



Universidade Federal de Pernambuco

Departamento de Ciência da Computação

Curso de Ciência da Computação

**Conformidade da biblioteca Material UI para React com a Web
Accessibility Content Guidelines**

Trabalho de Conclusão de Curso de Graduação

por

Italo Ribeiro Soares

Orientador: Prof. Kiev Santos da Gama

Recife, Dezembro / 2022

Italo Ribeiro Soares

**Conformidade da biblioteca Material UI para React com a Web Accessibility
Content Guidelines**

Monografia apresentada ao Curso de Ciência da Computação, como requisito parcial para a obtenção do Título de , Centro de Informática.

Orientador: Prof. Kiev Santos da Gama

Recife

2022

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Soares, Italo Ribeiro.

Conformidade da biblioteca material UI para react com a web accessibility
content guidelines / Italo Ribeiro Soares. - Recife, 2022.
44p, tab.

Orientador(a): Kiev Santos da Gama

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de
Pernambuco, Centro de Informática, Ciências da Computação - Bacharelado,
2022.

Inclui referências, apêndices.

1. Acessibilidade. 2. Web. 3. ARIA. 4. WCAG. 5. WAI. I. Gama, Kiev
Santos da. (Orientação). II. Título.

000 CDD (22.ed.)

ITALO RIBEIRO SOARES

**CONFORMIDADE DA BIBLIOTECA MATERIAL UI PARA
REACT COM A WEB ACCESSIBILITY CONTENT
GUIDELINES**

Trabalho de Conclusão de Curso apresentado à
Coordenação do Curso de Bacharelado em
Ciência da Computação da Universidade
Federal de Pernambuco, como parte dos
requisitos à obtenção do grau de Bacharel em
Ciência da Computação.

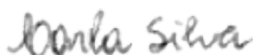
Data de Aprovação: 07/11/2022

Nota: 8,5

BANCA EXAMINADORA:



Prof. Dr. KIEV SANTOS DA GAMA (Orientador)
Centro de Informática - UFPE



Profa. Dra CARLA TACIANA LIMA LOURENÇO SILVA (1º Titular)
Centro de Informática - UFPE

RECIFE
2022

Agradecimentos

Agradeço, primeiramente ao meu orientador Prof. Kiev Santos da Gama por seus conselhos durante o desenvolvimento deste trabalho.

Agradeço também a minha esposa Wal, pelo seu suporte incondicional e a minha psicóloga Mirella por terem ajudado a manter minha ansiedade em xeque durante o processo de balancear trabalho e TCC.

Por fim, aos meus amigos Frodo, Jão e Juliano, afinal de contas, são minha família escolhida.

*Se soubéssemos o que era aquilo que estávamos
fazendo, não seria chamado de pesquisa.*

Albert Einstein

RESUMO

Cerca de 15% da população mundial tem algum tipo de deficiência. Este fato demonstra a relevância da inclusão digital para a participação dessas pessoas em todos os setores da sociedade. Neste escopo, existem trabalhos que buscam avaliar e medir o nível de acessibilidade das mais diversas tecnologias. Entretanto, tais abordagens são muito generalistas. Diante deste contexto, propomos avaliar uma tecnologia específica, a biblioteca Material UI, frente às diretrizes de acessibilidade web. Realizamos um estudo de acessibilidade usando a abordagem objetivo-pergunta-métrica e encontramos diversas violações de acessibilidade que culminaram na sugestão de melhorias à tecnologia em questão.

Palavras-chave: acessibilidade, WCAG, ARIA, material UI

ABSTRACT

About 15% of the world population has some form of disability. This fact demonstrates the relevance of digital inclusion for the participation of these people in all sectors of society. In such scope, there are many research papers that seek to evaluate and measure the level of accessibility of the most diverse digital technologies. However, such approaches are very generalistic. In this context, we propose to evaluate a specific technology, the Material UI library, against web accessibility guidelines. We carried out an accessibility study using the objective-question-metric approach and found several accessibility violations that culminated in the suggestion of improvements to the technology in question.

Keywords: material UI, accessibility, WCAG, ARIA

LISTA DE TABELAS

Tabela 1	Modelo GQM.	19
Tabela 2	Violações encontradas nos testes automatizados.	23
Tabela 3	Modelo de apresentação de uma violação.	24
Tabela 4	Violação Séria 1.	24
Tabela 5	Violação Séria 2.	24
Tabela 6	Violação Crítica 1.	25
Tabela 7	Violação Crítica 2.	25
Tabela 8	Caso de Teste Acordeão	27
Tabela 9	Resultados de testes manuais	28
Tabela 10	Problemas encontrados durante testes exploratórios com leitor de tela NVDA	29
Tabela 11	Caso de Teste Acordeão	40
Tabela 12	Caso de Teste Diálogo de Alerta	40
Tabela 13	Caso de Teste Checkbox	40
Tabela 14	Caso de Teste Radio Group	41
Tabela 15	Caso de Teste Slider	41
Tabela 16	Caso de Teste Botão	42
Tabela 17	Caso de Teste Autocomplete	42
Tabela 18	Caso de Teste Lista Aninhada	42
Tabela 19	Caso de Teste Seleção	42
Tabela 20	Caso de Teste Menu	43
Tabela 21	Caso de Teste Botão Menu	43
Tabela 22	Caso de Teste Link	44
Tabela 23	Caso de Teste Switch	44
Tabela 24	Caso de Teste Aba	44

LISTA DE SIGLAS

CS	Critério de Sucesso
CT	Critério de Teste
UFPE	Universidade Federal de Pernambuco
MUI	Material UI
Q1	Pergunta 1
Q2	Pergunta 2
M1	Métrica 1
M2	Métrica 2
W3C	<i>World Wide Web Consortium</i>
WAI	<i>Web Accessibility Initiative</i>
ARIA	<i>Accessible Rich Internet Applications</i>
WCAG	<i>Web Content Accessibility Guidelines</i>
PCD	Pessoa Com Deficiência
CRPD	<i>The UN Convention on the Rights of Persons with Disabilities</i>
GQM	<i>Goal Question Metric</i>

SUMÁRIO

1	INTRODUÇÃO	10
1.1	Motivação: O Problema e suas limitações	10
1.2	Relevância e abrangência do problema.....	11
1.3	Contextualização	12
2	FUNDAMENTAÇÃO TEÓRICA	14
2.1	WCAG.....	14
2.2	WAI-ARIA	15
2.3	Material UI.....	15
2.4	Testagem de Acessibilidade	16
3	METODOLOGIA DE TRABALHO ACADÊMICO	18
3.1	A abordagem objetivo-pergunta-métrica	18
3.2	Aplicação do modelo GQM a este trabalho	18
4	METODOLOGIA DE TESTES.....	20
4.1	Página de Testes	20
4.2	Roteiro de Testes.....	21
5	RESULTADOS E DISCUSSÕES	23
5.1	Resultados dos Testes Automatizados.....	23
5.1.1	Violações Sérias	24
5.1.2	Violações Críticas	24
5.1.3	Violações Que Requerem Revisão	25
5.2	Casos de Teste Manuais	25
5.2.1	Resultados dos Testes Manuais	27
5.3	Resultados dos Testes Exploratórios com Leitor de Tela	29
6	ANÁLISE E DISCUSSÃO	30
6.1	Pergunta de Pesquisa 1	30
6.2	Pergunta de Pesquisa 2	31
6.3	Discussão.....	33

7	CONCLUSÕES	34
7.1	Aplicações, Limitações e Trabalhos Futuros	35
8	APÊNDICE A - TESTES MANUAIS REALIZADOS	40

1 INTRODUÇÃO

1.1 Motivação: O Problema e suas limitações

Deficiência e pobreza tem uma ligação bidirecional. PCDs e suas famílias têm chances maiores de experimentar dificuldades sociais e econômicas, da mesma forma que o surgimento de deficiências na família pode piorar as mesmas condições sócio-econômicas através de vários canais, e.g. impactos na educação, emprego e acesso à saúde. [1]

Crianças com deficiência têm menores chances de frequentar a escola. [2] [3] PCDs têm maiores chances de serem desempregadas e de ganhar menos quando empregados. [4] Existe inclusive uma correlação negativa entre a severidade da deficiência e renda.

