



**UNIVERSIDADE  
FEDERAL RURAL  
DE PERNAMBUCO**



Construção de *guidelines* para tornar um *e-commerce* acessível para pessoas com deficiência visual

---

**Relatório Técnico relativo ao Trabalho de Conclusão Curso  
do Bacharelado em Sistemas de Informação na modalidade Empresa**

---

**Aluna**

Stefany Vasconcelos de Sá

**Orientador**

Cleviton Vinicius Fonsêca Monteiro  
Departamento de Estatística de Informática

Outubro de 2022

Stefany Vasconcelos de Sá

## **Construção de *guidelines* para tornar um e-commerce acessível para pessoas com deficiência visual**

Relatório Técnico apresentado ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Universidade Federal Rural de Pernambuco – UFRPE

Departamento de Estatística e Informática

Curso de Bacharelado em Sistemas de Informação

**Orientador: Cleviton Vinicius Fonsêca Monteiro**

Recife

Outubro de 2022

## Resumo

O surgimento do *e-commerce* modificou a forma como as pessoas fazem compras e como consequência disso houve um aumento significativo no número de lojas *on-line*. Um estudo recente publicado pela Accenture mostrou que pessoas com deficiência possuem poder aquisitivo e sentem vontade de efetuar compras e atrelado às suas limitações as lojas *on-line* se mostram como sendo uma ótima opção para efetuar suas compras sem ter a necessidade de sair de casa. O problema surge quando essas lojas *on-line* não se mostram preparadas o suficiente para suprir as necessidades das pessoas com deficiências, especificamente deficiência visual, e como resultado desse despreparo ocorre a marginalização desse público-alvo, fato que fere os direitos básicos desses indivíduos. Utilizando os conhecimentos adquiridos por meio de um estudo focado sobre diretrizes Web Content Accessibility Guidelines (WCAG) e padrões de acessibilidade na internet foi possível fazer a construção de alguns artefatos que serviram como guias para tomadas de decisões e padrões de implementações. Neste *guideline* estão presentes técnicas de desenvolvimento de *software* e design inclusivo.

**Palavras-chave:** acessibilidade; e-commerce; style guide; diretrizes WCAG; atributos WAI-ARIA; gráficos acessíveis.

## Abstract

The emergence of e-commerce has changed the way people shop and as a result there has been a significant increase in the number of online stores. A recent study published by Accenture showed that people with disabilities have purchasing power and feel like shopping, and linked to their limitations, online stores are shown to be a great option to make their purchases without having to leave their homes. House. The problem arises when these online stores are not prepared enough to meet the needs of people with disabilities, specifically visual impairment, and as a result of this unpreparedness, this target audience is marginalized, a fact that violates the basic rights of these individuals. Using the knowledge acquired through a study Based on Web Content Accessibility Guidelines (WCAG) and internet accessibility standards, it was possible to build some artifacts that served as guides for decision-making and implementation standards. In this guideline are present techniques of software development and inclusive design.

**Keywords:** accessibility; e-commerce; styleguide; WCAG guidelines; WAI-ARIA attributes; accessible graphics.

# Sumário

<b>1</b>	<b>Introdução</b>	<b>6</b>
1.1	Contexto . . . . .	6
1.1.1	Acessibilidade <i>Web</i> . . . . .	6
1.2	Problema e solução proposta . . . . .	7
1.3	Justificativa . . . . .	8
<b>2</b>	<b>A empresa e sua atuação</b>	<b>8</b>
<b>3</b>	<b>Desenvolvimento realizado na empresa</b>	<b>8</b>
3.1	A problemática e a solução proposta . . . . .	8
3.2	Fase 1: Pesquisa e entendimento . . . . .	9
3.2.1	Guia de acessibilidade . . . . .	10
3.2.2	Style Guide . . . . .	12
3.3	Fase 2: Adaptação e construção da <i>interface</i> do usuário acessível e usável com base no guia de acessibilidade e <i>style guide</i> . . . . .	16
3.3.1	Abordagens inteligentes e adaptação de componentes . . . . .	16
3.3.2	Adaptação de componentes para o uso de leitores de telas e navegação por teclado . . . . .	20
3.4	Fase 3: Construção do dashboard acessível . . . . .	23
3.5	Fase 4: Testes e avaliações de acessibilidade . . . . .	26
3.6	Tecnologias utilizadas . . . . .	27
3.7	Contribuição . . . . .	28
<b>4</b>	<b>Dificuldades encontradas</b>	<b>28</b>
<b>5</b>	<b>Impactos da sua formação no seu trabalho</b>	<b>29</b>
<b>6</b>	<b>Conclusão</b>	<b>29</b>
	<b>Referências Bibliográficas</b>	<b>32</b>
<b>A</b>	<b>Diretrizes WCAG</b>	<b>33</b>
A.1	Nível A . . . . .	33
A.2	Nível AA . . . . .	34



# 1 Introdução

Com a pandemia da COVID-19 houve várias mudanças no cenário econômico brasileiro, lojas físicas tiveram que fechar as portas enquanto as lojas virtuais, mais conhecidas como *e-commerce*, tiveram um crescimento muito acentuado. Uma pesquisa feita pelo BigData Corp em parceria com o PayPal Brasil [1] afirma que no ano de 2021 houve um aumento de 22,05% no número de lojas *on-line*, o que marca cerca de 1,6 milhões de sites de *e-commerce*. Esse aumento de lojas *on-line* reflete o hábito de consumo dos indivíduos, que sempre buscam uma forma de realizar suas compras de forma ágil e descomplicada. Porém existe uma parcela da população que não conseguem efetuar compras em lojas *on-line*, pois tais lojas não estão preparadas para lidar com as necessidades desse público.

## 1.1 Contexto

De acordo com a Lei Brasileira de Inclusão Da Pessoa com Deficiência [2], considera-se pessoa com deficiência aquela que tem impedimento de longo prazo de natureza física, mental, intelectual ou sensorial, o qual, em interação com uma ou mais barreiras, pode obstruir sua participação plena e efetiva na sociedade em igualdade de condições com as demais pessoas.

Ainda no ano de 2019, foi realizada uma pesquisa para entender melhor um pouco sobre o indivíduo com deficiência e segundo a Pesquisa Nacional de Saúde [3] existiam quase 7 milhões de pessoas com deficiência visual no Brasil onde aproximadamente 4,2% eram pessoas com 18 anos ou mais. Para entender um pouco mais sobre esse nicho de consumidores a Accenture produziu um relatório [4] sobre Design Inclusivo [5], esse relatório mostra as vantagens de se aplicar o design inclusivo a um negócio. Os dados do relatório demonstraram que no Brasil as pessoas com deficiência possuem uma renda disponível de 5,3 milhões de dólares. Ou seja, essa parcela da população apresenta um poder aquisitivo disponível para efetuar compras, mas muitas empresas ainda não pensaram ou se preocuparam em criar alternativas para vender a estas pessoas.

Como as lojas *on-line* não estão preparadas para lidar com essa parcela da população, um grande público acaba sendo marginalizado. Esse ato acaba ferindo um dos direitos básicos conferido de acordo com a lei brasileira de inclusão da pessoa com deficiência [2], o que torna obrigatória a acessibilidade em ambientes virtuais. Porém, mesma havendo uma legislação, isso não foi suficiente para que houvesse uma mudança dentro do ambiente *Web*.

### 1.1.1 Acessibilidade Web

Acessibilidade “é a possibilidade de qualquer pessoa usufruir todos os benefícios da sociedade, inclusive o de usar a Internet” [6]. A acessibilidade na *Web* possibilita que pessoas com deficiência possam navegar livremente na *internet* sem que haja interrupções ou bloqueios. Porém, tornar esse conteúdo acessível pode ser benéfico para um outro grupo de usuários, que podem não ter deficiência mas utilizam a aplicação de um modo diferente, como no caso das pessoas que possuem limitações de visão.

Quando um sistema é considerado acessível não há barreiras que impeçam o acesso dos usuários. O presente trabalho foi focado na diminuição de barreiras de acesso a pessoas com deficiências e limitações visuais. As necessidades que uma pessoa com deficiência visual tem ao navegar pelos sites são semelhantes a de uma pessoa sem essa deficiência: se atualizar com as últimas notícias, se entreter através de jogos, realizar operações bancárias, fazer compras, etc. A acessibilidade é um dos fatores que compõem a qualidade de um site, por isso a preocupação de diversas instituições e governos em relação a este assunto [7].

O processo de tornar um site ou sistema acessível deixou de ser tema exclusivo do universo das pessoas com deficiência e começou a se tornar um tema recorrente. O engajamento das pessoas para diminuir as barreiras de acesso é o fator mais importante para iniciar o processo de acessibilização. Dado esse início é necessário muito empenho para promover avaliações, correções e melhorias. Vale ressaltar que dificilmente um site ou aplicação será 100% acessível, uma vez que existem muitos tipos diferentes de deficiências com peculiaridades a seu modo, porém a eliminação de barreiras é algo que pode ser tangível e mensurável.

## 1.2 Problema e solução proposta

Diante da problemática, onde deficientes visuais, mesmo tendo seus direitos garantidos por lei [2] e sendo consumidores em potencial, acabam sendo negligenciados por empresas que optam por estender seu nicho de mercado para o meio *on-line*. Este trabalho tem como o principal objetivo propor *guidelines* para diminuir as barreiras de acesso e transformar um *e-commerce* acessível para pessoas com deficiência visual. Para que esse objetivo maior fosse satisfeito foi necessário passar por algumas etapas de construção e estudos, que são elas:

- A sumarização do conhecimento adquirido após o entendimento de como um deficiente visual interage com softwares, seja por navegação via teclado ou com o auxílio de leitores de tela e a documentação desse conhecimento em um *Style Guide* [8] que auxilia as pessoas desenvolvedoras nas tomadas de decisões e padrões necessários para a construção de um interface acessível.
- Entendimento das diretrizes Web Content Accessibility Guidelines (WCAG) [9] para que houvesse um isolamento dessas diretrizes por prioridades e por fim a construção de uma documentação que contava com formas de aplicação eficiente e abordagens já difundidas pela internet.
- Entendimento do *e-commerce* para que fosse possível a aplicação das melhores técnicas de acessibilidade a fim de obter a adaptação de sistemas antigos e sem acessibilidade em sistemas acessíveis.
- Aplicação de técnicas de acessibilidade em elementos gráficos complexos para torná-los identificáveis via navegação por teclado e por leitores de tela.

