

UNIVERSIDADE DE SÃO PAULO
FACULDADE DE FILOSOFIA, CIÊNCIAS E LETRAS DE RIBEIRÃO PRETO
DEPARTAMENTO DE COMPUTAÇÃO E MATEMÁTICA

VINÍCIUS DE SOUZA GONÇALVES

**Avaliação automática de acessibilidade em interfaces
de usuário de aplicações móveis**

Ribeirão Preto–SP

2021

VINÍCIUS DE SOUZA GONÇALVES

**Avaliação automática de acessibilidade em interfaces de usuário
de aplicações móveis**

Dissertação apresentada à Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto (FFCLRP) da Universidade de São Paulo (USP), como parte das exigências para a obtenção do título de Mestre em Ciências.

Área de Concentração: Computação Aplicada.

Orientador: Alessandra Alaniz Macedo

Ribeirão Preto–SP

2021

Vinícius de Souza Gonçalves

Avaliação automática de acessibilidade em interfaces de usuário de aplicações móveis. Ribeirão Preto–SP, 2021.

95p. : il.; 30 cm.

Dissertação apresentada à Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto da USP, como parte das exigências para a obtenção do título de Mestre em Ciências,

Área: Computação Aplicada.

Orientador: Alessandra Alaniz Macedo

1. Acessibilidade. 2. Técnicas de avaliação de acessibilidade. 3. Avaliação automática de acessibilidade. 4. Inclusão digital. 5. Tecnologia assistiva. 6. Interação Humano-Computador. 7. WCAG.

Agradecimentos

Primeiramente, agradeço a Deus, por ter me proporcionado a oportunidade de fazer mestrado e por ter estado comigo, quando mais necessitei de ajuda. Agradeço aos meus pais, José Reis e Luciana, pelos conselhos e pelo apoio financeiro durante esse mestrado. Quero agradecer a minha namorada Gabriela, a qual esteve comigo desde do começo do mestrado e que vivenciou minhas vitórias e minhas dificuldades, além de muitas vezes ter revisado meu texto e me concedido pareceres.

Gratulo a todos que participaram desta pesquisa, professores, pesquisadores e desenvolvedores de *softwares*, porque sem vocês parte dessa pesquisa não poderia ser concluída. Agradeço a Fernando Meloni pelo auxílio e orientações na etapa de construção dos resultados e discussões. Gratulo os professores(as) e pesquisadores(as) Kamila Rios, Renata Pontin e Renato Bulcão pelas contribuições concedidas nas defesas da qualificação e da dissertação.

Por fim, não menos importante, agradeço minha orientadora Alessandra Alaniz Macedo, a qual me concedeu a oportunidade de trabalhar no projeto SofiaFala, o qual tenho muito orgulho de ter participado. Além disso, agradeço pela grande colaboração e apoio para a publicação do artigo *Similarities and Divergences in Electronic Game Review Texts*, o qual publicamos na SBGAMES 2020, e por todas as orientações durante esses anos de mestrado.

*“Com sabedoria se constrói a casa,
e com discernimento se consolida.*

*Pelo conhecimento os seus
cômodos se enchem do que é
precioso e agradável.”*

(Bíblia Sagrada, Provérbios 24:3-4)

Resumo

Problemas de acessibilidade em aplicações móveis podem fazer com que um usuário não consiga utilizar adequadamente uma aplicação. A adoção de algumas diretrizes e práticas de desenvolvimento pode tornar as aplicações mais acessíveis e conseqüentemente diminuir frustrações, perdas de tempo e monetárias, além de aumentar a produtividade e o alcance de metas. Logo, estudos nas áreas de Ergonomia e Interação Humano-Computador (IHC) objetivam a formulação de diretrizes e princípios para auxiliar o desenvolvimento de aplicações móveis acessíveis ao público, em geral e com deficiências. Para tornar uma aplicação acessível, desenvolvedores de *software* devem conhecer e aplicar diretrizes, recomendações e princípios voltados à acessibilidade desde o início do processo de desenvolvimento de *software*. Dentro desse processo, a avaliação de acessibilidade de interfaces de usuário de aplicações móveis exige um vasto estudo e conhecimento de documentos que apresentam recomendações para tratar acessibilidade. Com o intuito de facilitar o processo de avaliação de acessibilidade de interfaces de aplicações móveis, uma ferramenta de avaliação de acessibilidade automatizada pode auxiliar: o desenvolvimento efetivo de interfaces acessíveis de aplicativos móveis e a conceder *feedbacks* ao desenvolvedor de *softwares* sobre a acessibilidade da aplicação móvel de forma automatizada. O objetivo deste trabalho é auxiliar a avaliação de acessibilidade de interfaces de usuário de aplicações móveis, por meio de uma ferramenta computacional que avalie automaticamente a acessibilidade de interfaces de usuário de aplicações móveis, seguindo as diretrizes da WCAG 2.1 propostas pelo W3C. A ferramenta de avaliação de acessibilidade proposta, denominada AETool (*Android Accessibility Evaluation Tool*), pode conceder *feedbacks* durante a concepção ao prototipar aplicações móveis, o desenvolvimento das aplicações e, ainda durante a avaliação de acessibilidade. Para avaliar a eficácia da AETool, desenvolveu-se uma prova de conceito, comparando a avaliação de acessibilidade da AETool em relação a avaliação de humanos e em relação ao *Accessibility Insights* e o *scanner* de acessibilidade da Google. A AETool encontrou 56,53% das violações de acessibilidade existentes na soma das quatro aplicações móveis utilizadas como objetos de estudo. A AETool julgou haver 340 violações de acessibilidade somando as violações encontradas em todas as quatro avaliações. Dentre as 340, 303 são violações reais, ou seja, 89,11%. Esse fator é importante de considerar, pois mostra a precisão da AETool em encontrar violações de acessibilidade reais.

Palavras-chave: Acessibilidade. Técnicas de avaliação de acessibilidade. Avaliação automática de acessibilidade. Inclusão digital. Tecnologia assistiva. Interação Humano-Computador. WCAG.

Abstract

Accessibility issues in mobile applications can make a user unable to properly use an application. The adoption of some development guidelines and practices can make applications more accessible and, consequently, reduce frustration, wasted time and money, in addition to increasing productivity and achieving goals. Therefore, studies in the areas of Ergonomics and Human-Computer Interaction (HCI) aim to formulate guidelines and principles to assist in the development of mobile applications accessible to the general public and those with disabilities. To make an application accessible, software developers must know and apply accessibility guidelines, recommendations and principles from the beginning of the software development process. Within this process, the accessibility assessment of mobile application user interfaces requires a vast study and knowledge of documents that present recommendations to address accessibility. In order to facilitate the accessibility assessment process of mobile application interfaces, an automated accessibility assessment tool can help: the effective development of accessible mobile application interfaces and provide feedbacks to the developer of softwares about the accessibility of the mobile application in an automated way. The objective of this work is to help assess the accessibility of mobile application user interfaces, through a computational tool that automatically assesses the accessibility of mobile application user interfaces, following the guidelines of WCAG 2.1 proposed by the W3C. The proposed accessibility assessment tool, called AETool (Android Accessibility Evaluation Tool), can provide feedbacks during the development or prototyping stage of the mobile application, to assess accessibility issues. To assess the effectiveness of AETool, it was developed a proof-of-concept, comparing AETool's accessibility rating against software developers' rating and against Accessibility Insights and scanner of Google accessibility. AETool found 56.53% of existing accessibility violations in the sum of the four mobile applications used as study objects. AETool judged there were 340 accessibility violations summing the violations found in all four assessments. Among the 340, 303 are real violations, that is, 89.11%. This factor is important to consider as it shows AETool's accuracy in finding real accessibility violations.

Keywords: Accessibility. Accessibility assessment techniques. Automatic accessibility assessment. Digital inclusion. Assistive technology. Human-Computer Interaction. WCAG.

Lista de figuras

Figura 1 – Interface de usuário do Access Monitor após avaliar página web do Uol.	32
Figura 2 – Interface de usuário do Accessibility Insights após avaliar aplicação móvel do SofiaFala - Fonoaudiólogos.	33
Figura 3 – Capturas de tela das aplicações móveis WhatsApp(a), Youtube(b) e Uber(c)(d).	39
Figura 4 – Capturas de tela das aplicações móveis do SofiaFala, módulos Fonoaudiólogos e Criança.	40
Figura 5 – Capturas de tela das aplicações móveis Minimal To Do(a)(b) e Sound Recorder(c)(d)	41
Figura 6 – Diagrama de Casos de Uso da AETool.	54
Figura 7 – Funcionamento da AETool.	55
Figura 8 – Interface de usuário da AETool.	56
Figura 9 – AETool apresentando os problemas de acessibilidade separados por interface de usuário.	56
Figura 10 – (a) Tela de listagem de pacientes. (b) Tela de cadastro de paciente. (c) Tela de cadastro de treino. (d) Tela de relatório de treino articulatorio.	60
Figura 11 – (a) Tela de listagem de treinos. (b) Tela de treino de beijo. (c) Tela de treino de bico-sorriso. (d) Tela de treino de movimento articulatorio.	61
Figura 12 – (b) Tela de áudios gravados com o Sound Recorder. (c) Tela com tarefas criadas no Minimal To do. (c) Tela para cadastro de tarefas no Minimal To do.	61
Figura 13 – Violações reais encontradas pela AETool e pelos humanos em relação a cada aplicação móvel	68
Figura 14 – Violações reais de acessibilidade encontradas pela AETool e pelos avaliadores (humanos) em relação a cada aplicação móvel. (a) AETool e Avaliador 1. (b) AETool e Avaliadores 1 e 2. (c) AETool e Avaliadores 1, 2 e 3. (d) AETool e Avaliadores 1, 2, 3 e 4. (e) AETool e Avaliadores 1, 2, 3, 4 e 5. (f) AETool e Avaliadores 1, 2, 3, 4, 5 e 6.	69
Figura 15 – Violações reais de acessibilidade encontradas pela AETool e pelos avaliadores (humanos) em relação a cada aplicação móvel e separadas por CS. (a) Sound Recorder. (b) Minimal To Do. (c) SofiaFala - Criança. (d) SofiaFala - Fonoaudiólogos.	71
Figura 16 – Violações reais de acessibilidade encontradas pela AETool, pelo scanner de acessibilidade da Google e pelo Accessibility Insights separadas por CS. (a) AETool versus scanner de acessibilidade da Google. (b) AETool versus Accessibility Insights.	73

Lista de tabelas

Tabela 1 – Sugestões de acessibilidade, segundo o <i>scanner</i> de acessibilidade da Google.	41
Tabela 2 – CSs selecionados para serem implementados na AETool.	42
Tabela 3 – Critérios de sucesso não selecionados para implementação na AETool, devido à Condição 1	49
Tabela 4 – Critérios de sucesso não selecionados para implementação na AETool, devido à Condição 2	49
Tabela 5 – Critérios de sucesso não selecionados para implementação na AETool, devido a Condição 3	50
Tabela 6 – Critérios de sucesso não selecionados para implementação na AETool, devido a Condição 4	51
Tabela 7 – Requisitos Funcionais da AETool.	51
Tabela 8 – Perfil dos Avaliadores.	63
Tabela 9 – Porcentagem de violações reais de acessibilidade encontradas pelos avaliadores (humanos) e pela ferramenta (AETool) separadas por CS .	70

Sumário

1	INTRODUÇÃO	21
1.1	Objetivo	23
1.2	Metodologia	23
1.3	Organização do Documento	24
2	REFERENCIAL TEÓRICO	27
2.1	Acessibilidade	27
2.1.1	Definição de Acessibilidade	27
2.1.2	Acessibilidade em Dispositivos e Aplicações Móveis	28
2.2	Diretrizes de Acessibilidade de Conteúdo da Web	29
2.3	Trabalhos Relacionados	31
2.3.1	Ferramentas Automáticas para Avaliação de Acessibilidade de Aplicações <i>Web</i>	31
2.3.2	Ferramentas Automáticas para Avaliação de Acessibilidade de Aplicações <i>Móveis</i>	32
2.3.3	Métodos Alternativos para Avaliação de Acessibilidade de Aplicações Móveis sem Uso de Ferramentas Automáticas	34
2.3.4	Considerações Finais	35
3	AETOOL	37
3.1	Elicitação e Análise de Requisitos	37
3.1.1	Investigação Prévia	37
3.1.2	Critérios de Sucesso Seleccionados	42
3.1.2.1	Conteúdo Não Textual (CS.1.1.1)	43
3.1.2.2	Informações e Relacionamentos (CS.1.3.1)	43
3.1.2.3	Orientação (CS.1.3.4)	44
3.1.2.4	Identificar Finalidade de Entrada (CS.1.3.5)	44
3.1.2.5	Contraste (mínimo) (CS.1.4.3)	44
3.1.2.6	Contraste (aprimorado) (CS.1.4.6)	45
3.1.2.7	Apresentação Visual (CS.1.4.8)	45
3.1.2.8	Contraste sem Texto (CS.1.4.11)	46
3.1.2.9	Título da Página (CS.2.4.2)	46
3.1.2.10	Objetivo do Link (no Contexto) (CS.2.4.4)	46
3.1.2.11	Objetivo do Link (Apenas Link) (CS.2.4.9)	46
3.1.2.12	Títulos de Seção (CS.2.4.10)	47
3.1.2.13	Gestos de Ponteiro (CS.2.5.1)	47

3.1.2.14	Tamanho do Alvo (CS.2.5.5)	47
3.1.2.15	Ajuda (CS.3.3.5)	47
3.1.2.16	Análise (CS.4.1.1)	48
3.1.3	Critérios de Sucesso não Selecionados	48
3.1.4	Requisitos Funcionais da AETool	51
3.2	Modelagem	52
3.3	Codificação	54
3.4	Considerações Finais	58
4	PROVA DE CONCEITO COM AETOOL	59
4.1	Aplicações para Prova de Conceito	59
4.2	AETool <i>versus</i> Julgamento Humano	62
4.3	AETool <i>versus</i> Outras Ferramentas	63
4.4	Considerações Finais	64
5	RESULTADOS E DISCUSSÃO	67
5.1	AETool <i>versus</i> Julgamento Humano	67
5.2	AETool <i>versus</i> Outras Ferramentas	72
5.3	Considerações Finais	74
6	CONCLUSÃO	75
6.1	Contribuições	76
6.2	Limitações	77
6.3	Trabalhos Futuros	78
REFERÊNCIAS		79
APÊNDICE A	CHECKLIST PARA AVALIAÇÃO DE APLICAÇÕES MÓVEIS UTILIZANDO AS WCAG 2.1	83
APÊNDICE B	PLANILHA COM DADOS DA AVALIAÇÃO COM AVALIADORES E COM A AETOOL	93

Introdução

Atualmente, os *smartphones* com sua capacidade computacional e de conectividade tornaram-se essenciais para realização das atividades humanas diárias (PARASURAMAN et al., 2017; SILVA; FERREIRA; SACRAMENTO, 2018). Uma característica advinda dos *smartphones* ou dispositivos móveis, em geral, é a disponibilidade de suportarem aplicações que fornecem serviços essenciais, úteis e desejáveis para os usuários. Porém, usuários com deficiências podem ser impedidos de fazer uso dos recursos promovidos pelas aplicações móveis, como a comunicação por textos ou mídias, a busca de informações na *web*, a visualização de informações e dentre outros, caso a aplicação móvel não forneça acessibilidades para a inclusão digital (TAN et al., 2014; ABDUL AZIZ; WAN AHMAD; BINTI ZULKIFLI, 2015; ALFREDO et al., 2015; BOTTINO et al., 2015; BLANKENHAGEL, 2019; SAMONTE et al., 2019).

A exclusão digital pode estar no fato da aplicação móvel não garantir tamanhos de itens clicáveis grandes o suficiente para que os usuários possam interagir facilmente ou no fato da interface de usuário ter orientação de exibição fixa, não permitindo assim adaptar a orientação de exibição para que funcione melhor ao usuário, e dentre outros problemas. Além disso, o acesso a produtos de Tecnologia de Informação e Comunicação (TIC) por pessoas com deficiência é um direito amparado pela Lei Brasileira de Inclusão de Pessoas com Deficiência N° 13.146:

"Parágrafo único. Serão estimulados, em especial o emprego de tecnologias da informação e comunicação como instrumento de superação de limitações funcionais e de barreiras à comunicação, à informação, à educação e ao entretenimento da pessoa com deficiência" (BRASIL, 2015)

A CIF (Classificação Internacional de Funcionamento, Incapacidade e Saúde) que apoia a definição de leis e direitos para pessoas com deficiências classifica os tipos de deficiências em (i) visual, (ii) auditiva, (iii) motora e (iv) intelectual (SIEBRA et al., 2017). As deficiências visuais podem ser divididas em dois grupos, sendo cegueira e pouca

visão. Por outro lado, deficiências auditivas são divididas em perda total ou parcial da habilidade de ouvir sons, variando em diferentes graus e níveis, os quais podem ser medidos em decibéis (dB). Deficiências motoras podem ser congênitas ou adquiridas, afetando a mobilidade dos indivíduos. Por fim, tem-se as deficiências intelectuais ou cognitivas, que, comumente, presume-se ser causadas devido alguma disfunção do sistema nervoso, caso, o qual acontece, por exemplo, em pessoas com Síndrome de Down (SD), doença de mutação genética, a qual tem como uma das suas consequências a disfunção cognitiva (SIEBRA et al., 2017).

No contexto de aplicações móveis, a acessibilidade diz respeito a tornar as aplicações móveis possíveis de serem utilizadas por pessoas com deficiências (SILVA; FERREIRA; SACRAMENTO, 2018; FRANKLIN; MYNENI, 2018). Além de tornar às aplicações acessíveis as pessoas com deficiências, a adoção da prática de desenvolvimento de *softwares* acessíveis pode reduzir perdas monetárias, aumentar produtividade e alcançar metas e lucros (BARUA; MANI; MUKHERJEE, 2012; NIELSEN, 2012; SIEBRA et al., 2017). Nesse contexto, os estudos sobre acessibilidade em interfaces de usuário de aplicações móveis levam em consideração que podem haver usuários com alguma deficiência dentro do público-alvo da aplicação.

Os indivíduos com deficiências podem ter dificuldades para desempenharem uma participação efetiva na sociedade (HELATHY, 2015; LYON, 1996). Eles podem ter dificuldades para utilizar um dispositivo móvel, bem como, as aplicações (ALFREDO et al., 2015; ABDUL AZIZ; WAN AHMAD; BINTI ZULKIFLI, 2015). Nesse sentido, surgem questões sobre o tratamento de acessibilidade em aplicações móveis, quando usados por pessoas com deficiências.

Nesse contexto, como sendo a hipótese geral deste trabalho, será que existem técnicas para avaliação de acessibilidade de aplicações móveis? Em relação às deficiências motoras, Alfredo et al. (2015) analisaram os oito gestos mais comumente utilizados em aplicações móveis, sendo realizados por pessoas com Síndrome de Down (SD). No trabalho de SIEBRA et al. (2017), foi realizado um estudo a fim de levantar propostas de diretrizes que deveriam ser consideradas no desenvolvimento de aplicações móveis, tendo como objetivo de assegurar a acessibilidade. Por fim, CALVO, SEYEDARABI e SAVVA (2016) realizaram um estudo com o intuito de prover recomendações para que os desenvolvedores de *software* possam criar sites acessíveis.

Além desses trabalhos encontrados na literatura, há as organizações e empresas que realizam estudos e formulam diretrizes com o propósito de auxiliar os desenvolvedores a criarem aplicações acessíveis ao público em geral. Empresas como Apple e Google possuem princípios de desenvolvimento de interfaces de usuário, a fim de ajudar os desenvolvedores a projetarem aplicações mais acessíveis aos seus respectivos sistemas operacionais (iOS e Android) (SILVA; FERREIRA; SACRAMENTO, 2018).

O W3C, considerado como principal organização de padronização da *World Wide Web* (WWW), desenvolveu as Diretrizes de Acessibilidade para o Conteúdo da Web (WCAG), as quais atualmente encontram-se na versão 2.1 (W3C, 2011). As WCAG pretendem que o conteúdo *web* se torne mais acessível aos usuários em geral. O cumprimento destas diretrizes faz com que o conteúdo se torne acessível a um maior número de pessoas com deficiências, incluindo cegueira e baixa visão, surdez e baixa audição, dificuldades de aprendizagem, distúrbios fonológicos e limitações cognitivas. Além disso, essas diretrizes auxiliam o desenvolvimento e avaliação de *softwares*, a fim de torná-los mais acessíveis (WCAG2.1, 2018).

A realização de uma avaliação, a fim de desenvolver uma interface acessível, pode exigir uma vasta leitura de documentos extensos, além de exigir que tenham-se especialistas que conheçam tais diretrizes. Dado o contexto apresentado, o problema de pesquisa tratado neste trabalho é a dificuldade de realizar avaliações de interfaces de usuários para aplicativos móveis, as quais podem exigir uma vasta leitura de documentos técnicos e especialidade de conhecimento nas diretrizes de acessibilidade.

Com a motivação de facilitar o procedimento para avaliação de acessibilidade, uma ferramenta de avaliação de acessibilidade pode surgir como uma forma auxiliar a avaliação de acessibilidade. Nesse contexto, tem-se como hipótese específica desta dissertação que: uma ferramenta para avaliar a acessibilidade de interfaces de usuário de aplicações móveis automaticamente pode surgir como um diferencial no processo de desenvolver aplicações móveis. Além de poder auxiliar o desenvolvedor de *softwares* a captar problemas de acessibilidade na aplicação móvel, a fim de solucioná-los.

1.1 Objetivo

O objetivo deste trabalho é auxiliar a avaliação de acessibilidade de interfaces de usuário de aplicações móveis, por meio de uma ferramenta que avalie automaticamente a acessibilidade de interfaces de usuário de aplicações móveis, seguindo as diretrizes da WCAG 2.1.

1.2 Metodologia

A ferramenta, denominada AETool (*Android Accessibility Evaluation Tool*), foi desenvolvida para avaliar códigos fontes de aplicações móveis desenvolvidas para Android nativo e para conceder *feedbacks* ao desenvolvedor durante o desenvolvimento de uma aplicação móvel.

Para construção da AETool e avaliação da sua eficácia foram as seguidas as etapas: (i) elicitação e análise de requisitos, (ii) modelagem da AETool, (iv) codificação da AETool,

(v) prova de conceito com a AETool. A elicitación e análise de requisitos consistiu de um levantamento de ferramentas de avaliação de acessibilidade de aplicações móveis existentes, do levantamento de requisitos da AETool e de uma análise dos Critérios de Sucesso (CSs) da WCAG 2.1, a fim de selecionar quais seriam avaliados pela AETool.

Para avaliar a acessibilidade da interface de usuário de uma aplicação móvel por meio da AETool, necessita-se do código-fonte da interface de usuário, o qual precisa estar no formato *Extensible Markup Language (XML)*. Nesse trabalho, optou-se em focar no desenvolvimento de uma ferramenta para avaliação de acessibilidade para aplicações móveis desenvolvidas nativamente para o sistema operacional Android. Porém, esse estudo pode ser ampliado para avaliar a acessibilidade de interfaces de usuários de aplicações móveis desenvolvidas para iOS. A AETool avalia os CSs, independente do tipo de deficiência, que os mesmos abordam, desde que seja possível de avaliar as recomendações do CS da WCAG 2.1 avaliando o código-fonte da interface de usuário em formato *XML*.

A etapa de modelagem consistiu no desenvolvimento de um diagrama de casos de usos da AETool com os processos de cada ator e o diagrama de fluxo de entrada e saída de dados da AETool. A etapa de codificação consistiu na implementação do código-fonte da AETool. Por fim, a etapa de prova de conceito da AETool consistiu na avaliação da eficácia da AETool comparando seus resultados de avaliação com avaliações de acessibilidade realizadas por humanos e por outras ferramentas de avaliação de acessibilidade de aplicações móveis.

Algumas aplicações móveis foram selecionadas para a prova de conceito, a fim de levantar problemas de acessibilidade sendo essas: (i) SofiaFala - Fonoaudiólogos, (ii) SofiaFala - Criança, (iii) Sound Recorder e (iv) Minimal To Do. Os humanos avaliaram a acessibilidade das aplicações móveis citadas, seguindo as diretrizes do WCAG 2.1 do W3C. Com o comparativo entre as avaliações realizadas pela AETool em relação as dos humanos, foi possível avaliar a eficácia da ferramenta. Além disso, um comparativo das avaliações da AETool em relação a outras ferramentas de avaliação de acessibilidade existentes foi realizada, a fim de avaliar o desempenho da AETool em relação a outras ferramentas.

1.3 Organização do Documento

O presente documento está organizado na seguinte estrutura:

- O Capítulo 2 descreve os aspectos teóricos referentes à acessibilidade, às tecnologias para desenvolvimento de aplicações móveis, às diretrizes de acessibilidade e os trabalhos relacionados.

-
- O Capítulo 3 apresenta a metodologia utilizada neste trabalho, especialmente, em relação a análise de requisitos, modelagem e codificação da AETool.
 - O Capítulo 4 apresenta a metodologia aplicada para prova de conceito para avaliar a eficácia da AETool.
 - O Capítulo 5 apresenta os resultados obtidos com a prova de conceito com a AETool, mostrando comparativo entre a avaliação de acessibilidade de aplicações móveis realizada pela AETool em relação a dos humanos e em relação a outras ferramentas.
 - O Capítulo 6 apresenta as conclusões e contribuições logradas ao término do trabalho e as perspectivas inerentes a trabalhos futuros correlacionados com o tema de estudo.

Referencial Teórico

O referencial teórico utilizado para realização deste trabalho é descrito neste capítulo. A Seção 2.1 descreve os conceitos de acessibilidade. A Seção 2.2 apresenta as Diretrizes de Acessibilidade de Conteúdo da Web (WCAG). Por fim, a Seção 2.3 descreve os trabalhos relacionados utilizados como base, discussão e comparação com este presente trabalho.