Embora o acesso à tecnologia não possa resolver todos esses problemas, a inclusão em um mundo cada vez mais digital e informatizado é um catalisador de desenvolvimento e inclusão. Entretanto, a maior parte dos serviços e ferramentas disponíveis não têm acessibilidade como prioridade e, conseqüentemente, acrescentam aos problemas de inclusão e menor status socioeconômico de PCDs.

Desde a publicação da CRPD em 2006 [5], vários projetos têm sido desenvolvidos para promover a acessibilidade em diversas áreas. Esses projetos incluem iniciativas governamentais e organizações privadas, com focos específicos ou gerais. Entre as iniciativas de cunho privado existem a Ceweb.br [6] o Centro de Estudos sobre Tecnologias Web e o A11Y Project [7]. Ambos visam pesquisar e promover a adoção de políticas de acessibilidade em ambientes tecnológicos.

Além do empurrão privado para maior acessibilidade tecnológica, também existem iniciativas governamentais para promovê-la. No Brasil, temos a lei de acessibilidade que estabeleceu o Estatuto da Pessoa com Deficiência de 2016 [8] e a Lei de Acesso à Informação de 2011 [9] que, apesar de não ser focada em acessibilidade, trouxe inovações e aparatos regulatórios para a promoção da acessibilidade em contextos de acesso à informações sobre ações governamentais. A União Europeia aprovou em 2016 a Diretiva de Acessibilidade Web [10] que aumenta requerimentos de acessibilidade para o setor público.

Por mais que esforços de escopo abrangente existam por parte de organizações governamentais e privadas existam, falta pesquisa de cunho mais específico, focando em técnicas e ferramentas usadas para a construção da web moderna, como frameworks e bibliotecas de UI e javascript. Tal falta de dados e resultados é problemática por várias razões:

tais ferramentas são usadas para construir centenas de milhares de páginas e aplicações, essas aplicações são consumidas por diversos aparelhos nas mais variadas circunstâncias, o que aumenta potencial de falhas, componentes de interface podem ser tornados mais acessíveis de várias formas, o que leva ao questionamento se devem existir diretrizes ou regras para garantir ou expandir a acessibilidade desses componentes e, por fim, a multiplicidade de técnicas e ferramentas combinadas à ausência de dados sobre sua acessibilidade leva à dificuldade de testagem automatizada de acessibilidade e faz necessário testagem manual que, por sua natureza, é sujeita a erros e interpretações.

O objetivo deste trabalho é examinar o quão e como uma ferramenta específica, a MUI, conforma ou não com diretrizes já existentes de acessibilidade para determinar a suficiência ou não de testagem automática e também fazer sugestões sobre seu uso de forma a melhorar a acessibilidade de aplicações construídas com esta ferramenta. Este exame se dará via análise automatizada, um exame manual das práticas autorais ARIA [11] e uma análise via programa assistivo de leitura de tela.

1.2 Relevância e abrangência do problema

Visto que simplesmente definir deficiência já é um problema em si, medi-la é ainda mais complicado. Dada sua dimensão multifacetada e dinâmica, que muda de acordo com as interações com o ambiente, pesquisas feitas com o mesmo grupo e tempo geram resultados distintos baseado no propósito dos dados, no conceito adotado de deficiência, nos quesitos focais e aspectos examinados. Dados coletados para direcionamento de políticas públicas de saúde, por exemplo, podem refletir resultados muito diferentes de uma pesquisa feita com o intuito de informar a elaboração de uma nova lei antidiscriminação.

Além disso, o paradigma usado pela maioria das contagens e pesquisas coloca tipos de deficiência como um conjunto de categorias discretas que, com frequência, são definidas com base em critérios puramente médicos. Ignorando aspectos sociais, econômicos e ambientais. Essas definições com frequência excluem pessoas que sofrem com limitações advindas de condições crônicas mas que são deixadas de fora de pesquisas sobre deficiência. Também existe a premissa tácita de que cada tipo de deficiência tem seus desafios particulares e que esses desafios diferem entre cada tipo de deficiência. Quando, na verdade, diferentes intervenções podem ser necessárias para pessoas com o mesmo tipo de deficiência ou o contrário, intervenções semelhantes para pessoas com tipos diferentes de

deficiência. [12]

Pelos dados da World Health Survey (WHS) da Organização Mundial da Saúde (OMS), existiam cerca de 650 milhões de pessoas com deficiência em 2004. [13] A WHS foi uma pesquisa feita no triênio 2002-2004 e era, até 2011, a maior pesquisa de saúde e deficiência multinacional que usa métodos consistentes e questionário único. 70 países foram examinados, as escolhas foram feitas levando em consideração critérios de renda, para obter cobertura abrangente e critérios de dados disponíveis, para obter algum tipo de dado em regiões sub representadas em pesquisas independentes, como a África sub saariana.

Já pelos dados da pesquisa Global Burden of Disease (GBD) do mesmo ano, existiam cerca de 978 milhões de pessoas com deficiência no mundo. [13] A GBD é uma pesquisa comissionada pelo Banco Mundial desde 1990 para avaliar o estado das doenças no mundo e como isso afeta populações e economias.

Segundo estimativas da WHS e GBD, hoje, devem existir mais de 1 bilhão de pessoas com deficiência no mundo, ou cerca de 15% da população mundial, maior que a estimativa de 10% feita pela OMS em 1970. [14]

1.3 Contextualização

Em um trabalho sobre acessibilidade, é útil começar com uma definição do que é considerado deficiência ou incapacidade. Como em muitas questões complexas, uma definição única e abrangente é esquivada e difícil de formular. Em diversos casos é possível encontrar casos extremos onde a definição se faz abrangente demais ou de menos. Incluindo pessoas que escapam ao estereótipo ou ideia que se tem do que é deficiência ou o contrário, deixando de fora pessoas que se encaixam em tais categorias. A interação com o ambiente e possíveis dispositivos de assistência é parte integral do modelo bio psicossocial. Sendo assim, uma pessoa, pode, por exemplo, ser considerada deficiente visual ao não conseguir ler o letreiro de um ônibus a uma certa distância, o que impacta em sua habilidade de funcionar no dia-a-dia. Mas não ser considerada deficiente ao realizar seu trabalho em computador, onde fontes podem ser ajustadas de forma a acomodar a acuidade visual da pessoa em questão e fazendo com que esta funcione de maneira igual a seus pares sem alterações visuais.

Ao longo das últimas décadas, pesquisas nas áreas de ciências sociais e médicas [15]

têm contribuído para redefinir ou, retificar, a definição arcaica de deficiência meramente como função da biologia da pessoa com deficiência (PCD). Tal mudança de pensamento trouxe à tona o modelo social de deficiência que vê PCDs como pessoas desafiadas mais pela sociedade e meio que pela sua biologia.

Tal modelo, entretanto, é extremo demais e falha em reconhecer as claras circunstâncias que advém de suas condições médicas. Diante dessa dicotomia, a visão predominante na comunidade científica [15] e adotada também por grandes documentos de referência tais como a Classificação Internacional de Funcionalidade, Incapacidade e Saúde (ICF, na sigla em inglês) [16] e a Convenção sobre os Direitos das Pessoas com Deficiência (CRPD, na sigla em inglês) [5] é a do modelo biopsicossocial, onde deficiência é um termo genérico que compreende incapacidades, limitadores de atividade e restrições de participação referindo-se às interações negativas entre o contexto e o ambiente de uma PCD com uma condição médica que dificultem sua participação efetiva e completa na sociedade em igualdade com as outras.

Tal definição, além de estar em concordância com a visão científica contemporânea [15] tem como corolário que a melhoria da qualidade de vida de PCDs passa, necessariamente, não só por intervenções de natureza pessoal e individual mas também por endereçar o ambiente, o contexto e a sociedade nas quais essas PCDs estão inseridas. Isso inclui, logicamente, as tecnologias e ferramentas digitais usadas por essas sociedades para se comunicar, entreter, realizar transações e exercer a cidadania e as ferramentas que permitem a sua construção e funcionamento, que são o foco deste trabalho.