Desse modo, o presente estudo apresenta um relato de abordagens eficientes aplicadas ao *e-commerce*, sendo baseadas em um guia de acessibilidade construído com o propósito de facilitar o entendimento de diretrizes WCAG e um *style guide* para nortear os desenvolvedores quanto a comportamentos esperados e que faziam sentido no contexto de limitações visuais.

## 1.3 Justificativa

Existe uma ideia equivocada de que pessoas com deficiência possuem apenas necessidades relacionadas exclusivamente a sua deficiência, quando na verdade essas pessoas necessitam de tudo o que uma pessoa sem deficiência necessita somado as particularidades atreladas às suas limitações. A acessibilidade *web* surge como um modificador de vida trazendo conforto e praticidade para os indivíduos. Porém esse conforto e praticidade não costuma ocorrer tão facilmente. Mesmo as pessoas com deficiência tendo voz, poder e direitos dentro da sociedade eles passam a ser escanteados pelas potenciais empresas que criam lojas *on-line*.

Por mais que a internet esteja difundida nos dias de hoje, as pessoas com deficiência, consumidores em potencial, acabam por serem esquecidos e negligenciados. Logo esse trabalho se faz relevante por apresentar opções para assegurar os direitos do público-alvo em questão e demonstrar a importância da construção de *e-commerce* acessíveis, pois existe um nicho de mercado que está sendo deixado de lado e poderia ser uma vantagem competitiva às empresas que adotassem as técnicas de acessibilidade em seus sites de *e-commerce*.

## 2 A empresa e sua atuação

A empresa tem sede no Recife e é de grande relevância no ecossistema do Porto Digital. Atualmente ela conta com mais de 20 anos no mercado e sua área de atuação está em todo o ciclo de inovação, desde pesquisa, formação de pessoas, experimentação e criação de novos modelos de negócios, até o desenvolvimento de soluções robustas e escaláveis. Muitas empresas dos mais diversos setores buscam parcerias para desenvolverem suas soluções juntamente com a organização o que demonstra a alta relevância e o profissionalismo entregue por todos que a compõem.

A organização conta com um time diverso e multidisciplinar de mais de mil e trezentos colaboradores, incluindo designers, desenvolvedores, consultores, estrategistas, empreendedores, pesquisadores e educadores.

## 3 Desenvolvimento realizado na empresa

Esta seção apresenta a ferramenta desenvolvida como solução ao problema de acessibilidade em *e-commerce* da organização, abordando as técnicas e tecnologias utilizadas e contribuições para a empresa.

### 3.1 A problemática e a solução proposta

A problemática tratava-se de transformar um *e-commerce* legado sem práticas de acessibilidade em um *e-commerce* acessível e bem construído focando em dar suporte diminuir as barreiras de acesso que as pessoas com deficiências visuais se deparam ao acessar um sistema como esse. Como requisitos principais para a solução ser aceita se fazia necessário construir ou adaptar os

componentes incluindo práticas de acessibilidade, dar suporte a diversos tipos de navegações que o usuário pudesse escolher, construir um painel administrativo onde apresentasse um *dashboard* com gráficos acessíveis para o acompanhamento das vendas.

Para tratar essa problemática foi necessário dividi-la em pequenas fases e trabalhar isoladamente em cada uma delas, visto que essas fases dependiam uma da outra e elas foram desenvolvidas de forma cronológica partindo de pesquisas a testes.

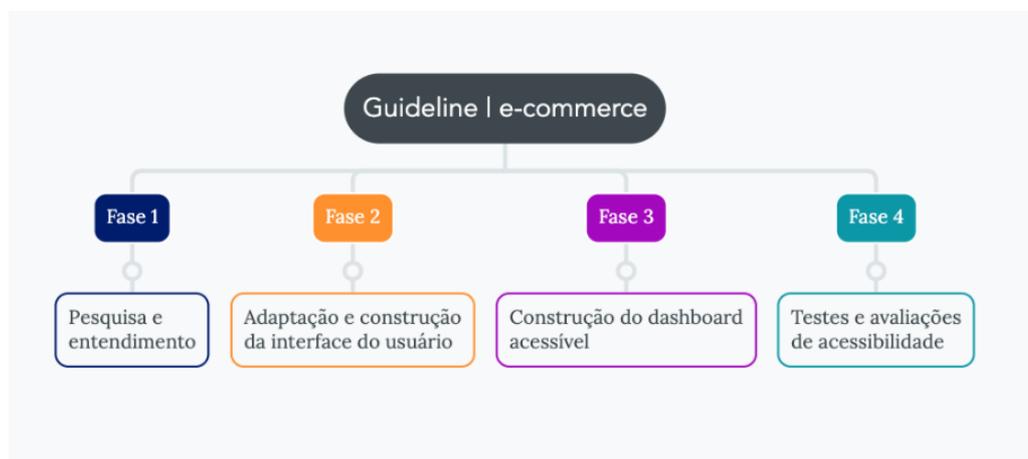


Figura 1: Fases do Guideline

Fonte: a autora (2022)

No total foram preciso quatro fases, como exemplificado na figura 1, sendo a primeira fase voltada para o entendimento do problema e do uso das novas tecnologias como também entender as limitações do projeto. A segunda fase estava voltada na construção e adaptação da interface do usuário seguindo tudo o que foi levantado na fase de entendimento do problema. A terceira fase teve foco na construção do painel administrativo e no *dashboard*, além do planejamento de componentes e na construção de gráficos acessíveis. Por fim a quarta fase, onde se concentrou os testes, foram construídas abordagens de testes para a certificação de que todos os requisitos estavam sendo satisfeitos e se as diretrizes estavam sendo seguidas.

### 3.2 Fase 1: Pesquisa e entendimento

Para começar a construção do *e-commerce* se fazia necessário entender melhor o problema que iria ser enfrentado e as tecnologias usadas a fim de entender as limitações do projeto em questão de performance. Com o entendimento das limitações do projeto os desenvolvedores teriam a visão da possibilidade ou não de adicionar novas bibliotecas ao decorrer do desenvolvimento caso fosse necessário. Na figura 2 é destrinchado o que compreende a fase 1 na construção do *guideline*.



Figura 2: Fase 1 do Guideline

Fonte: a autora (2022)

Após concluída a fase de estudo e exploração surgiu a necessidade da construção de um guia de acessibilidade que seria usado na fase de desenvolvimento do projeto, essa decisão se deu porque a documentação das diretrizes de acessibilidade se mostrou bastante extensa e englobava muitos tipos de deficiência, tornando o entendimento muito dificultoso para todos que estão lendo. Para complementar o guia de acessibilidade foi construído um *style guide* para guiar todos os desenvolvedores a respeito de aspectos importantes que devem estar presentes na *User Interface (UI)*

### 3.2.1 Guia de acessibilidade

Ao se tocar no quesito acessibilidade é necessário citar a *WCAG* que trata-se de um conjunto de diretrizes que estipula os padrões de acessibilidade digital que devem ser seguidos pelos desenvolvedores para a produção de aplicações *on-line*. Essas recomendações foram todas desenvolvidas pelo consórcio *World Wide Web (W3C)* [10], .

O *WCAG* é responsável por orientar de forma muito simples como identificar e implementar técnicas que eliminam barreiras de acesso para pessoas com deficiência e ele foi construído sob quatro princípios:

- **Perceptível:** As informações e os componentes da interface do usuário devem ser apresentados em formas que possam ser percebidas pelo usuário.
- **Operável:** Os componentes da interface e a navegação devem ser operáveis, ou seja o usuário não pode ter impedimentos para utilizar a interface.
- **Compreensível:** A informação e a operação da interface de usuário devem ser compreensíveis, elas devem fazer sentido e bem apresentadas na interface do usuário.
- **Robusto:** O conteúdo deve ser robusto o suficiente para poder ser interpretado de forma confiável por uma ampla variedade de agentes de usuário, incluindo tecnologias assistivas. A aplicação deve ser bem construída, de forma a ser acessível para uma gama maior de navegadores e tecnologias assistiva.

Guiado por esses quatro princípios surgiram as diretrizes, que são os itens que fornecem os objetivos básicos que devem ser atingidos dentro de cada princípio, que são tópicos menores e mais

específicos. Dentro das diretrizes existe critérios de sucesso, esse sucesso está atrelado a nível de conformidade. Então quanto mais conforme o *e-commerce* está adequado àquela diretriz maior é o critério de sucesso. Os critérios são definidos três níveis de conformidade:

- No nível A estão os critérios mais simples, que representam apenas barreiras mais significativas de acessibilidade. Ao obter sucesso nesse critério já é possível eliminar uma boa quantidade de barreiras de acesso. 3
- No nível AA, é um nível acima do A, o que significa que atingindo esse nível as barreiras do nível A foram somadas a mais barreiras e todas foram eliminadas. 4
- Nível AAA, geralmente são um refinamento das anteriores, sendo mais detalhadas e que trazem um nível mais sofisticado de acessibilidade. 5

Dentre as 78 diretrizes WCAG [9] que existem foram escolhidas algumas que fizessem sentido no contexto de diminuição de barreiras de acesso para deficientes visuais. Toda a construção do *e-commerce* foi baseada nas diretrizes listadas no Apêndice A:

### 3.2.2 Style Guide

O *style guide* [8] é um documento que concentra as diretrizes de design de um projeto para ajudar e alinhar todos os desenvolvedores. Com o *style guide* é possível manter a consistência visual dentro do projeto. Para a construção do *style guide* do *e-commerce* o foco se deu em três informações básicas:

#### Cores

Quando o foco é sobre as cores é preciso ter em mente contraste e luminosidade. Existe algumas deficiências e limitações visuais que fazem com que a cor sofra mudanças então é preciso que o *e-commerce* tenha uma paleta de cores acessível que garanta o contraste ideal entre as cores 4. No exemplo abaixo da figura 3 está sendo demonstrado uma paleta de cores inclusiva criada pela ferramenta *Adobe Color Wheel* [11].

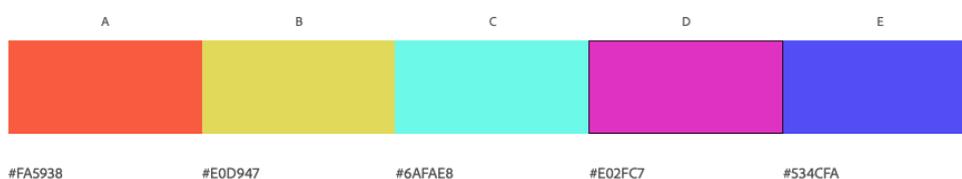


Figura 3: Paleta de cores acessíveis construída pelo site da *Adobe Color* [11].