2.1 Acessibilidade

Esta seção apresenta a definição de acessibilidade e a acessibilidade no contexto de dispositivos e aplicações móveis.

2.1.1 Definição de Acessibilidade

Conforme a Lei Brasileira de Inclusão N° 13.146:

"Acessibilidade é a possibilidade e condição de alcance para utilização, com segurança e autonomia, de espaços, mobiliários, equipamentos urbanos, edificações, transportes, informação e comunicação, inclusive seus sistemas e tecnologias, bem como de outros serviços e instalações abertas ao público, de uso público ou privado, tanto na zona urbana como rural, por pessoas com deficiência ou mobilidade reduzida" (BRASIL, 2015).

Segundo a ISO 9241, a acessibilidade de sistemas interativos pode ser definida como sendo a usabilidade de um produto, ambiente, serviço por pessoas com deficiência (ISO/TR-9241-210, 2010). A ISO 9241 considera que em determinados casos, os usuários de *softwares* podem necessitar de tecnologias assistivas para fazer uso de algum sistema, como por exemplo, o uso de um leitor de telas para pessoas com deficiências visuais (CARVALHO; FREIRE, 2017).

O acesso a produtos de Tecnologia de Informação e Comunicação (TIC) por pessoas com deficiência é um direito amparado pela Lei Brasileira de Inclusão de Pessoas com Deficiência:

"Parágrafo único. Serão estimulados, em especial o emprego de tecnologias da informação e comunicação como instrumento de superação de limitações funcionais e de barreiras à comunicação, à informação, à educação e ao entretenimento da pessoa com deficiência" (BRASIL, 2015)

No Artigo 66, presente no Capítulo 2, da Lei Brasileira de Inclusão N° 13.146, é mencionado a importância de acessibilidade para dispositivos móveis da seguinte maneira:

"Cabe ao poder público incentivar a oferta de aparelhos de telefonia fixa e móvel celular com acessibilidade que, entre outras tecnologias assistivas, possuam possibilidade de indicação e de ampliação sonoras de todas as operações e funções disponíveis" (BRASIL, 2015)

2.1.2 Acessibilidade em Dispositivos e Aplicações Móveis

Dispositivos móveis são aparelhos que permitem que os usuários façam uso de recursos como câmera, sensores de GPS e bússola, internet e aplicações móveis, além de permitir ligações e envio de mensagens de texto (KARDYS, 2016). Segundo a 29ª Pesquisa Anual de uso de TI de 2018, no Brasil, há 220 milhões de smartphones em uso (MEIRELLES, 2018).

Em geral as interfaces físicas possuem poucas teclas, se comparado aos seus precursores, os celulares tradicionais. Comumente, o uso da interface de um *smartphone* é por meio de gestos e toques na tela. Nesse sentido, dispositivos móveis podem trazer uma complexidade extra para tratar acessibilidade, pois há características específicas como o próprio mecanismo de interação por gestos e toques, além do tamanho da tela do dispositivo, portabilidade e contextos de uso (BILLI et al., 2010; HARPER; THIESSEN; YESILADA, 2018; BUDIU, 2015; EMAG, 2018).

Com o intuito de apoiar a acessibilidade em aplicações móveis, a Google e a Apple disponibilizam recursos, para pessoas com deficiência, integrados aos seus sistemas operacionais móveis, Android e iOS. Dentre os recursos disponibilizados, há a possibilidade de aumentar e mudar cor de fontes de texto e imagens, de conectar teclados braile via *bluetooth*, de fazer uso de assistentes de voz e de utilizar os *softwares* de leitores de tela instalados de fábrica (ANDROID, 2018; APPLE, 2018).

Além desses recursos, essas empresas disponibilizam documentos para auxiliar os desenvolvedores de *softwares* a proverem acessibilidade nas aplicações móveis desenvolvidas. A Apple disponibiliza o guia *iOS Human Interface Guidelines* (APPLE, 2018), no qual são apresentados conceitos referentes a elementos de interfaces de usuários. Embora seja um documento considerado abrangente, esse guia está fortemente ligado às exigências tecnológicas presentes no sistema operacional iOS. De forma semelhante, a Google disponibiliza um guia denominado *Android Design* (ANDROID, 2018), o qual apresenta princípios de design e recomendações para desenvolvimento de aplicações para o sistema operacional Android.

O W3C, principal organização de padronização da WWW, publicou os documentos *Mobile Web Best Practices* (MWBP) e *Mobile Web Application Best Practices* (MWABP), nos quais ambos possuem orientações sobre como desenvolver aplicações móveis que sejam acessíveis (ZEIN; SALLEH; GRUNDY, 2016). Nesse contexto de acessibilidade, o MWBP e as WCAG buscam auxiliar a interação dos usuários que enfrentam dificuldades de acesso, seja por limitação física do usuário ou pelo dispositivo móvel utilizado (W3C, 2009).

2.2 Diretrizes de Acessibilidade de Conteúdo da Web

Atualmente, as Diretrizes de Acessibilidade ao Conteúdo da Web, desenvolvidas pelo W3C, encontram-se na versão 2.1. O intuito das WCAG 2.1 são de auxiliar com recomendações que tornem o conteúdo mais acessível as pessoas com deficiências visuais, auditivas, motoras, fonológicas, intelectuais e dentre outras (WCAG2.1, 2018). As WCAG 2.1 abordam acessibilidade do conteúdo da web em computadores, *laptops*, *tablets* e dispositivos móveis. Além disso, as WCAG 2.1 são uma extensão das WCAG 2.0. As WCAG 2.1 podem ser utilizadas para auxiliar no desenvolvimento de aplicações móveis acessíveis.

Segundo (WCAG2.1, 2018), para que as aplicações *web* ou móveis sejam acessíveis, um conjunto de orientações é necessário para atender às necessidades da maioria do público, sendo divididas em níveis denominados (i) princípios, (ii) diretrizes, (iii) critérios de sucesso e (iv) técnicas suficientes e consultivas.

Princípios fornecem a base para a acessibilidade de conteúdos *web*, sendo divididos em quatro princípios: perceptível, operável, compreensível e robusto. Diretrizes fornecem os objetivos básicos, aos quais os desenvolvedores devem trabalhar a fim de tornar o conteúdo mais acessível aos usuários com necessidades especiais. As diretrizes das WCAG 2.1 podem ser observados a seguir:

- **Diretriz 1.1 Textos Alternativos** é o fornecimento de alternativas textuais para

conteúdos não textuais para que possa ser realizada a alteração para diferentes tipos de formulários, como braille e fala;

- **Diretriz 1.2 Mídia Baseada em Tempo** é o fornecimento de alternativas textuais ou de outro tipo para mídias baseadas em tempo;
- **Diretriz 1.3 Adaptabilidade** é a criação de conteúdos que possam ser apresentados de diferentes formas sem que haja perda de informações ou de estrutura;
- **Diretriz 1.4 Distinguível** é facilitar para que os usuários possam ver e ouvir o conteúdo, incluindo a separação entre primeiro e segundo plano;
- **Diretriz 2.1 Teclado Acessível** é a disponibilização de todas as funcionalidades da aplicação para serem utilizadas a partir de um teclado;
- **Diretriz 2.2 Tempo Suficiente** é o fornecimento de tempo suficiente para que os usuários leiam e usem o conteúdo;
- **Diretriz 2.3 Convulsões e Reações Físicas** recomenda não criar conteúdos que sejam conhecidos por causarem convulsões ou reações físicas;
- **Diretriz 2.4 Navegável** é prover formas de ajudar os usuários a navegarem e a encontrarem os conteúdos na aplicação;
- **Diretriz 2.5 Modalidades de Entrada** é facilitar a experiência do usuário através de entradas que estendam ao teclado;
- **Diretriz 3.1 Legível** é o fornecimento de conteúdos de texto legíveis e entendíveis;
- **Diretriz 3.2 Previsível** é o fornecimento de páginas ou telas que apareçam e operem de formas previsíveis;
- **Diretriz 3.3 Assistência de entrada** é o fornecimento de ajuda aos usuários para que erros sejam evitados e corrigidos;
- **Diretriz 4.1 Compatível** é a maximização de compatibilidade com agentes usados por usuários atuais e futuros, como tecnologias assistivas.

Cada diretriz tem critérios de sucesso testáveis que permitam a utilização das WCAG, os quais têm o intuito de atender as necessidades dos diferentes grupos de usuários. Para isso, três níveis de conformidade são definidos, sendo A (menor), AA e AAA (maior). As técnicas são informativas e se dividem em duas categorias: (i) suficientes para atender aos critérios de sucesso, como, por exemplo, o uso de marcações para definir que um campo do formulário é obrigatório; (ii) consultivas que vão além do que é exigido pelos critérios individuais de sucesso, permitindo aos autores abordarem melhor as diretrizes.

A Google possui um *scanner* de acessibilidade que baseia-se em algumas diretrizes semelhantes as das WCAG 2.1. Esse *scanner* da Google foi utilizado neste trabalho de mestrado, a fim de realizar uma análise prévia sobre ferramentas automatizadas para avaliação de aplicações móveis já existentes (WCAG2.1, 2018; CHOO; BALAN; LEE, 2019). O *scanner* e outros trabalhos relacionados são apresentados na Seção 2.3.

2.3 Trabalhos Relacionados

Essa seção apresenta trabalhos relacionados com esta dissertação de mestrado. Os trabalhos são divididos em: (i) Ferramentas automáticas para avaliação de acessibilidade de aplicações *web*, (ii) Ferramentas automáticas para avaliação de acessibilidade de aplicações móveis e (iii) Métodos alternativos para avaliação de acessibilidade de aplicações móveis sem uso de ferramentas automáticas.

2.3.1 Ferramentas Automáticas para Avaliação de Acessibilidade de Aplicações *Web*

Dentre as ferramentas de avaliação de acessibilidade de aplicações *web*, pode-se destacar o *Access Monitor*, o qual é uma ferramenta que avalia automaticamente a acessibilidade de páginas de *sites* (ACCESSMONITOR, 2021). O *Access Monitor* realiza a avaliação, por meio do fornecimento da URL ou do código-fonte da página *web*. Dada a entrada do código-fonte ou da URL, ocorrerá a avaliação. Ao término da avaliação, o *Access Monitor* fornece um relatório com nota variando de zero a dez para a página *web*, bem como, uma lista de erros e avisos que podem auxiliar a melhorar a acessibilidade da página *web* (ACCESSMONITOR, 2021).

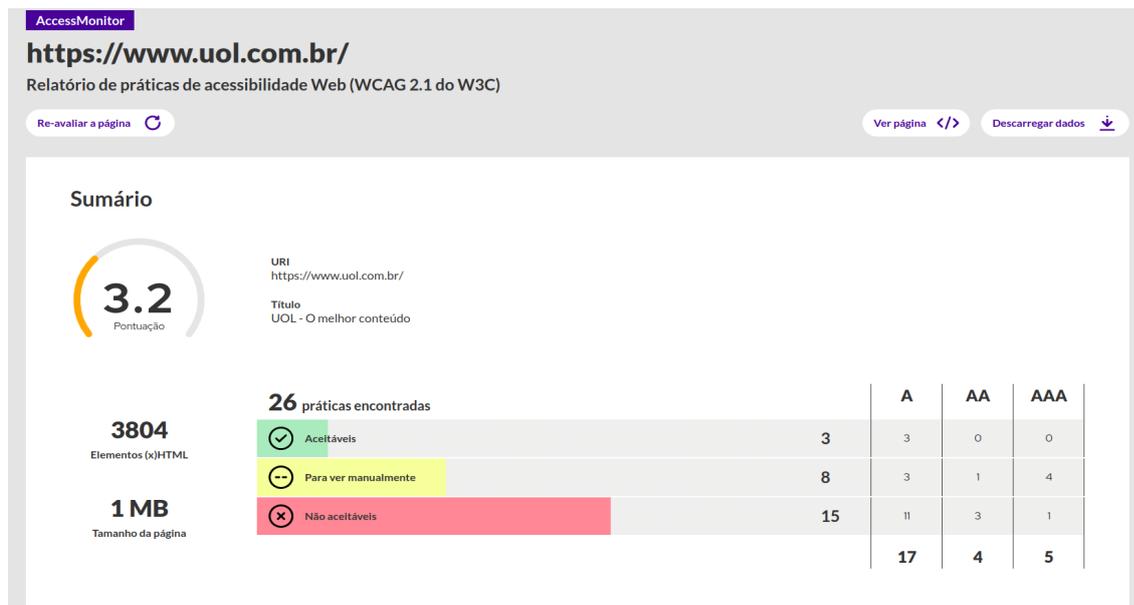
Na Figura 1 é possível observar um exemplo de avaliação de acessibilidade feito pelo *Access Monitor* com a página inicial do UOL¹. Como descrito anteriormente, a ferramenta concede uma nota a página *web* e apresenta os erros e avisos presentes nos conteúdos da página, bem como, os CSs violados, o nível de conformidade de cada CS violado e os detalhes do código-fonte, no qual encontra-se o problema de acessibilidade.

Outras ferramentas de avaliação de acessibilidade, que utilizam as WCAG, podem ser encontradas em <https://www.w3.org/WAI/ER/tools/>. Por exemplo, o *Color Contrast Accessibility Validator*² é uma ferramenta que identifica problemas de acessibilidade com o contraste de cores de uma página *web*, por meio de uma avaliação seguindo as diretrizes WCAG 2.1. O contraste de cor refere-se a como as cores claras ou escuras contrastam

¹ <https://www.uol.com.br>

² <https://color.a11y.com/?wc3>

Figura 1 – Interface de usuário do Access Monitor após avaliar página web do Uol.



Fonte - <https://accessmonitor.acessibilidade.gov.pt/>.

entre si. Quando se trata de acessibilidade de aplicações *web* ou móveis, o contraste entre o texto e o seu plano de fundo é uma preocupação para usuários que possuem problemas visuais, especialmente para os daltônicos (A11Y, 2019).

2.3.2 Ferramentas Automáticas para Avaliação de Acessibilidade de Aplicações Móveis

Os trabalhos de KRAINZ, FEINER e FRUHMANN (2016) e ZOUHAIER, HLAOUI e AYED (2017) desenvolveram ferramentas para criação de interfaces de usuário com regras de acessibilidade embutidas para tratar acessibilidades para deficientes visuais durante a etapa de prototipação da aplicação móvel. KRAINZ, FEINER e FRUHMANN (2016) propuseram um modelo, no qual são levantados requisitos de acessibilidade, os quais são usados na criação de interfaces de usuário. O desenvolvedor cria a interface de usuário utilizando o modelo proposto e a ferramenta codifica a interface de usuário para linguagem de marcação utilizada em aplicações móveis nativas para Android.

Por outro lado, no trabalho de ZOUHAIER, HLAOUI e AYED (2017), a interface de usuário da aplicação móvel é adaptada seguindo os requisitos obtidos ao avaliar a deficiência visual do usuário. Nos trabalhos de KRAINZ, FEINER e FRUHMANN (2016), ZOUHAIER, HLAOUI e AYED (2017), a avaliação de acessibilidade de interfaces de usuário pode ser realizada durante a prototipação das interfaces de usuário da aplicação móvel.

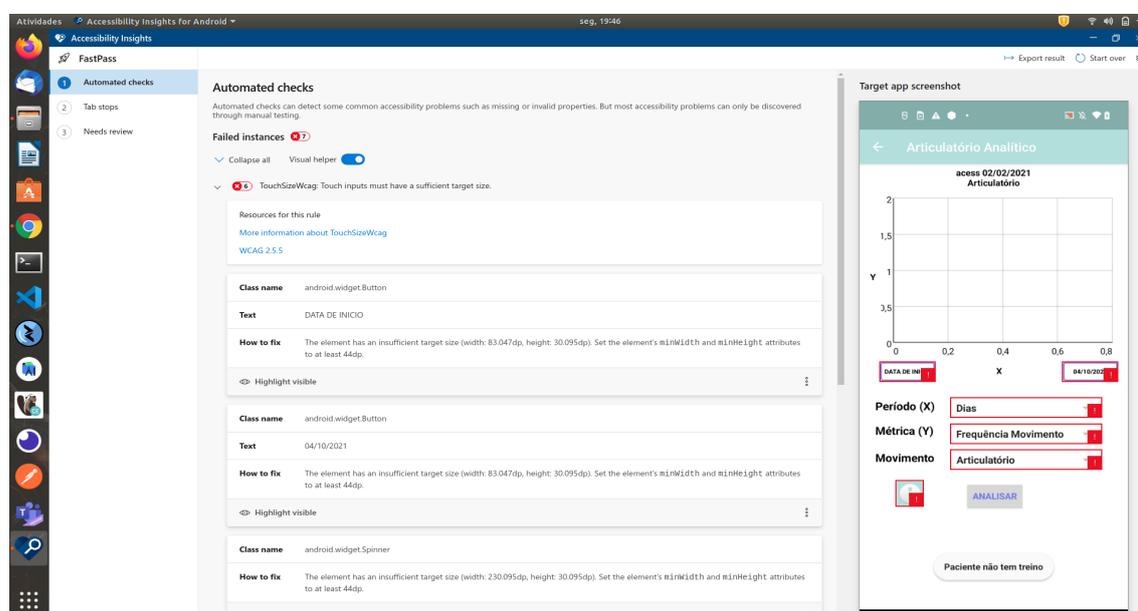
No trabalho de PATIL, BHOLE e SHETE (2016) foi desenvolvida uma funci-

onalidade para ser colocada no *software UI Automator Viewer*, presente no *Software Development Kit* (SDK) da plataforma Android, o qual é utilizado para avaliar componentes de interface de usuário. Essa nova funcionalidade teve o intuito de avaliar se as cores de componentes presentes nas interfaces de usuários estão acessíveis para daltônicos.

O trabalho de Oliveira (2019) propôs uma ferramenta de avaliação estática de código-fonte para verificar a acessibilidade de aplicações móveis nativas para Android. Nesse trabalho, o autor se apoiou em estudos na literatura e na WCAG 2.1 para levantar as regras de avaliação de acessibilidade. No trabalho de CHOO, BALAN e LEE (2019) foram examinados os efeitos do uso de um sistema proposto para apoiar os desenvolvedores de *software* a tratarem acessibilidades para deficientes visuais durante a etapa de codificação do *software*. Vale ressaltar que nos trabalhos de Oliveira (2019), CHOO, BALAN e LEE (2019) o foco foi manipular as diretrizes de acessibilidade voltadas para o público com deficiência visual.

O *Accessibility Insights*³ possui um módulo para avaliação de páginas *web* e um módulo para avaliação de aplicação móveis, os quais são voltados a ajudar os desenvolvedores de *software* a encontrarem e corrigirem problemas de acessibilidades em aplicações *web* ou móveis (ACCESSIBILITYINSIGHTS, 2021).

Figura 2 – Interface de usuário do *Accessibility Insights* após avaliar aplicação móvel do *SofiaFala - Fonoaudiólogos*.



Fonte - Autor.

Na Figura 2, pode ser observado um exemplo de avaliação de acessibilidade de uma aplicação móvel utilizando o *Accessibility Insights*. Para utilizar o *Accessibility Insights*, é necessário instalar um serviço do *Accessibility Insights* no dispositivo móvel e instalar uma aplicação no computador pessoal para realizar as avaliações de acessibilidade. Após isso,

³ <https://accessibilityinsights.io/>

conecta-se o dispositivo móvel ao computador pessoal, por meio de um cabo USB, e, ao clicar no botão de avaliação na aplicação do *Accessibility Insights*, o mesmo realiza uma captura de tela no momento do clique e apresenta as violações de acessibilidade, segundo a WCAG 2.1. O *Accessibility Insights* encontra violações de acessibilidade referentes aos CSs 1.1.1, 1.4.3, 1.4.11, 2.5.5 e dentre outras, mas, apenas as quatro citadas serão consideradas nesta dissertação para fins de análise.

O *scanner* de acessibilidade da Google avalia acessibilidade de uma aplicação móvel levando em consideração os quatro critérios a seguir:

- **Rótulos de conteúdo:** Verifica-se a existência de textos alternativos em *ImageViews*, *ImageButtons*, *CheckBoxs*, *EditTexts*, *TextViews* e dentre outras *Views* que transmitam informações ou realizem ações. Assemelha-se ao CS.1.1.1 Conteúdo não textual da WCAG 2.1.
- **Tamanho da área de toque:** Verifica-se a existência de componentes que têm ação de toque e que possuem no mínimo 48 de altura e 48 dp de largura. Assemelha-se ao CS.2.5.5 Tamanho do Alvo da WCAG 2.1.
- **Itens clicáveis:** Avalia-se a hierarquia presente na interface de usuário, a fim de verificar se o usuário pode ser prejudicado na realização de uma ação, devido a algum problema de codificação da interface de usuário como, por exemplo, quando há componentes sobrepostos.
- **Contraste de textos e imagens:** É verificado se a taxa de contraste entre texto e seu fundo possuem no mínimo 4,5:1 para textos pequenos (abaixo de 18 dp, normal, ou 14 dp, negrito) ou 3,0:1 para textos grandes (acima de 18 dp, normal, ou 14 dp, negrito). Assemelha-se aos CSs 1.4.3 Contraste (mínimo) e 1.4.11 Contraste sem texto da WCAG 2.1.

Uma investigação prévia com o *scanner* de acessibilidade da Google foi realizada, a qual é descrita na Seção 3.1.1.

2.3.3 Métodos Alternativos para Avaliação de Acessibilidade de Aplicações Móveis sem Uso de Ferramentas Automáticas

No trabalho de ANGGANANON, WALD e GILBERT (2015), foi realizado um estudo que apresenta um *framework*, o qual tem como intuito apoiar o desenvolvedor na tomada de decisão para criar aplicações acessíveis. O *framework* auxilia a organização dos requisitos

de interação da aplicação. Esse *framework* pode ser aplicado em fases iniciais de um projeto para definir os requisitos da aplicação.

O projeto *Websites Atendendo a Requisitos de Acessibilidade e Usabilidade* (WARAU) teve início com um *website* com o objetivo de ser um espaço de discussão sobre normas, diretrizes, técnicas e boas práticas para a criação de interfaces de usuário de aplicações *web* acessíveis e usáveis (SANTANA; ALMEIDA; MARIA, 2018). O WARAU envolveu um estudo realizado com uma equipe de desenvolvimento de sistemas *web* da UNICAMP em que foram identificadas dificuldades e desafios relacionados ao desenvolvimento *web* acessível combinando HTML, CSS, JavaScript, Acessibilidade e Usabilidade. Como forma de lidar com esses desafios, o WARAU foi criado e disponibilizado para a comunidade de mantenedores de *websites* (desenvolvedores, designers e redatores). O *website* não está mais disponível, mas seu conteúdo foi revisitado e reestruturado em formato de livro⁴.

2.3.4 Considerações Finais

As considerações retiradas por meio da leitura e análise desses trabalhos e ferramentas é de que existem abordagens que tratam a acessibilidade da aplicação nos estágios de levantamento e análise requisitos do projeto, ou seja, antes de iniciar a codificação, como no trabalho de ANGKANANON, WALD e GILBERT (2015). Por outro lado, existem abordagens que avaliam acessibilidade no processo de prototipação da aplicação móvel (antes de entrar na codificação), como no caso dos trabalhos de KRAINZ, FEINER e FRUHMANN (2016), ZOUHAIER, HLAOUI e AYED (2017).

Ademais, há ferramentas e abordagens que fazem a avaliação de acessibilidade durante os estágios de prototipação e codificação da aplicação, como nos casos do *Access Monitor*, dos trabalhos de PATIL, BHOLE e SHETE (2016), Oliveira (2019), do *Accessibility Insights* e do *scanner* de acessibilidade da Google.

Nesta dissertação, a ferramenta desenvolvida, a AETool, pode avaliar a acessibilidade durante as etapas de prototipação e codificação da aplicação móvel e pode ser utilizada, quando a aplicação móvel já está finalizada, estando na etapa de manutenção do *software*. O diferencial deste trabalho em relação aos trabalhos de PATIL, BHOLE e SHETE (2016), Oliveira (2019), CHOO, BALAN e LEE (2019), a AETool não focou apenas em diretrizes e avaliação de acessibilidade no contexto de deficiência visual. A ideia foi tratar diferentes Critérios de Sucesso (CSs) presentes na WCAG 2.1, independente do tipo de deficiência, desde que seja possível de avaliar as recomendações do CS da WCAG 2.1, analisando o código-fonte da interface de usuário em formato *XML*.

⁴ <https://leanpub.com/warau>

AETool

Cada etapa da metodologia utilizada para construção da AETool é descrita neste capítulo na seguinte ordem: (i) elicitación e análise de requisitos, (ii) modelagem da AETool e (iii) codificação da AETool. A Seção 3.1 descreve os requisitos necessários para desenvolvimento da AETool, juntamente com uma investigação prévia utilizando o *scanner* de acessibilidade da Google. Adiante, a Seção 3.2 apresenta a modelagem lógica utilizada pela AETool para avaliar se uma aplicação móvel está seguindo as recomendações dos CSs da WCAG 2.1. O algoritmo da AETool é apresentado na Seção 3.3. Por fim, as decisões de projeto tomadas durante o processo de construção da AETool são apresentadas na Seção 3.4,

3.1 Elicitación e Análise de Requisitos

Esta seção apresenta a investigação prévia do *scanner* de acessibilidade da Google para levantar as funcionalidades e os problemas captados pelo *scanner*. Além disso, esta seção apresenta os Critérios de Sucesso (CS) da WCAG 2.1 que são abordados na AETool.