Nos capítulos a seguir, iremos determinar se a MUI é suficientemente acessível e sugerir, se for o caso, maneiras de tornar a MUI mais acessível através da metodologia GQM, que norteará a testagem de acessibilidade através de ferramenta automatizada, checagem manual de diretrizes de acessibilidade e verificação manual com leitor de tela e, por fim, a sugestão de melhorias baseadas nos resultados desses testes.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção, apresentaremos conceitos centrais para a compressão e desenvolvimento do trabalho. Apresentaremos e definiremos os principais conjuntos de diretrizes de acessibilidade web: WCAG e ARIA. Também traremos mais informações sobre o objeto de pesquisa, a biblioteca MUI e discutiremos as peculiaridades e pormenores dos métodos de testagem de acessibilidade.

2.1 WCAG

A *Web Content Accessibility Guidelines* (WCAG) é um conjunto de diretrizes e princípios de desenvolvimento publicados pelo *World Wide Web Consortium* (W3C). [17] A versão mais recente, a 2.0 foi publicada em 2008 e, desde então, foi estendida em 2018 com mais 17 critérios. Neste trabalho, usaremos a versão 2.1 como referência uma vez que incorpora todos os elementos da versão 2.0 e extensões.

A WCAG 2.1 é separada em 4 princípios norteadores: percepção, operação, compreensão e robustez. O primeiro princípio, percepção, diz respeito como usuários recebem informação e como conteúdo é representado para eles. O segundo princípio, operação, diz respeito à formas de interação dos usuários com o conteúdo da página e sobre meios alternativos de interagir com tal conteúdo. O terceiro princípio, compreensão visa garantir que usuários entendam informação, navegação e o funcionamento da página. O quarto e último princípio, robustez, diz respeito aos aspectos técnicos de compatibilidade de uma página com uma vasta gama de dispositivos de visualização, operação e assistência.

Cada princípio é, então, subdividido em critérios testáveis que precisam ser aprovados com sucesso para obter algum dos graus de conformidade. Os graus de conformidade são A (o mais baixo), AA (o intermediário) e AAA (o mais alto). O documento WCAG define que o grau de conformidade A é o considerado essencial e o mais básico, sem o qual uma página, mesmo com tecnologia assistiva, se torna não acessível. O grau de conformidade AA é considerado o grau padrão ao qual todos os desenvolvedores preocupados com o alcance de suas páginas à PCDs devem esperar atingir e fornece cobertura suficiente para garantir a acessibilidade do conteúdo. Já o terceiro e mais alto grau, AAA, é considerado um grau extra em que desenvolvedores podem atingir para poder obter um maior grau de acessibilidade e garantir total conformidade com as diretrizes da WCAG.

2.2 WAI-ARIA

Outro conjunto de diretrizes importantes para acessibilidade web é o *Rich Internet Applications* (ARIA) publicado pela *Web Accessibility Initiative* (WAI). [18] Mais conhecido como WAI-ARIA, é uma especificação independente da WCAG mas com frequência usada como extensão ou complemento devido a sua natureza focada em aplicações “ricas”. Aplicações ricas são definidas por Fraternali et al. de maneira genérica como um conjunto de soluções com o objetivo comum de adicionar novas capacidades à arquitetura convencional de hipertexto da web. Ou, de maneira mais específica, aplicações que se assemelham a programas e aplicações desktop e mobile completo com recursos avançados como multimídia, acesso à arquivos, comunicação assíncrona e processamento e renderização distribuída entre cliente e servidor. [19]

WAI-ARIA é complementar à especificação HTML e tem o objetivo de fornecer mais informação a dispositivos e programas assistivos, como leitores de tela. Contudo, é importante destacar que a especificação WAI-ARIA não substitui o uso correto de tags semânticas da especificação HTML e requer que tais tags sejam usadas corretamente para proporcionar acessibilidade completa. Por mais que atributos WAI-ARIA possam ser usados em qualquer tag HTML e prover algum nível de informação adicional, isso é considerado má prática [20] uma vez que ‘Nenhuma ARIA é melhor que ARIA ruim’ [21]. A ausência de técnicas WAI-ARIA significa que uma página é inacessível mas o uso incorreto significa que a página pode conter layout, organização e informações errôneas que, além de dificultar, podem induzir o usuário ao erro e ao executar tarefas não desejadas.

2.3 Material UI

O framework de interface Material UI foi criado em 2014 com licença open source para tornar o design, coincidentemente, mais acessível. Entretanto, a acessibilidade que os criadores têm em mente não se trata da acessibilidade de uso dos produtos construídos com esta tecnologia, mas sim da construção desses produtos em si. Os criadores desejam fazer com que qualquer um possa construir aplicações web e móveis de maneira fácil, atrativas esteticamente e responsivas, mas sem extenso treinamento em design ou UX.

Desde então mais de 2 milhões de desenvolvedores e grandes empresas e organizações governamentais como Spotify, Amazon, Netflix e NASA usaram a tecnologia

criada por mais de 2400 desenvolvedores e contribuintes diversos. O framework foi escolhido para este trabalho justamente pelo seu caráter open source e popularidade como ferramenta, o que dá aos resultados grande abrangência. [22]. Usar um framework já popular e construir funcionalidades de acessibilidade sobre ele trás benefícios de adoção uma vez que um framework já maduro e popular é capaz de aumentar o nível de acessibilidade das páginas que o utilizam mais rapidamente que construir um framework exclusivamente focado em acessibilidade e convencer toda uma comunidade de usuários a adotar uma nova ferramenta.

2.4 Testagem de Acessibilidade

Existem 2 grandes grupos de práticas de testagem de acessibilidade. Pode ser manual ou automatizado. Ambas possuem suas vantagens e desvantagens. Testes automáticos são melhores que testes manuais no que diz respeito ao determinismo, constância e velocidade. Um teste automático sempre irá retornar o mesmo resultado se executado na mesma página e com os mesmos parâmetros. Além disso, testes automáticos são capazes de examinar grandes grupos de páginas em intervalos curtos e produzir relatórios em intervalos igualmente curtos.

Entretanto, ferramentas automáticas são muito menos exatas que testadores manuais. As melhores obtendo, no máximo, 50% de cobertura dos critérios WCAG. [23] Além disso, quanto maior a cobertura, maior o número de falsos positivos encontrados, o que pode fazer com que ferramentas automáticas se mostrem contraproducentes naquilo que se propõem a fazer: analisar rapidamente sites em busca de problemas de acessibilidade uma vez que podem requerer muita interação humana para checar ou corrigir falsos positivos e/ou negativos [23].

Ferramentas automáticas também não são uniformes. Ferramentas diferentes possuem forças e fraquezas. Algumas ferramentas podem obter altos índices de cobertura e corretude em critérios de percepção mas baixos nos outros 3 critérios. Por outro lado, o inverso também ocorre, em que uma ferramenta produz bons resultados em critério de robustez mas não nos outros. Sendo assim, para obter uma boa cobertura e corretude são necessárias usar múltiplas ferramentas. O que acarreta custos maiores, processos mais complicados e revisões mais laboriosas.

Já no campo da testagem manual, existe o problema da sujeição a erros, demora

e custos associados, especialmente quando se considera que são necessários vários analistas com conhecimento de área para a produção de resultados confiáveis. Por exemplo, num estudo de 2011, Brajnik et al chegaram ao número de 3 analistas profissionais para obter resultados confiáveis [24]. Diante dessas considerações, fica claro que definir a melhor abordagem de testes de acessibilidade não é tarefa trivial e deve ser considerada cuidadosamente por empresas e instituições que desejam produzir sites acessíveis.

3 METODOLOGIA DE TRABALHO ACADÊMICO

3.1 A abordagem objetivo-pergunta-métrica

A abordagem GQM (*Goal Question Metric* na sigla em inglês) é uma abordagem de mensuração e refinamento de produtos e processos originalmente criada no Centro de Voo Espacial Goddard da NASA [25]. A abordagem tem como premissa que para que uma organização possa medir indicadores operacionais é necessário, primeiro, a definição de objetivos e então a conexão desses objetivos com as métricas necessárias para definir esses objetivos.

A abordagem pode ser dividida em 3 níveis da seguinte maneira:

1. **Conceitual** (GOAL ou OBJETIVO)

Um objetivo tem parâmetros: um objeto, um ponto de vista e um problema. Um objeto pode ser um produto, um processo ou recursos. O ponto de vista precisa ser definido de forma relevante para a solução do problema e questão e o problema precisa ser central para a melhoria do objeto a ser analisado.