Fonte: a autora (2022)

É de extrema importância verificar como ficariam essas cores pela visão de pessoas que possuem algum tipo de daltonismo. Ao perceber que a paleta de cores ainda está com um contraste aceitável entre o primeiro e o segundo plano de no mínimo 7:1 ela está pronta para o uso, seguindo a diretriz nível AAA de contraste 5. Na figura 4 é possível observar os diferentes tipos de daltonismo e como a paleta de cores se comporta diante das limitações de visão de pessoas com daltonismo. Essa paleta foi criada utilizando as ferramentas de acessibilidade do *Adobe Color Wheel* [11].



Figura 4: Comportamento de uma paleta de cores acessível diante dos tipos de daltonismo.

Fonte: a autora (2022)

## Tipografia

A tipografia não é apenas escolher uma fonte legal para o seu projeto, junto com ela devemos levar em consideração vários aspectos que são importantes e foram listados nas diretrizes:

- Espaçamento
  - O espaçamento deve ser bem planejado para que seja possível aplicar um zoom de 200% ou ainda em caso de redimensionamento de tela o texto deve permanecer legível. 4
- Altura de linhas
  - Sempre que houver um redimensionamento os textos não devem perder legibilidade. 4
- Hierarquia tipográfica
  - Sempre que o conteúdo da tela for dividido em sessões, todas devem possuir títulos claros, com níveis de hierarquia bem definidos, para assim manter a hierarquia da informação de uma forma concisa. 5
- Pesos, cores e contraste.
  - O contraste entre textos e telas também deve ser mantido num padrão de no mínimo 4:5:1 entre o primeiro e o segundo plano da tela. 3

A figura 5 foi criada usando os recursos de acessibilidade presentes na ferramenta *Adobe Color Wheel* [11]. A partir da cor do texto e da cor de fundo é possível saber a proporção de contraste e se está dentro do esperado nas diretrizes WCAG [12].



Figura 5: Checagem de contraste entre texto e fundo realizado por meio do site *Adobe Color Wheel* [11]

Fonte: a autora (2022)

## Elementos da *interface* do usuário

Este tópico compreende os elementos que estão presentes na *interface*, logo é apresentados exemplos e listagem desses componentes. Na construção desses elementos é feito o detalhamento visual, ou seja, os tamanhos, espaços, margens, cores que ele deve possuir como também os seus estados comportamentais, se houver um erro como ele se comporta, em caso de elemento focado como deve ficar o elemento.

Com o componente *Card* da figura 6 abaixo é possível analisar onde as diretrizes foram aplicadas e como elas impactam na construção de componentes mais complexos da *interface* do usuário.

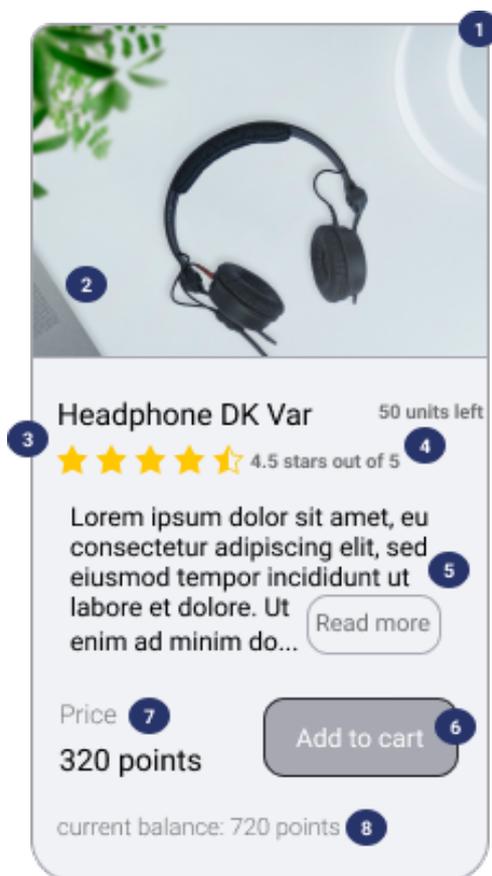


Figura 6: Exemplo de um componente montado com elementos da interface do usuário

Fonte: a autora (2022)

No quadro abaixo 1 foi referenciado os identificadores de 1 a 8 apontados na Figura 6 com as diretrizes aplicadas em cada ponto juntamente com sua breve descrição. Foi especificado o nível em que cada diretriz se encontra. Para maiores detalhes sobre as diretrizes consultar o apêndice A.

Quadro 1: Lista de identificadores de 1 a 8 e as diretrizes contempladas

Identificador	Diretriz	Nome	Nível	Descrição
1	1.3.1	Informações e Relações	A	As estruturas da tela devem ser construída de forma que sua arquitetura de informação faça sentido tanto para todos, sejam ouvintes ou leitores.
	1.3.2	Sequência com significado	A	A apresentação das informações na tela sempre deverá ter uma sequência lógica.
	2.4.7	Foco visível	A	Ao se interagir por teclado, qualquer pessoa deve conseguir identificar qual é a sua localização na tela através de um foco visível.
	3.2.1	Em foco	A	O foco sempre deve se manter durante a navegação, guiando o usuário e evitando mudanças que causem desorientação.
2	1.1.1	Conteúdo Não Textual	A	Todo o conteúdo não textual que é exibido ao usuário tem uma alternativa textual que serve a um propósito equivalente.
	1.4.11	Contraste Não textual	AA	apresentação visual a seguir tem um relação de contraste de pelo menos 3:1 contra cor(es) adjacente(s).
3	1.4.3	Contraste (mínimo)	AA	Textos devem ter uma relação de contraste entre primeiro e segundo plano de ao menos 4.5:1.
	2.4.6	Cabeçalhos e rótulos	AA	Todos os títulos e rótulo devem descrever claramente a finalidade dos conteúdos, não deve haver ambiguidade em seu entendimento.
4	1.4.6	Contraste (melhorado)	AAA	Textos devem ter uma relação de contraste entre primeiro e segundo plano de ao menos 7:1.
	1.4.11	Contraste Não textual	AA	A apresentação visual a seguir tem um relação de contraste de pelo menos 3:1 contra cor(es) adjacente(s).
5	1.4.12	Espaçamento de texto	AA	Sempre que houver um redimensionamento os textos não devem perder legibilidade
	1.4.13	Conteúdo em foco por mouse ou teclado	AA	Conteúdos adicionais não devem ser acionados apenas com foco por mouse ou teclado.
	2.1.1	Teclado	A	Todas as funcionalidades devem ser acionadas via teclado.
	2.1.2	Sem bloqueio de teclado	A	Ao se interagir via teclado, a navegação por todos os elementos "cliqueáveis" deve ocorrer sem que haja bloqueios ou interrupções.
	2.1.3	Teclado (sem exceção)	AAA	Todas as funcionalidades devem ser acionadas via teclado, sem exceção.
	2.4.7	Foco visível	A	Ao se interagir por teclado, qualquer pessoa deve conseguir identificar qual é a sua localização espacial na tela através de um foco visível.
	2.5.3	Rótulo no Nome acessível	A	Rótulos em botões, ícones acionáveis ou qualquer controle interativo, devem ter uma descrição significativa.
	2.5.5	Tamanho da área clicável	AAA	O tamanho das áreas acionáveis por clique ou toque devem possuir no mínimo 44x44 pixels de espaçamento
6	3.2.1	Em foco	A	O foco sempre deve se manter durante a navegação, guiando o usuário e evitando mudanças que causem desorientação.
	1.4.12	Espaçamento de texto	AA	Sempre que houver um redimensionamento os textos não devem perder legibilidade
	1.4.13	Conteúdo em foco por mouse ou teclado	AA	Conteúdos adicionais não devem ser acionados apenas com foco por mouse ou teclado.
	2.1.1	Teclado	A	Todas as funcionalidades devem ser acionadas via teclado.
	2.1.2	Sem bloqueio de teclado	A	Ao se interagir via teclado, a navegação por todos os elementos "cliqueáveis" deve ocorrer sem que haja bloqueios ou interrupções.
	2.1.3	Teclado (sem exceção)	AAA	Todas as funcionalidades devem ser acionadas via teclado, sem exceção.
	2.4.3	Ordem do foco	A	A interação por elementos focáveis na tela sempre deverá ser sequencial e lógica de acordo com o conteúdo apresentado.
	2.4.7	Foco visível	A	Ao se interagir por teclado, qualquer pessoa deve conseguir identificar qual é a sua localização na tela através de um foco visível.
	2.4.8	Localização	AAA	Qualquer pessoa deve conseguir se localizar ou se orientar facilmente em qualquer nas telas.
	2.5.2	Cancelamento de acionamento	A	Deve se fornecer um modo de cancelar acionamentos feitos de forma não proposital.
	3.2.3	Navegação consistente	AA	Deve-se manter a consistência com relação ao formato de apresentação, interação e localização na tela.
7	3.2.4	Identificação consistente	AA	Deve-se manter a consistência com relação a diferentes formatos de elementos, mas que possuem uma mesma funcionalidade.
	1.4.3	Contraste (mínimo)	AA	Textos devem ter uma relação de contraste entre primeiro e segundo plano de ao menos 4.5:1.
8	1.4.6	Contraste (melhorado)	AAA	Textos devem ter uma relação de contraste entre primeiro e segundo plano de ao menos 7:1.
	4.1.3	Mensagens de status	AA	Qualquer tipo de mensagem que é resultado de uma ação deve ser transmitida sem que ocorra uma mudança de foco na tela.

### 3.3 Fase 2: Adaptação e construção da *interface* do usuário acessível e usável com base no guia de acessibilidade e *style guide*

A segunda fase se concentrou na adaptação da *interface* do usuário e na construção de novos componentes para o *e-commerce*, sempre seguindo os guias, como exemplificado na figura 9. Nessa fase o foco principal se dividiu em duas vertentes:

- Criar abordagens inteligentes baseadas em casos de uso para problemas de acessibilidade em imagens, formulários e navegação e a adaptação de componentes com *HyperText Markup Language* (HTML) [13] não semântico para semântico.
- Adaptar e criar componentes para que as tecnologias assistivas pudessem entendê-los corretamente e assim ter uma a navegação por teclado eficaz.



Figura 7: Fase 2 do guideline

Fonte: a autora (2022)

#### 3.3.1 Abordagens inteligentes e adaptação de componentes

O primeiro passo a se fazer foi organizar o *HTML* de forma lógica e semântica, onde cada *tag* deve ser usada para o fim que ela foi criada, seguindo uma ordem que seja compreensível além de corresponder ao conteúdo desejado. Nesse ponto em um sistema legado muitos componentes estavam escritos de maneira errada, algumas *tags HTML* estavam sendo usadas com um propósito diferente do que ela foi criada e para esses casos foi utilizado a abordagem *Accessible Rich Internet Applications Suite* (WAI-ARIA) [14] para que o leitor de tela conseguisse entender o que era cada *tag* daquela e também fosse possível fazer a navegação via teclado.