3.1.1 Investigación Prévia

Avaliações de acessibilidade em aplicações móveis utilizando o *scanner* de acessibilidade da Google foram realizadas a fim de efetuar uma investigação para levantar os problemas de acessibilidade encontrados por meio dessa ferramenta. Como pode ser observado na Figura 3, algumas telas dos aplicativos WhatsApp, Uber e Youtube foram escaneadas. A escolha dessas aplicações, para realizar essa investigação prévia, foi devido à popularidade dessas aplicações em nível mundial. Outra aplicação escaneada, foi o SofiaFala¹, o qual

¹ O SofiaFala foi construído com o objetivo de auxiliar a terapia de fala de pessoas com distúrbios fonológicos. Essa aplicação móvel possui dois módulos (Fonoaudiólogos e Criança), os quais trabalham de forma integrada. O módulo Fonoaudiólogos é voltado aos profissionais em fonoaudiologia, o qual tem o intuito de realizar cadastros de pacientes e de treinos personalizados, a produção de mídias

é um projeto desenvolvido pelo grupo de pesquisa coordenado pela orientadora desta dissertação na USP de Ribeirão Preto e objeto de estudo deste trabalho (SOFIAFALA, 2019).

Com a avaliação dos módulos Fonoaudiólogos e Criança do SofiaFala, esta dissertação poderá contribuir com o projeto SofiaFala com resultados que poderão ser utilizados no aprimoramento das interfaces de usuário dos módulos Fonoaudiólogos e Criança. Segundo Android (2020a), o *scanner* avalia acessibilidade de uma aplicação móvel levando em consideração os quatro critérios a seguir:

- **Rótulos de conteúdo:** Verifica-se a existência de textos alternativos em *ImageViews*, *ImageButtons*, *CheckBoxes*, *EditTexts*, *TextViews* e dentre outras *Views* que transmitam informações ou realizem ações. Assemelha-se ao CS.1.1.1 Conteúdo não textual da WCAG 2.1.
- **Tamanho da área de toque:** Verifica-se a existência de componentes que têm ação de toque e que possuem no mínimo 48 de altura e 48 dp de largura. Assemelha-se ao CS.2.5.5 Tamanho do Alvo da WCAG 2.1.
- **Itens clicáveis:** Avalia-se a hierarquia presente na interface de usuário, a fim de verificar se o usuário pode ser prejudicado na realização de uma ação, devido a algum problema de codificação da interface de usuário como, por exemplo, quando há componentes sobrepostos.
- **Contraste de textos e imagens:** É verificado se a taxa de contraste entre texto e seu fundo possuem no mínimo 4,5:1 para textos pequenos (abaixo de 18 dp, normal, ou 14 dp, negrito) ou 3,0:1 para textos grandes (acima de 18 dp, normal, ou 14 dp, negrito). Assemelha-se aos CSs 1.4.3 Contraste (mínimo) e 1.4.11 Contraste sem texto da WCAG 2.1.

Por fim, outras duas aplicações móveis escaneadas, foram o Sound Recorder, que é um gravador de áudios de código-fonte aberto, e o Minimal To Do, a qual é uma aplicação móvel para criação e notificação de tarefas, funcionando com uma espécie de agenda (SOUNDRECORDER., 2017; SEKHON, 2021). O Sound Recorder e o Minimal To Do são objetos de estudos desta dissertação junto ao SofiaFala.

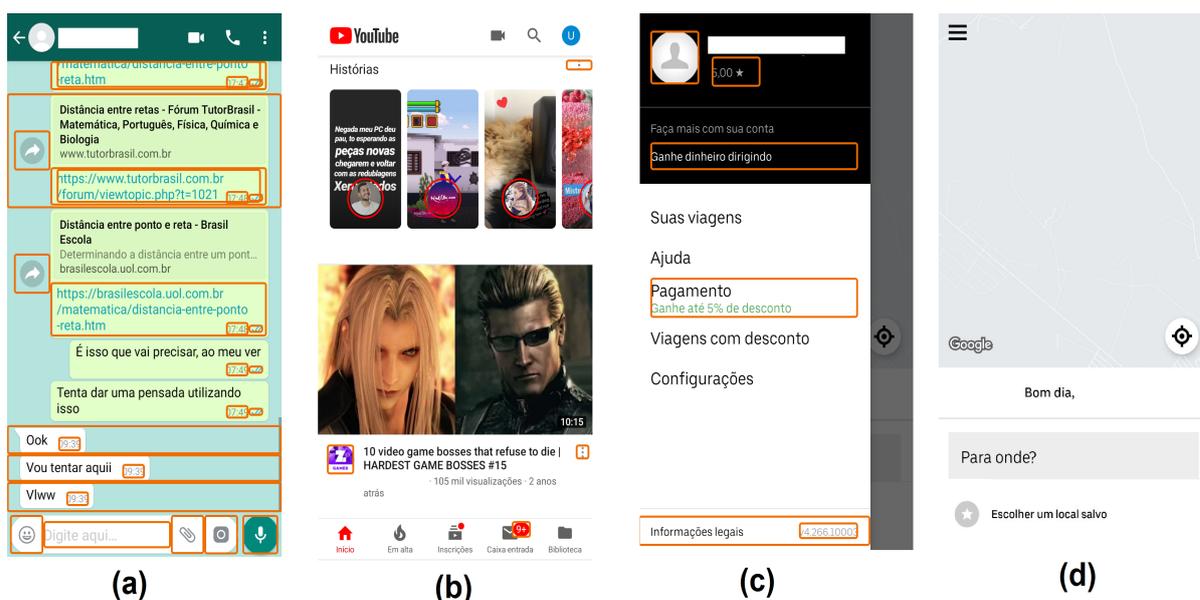
Na Figura 3, é possível observar as capturas de tela feitas pelo *scanner* de acessibilidade da Google. O *scanner* marca, com um retângulo, as áreas, as quais considerou haver problemas de acessibilidade. Para cada item marcado, é concedido uma sugestão

que demonstram como executar o treino e a avaliação de relatórios construídos a partir dos dados recebidos pelo paciente no módulo Criança. Por outro lado, o módulo Criança recebe e organiza os treinos prescritos pelo fonoaudiólogo(a), analisa os áudios e vídeos produzidos pelo paciente durante os treinos e gera e envia os resultados da performance do paciente para que possa ser visualizado pelo profissional em fonoaudiologia (SOFIAFALA, 2019).

de como melhorar a acessibilidade da interface de usuário, seguindo os critérios que o *scanner* utiliza na avaliação. A Figura 3(a) é uma captura de tela de uma conversa no WhatsApp, na qual o *scanner* concedeu trinta e uma melhorias de acessibilidade nessa tela. Na Figura 3(b), tem-se uma captura de tela da aplicação móvel do Youtube. Nessa captura de tela, o *scanner* sugeriu quatro mudanças para melhorar a acessibilidade.

Por fim, as Figuras 3(c) e (d) são capturas de tela da aplicação móvel Uber. Realizou-se a avaliação de duas capturas de telas no Uber, pois na captura de tela observada na Figura 3(d), o *scanner* não sugeriu nenhuma modificação para melhorar a acessibilidade da interface de usuário. Por outro lado, na captura de tela ilustrada na Figura 3(c), pode-se observar que o *scanner* concedeu seis sugestões. A quantidade de sugestões concedidas pelo *scanner* da Google para cada item de acessibilidade pode ser observada na Tabela 1. As sugestões concedidas pelo *scanner* são úteis para incluir acessibilidades para atender pessoas com deficiências visuais e motoras, pois, o *scanner* aborda os contrastes de textos e imagens e os tamanhos de áreas de toque de elementos que tenham ação vinculada.

Figura 3 – Capturas de tela das aplicações móveis WhatsApp(a), Youtube(b) e Uber(c)(d).



Fonte - Autor.

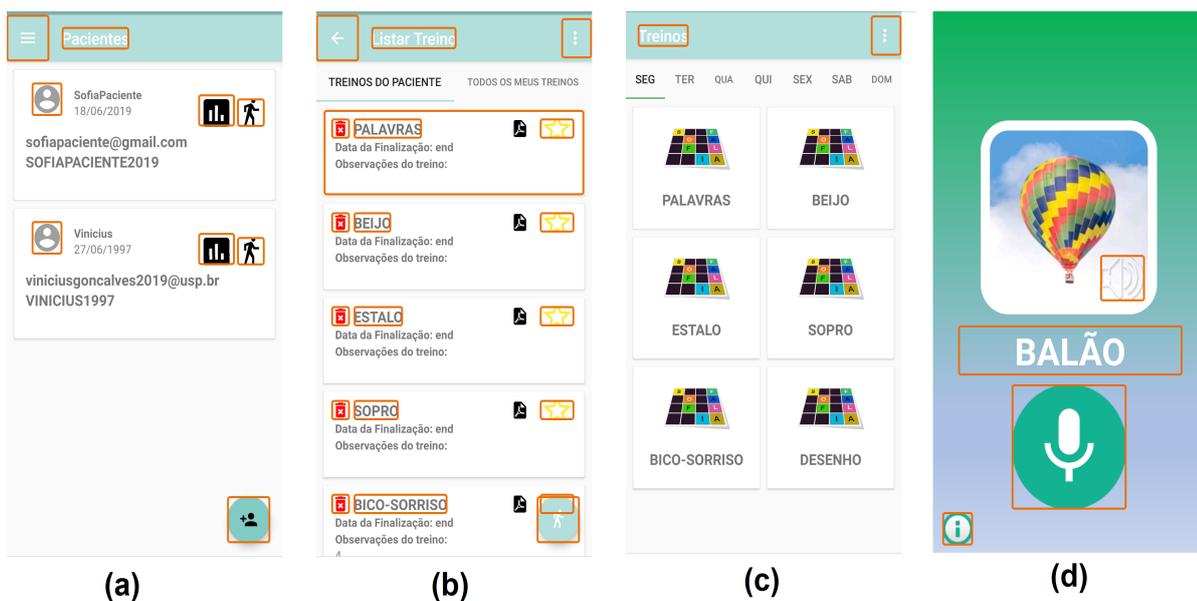
Em relação a aplicação móvel SofiaFala, nas Figuras 4(a) e (b), podem ser observados duas capturas de tela da aplicação SofiaFala - Fonoaudiólogos, as quais foram analisadas pelo *scanner*. Na captura de tela, correspondente a Figura 4(a), foram sugeridas dezoito alterações pelo *scanner*, sendo sete referentes aos rótulos dos itens, problema o qual pode ser prejudicial para pessoas com deficiências visuais. Ainda na Figura 4(a), seis problemas foram referentes as áreas de toque, as quais, segundo o *scanner*, foram consideradas pequenas para interação gestual, enquanto cinco problemas estão relacionados ao contraste de textos e imagens. Nenhuma sugestão relativa a descrição dos itens da interface de

usuário foi concedida.

Na captura de tela observada na Figura 4(b), o *scanner* sugeriu trinta e seis mudanças, consistindo de onze problemas relacionados aos rótulos dos itens, quinze referentes as áreas do toque, uma sobre a descrição do item e nove relativas ao contraste de texto e imagens.

Nas Figuras 4(c)(d) podem ser observadas as capturas de tela do SofiaFala - Criança, onze sugestões de acessibilidade foram observadas, o que faz perceber que o módulo Criança, se comparado ao Fonoaudiólogos, possui menos problemas de acessibilidade, segundo o *scanner* da Google. Percebe-se que o SofiaFala precisa passar por correções a fim de conceder mais acessibilidades aos usuários, principalmente no módulo Fonoaudiólogos. Observou-se que o módulo Fonoaudiólogos recebeu uma quantidade considerável de sugestões para melhoria da acessibilidade, principalmente, no que refere-se à área de toque.

Figura 4 – Capturas de tela das aplicações móveis do SofiaFala, módulos Fonoaudiólogos e Criança.



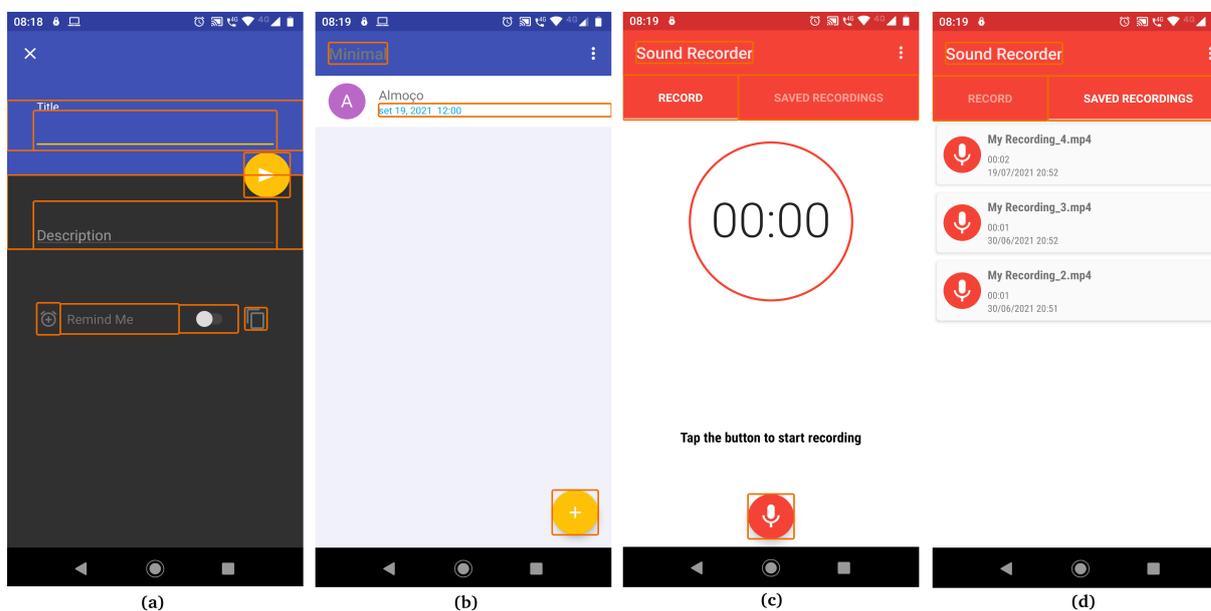
Fonte - Autor.

Em relação a aplicação móvel Minimal To Do, nas Figuras 5(a) e (b), podem ser observados duas capturas de tela da aplicação, as quais foram analisadas pelo *scanner*. Na captura de tela, correspondente a Figura 5(a), foram sugeridas dez melhorias de acessibilidade pelo *scanner*, enquanto que na captura de tela observada na Figura 5(b), foram sugeridas quatro melhorias.

No que diz respeito a aplicação móvel Sound Recorder, nas Figuras 5(c) e (d), podem ser observados duas capturas de tela da aplicação analisadas pelo *scanner*. Na captura de tela, correspondente a Figura 5(c), foram sugeridas quatro melhorias de acessibilidade

pelo *scanner*, enquanto que na captura de tela observada na Figura 5(d), foram sugeridas três melhorias.

Figura 5 – Capturas de tela das aplicações móveis Minimal To Do(a)(b) e Sound Recorder(c)(d)



Fonte - Autor.

Tabela 1 – Sugestões de acessibilidade, segundo o *scanner* de acessibilidade da Google.

Captura de Tela	Rótulos do item	Área do toque	Descrições de itens	Contraste do texto ou imagem
Figura 3(a) WhatsApp	0	9	4	19
Figura 3(b) Youtube	0	3	0	1
Figura 3(c) Uber	1	3	0	2
Figura 3(d) Uber	0	0	0	0
Figura 4(a) SofiaFala- Fonoaudiólogos	7	6	0	5
Figura 4(b) SofiaFala- Fonoaudiólogos	11	15	1	9
Figura 4(c) SofiaFala- Criança	0	0	0	2
Figura 4(d) SofiaFala- Criança	3	2	0	4
Figura 5(a) Minimal To Do	3	3	2	2
Figura 5(b) Minimal To Do	1	0	0	3
Figura 5(c) Sound Recorder	1	0	0	3
Figura 5(d) Sound Recorder	0	0	0	3

Fonte - Autor.

Com essa investigação prévia, foi possível analisar os critérios utilizados pelo *scanner* de acessibilidade da Google e quais os componentes de interfaces de usuário de aplicações móveis para Android que o *scanner* avalia. Esse conhecimento adquirido foi útil para levantar requisitos para construção da AETool, como na avaliação dos tamanhos de componentes clicáveis, pois em aplicações *web* usa-se a unidade de *pixels*, enquanto

que em aplicações móveis nativas para Android usa-se dp^2 . Além disso, percebendo que o *scanner* da Google realiza a avaliação de uma interface de usuário por vez, necessitando assim de um esforço manual, optou-se por implementar na AETool um mecanismo que possibilitasse avaliar todas as interfaces de usuário de uma aplicação móvel de uma só vez, podendo assim, trazer rapidez à avaliação.

3.1.2 Critérios de Sucesso Selecionados

Os Critérios de Sucesso (CS) selecionados para serem implementados na AETool são descritos nesta seção, seguindo a definição da WCAG2.1 (2018). A diretriz, o nível de conformidade e o público de deficientes beneficiados de cada CS selecionado podem ser observados na Tabela 2.

Tabela 2 – CSs selecionados para serem implementados na AETool.

CS	Diretriz	Nível de Conformidade	Público de Deficientes Beneficiados
1.1.1 Conteúdo não textual	1.1	A	Auditivos, Motores e Visuais
1.3.1 Informações e relacionamentos	1.3	A	Visuais
1.3.4 Orientação	1.3	AA	Motores e Visuais
1.3.5 Identificar finalidade de entrada	1.3	AA	Visuais
1.4.3 Contraste (mínimo)	1.4	AA	Visuais
1.4.6 Contraste (aprimorado)	1.4	AAA	Visuais
1.4.8 Apresentação visual	1.4	AAA	Intelectuais e Visuais
1.4.11 Contraste sem texto	1.4	AA	Visuais
2.4.2 Título da página	2.4	A	Intelectuais, Motores e Visuais
2.4.4 Objetivo do Link (no Contexto)	2.4	A	Intelectuais e Visuais
2.4.9 Objetivo do Link (Apenas Link)	2.4	AAA	Intelectuais e Visuais
2.4.10 Títulos de Seção	2.4	AAA	Intelectuais e Visuais
2.5.1 Gestos de ponteiro	2.5	A	Intelectuais e Motores
2.5.5 Tamanho do alvo	2.5	AAA	Intelectuais, Motores e Visuais
3.3.5 Ajuda	3.3	AAA	Intelectuais
4.1.1 Análise	4.1	A	Visuais

Fonte - Autor.

As avaliações da AETool podem auxiliar o desenvolvimento de interfaces de usuários

² Unidade relativa à resolução da tela. Por exemplo, se a resolução da tela é de 160 dp, então significa que 1 dp representa 1 pixel em um total de 160

acessíveis às pessoas com deficiências, auditivas, motoras, intelectuais e visuais, como pode ser observado na Tabela 2.

3.1.2.1 Conteúdo Não Textual (CS.1.1.1)

Para atender o CS.1.1.1, todo conteúdo não textual (imagens, entradas de texto, botões, elementos gráficos clicáveis e etc) deve ter uma alternativa de texto que descreva sua finalidade, exceto:

- **Controles, entrada:** Conteúdo não textual que seja um controle de entrada do usuário;
- **Mídia baseada no tempo:** Conteúdo não textual que seja uma mídia baseada no tempo;
- **Teste:** Conteúdo não textual que for um teste ou exercício que seria invalidado, caso fosse apresentado em texto;
- **Sensorial:** Conteúdo não textual destinado a criar uma experiência sensorial;
- **CAPTCHA:** Conteúdo não textual para confirmar que uma pessoa está acessando determinado conteúdo e não um computador;
- **Decoração, formatação, invisível:** Conteúdo não textual que é apenas para decoração ou que não seja visível pelo usuário.

As alternativas de texto permitem tornar as informações presentes em uma interface de usuário acessíveis, pois podem ser processadas por meio de modalidades sensoriais (visual, auditiva ou tátil) a fim de atender as necessidades do usuário. Além disso, a aplicação de alternativas de texto permite o processamento das informações por agentes de usuário (*software* que age pelo usuário, que recupera, renderiza e facilita a interação do usuário final com o conteúdo *web* ou móvel). Por exemplo, uma pessoa com deficiência visual pode fazer uso de um agente de usuário que leia a alternativa de texto de uma imagem e reproduza em voz sintetizada. Esse critério de sucesso traz benefícios para pessoas com deficiência visual, pessoas que possuem dificuldades de compreender o significados de fotografias, desenhos e outras imagens e para pessoas com deficiência auditiva.

3.1.2.2 Informações e Relacionamentos (CS.1.3.1)

A intenção do CS.1.3.1 é de preservar as informações e relações implícitas na formatação visual ou auditiva, quando o formato de apresentação for alterado. Nessa dissertação,

o CS.1.3.1 foi trabalhado juntamente com o CS.1.1.1. O CS.1.3.1 aborda outras recomendações de acessibilidade, as quais não foram tratadas nessa dissertação, devido ao fato de que não era possível avaliar as recomendações deste CS, por meio da avaliação de código-fonte *XML*. A aplicação do CS.1.3.1 pode beneficiar diferentes deficiências, especialmente, usuários cegos, os quais utilizam leitores de tela, devido as alternativas de texto.

3.1.2.3 Orientação (CS.1.3.4)

Para atender o CS.1.3.4, o conteúdo de uma interface de usuário não pode ter sua visualização e operação restringida a uma única orientação de exibição, como retrato ou paisagem, exceto que uma orientação de exibição específica seja necessária. Esse CS beneficia usuários com deficiência de destreza, que possuem um dispositivo montado em uma orientação fixa (braço de uma cadeira de rodas) e usuários com deficiência visual, pois podem visualizar o conteúdo apresentado em uma interface de usuário na orientação que funcionar melhor para eles.

3.1.2.4 Identificar Finalidade de Entrada (CS.1.3.5)

A intenção do CS.1.3.5 é assegurar que entradas de formulário, as quais coletam informações sobre o usuário, tenham sua finalidade determinada programaticamente. Por exemplo, definir que um campo do formulário precisa ter um email válido e que um campo de telefone permitirá apenas a números. Isso facilita o preenchimento de formulários, principalmente por pessoas com deficiência intelectual, pois permite que a pessoa evite digitar informações erradas. Para pessoas com deficiência motora, a necessidade de esforço manual pode ser reduzida.

3.1.2.5 Contraste (mínimo) (CS.1.4.3)

A intenção do CS.1.4.3 é garantir que a apresentação visual entre a cor do texto e do seu fundo possuam o mínimo de contraste para que possa ser lido por pessoas com deficiência visual (ex., pessoas com baixa visão ou com daltonismo). Para atender esse CS, o contraste entre o texto e o seu fundo deve ser de pelo menos 4,5:1³, exceto para os seguintes casos:

- **Texto Grande:** Em textos em grande escala, a taxa de contraste deve ser de pelo menos 3:1;

³ Razão de contraste é uma propriedade, definida como sendo a razão entre a luminância da cor mais clara e a luminância da cor mais escura

- **Incidental:** Conteúdos de interface de usuários que são apenas decorativos;
- **Logotipos:** Texto que faz parte de logotipo ou nome de marca.

3.1.2.6 Contraste (aprimorado) (CS.1.4.6)

O objetivo do CS.1.4.6 é de garantir que a apresentação visual entre a cor do texto e do seu fundo possuam contraste que permita a leitura por pessoas com deficiência visual. Para atender esse CS, o contraste entre o texto e o seu fundo deve ser de pelo menos 7:1, exceto para os seguintes casos:

- **Texto Grande:** Em textos em grande escala, a taxa de contraste deve ser de pelo menos 4,5:1;
- **Incidental:** Conteúdos de interface de usuários que são apenas decorativos;
- **Logotipos:** Texto que faz parte de logotipo ou nome de marca.

3.1.2.7 Apresentação Visual (CS.1.4.8)

O CS.1.4.8 tem como objetivo de garantir que o *layout* da aplicação não interfira na legibilidade do texto apresentado, pois pessoas com deficiências cognitivas, linguísticas, intelectuais ou visuais podem não conseguir ler o texto adequadamente, caso a apresentação do mesmo esteja dificultando a leitura. Algumas recomendações de apresentação visual são de:

- Não apresentar o texto justificado, sempre apresentá-los com alinhamento a esquerda ou a direita;
- Não haver mais de 80 caracteres ou glifos de largura (40 se CJK⁴);
- Colocar o espaçamento de linha (entrelinha) de pelo menos um espaço e meio dentro de parágrafos.

O uso correto do CS.1.4.8 traz benefícios aos usuários que tenham deficiências visuais, facilitando visualização de textos; e aos usuários com deficiências cognitivas, linguísticas, intelectuais, o uso desse CS permite que esses usuários percebam o texto e rastreiem sua localização dentro de blocos de texto. Os seguintes itens desse CS foram tratados na AETool: (i) se há justificação do texto e (ii) se o espaçamento entre linhas dentro de um paragrafo é de 1,5.

⁴ Caracteres chineses, japoneses e coreanos

3.1.2.8 Contraste sem Texto (CS.1.4.11)

O objetivo do CS.1.4.11 é de garantir que componentes da interface de usuário sejam distinguíveis por pessoas com deficiência visual. Controles de baixo contraste são mais complicados de serem percebidos por pessoas com baixa visão. A apresentação do visual dos componentes da interface de usuário devem ter uma relação de contraste de pelo menos 3.1 em relação as cores adjacentes.

3.1.2.9 Título da Página (CS.2.4.2)

A intenção do CS.2.4.2 é de auxiliar os usuários a identificarem um conteúdo presente em uma página/tela, por meio do título descritivo da mesma. Esse CS beneficia todos os usuários, pois permite a identificação rápida e fácil das informações presentes na página de um site/aplicativo. Pessoas com deficiências visuais e intelectuais são beneficiadas com a possibilidade de diferenciar o conteúdo da página por meio do título, bem como, as pessoas com deficiências motoras, cujo o modo de operação depende de uma navegação via áudio.

3.1.2.10 Objetivo do Link (no Contexto) (CS.2.4.4)

A intenção do CS.2.4.4 é de ajudar os usuários a identificarem as finalidades de *links* existentes na interface de usuário da aplicação para que possam decidir a ação que irão realizar. O CS.2.4.4 foi implementado em consonância com o CS 2.4.2, sendo empregada a mesma avaliação, pois, segundo WCAG2.1 (2018), nesse CS, o título da interface de usuário da aplicação (*web* ou móvel) seria suficiente para descrever o propósito da mesma.