2. **Operacional** (QUESTION ou PERGUNTA)

No critério operacional, perguntas devem caracterizar o produto com respeito ao ponto de vista e problema selecionados na etapa anterior.

3. **Quantitativo** (METRIC ou MÉTRICA)

Um conjunto de dados associados a cada pergunta com o objetivo de responde-las de maneira quantitativa. É importante destacar que os dados podem ser de natureza objetiva ou subjetiva e que o mesmo conjunto de métricas pode ser usado para responder perguntas diferentes.

3.2 Aplicação do modelo GQM a este trabalho

O modelo GQM aplicado aos objetivos e métricas deste trabalho é o que segue:

Tabela 1: Modelo GQM.

Obrjetivo	Propósito	Avaliar
	Problema	o o quão conformante é
	Objetivo	a biblioteca de componentes React Material UI
	Ponto de Vista	sob o ponto de vista do usuário
Pergunta	Q1	Quanto MUI acata os princípios de acessibilidade?
Métrica	M1	Razão de problemas de acessibilidade encontrados pela ferramenta de testes automatizados Axe sobre o total de componentes utilizados.
	M2	Razão de problemas de acessibilidade encontrados pela verificação de casos de teste manuais sobre o total de componentes utilizados.
Pergunta	Q2	Como tornar as aplicações que empregam a MUI mais acessíveis?
Métrica	M3	Métricas M1 e M2

4 METODOLOGIA DE TESTES

4.1 Página de Testes

Para testar as características de acessibilidade da MUI, foi criado um site com todos os componentes visíveis da biblioteca. Sendo estes:

- Autocomplete
- Button
- Button Group
- Checkbox
- Floating Action Button
- Radio Group
- Rating
- Select
- Slider
- Switch
- Text Field
- Transfer List
- Toggle Button
- Avatar
- Badge
- Chip
- List
- Table
- Alert
- Backdrop
- Dialog
- Progress
- Skeleton
- SnackBar
- Accordion
- App Bar
- Card
- Paper
- Bottom Navigation
- Breadcrumbs
- Drawer
- Link
- Pagination
- Speed Dial
- Stepper
- Tab

Componentes invisíveis, de layout, foram excluídos pois a acessibilidade desses componentes é altamente dependente da implementação realizada pelo desenvolvedor em questão. Não sendo muito representativo da acessibilidade da biblioteca a ser testada. O site foi criado de maneira extremamente simples, sem quaisquer adições de livrarias ou código adicional, apenas o básico do React e da MUI. Os componentes foram colocados na página seguindo a ordem natural de organização do HTML para componentes block ou inline, salvo em casos onde alterações simples de layout foram utilizadas para fins didáticos de manter todas as partes de um determinado componente com tamanhos adequados para visualização. A maioria dos componentes foi implementada da maneira mais simples possível, apenas com apenas os argumentos meramente essenciais para seu funcionamento.

4.2 Roteiro de Testes

Existem muitas potenciais ferramentas para avaliar a acessibilidade de páginas. Entretanto nenhuma delas atinge um nível maior que 38% de completude [23] definida por Vigo et al. como o número de positivos verdadeiras encontrados pelas ferramentas e, embora possam atingir cerca de 55% de completude ao combinar diversas ferramentas de acordo com suas vantagens e desvantagens, esse percentual ainda é baixo se comparado a testes realizados com profissionais humanos. O estudo argumenta, entretanto, que ferramentas automatizadas têm o seu valor visto que testagem realizada manualmente é mais demorada, cara e sujeita a erros. Diante disso, achamos o uso de pelo menos uma ferramenta é essencial para demonstrar, se não a cobertura das diretrizes WCAG 2.1 por parte da MUI, pelo menos o cenário esperado por desenvolvedores ao recorrerem a tais ferramentas.

Escolhemos a ferramenta Axe [26] por ter sido a segunda ferramenta com maior cobertura no trabalho de Vigo et al. [23] com cerca de 42%. Sendo cobertura uma métrica definida por quantos verdadeiros positivos uma ferramenta de testes obtém. Escolhemos a segunda ferramenta com mais cobertura e não a primeira pois a primeira ferramenta, TAW [27], pois está consideravelmente defasada, suportando apenas a versão 1.0 da WCAG publicada em 1999 [27] [28].

Diante da insuficiência de testes automatizados para atestar a acessibilidade de uma página, também serão realizados testes manuais e uma revisão do código da página de acordo com diretrizes WCAG e WAI-ARIA. Os testes manuais seguirão a seguinte

ordem:

1. Verificação da conformidade de itens conforme os CSs aplicáveis listados nas diretrizes WCAG e WAI-ARIA
2. Testagem exploratória com objetivo de verificar a conformidade da página de testes com dispositivos assistivos. Note que aqui usamos 'dispositivos assistivos' como qualquer tipo de artifício que assista usuários na navegação, percepção e compreensão da página. Seja *hardware* ou *software*. Usaremos o leitor de tela NVDA [29] e navegação via teclado.

Quanto as opções e parâmetros dos testes automatizados, estes serão os seguintes:

1. Linguagem: padrão do navegador utilizado (inglês)
2. Nível WCAG: AA
3. Versão Axe-core: 4.x
4. Melhores Práticas: Desabilitado

A linguagem usada (inglês) foi usada por ser uma linguagem universal e ser a padrão da ferramenta e do navegador. O nível de conformidade AA foi usado em detrimento do nível AAA por ser menos crucial, porque a maioria das ferramentas sequer suporta o nível AAA e o W3C-WAI explicitamente avisa sobre a impossibilidade de satisfazer os critérios AAA para alguns conteúdos. [30]. A versão do axe-core diz respeito à última versão da plataforma usada. A opção de melhores práticas foi desabilitada por incluir regras e práticas que não necessariamente fazem parte das diretrizes WCAG, o que foge ao escopo deste trabalho.

5 RESULTADOS E DISCUSSÕES

5.1 Resultados dos Testes Automatizados

Os resultados da primeira execução estão compilados na seguinte tabela:

Tabela 2: Violações encontradas nos testes automatizados.

Tipo	Número	Porcentagem
Totais	54	100%
Requer Revisão	40	74.07%
Crítico	12	22.22%
Sério	2	3.70%
Moderado	0	0%
Menor	0	0%

Os possíveis tipos de violação são:

- **Requer Revisão**
- **Crítico**
- **Sério**
- **Moderado**
- **Menor**

As violações intituladas 'Requer Revisão' são violações as quais a ferramenta não tem certeza sobre o resultado e requer assistência humana para confirmar ou não a violação. O que corrobora os resultados encontrados por Vigo et al. [23] sobre a insuficiência das ferramentas e é um obstáculo à ferramentas de refatoração que prometem melhorar automaticamente a acessibilidade de páginas como a Automatica11y [31]

É importante frisar que as violações não necessariamente são únicas, um mesmo tipo de violação pode ser contabilizado múltiplas vezes.

As violações serão listadas nas próximas seções, agrupadas por severidade e observando o formato da tabela abaixo. Quando uma violação afetar múltiplos componentes, os componentes serão mostrados abaixo da tabela detalhando a violação.

Tabela 3: Modelo de apresentação de uma violação.

Componente	Nome do componente afetado
Problema	Problema apontado pelo Axe
Descrição	Detalhes adicionais providos pelo Axe
Captura de Tela do Componente	Uma captura de tela do componente em questão para facilitar visualização

5.1.1 Violações Sérias

Tabela 4: Violação Séria 1.



Componente	Barra de progresso circular
Problema	O componente barra de progresso deve ter um nome acessível
Descrição	Garante que todos os elementos barra de progresso tenham um nome acessível
Captura de Tela do Componente	

Tabela 5: Violação Séria 2.

Componente	Avatar Chip
Problema	Os elementos devem ter contraste de cor suficiente
Descrição	Garante que o contraste entre as cores do primeiro plano e plano de fundo atende a razão indicada na WCAG 2 AA
Captura de Tela do Componente	

5.1.2 Violações Críticas

Existem 12 violações críticas totais, mas apenas 2 tipos únicos de violações. Sempre que uma violação ocorrer múltiplas vezes em um componente, este terá o número de violações em parênteses. As capturas de tela serão todas agrupadas na célula apropriada.

Tabela 6: Violação Crítica 1.