##### **Caso de uso 1:** Transformando botão inacessível em acessível

Adaptação de um botão que foi construído de forma não acessível, como exemplificado no código 1, e por isso não era lido por leitores de tela e nem navegável via teclado. No exemplo do código 2 foi usado alternativas para deixar o botão acessível, foi adicionado o atributo `role` do *WAI-ARIA*[14] para dar o comportamento de botão a essa `div` e com isso ganhar todos os atributos e comportamentos que um botão teria.

Listing 1: Componente de botão antes de receber boas práticas de acessibilidade.

```

1 | <div
2 |   className=#button__add"
3 |   onClick={event => onClick( event)}
4 | >
5 |   Adicionar ao carrinho
6 | </div >

```

Listing 2: Adaptação do componente de botão usando o atributo *role* presente no *WAI-ARIA* [14].

```

1 | <div
2 |   role=" button"
3 |   tabIndex="0"
4 |   className=#button__add"
5 |   onClick={event => onClick( event)}
6 |   onKeyDown={event => keyHandler( event,  props.onClick)}
7 |   {...  props}
8 | >
9 |   Adicionar ao carrinho
10 | </div >

```

**Caso de uso 2:** Adaptação e construção de imagens acessíveis. As imagens que possuem algum tipo de conteúdo devem ser acessíveis, elas precisam de alguma descrição, seja ela visível ou não. Existe o atributo `alt` para a `tag` `img` como exemplificado no código 3, onde ele serve para descrever o que está presente na imagem. Essa descrição do `alt` não aparece visualmente, mas ela é lida pelo leitor de tela, quando o usuário, navegando pelo teclado, passar pela imagem.

Listing 3: Uso atributo `alt` para descrever o item da imagem.

```

1 | 

```

Caso haja necessidade de fornecer uma informação contextualizada é necessário fazer uso do atributo `title` e nele colocar a informação desejada. Nesse caso, a maioria dos leitores de tela lerá o texto alternativo, o atributo de título e o nome do arquivo. Além disso, os navegadores exibem o texto do título como dicas de ferramentas quando estão sobre o *mouse*, ou seja quando o mouse está parado em cima da *tag* que possui esse atributo . No exemplo do código 4 foi aplicado duas *tags* semânticas presentes no *HTML 5*, esses dois elementos o `<figure >` e `figcaption >`, servem para associar uma figura a uma legenda.

Listing 4: Uso do atributo `alt` em conjunto com as *tags* `figure` e `figcaption`.

```

1 | <figure >
2 |   
3 |   <figcaption>Headphonecom microfone acoplado na cor preta, com um fio de ótima
4 |     construção</figcaption >
5 | </figure >

```

Ainda há os casos onde as imagens são apenas decorativas e por elas não carregarem conteúdo

elas devem ser ignoradas pelos recursos de tecnologia assistivas. Para esse caso há três tipos de abordagens:

- Por ser decorativa a melhor abordagem é incluí-las na página como imagens de fundo através de *Cascading Style Sheets* (CSS) [15], como exemplificado no código 5 e caso a imagem transmita significado ao conteúdo mas tenha sido inserida via CSS, pode-se utilizar recursos *WAI-ARIA*, exemplificado no código 7 .

Listing 5: Adição de imagens via CSS.

```
1 | <style >
2 | div {
3 |     background-image url ( "./ images/logotipo . png" );
4 | }
5 | </style >
```

- Em casos de imagens que usam a *tag* `img` basta ter uma descrição `alt` vazia. Isso faz com que os leitores de tela reconheçam a imagem mas não tentem descrevê-la, como exemplificado no código 6 abaixo. A razão para usar um `alt` vazio ao invés de não incluí-lo é porque muitos leitores de tela anunciam o *Uniform Resource Locator* (URL) da imagem inteira se nenhum `alt` for fornecido.

Listing 6: Uso do atributo `alt` vazio.

```
1 | 
```

- Como exemplificado no código 7 em casos de imagens incluídas em *tags* que não são `img` aconselha-se fazer uso do atributo *WAI-ARIA* `role` ( `role="presentation"` ), isso também impede que os leitores de telas leiam textos alternativos.

Listing 7: Uso atributo *WAI-ARIA* [14] `role presentation`.

```
1 | <div role="presentation" src="logotipo . jpg" />
```

### Caso de uso 3: Criando formulários acessíveis.

O primeiro passo para criar um formulário acessível é criar uma estrutura *HTML* com sequência lógica, essa sequência é definida pela ordem que o *HTML* é escrito. Fazendo isso a navegação pelos campos do formulário ficará mais fácil tanto por teclado quanto para leitores de tela. Para se construir formulários acessíveis deve-se tomar os seguintes cuidados na hora da implementação:

- A associação de um campo de entrada a um rótulo fará diferença para o usuário final. E quando não for possível fazer essa associação por *tags HTML* existe outras opções como a forma *WAI-ARIA* usando o atributo `aria-label` .
- É necessário o fornecimento de instruções claras sobre os dados de entradas pedidos no campo. As entradas devem ser facilitadas, os caracteres especiais em campos numéricos devem ser retirados etc.

- Para campos com obrigatoriedade de dados, deve-se ter um indicador visual e escrito de obrigatoriedade. O atributo `required` auxilia no quesito de obrigatoriedade de um campo.
  - Caso necessário há a possibilidade de se adicionar o atributo de estado `aria-invalid` para indicar se o campo está válido ou inválido e ele pode ter os valores `true` ou `false`.
- A identificação e descrição de erros de entrada e confirmação de envio de informações devem ser claras para o usuário.
    - A indicação do erro é feita de três formas: com uma cor, um texto informativo e um ícone de alerta. Assim, não é usada somente uma forma para representar esta informação, beneficiando também pessoas com limitações visuais e com deficiências cognitivas que usam leitor de telas ou que possuem daltonismo;
    - Uma área para mensagem informativa em um formulário sempre será bem-vinda. Esta área de mensagem informativa do formulário possui os atributos `role="alert"`, que avisa que é uma mensagem de alerta e `aria-live="assertive"` que informa que é um campo que pode sofrer modificações no conteúdo e o leitor de telas é informado quando há modificação neste campo;
  - É necessário haver distinção visual de qual campo está recebendo o foco no momento, mesmo aplicando o `outline` padrão do navegador. A customização desse `outline` pode ser feita para que ele sempre fique visível qual campo contém o foco.

O código construído no exemplo 8 é possível observar todas as enumerações da lista acima desde a construção do HTML [?] ordenado, passando por associação de rótulos a campos e instruções sobre o tipo de entrada que é esperado como também o *feedback* conciso para eventuais indicações de erros. Como resultado do código 8 após a compilação do código desenvolvido temos a imagem 8 que apresenta de forma visual todas as enumerações desenvolvidas no código.

Listing 8: Formulário acessível usando abordagens citadas acima

```

1 <form>
2   <output id="formMessage" role="alert" aria-live="assertive" tabindex="0" class="
3     error" >
4     O formulário apresenta erros que impedem a finalização do seu cadastro.
5     Confira se todos os campos obrigatórios foram preenchidos e tente
6     novamente
7   </output>
8
9   <p>
10    <label for="email1" >
11      E-mail <span class="required" >(obrigatório) </span>:
12    </label >
13    <input type="email" name="email1" id="email1" required="" aria-invalid="true"
14      aria-describedby="emailMessage" >
15    <span class="fieldMessage" id="emailMessage" >
16      O e-mail não pode ficar em branco e deve ser válido.
17    </span>
18  </p>
19 </form>

```

```

17 <label for="phone1">
18     Telefone <span class="optional">(opcional)</span>:
19 </label>
20 <input type="tel" name="phone1" id="phone1" aria-describedby="phoneTip">
21 <span class="fieldTip" id="phoneTip">
22     Formata (99) 99999-9999.
23 </span>
24 </p>
25
26 <button class="formButton" id="formButtonARIA" type="submit">Finalizar cadastro </
27 </form>

```

O formulário apresenta erros que impedem a finalização do seu cadastro.  
Confira se todos os campos obrigatórios foram preenchidos e tente novamente.

E-mail (obrigatório):

john.doe@gmail.com 

O e-mail não pode ficar em branco e deve ser válido

Telefone (opcional):

Formato: (99) 99999-9999

**FINALIZAR CADASTRO**

Figura 8: Exemplifica o código 8 após a compilação.

Fonte: a autora 2022

### 3.3.2 Adaptação de componentes para o uso de leitores de telas e navegação por teclado

Antes de falar de adaptação de componentes é necessário entender como funciona o leitor de telas. O leitor de telas basicamente é um software que vai percorrendo textos e imagens e lendo em voz alta tudo o que ele encontra na tela, a partir do momento que o usuário começa a interagir com o software para efetuar uma navegação ele vai percorrendo o documento saltando entre elementos interativos e cabeçalhos. Para que essa navegação ocorra tudo deve está funcionando direito, pois caso contrário o usuário pode se perder na página e ter dificuldades em entender como a informação

está organizada. No leitor de telas a navegação ocorre de três maneiras:

- Navegação com as setas direcionais: Usando a navegação por setas direcionais o usuário consegue acessar as informações textuais.
- Navegação por tecla *tab*: O leitor sai pulando para os links espalhados na página.
- Navegação com a tecla *h*: O leitor salta entre os cabeçalhos marcados na página.

#### **Caso de uso 1:** Navegação por cabeçalho.

Cabeçalhos compõem a hierarquia de informação do site, essa hierarquia é usada pelos leitores de tela para navegação dentro da página. A grande maioria dos leitores de tela dispõem de atalhos para acesso rápido aos cabeçalhos marcados na página. Para garantir que tecnologias assistivas acessem o conteúdo dos cabeçalho sem sua ordem de importância é preciso tomar alguns cuidados:

- Cuidado com múltiplas *tags h1* na página. Mesmo que haja separação de contexto para a utilização de mais de um *h1* por página o leitor de tela não consegue identificar a diferença e a importância de *h1* contidos em outras *tags*.
- Para essa navegação funcionar, o código precisa ter sido escrito com a semântica correta, ou seja, cabeçalhos devem ser definidos como cabeçalhos e não usar qualquer outro artifício visual para se assemelhar a um cabeçalho, como exemplificado abaixo.