O CS.2.4.4 pode beneficiar eu não tivesse os dois, eu entrava em contato para comprar kkkiciar pessoas com deficiências cognitivas e com deficiências visuais, pois os mesmos poderão determinar a finalidade da interface de usuário e não ficarão desorientados em relação ao que é o conteúdo mostrado na interface de usuário.

3.1.2.11 Objetivo do Link (Apenas Link) (CS.2.4.9)

A intenção do CS.2.4.9 é de ajudar os usuários a identificarem as finalidades de *links* existentes na interface de usuário da aplicação para que possam decidir a ação que irão realizar. O CS.2.4.9 foi implementado em consonância com o critério 2.4.2, sendo empregada a mesma avaliação, pois, segundo WCAG2.1 (2018), nesse CS, o título da interface de usuário da aplicação (*web* ou móvel) seria suficiente para descrever o propósito da mesma.

O CS.2.4.9 pode beneficiar pessoas com deficiências cognitivas e com deficiências visuais, pois os mesmos poderão determinar a finalidade da interface de usuário e não ficarão desorientados em relação ao que é o conteúdo mostrado na interface de usuário. A diferença do CS.2.4.9 com o CS.2.4.4 é que no caso do CS.2.4.4 leva-se em consideração que a avaliação da finalidade do *link* pode ser feita por meio do contexto do *link*, enquanto que no CS.2.4.9 é feita apenas pelo texto do *link*.

3.1.2.12 Títulos de Seção (CS.2.4.10)

O objetivo do CS.2.4.10 é de fornecer títulos para seções de uma página/tela. O uso adequado do CS.2.4.10 pode beneficiar pessoas:

- Cegas a saberem, quando trocar de página/tela;
- Com dificuldades de aprendizagem a identificarem o conteúdo mostrado na página/tela.

3.1.2.13 Gestos de Ponteiro (CS.2.5.1)

A intenção do CS.2.5.1 é de garantir que conteúdos de uma interface de usuário possam ser controlados pelo usuário por meio de uma variedade de dispositivos apontadores, ações gestuais e tecnologias assistivas. Esse CS beneficia pessoas que possuem dificuldades de realizar determinados gestos com precisão, principalmente pessoas com deficiência motora ou intelectual.

3.1.2.14 Tamanho do Alvo (CS.2.5.5)

A intenção do CS.2.5.5 é de garantir que tamanhos de itens clicáveis (campo de formulário, botões, imagens e etc) sejam grandes o suficiente para que os usuários possam interagir facilmente. Em aplicações móveis, segundo Android (2018), GooglePlay (2019), o tamanho mínimo deve ser de 48 dp de altura e 48 dp de largura.

3.1.2.15 Ajuda (CS.3.3.5)

O intuito do CS.3.3.5 é de evitar que os usuários cometam erros. Nesse sentido, um exemplo, é fazer uso de auto-completar ou especificar os caracteres permitidos em um campo de entrada de texto para que o usuário não digite caracteres inválidos. O uso adequado desse CS pode beneficiar indivíduos com deficiências de escrita, leitura e intelectual, pois tais indivíduos podem ter dificuldades de escrever em formulários.

3.1.2.16 Análise (CS.4.1.1)

A intenção do CS.4.1.1 é de garantir que o código-fonte da interface de usuário esteja escrito adequadamente, ou seja, que *tags* e marcadores estejam fechados corretamente e que não haja componentes com os mesmos identificadores na mesma página/tela. O uso adequado do CS.4.1.1 beneficia tecnologias assistivas como leitores de tela, para que consigam avaliar o código corretamente.

3.1.3 Critérios de Sucesso não Selecionados

Alguns CSs não foram selecionados para implementação, devido à impossibilidade de avaliar as recomendações do CS, por meio das avaliações dos códigos fonte das interfaces de usuário em formato *XML*. Essa impossibilidade foi considerada utilizando as condições a seguir:

- **Condição 1:** O elemento de interface de usuário que é avaliado no CS é uma mídia de áudio/vídeo ou arquivos.
- **Condição 2:** Não há recursos (tag, marcações e etc) para avaliação do CS via código-fonte da interface de usuário em formato *XML* de uma aplicação nativa para Android.
- **Condição 3:** É necessário uma intervenção, avaliação e interpretação técnica por parte do desenvolvedor de aplicações móveis para avaliar se a aplicação está seguindo as recomendações do CS. Necessita-se do desenvolvedor de aplicações móveis para avaliar as regras de negócio da aplicação, podendo ser necessário o tratamento das recomendações de acessibilidade via programação em Java/Kotlin.
- **Condição 4:** Necessita-se de avaliar a integração da aplicação com *hardware/software* externo.

Na Tabela 3, podem ser observados os CSs não selecionados para implementação, devido a Condição 1, pois para saber se a aplicação móvel atende as recomendações desses CSs, necessita-se da avaliação de mídias de áudio, vídeo ou arquivos. Por exemplo, avaliar se mídias de áudio/vídeo pré-gravado têm legendas ou linguagens de sinais como alternativas à mídia (CSs 1.2.2, 1.2.4, 1.2.6 e dentre outros) ou se na tela/página da aplicação móvel não há animações ou outro tipo de conteúdo que pisque mais de três vezes dentro do período de um segundo (CSs 2.3.1 e 2.3.2).

Tabela 3 – Critérios de sucesso não selecionados para implementação na AETool, devido à Condição 1

Diretriz	CS
1.2	1.2.1 Apenas Áudio e Apenas Vídeo (Pré-gravado)
1.2	1.2.2 Legendas (Pré-gravadas)
1.2	1.2.3 Audiodescrição ou Mídia Alternativa (Pré-gravada)
1.2	1.2.4 Legendas (Ao Vivo)
1.2	1.2.5 Audiodescrição (Pré-gravada)
1.2	1.2.6 Língua de sinais (Pré-gravada)
1.2	1.2.7 Audiodescrição Estendida (Pré-gravada)
1.2	1.2.8 Mídia Alternativa (Pré-gravada)
1.2	1.2.9 Apenas Áudio (Ao Vivo)
1.4	1.4.7 Áudio Baixo ou Sem Áudio de Fundo
2.3	2.3.1 Três flashes ou abaixo do limite
2.3	2.3.2 Três Flashes

Fonte - Autor.

Na Tabela 4, podem ser observados os CSs não selecionados para implementação, devido à Condição 2, pois não há recursos (tag, marcações e etc) para atender as recomendações desses CSs via código-fonte da interface de usuário em formato *XML* de uma aplicação nativa para Android. Em programação de interfaces de usuário em formato *XML* para aplicações nativas para Android, não há marcações para definir idiomas (CS.3.1.1).

Na Tabela 5, podem ser observados os CSs não selecionados para implementação, devido à Condição 3, pois para saber se a aplicação móvel atende as recomendações desses CSs, necessita-se de uma intervenção, avaliação e interpretação técnica por parte do desenvolvedor de aplicações móveis. Por exemplo, no CS.1.4.1, precisa de intervenção, por parte do desenvolvedor para avaliar se há o uso de cores para indicar os campos obrigatórios em formulários, pois só com uma análise das regras de negócio será possível dizer quais campos devem ser obrigatórios. No CS.1.3.2, precisa-se do desenvolvedor para avaliar as posições dos conteúdos em uma página.

Tabela 4 – Critérios de sucesso não selecionados para implementação na AETool, devido à Condição 2

Diretriz	CS
1.4	1.4.12 Espaçamento do Texto
2.1	2.1.1 Teclado
2.1	2.1.2 Sem armadilha de teclado
2.1	2.1.3 Teclado (sem exceção)
2.1	2.1.4 Atalhos-chave de caracteres
3.1	3.1.1 Idioma da página
3.1	3.1.2 Linguagem das peças

Fonte - Autor.

No CS.2.2.1, é necessário do desenvolvedor de *software* para avaliar se o tempo do usuário para realizar determinadas ações estão ajustáveis ou razoáveis, permitindo que o usuário tenha tempo suficiente para realizar uma ação (seja de leitura, preenchimento de formulários e etc). No CS.2.3.3, é necessário de intervenção do desenvolvedor para avaliar as animações a fim de verificar juntamente com as recomendações do CS, se determinada animação pode causar problemas de acessibilidade. Por fim, mais alguns exemplos de CSs que necessitam de intervenção do desenvolvedor de *software* são os referentes a identificação, tratamento e prevenção de erros (CSs 3.3.1, 3.3.3 e 3.3.4). Nesses CSs, o desenvolvedor precisa avaliar as regras de negócio da aplicação, juntamente com as recomendações de acessibilidade do CS, para poder criar mecanismos para identificação, prevenção e tratamento de erros em ações de transações bancárias, preenchimento e submissão de formulários e dentre outras situações.

Tabela 5 – Critérios de sucesso não selecionados para implementação na AETool, devido a Condição 3

Diretriz	CS	Diretriz	CS
1.3	1.3.2 Sequência com Significado	2.4	2.4.7 Foco Visível
1.3	1.3.3 Características Sensoriais	2.4	2.4.8 Localização
1.3	1.3.6 Identificar Objetivo	2.5	2.5.2 Cancelamento de Ponteiro
1.4	1.4.1 Uso de Cor	2.5	2.5.3 Rótulo no Nome
1.4	1.4.2 Controle de Áudio	2.5	2.5.4 Atuação de Movimento
1.4	1.4.4 Redimensionar texto	3.1	3.1.3 Palavras incomuns
1.4	1.4.5 Imagens de texto	3.1	3.1.4 Abreviações
1.4	1.4.9 Imagens de texto (sem exceção)	3.1	3.1.5 Nível de leitura
1.4	1.4.10 Reflow	3.1	3.1.6 Pronúncia
1.4	1.4.13 Conteúdo em foco ou foco	3.2	3.2.1 Em Foco
2.2	2.2.1 Tempo ajustável	3.2	3.2.2 Na entrada
2.2	2.2.2 Pausar, Parar, Ocultar	3.2	3.2.3 Navegação consistente
2.2	2.2.3 Sem tempo	3.2	3.2.4 Identificação Consistente
2.2	2.2.4 Interrupções	3.2	3.2.5 Alteração mediante solicitação
2.2	2.2.5 Reautenticação	3.3	3.3.1 Identificação de Erro
2.2	2.2.6 Timeouts	3.3	3.3.2 Rótulos ou instruções
2.3	2.3.3 Animação de Interações	3.3	3.3.3 Sugestão de erro
2.4	2.4.1 Blocos de contorno	3.3	3.3.4 Prevenção de erros (jurídico, financeiro, dados)
2.4	2.4.3 Ordem de Foco	3.3	3.3.6 Prevenção de erros (sem exceção)
2.4	2.4.5 Formas Múltiplas	4.1	4.1.3 Mensagens de Status
2.4	2.4.6 Títulos e rótulos		

Fonte - Autor.

Na Tabela 6 podem ser observados os CSs não selecionados para implementação, devido à Condição 4. Esses CSs não foram selecionados, pois para saber se a aplicação móvel atende as recomendações desses CSs é necessário avaliar a integração da aplicação

com *hardware e software* externo. Como, por exemplo, a integração com um mini-aplicativo Java que usa uma API de acessibilidade (CS.4.1.2) ou o uso de *hardwares* externos como entrada (teclados, mouses, tecnologias assistivas e dentre outros).

Tabela 6 – Critérios de sucesso não selecionados para implementação na AETool, devido a Condição 4

Diretriz	CS
2.5	2.5.6 Mecanismos de entrada simultânea
4.1	4.1.2 Nome, Função, Valor

Fonte - Autor.

3.1.4 Requisitos Funcionais da AETool

Os requisitos funcionais da AETool podem ser observados na Tabela 7.

Tabela 7 – Requisitos Funcionais da AETool.

Requisito	Descrição
RQ1	Possuir campos para inserir os arquivos <i>XML</i> da aplicação móvel.
RQ2	Identificar os problemas de acessibilidade da aplicação móvel seguindo os CSs apresentados na Subseção 3.1.2.
RQ3	Identificar para cada problema de acessibilidade encontrado o (i) a interface de usuário; (ii) o identificador (id) do componente da interface de usuário (caso haja); (iii) o CS não atendido, juntamente com seu nível de conformidade (A, AA, AAA); (iv) a descrição do problema de acessibilidade; (v) o trecho do código-fonte, no qual o problema de acessibilidade foi encontrado; (vi) o link redirecionando para a descrição do CS na documentação da WCAG 2.1.
RQ4	Apresentar visualmente os problemas de acessibilidade encontrados e separando-os por interface de usuário ou por CS.
RQ5	Apresentar visualmente a quantidade de problemas de acessibilidade encontrados e separando-os por CS.
RQ6	Apresentar visualmente para cada problema de acessibilidade encontrado o (i) a interface de usuário; (ii) o identificador (id) do componente da interface de usuário (caso haja); (iii) o CS não atendido, juntamente com seu nível de conformidade (A, AA, AAA); (iv) a descrição do problema de acessibilidade; (v) o trecho do código-fonte, no qual o problema de acessibilidade foi encontrado; (vi) o link redirecionando para a descrição do CS na documentação da WCAG 2.1.

Fonte - Autor.

O requisito RQ1 refere-se ao fato de haver campos para que o usuário possa inserir os arquivos *XML* da aplicação móvel na AETool. RQ2 e RQ3 referem-se ao processo

da AETool processar os arquivos *XML* da aplicação móvel na AETool e identificar os problemas de acessibilidade seguindo os CSs apresentados na Subseção 3.1.2. Por fim, RQ4, RQ5 e RQ6 são requisitos referentes a apresentar ao usuário tabelas, listas e detalhes sobre os problemas de acessibilidade encontrados para fins de análise.

3.2 Modelagem

Para avaliar as recomendações do CS.1.1.1 em aplicações móveis desenvolvidas nativamente para Android, a AETool avalia se há o uso da marcação *contentDescription* em componentes que possuam imagens (*ImageView* e *ImageButton*) e se há o uso da marcação *hint* em campos de entrada de texto (*EditText*). Além disso, a AETool avalia se há o uso de campos de texto (*TextView*) como rótulos de campos de entrada de texto, desde que o campo de texto tenha a marcação *labelFor*. Para avaliação de botões (*Button*, *RadioButton*, *ToggleButton* e *FloatingActionButton*), a AETool avalia se existe o uso da marcação *text*. Em relação a itens clicáveis, é avaliado se os mesmos possuem as marcações *contentDescription* ou *text* e, por fim, conteúdos que sejam apenas decorativos não podem ter a marcação *focusable* com valor *true*. A mesma lógica utilizada pela AETool para avaliar as recomendações do CS.1.1.1 é utilizada nas recomendações do CS.1.3.1, no que refere-se a campos de entrada de texto.

No que refere-se ao CS.1.3.4, para avaliar as recomendações desse CS por meio da AETool, foi necessário avaliar o arquivo *AndroidManifest.xml* (Manifesto Android) da aplicação. Todo projeto tem o Manifesto Android, o qual é responsável por descrever informações essenciais sobre o aplicativo para as ferramentas de compilação do Android, para o sistema operacional Android e para o Google Play (ANDROID, 2020b). Necessita-se avaliar o Manifesto Android, pois é nele que são descritas as *activities*, as quais são as interfaces de usuário. Nesse sentido, para avaliar se a interface usuário está com uma orientação fixa, a AETool avalia se há a marcação *screenOrientation* e o valor dentro dessa marcação na descrição da *activity*.

Para avaliar as recomendações dos CSs 1.3.5 e 3.3.5, a AETool avalia se os campos de entrada de texto possuem a marcação *inputType*, a qual é utilizada para definir o tipo de entrada de texto para esse campo de entrada de texto.

No que diz respeito aos CS 1.4.3 e 1.4.6, para avaliar as recomendações desse CS, a AETool captura os valores das cores (em hexadecimal) presentes nas marcações *background*, *backgroundTint* (as quais tem como função de denotar a cor de fundo de um componente) e *textColor* (a qual é utilizada para denotar a cor do texto presente em um componente). Caso o componente da interface de usuário possua *textColor*, mas não tenha a marcação *background* e *backgroundTint*, a AETool captura o *background* e

backgroundTint do componente pai (caso o componente pai tenha as marcações). Com a captura dos valores das cores, é calculado o contraste entre o texto e o seu fundo. No caso do CS.1.4.3, o contraste deve ser de pelo menos 4,5:1, enquanto no CS.1.4.6 deve ser de pelo menos 7:1.

No que refere-se os CS.1.4.8, foram tratados os seguintes itens desse CS na AETool: (i) se há justificação do texto e (ii) se o espaçamento entre linhas dentro de um parágrafo é de 1,5. Para avaliar a justificação do texto, a AETool obtém o valor das marcações *gravity* ou *layout_gravity* (utilizadas em alinhamento de texto e posição do componente) do componente de interface de usuário, caso o componente também tenha a marcação *text* (utilizada para colocar texto). Caso o valor seja *justify*, então há um problema de acessibilidade. Por outro lado, para avaliar o espaçamento entre linhas em componentes que possuíam texto, a AETool avalia o valor dentro da marcação *lineSpacingMultiplier* (responsável por marcar o valor do espaçamento entre linhas) é diferente de 1,5, caso seja, então há um problema de acessibilidade. Além disso, caso não haja identificação de espaçamento entre linhas no parágrafo, a AETool também sinaliza nos resultados de sua avaliação.

Para avaliar as recomendações do CS.1.4.11, a AETool captura os valores das cores (em hexadecimal) presentes nas marcações *background* e *backgroundTint* de componentes adjacentes. Caso o contraste dos componentes adjacentes seja menor que 3.1, a AETool acusa o problema de acessibilidade.

No que diz respeito aos CSs 2.4.2, 2.4.4, 2.4.9 e 2.4.10, para avaliar as recomendações desses CSs é necessário avaliar o arquivo *AndroidManifest.xml* da aplicação. Nesse sentido, para avaliar se a interface usuário tem título descrevendo seu conteúdo, a AETool avalia se há a marcação *label* na descrição da *activity* e se o valor da mesma não é nulo.

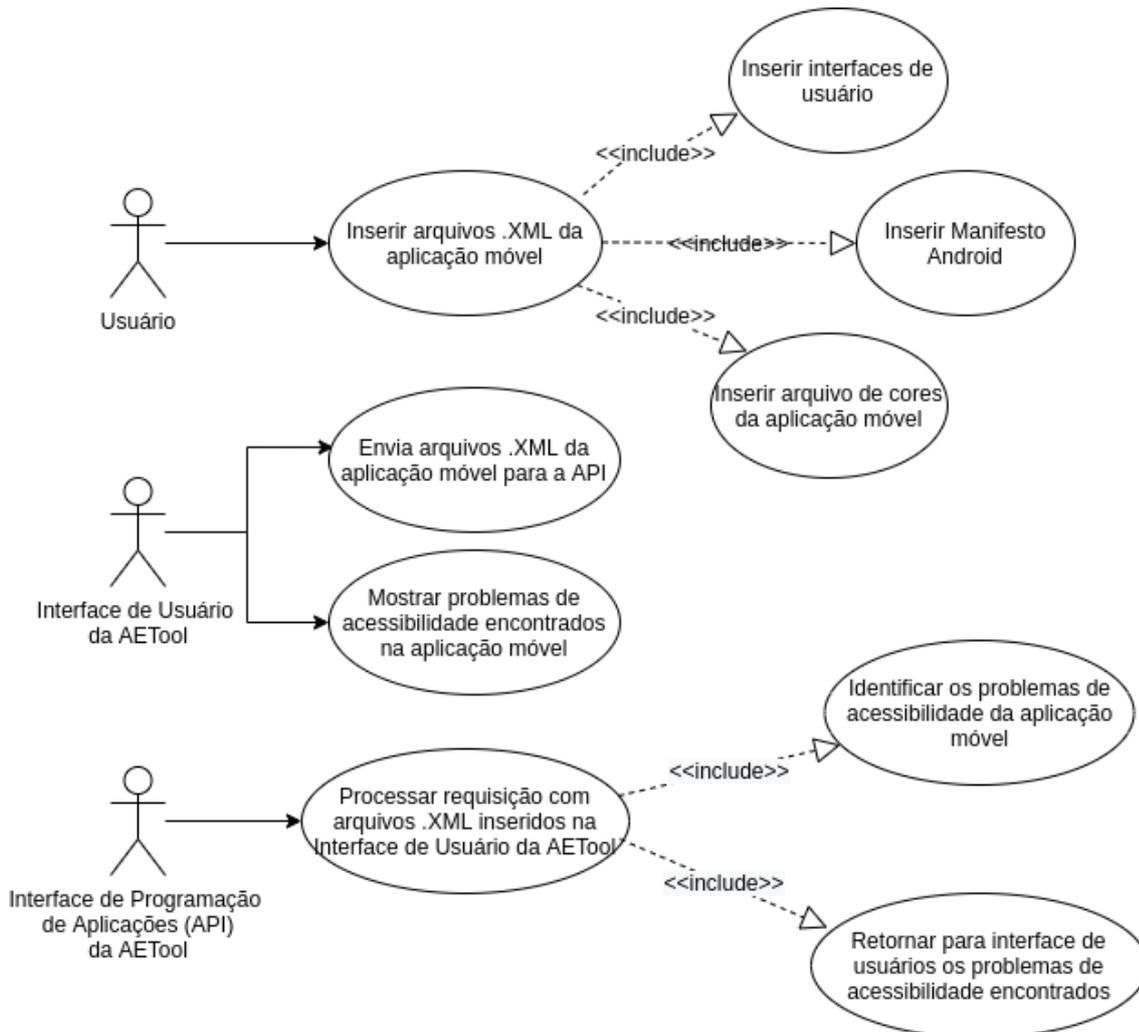
Para avaliar as recomendações do CS.2.5.1, a AETool verifica se nas interfaces de usuário há o componente *SeekBar*, pois o mesmo exige mais de um gesto para execução, podendo causar dificuldades para o usuário. No que refere-se ao CS.2.5.5, a AETool avalia a largura (*layout_width*) e altura (*layout_height*) de todos os componentes das interfaces de usuário que podem ser clicados pelo usuário para execução de ações.

Neste trabalho, a AETool avalia as recomendações do CS.3.2.4 identificando botões ou componentes de entrada de texto que possuem o mesmo texto visualmente e que tenham as mesmas alternativas de texto. Por fim, para avaliar as recomendações do CS.4.1.1, a AETool avalia a interface de usuário a fim de verificar se há componentes diferentes com o mesmo identificador.

Ademais, o diagrama de casos de uso da AETool pode ser observado na Figura 6. O usuário, o qual pode ser um desenvolvedor de *softwares* ou um especialista em IHC, insere os arquivos *XML*: (i) das interfaces de usuário da aplicação móvel nativa para Android,

(ii) do manifesto android e (iii) de cores da aplicação móvel. A Interface de Programação de Aplicações (API) da AETool processa a requisição com os arquivos *XML* enviados pela interface de usuário da AETool para identificar os problemas de acessibilidade da aplicação móvel. Por fim, a interface de usuário da AETool mostra os problemas de acessibilidade encontrados na aplicação móvel, após retorno da API.

Figura 6 – Diagrama de Casos de Uso da AETool.



Fonte - Autor.

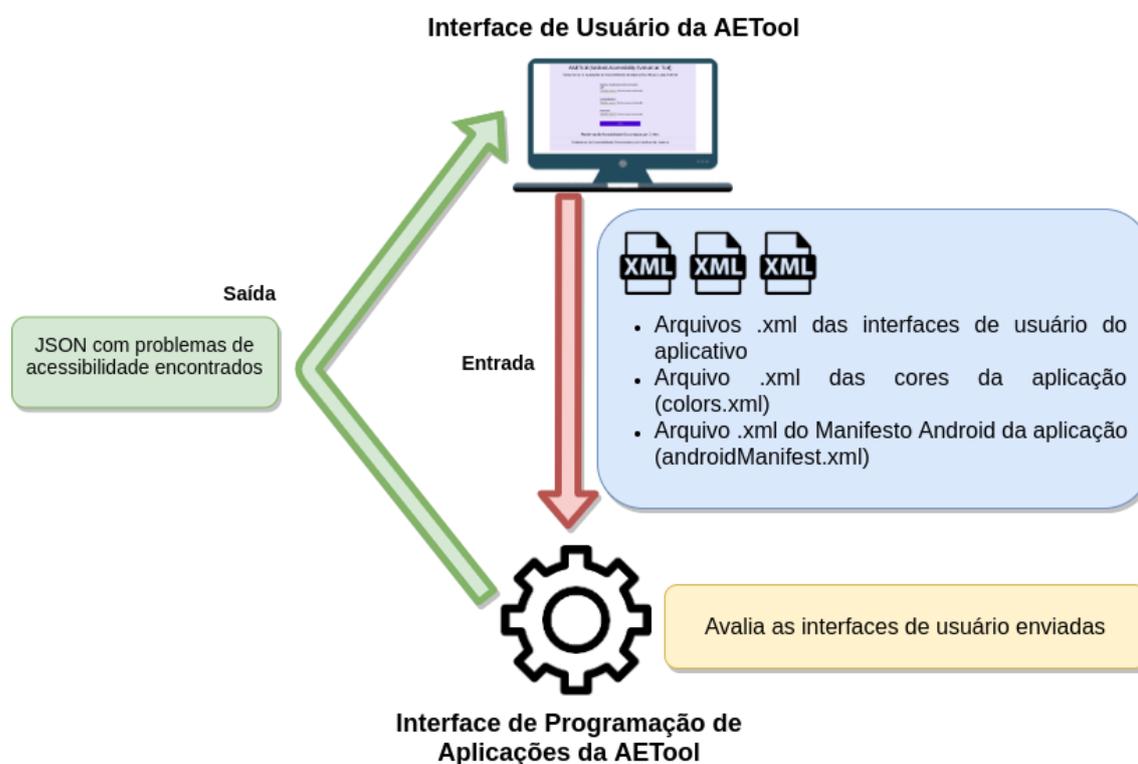
3.3 Codificação

A AETool foi construída utilizando a linguagem de programação Python e com uso do *framework* Flask⁵. A AETool possui uma Interface de Programação de Aplicações (API), como observado na Figura 7, a qual recebe uma requisição *POST* (método *Hypertext Transfer Protocol*) com: (i) os arquivos *XML* das interfaces de usuário (*activities*) da

⁵ Flask é um *framework web* escrito em Python (<https://flask.palletsprojects.com/en/1.1.x/>).

aplicação a ser avaliada; (ii) o arquivo *XML* do Manifesto Android (*androidManifest.xml*), no qual as *activities* são descritas; (iii) o arquivo *XML* de cores da aplicação (*colors.xml*), no qual são definidas as cores utilizadas na aplicação.