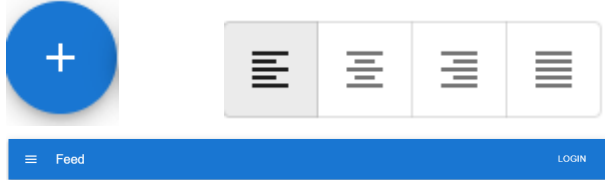

Componente	Botão de Ação Flutuante, Botões de Alternância (x4), Barra de Aplicativo
Problema	Botões devem ter texto discernível
Descrição	Garante que botões têm texto discernível
Captura de Tela do Componente	

Tabela 7: Violação Crítica 2.

Componente	Caixa de Seleção (x4), Controle Deslizante, Interruptor
Problema	Elementos de formulário devem ter rótulos
Descrição	Garante que cada elemento de formulário tenha um rótulo
Captura de Tela do Componente	

5.1.3 Violações Que Requerem Revisão

Todos os problemas apontados foram por falta de contraste entre cores, em um grande número de componentes, 40. Entretanto, a ferramenta Axe [26] não pôde determinar com exatidão esses problemas por conta de problemas com sobreposição de elementos na página. Em outra seara, a ferramenta parece ter problemas com componentes que têm sua visibilidade condicionada a algum estado. Como por exemplo, menus colapsáveis. Não detectando esses componentes sejam eles visíveis ou não no momento da testagem.

5.2 Casos de Teste Manuais

Os testes manuais foram realizados usando as práticas autorais ARIA [11] que contem convenções fundamentais relacionadas à navegação com teclado e manipulação de foco em elementos da página. Entretanto, se um determinado componente não está

incluso nas práticas autorais, usaremos os critérios fundamentais descritos nas diretrizes WAI-ARIA. É importante destacar que as práticas autorais WAI-ARIA diferenciam componentes por função e não por aparência. Logo, vários componentes na página de testes podem obedecer os mesmos critérios de acessibilidade. As tabelas no apêndice compilam os testes realizados. Contém o ou os componentes em questão, critérios de teste, descrições e resultados esperados. Listamos aqui todos os componentes testados e mostramos uma das tabelas disponíveis no apêndice como exemplo.

- Acordeão
- Diálogo de Alerta
- Checkbox
- Radio Group
- Slider
- Botão
- Autocomplete
- Lista Aninhada
- Seleção
- Menu
- Botão Menu
- Link
- Switch
- Aba

Tabela 8: Caso de Teste Acordeão

Componente	CT	Descrição	Comportamento Esperado
Acordeão	1.1	Checar suporte a navegação com TAB	Move o foco para o próximo painel disponível
	1.2	Checar se ENTER ou ESPAÇO podem ser usados para expandir e colapsar painéis	ENTER e ESPAÇO devem estender painéis colapsados e esconder painéis estendidos
	1.3	Checar suporte a navegação com SHIFT + TAB	Move o foco para o painel disponível anterior

5.2.1 Resultados dos Testes Manuais

Tabela 9: Resultados de testes manuais

Componente	CT	Aprovado
Acordeão	1.1	S
	1.2	S
	1.3	S
Diálogo de Alerta	2.1	S
	2.2	S
	2.3	S
Checkbox	3.1	S
Radio Group	4.1	S
	4.2	S
	4.3	S
Rating	5.1	S
	5.2	N
	5.3	N
	5.4	S
	5.5	N
	5.6	N
	5.6	N
Slider	6.1	S
	6.2	S
	6.3	S
	6.4	S
	6.5	N
	6.6	N
	6.6	N
Botão	7.1	S
	7.2	N
Botão Flutuante	8.1	S
	8.2	N
Chip	9.1	N
	9.2	N
Autocomplete	10.1	S
	10.2	S
	10.3	S
Lista Aninhada	11.1	S
	11.2	S
Seleção	12.1	N
	12.2	S
	12.3	S
	12.4	S
Menu Inferior	13.1	N
	13.2	N
	13.3	S
	13.4	N
	13.5	N
	13.6	N
	13.7	N
	13.8	N
	13.9	N
	13.10	N

Componente	CT	Aprovado
Drawer	14.1	N
	14.2	N
	14.3	S
	14.4	N
	14.5	N
	14.6	N
	14.7	N
	14.8	N
	14.9	N
	14.10	S
Speed Dial	15.1	S
	15.2	S
	15.3	S
	15.4	S
	15.5	S
	15.6	S
	15.7	S
	15.8	N
	15.9	N
	15.10	S
Botão Menu	16.1	S
	16.2	S
Link	17.1	N
Switch	18.1	S
Aba	19.1	S
	19.2	S
	19.3	S
	19.4	S
	19.5	S

5.3 Resultados dos Testes Exploratórios com Leitor de Tela

Esta seção apresenta problemas encontrados com nosso teste exploratório com o leitor de tela NVDA. Os problemas relacionados a ARIA e seus atributos foram omitidos para evitar redundância, uma vez que já foram detectados pela ferramenta Axe [26] nos testes automatizados.

Tabela 10: Problemas encontrados durante testes exploratórios com leitor de tela NVDA

Componente	Problema Encontrado
Checkbox	NVDA não lê checkbox desabilitada
Fab	NVDA lê apenas como 'botão'. Não lê ícone usado.
Star Rating	Apresenta-se como radio group com 6 estados, o que é tecnicamente verdade mas induz ao erro do usuário.
Select	Não lê informação de título
Toggle Button	Não há título disponível. Apenas um estado é lido.
Badges e Icons	São completamente ignorados pelo leitor de tela. Para usuários com deficiência visual, são como se não existissem.
Chip	Botão de ação não é focalizável e não há retorno audível ao passar o cursor sobre.
Lista Aninhada	É perdida informação de hierarquia e relação entre itens. Apenas os nomes são lidos.
Alerts	Não há informação contextual de severidade. Apenas título.
Backdrop	Completamente ignorado.
Circular Progress	Completamente ignorado.
Skeleton	Completamente ignorado.
Papercard	Título visível mas sem informação de elevação.
Breadcrumbs	Sem informação de hierarquia ou ordem.
Speed Dial	O botão em si é focalizável e interativo. Mas os sub-itens são completamente ignorados.
Horizontal Linear Stepper	Botões são focalizáveis mas texto é ignorado.

6 ANÁLISE E DISCUSSÃO

Esta seção visa responder as perguntas elencadas na tabela 1 através do modelo GQM baseado nos dados coletados na seção 5.

6.1 Pergunta de Pesquisa 1

A pergunta Q1 diz respeito à conformidade da MUI com diretrizes de acessibilidade, especificamente aos critérios de nível AA das práticas autorais ARIA [11]. Foram encontrados mais de 50 problemas com a ferramenta automatizada Axe, mais de 30 problemas por busca manual e mais de 15 problemas relacionados a leitor de tela. Estes fatos, dentro dos limites deste trabalho, sugerem que a MUI não conforma, pelo menos não totalmente, às diretrizes de acessibilidade WCAG [17] e ARIA [18].

Começando pela métrica 1 (M1), a quantidade total de componentes usados foi 37. Foram contabilizados somente componentes únicos, variações de componentes não entraram no total. O total de problemas detectados pela ferramenta Axe foi 54. O que dá uma razão de 1.45 violação de acessibilidade por componente usado. Destes 54 entretanto, apenas 14 foram encontrados de forma "independente", sem a necessidade de intervenção humana para confirmar a violação, uma razão de 0.37 violação por componente.

Para contextualizar essas métricas, usaremos dados da última pesquisa *WebAIM Million* [32]. Um projeto do Instituto para Pesquisa, Políticas Públicas e Práticas em Deficiência da Universidade Estadual de Utah, nos Estados Unidos. O projeto faz uma revisão programática das páginas iniciais dos 1 milhão de sites mais populares da internet. Considerando que o site médio tem 955 elementos em sua construção, a nossa densidade de erros aponta que poderíamos esperar 1384.75 violações de acessibilidade num site construído com MUI e uso variado de componentes! Este é um número excessivamente alto de violações e demonstra a dificuldade de cumprir todas as requisições de acessibilidade em grandes projetos da indústria.