#### Listing 9: cabeçalhos com hierarquia

```
1 | <h1>Cabeçalho Principal </h1>  
2 | <p>Exemplo de um texto de um parágrafo </p>  
3 | <h2>Cabeçalho Secundário </h2>  
4 | <p>Mais texto de um parágrafo.</p>
```

#### **Caso de uso 2:** Navegação por tabulação e leitores de tela

A ordem da navegação por tabulação é derivada da forma como o *layout* do *site* está escrito, contudo a ordem padrão pode não corresponder necessariamente a ordem visual. Ao utilizar a tecla *tab*, o leitor de tela realiza o foco e lê em voz alta os *links* e componentes interativos presentes na página. A tabulação é utilizada somente para navegar entre os *links*, e com isso todos os *links* terão que ser localizados através destas duas formas, com a tabulação ou com as setas. O que diferencia uma forma da outra, é que com as setas, o usuário poderá ler toda a página (incluindo texto entre os *links*), e com a tabulação, o usuário só pode localizar os *links*, botões, campos de edição, e caixas de seleção.

Navegar por entre os *links* é uma forma de observar o texto rapidamente, especialmente se os usuários estiverem tentando encontrar uma seção específica do *website*. Para ter uma navegação por tabulação eficiente é preciso passar por alguns pontos:

1. Ordene de forma lógica e intuitiva a leitura e tabulação. Preparar o *HTML* para ter uma navegação consistente e eficaz é essencial para o bom funcionamento da navegação atrelada a leitores de tela.

2. Usar a tecla *tab* para passar pelos *links* e controles de formulários das páginas, certificando-se de que todos os links e controles de formulários podem ser acessados, bem como se os links indicam claramente para onde levam.
3. Com atributo `tabindex` é possível definir uma ordem de foco nos elementos dispostos na tela. Usando o `tabindex=0` é inserido um elemento na ordem natural de tabulação, como visto no exemplo do código 10, e com `tabindex="-1"` um elemento é removido da ordem e tabulação.

Listing 10: Uso do atributo *tabindex* para adicionar um elemento a ordem de foco

```

1 | <label >
2 |     Primeiro na lista de tabulação:<input type="text" >
3 | </label >
4 | <div tabindex="0" >
5 |     Próximo item na lista de tabulação, mesmo não sendo um elemento que
6 |     receberia o foco natural
7 | </div >
8 | <div >Não será focado pois está sem o tabindex</div >

```

4. Adicione o atributo `title` aos *links*, ele mostrará uma descrição do lugar pra onde o *link* leva, melhorando a navegação. Ao ter o elemento focado o leitor conseguirá ler o atributo `title` e saber a direção que o redirecionamento poderá tomar ou se um elemento novo surgirá, como exemplificado no código 11.

Listing 11: Uso do atributo *title* para dar uma explicação sobre onde o link leva

```

1 | <a href="#" onclick="abrePopu() " onkeypress="abrePopu() " title ="Abre uma
2 |     janela pop-up com Javascript ">Ver mais informações</a>

```

5. No exemplo exposto no código 11 é usada a abordagem de *landmarks*, que consiste no fornecimento de âncoras para ir direto a um bloco de conteúdo. São comumente chamados de *landmarks* é um tipo de região em uma página *web* a qual uma pessoa pode querer acessá-la rapidamente. Um dos principais benefícios em se usar pontos de referência é a possibilidade de o usuário acessar diretamente uma região da página sem a necessidade de seguir a ordem natural do conteúdo. Isso é especialmente relevante para o caso de pessoas que utilizam leitores de tela para navegar na *web*. Para essa abordagem é muito utilizado o CSS [15] deixando-os *links* invisíveis e apenas disponíveis quando a navegação por tabulação se iniciar.

Listing 12: Criação de *landmarks* para acessar pontos de referências através do atalho p.

```

1 | <div role=" heading" id=" cabecalho" >
2 |     <h1>O Cabeçalho</h1>
3 |     <a href="#conteudo" accesskey="p">Pular para o conteúdo principais </a>
4 | </div >
5 | <div role=" navigation " id=" navegacao" > conteúdo da navegação </div >
6 | <div role=" main" id=" conteudo" > conteúdo da main </div >

```

6. É aconselhável não tabelas para diagramação. As tabelas devem ser utilizadas apenas para dados tabulares e não para efeitos de disposição dos elementos na página. O leitor de telas

sairá pulando por cada campo das tabelas e o usuário poderá ficar perdido na navegação. Para efeitos de diagramação a melhor opção está na utilização do CSS.

7. Utilização de separação dos *links* adjacentes. Em uma sequência de *links*, além do espaço, é importante o uso de separadores ou elementos do HTML adequados para que as pessoas com deficiência identifiquem claramente onde termina e começa um novo *link*. É recomendado o uso de listas, assim como no exemplo do código 13 , em que cada elemento dentro da lista é um link.

Listing 13: Separação de *links* adjacentes usando a *tag* li.

```
1 | <ul id="menü">
2 |   <li > <a href="homehtml">Home</a> </li >
3 |   <li > <a href="pesquisa.html">Pesquisa</a> </li >
4 | </ul >
```

8. É desencorajado a abertura novas instâncias sem a solicitação do usuário. O impedimento de redirecionamento automático de páginas é algo necessário. Forçar a abertura de *links* em uma nova janela não é algo esperado e o usuário se perderá na navegação. Caso implemente essa funcionalidade, o *link* deve possuir uma iconografia óbvia e explicativa, usando pseudo-elemento `::after` no CSS. No CSS o conteúdo de pseudo-elementos é lido em voz *alt* a por leitores de tela, portanto é de extrema importância de que o conteúdo seja com informações relevantes.
9. O conteúdo deve possuir o foco por mouse ou teclado e esse foco deve ser visível. Para pessoas com baixa visão, é muito importante que seja possível perceber facilmente onde está o foco do teclado, garantindo uma maior facilidade de navegação. Como no exemplo do código 14 pseudo-classe `:focus` é utilizada para definir o estilo de qualquer elemento HTML que receber o foco do teclado, como *links* e elementos de formulário.

Listing 14: Exemplo de como estilizar o o elemento nos estados de foco e hover.

```
1 | a: focus, a: hover {
2 |   border: 2px solid #F00;
3 | }
```

10. Todas as funcionalidades que estão disponíveis via mouse deverão estar disponíveis para serem acionadas via teclado. Só dessa forma o site estará acessível para pessoas com deficiências visuais.

### 3.4 Fase 3: Construção do dashboard acessível

Para a fase de construção do dashboard acessível o foco se manteve na construção dos gráficos de forma acessível, navegável e reutilizável como exemplificado na figura 9.

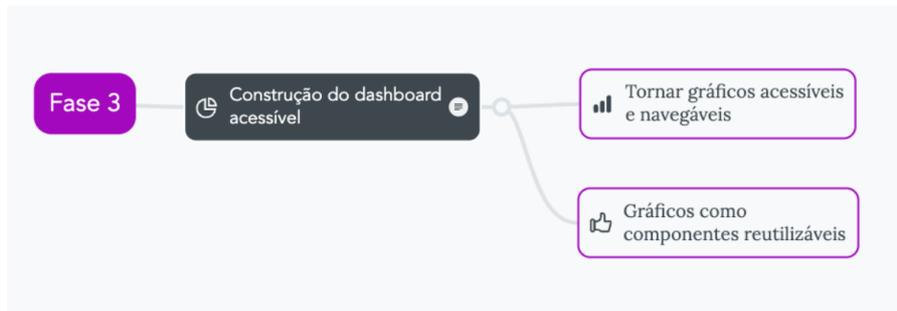


Figura 9: Fase 3 do guideline

Fonte: a autora 2022

Para a construção dos gráficos do *dashboad* foi utilizada a biblioteca D3.js [16] que é usada para visualização de informação. Essa biblioteca permite que sejam construídas aplicações em que os dados entram puros e são dinamicamente associados em representações gráficas como exemplificado em 10. Com o D3 o *Document Object Model* (DOM) [17] pode ser facilmente manipulado já que ele faz uso dos padrões da *web* como *HTML*, *CSS* e *Scalable Vector Graphics* (SVG) para renderizar gráficos de visualização poderosos.

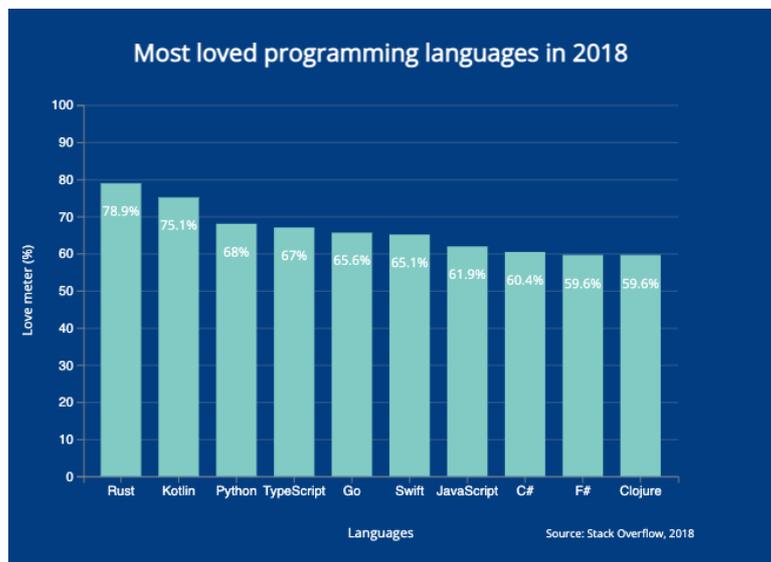


Figura 10: Um gráfico desenvolvido com D3.js

Fonte: a autora 2022

### Construção de gráficos com D3.js

No *e-commerce* foi decidido não construir o gráfico inteiramente com D3.js, pois a abordagem normal presente não facilitaria o reaproveitamento para a criação dos componentes. Quando um gráfico é plotado somente com D3.js pode haver problemas de performance, visto que o React [18] não estava preparado para esse tipo de renderização contínua criada ao se aplicar filtros e interações nos gráficos. Dessa forma, a abordagem mais coerente era usar as *tags HTML* normais e quando era necessário feito a criação do elementos e interações usando o D3.js.

Para a manipulação do DOM, é necessário aprender a utilizar os recursos presente no D3. A biblioteca usa JavaScript para realizar a maioria das tarefas de seleção, transição e vinculação de dados.

