As demais sugestões ainda estão sendo apuradas.

5. CONCLUSÃO

Na métrica de desempenho e boa experiência de usuário com a acessibilidade de um site ou sistema da web, provida pelo Google Lighthouse, a classificação do AVA da Unifeso está em laranja – pontuação de 50 a 89 – que define o nível de acessibilidade do ambiente como “precisa de melhoria”. O Lighthouse também recomenda a coloração verde, que classifica o sistema com “bom nível de acessibilidade”, para uma experiência de usuário satisfatória e esperada da Web – projetada para ser utilizada por qualquer pessoa e que fornece diretrizes de acessibilidade que tornam o conteúdo na internet acessível a um maior número de pessoas com deficiência quando implementadas.

Este trabalho foi um estudo de caso com o intuito de tornar o AVA da Unifeso mais acessível, identificando suas vulnerabilidades e propondo melhorias nos pontos-chaves que mais afetam as páginas do sistema. Foi realizado um levantamento das principais tecnologias e stacks envolvidas na realização de testes automatizados de acessibilidade, além dos métodos e das principais abordagens utilizadas para testar a acessibilidade manualmente – complementando a análise com aspectos que somente os testes automatizados não conseguem identificar. A principal ferramenta escolhida para fazer os testes automatizados foi o próprio Lighthouse, que recentemente incorporou o axe-core como uma de suas bibliotecas e atualmente está integrado ao Chrome DevTools e disponível no Google Chrome a partir da versão 60.

Além das estatísticas que a pesquisa levantou, que ajudam a identificar os problemas-chaves de acessibilidade, mais frequentes e como estão distribuídos, também foram encontradas 13 propostas de soluções que impactariam consideravelmente na pontuação de acessibilidade do AVA, inserindo a plataforma na classificação verde – considerada ideal pelo Google Lighthouse.

6. TRABALHOS FUTUROS

1.1.1. Checagem de SEO com Lighthouse e impactos da acessibilidade no ranqueamento do Google: Estatísticas do antes e depois da melhoria de acessibilidade do AVA da Feso

Com as estatísticas de acessibilidade do AVA levantadas e documentadas antes das propostas de melhorias sugeridas nesse trabalho terem sido implementadas, sabendo-se que o Google Lighthouse também faz auditoria de SEO, uma nova análise automatizada após a realização dessas melhorias demonstraria a relação do nível de acessibilidade com o *ranqueamento* das páginas pelo Google.

7. REVISÃO BIBLIOGRÁFICA

Não foram encontrados na literatura estudos que correlacionassem testes automatizados à verificação de acessibilidade segundo as Diretrizes de Acessibilidade para o Conteúdo da Web (WCAG) especificamente no campo do desenvolvimento de Aplicações Web.

Apesar disso, alguns estudos analisam as potencialidades das Tecnologias de Informação e Comunicação (TICs) como ferramenta imprescindível à democracia. O próprio estudo d'O reencantamento do mundo e acesso à informação [MORIGI, V. J., et al. 2016] admite uma quebra do antigo modelo de comunicação, fundamentalmente marcado pela transmissão da informação pelo denominado centro (emissor) à periferia (receptor). Com as TICs, é possível uma melhor distribuição dos emissores, uma vez que a nova forma de interação social democratiza a produção da informação por diversos agentes da informação.

Por outro lado, apesar do potencial democrático da internet, estudos como A História da sociedade da informação de Mattelart (2002) evidencia um fenômeno conhecido como “marginalização informacional”, quando o acesso às tecnologias não se dá de forma uniforme. Isso pode ser relacionado ao trabalho de João Pissarra Esteves (2011), no seu estudo Novos media e deliberação: sobre redes, tecnologia, informação e comunicação, que associa essa exclusão social a um novo tipo de digital divide para além da pobreza, compreendendo também usuários que apesar de se conectarem à internet não têm acesso às informações.

Podemos ver no estudo Portadores de deficiência: a questão da inclusão social [MACIEL, M. R. C., et al. 2010] que apesar do tema Inclusão Social e Digital vir sendo objeto de debates no meio acadêmico, governamental e empresarial, a questão da inclusão de pessoas portadoras de necessidades especiais, em todos os recursos da sociedade, ainda é muito incipiente no Brasil. Porém, a garantia da qualidade de sistemas na Web é um tema cada vez mais discutido, além da usabilidade e outras peculiaridades de interação das pessoas com o produto digital. Essa visão, baseada na perspectiva e necessidades dos usuários, abrem margem para integração e normalização das verificações de acessibilidade nos testes automatizados, que também fazem parte da qualidade de um produto e além de melhorariam a experiência dos usuários portadores de alguma deficiência também faz com que todos os tipos de usuário sejam beneficiados com as melhorias de acessibilidade, como enfatizado pelas Diretrizes de Acessibilidade para Conteúdo Web (WCAG) .