Figura 7 – Funcionamento da AETool.



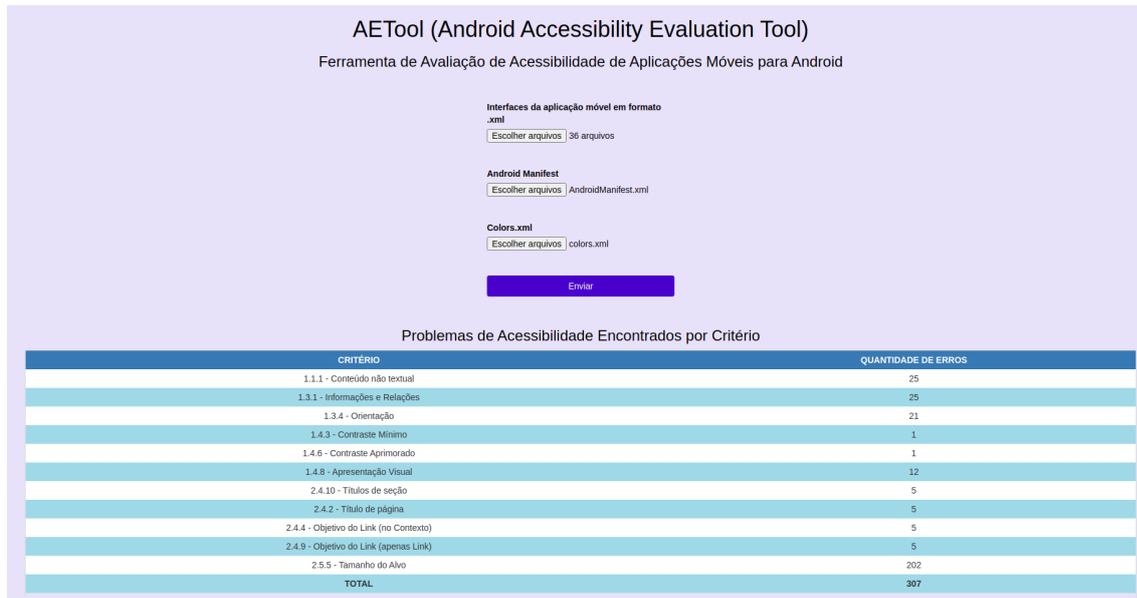
Fonte - Autor.

A AETool avalia o código-fonte de cada interface de usuário, de acordo com os CSs descritos na Subseção 3.1.2. Na Figura 7, pode ser observado o funcionamento da AETool. A AETool retorna como saída os problemas de acessibilidade encontrado(s) na(s) interface(s) de usuário, em formato JSON (*JavaScript Object Notation*), identificando: (i) a interface de usuário; (ii) o identificador (*id*) do componente da interface de usuário (caso haja); (iii) o CS não atendido, juntamente com seu nível de conformidade (A, AA, AAA); (iv) a descrição do problema de acessibilidade; (v) o trecho do código-fonte, no qual o problema de acessibilidade foi encontrado; (vi) o *link* redirecionando para a descrição do CS na documentação da WCAG 2.1.

Ao receber os arquivos *XML* das interfaces de usuário, o algoritmo presente na API da AETool avalia as recomendações de cada CS implementado (Seção 3.1.2) em cada uma das interfaces de usuário da aplicação móvel, como pode ser observado nas linhas 6 a 21 do Algoritmo 1.

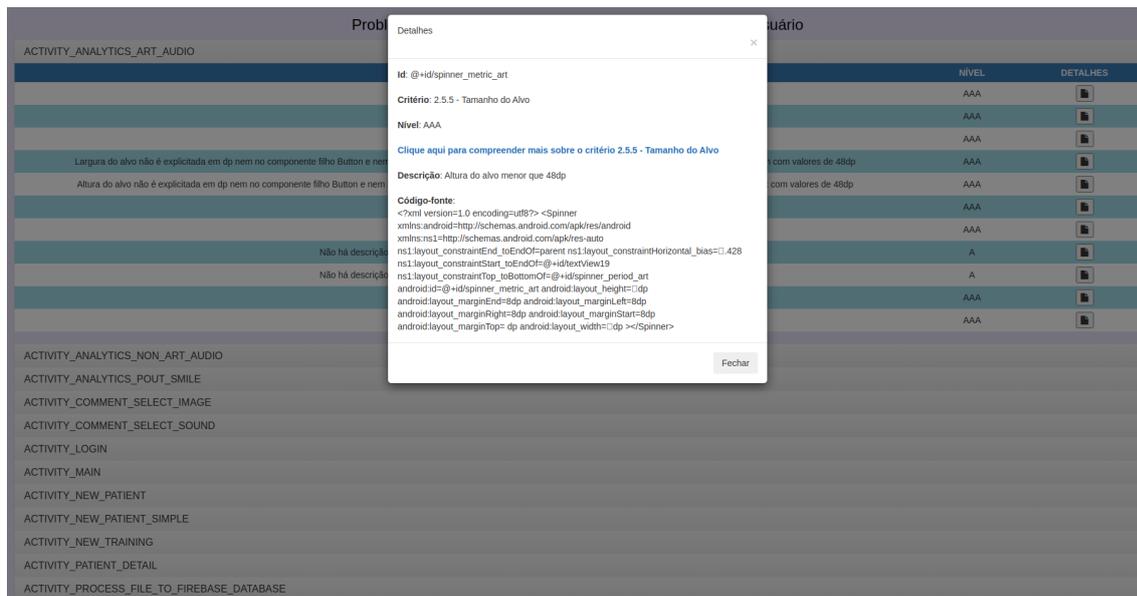
Na Figura 8 pode ser observado a interface de usuário da AETool. O usuário possui três campos para entrada de arquivos *xml*, um para colocar os arquivos de interface de usuário da aplicação móvel, um para o Manifesto Android e um para o arquivo de cores da aplicação móvel.

Figura 8 – Interface de usuário da AETool.



Fonte - Autores.

Figura 9 – AETool apresentando os problemas de acessibilidade separados por interface de usuário.



Fonte - Autores.

Como pode ser observado na Figura 8, após colocar os arquivos e clicar no botão "Enviar", a AETool avalia as interfaces de usuário da aplicação móvel seguindo os CSs apresentados na Subseção 3.1.2. Depois da avaliação, a AETool tabula os resultados e coloca a quantidade de problemas de acessibilidade encontrados por CS. Além disso, como pode ser observado na Figura 9, a AETool lista os problemas de acessibilidade por interface de usuário. Cada item da lista é nomeado com o nome do arquivo da interface de usuário da aplicação móvel e, ao clicar no item da lista, são mostrados os problemas de acessibilidade encontrados nessa interface de usuário. O usuário pode visualizar nessa

avaliação por interface de usuário, ao clicar em detalhes: (i) o identificador do componente da interface de usuário, caso tenha, (ii) o CS, (iii) o nível de conformidade do CS, (iv) o código-fonte do componente da interface de usuário e (v) a descrição do problema de acessibilidade.

Algorithm 1: Algoritmo da AETool para avaliar interfaces de usuário

```
1 Data: interfaces de usuário (.xml), androidManifest.xml, colors.xml;  
2 Result: JSON com problemas de acessibilidade encontrado(s) na(s) interface(s)  
   de usuário  
3 Início;  
4 lê androidManifest.xml e colors.xml;  
5 JSON = [ ];  
6 while tiver interfaces de usuário do  
7   lê código da interface de usuário;  
8   i = 0;  
9   while tiver CSs para serem avaliados do  
10    if interface de usuário possui problema de acessibilidade em relação ao  
      CS[i] then  
11      JSON[ ][idComponent] = identificador (id) do componente da interface  
        de usuário;  
12      JSON[ ][criterio] = CS[i];  
13      JSON[ ][nivel] = nível de conformidade do CS[i];  
14      JSON[ ][description] = descrição do problema de acessibilidade;  
15      JSON[ ][arq] = nome da interface de usuário;  
16      JSON[ ][link] = link redirecionando para a descrição do CS no manual  
        da WCAG 2.1;  
17      JSON[ ][component] = trecho do código-fonte que encontra-se o  
        problema de acessibilidade.;  
18    end  
19    i++;  
20  end  
21 end
```

3.4 Considerações Finais

Com a investigação prévia apresentada na Seção 3.1.1, foi possível analisar os critérios utilizados pelo *scanner* de acessibilidade da Google e quais os componentes de interfaces de usuário de aplicações móveis para Android que o *scanner* avalia. Esse conhecimento adquirido foi útil para levantar requisitos para construção da AETool, como na avaliação dos tamanhos de componentes clicáveis, pois em aplicações *web* usa-se a unidade de *pixels*, enquanto que em aplicações móveis nativas para Android usa-se *dp*. Além disso, percebendo que o *scanner* da Google realiza a avaliação de uma interface de usuário por vez, necessitando assim de um esforço manual, optou-se por implementar na AETool um mecanismo que possibilitasse avaliar todas as interfaces de usuário de uma aplicação móvel de uma só vez, podendo assim, trazer rapidez a avaliação.

Selecionou-se apenas os CSs da WCAG 2.1 para serem avaliados pela da AETool, caso não houvesse alguma impossibilidade de avaliar as recomendações do CS, por meio das avaliações dos códigos fonte das interfaces de usuário em formato *XML*. Essa impossibilidade foi considerada utilizando as condições a seguir:

- **Condição 1:** O elemento de interface de usuário que é avaliado no CS é uma mídia de áudio/vídeo ou arquivos.
- **Condição 2:** Não há recursos (tag, marcações e etc) para avaliação do CS via código-fonte da interface de usuário em formato *XML* de uma aplicação nativa para Android.
- **Condição 3:** É necessário uma intervenção, avaliação e interpretação técnica por parte do desenvolvedor de aplicações móveis para avaliar se a aplicação está seguindo as recomendações do CS. Necessita-se do desenvolvedor de aplicações móveis para avaliar as regras de negócio da aplicação, podendo ser necessário o tratamento das recomendações de acessibilidade via programação em Java/Kotlin.
- **Condição 4:** Necessita-se de avaliar a integração da aplicação com *hardware/software* externo.

Nessa dissertação, optou-se em focar no desenvolvimento de uma ferramenta para avaliação de acessibilidade para aplicações móveis desenvolvidas nativamente para o sistema operacional Android. Além disso, selecionou-se CSs, independente do tipo de deficiência, que os mesmos abordassem, desde que fosse passível de avaliar as recomendações do CS da WCAG 2.1 avaliando o código-fonte da interface de usuário em formato *XML*.

Prova de Conceito com AETool

Este capítulo apresenta as aplicações móveis que foram utilizadas para a Prova de Conceito (POC) (Seção 4.1), a metodologia aplicada para comparar a AETool em relação ao julgamento de humanos (Seção 4.2) e a metodologia aplicada para comparar a AETool em relação a outras ferramentas (Seção 4.3).

Os dados obtidos na etapa de comparar a AETool com o julgamento de humanos e com outras ferramentas foram normalizados utilizando a frequência relativa das violações de acessibilidade encontradas pelos desenvolvedores de *software*, pela AETool ou pelas outras ferramentas em relação a quantidade de violações de acessibilidade realmente existentes na aplicação móvel. O padrão ouro consistiu de uma avaliação manual para encontrar todas as violações de acessibilidade de uma aplicação móvel, utilizando a WCAG 2.1. No padrão ouro, para não haver subjetividade foram calculados, por exemplo, a altura e largura de áreas de toque e o contraste entre cores para ter um valor real.

A quantidade de violações de acessibilidade encontradas por desenvolvedores de *software*, pela AETool e pela as outras ferramentas foram divididas em violações de acessibilidade encontradas (violações julgadas) e violações de acessibilidade encontradas que fazem parte do padrão ouro (violações reais).

4.1 Aplicações para Prova de Conceito

Quatro aplicações móveis foram utilizadas para Prova de Conceito (POC) com AETool, a fim de validar a eficácia da ferramenta, sendo as aplicações: (i) SofiaFala - Fonoaudiólogos¹, (ii) SofiaFala - Criança, (iii) Sound Recorder², (iv) Minimal To Do³. Todas essas aplicações citadas fizeram parte da investigação prévia realizada neste trabalho, a qual foi descrita na Subseção 3.1.1. As aplicações móveis WhatsApp, Uber e Youtube, as quais também

¹ <https://dcm.ffclrp.usp.br/sofiafala/download.php>

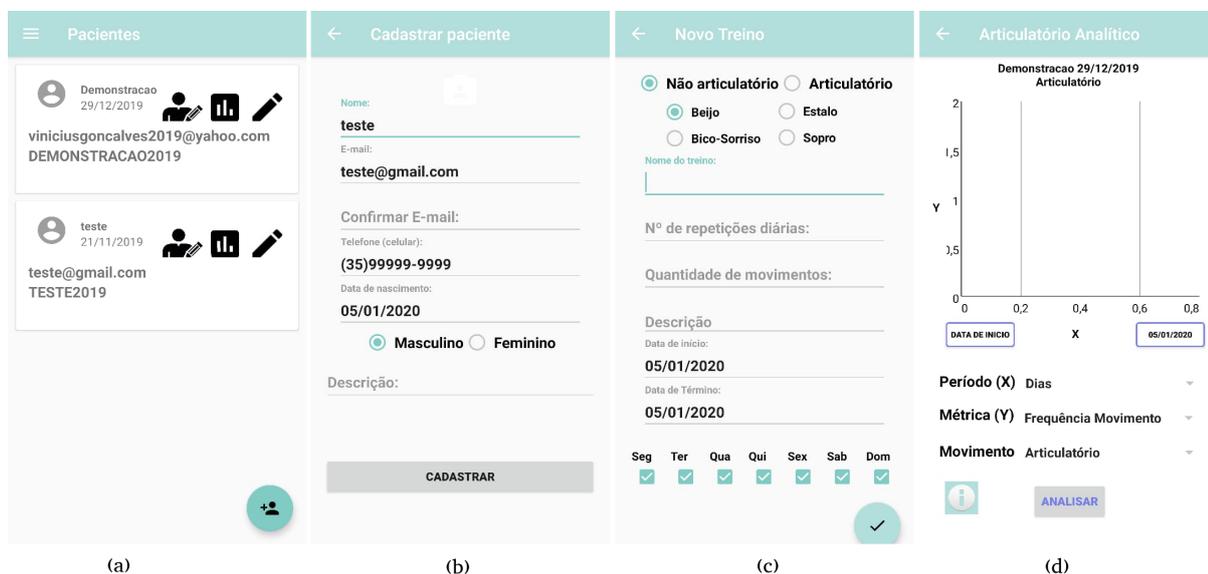
² <https://play.google.com/store/apps/details?id=com.danielkim.soundrecorder>

³ <https://play.google.com/store/apps/details?id=com.avjindersinghsekhon.minimaltodo>

fizeram parte da investigação prévia, não desempenharam parte da avaliação, devido a não terem código-fonte aberto.

As aplicações foram selecionadas devido a terem código-fonte aberto, no caso do Sound Recorder e do Minimal To Do, e também pelo grupo de pesquisa do SofiaFala ter disponibilizado os códigos fontes do SofiaFala. Além da questão da disponibilidade do código-fonte, selecionou-se aplicações que tivessem contextos diferentes. O SofiaFala - Fonoaudiólogos é uma aplicação voltada para profissionais de fonoaudiologia para prescrição de treinos, enquanto o SofiaFala - Criança é voltado para execução de treinos fonoaudiológicos. Ademais, o Sound Recorder é uma aplicação para gravação de áudios e o Minimal To Do é uma aplicação para agendamento de lembretes.

Figura 10 – (a) Tela de listagem de pacientes. (b) Tela de cadastro de paciente. (c) Tela de cadastro de treino. (d) Tela de relatório de treino articulatório.



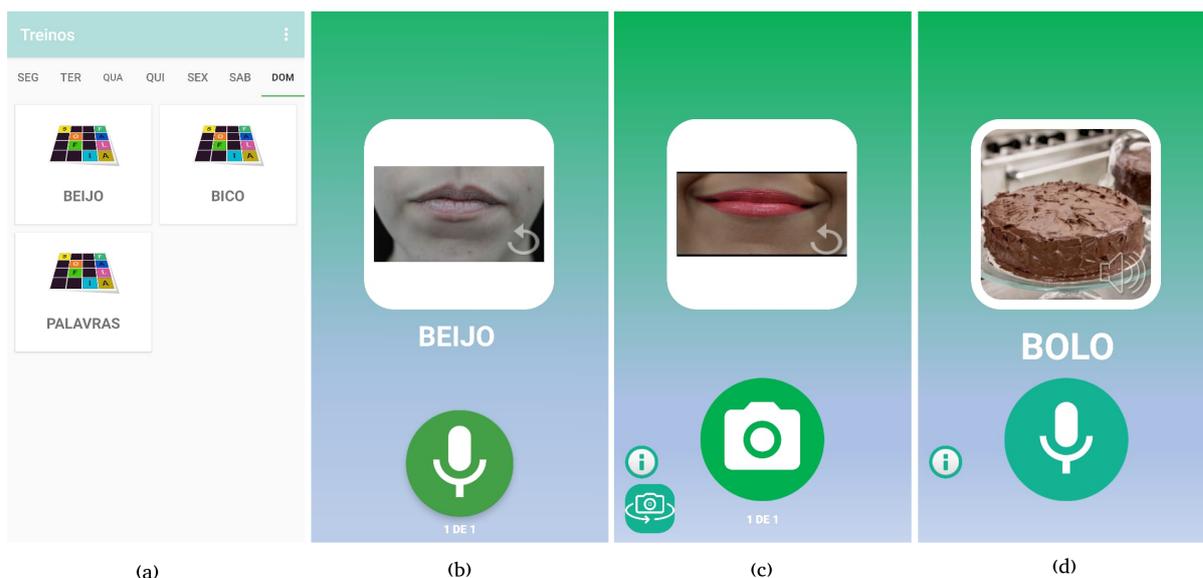
Fonte - Autores.

Na Figura 10, é possível ver algumas das interfaces de usuário do módulo Fonoaudiólogos do SofiaFala. Na Figura 10(a), pode ser observada a interface de usuário, na qual são listados os pacientes do(a) fonoaudiólogo(a). Por outro lado, nas Figuras 10(b) e (c) são possíveis de se observar as interfaces de usuário de cadastro de pacientes e de treinos, respectivamente. Por fim, na Figura 10(d), é ilustrado a interface de usuário para visualização dos relatórios dos treinos do paciente. A Figura 11 ilustra interfaces de usuário do módulo Criança. Na Figura 11(a), é ilustrada a interface de usuário que lista os treinos do(a) paciente e, nas Figuras 11(b), (c), (d) são possíveis de se observar as interfaces de usuário dos treinos de beijo, bico-sorriso e palavras, respectivamente.

Para poder realizar as avaliações de aplicações móveis com a AETool, foi necessário acesso aos códigos fontes das interfaces de usuário das aplicações. Os códigos fontes do SofiaFala foram possíveis de ter acesso, devido ao grupo de pesquisa deste presente

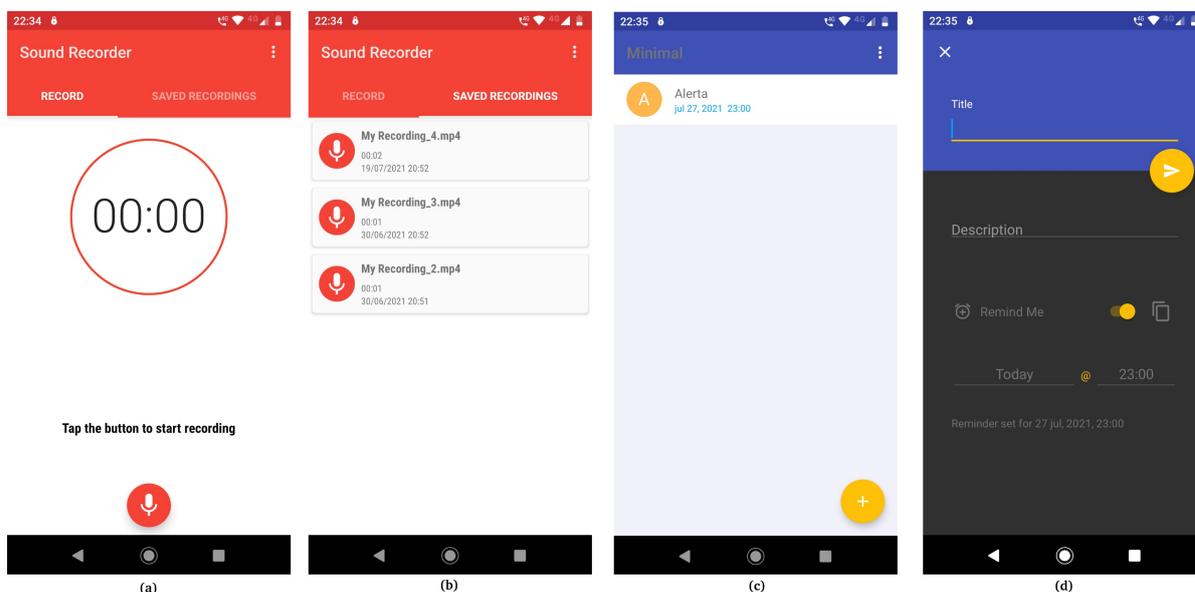
trabalho estar interligado com o projeto. Além disso, a avaliação de acessibilidade dos módulos Fonoaudiólogos e Criança permitirá ao grupo de pesquisa do projeto SofiaFala tratar possíveis problemas de acessibilidade presentes nos módulos. As outras aplicações móveis selecionadas para teste possuem seus códigos fontes abertos para a comunidade do GitHub (<https://github.com/>).

Figura 11 – (a) Tela de listagem de treinos. (b) Tela de treino de beijo. (c) Tela de treino de bico-sorriso. (d) Tela de treino de movimento articulatorio.



Fonte - Autores.

Figura 12 – (b) Tela de áudios gravados com o Sound Recorder. (c) Tela com tarefas criadas no Minimal To do. (c) Tela para cadastro de tarefas no Minimal To do.



Fonte - Autores.

A aplicação Sound Recorder é gravador de áudios de código-fonte aberto (SOUNDRECORDER., 2017). Na Figura 12(a)(b), podem ser observadas as interfaces de usuário do Sound Recorder. Outra aplicação selecionada, que também possui código-aberto, é o Minimal To Do, a qual é uma aplicação móvel criação e notificação de tarefas, funcionando como uma espécie de agenda (SEKHON, 2021). Nas Figuras 12(c)(d) podem ser observadas interfaces de usuário da aplicação Minimal To Do.

4.2 AETool *versus* Julgamento Humano

A avaliação de acessibilidade com humanos é um método manual, porém, podem ser utilizadas ferramentas para auxiliar a avaliação, como calculadoras de contraste e validadores de código-fonte. Para avaliar a acessibilidade de uma interface de usuário, o humano deve ter o conhecimento em acessibilidade (BRAJNIK; YESILADA; HARPER, 2010; BRAJNIK, 2008). Nesse sentido, o humano pode ser um pesquisador, um professor ou um desenvolvedor de *softwares* que tenha conhecimento na área de acessibilidade.

Nesta dissertação, a prova de conceito, para comparar a AETool em relação aos humanos, ocorreu com desenvolvedores de *softwares*, a qual consistiu de uma entrevista, na qual aplicou-se o método *Think Aloud*) e um *checklist* presente no APÊNDICE A. *Think Aloud* é um método usado para coletar dados, que consiste de instigar o participante a pensar em voz alta, enquanto o mesmo executa um conjunto de tarefas específicas (SUTTON; FISCHER, 2021).

O *checklist* aplicado na entrevista foi construído com base no *checklist* da WebAIM⁴, o qual possui como itens de avaliação, os CSs da WCAG 2.1.

No *checklist*, o avaliador marca uma caixa de checagem, caso a aplicação móvel esteja atendendo determinada recomendação do CS ou deixa desmarcado a caixa de checagem, caso a aplicação não esteja atendendo a recomendação do CS. Além disso, para cada CS abordado no *checklist*, o avaliador coloca quantos problemas de acessibilidade o mesmo encontrou em relação a cada CS. O *checklist* possui sete seções, as quais possuem, ao final, um espaço para o avaliador descrever problemas, críticas, recomendações e sugestões, dentro do contexto dos CSs abordados na presente seção do *checklist* e referentes à aplicação móvel.

Seis avaliadores foram convidados para avaliarem as aplicações móveis, SofiaFala - Fonoaudiólogos, SofiaFala - Criança, Sound Recorder e Minimal To Do. Na Tabela 8,

⁴ A WebAIM é uma organização sem fins lucrativos, com sede na Universidade do Estado do Utah nos Estados Unidos, que fornece soluções de acessibilidade *web* desde 1999, como treinamento, monitoramento e relatórios do *website*, certificação, consultoria, avaliação de acessibilidade de *websites* e dentre outros (WEBAIM., 2015). Devido a influência da WebAim na área de acessibilidade para *web* em contexto global, optou-se por basear no *checklist* dessa organização (<https://webaim.org/standards/wcag/checklist>).

os dados do perfil de cada avaliador podem ser observados, como idade, sexo, profissão e conhecimento de cada um nas diretrizes da WCAG 2.1 (julgado pelo próprio avaliador ao ser questionado sobre seu conhecimento), variando de 1 a 5, sendo que quanto maior o valor, maior o conhecimento. Todos os avaliadores eram desenvolvedores de *software*⁵ e cada um avaliou as quatro aplicações móveis, totalizando seis avaliações de desenvolvedores por aplicação.