A primeira coisa necessária para construir um gráfico com D3.js é adicionar a biblioteca ao projeto, seja por gerenciador de pacotes ou por importação direta na `tag script`. Feito isso, é preciso ter uma `tag` como base onde as partes do gráfico serão injetadas, geralmente é usada a `tag svg` pois ela auxilia na construção de formas geométricas que será utilizado no gráfico. Após adicionar a `tag svg` precisamos adicionar uma nova `tag g` para que o gráfico comece a tomar forma, essa `tag g` é um recipiente utilizado para agrupar objetos.

Para começar a desenhar qualquer tipo de gráfico é necessário colocar os eixos das abscissas e ordenadas. Na construção desses eixos deve-se estabelecer qual o domínio de cada eixo e após isso iniciar a construção das escalas, para que assim todos os dados se comportem dentro dos limites estabelecidos em função da escala. O próprio D3 já fornece alguns tipos de funções que auxiliam na criação dessas escalas, as funções são: `d3.scaleBand()`, esta função divide o intervalo em n bandas onde n é o número de valores na matriz de domínio e `d3.scaleLinear`, que é usada para criar um ponto de escala visual. São criados duas escalas, uma para o eixo X usando a `d3.scaleBand()` e outra para o eixo Y usando a `d3.scaleLinear`.

O próximo passo é criar por fim o conteúdo do gráfico de forma visual. Com o apoio de dados vindos de uma API é possível atribuir ao longo do eixo X e do eixo Y os valores e formas visuais necessárias. Utilizando a função `attr` é definido os atributos do elemento, uma vez que essa função recebe dois parâmetros o nome do atributo e o valor desse atributo. Graças a essa função são definidas a altura, largura e os valores de cada ponto dos eixos.

Depois que todos os valores dos gráficos estiverem corretamente plotados é o momento de lidar com as interações que estarão no gráfico. Todas as interações presentes via mouse devem também estarem disponíveis via teclado, ou seja, se houver uma interação de clique de mouse deve haver uma interação de pressão de tecla. Para auxiliar na construção dessas interações pode-se usar a função `.on()` que ficará responsável por agir como um *event listener* de um elemento, onde qualquer tipo de interação será percebida. Essa função recebe dois argumentos, sendo o primeiro o tipo de evento e o segundo uma função que será disparada quando esse evento for detectado.

### **Tornando os gráficos acessíveis a leitores de tela**

Além de tornar os gráficos acessíveis a qualquer tipo de navegação e interação é preciso que as imagens e os elementos presentes também estejam acessíveis e para isso existem algumas abordagens eficientes.

O gráfico exemplificado abaixo 11 foi construído levando em consideração as barreiras de acessibilidade que um deficiente visual frequentemente se depara. Logo, para diminuir essas barreiras as decisões tiveram como apoio as diretrizes WCAG que se adequam ao propósito do *e-commerce* e a construção do gráfico com D3.js se deu com o apoio das mesmas.

Tomando como exemplo a figura 11, a tabela 2 logo abaixo exemplifica cada barreira de acessibilidade e sua respectiva tomada de decisão.

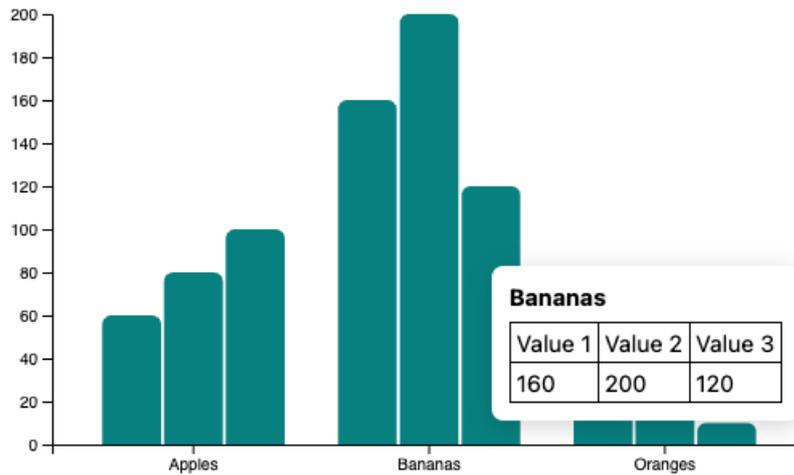


Figura 11: Gráfico do tipo de barras estacadas.

Fonte: a autora 2022

Barreira de acessibilidade	Tomada de decisão
Falta de atributo <code>alt</code> ou atributo não descritivo	Se faz necessário o uso do atributo <code>alt</code> . Os leitores de tela falam texto alternativo sem permitir que os usuários acelerem ou pulem, portanto, as informações sejam descritivas, mas sucintas.
Falta de rotulação nos dados	Legendas podem ser não eficientes para usuários daltônicos ou deficientes visuais que podem ter dificuldade em combinar as cores do gráfico com as da legenda. Rotulando também diminuirá o trabalho de varredura para frente e para trás tentando combinar a legenda com os dados.
Dados fornecidos apenas por gráficos	Forneça os dados de algum outro modo que não somente por gráficos: Incluir um link para um CSV ou outro formato de dados legível por máquina para que pessoas com deficiência visual possam navegar pelos dados do gráfico com um leitor de tela é essencial.
Pouco contraste entre o fundo do gráfico e o conteúdo do gráfico	Verifique se o contraste de cores do gráfico está dentro dos padrões da WCAG.
Separação não clara entre os setores dos gráficos	Utilizar espaços em branco para separar gráficos que são empilhados. O uso criterioso do espaço em branco aumenta a legibilidade, ajudando a demarcar e distinguir as diferentes seções sem depender apenas da cor. Isso também pode complementar as opções de cores acessíveis, ajudando os usuários a distinguir a diferença entre as cores que identificam seções separadas.

Quadro 2: Exemplificação de barreiras e decisões na construção de um gráfico acessível

### Gráficos como componentes reutilizáveis

Quando se trabalha com React ao se criar um componente é preciso ter a garantia de que esse componente possa ser reutilizável, para que isso seja possível no momento da criação do componente deve-se ter em mente que quanto menor e mais genérico for o componente mais ele será escalável e com isso facilitará a construção de componentes maiores.

Para facilitar a construção como módulos separados cada tipo de gráfico se tornou um componente reutilizável. Em cada componente desses contaria com suas funções e particularidades, porém com a flexibilidade de ser um componente genérico, onde todos os dados são populados através de propriedades que são repassadas do componente pai para o componente filho.

## 3.5 Fase 4: Testes e avaliações de acessibilidade

Durante e após a construção do *e-commerce* de acordo com os padrões *web* e as diretrizes de acessibilidade, é necessário efetuar testes para garantir sua acessibilidade. Todos os testes descritos nessa seção foram executados por pessoas que não são os *stakeholders* do projeto mas

fazem parte da equipe de *quality assurance*.

Com o objetivo de cobrir o máximo de casos de testes possível as avaliações abordaram dois modos de testes: a avaliação automático e a manual, como exemplificado na figura 12.

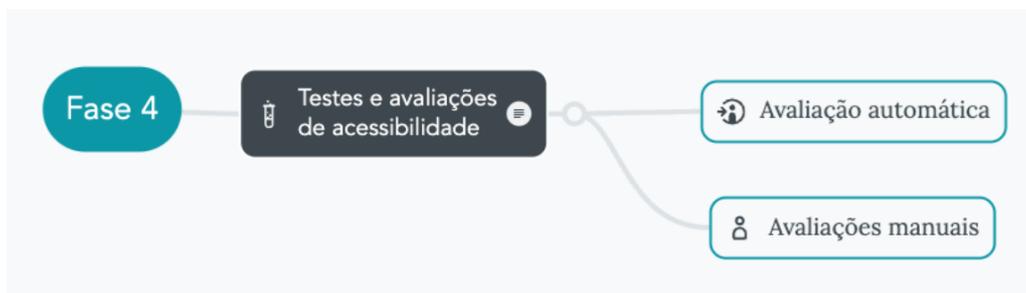


Figura 12: Fase 4 do guideline

Fonte: a autora 2022

- **Avaliação automática:** São as avaliações feitas de maneira automática por meio de *plugins* de verificação e de *websites* de validação. Quando os testes automáticos não atingiam os requisitos de aceitação era gerado um relatório contendo os pontos de falhas dentro do teste, esse relatório era repassado para a equipe de desenvolvimento para que as devidas correções fossem feitas.
  - *Google Accessibility Developer Tools* (Plugins para navegador *Chrome*).
  - *DevTools* (Plugins para navegador *Chrome*).
  - *Contrast Checker* (*website* - Validação de contraste de cores).
- **Avaliações manuais:** Essa etapa se deu usando o leitor de tela e a navegação por teclado, onde foi checado se todos os comportamentos e ações esperados estão sendo realmente respeitados e executados. Quando os testes manuais não atingiam os requisitos de aceitação a equipe de *quality assurance* reportava aos desenvolvedores os pontos de falhas para que fossem corrigidos.

### 3.6 Tecnologias utilizadas

A linguagem de programação JavaScript foi escolhida para a elaboração deste trabalho devido a sua grande relevância no *front-end*. Entre as principais bibliotecas utilizadas neste projeto destaca-se:

- **HTML:** é uma linguagem de marcação utilizada na construção de páginas na *Web*.
- **Diretrizes WCAG:** é um conjunto de regras que definem a forma de como tornar o conteúdo da *Web* mais acessível para pessoas com deficiência.
- **Atributos WAI-ARIA:** Especificações técnicas que oferecem maneiras de tornar as aplicações mais acessíveis a uma diversidade maior de pessoas, incluindo quem utiliza tecnologias assistivas, como leitores de telas.

- React.js: É uma biblioteca *JavaScript* para criar interfaces de usuário, ela é declarativa e baseada em componentes.
- Sass: Ou *Syntactically Awesome Style Sheets* é uma linguagem de extensão do CSS e tem como objetivo tornar o processo de desenvolvimento mais simples e eficiente.
- D3.js: Trata-se de uma biblioteca JavaScript para construção de visualizações de dados. O *D3* permite que visualizações sejam criadas diretamente em páginas *HTML* através de gráficos vetorizados [SVG](#)
- Tecnologias assistivas: Para a construção do *e-commerce* foram usados alguns tipos de leitores de telas que transcrevem para linguagem falada o conteúdo textual presente na tela do dispositivo eletrônico.
  - VoiceOver: Leitor de tela presente nos produtos e ambientes virtuais confeccionados pela *Apple*.
  - NVDA: Leitor de telas voltado para ambientes que usam *Windows*
  - ORCA: Leitor de telas voltado para ambientes *Linux*

### 3.7 Contribuição

A atuação da autora desse estudo consistiu na construção de soluções de interface que melhorem a experiência do usuário e no desenvolvimento de tais soluções em softwares. Para criar esses softwares aplicando essas soluções foram utilizadas tecnologias de *front-end*. Portanto, a autora esteve presente desde a fase de ideação de *design* até o desenvolvimento do *e-commerce* fazendo o uso de JavaScript e da biblioteca React para a construção do mesmo. A solução apresentada não se encontra em ambiente de produção pois há outros épicos que não englobam acessibilidade que ainda estão sendo construídos dentro do *e-commerce*. Entretanto, todo o épico de acessibilidade e *Data Visualization* foi muito bem avaliado pelo cliente que se mostrou muito feliz com o que foi apresentado e aprovou todo o progresso feito.