Quanto à densidade de erros, não foram encontradas discrepâncias na distribuição de violações entre componentes, mas sim entre tipos de violação. A violação mais comum foi, de longe, a violação de contraste insuficiente de texto com 40 ocorrências ou 74.07%. Esses resultados se mostram consistentes com dados encontrados na pesquisa *WebAIM* [32] onde 83.9% das páginas pesquisadas apresentaram problemas com contraste de texto.

Já no campo da verificação manual feita na seção 5.2. O problema de maior prevalência foi a insuficiência em navegação com teclado. Alguns componentes atendem apenas parcialmente os requisitos das práticas autorais ARIA [11]. Foram encontrados 31 problemas dentre os 37 componentes usados, o que resulta um valor de 0.83 violações por componente para a métrica 2 (M2). Um resultado visivelmente muito melhor que a métrica 1 (M1) mas fazendo-se a ressalva de ter se concentrado em itens relacionados à navegação via teclado, como determinam as práticas autorais ARIA [11]. A verificação com leitor de tela na seção 5.3 também mostrou problemas, mas em menor extensão. Os problemas mais frequentes foram componentes ignorados pelo programa e componentes sem títulos ou rótulos que os identificassem sua função e posição na estrutura da página. Dentre esses problemas, o mais simples de reparar é o de rótulos ausentes, estes também são os problemas cuja responsabilidade recaem sobre os desenvolvedores que fazem uso da MUI e dependem bem menos de mudanças na biblioteca em si.

Finalmente, é claro que a MUI apresenta numerosos problemas de acessibilidade e que desenvolvedores devem fazer suas devidas diligências sempre mantendo em mente as práticas autorais ARIA [11] e WCAG [17] nível AA.

6.2 Pergunta de Pesquisa 2

Algumas das mudanças necessárias para melhorar o índice de conformidade com diretrizes de acessibilidade da MUI são de natureza interna, sendo a MUI uma biblioteca de código aberto, é possível sugerir alterações até mesmo fazê-las localmente. Outras sugestões dizem respeito ao uso dessas tecnologias. Uma vez que o ideal seria um sistema automatizado que sempre garante cobertura de todos os critérios de acessibilidade, tal cenário é excessivamente otimista. Dentro dos resultados obtidos nas seções 5 e 6.1. Algumas recomendações, separadas por responsabilidade são:

1. Responsabilidade: MUI

- (a) Conforme seção 5.1.3, a maioria das violações encontradas de forma automatizada dizem respeito ao contraste de cores dos componentes. Diante disso, recomendamos a escolha de combinação de texto e esquema de cores de forma a maximizar ou, ao menos, alcançar contraste mínimo de 4.5:1 conforme recomendado pelas diretrizes WCAG [17].

- (b) Escolha de arquitetura de software que implemente um padrão de construção dos componentes de forma a não haver sobreposição de componentes na construção de uma página. Esta recomendação tem como objetivo melhorar a capacidade de detecção de problemas por ferramentas automatizadas, como visto na seção 5.1.3.
- (c) Criação de seção dedicada à acessibilidade na documentação. Alguns trechos da documentação têm seções dedicadas à acessibilidade mas essas seções não são uniformemente disponíveis na documentação. [33]
- (d) Adequação dos componentes às interações de teclado presentes nas diretrizes das práticas autorais ARIA [11], com o objetivo de melhorar a navegação via teclado, de acordo com os resultados encontrados na seção 5.2 e no apêndice A.
- (e) Adequação dos componentes ignorados por leitores de tela, fazendo com que passem a ser percebidos e lidos por programas de acessibilidade de forma a prover contexto e informação de funcionalidade aos usuários uma vez que estes foram os principais problemas encontrados na seção 5.3.

2. Responsabilidade: Desenvolvedor

- (a) Atribuição correta de nomes e rótulos aos componentes de forma a deixar claro posições na página, funções e relações de hierarquia entre seus componentes irmão, pais ou filhos. Tal recomendação é importante para prover contexto e títulos para leitores de tela e outras tecnologias assistivas [34].
- (b) Atribuição manual de atalhos a botões e elementos de interface quando estes não implementarem os comportamentos esperados adequadamente. Essa recomendação é útil para mitigar o principal problema encontrado com navegação via teclado na testagem manual, como demonstrado na seções 5.2 e Apêndice A.

3. Responsabilidade: Designer de interfaces

- (a) Conforme seção 5.1.3, a maioria das violações encontradas de forma automatizada dizem respeito ao contraste de cores dos componentes. Diante disso, recomendamos a escolha de combinação de texto e esquema de cores de forma

a maximizar ou, ao menos, alcançar contraste mínimo de 4.5:1 conforme recomendado pelas diretrizes WCAG [17].

- (b) Manter em mente que uma interface precisa funcionar em uma gama de dispositivos, logo, evitar elementos que requerem um método de interação específico como mouse ou tela ou, pelo menos, prover alternativas adequadas, conforme o princípio de robustez da WCAG [35].

É notável no item 1.a que a responsabilidade de garantir contraste suficiente de texto e elementos é compartilhada entre designers, desenvolvedores que usam bibliotecas e os desenvolvedores da biblioteca em si. No item 1.b, já existem alguns componentes com seções dedicadas à acessibilidade em suas páginas, mas não existe uma padronização ou presença completa de tais seções.

Além disso, é importante ressaltar que o item 1.c não necessariamente aumenta o nível de acessibilidade de uma página, mas melhora a acurácia de ferramentas de teste automatizadas como a Axe [26] usada neste trabalho e retorna um número menor de falsos positivos. Levando a uma verificação manual menos custosa e facilitando a implementação de políticas de acessibilidade por parte de desenvolvedores.

6.3 Discussão

Neste trabalho, nosso objetivo foi realizar uma análise do estado da acessibilidade na biblioteca de componentes Material UI (MUI) e também de sugerir possíveis melhorias tanto na implementação da biblioteca em si quanto da formas em que é usada para aumentar o nível de conformidade com as diretrizes de acessibilidade WCAG [17] e ARIA [18]. Acreditamos que nossas descobertas respondem à ambas as perguntas de pesquisa elencadas na seção 3 e mostram que a MUI tem muitos problemas de acessibilidade, mas que podem ser, pelo menos em parte, mitigados pelo uso correto e responsável de suas funcionalidades como determinam as práticas autorais ARIA [11]. Também entendemos que a documentação da MUI é deficiente no quesito de acessibilidade e precisa de melhorias. Recomendamos o uso de ferramentas automatizadas de teste. Apesar de serem bastante dependentes de interação manual e não serem muito autônomas, acreditamos que seu uso é benéfico no cômputo geral e podem ajudar a encontrar problemas de acessibilidade mais facilmente em grandes bases de código.

7 CONCLUSÕES

Nosso propósito com este trabalho foi determinar o quão acessível a biblioteca de componentes Material UI é e, além disso, sugerir possíveis melhorias que possam expandir a cobertura de critérios de acessibilidade da tecnologia, a fim de contribuir com a inclusão de pessoas com deficiência no cenário digital. Para isso, primeiro definimos conceitos chave para a elaboração e entendimento deste trabalho, como os principais grupos de diretrizes de acessibilidade, como WCAG e ARIA, trouxemos também o contexto de melhores práticas de testagem de acessibilidade e introduzimos o objeto de estudo em questão, a Material UI.

Em seguida, definimos de maneira mais estruturada os objetivos do trabalho fazendo uso do modelo GQM, incluindo também uma breve explicação sobre o modelo usado. Então introduzimos a página de testes usada para demonstrar e, principalmente, testar os componentes objeto de estudo deste trabalho. Explicamos o roteiro de testes usado, seguindo a ordem de testes automatizados em conformidade com o nível AA da WCAG, testes manuais exploratórios e com leitor de tela de acordo com as práticas autorais WAI-ARIA.

A apresentação bruta dos resultados destes testes ficou limitada ao apêndice deste trabalho, reservamos a seção de resultados para uma visão geral e análise dos resultados obtidos. Encontramos diversas violações com graus de severidade distintos mas com uma distribuição clara, onde a maior parte das violações de acessibilidade dizem respeito à visibilidade e contraste de cores. Além disso, encontramos como limitação a insuficiência de ferramentas automatizadas para a testagem de acessibilidade. Sendo, portanto, necessária a intervenção humana no processo de testes para obter maior corretude.