REFERÊNCIAS

MORIGI, V. J., et al. O reencantamento do mundo e acesso à informação: as potencialidades das Tecnologias de Informação e Comunicação (TICs) na construção e reforço da democracia Mídia, cidadania e utopia no Brasil. In: SOUSA, C. M., org. Um convite à utopia [online]. Campina Grande: EDUEPB, 2016. Um convite à utopia collection, vol. 1, pp. 303-339. ISBN: 978-85-7879-488-0. Available from: doi: 10.7476/9788578794880.0009. Also available in ePUB from: <http://books.scielo.org/id/kcdz2/epub/sousa-9788578794880.epub>.

MATELLART, A. História da sociedade da informação. São Paulo: Loyola, 2002.

ESTEVES, João Pissarra. Novos media e deliberação: sobre redes, tecnologia, informação e comunicação. Revista Media & Jornalismo, v. 18, n. 10, 2011.

ALONSO, Luiza Beth Nunes; FERNEDA, Edilson; SANTANA, Gislane Pereira. Inclusão digital e inclusão social: contribuições teóricas e metodológicas. Barbaroi, Santa Cruz do Sul, n. 32, p. 154-177, jun. 2010. Disponível em <http://pepsic.bvsalud.org/scielo.php?script=sci_arttext&pid=S0104-65782010000100010&lng=pt&nrm=iso>. acessos em 12 jan. 2021.

MACIEL, MARIA REGINA CAZZANIGA. Portadores de deficiência: a questão da inclusão social. São Paulo Perspec., São Paulo, v. 14, n. 2, p. 51-56, June 2000. Available from <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0102-88392000000200008&lng=en&nrm=iso>. access on 12 Jan. 2021. <https://doi.org/10.1590/S0102-88392000000200008>.

TORRES, Elisabeth Fátima; MAZZONI, Alberto Angel. Conteúdos digitais multimídia: o foco na usabilidade e acessibilidade. Ci. Inf., Brasília, v. 33, n. 2, p. 152-160, Aug. 2004. Available from <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-19652004000200016&lng=en&nrm=iso>. access on 04 Feb. 2021. <https://doi.org/10.1590/S0100-19652004000200016>.

W3C Working Draft. Web Content Accessibility Guidelines 2.1 (WCAG 2.1). B. Caldwell, W. Chisholm, G. Vanderheiden, J. White, eds. World Wide Web Consortium (MIT, ERCIM, Keio). Disponível em: <<https://www.w3.org/TR/WCAG21>>. Acesso em: 13 jan. 2021.

BERNARDO, P. C.; KON, F. A importância dos testes automatizados. Engenharia de Software Magazine, 3. ed., 2008

ALMEIDA, Luís Fernando Magnanini de et al. Avaliação do desempenho em agilidade na gestão de projetos. Prod., São Paulo, v. 26, n. 4, p. 757-770, dez. 2016. Disponível em <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-65132016000400757&lng=pt&nrm=iso>. acessos em 04 fev. 2021. Epub 10-Nov-2015. <https://doi.org/10.1590/0103-6513.116213>.

<https://www.w3.org/TR/wai-aria/#listbox>

<https://www.w3.org/TR/wai-aria/#option>

<https://www.w3.org/TR/WCAG21/#contrast-minimum>

<https://www.w3.org/TR/2016/NOTE-WCAG20-TECHS-20161007/H93>

<https://www.w3.org/TR/html52/sections.html#headings-and-sections>

FUNDAÇÃO EDUCACIONAL SERRA DOS ÓRGÃOS - FESO
CENTRO UNIVERSITÁRIO SERRA DOS ÓRGÃOS - UNIFESO
CENTRO DE CIÊNCIA E TECNOLOGIA - CCT
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

DECLARAÇÃO DE PRÓPRIA AUTORIA

Teresópolis, 20/04/2021

Eu, Lucas da Silva Cruz, declaro para fins de conclusão do Curso de Bacharelado em Ciência da Computação do UNIFESO, que este Trabalho de Conclusão de Curso é de minha própria autoria, estando ciente das consequências disciplinares a que estarei sujeito caso seja comprovada fraude ou má-fé.

Sem mais, subscrevo-me,

Atenciosamente,



Nome do discente