O projeto, o guia construído, o *style guide* e o conhecimento adquirido para a construção da solução se tornou um case para outros desenvolvedores e designers que buscam aprender mais sobre a acessibilidade e aplicá-la de um modo descomplicado e correto, sendo apresentado como case em reuniões de troca de conhecimentos entre pessoas desenvolvedoras e designers.

## 4 Dificuldades encontradas

Uma das maiores dificuldades encontradas ao longo do desenvolvimento deste estudo foi o fato de ter que lidar com uma documentação extremamente extensa do WCAG. Como a WCAG são as diretrizes de acessibilidades para vários tipos de deficiências foi preciso ler toda a documentação para depois focar nas diretrizes que deveriam ser aplicadas de fato. Isso exigiu muito esforço, pois além de ter muita informação em cada diretriz deveria ser feita uma avaliação para ter a certeza se a diretriz em questão se aplicava nas necessidades e requisitos levantados pelo cliente.

Adicionalmente, lidar com um sistema legado nunca foi fácil e torná-lo acessível foi verdadeiramente um desafio para o time inteiro, pois havia muitos componentes inacessíveis que deveriam ser adaptados ou refeitos. Essa escolha entre adaptar ou refazer se tornou complicada, pois refazer por vezes se torna mais atrativo, mas quando lidamos com prazo de entrega essa escolha deve ser ponderada pelo time inteiro.

O sistema já tinha certos problemas de performance e isso foi uma preocupação para o time quando entramos na parte de *Data Visualization*, pois a plotagem de gráfico pode se tornar algo custoso ao sistema. Para minimizar os problemas de performance foi feita uma varredura no sistema para detectar os pontos de gargalo e no dashboard foi usado o [lazy loading](#) nativo do JavaScript que ajudou em possíveis gargalos ao se plotar muitos gráficos de uma só vez.

## 5 Impactos da sua formação no seu trabalho

O curso de Bacharelado em Sistemas de Informação busca formar os alunos com um perfil multidisciplinar sempre com o foco para o mercado de TI. Nesta seção são exploradas as disciplinas curriculares do curso que foram de extrema importância na vivência profissional.

Metodologia ágil é algo bem forte dentro do mercado e uma das formas mais eficientes para se trabalhar com entregas. Os conhecimentos adquiridos na disciplina de gerência de projetos de software sobre entregas com *sprints* semanais, reuniões como *daily*, *planning* e *sprint review* foram essenciais para entender e aplicar essa forma de trabalho.

A linguagem de programação JavaScript, abordada na disciplina de desenvolvimento de aplicações para *web*, é essencial para qualquer solução que sejam desenvolvida para a camada do *frontend*. Juntamente com todos os conhecimentos adquiridos na disciplina de Interfaces Homem-Máquina foram essenciais para entender as necessidades de se construir uma interface que faça sentido para todos os usuários e com isso construir uma solução eficiente tanto na parte visual quanto na parte de eficiência de código.

Todas as disciplinas que contaram com trabalhos feitos em grupos foram de extrema importância para despertar a consciência e responsabilidade por prazos e entregas. Por estar em grupo ficou muito evidente a importância de se manter a comunicação para a saúde do projeto e esses pontos foram essenciais na construção da postura profissional.

## 6 Conclusão

A praticidade de conseguir efetuar compras sem sair de casa é algo que vem sendo amplamente explorado por consumidores e comerciantes. Como foi visto ao longo desse trabalho, essa praticidade não se aplica para todos os tipos de consumidores. Existe uma parcela da população que não consegue fazer suas compras, mesmo esse sendo um direito conferido por lei.

Pela internet ser algo tão inerente e necessária à vida humana, a experiência com o seu uso deve ser semelhante para todas as pessoas, sejam elas deficientes ou não. Para promover essa semelhança de experiência foi necessário repensar o *e-commerce* nos mínimos detalhes levando

em conta a própria complexidade de construção sem ter a acessibilidade devida.

Um ponto levado em consideração neste trabalho foi que dificilmente um site consegue ser 100% acessível, pois muitas das diretrizes se sobrepõem ou desfazem a construção de outra. Sendo assim, deve-se trabalhar com a maximização de quebra de barreiras de acesso. No caso do *e-commerce* construído e detalhado nesse trabalho o foco das quebras barreiras se deu para com as pessoas que possuem alguma limitação ou deficiência visual.

Toda a construção desse trabalho converge em prover abordagens sobre a falta de acessibilidade em lojas *on-line*. Logo, o principal objetivo foi transformação de um *e-commerce* não acessível em acessível e para alcançá-lo durante o processo de desenvolvimento ficou claro a necessidade de documentos que apoiassem a pessoa desenvolvedora durante a construção, visto que acessibilidade é um ponto que acaba sendo deixado de lado pois não há muita procura partindo da comunidade de desenvolvedores. Esses documentos especificavam decisões de design e de comportamento diante da *interface* do usuário e se mostraram excepcionais como apoio para tomadas de decisões, esclarecimento de dúvidas além de ser o guia de padrões para a construção da UI.

Outro ponto de extrema importância presente nesse trabalho e pouco difundido diante dos padrões de acessibilidade e desenvolvimento está na construção de gráficos que são acessíveis e navegáveis. O objetivo principal de um gráfico é trazer a visualização da combinação de dados de forma facilitada e para isso se faz uso de cores e em termos de acessibilidade o maior desafio é conseguir construir o mesmo tipo de visualização facilitada para as pessoas que possuem limitações visuais.

A visualização facilitada entregue por um gráfico é um recurso muito importante no entendimento geral do conteúdo daquele gráfico e as pessoas que são deficientes visuais devem estar cientes de tudo o que se passa no gráfico e em seus dados. Para eliminar algumas barreiras de acesso quanto a essa visualização foi mostrada algumas técnicas de construção de gráficos de forma acessível utilizando a biblioteca D3.js. As barreiras de acesso mais abordadas quanto a gráficos foram a dificuldade de navegação que os usuário enfrentam e a falta de rotulação para cada parte que o gráfico dispõe, pois só não há como navegar por algo que não é lido pelo leitor de telas.

Mesmo tendo todas essas abordagens para diminuir as barreiras encontradas diante de *Data Visualization* existem casos complexos que apenas o fornecimento dos dados de uma forma que não seja por meio de gráficos será clara o suficiente para se tornar entendível para uma pessoas com deficiência visual.

Por vezes a acessibilidade passa despercebido por uma pessoa que não tem deficiência, mas esse panorama vem sendo modificado aos poucos. A acessibilidade deixou de ser um tema apenas difundido por pessoas com deficiência e começou a se tornar um tema costumeiro. A acessibilidade vem muito antes de transformar um *wireframe* em um *website*, ela vem desde a fase de planejamento.

Hoje em dia existem tecnologias e padrões de apoio para os desenvolvedores se empenharem em tornar a *web* acessível para todos. Contudo, há um débito comportamental por parte dos desenvolvedores de *software* uma vez que em algum momento do desenvolvimento não foi considerado os diferentes tipos de pessoas e necessidades. A *web* acessível depende daqueles que a constroem, acessibilidade não é opção é um direito e deve ser preservado por todos.

No que se refere às melhorias e trabalhos futuros, há alguns pontos a serem implementados de

modo a complementar a acessibilidade e usabilidade do *e-commerce*:

1. Realizar testes com os *stakeholders* e levantar pontos de melhorias mais focados nas necessidades de deficientes visuais.
2. Possibilidade da criação de um reconhecimento de padrões por meio de aprendizado de máquina para sugerir itens de compras de acordo com os gostos do usuário.
3. Implementação de resumos de visualização de dados de acordo com os filtros mais usados por aquele usuário, onde esses resumos devem apresentar duas formas de visualização: em texto corrido com a sua estrutura navegável e a outra em um documento pdf.
4. Implementação de filtros preferenciais e pré-programados no dashboard.

## Referências Bibliográficas

- [1] BigDataCorp. (2022, Setembro) Perfil do e-commerce brasileiro. [Online]. Available: <https://public.flourish.studio/story/947803/>
- [2] “Lei nº 913.146 de 06 de julho de 2015. vigência institui a lei brasileira de inclusão da pessoa com deficiência (estatuto da pessoa com defici).” *Diário Oficial D República Federativa do Brasil*, 07 de Julho. 2022. [Online]. Available: [http://www.planalto.gov.br/ccivil\\_03/\\_ato2015-2018/2015/lei/l13146.htm](http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2015/lei/l13146.htm)
- [3] IBGE. (2022, Setembro) Pns 2019: país tem 17,3 milhões de pessoas com algum tipo de deficiência. [Online]. Available: <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>
- [4] ACCENTURE. (2022, Setembro) The accessibility advantage: Why businesses should care about inclusive design. [Online]. Available: [https://www.accenture.com/\\_acnmedia/pdf-91/accenture-accessibility-advantage-pov-final.pdf](https://www.accenture.com/_acnmedia/pdf-91/accenture-accessibility-advantage-pov-final.pdf)
- [5] H. S. H., P. I., P. H., and L. W. L. (2022, Setembro) Inclusive design principles. [Online]. Available: <https://inclusivedesignprinciples.org/>
- [6] S. B. L. Ferreira, “E-acessibilidade: tornando visível o invisível,” *Revista Morpheus - Estudos Interdisciplinares em Memória Social*, vol. 6, no. 10, mar. 2015. [Online]. Available: <http://www.seer.unirio.br/morpheus/article/view/4780>
- [7] S. Lima, F. Lima, and K. De Oliveira, “Avaliação da acessibilidade de sítios web por meio de métricas de software,” *Anais do Simpósio Brasileiro de Qualidade de Software (SBQS)*, 2007. [Online]. Available: <https://sol.sbc.org.br/index.php/sbqs/article/view/15570>
- [8] P. J. Lynch and S. Horton, *Web style guide: Foundations of user experience design*. Yale University Press, 2016.
- [9] C. W. W. Web. (2022, Setembro) Diretrizes de acessibilidade para conteúdo web (wcag) 2.1. [Online]. Available: <https://www.w3c.br/traducoes/wcag/wcag21-pt-BR/>