Nos testes manuais, encontramos diversas violações de navegação, que prejudicam ou completamente inviabilizam a utilização de certos componentes da página teste e, por extensão, de produtos que se utilizem da Material UI. Finalizamos com respostas às perguntas de pesquisa elencadas na seção 3.2 e uma discussão sobre as implicações dessas informações complementada por sugestões de melhorias para a tecnologia objeto de estudo.

7.1 Aplicações, Limitações e Trabalhos Futuros

Este trabalho se encerra com 3 contribuições principais:

- Descoberta da extensão da acessibilidade da biblioteca Material UI. Ao fazer um estudo abrangente sobre as funcionalidades de acessibilidade da biblioteca, descobrimos o quanto e em quais aspectos a biblioteca é deficiente ou suficiente a respeito de acessibilidade.
- Formulação de um método de testagem de acessibilidade relativamente genérico e, portanto, que pode ser usado para testar o nível de acessibilidade de outras tecnologias. Ao aliar ferramentas automatizadas, testagem exploratória manual e testagem manual com leitor de tela, todas essas estratégias balizadas pelas perguntas elencadas via abordagem GQM, criamos um método que pode ser usado para determinar o nível de acessibilidade e sugerir melhorias em outras tecnologias de componentes usadas para a construção de interfaces gráficas de produtos e serviços digitais, desde que haja um conjunto de diretrizes de acessibilidade já estabelecido para a tecnologia em questão, como a WCAG ou ARIA no caso de aplicações web.
- A sugestão de melhorias com base nas violações de acessibilidade encontradas no curso do trabalho. A implementação destas sugestões tem o potencial de elevar o quão acessível a MUI pode ser e, dessa forma, contribuir para a inclusão de potencialmente milhões de pessoas com deficiência pelo mundo.

Dentre suas limitações, acreditamos que uma testagem mais abrangente, incluindo critérios de nível AAA, poderia resultar em mais achados de violações de acessibilidade, entretanto, com menor relevância e escopo. Este trabalho também não aponta um método completamente genérico de testagem de acessibilidade que possa ser usado para testar todo e qualquer tipo de tecnologia digital. Por fim, não propõe uma análise completamente abrangente do estado de acessibilidade na tecnologia React, na qual a MUI se baseia. Quanto a potenciais trabalhos futuros, existem diversas opções como a testagem de outras bibliotecas de componentes populares na web moderna e a testagem em ambientes móveis ou com conexões e periféricos limitados.

REFERÊNCIAS

- [1] JENKINS, S. P.; RIGG, J. A. Disability and Disadvantage: Selection, Onset, and Duration Effects. *Journal of Social Policy*, v. 33, n. 3, p. 479–501, jul. 2004. ISSN 0047-2794, 1469-7823. Available at: <https://www.cambridge.org/core/product/identifier/S0047279404007780/type/journal_article>.
- [2] BURCHARDT, T. *The education and employment of disabled young people*. [S.l.]: Policy Press, 2005.
- [3] FILMER, D. Disability, Poverty, and Schooling in Developing Countries: Results from 14 Household Surveys. *The World Bank Economic Review*, v. 22, n. 1, p. 141–163, jan. 2008. ISSN 1564-698X, 0258-6770. Available at: <<https://academic.oup.com/wber/article/22/1/141/1682417>>.
- [4] MITRA, S.; POSARAC, A.; VICK, B. C. Disability and Poverty in Developing Countries: A Snapshot from the World Health Survey. *SSRN Electronic Journal*, 2011. ISSN 1556-5068. Available at: <<http://www.ssrn.com/abstract=1908128>>.
- [5] BANTEKAS, I. *The UN Convention on the Rights of Persons with Disabilities, a commentary*. First edition. [S.l.]: Oxford University Press, 2018.
- [6] CEWEB.BR - Centro de Estudos sobre Tecnologias Web. Available at: <<https://ceweb.br>>.
- [7] HOME - The A11Y Project. Available at: <<https://a11yproject.com/>>.
- [8] L13146. Available at: <http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2015/Lei/L13146.htm>.
- [9] L12527. Available at: <http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2011/lei/l12527.htm>.
- [10] WEB Accessibility | Shaping Europe's digital future. Available at: <<https://digital-strategy.ec.europa.eu/en/policies/web-accessibility>>.

- [11] (WAI), W. W. A. I. *Patterns*. Available at: <https://www.w3.org/WAI/ARIA/apg/patterns/>.
- [12] MONT, D. Measuring health and disability. *The Lancet*, v. 369, n. 9573, p. 1658–1663, maio 2007. ISSN 01406736. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0140673607607521>.
- [13] DEMOGRAPHIC Yearbook. 56th edition. ed. [S.l.: s.n.], 2004.
- [14] WHO Expert Committee on Disability Prevention and Rehabilitation. *Disability prevention and rehabilitation*. [S.l.: s.n.], 1981.
- [15] LEONARDI, M. et al. The definition of disability: what is in a name? *The Lancet*, v. 368, n. 9543, p. 1219–1221, out. 2006. ISSN 01406736. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0140673606694981>.
- [16] INTERNATIONAL classification of functioning, disability and health: ICF. [S.l.]: World Health Organization, 2001.
- [17] WEB Content Accessibility Guidelines (WCAG) 2.1. Available at: <https://www.w3.org/TR/WCAG21/>.
- [18] ACCESSIBLE Rich Internet Applications (WAI-ARIA) 1.1. Available at: <https://www.w3.org/TR/wai-aria-1.1/>.
- [19] FRATERNALI, P.; ROSSI, G.; SÁNCHEZ-FIGUEROA, F. Rich Internet Applications. *IEEE Internet Computing*, v. 14, n. 3, p. 9–12, maio 2010. ISSN 1089-7801. Available at: <http://ieeexplore.ieee.org/document/5481362>.
- [20] ARIA: navigation Role - Accessibility | MDN. Available at: https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Roles/navigation_role.
- [21] WAI-ARIA Authoring Practices 1.2. Available at: https://w3c.github.io/aria-practices/no_aria_better_ad_aria.
- [22] ABOUT us - MUI. Available at: <https://mui.com/about/>.
- [23] VIGO, M.; BROWN, J.; CONWAY, V. Benchmarking web accessibility evaluation tools: measuring the harm of sole reliance on automated tests. In: *Proceedings of*

- the 10th International Cross-Disciplinary Conference on Web Accessibility - W4A '13*. Rio de Janeiro, Brazil: ACM Press, 2013. p. 1. ISBN 9781450318440. Available at: <http://dl.acm.org/citation.cfm?doid=2461121.2461124>.
- [24] BRAJNIK, G.; YESILADA, Y.; HARPER, S. The expertise effect on web accessibility evaluation methods. *HUMAN-COMPUTER INTERACTION*, v. 26, p. 246–283, 07 2011.
- [25] SOLINGEN, R. van et al. Goal Question Metric (GQM) Approach. In: MARCINIAK, J. J. (Ed.). *Encyclopedia of Software Engineering*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2002. p. sof142. ISBN 9780471028956. Available at: <https://onlinelibrary.wiley.com/doi/10.1002/0471028959.sof142>.
- [26] AXE: Accessibility Testing Tools and Software. Available at: <https://www.deque.com/axe/>.
- [27] TAW. Available at: <https://www.tawdis.net/index>.
- [28] WEB Content Accessibility Guidelines 1.0. Available at: <https://www.w3.org/TR/WAI-WEBCONTENT/>.
- [29] NV Access. Available at: <https://www.nvaccess.org/>.
- [30] UNDERSTANDING Conformance | Understanding WCAG 2.0. Available at: <https://www.w3.org/TR/UNDERSTANDING-WCAG20/conformance.html#conformance-requirements-head>.
- [31] IKHSAN, I. N.; CANDRA, M. Z. C. Automatically: An Automated Refactoring Method and Tool for Improving Web Accessibility. In: *2018 5th International Conference on Data and Software Engineering (ICoDSE)*. Mataram, Lombok, Indonesia: IEEE, 2018. p. 1–6. ISBN 9781538672594. Available at: <https://ieeexplore.ieee.org/document/8705894>.
- [32] WebAIM: The WebAIM Million - The 2022 report on the accessibility of the top 1,000,000 home pages. Available at: <https://webaim.org/projects/million/>.
- [33] REACT Components - Material UI. Available at: <https://mui.com/pt/material-ui/>.