Tabela 8 – Perfil dos Avaliadores.

Profissão	Anos de Profissão	Idade	Sexo	Conhecimento nas Diretrizes da WCAG 2.1
Desenvolvedor de <i>softwares</i>	5	30	Masculino	3
Desenvolvedor de <i>softwares</i>	3	24	Masculino	4
Desenvolvedor de <i>softwares</i>	3	24	Masculino	3
Desenvolvedor de <i>softwares</i>	3	26	Masculino	4
Desenvolvedor de <i>softwares</i>	4	26	Masculino	4
Desenvolvedor de <i>softwares</i>	6	32	Masculino	5

Fonte - Autor.

No *checklist*, apenas os CSs avaliados pela AETool foram abordados, pois a intenção é comparar os resultados obtidos da avaliação de acessibilidade com a AETool em relação a dos desenvolvedores de *software*. Não teria o que comparar em relação aos CSs não avaliados pela AETool. O único CS dos mencionados na Seção 3.1.2, que não foi abordado no *checklist*, foi o CS.4.1.1, pois não é necessário de ser avaliado pelos desenvolvedores de *software*, devido ao fato das próprias ferramentas para desenvolvimento nativo para Android já realizarem a tratativa das recomendações desse CS.

4.3 AETool versus Outras Ferramentas

Na prova de conceito para comparar AETool com outras ferramentas, foi realizado um comparativo entre a avaliação de acessibilidade da AETool em relação a do *scanner* de acessibilidade da Google e a do *Accessibility Insights*. Vale lembrar, que o *scanner* de acessibilidade da Google avalia aplicações móveis seguindo os quatro critérios a seguir:

- **Rótulos de conteúdo:** Verifica-se a existência de textos alternativos em *ImageViews*, *ImageButtons*, *CheckBoxes*, *EditTexts*, *TextViews* e dentre outras *Views* que transmitam informações ou realizem ações. Para fins de comparação entre a AETool e o *scanner*, comparou-se esse critério com o CS.1.1.1, devido a semelhança.

⁵ Os avaliadores selecionados são parte do grupo de trabalho do autor do presente trabalho em uma empresa privada.

- **Tamanho da área de toque:** Verifica-se a existência de componentes que têm ação de toque e que possuem no mínimo 48 de altura e 48 dp de largura. Para fins de comparação entre a AETool e o *scanner*, comparou-se esse critério com o CS.2.5.5, devido a semelhança.
- **Itens clicáveis:** Avalia-se a hierarquia presente na interface de usuário, a fim de verificar se o usuário pode ser prejudicado na realização de uma ação, devido a algum problema de codificação da interface de usuário como, por exemplo, quando há componentes sobrepostos.
- **Contraste de textos e imagens:** É verificado se a taxa de contraste entre texto e seu fundo possuem no mínimo 4,5:1 para textos pequenos (abaixo de 18 dp, normal, ou 14 dp, negrito) ou 3,0:1 para textos grandes (acima de 18 dp, normal, ou 14 dp, negrito). Para fins de comparação entre a AETool e o *scanner*, comparou-se esse critério com o CS.1.4.3, quando a violação referia-se ao contraste de texto, e com o CS.1.4.11, quando a violação arremetia ao contraste de imagens.

Realizar as comparações citadas foi possível devido a regra utilizada pela AETool e pelo *scanner* de acessibilidade da Google seguirem a mesma lógica. Por outro lado, o *Accessibility Insights* avalia os seguintes CSs em aplicação móveis: CS.1.1.1, CS.1.4.3, CS.1.4.11 e CS.2.5.5. Nesse sentido, o comparativo da AETool em relação as outras ferramentas foi em relação aos CSs 1.1.1, 1.4.3, 1.4.11, 2.5.5.

4.4 Considerações Finais

Os dados obtidos na etapa de comparar a AETool com o julgamento de humanos e com outras ferramentas foram normalizados utilizando a frequência relativa das violações de acessibilidade encontradas pelos desenvolvedores de *software*, pela AETool ou pelas outras ferramentas em relação a quantidade de violações de acessibilidade realmente existentes na aplicação móvel.

A quantidade de violações de acessibilidade encontradas por desenvolvedores de *software*, pela AETool e pela as outras ferramentas foram divididas em violações de acessibilidade encontradas (violações julgadas) e violações de acessibilidade encontradas que fazem parte do padrão ouro (violações reais).

O padrão ouro consistiu de uma avaliação manual para encontrar todas as violações de acessibilidade de uma aplicação móvel, utilizando a WCAG 2.1. O padrão ouro foi criado e definido neste trabalho a fim de ser um referência do que seriam violações julgadas e reais e também para que fosse parâmetro para normalização dos dados.

As aplicações para prova de conceito foram selecionadas devido a terem código-fonte aberto, no caso do Sound Recorder e do Minimal To Do, e também pelo grupo de pesquisa do SofiaFala ter disponibilizado os códigos fontes do SofiaFala. Além da questão da disponibilidade do código-fonte, selecionou-se aplicações que tivessem contextos diferentes. O SofiaFala - Fonoaudiólogos é uma aplicação voltada para profissionais de fonoaudiologia para prescrição de treinos, enquanto o SofiaFala - Criança é voltado para execução de treinos fonoaudiológicos. Ademais, o Sound Recorder é uma aplicação para gravação de áudios e o Minimal To Do é uma aplicação para agendamento de lembretes.

Nesta dissertação, a prova de conceito, para comparar a AETool em relação aos humanos, ocorreu com desenvolvedores de *softwares*, a qual consistiu de uma entrevista, na qual aplicou-se o método *Think Aloud*) e um *checklist*, o qual foi construído com base no *checklist* da WebAIM. Optou-se por basear no *checklist* da WebAIM, devido a referência mundial que é WebAIM na área de acessibilidade.

No *checklist*, apenas os CSs avaliados pela AETool foram abordados, pois a intenção era comparar os resultados obtidos da avaliação de acessibilidade com a AETool em relação a dos desenvolvedores de *software*, pois não teria o que comparar em relação aos CSs não avaliados pela AETool. O único CS dos mencionados na Seção 3.1.2, que não foi abordado no *checklist*, foi o CS.4.1.1, pois não é necessário de ser avaliado pelos desenvolvedores de *software*, devido ao fato das próprias ferramentas para desenvolvimento nativo para Android já realizarem a tratativa das recomendações desse CS.

Todos os avaliadores convidados eram desenvolvedores de *software*, sendo que os mesmos foram selecionados por fazerem parte do grupo de trabalho do autor do presente trabalho em uma empresa privada.

Resultados e Discussão

Este capítulo apresenta os comparativos entre os resultados das provas de conceito comparando os resultados das avaliações de acessibilidade da AETool em relação a dos humanos (Seção 5.1) e da AETool em relação a outras ferramentas (Seção 5.2).

5.1 AETool *versus* Julgamento Humano

No Apêndice B, pode ser observada a quantidade de violações julgadas e reais encontradas pela AETool e pelos humanos, a quantidade de violações que cada aplicação móvel tem (padrão ouro), a quantidade total de violações divididas por aplicação móvel e CS e dentre outras informações.

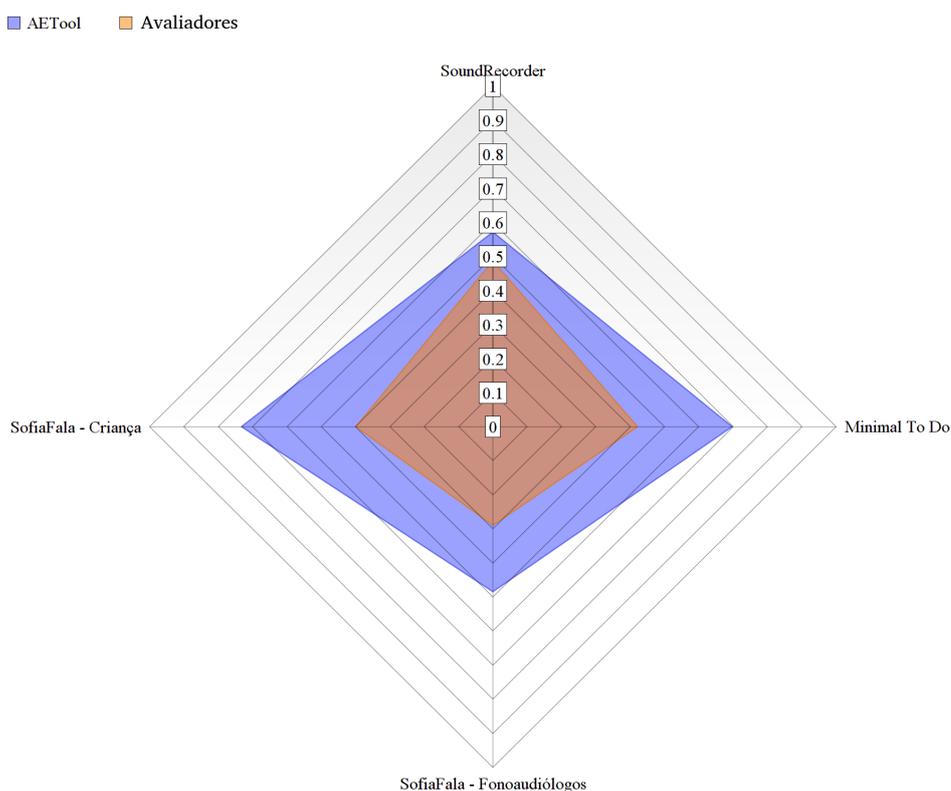
Os resultados apresentados nesta seção são em relação as violações de acessibilidade reais. Outras informações, além das violações reais, podem ser consultadas no Apêndice B. Em relação as violações reais encontradas pela AETool e pelos humanos, independente do Critério de Sucesso (CS) e da aplicação móvel, a AETool encontrou 56,53% das violações de acessibilidade existentes na soma das quatro aplicações móveis utilizadas como objetos de estudo desta dissertação. Por outro lado, os humanos encontraram em média 33,33%, tendo um Coeficiente de Variação (CV) de 6,10%. Em uma avaliação geral, não levando em consideração o CS e a aplicação móvel, o desempenho da AETool foi melhor que dos humanos.

A AETool julgou haver 340 violações de acessibilidade somando as violações encontradas em todas as quatro aplicações. Dentre as 340, 303 são violações reais, ou seja, 89,11%. Esse fator é importante de considerar, pois mostra a precisão da ferramenta em encontrar violações de acessibilidade reais.

Na Figura 13, pode ser observado as violações de acessibilidade reais encontradas pela AETool e pelos humanos em relação a cada aplicação móvel. A AETool encontrou 57,14% das violações de acessibilidade existentes no Sound Recorder, enquanto os humanos

encontraram em média de 48,64% (CV de 10,75%). Em relação a aplicação móvel Minimal To Do, a AETool cobriu 69,70% das violações de acessibilidade existentes na aplicação comparado a média de 42,17% (CV de 10,75%) dos humanos. No que refere-se ao SofiaFala - Fonoaudiólogos, aplicação com maior quantidade de violações de acessibilidade, a AETool encontrou 48,43% das violações de acessibilidade existentes, enquanto que os humanos encontraram em média de 28,96% (CV de 10,63%). Por fim, no que refere-se ao SofiaFala - Criança, a AETool encontrou 73,27% das violações de acessibilidade existentes na aplicação, enquanto os humanos encontraram em média 33,99% (CV de 7,52%). Em relação as médias dos humanos para cada aplicação móvel, o desvio padrão e coeficiente de variação foram baixos, o que indica que os dados estão próximos da média, tendo assim, uma amostra homogênea.

Figura 13 – Violações reais encontradas pela AETool e pelos humanos em relação a cada aplicação móvel



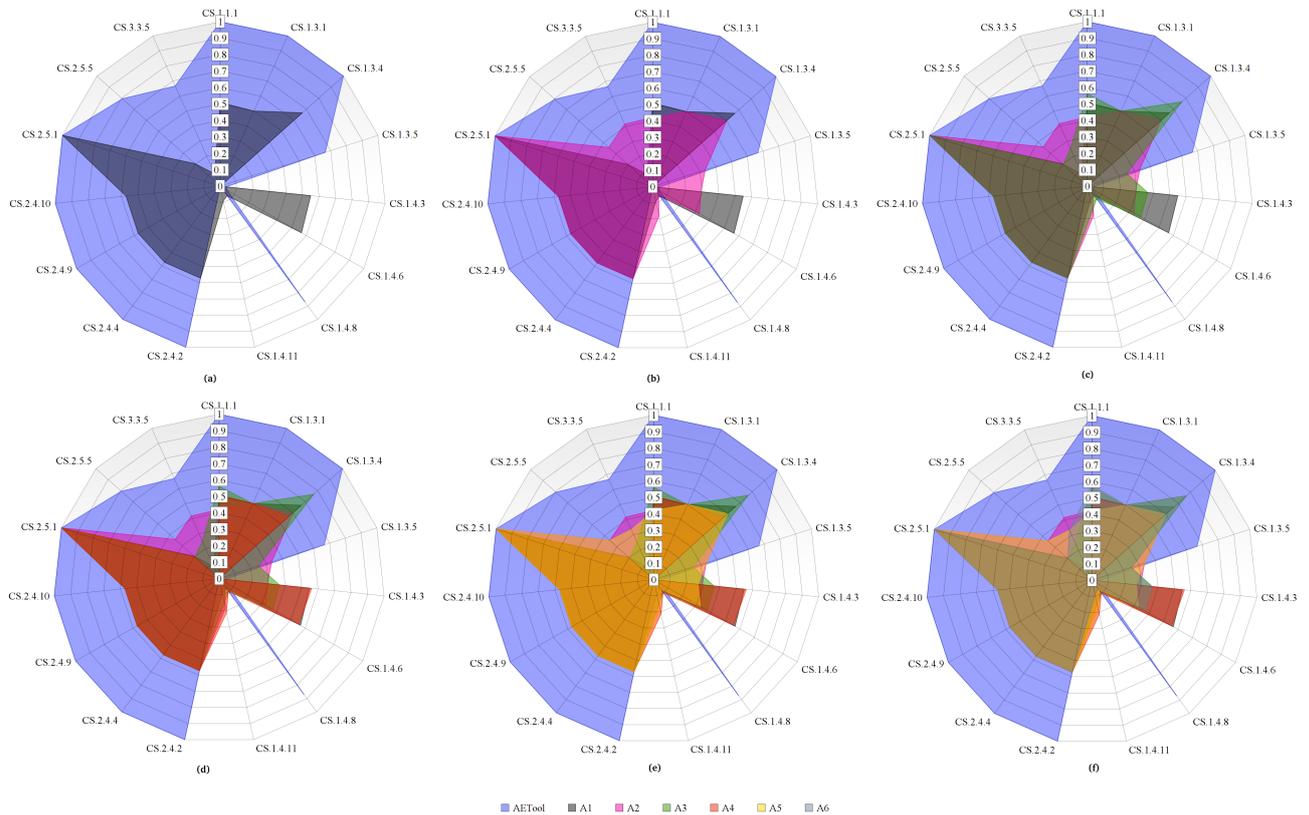
Fonte - Autor.

A AETool teve um desempenho melhor que os humanos, devido ao fato de ter encontrado mais violações de acessibilidade, independente da aplicação móvel. Com esse resultado, pode-se concluir que a AETool possui um bom desempenho para encontrar violações de acessibilidade para diferentes tipos de aplicação. A quantidade de violações de acessibilidade reais encontradas pelos humanos foi menor no SofiaFala - Fonoaudiólogos e maior no Sound Recorder (porém a AETool ainda cobriu mais violações reais de acessibilidade no Sound Recorder). Por outro lado, a quantidade de violações de acessibilidade reais

encontradas pela AETool foi menor no SofiaFala - Fonoaudiólogos e maior no SofiaFala - Criança.

O fato da AETool ter tido um melhor desempenho pode estar atrelado a capacidade da ferramenta de avaliar todos os componentes presentes no código-fonte da interface de usuário com uma regra definida que segue a WCAG 2.1, não deixando assim um problema de acessibilidade sem ser indicado, caso a regra definida na ferramenta esteja sendo violada. Por outro lado, os humanos podem ter esquecido de avaliar algum componente de interface de usuário ou podem não ter considerado que havia uma violação de acessibilidade.

Figura 14 – Violações reais de acessibilidade encontradas pela AETool e pelos avaliadores (humanos) em relação a cada aplicação móvel. **(a)** AETool e Avaliador 1. **(b)** AETool e Avaliadores 1 e 2. **(c)** AETool e Avaliadores 1, 2 e 3. **(d)** AETool e Avaliadores 1, 2, 3 e 4. **(e)** AETool e Avaliadores 1, 2, 3, 4 e 5. **(f)** AETool e Avaliadores 1, 2, 3, 4, 5 e 6.



Fonte - Autor.

Na Figura 14, pode ser observado gráficos do tipo radar com a quantidade de violações reais de acessibilidade encontradas pela AETool e pelos humanos (também denominados de avaliadores) separadas por CS e normalizadas. Para fins de análise e visualização, não se usou a média dos humanos para cada CS para fins comparativos, devido ao fato de ter ocorrido altas variações entre humanos para alguns dos CSs, como pode ser observado na Tabela 6 do Apêndice B. Nesse sentido, o resultado de cada humano

pode ser observado separadamente.

Tabela 9 – Porcentagem de violações reais de acessibilidade encontradas pelos avaliadores (humanos) e pela ferramenta (AETool) separadas por CS

CS	A1	A2	A3	A4	A5	A6	AETool
CS.1.1.1	50,88%	42,11%	56,14%	50,88%	42,11%	57,89%	100%
CS.1.3.1	50,00%	50,00%	50,00%	50,00%	50,00%	50,00%	100%
CS.1.3.4	66,67%	60,00%	76,67%	56,67%	60,00%	76,67%	100%
CS.1.3.5	0,00%	33,33%	25,00%	0,00%	33,33%	25,00%	66,67%
CS.1.4.3	54,88%	29,27%	36,59%	56,10%	26,83%	36,59%	3,66%
CS.1.4.6	56,63%	32,53%	37,35%	55,42%	32,53%	37,36%	4,82%
CS.1.4.8	7,14%	5,95%	9,52%	9,52%	7,14%	5,95%	86,90%
CS.1.4.11	9,09%	18,18%	15,15%	21,21%	18,18%	9,09%	0,00%
CS.2.4.2	57,14%	57,14%	57,14%	57,14%	57,14%	57,14%	100%
CS.2.4.4	57,14%	57,14%	57,14%	57,14%	57,14%	57,14%	100%
CS.2.4.9	57,14%	57,14%	56,14%	57,14%	57,14%	57,14%	100%
CS.2.4.10	57,14%	57,14%	56,14%	57,14%	57,14%	57,14%	100%
CS.2.5.1	100%	100%	100%	100%	100%	100%	100%
CS.2.5.5	20,53%	35,71%	19,64%	19,64%	35,71%	19,64%	79,46%
CS.3.3.5	8,83%	41,67%	25,00%	0,00%	33,33%	33,33%	66,67%

Fonte - Autor.

Na Figura 14(f), pode ser observada a quantidade de violação de acessibilidade reais encontradas pelas AETool para cada CS comparada a dos seis humanos que participaram da avaliação de acessibilidade. A AETool teve um desempenho melhor que os humanos em geral nos CSs 1.1.1, 1.3.1, 1.3.4, 1.3.5, 1.4.8, 2.4.2, 2.4.4, 2.4.9, 2.4.10, 2.5.5 e 3.3.5. A AETool teve um desempenho igual a dos humanos no CS.2.5.1, pois a AETool e os seis humanos encontraram a violação de acessibilidade, referente ao CS.2.5.1, existente na aplicação Sound Recorder (única violação dentre as quatro aplicações). Por fim, a AETool teve desempenhos piores que os humanos nos CSs 1.4.3, 1.4.6 e 1.4.8, todos referentes a análise de contraste entre cores.

A Figura 14 apresenta também diferenças nos desempenhos dos humanos. O humano 1 (A1) comparado ao 2 (A2) encontrou mais violações de acessibilidade referentes aos CSs 1.1.1, 1.3.1, 1.3.4, 1.4.3 e 1.4.6, porém, teve desempenho inferior nos CSs 1.3.5, 1.4.11, 2.5.5 e 3.3.5, como pode ser observado na Figura 14(b). A1 teve um desempenho melhor que os outros humanos também, exceto em relação ao humano 4 (A4), nos CSs 1.4.3 e 1.4.6, mostrando que o mesmo conseguiu reconhecer mais violações de acessibilidade referentes a esses CSs, junto a A4. Em relação aos perfis dos humanos, todos tinham o mesmo perfil, sendo os mesmos humanos com mais de dois anos de experiência em desenvolvimento de interfaces de usuário.

Figura 15 – Violações reais de acessibilidade encontradas pela AETool e pelos avaliadores (humanos) em relação a cada aplicação móvel e separadas por CS. (a) Sound Recorder. (b) Minimal To Do. (c) SofiaFala - Criança. (d) SofiaFala - Fonoaudiólogos.



Fonte - Autor.

Ainda em relação aos comparativos entre os humanos, A3 e A6 tiveram melhores desempenhos em relação aos outros humanos nos CSs 1.1.1 e 1.3.4. Por fim, outro resultado interessante, que pode ser citado, é que no que refere-se ao CS.3.3.5, A2 teve o melhor desempenho entre os humanos. Esses exemplos citados são alguns dentro de uma análise entre os desempenhos dos humanos. Na Tabela 9, pode ser observado a porcentagem de violações reais de acessibilidade encontradas pelos humanos e pela AETool separadas por CS.

Levando em consideração uma análise por CSs, a AETool continuou tendo melhores

desempenhos em relação aos humanos, devido ao fato de como citado anteriormente, da ferramenta fazer uma avaliação completa e detalhada de todos os componentes presentes nos códigos fontes das interfaces de usuários das aplicações, não deixando de avaliar nenhum componente ou tendo subjetividade na avaliação. Nesse sentido, é possível observar que a ferramenta consegue um bom desempenho em identificar as violações de acessibilidade dentro dos CSs que a mesma avalia, independente de qual CS seja. O único fator a ser levado em consideração, é seu baixo desempenho em relação a CSs referentes á contraste, o que pode ser uma necessidade de análise de um futuro processo de mudança na lógica aplicada na avaliação dos CSs 1.4.3, 1.4.6 e 1.4.11.

Na Figura 15, gráficos com a quantidade de violações de acessibilidade encontradas pela AETool e pelos humanos para cada aplicação móvel e separadas por CS podem ser observados. Em relação ao Sound Recorder (Figura 15(a)), a AETool teve melhores desempenhos em relação aos humanos nos CSs 1.1.1, 1.3.5, 1.4.8 e 2.5.5. Como observado anteriormente na Figura 14, a AETool teve uma dificuldade em identificar violações de acessibilidade referentes aos CSs de contraste de cores (1.4.3, 1.4.6 e 1.4.11), fato o qual se repete na Figura 15.

Por outro lado, a AETool teve bons desempenhos em identificar violações de acessibilidade em relação ao restante dos CS, tendo conseguido identificar todas as violações de acessibilidade de todas as aplicações móveis no que se diz respeito aos CSs 1.1.1, 1.3.1, 1.3.4, 2.4.2, 2.4.4, 2.4.9 e 2.4.10. No SofiaFala - Criança (Figura 15(c)), a AETool encontrou 73,26% das violações de acessibilidade existentes na aplicação móvel comparado a média de 33,99% dos humanos. Nesse sentido, a AETool surge como uma grande aliada no processo de identificar violações de acessibilidade, podendo assim, ajudar desenvolvedores de aplicações móveis a resolver tais violações.

A AETool teve bons desempenhos em identificar violações de acessibilidade referentes ao CS.2.5.5, tendo conseguido encontrar todas as violações existentes nas aplicações móveis Sound Recorder, Minimal To Do, SofiaFala - Criança e 74,73% das violações existentes no SofiaFala - Fonoaudiólogos. Fato importante, porque problemas de acessibilidade referentes ao CS.2.5.5 podem prejudicar usuários com problemas motores e visuais. Logo com a identificação das violações de acessibilidade referentes ao CS.2.5.5, o desenvolvedor pode trabalhar na correção dessas violações a fim de conceder mais acessibilidades as pessoas com deficiência.

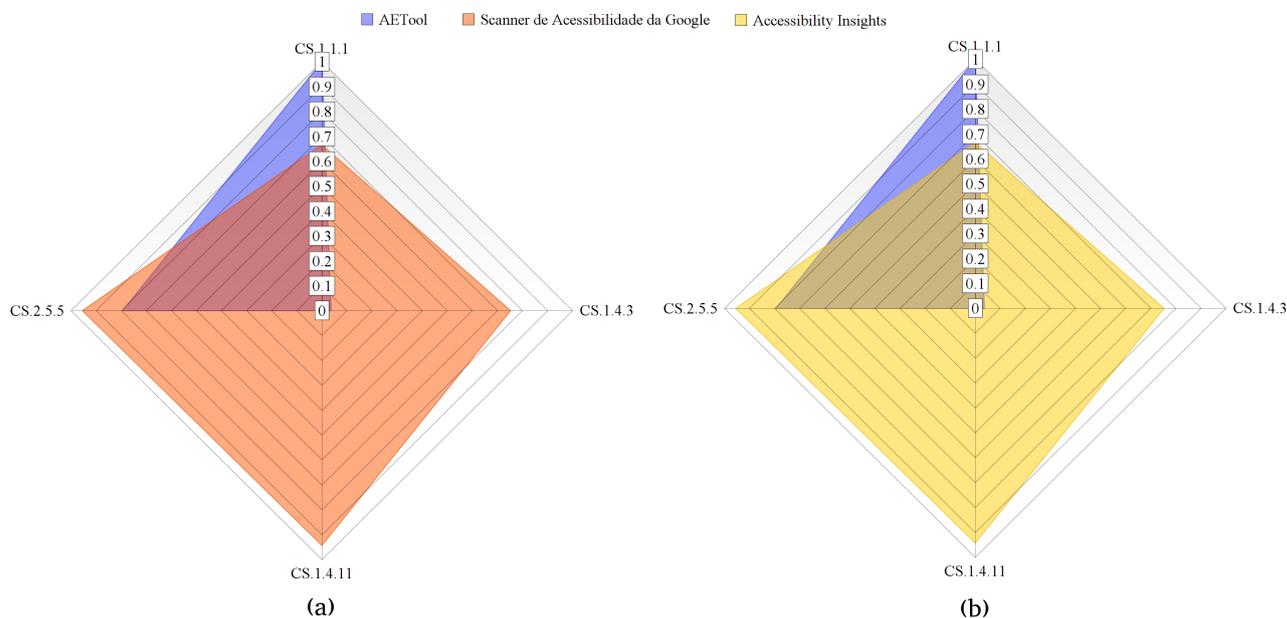
5.2 AETool *versus* Outras Ferramentas

Na Figura 16, pode ser observado gráficos com quantidade de violações reais de acessibilidade encontradas pela AETool, pelo *scanner* da Google e pelo *Accessibility Insights*

separadas por CS e normalizadas.