- [10] —. (2022, Setembro) W3c - consórcio world wide web. [Online]. Available: <https://www.w3c.br/Sobre/>
- [11] I. Adobe. (2022, Setembro) Adobe color wheel. [Online]. Available: <https://color.adobe.com/pt/create/color-wheel>
- [12] C. W. W. Web. (2022, Setembro) Diretrizes de acessibilidade para conteúdo web (wcag) 2.0. [Online]. Available: <https://www.w3.org/Translations/WCAG20-pt-br/>
- [13] F. Flatschart, *HTML 5-Embarque Imediato*. Brasport, 2011.
- [14] W. contributors. (2022, Setembro) Accessible rich internet applications (wai-aria) 1.2. [Online]. Available: <https://www.w3.org/TR/wai-aria-1.2/>
- [15] M. contributors. (2022, Setembro) Css - folhas de estilo em cascata. [Online]. Available: <https://developer.mozilla.org/pt-BR/docs/Web/CSS>
- [16] M. Bostock. (2022, Setembro) D3.js - data-driven documents. [Online]. Available: <https://d3js.org/>
- [17] M. contributors. (2022, Setembro) Document object model (dom). [Online]. Available: [https://developer.mozilla.org/pt-BR/docs/Web/API/Document\\_Object\\_Model/Introduction](https://developer.mozilla.org/pt-BR/docs/Web/API/Document_Object_Model/Introduction)
- [18] I. Meta Platforms. (2022, Setembro) React - uma biblioteca javascript para criar interfaces de usuário. [Online]. Available: <https://pt-br.reactjs.org/>

## A Diretrizes WCAG

O apêndice em questão lista todas as diretrizes WCAG que foram usadas para serem aplicadas no *e-commerce* a fim de diminuir barreiras de acesso para deficientes visuais.

### A.1 Nível A

Quadro 3: Diretrizes do Nível A

Número	Nome	Descrição	Aplicação no projeto
1.1.1	Conteúdo Não Textual	Todo o conteúdo não textual que é exibido ao usuário tem uma alternativa textual que serve a um propósito equivalente, exceto para algumas situações.	sim
1.3.1	Informações e Relações	As estruturas da tela devem ser construída de forma que sua arquitetura de informação faça sentido tanto para todos, sejam ouvintes ou leitores.	sim
1.3.2	Sequência com significado	A apresentação das informações na tela sempre deverá ter uma sequência lógica.	sim
1.4.1	Utilização de cores	As cores não devem carregar significado lógico, elas não devem ser utilizadas como única maneira de transmitir conteúdo ou distinguir elementos visuais.	sim
2.1.1	Teclado	Todas as funcionalidades devem ser acionadas via teclado, com exceção se a funcionalidade não possibilite o controle apenas por teclado.	sim
2.1.2	Sem bloqueio de teclado	Ao se interagir via teclado, a navegação por todos os elementos "clicáveis" deve ocorrer sem que haja bloqueios ou interrupções.	sim
2.4.1	Ignorar blocos	Deve ser fornecido um tipo de controle para que as pessoas possam ignorar determinados conteúdos repetitivos e assim continuar com a navegação.	sim
2.4.2	Página com título	Todas as telas devem ter um título principal e que descreva claramente a sua finalidade.	sim
2.4.3	Ordem do foco	A interação por elementos focáveis na tela sempre deverá ser sequencial e lógica de acordo com o conteúdo apresentado.	sim
2.4.4	Finalidade do link (em contexto)	A finalidade de um link deve ser determinada a partir do texto do próprio link ou a partir do contexto no entorno deste link.	sim
2.4.7	Foco visível	Ao se interagir por teclado, qualquer pessoa deve conseguir identificar qual é a sua localização espacial na tela através de um foco visível identificador de sua localização.	sim
2.5.2	Cancelamento de acionamento	Deve se fornecer um modo de cancelar acionamentos feitos de forma não proposital.	sim
2.5.3	Rótulo no Nome acessível	Rótulos em botões, ícones acionáveis ou qualquer controle interativo, devem ter uma descrição significativa.	sim
3.1.1	Idioma da página	Declarar adequadamente o idioma da tela faz com que leitores de telas utilizem uma entonação correta para citar conteúdos. Sempre os declare.	sim
3.2.1	Em foco	O foco sempre deve se manter durante a navegação, sempre evitar mudança contextual que possa desorientar alguém.	sim
3.3.1	Identificação do erro	Sempre que uma mensagem de erro for exibida, ela deve identificar claramente qual é o elemento que gerou o erro de forma visual e audível.	sim
3.3.2	Rótulos e instruções	Todos os rótulos devem descrever claramente e sem ambiguidades a finalidade dos campos de formulário.	sim
4.1.1	Análise	Deve ser fornecido código semanticamente correto e sem erros significativos.	sim
4.1.2	Nome, função, valor	Toda tecnologia assistiva faz uso das propriedades de nome, função e valor para identificar adequadamente os elementos padronizados do HTML. Qualquer componente customizado deve trazer também essas marcações de forma adequada.	sim

## A.2 Nível AA

Quadro 4: Diretrizes do Nível AA

Número	Nome	Descrição	Aplicação no projeto
1.3.5	Identificar o objetivo de entrada	Deve ser claro para as pessoas o que deve ser preenchido em campos de formulários.	sim
1.4.3	Contraste (mínimo)	Textos devem ter uma relação de contraste entre primeiro e segundo plano de ao menos 4.5:1.	sim
1.4.4	Redimensionar texto	Ao se aplicar zoom de até 200 % na tela, deve ocorrer a responsividade dos textos apresentados de forma que sua leitura e legibilidade continuem adequados sem qualquer quebra na apresentação das informações.	sim
1.4.11	Contraste Não-Textual	Componentes de interface e imagens essenciais para o entendimento do conteúdo devem ter uma relação de contraste entre primeiro e segundo plano de ao menos 3:1.	sim
1.4.12	Espaçamento de texto	Sempre que houver um redimensionamento os textos não devem perder legibilidade.	sim
1.4.13	Conteúdo em foco por mouse ou teclado	Conteúdos adicionais não devem ser acionados apenas com foco por mouse ou teclado.	sim
2.4.5	Várias formas	Deve ser fornecido mais de uma forma de as pessoas encontrarem um determinado conteúdo.	sim
2.4.6	Cabeçalhos e rótulos	Todos os títulos e rótulo devem descrever claramente a finalidade dos conteúdos, não deve haver ambiguidade em seu entendimento.	sim
3.1.2	Idioma das partes	O idioma de uma determinada palavra ou frase contendo idioma diferente do original da tela, deve ser definido e corretamente identificado para que também ocorra uma correta entonação e pronúncia adequada via leitores de tela.	sim
3.2.3	Navegação consistente	Deve-se manter a consistência com relação ao formato de apresentação, interação e localização na tela.	sim
3.2.4	Identificação consistente	Deve-se manter a consistência com relação a diferentes formatos de elementos, mas que possuem uma mesma funcionalidade.	não
3.3.3	Sugestão de erro	Sempre que uma mensagem de erro for exibida, ela deve também dar dicas de como resolver o erro.	sim
3.3.4	Prevenção de erro (legal, financeiro, dados)	Deve ser fornecida uma forma de confirmação de dados ou a possibilidade de cancelamento do envio, sempre que campos de formulários exigirem o preenchimento de dados que envolvam responsabilidade jurídica, financeira ou contenham dados sensíveis.	sim
4.1.3	Mensagens de status	Qualquer tipo de mensagem que é resultado de uma ação ou que informa o andamento de um processo e que seja relevante para a pessoa, deve ser transmitida sem que ocorra uma mudança de foco na tela.	sim

## A.3 Nível AAA

Quadro 5: Diretrizes do Nível AAA

Número	Nome	Descrição	Aplicação no projeto
1.4.6	Contraste (melhorado)	Textos devem ter uma relação de contraste entre primeiro e segundo plano de ao menos 7:1.	sim
2.1.3	Teclado (sem exceção)	Todas as funcionalidades devem ser acionadas via teclado, sem exceção.	sim
2.2.3	Sem limite de tempo	Nenhuma funcionalidade em tela deve possuir algum tipo de execução mediante o cumprimento em um determinado período de tempo.	não
2.2.4	Interrupções	Deve ser possível adiar ou desligar qualquer tipo de interrupção acionada no sistema.	não
2.2.5	Nova autenticação	Quando uma sessão autenticada expira, o usuário deve ser capaz de continuar sua atividade sem que haja perda de dados até que seja feita uma nova autenticação.	sim
2.4.9	Finalidade do link (apenas link)	A finalidade de um link deve ser determinada a partir do texto do próprio link.	sim
2.4.10	Cabeçalhos da seção	Sempre que o conteúdo da tela for dividido em sessões, todas devem possuir títulos claros, com níveis de hierarquia bem definidos, facilitando a identificação das áreas.	sim
2.4.8	Localização	Qualquer pessoa deve conseguir se localizar ou se orientar facilmente em qualquer tela.	sim
2.5.5	Tamanho da área clicável	O tamanho das áreas acionáveis por clique ou toque devem possuir no mínimo 44x44 pixels de espaçamento, a não ser quando essa área esteja em uma frase localizada em um bloco de texto.	sim
3.1.3	Palavras incomuns	O uso de gírias, jargões, metáforas e figuras de linguagem pode ser um empecilho para a compreensão da informação, nesse sentido deve-se fornecer uma forma de tradução ou explicação da informação.	não
3.1.4	Abreviações	Nem sempre uma abreviação ou um acrônimo é compreensível por todas as pessoas, nesse sentido deve-se fornecer uma forma de identificação de seu significado real.	não
3.1.6	Pronúncia	Palavras regionais específicas e nomes próprios costumam ter pronúncias também específicas. Deve ser fornecida uma forma de possibilitar a correta compreensão da pronúncia em alguns casos.	não
3.2.5	Alteração a pedido	Qualquer mudança de contexto que possa desorientar as pessoas, deve ocorrer apenas quando solicitada pela pessoa que está utilizando.	sim
3.3.6	Prevenção de erro (todos)	Deve ser fornecida uma forma de confirmação de dados ou a possibilidade de cancelamento do envio, sempre que campos de formulários exigirem o preenchimento de dados.	sim