- [34] (WAI), W. W. A. I. *Providing Accessible Names and Descriptions*. Available at: [<https://www.w3.org/WAI/ARIA/apg/practices/names-and-descriptions/>](https://www.w3.org/WAI/ARIA/apg/practices/names-and-descriptions/).
- [35] WCAG 2.1 Principles Explained: Robustness. Available at: <https://www.boia.org/blog/wcag-2.1-principles-explained-robustness>.

8 APÊNDICE A - TESTES MANUAIS REALIZADOS

Tabela 11: Caso de Teste Acordeão

Componente	CT	Descrição	Comportamento Esperado
Acordeão	1.1	Checar suporte a navegação com TAB	Move o foco para o próximo painel disponível
	1.2	Checar se ENTER ou ESPAÇO podem ser usados para expandir e colapsar painéis	ENTER e ESPAÇO devem estender painéis colapsados e esconder painéis estendidos
	1.3	Checar suporte a navegação com SHIFT + TAB	Move o foco para o painel disponível anterior

Tabela 12: Caso de Teste Diálogo de Alerta

Componente	CT	Descrição	Comportamento Esperado
Diálogo de Alerta	2.1	Checar suporte a navegação com TAB	Move o foco para o próximo botão ou opção disponível
	2.2	Checar se ESC fecha o modal	ESC deve fechar o modal de alerta
	2.3	Checar suporte a navegação com SHIFT + TAB	Move o foco para o botão ou opção disponível anterior

Tabela 13: Caso de Teste Checkbox

Componente	CT	Descrição	Comportamento Esperado
Checkbox	3.1	Checar se espaço marca e desmarca checkbox	Checar se a barra de espaço alterna o estado do checkbox

Tabela 14: Caso de Teste Radio Group

Componente	CT	Descrição	Comportamento Esperado
Radio Group	4.1	Checar se TAB move o foco entre radio groups	Move o foco para o próximo radio group disponível
	4.2	Checar teclas de seta movem o foco entre as opções	Teclas de seta devem mover o foco entre opções dentro do radio group
	4.3	Checar suporte uso da barra de espaço para marcar e desmarcar opções	Barra de espaço deve poder alternar os estados da opção focada

Tabela 15: Caso de Teste Slider

Componente	CT	Descrição	Comportamento Esperado
Slider Rating	5.1	Seta para direita incrementa valor do slider	Checar se seta para direita incrementa o valor do slider em uma unidade
	5.2	Seta para cima incrementa valor do slider	Checar se seta para cima incrementa o valor do slider em uma unidade
	5.3	Seta para baixo decrementa valor do slider	Checar se seta para baixo decrementa o valor do slider em uma unidade
	5.4	Seta para esquerda decrementa valor do slider	Checar se seta para esquerda decrementa o valor do slider em uma unidade
	5.5	Tecla HOME define valor mínimo do slider	Checar se tecla HOME define o valor do slider para o mínimo valor permitido
	5.6	Tecla END define valor máximo do slider	BChecar se tecla END define o valor do slider para o máximo valor permitido

Tabela 16: Caso de Teste Botão

Componente	CT	Descrição	Comportamento Esperado
Botão Botão Flutuante Chip	6.1	Checar se ESPAÇO aciona o botão	Tecla ESPAÇO deve pressionar o botão
	6.2	Checar se ENTER aciona o botão	Tecla ENTER deve pressionar o botão

Tabela 17: Caso de Teste Autocomplete

Componente	CT	Descrição	Comportamento Esperado
Autocomplete	7.1	Checar se seta para baixo move o foco para o popup	Tecla de seta para baixo move o foco para o popup se estiver visível
	7.2	Checar se ESC oculta o popup	Tecla ESC deve ocultar o popup se este estiver visível
	7.3	Checar se ENTER aceita sugestão presente no popup	Tecla ENTER aceita o valor sugerido no popup se este estiver presente e posiciona o cursor ao final do valor aceito

Tabela 18: Caso de Teste Lista Aninhada

Componente	CT	Descrição	Comportamento Esperado
Lista Aninhada	8.1	Checar se ESPAÇO aciona a lista	ESPAÇO deve alternar o estado de visibilidade da lista.
	8.2	Checar se ENTER aciona a lista	ENTER deve alternar o estado de visibilidade da lista.

Tabela 19: Caso de Teste Seleção

Componente	CT	Descrição	Comportamento Esperado
Seleção	9.1	Checar se ESPAÇO seleciona a opção destacada	ESPAÇO deve selecionar a opção visualmente destacada.
	9.2	Checar se ENTER seleciona a opção destacada	ENTER deve selecionar a opção visualmente destacada.
	9.3	Checar se tecla para cima move o foco para a opção anterior	Tecla para cima deve mover o foco para a opção anterior.
	9.4	Checar se tecla para baixo move o foco para a próxima opção	Tecla para baixo deve mover o foco para a opção anterior.

Tabela 20: Caso de Teste Menu

Componente	CT	Descrição	Comportamento Esperado
Menu Inferior Drawer Speed Dial	10.1	TAB move foco para dentro e fora do menu	TAB deve mudar o foco para dentro do menu mas não entre itens do menu.
	10.2	SHIFT+TAB move foco para dentro e fora do menu	SHIFT+TAB deve mudar o foco para dentro do menu mas não entre itens do menu.
	10.3	ENTER abre submenu ou ativa item do menu	ENTER abre submenu e coloca foco no primeiro item do submenu ou ativa o item selecionado se este não for um submenu.
	10.4	SETA PARA BAIXO move foco entre itens ou abre submenu	SETA PARA BAIXO move foco entre itens do menu ou, se houver um submenu, abre o submenu e coloca o foco no primeiro item.
	10.5	SETA PARA CIMA move foco entre itens	SETA PARA CIMA move foco entre itens do menu.
	10.6	SETA PARA DIREITA move foco entre itens ou abre submenu	SETA PARA DIREITA move foco entre itens do menu ou, se houver um submenu, abre o submenu e coloca o foco no primeiro item.
	10.7	SETA PARA ESQUERDA move foco entre itens ou abre submenu	SETA PARA ESQUERDA move foco entre itens do menu ou, se houver um submenu, abre o submenu e coloca o foco no primeiro item.
	10.8	HOME move foco para o primeiro item no menu atual	Se quebra de linha com setas não for suportada, coloca o foco no primeiro item do menu.
	10.9	END move foco para o primeiro item no menu atual	Se quebra de linha com setas não for suportada, coloca o foco no último item do menu.
	10.10	ESC fecha menu e retorna foco	ESC fecha menu atual e retorna foco para elemento pai ou elemento que continha foco inicialmente.

Tabela 21: Caso de Teste Botão Menu

Componente	CT	Descrição	Comportamento Esperado
Botão Menu	11.1	ENTER abre menu e seleciona foco.	ENTER abre menu e coloca foco no primeiro item do menu.
	11.2	ESPAÇO abre menu e seleciona foco.	ESPAÇO abre menu e coloca foco no primeiro item do menu.

Tabela 22: Caso de Teste Link

Componente	CT	Descrição	Comportamento Esperado
Link	12.1	ENTER abre link e move foco	ENTER abre executa link e move foco para o alvo do link.

Tabela 23: Caso de Teste Switch

Componente	CT	Descrição	Comportamento Esperado
Switch	13.1	ESPAÇO aciona switch.	ESPAÇO executa switch se o foco estiver no componente em questão.

Tabela 24: Caso de Teste Aba

Componente	CT	Descrição	Comportamento Esperado
Aba	14.1	TAB posiciona foco na primeira aba.	Se o foco está fora das abas, TAB deve colocar o foco na primeira aba. Se o foco está na primeira aba, TAB move o foco para o próximo elemento fora da lista de abas.
	14.2	SETA PARA ESQUERDA move foco para aba anterior.	SETA PARA ESQUERDA deve mudar o foco para a aba anterior se esta existir e retornar para a primeira aba se não existir.
	14.3	SETA PARA DIREITA move foco para aba seguinte.	SETA PARA DIREITA deve mudar o foco para a aba seguinte se esta existir e retornar para a última aba se não existir.
	14.4	ESPAÇO aciona a aba..	ESPAÇO aciona a aba se esta não for ativada automaticamente em foco.
	14.5	SHIFT+F10 abre menu pop up.	SHIFT+F10 abre menu pop up associado à aba se este existir.