Como observado na Figura 16, as outras ferramentas tiveram melhores resultados em encontrar violações de acessibilidade referentes aos CSs 1.4.3 e 1.4.11, se comparado com a AETool. A AETool em relação aos CS 1.4.3, 1.4.6 e 1.4.11, referentes a contraste de texto e imagem, não obteve bons desempenhos, fator discutido anteriormente na Seção 5.1. Vale ressaltar que, tanto o *scanner* de acessibilidade da Google e o *Accessibility Insights* tiveram o mesmo desempenho para todos os CSs.

Figura 16 – Violações reais de acessibilidade encontradas pela AETool, pelo *scanner* de acessibilidade da Google e pelo *Accessibility Insights* separadas por CS. (a) AETool versus *scanner* de acessibilidade da Google. (b) AETool versus *Accessibility Insights*.



Fonte - Autor.

Em relação ao CS.1.1.1, a AETool teve um melhor desempenho em relação as outras ferramentas, lembrando que, a AETool encontrou todas as violações de acessibilidade referentes a esse CS em todas as aplicações móveis, enquanto as outras ferramentas encontraram 66,67% das violações. Por fim, em relação ao CS.2.5.5, a AETool encontrou 79,46% das violações de acessibilidade existentes nas quatro aplicações comparado a 95,53% das outras ferramentas, ou seja, as outras ferramentas tiveram um melhor desempenho em relação a esse CS.

Em termos gerais, as outras ferramentas de avaliação de acessibilidade tiveram melhores desempenhos em relação a maioria dos CSs analisados, porém, percebe-se um bom desempenho das três ferramentas em relação ao CS.2.5.5 e um desempenho melhor da AETool em relação ao CS.1.1.1. Por fim, a AETool possui um diferencial por avaliar mais CSs, se comparado as outras ferramentas.

Em termos operacionais, a AETool possibilita obter uma avaliação completa da aplicação móvel durante uma única execução, diferente das outras ferramentas, as quais avaliam uma interface de usuário por vez. Além disso, a AETool provém ao final de sua avaliação informações como: (i) Quantidade de violações de acessibilidade por CS e (ii) Quantidade de violações de acessibilidade por interface de usuário, diferente das outras ferramentas, as quais não fornecem essa informação, seria necessário um esforço manual em obtê-las. Como trabalho futuro, pode-se ser trabalhado em melhorar os resultados da AETool em relação aos CSs 1.4.3, 1.4.6 e 1.4.11, o que traria um ganho para a AETool.

5.3 Considerações Finais

A AETool encontrou 56,53% das violações de acessibilidade existentes na soma das quatro aplicações móveis utilizadas como objetos de estudo desta dissertação. Por outro lado, os humanos encontraram em média 33,33%, tendo um Coeficiente de Variação (CV) de 6,10%. Em uma avaliação geral, não levando em consideração o CS e a aplicação móvel, o desempenho da AETool foi melhor que dos humanos.

A AETool julgou haver 340 violações de acessibilidade somando as violações encontradas em todas as quatro aplicações. Dentre as 340, 303 são violações reais, ou seja, 89,11%. Esse fator é importante de considerar, pois mostra a precisão da ferramenta em encontrar violações de acessibilidade reais.

A AETool encontrou problemas de acessibilidade, os quais não foram percebidos ou considerados como problemas pelos humanos, pois a ferramenta encontrou 56,53% dos problemas de acessibilidade existentes nas quatro aplicações móveis, enquanto os humanos encontraram em média 33,33%. Como exemplo, a AETool conseguiu identificar todas as violações de acessibilidade das aplicações móveis referentes aos CSs 1.1.1, 1.3.1, 1.3.4, 2.4.2, 2.4.4, 2.4.9 e 2.4.10, diferentemente dos humanos. Por fim, independente da aplicação móvel e do CS, a AETool mostrou bons desempenhos na identificação de violações de acessibilidade, sendo assim, uma aliada no processo de identificar violações de acessibilidade, podendo assim, ajudar desenvolvedores de aplicações móveis a resolver tais violações.

Em relação a outras ferramentas de avaliação de acessibilidade, a AETool avalia mais CSs se comparado com o *scanner* de acessibilidade da Google e com o *Accessibility Insights* e consegue avaliar todas interfaces de usuário de uma só vez.

A AETool teve um baixo desempenho em relação a CSs referentes a contraste, o que pode ser uma necessidade de análise de um futuro processo de mudança na lógica aplicada na avaliação dos CSs 1.4.3, 1.4.6 e 1.4.11.

Conclusão

Neste trabalho, o objetivo foi auxiliar a avaliação de acessibilidade de interfaces de usuário de aplicações móveis, por meio de uma ferramenta computacional que avalie automaticamente a acessibilidade de interfaces de usuário de aplicações móveis, seguindo as diretrizes da WCAG 2.1. A ferramenta de avaliação de acessibilidade proposta, denominada AETool, avalia a acessibilidade de aplicações móveis nativas para Android, por meio da avaliação dos arquivos *XML* das interfaces de usuário.

A AETool avalia o código-fonte de cada interface de usuário, de acordo com os CSs descritos na Subseção 3.1.2 desta dissertação. Ao final da avaliação, a AETool retorna os problemas de acessibilidade encontrado(s) na(s) interface(s) de usuário, identificando: (i) a interface de usuário, (ii) o identificador (*id*) do componente da interface de usuário (caso haja), (iii) o CS violado, juntamente com seu nível de conformidade (A, AA, AAA), (iv) a descrição do problema de acessibilidade, (v) o trecho do código-fonte, no qual o problema de acessibilidade foi encontrado e (vi) o *link* redirecionando para a descrição do CS na documentação da WCAG 2.1, como pode ser observado na Figura 7 presente na Seção 3.3 desta dissertação.

Para avaliar a eficácia da AETool, foi realizado um comparativo entre os resultados obtidos pela AETool na avaliação de acessibilidade de quatro aplicações (SofiaFala - Fonoaudiólogos, SofiaFala - Criança, Sound Recorder e Minimal To Do) em relação aos resultados das avaliações de desenvolvedores de *software*.

Por meio do comparativo realizado, percebeu-se as seguintes vantagens de utilizar a AETool para avaliar acessibilidade de aplicações móveis: (i) a rapidez em encontrar os problemas de acessibilidade, se comparado a um especialista humano; (ii) a identificação da parte do código-fonte, na qual encontra-se o problema de acessibilidade; e (iii) a utilidade da ferramenta em avaliar a acessibilidade de aplicações móveis durante o desenvolvimento de uma aplicação móvel. Nesse sentido, a AETool pode ajudar os desenvolvedores de *software* a encontrarem os problemas de acessibilidade dentro da aplicação com mais rapidez do que caso fosse realizar uma avaliação manual. Além de que, com a avaliação da

AETool, os desenvolvedores de *software* podem corrigir problemas de acessibilidade para trazer mais acessibilidades ao público com deficiências.

Para um desenvolvedor de *software*, a avaliação de código-fonte, de uma interface de usuário de uma aplicação móvel nativa para Android, exige conhecimento do mesmo em relação a linguagem de marcação *XML*, além de ser um processo exaustivo de ter que avaliar cada componente no código-fonte. Além disso, alguns problemas de acessibilidade podem não ser perceptíveis sem avaliação do código-fonte, como, por exemplo, um componente que possui altura e largura adaptáveis, mas que não possui uma marcação para definir um tamanho mínimo. Nesse contexto, o componente pode estar acessível para um determinado tamanho de tela de dispositivo móvel, porém, pode não estar acessível em outra configuração de tela, sendo uma violação ao CS.2.5.5.

A AETool encontrou problemas de acessibilidade, os quais não foram percebidos ou considerados como problemas pelos humanos, pois a ferramenta encontrou 56,53% dos problemas de acessibilidade existentes nas quatro aplicações móveis, enquanto os desenvolvedores de *software* encontraram em média 33,33%. Como exemplo, a AETool conseguiu identificar todas as violações de acessibilidade das aplicações móveis referentes aos CSs 1.1.1, 1.3.1, 1.3.4, 2.4.2, 2.4.4, 2.4.9 e 2.4.10, diferentemente dos desenvolvedores de *software*. Por fim, independente da aplicação móvel e do CS, a AETool mostrou bons desempenhos na identificação de violações de acessibilidade, sendo assim, uma aliada no processo de identificar violações de acessibilidade, podendo assim, ajudar desenvolvedores de aplicações móveis a resolver tais violações.

Em relação a outras ferramentas de avaliação de acessibilidade, a AETool avalia mais CSs se comparado com o *scanner* de acessibilidade da Google e com o *Accessibility Insights* e consegue avaliar todas interfaces de usuário de uma só vez. Os códigos fonte da AETool estão disponibilizados para contribuições de código e uso nos seguintes repositórios: (i) API AETool: <https://github.com/vinisouzax/accessibility-ME-api>; (ii) Interface de Usuário da AETool: <https://github.com/vinisouzax/aceessibilidade-ME-front>.

6.1 Contribuições

Esse trabalho foi realizado no contexto de avaliação de acessibilidade, o qual atualmente se baseia em respostas a *checklists* de diretrizes. Portanto, automatizar esse processo pode contribuir com desenvolvedores de *software* em diferentes etapas do processo de desenvolvimento. Na concepção, a AETool pode ser útil para prototipar interfaces de usuários acessíveis. Na codificação, a AETool pode auxiliar no processo de criação de interfaces de usuários acessíveis, provendo *feedbacks* ao desenvolvedor de problemas de acessibilidade existentes.

Além da questão de desenvolvimento, a AETool promove um contexto educativo dos desenvolvedores para mostrar como tratar acessibilidade de aplicações móveis. Além da AET e das questões educativas e de suporte ao desenvolvimento de aplicações, este trabalho pode contribuir com a análise realizada na Seção 3.1, na qual foi levantado os CSs que seriam implementados e o que cada CS pode contribuir para pessoas com deficiências. Uma análise de todos os CSs da WCAG 2.1 foi realizada a fim de levantar quais CSs seriam passíveis de serem implementados desde que não fossem impossibilitados pelas seguintes condições estipuladas:

- **Condição 1:** O elemento de interface de usuário que é avaliado no CS é uma mídia de áudio/vídeo ou arquivos.
- **Condição 2:** Não há recursos (tag, marcações e etc) para avaliação do CS via código-fonte da interface de usuário em formato *XML* de uma aplicação nativa para Android.
- **Condição 3:** É necessário uma intervenção, avaliação e interpretação técnica por parte do desenvolvedor de aplicações móveis para avaliar se a aplicação está seguindo as recomendações do CS. Necessita-se do desenvolvedor de aplicações móveis para avaliar as regras de negócio da aplicação, podendo ser necessário o tratamento das recomendações de acessibilidade via programação em Java/Kotlin.
- **Condição 4:** Necessita-se de avaliar a integração da aplicação com *hardware/software* externo.

6.2 Limitações

A AETool não dispensa a avaliação de um especialista humano, primeiramente, porque a AETool não avalia as recomendações de todos os CSs da WCAG 2.1 e porque pode acontecer do problema de acessibilidade não ser captado, via avaliação de código-fonte *XML*, mas perceptível pelo especialista humano. Como, por exemplo, no caso das violações referentes aos CSs 1.4.3, 1.4.6 e 1.4.8, as quais os desenvolvedores de *software* conseguiram um melhor desempenho na identificação, se comparado a AETool. Além disso, nesta dissertação, abordou-se apenas a avaliação de interfaces de usuários codificadas em *XML* de aplicações móveis nativas para Android. Interfaces desenvolvidas com outras tecnologias não são contemplados nesta versão da ferramenta.

6.3 Trabalhos Futuros

Pode-se expandir a avaliação de acessibilidade da AETool para interfaces de usuários codificadas em outras linguagens de marcação, além de que se pode expandir para avaliação de interfaces de usuários de aplicações híbridas e de aplicações nativas para iOS. Além disso, pode-se ser trabalhado em melhorar os resultados da AETool em relação aos CSs 1.4.3, 1.4.6 e 1.4.11, o que traria um ganho para a AETool. Uma metodologia de avaliação de acessibilidade híbrida utilizando a AETool pode ser criada, a fim de abordar outros CSs, os quais necessitam de intervenção de um especialista humana, podendo assim, alimentar uma possível base de dados da AETool.

Ademais, a análise realizada na Seção 3.1, na qual foi levantado os CSs que seriam implementados e o que cada CS pode contribuir para pessoas com deficiências, pode ser utilizada para criar um guia de acessibilidade para desenvolvedores de *software*.

Por fim, outro trabalho futuro seria em relação a avaliar a utilidade da AETool em um ambiente real de trabalho por meio do Modelo de Aceitação de Tecnologia (TAM), o qual possui dois fatores principais que influenciam a intenção de um indivíduo de usar uma nova tecnologia: facilidade de uso percebida e utilidade percebida (DAVIS; BAGOZZI; WARSHAW, 1989).

Referências

A11Y. *Color Contrast Accessibility Validator*. 2019. Disponível em: <<https://color.a11y.com/?wc3>>. Acesso em: 06 jan. 2020.

ABDUL AZIZ, N. S.; WAN AHMAD, W. F.; BINTI ZULKIFLI, N. J. User experience on numerical application between children with down syndrome and autism. *Proceedings of the International HCI and UX Conference in Indonesia, ACM*, p. 23–31, 2015.

ACCESSIBILITYINSIGHTS. *Accessibility Insights*. 2021. Disponível em: <<https://accessibilityinsights.io/>>. Acesso em: 04 out. 2021.

ACCESSMONITOR. *AccessMonitor*. 2021. Disponível em: <<http://www.acessibilidade.gov.pt/accessmonitor/>>. Acesso em: 19 mar. 2021.

ALFREDO, M. G. et al. Analyzing learnability of common mobile gestures used by down syndrome users. *Proceedings of the XVI International Conference on Human Computer Interaction, ACM*, p. 1, 2015.

ANDROID. *Em operação com o material design*. 2018. Disponível em: <<https://developer.android.com/design/index.html>>. Acesso em: 06 jan. 2020.

ANDROID. *Ajuda de Acessibilidade no Android*. 2020. Disponível em: <<https://support.google.com/accessibility/android/answer/7158390>>. Acesso em: 09 set. 2020.

ANDROID. *Visão geral do manifesto do aplicativo*. 2020. Disponível em: <<https://developer.android.com/guide/topics/manifest/manifest-intro>>. Acesso em: 02 nov. 2020.

ANGKANANON, K.; WALD, M.; GILBERT, L. Technology enhanced interaction framework and method for accessibility in thai museums. *Information and Communication Technology (ICoICT), 2015 3rd International Conference on. IEEE*, p. 316–321, 2015.

APPLE. *Acessibilidade*. 2018. Disponível em: <<https://www.apple.com/br/accessibility/iphone/>>. Acesso em: 06 jan. 2020.

BARUA, A.; MANI, D.; MUKHERJEE, R. Measuring the business impacts of effective data. *Report accessed at <http://www.sybase.com/files/WhitepapersonSep>, v. 15, p. 2012, 2012.*

BILLI, M. et al. A unified methodology for the evaluation of accessibility and usability of mobile applications. *Universal Access in the Information Society*, v. 9, n. 4, p. 337–356, 2010.

BLANKENHAGEL, K. J. Identifying usability challenges of ehealth applications for people with mental disorders: Errors and design recommendations. *Proceedings of the 13th EAI International Conference on Pervasive Computing Technologies for Healthcare, ACM*, p. 91–100, 2019.

BOTTINO, R. M. et al. Cloud-mobile assistive technologies for people with intellectual impairments: A microsoft azure-based solution. *Proceedings of the 5th International Conference on Digital Health 2015, ACM*, p. 103–104, 2015.

BRAJNIK, G. A comparative test of web accessibility evaluation methods. *Proceedings of the 10th international ACM SIGACCESS Conference on Computers and Accessibility*, p. 113–120, 2008.

BRAJNIK, G.; YESILADA, Y.; HARPER, S. Testability and validity of wcag 2.0: the expertise effect. *Proceedings of the 12th international ACM SIGACCESS conference on Computers and accessibility*, p. 43–50, 2010.

BRASIL. *LEI Nº 13.146, DE 6 DE JULHO DE 2015*. 2015. Disponível em: <http://www.planalto.gov.br/ccivil_03/_Ato2015-2018/2015/Lei/L13146.htm#art112>. Acesso em: 06 jan. 2020.

BUDIU, R. Mobile user experience: limitations and strengths. *Nielsen Norman Group*, v. 19, p. 5, 2015.

CALVO, R.; SEYEDARABI, F.; SAVVA, A. Beyond web content accessibility guidelines: expert accessibility reviews. *Proceedings of the 7th international conference on software development and technologies for enhancing accessibility and fighting info-exclusion, ACM*, p. 77–84, 2016.

CARVALHO, L. P.; FREIRE, A. P. Native or web-hybrid apps?: An analysis of the adequacy for accessibility of android interface components used with screen readers. *Proceedings of the 16th Brazilian Symposium on Human Factors in Computing Systems*, p. 362–371, 2017.

CHOO, K. T. W.; BALAN, R. K.; LEE, Y. Examining augmented virtuality impairment simulation for mobile app accessibility design. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, ACM*, p. 375, 2019.

DAVIS, F.; BAGOZZI, R.; WARSHAW, R. User acceptance of computer technology: A comparison of two theoretical models. *Management Science*, v. 35, p. 982–1003, 1989.

EMAG. *eMAG - Modelo de Acessibilidade em Governo Eletrônico*. [S.l.], 2018. Disponível em: <<http://emag.governoeletronico.gov.br>>. Acesso em: 06 jan. 2020.

FRANKLIN, A.; MYNENI, S. Engagement and design barriers of mhealth applications for older adults. *Proceedings of the Technology, Mind, and Society, ACM*, p. 9, 2018.

GOOGLEPLAY. *Scanner de acessibilidade*. 2019. Disponível em: <https://play.google.com/store/apps/details?id=com.google.android.apps.accessibility.auditor&hl=pt_BR>. Acesso em: 06 jan. 2020.

HARPER, S.; THIESSEN, P.; YESILADA, Y. *Research Report on Mobile Web Accessibility*. [S.l.], 2018. Disponível em: <<https://www.w3.org/WAI/RD/2012/mobile/note/ED-mobile>>. Acesso em: 06 jan. 2020.

HELATHY, C. *What are Learning Disabilities?* 2015. Disponível em: <<https://www.healthychildren.org/English/health-issues/conditions/learning-disabilities/Pages/What-are-Learning-Disabilities.aspx>>. Acesso em: 31 mai. 2019.

ISO/TR-9241-210. *Ergonomics of human-system interaction - Part 100: Introduction to standards related to software ergonomics*. [S.l.], 2010. Disponível em: <<https://www.iso.org/obp/ui/#iso:std:iso:9241:-210:ed-1:v1:en>>. Acesso em: 06 jan. 2020.

KARDYS, P. A new android application for blind and visually impaired people. *IEEE*, p. 152–155, 2016.

KRAINZ, E.; FEINER, J.; FRUHMANN, M. Accelerated development for accessible apps—model driven development of transportation apps for visually impaired people. *Human-Centered and Error-Resilient Systems Development*. Springer, Cham, p. 374–381, 2016.

LYON, G. R. Learning disabilities. *The future of children*, p. 54–76, 1996.

MEIRELLES, F. S. *29ª Pesquisa Anual sobre Administração e Uso da TI nas Empresas*. 2018. Disponível em: <http://www.tvclassificados.com/uploads/docs/doc_619_1_FGV%20Smartphone_1530837863.pdf>. Acesso em: 24 mai. 2021.

NIELSEN, J. *Usability 101: Introduction to Usability*. [S.l.], 2012. Disponível em: <<https://www.nngroup.com/articles/usability-101-introduction-to-usability/>>. Acesso em: 06 jan. 2020.

OLIVEIRA, A. F. B. A. *Ferramenta para análise estática de código visando acessibilidade para deficiência visual em aplicações nativas Android*. Dissertação (Mestrado) — Instituto de Pesquisas Tecnológicas do Estado de São Paulo, São Paulo, 2019.

PARASURAMAN, S. et al. Smartphone usage and increased risk of mobile phone addiction: A concurrent study. *International journal of pharmaceutical investigation*, v. 7, n. 3, p. 125–131, 2017.

PATIL, N.; BHOLE, D.; SHETE, P. Enhanced ui automator viewer with improved android accessibility evaluation features. *International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*, IEEE, p. 977–983, 2016.

SAMONTE, M. J. C. et al. Assistive gamification and speech recognition e-tutor system for speech impaired students. *Proceedings of the 2nd International Conference on Image and Graphics Processing, ACM*, v. 6, p. 37–41, 2019.

SANTANA, V. F.; ALMEIDA, L. D.; MARIA, C. B. *Websites Atendendo a Requisitos de Acessibilidade e Usabilidade*. [S.l.]: Leanpub, 2018.

SEKHON, A. *Minimal To Do*. 2021. Disponível em: <https://play.google.com/store/apps/details?id=com.avjindersinghsekhon.minimaltodo&hl=pt_BR&gl=US>. Acesso em: 5 jun. 2021.

SIEBRA, C. et al. Toward accessibility with usability: understanding the requirements of impaired users in the mobile context. *Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication, ACM*, p. 6, 2017.

- SILVA, C. F. D.; FERREIRA, S. B. L.; SACRAMENTO, C. Mobile application accessibility in the context of visually impaired users. *Proceedings of the 17th Brazilian Symposium on Human Factors in Computing Systems, ACM*, p. 32, 2018.
- SOFIAFALA. *SofiaFala*. 2019. Disponível em: <<http://dcm.ffclrp.usp.br/sofiafala/>>. Acesso em: 06 jan. 2020.
- SOUNDRECORDER. *SoundRecorder*. 2017. Disponível em: <<https://github.com/dkim0419/SoundRecorder>>. Acesso em: 5 jun. 2021.
- SUTTON, J.; FISCHER, L. M. Understanding visual risk communication messages: an analysis of visual attention allocation and think-aloud responses to tornado graphics. *Computer Science Review*, v. 27, n. 1, p. 173–188, 2021.
- TAN, C. T. et al. Retrogaming as visual feedback for speech therapy. *SIGGRAPH Asia 2014 Mobile Graphics and Interactive Applications, ACM*, 2014.
- W3C. *Relationship between Mobile Web Best Practices (MWBP) and Web Content Accessibility Guidelines (WCAG)*. 2009. Disponível em: <<https://www.w3.org/TR/mwbp-wcag/>>. Acesso em: 06 jan. 2020.
- W3C. *Sobre*. 2011. Disponível em: <<http://www.w3c.br/Sobre/>>. Acesso em: 06 jan. 2020.
- WCAG2.1. *Web Content Accessibility Guidelines (WCAG) 2.1*. [S.l.]: WWW Consortium (W3C), 2018.
- WEBAIM. *WebAim - web accessibility in mind*. 2015. Disponível em: <<https://webaim.org/>>. Acesso em: 5 jun. 2021.
- ZEIN, S.; SALLEH, N.; GRUNDY, J. A systematic mapping study of mobile application testing techniques. *Journal of Systems and Software*, v. 117, p. 334–356, 2016.
- ZOUHAIER, L.; HLAOUI, Y.; AYED, L. Users interfaces adaptation for visually impaired users based on meta-model transformation. *Computer Software and Applications Conference (COMPSAC), 2017 IEEE 41st Annual*, p. 881–886, 2017.



Checklist para avaliação de aplicações móveis utilizando as WCAG 2.1

Checklist para avaliação de aplicações móveis utilizando a WCAG 2.1

Este formulário pretende apoiar a avaliação de acessibilidade de uma aplicação móvel de acordo com os seguintes Critérios de Sucesso (CS) da WCAG 2.1:

- 1.1.1 Conteúdo não textual
- 1.3.1 Informações e relacionamentos
- 1.3.4 Orientação
- 1.3.5 Identificar finalidade de entrada
- 1.4.3 Contraste (mínimo)
- 1.4.6 Contraste (aprimorado)
- 1.4.8 Apresentação visual
- 1.4.11 Contraste sem texto
- 2.4.2 Título da página
- 2.4.4 Objetivo do Link (no Contexto)
- 2.4.9 Objetivo do Link (Apenas Link)
- 2.4.10 Títulos de Seção
- 2.5.1 Gestos de ponteiro
- 2.5.5 Tamanho do alvo
- 3.3.5 Ajuda
- 4.1.1 Análise (Não necessário de ser avaliado, pois as próprias ferramentas para desenvolvimento para Android nativo já fazem a tratativa desse critério de sucesso)

Observações:

1 - Este formulário foi estruturado em formato de checklist, no qual o avaliador marca uma caixa de checagem, caso a aplicação móvel esteja atendendo determinada recomendação do Critério de Sucesso (CS) ou deixa sem marcar a caixa de checagem, caso a aplicação não esteja atendendo a recomendação do CS. Além disso, para cada CS abordado no checklist, a quantidade de irregularidades de acessibilidade encontradas em relação a cada CS é perguntada.

2 - O formulário possui 8 seções, as seções de 2 a 8 fazem parte do processo de avaliação da aplicação móvel. O avaliador ao fim de cada seção, possui um espaço para descrever problemas, críticas, recomendações e sugestões, dentro do contexto dos CSs abordados na presente seção, referentes a aplicação móvel.

3 - Este checklist faz parte do trabalho "Avaliação automática de acessibilidade em interfaces de usuário de aplicações móveis" desenvolvido pelos autores(as): Vinícius de Souza Gonçalves (Mestrando em Computação Aplicada pela Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto) e Alessandra Alaniz Macedo (Docente do Departamento de Computação e Matemática da Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto)

4 - Só foram abordados neste checklist os CSs citados anteriormente, pois o intuito deste checklist é realizar um comparativo entre uma ferramenta desenvolvida para avaliação automatizada de acessibilidade em interfaces de usuário de aplicações móveis, a qual usa como critério de avaliação, os CSs citados anteriormente.

Aplicativos:

MinimalTodo:

Link Download: https://drive.google.com/file/d/1VExgDeJ_TpOZQqCh3gAjMpHkokIHJkrV/view?usp=sharing

Instruções:

Nenhuma

SoundRecorder:

Link Download: <https://drive.google.com/file/d/1wDRqF2ZHTGBLEaIMW6rFtFq9SfLySjH/view?usp=sharing>

Instruções:

Nenhuma

SofiaFala - Criança:

Link Download: <https://drive.google.com/file/d/1dgDGfUwI8FLBr6PVD88MhNU-Xj-0oNO/view?usp=sharing>

Instruções:

Pode usar o usuário de teste:

E-mail: vgondev@gmail.com

Senha: adm123

SofiaFala - Fonoaudiólogos:

Link Download: https://drive.google.com/file/d/1Gxk4XkSk65dJKY7DnH8HxODod297ak1_/view?usp=sharing

Pode cadastrar um usuário ou usar o usuário de teste:

E-mail: ygondev@gmail.com

Senha: adm123

***Obrigatório**

1. Qual a aplicação móvel que está sendo avaliada? *

Marcar apenas uma oval.

SofiaFala - Fonoaudiólogos

SofiaFala - Criança

Sound Recorder

Minimal Todo

2. Por favor, digite as iniciais do seu nome completo (Ex. José de Souza Silva = JSS) *

Diretriz 1.1 -
Textos
Alternativos

"Forneça alternativas de texto para qualquer conteúdo não textual" (WCAG 2.1)

CS 1.1.1 Conteúdo não textual: A intenção deste Critério de Sucesso é tornar as informações transmitidas por conteúdo não textual acessíveis por meio do uso de uma alternativa em texto. As alternativas de texto são uma forma primária de tornar as informações acessíveis, porque podem ser processadas por meio de qualquer modalidade sensorial (por exemplo, visual, auditiva ou tátil) para atender às necessidades do usuário (WCAG 2.1). Ex., Fornecer alternativas de texto para descrever imagens (Logo da empresa), campos de entrada de texto (Digite seu e-mail), botões (Salvar) e dentre outros componentes de interface de usuário

3. 1.1.1 Conteúdo não textual

Marque todas que se aplicam.

Imagens, botões com imagem de formulário e pontos de acesso de mapa de imagem têm texto alternativo equivalente e apropriado.

As imagens que não transmitem conteúdo, são decorativas ou contêm conteúdo já transmitido em texto e recebem um texto alternativo vazio (contentDescription = ""). Todas as imagens vinculadas têm um texto alternativo descritivo.

Alternativas de texto para imagens complexas são fornecidas no contexto ou em uma página vinculada.

Os botões de formulário têm um valor descritivo.

As entradas de texto de formulário possuem rótulos de texto associados.

4. Quantidade de erros encontrados para CS.1.1.1 Conteúdo não textual

5. Caso tenha considerações em relação à acessibilidade da aplicação móvel avaliada levando em consideração o(s) CS(s) dessa seção, por favor, descreva-as

Diretriz
1.3 -
Adaptável

"Crie conteúdos que podem ser apresentados de diferentes maneiras (por exemplo, layout mais simples) sem perder informações ou estrutura" (WCAG 2.1)

CS 1.3.1 Informações e relacionamentos: A intenção deste Critério de Sucesso é garantir que as informações e as relações implícitas na formatação visual ou auditiva sejam preservadas quando o formato da apresentação for alterado. Por exemplo, o formato da apresentação muda quando o conteúdo é lido por um leitor de tela ou quando uma folha de estilo do usuário é substituída pela folha de estilo fornecida pelo autor (WCAG 2.1).

CS 1.3.4 Orientação: A intenção deste Critério de Sucesso é garantir que o conteúdo seja exibido na orientação (retrato ou paisagem) preferida pelo usuário. Alguns aplicativos definem e restringem automaticamente a tela a uma orientação de exibição específica e esperam que os usuários respondam girando o dispositivo para corresponder, mas isso pode criar problemas (WCAG 2.1).

CS 1.3.5 Identificar a finalidade de entrada: A intenção deste Critério de Sucesso é garantir que a finalidade de uma entrada de formulário que coleta informações sobre o usuário possa ser determinada de forma programática, para que os agentes do usuário possam extrair e apresentar essa finalidade aos usuários usando diferentes modalidades. A capacidade de declarar programaticamente o tipo específico de dados esperado em um determinado campo torna o preenchimento de formulários mais fácil, especialmente para pessoas com deficiências cognitivas (WCAG 2.1). Ex., Um campo para o usuário digitar um número de telefone possui uma máscara de como deve ser o formato, a fim de auxiliar o preenchimento.

6. 1.3.1 Informações e relacionamentos

Marque todas que se aplicam.

Rótulos de texto são associados a elementos de entrada de texto de formulário. Elementos de formulário relacionados são agrupados com conjunto de campos/legenda.

7. Quantidade de erros encontrados para CS.1.3.1 Informações e relacionamentos

8. 1.3.4 Orientação

Marque todas que se aplicam.

A orientação do conteúdo da interface de usuário não se restringe apenas a retrato ou paisagem, a menos que uma orientação específica seja necessária.

9. Quantidade de erros encontrados para CS.1.3.4 Orientação

10. 1.3.5 Identificar a finalidade da entrada

Marque todas que se aplicam.

Os campos de entrada de texto que coletam certos tipos de informações do usuário têm um atributo de auxílio de preenchimento apropriado definido.

11. Quantidade de erros encontrados para CS.1.3.5 Identificar a finalidade da entrada

12. Caso tenha considerações em relação à acessibilidade da aplicação móvel avaliada levando em consideração o(s) CS(s) dessa seção, por favor, descreva-as

Diretriz 1.4

-

Distinguível

"Facilite para os usuários ver e ouvir o conteúdo, incluindo a separação do primeiro plano do segundo plano" (WCAG 2.1)

CS 1.4.3 Contraste (mínimo): A intenção deste Critério de Sucesso é fornecer contraste suficiente entre o texto e o fundo para que possa ser lido por pessoas com visão moderadamente baixa (que não usam tecnologia de assistência de aumento de contraste) (WCAG, 2.1).

CS 1.4.6 Contraste (aprimorado): A intenção deste Critério de Sucesso é fornecer contraste suficiente entre o texto e o fundo para que possa ser lido por pessoas com visão moderadamente baixa (que não usam tecnologia de assistência de aumento de contraste) (WCAG, 2.1).

CS 1.4.8 Apresentação visual: A intenção deste Critério de Sucesso é garantir que o texto renderizado visualmente seja apresentado de forma que possa ser percebido sem que o layout interfira na legibilidade. Pessoas com algumas dificuldades cognitivas, de linguagem e de aprendizagem e alguns usuários de baixa visão não conseguem perceber o texto e/ou perdem seu lugar de leitura se o texto for apresentado de uma maneira que seja difícil para eles lerem (WCAG, 2.1).

CS 1.4.11 Contraste sem texto: A intenção deste Critério de Sucesso é garantir que os componentes ativos da interface do usuário (ou seja, controles) e gráficos significativos sejam distinguíveis por pessoas com visão moderadamente baixa (WCAG, 2.1).

13. 1.4.3 Contraste (mínimo)

Marque todas que se aplicam.

Texto e imagens de texto têm uma taxa de contraste mínima que permite a visualização do conteúdo.

14. Quantidade de erros encontrados para CS.1.4.3 Contraste (mínimo)

15. 1.4.6 Contraste (aprimorado)

Marque todas que se aplicam.

- Texto e imagens de texto têm uma taxa de contraste que permite a visualização do conteúdo sem problemas.

16. Quantidade de erros encontrados para CS.1.4.6 Contraste (aprimorado)

17. 1.4.8 Apresentação Visual

Marque todas que se aplicam.

- Blocos de texto em uma frase de comprimento não estão totalmente justificados (alinhados às margens esquerda e direita).

- Blocos de texto em uma frase de comprimento têm espaçamento de linha adequado.

18. Quantidade de erros encontrados para CS.1.4.8 Apresentação Visual

19. 1.4.11 Contraste sem texto

Marque todas que se aplicam.

- Há um contraste suficiente que permite diferenciar objetos gráficos (como ícones e componentes de tabelas ou gráficos) e componentes de interface personalizados pelo autor (como botões, controles de formulário e indicadores/contornos de foco).

- Um contraste suficiente deve ser fornecido nos vários estados (foco, foco, ativo, etc.) de componentes interativos personalizados pelo autor.

20. Quantidade de erros encontrados para CS.1.4.11 Contraste sem texto

21. Caso tenha considerações em relação à acessibilidade da aplicação móvel avaliada levando em consideração o(s) CS(s) dessa seção, por favor, descreva-as

Diretriz
2.4 -
Navegável

"Forneça maneiras de ajudar os usuários a navegar, encontrar conteúdo e determinar onde estão" (WCAG 2.1)

CS 2.4.2 Título da página: A intenção deste Critério de Sucesso é ajudar os usuários a encontrar conteúdo e se orientar nele, garantindo que cada página da Web tenha um título descritivo. Os títulos identificam a localização atual sem exigir que os usuários leiam ou interpretem o conteúdo da página (WCAG, 2.1).

CS 2.4.4 Objetivo do link (no contexto): A intenção deste Critério de Sucesso é ajudar os usuários a entender a finalidade de cada link para que possam decidir se desejam seguir o link. Sempre que possível, forneça um texto de link que identifique a finalidade do link sem a necessidade de contexto adicional (WCAG, 2.1).

CS 2.4.9 Objetivo do link (apenas link): A intenção deste Critério de Sucesso é ajudar os usuários a compreender a finalidade de cada link no conteúdo, para que possam decidir se desejam segui-lo. A melhor prática é que links com o mesmo destino tenham as mesmas descrições, mas links com finalidades e destinos diferentes tenham descrições diferentes (WCAG, 2.1).

CS 2.4.10 Títulos de Seção: A intenção deste Critério de Sucesso é fornecer títulos para as seções de uma página da Web, quando a página está organizada em seções (WCAG, 2.1).

22. 2.4.2 Título da página

Marque todas que se aplicam.

A página possui um título de página descritivo e informativo.

23. Quantidade de erros encontrados para CS.2.4.2 Título da página

24. 2.4.4 Objetivo do link (no contexto)

Marque todas que se aplicam.

A página possui um título de página descritivo e informativo.

25. Quantidade de erros encontrados para CS.2.4.4 Objetivo do link (no contexto)

26. 2.4.9 Objetivo do link (apenas link)

Marque todas que se aplicam.

A página possui um título de página descritivo e informativo.

27. Quantidade de erros encontrados para CS.2.4.9 Objetivo do link (apenas link)

28. 2.4.10 Títulos de Seção

Marque todas que se aplicam.

A página possui um título de página descritivo e informativo.

29. Quantidade de erros encontrados para CS.2.4.10 Títulos de Seção

30. Caso tenha considerações em relação à acessibilidade da aplicação móvel avaliada levando em consideração o(s) CS(s) dessa seção, por favor, descreva-as

**Diretriz 2.5 -
Modalidades
de entrada**

"Torne mais fácil para os usuários operar a funcionalidade por meio de várias entradas além do teclado" (WCAG 2.1)

CS 2.5.1 Gestos de ponteiro: A intenção deste Critério de Sucesso é garantir que o conteúdo possa ser controlado com uma variedade de dispositivos apontadores, habilidades e tecnologias assistivas. Algumas pessoas não conseguem realizar gestos de maneira precisa ou podem usar um dispositivo de entrada especializado ou adaptado, como um apontador de cabeça, sistema de olhar fixo ou emulador de mouse controlado por fala. Alguns métodos de apontamento não têm capacidade ou precisão para realizar gestos multiponto ou baseados em caminho (WCAG, 2.1).

CS 2.5.5 Tamanho do alvo: A intenção deste critério de sucesso é garantir que os tamanhos de destino sejam grandes o suficiente para que os usuários os ativem facilmente, mesmo se o usuário estiver acessando o conteúdo em um pequeno dispositivo portátil com tela de toque, tiver destreza limitada ou tiver problemas para ativar pequenos destinos por outros motivos (WCAG, 2.1).

31. 2.5.1 Gestos de Ponteiro

Marque todas que se aplicam.

Se gestos multiponto ou baseados em caminho (como beliscar, deslizar ou arrastar pela tela) não forem essenciais para a funcionalidade, a funcionalidade também pode ser realizada com uma ativação de ponto único (como ativar um botão).

32. Quantidade de erros encontrados para CS.2.5.1 Gestos de Ponteiro

33. 2.5.5 Tamanho do alvo

Marque todas que se aplicam.

Os alvos clicáveis têm altura e largura adequadas, a menos que um alvo alternativo desse tamanho seja fornecido, o alvo seja embutido (como um link dentro de uma frase), o alvo não seja modificado pelo autor (como uma caixa de seleção padrão), ou o tamanho pequeno do alvo é essencial para a funcionalidade.

34. Quantidade de erros encontrados para CS.2.5.5 Tamanho do alvo

35. Caso tenha considerações em relação à acessibilidade da aplicação móvel avaliada levando em consideração o(s) CS(s) dessa seção, por favor, descreva-as

Diretriz 3.3

-

Assistência de entrada

"Ajude os usuários a evitar e corrigir erros" (WCAG 2.1)

CS 3.3.5 Ajuda: A intenção deste Critério de Sucesso é ajudar os usuários a evitar erros. Alguns usuários com deficiência podem ser mais propensos a cometer erros do que usuários sem deficiência. Usando a ajuda sensível ao contexto, os usuários descobrem como realizar uma operação sem perder o controle do que estão fazendo (WCAG, 2.1).

36. 3.3.5 Ajuda

Marque todas que se aplicam.

Instruções e dicas são fornecidas no contexto para ajudar no preenchimento e envio do formulário.

37. Quantidade de erros encontrados para CS.3.3.5 Ajuda

38. Caso tenha considerações em relação à acessibilidade da aplicação móvel avaliada levando em consideração o(s) CS(s) dessa seção, por favor, descreva-as

Este conteúdo não foi criado nem aprovado pelo Google.

Google Formulários

B

Planilha com dados da avaliação com avaliadores e com a AETool

Tabela 1 - Quantidade de violações de acessibilidade existentes e encontradas pelos avaliadores e pela ferramenta referentes à aplicação móvel Sound Recorder																							
CS	APP	Ouro	A1		A2		A3		A4		A5		A6		Ferramenta		A1 - FRI	A2 - FRI	A3 - FRI	A4 - FRI	A5 - FRI	A6 - FRI	Ferramenta - FRI
			O	H	I	H	I	H	I	H	I	H	I	F	I								
CS.1.1.1	SoundRecorder	4	3	3	3	3	3	3	3	3	3	3	3	3	4	4	0,75	0,75	0,75	0,75	0,75	0,75	1
CS.1.3.1	SoundRecorder	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
CS.1.3.4	SoundRecorder	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1
CS.1.3.5	SoundRecorder	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1
CS.1.4.3	SoundRecorder	10	4	4	6	6	6	6	6	6	4	4	6	6	1	1	0,4	0,6	0,6	0,6	0,4	0,6	0,1
CS.1.4.6	SoundRecorder	10	6	6	6	6	6	6	6	6	6	6	6	6	1	1	0,6	0,6	0,6	0,6	0,6	0,6	0,1
CS.1.4.8	SoundRecorder	15	3	3	2	2	2	5	5	5	3	3	2	2	15	15	0,2	0,33333333	0,33333333	0,33333333	0,2	0,33333333	0,1
CS.1.4.11	SoundRecorder	3	1	1	1	1	3	3	3	3	1	1	1	1	0	0	0,33333333	0,33333333	1	1	0,33333333	0,33333333	0
CS.2.4.2	SoundRecorder	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS.2.4.4	SoundRecorder	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS.2.4.9	SoundRecorder	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS.2.4.10	SoundRecorder	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS.2.5.1	SoundRecorder	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
CS.2.5.5	SoundRecorder	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1
CS.3.3.5	SoundRecorder	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Total		48	22	22	23	23	27	27	27	27	21	21	23	23	28	28							
FR		1	0,4486795918	0,4486795918	0,4693877551	0,4693877551	0,5510204082	0,5510204082	0,5510204082	0,5510204082	0,4285714286	0,4285714286	0,4693877551	0,4693877551	0,5714285714	0,5714285714							
																	Média FRI de A por App	Desvio Padrão de FRI de A por App					Coefficiente de Variação
																	0,4863846578	0,0522969536					10,75106145

Legenda
CS = Critério de Sucesso
A = Avaliador
I = Quantidade de violações encontradas que fazem parte do padrão ouro
H = Quantidade de violações encontradas pelos avaliadores
F = Quantidade de violações encontradas pela ferramenta (AETool)
FR = H / O ou I / O ou F / O (Normalização dos dados em Freqüência Relativa)
FRI = Freqüência relativa em relação a I
O = Quantidade de violações existentes na aplicação móvel (Padrão Ouro)

Tabela 2 - Quantidade de violações de acessibilidade existentes e encontradas pelos avaliadores e pela ferramenta referentes à aplicação móvel Minimal To do																							
CS	APP	Ouro	A1		A2		A3		A4		A5		A6		Ferramenta		A1 - FRI	A2 - FRI	A3 - FRI	A4 - FRI	A5 - FRI	A6 - FRI	Ferramenta - FRI
			O	H	I	H	I	H	I	H	I	H	I	F	I								
CS.1.1.1	Minimal To do	10	4	4	3	3	4	4	4	4	3	3	4	4	12	10	0,4	0,3	0,4	0,4	0,3	0,4	0,1
CS.1.3.1	Minimal To do	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0
CS.1.3.4	Minimal To do	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1
CS.1.3.5	Minimal To do	2	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0	0	0	0	0	0	0
CS.1.4.3	Minimal To do	9	7	7	7	3	3	8	8	8	3	3	8	8	0	0	0,77777778	0,33333333	0,88888889	0,88888889	0,33333333	0,88888889	0,1
CS.1.4.6	Minimal To do	10	7	7	6	6	8	8	8	8	6	6	8	8	1	1	0,7	0,6	0,8	0,8	0,6	0,8	0,1
CS.1.4.8	Minimal To do	14	0	0	0	0	0	0	0	0	0	0	0	0	14	14	0	0	0	0	0	0	1
CS.1.4.11	Minimal To do	2	2	2	2	2	2	2	2	2	2	2	2	2	0	0	1	1	1	1	1	1	1
CS.2.4.2	Minimal To do	2	1	1	1	1	1	1	1	1	1	1	1	1	2	2	0,5	0,5	0,5	0,5	0,5	0,5	0,1
CS.2.4.4	Minimal To do	2	1	1	1	1	1	1	1	1	1	1	1	1	2	2	0,5	0,5	0,5	0,5	0,5	0,5	0,1
CS.2.4.9	Minimal To do	2	1	1	1	1	1	1	1	1	1	1	1	1	2	2	0,5	0,5	0,5	0,5	0,5	0,5	0,1
CS.2.4.10	Minimal To do	2	1	1	1	1	1	1	1	1	1	1	1	1	2	2	0,5	0,5	0,5	0,5	0,5	0,5	0,1
CS.2.5.1	Minimal To do	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS.2.5.5	Minimal To do	7	3	3	4	4	2	2	2	2	4	4	2	2	10	7	0,4285714286	0,5714285714	0,2857142857	0,2857142857	0,5714285714	0,2857142857	0,1
CS.3.3.5	Minimal To do	2	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0	0	0	0	0	0	0
Total		66	29	29	24	24	30	30	30	30	24	24	30	30	53	46							
FR		1	0,4393939394	0,4393939394	0,3636363636	0,3636363636	0,4545454545	0,4545454545	0,4545454545	0,4545454545	0,3636363636	0,3636363636	0,4545454545	0,4545454545	0,803030303	0,698969897							
																	Média FRI de A por App	Desvio Padrão de FRI de A por App					Coefficiente de Variação
																	0,4217171717	0,04537029229					10,75846452

Legenda
CS = Critério de Sucesso
A = Avaliador
I = Quantidade de violações encontradas que fazem parte do padrão ouro
H = Quantidade de violações encontradas pelos avaliadores
F = Quantidade de violações encontradas pela ferramenta (AETool)
FR = H / O ou I / O ou F / O (Normalização dos dados em Freqüência Relativa)
FRI = Freqüência relativa em relação a I
O = Quantidade de violações existentes na aplicação móvel (Padrão Ouro)

Tabela 3 - Quantidade de violações de acessibilidade existentes e encontradas pelos avaliadores e pela ferramenta referentes à aplicação móvel SofaFala - Criança																							
CS	APP	Ouro	A1		A2		A3		A4		A5		A6		Ferramenta		A1 - FRI	A2 - FRI	A3 - FRI	A4 - FRI	A5 - FRI	A6 - FRI	Ferramenta - FRI
			O	H	I	H	I	H	I	H	I	H	I	F	I								
CS.1.1.1	Criança	14	9	9	9	9	9	8	8	9	9	9	9	9	15	14	0,6428571429	0,6428571429	0,5714285714	0,6428571429	0,6428571429	0,6428571429	0,1
CS.1.3.1	Criança	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
CS.1.3.4	Criança	9	6	6	3	3	6	6	6	3	3	6	6	9	9	0,666666667	0,3333333333	0,666666667	0,3333333333	0,3333333333	0,666666667	0,1	
CS.1.3.5	Criança	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS.1.4.3	Criança	8	3	3	1	1	3	3	1	1	1	1	3	3	2	2	0,375	0,125	0,375	0,125	0,125	0,375	0,25
CS.1.4.6	Criança	8	3	3	1	1	3	3	1	1	1	1	3	3	2	2	0,375	0,125	0,375	0,125	0,125	0,375	0,25
CS.1.4.8	Criança	18	0	0	0	0	0	0	0	0	0	0	0	0	11	11	0	0	0	0	0	0	0,6111111111
CS.1.4.11	Criança	8	1	0	3	2	1	0	3	2	3	2	1	0	0	0	0	0,25	0	0,25	0,25	0	0
CS.2.4.2	Criança	5	3	3	3	3	3	3	3	3	3	3	3	3	7	5	0,6	0,6	0,6	0,6	0,6	0,6	0,1
CS.2.4.4	Criança	5	3	3	3	3	3	3	3	3	3	3	3	3	7	5	0,6	0,6	0,6	0,6	0,6	0,6	0,1
CS.2.4.10	Criança	5	3	3	3	3	3	3	3	3	3	3	3	3	7	5	0,6	0,6	0,6	0,6	0,6	0,6	0,1
CS.2.5.1	Criança	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CS.2.5.5	Criança	13	6	4	4	4	4	4	4	4	4	4	6	4	13	13	0,3076923077	0,3076923077	0,3076923077	0,3076923077	0,3076923077	0,3076923077	0,1
CS.3.3.5	Criança	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
Total		101	40	37	33	32	37	36	33	32	33	32	40	37	83	74							
FR		1	0,396039604	0,3663366337	0,3267326733	0,3168316832	0,3663366337	0,3564356436	0,3267326733	0,3168316832	0,3267326733	0,3168316832	0,396039604	0,3663366337	0,8217621762	0,7326732673							
																	Média FRI de A por App	Desvio Padrão de FRI de A por App					Coefficiente de Variação
																	0,3399339934	0,02556424651					7,520356012

Legenda
CS = Critério de Sucesso
A = Avaliador
I = Quantidade de violações encontradas que fazem parte do padrão ouro
H = Quantidade de violações encontradas pelos avaliadores
F = Quantidade de violações encontradas pela ferramenta (AETool)
FR = H / O ou I / O ou F / O (Normalização dos dados em Freqüência Relativa)
FRI = Freqüência relativa em relação a I
O = Quantidade de violações existentes na aplicação móvel (Padrão Ouro)

Tabela 4 - Quantidade de violações de acessibilidade existentes e encontradas pelos avaliadores e pela ferramenta referentes à aplicação móvel SofaFala - Fonouduólogos																							
CS	APP	Ouro	A1		A2		A3		A4		A5		A6		Ferramenta		A1 - FRI	A2 - FRI	A3 - FRI	A4 - FRI	A5 - FRI	A6 - FRI	Ferramenta - FRI
			O	H	I	H	I	H	I	H	I	H	I	F	I								
CS.1.1.1	Fono	29	13	13	9	9	17	17	13	13	9	9	17	17	29	29	0,4482758621	0,3103448276	0,5862068966	0,4482758621	0,3103448276	0,5862068966	0,1
CS.1.3.1	Fono	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
CS.1.3.4	Fono	17	10	10	11	11	13	13	10	10	11	11	13	13	17	17	0,5882352941	0,6470588235	0,7647058824	0,5882352941	0,6470588235	0,7647058824	0

